Proceedings of the 26th Philippine Computing Science Congress Volume II: Short Papers

Editors: Dr. Judith J. Azcarraga and Mr. Lester Sityar

Copyright © 2026 Computing Society of the Philippines

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, without the prior written permission of the copyright owner. Individual papers may be uploaded to institutional repositories or other academic sites for self-archival purposes.

ISSN 1908-1146

Computing Society of the Philippines
csp.org.ph

April 23–25, 2026
University of the Immaculate Conception, Davao City

Contents

About the Philippine Computing Science Congress 1

About the Computing Society of the Philippines 1

Organizing Committee 2

Accepted Short Papers

Early Dengue Diagnosis using Machine Learning on Structured Clinical and Hematological Data 3

Queue Management System with Smart Feedback, Sentiment Analysis and Geolocation in Holy Child College of Davao 9

Comparative Study of MobileNetV2 and Hybrid MobileNetV2 with SVM Classifier for Efficient Waste Classification in Low-Resource Environments 15

A Swin-SNN Cross-Modal Attention Framework for Identifying Systemic Lupus Erythematosus from Ocular Fundus Imaging and Transcriptomic Interferon Signatures 20

Solving the Senior High School Timetabling Problem using a Multi-Agent Reinforcement Learning Approach 25

Text Mining Analysis of Review Bombing Incidents in Filipino Films 33

Judo Throw Recognition Using Computer Vision Techniques 39

FITPOSE: A COMPUTER VISION-BASED SYSTEM FOR REAL-TIME POSTURE ASSESSMENT IN HOME-BASED WORKOUT ENHANCEMENT 45

GaitAware: A Lightweight Hybrid Gait Analysis System Using BlazePose-BiLSTM and CNN for Identifying Potential Gait Abnormalities 52

DermaTector: Increasing Efficiency in Diagnostics of Sun-Induced Skin Diseases Utilizing EfficientNet 58

Augmenting a Force Directed Layout Algorithm in Drawing Adjacent Transposition Graphs 64

Analysis of Machine Learning Approaches for Real-Time Human Exercise Classification Using Pose Estimation Joint Angles 70

ATHENA: An Image Captioning Pipeline for Constrained Underwater Environments 76

Towards More Empathic Programming Environments - An Experimental Empathic AI-Enhanced IDE 82

Novel CUDA and Segment Tree-Accelerated Simulated Annealing Approaches to the Time-Dependent Knapsack Problem 90

Scrum vs. Extreme Programming (XP) vs Kanban in the Age of AI and DevOps: A Comparative Analysis of Cultural Resilience to Modern Development Challenges 97

Self-Training an Image Binary Classification Model using a Multi-Agent System 103

FIRE!: A Fire Simulation Game that Educates Filipinos on Fire Safety 109

Birdview: Human Crowd Density Visualization Tool from Video Feed to Floor Plan using Deep Learning and Homography 115

Neuro-Symbolic Structure-Aware Visual Parsing of Handwritten Boolean Logic using Swin-Transformer and mBART Decoders 123

ReHaPT: Integrating Machine Learning and Computer Vision for a Hand Rehabilitation Recognition System 131

Sign Lang: Developing a VR Game for FSL Learning 139

Technical Validation of an MDP-Based Adaptive Geometry Learning System via Q-Learning and LLM Hints 145

Evaluation of Target Encoding Methods Across Linear and Tree-Based Models for Cardiovascular Disease Prediction 151

Author Index 157

Institution Index 158

April 23–25, 2026
University of the Immaculate Conception, Davao City

About the Philippine Computing Science Congress

The **Philippine Computing Science Congress (PCSC)** is organized by the Computing Society of the Philippines to enable local and neighboring computing educators, researchers, information and communications technology (ICT) professionals, and students to interact and share their work in computing, computer science, computational science, and ICT. The conference features special lectures by prominent researchers and educators and contributed papers in ICT, computing, computer science, computational science, and related disciplines.

About the Computing Society of the Philippines

The **Computing Society of the Philippines (CSP)** is an organization promoting research and development in computing science, computer science, computer engineering, computational science, computing education, and related disciplines by promoting the exchange of knowledge in this field, especially at the tertiary and graduate levels. CSP's main objective is to advocate progressive policies and develop programs that affect the computing sector. CSP establishes linkages with other organizations in the pursuit of common goals.

April 23–25, 2026
University of the Immaculate Conception, Davao City

Organizing Committee

Program Chair

Judith J. Azcarraga, Ph.D.

Local Host Chair

Kristine Mae Adlaon, Ph.D.

Organizing Committee Members

Cherry Lyn Sta. Romana, MSCS, DIT
Kristine Mae Adlaon, Ph.D.
Tita Herradura
Judith J. Azcarraga, Ph.D.
Ramon Rodriguez
Cherry Lyn Sta. Romana, MSCS, DIT

Organizers

Computing Society of the Philippines

University of Immaculate Conception

Early Dengue Diagnosis Using Machine Learning on Structured Clinical and Hematological Data

Rani Jay M. Natavio
South Philippine Adventist College
Matanao, Davao del Sur, Philippines
nataviorani@gmail.com

Katelyn Leigh Guanzon
South Philippine Adventist College
Matanao, Davao del Sur, Philippines
leighguanzonyata@gmail.com

Neila M. Paglinawan-Muñez
South Philippine Adventist College
Matanao, Davao del Sur, Philippines
nmpmunez@gmail.com

ABSTRACT

Early and accurate diagnosis of dengue infection is critical for timely clinical management and prevention of severe complications. In resource-limited healthcare settings, confirmatory laboratory tests are often delayed or unavailable during early illness stages. This study investigates machine learning models to support early dengue diagnosis using routinely available clinical and hematological data as a practical decision-support tool for clinicians. The study utilizes the Structured Clinical and Hematological Dataset for Early Dengue Diagnosis in Bangladesh, containing patient-level clinical symptoms and laboratory measurements. Four supervised machine learning classifiers—Support Vector Machine (SVM), Random Forest, Extreme Gradient Boosting (XGBoost), and AdaBoost—were developed and evaluated. To improve generalizability, location-based attributes were excluded, allowing models to rely on universally applicable clinical and hematological features. The dataset was divided using stratified 70:30 train–test split, with class imbalance addressed through Synthetic Minority Oversampling Technique (SMOTE) applied to training data. Model performance was evaluated using accuracy, sensitivity, and specificity, emphasizing sensitivity due to its clinical importance in minimizing missed diagnoses. All models achieved high predictive performance, with AdaBoost demonstrating the strongest overall accuracy (99.35%) and sensitivity (99.52%), followed by SVM. Feature importance analysis identified platelet count and white blood cell count as the most influential predictors, consistent with established clinical understanding. These findings demonstrate that machine learning models can effectively support early dengue diagnosis at the patient level, assisting healthcare professionals in identifying suspected cases, prioritizing diagnostic testing, and improving clinical decision-making in settings where rapid laboratory confirmation is limited.

KEYWORDS

Machine Learning, Dengue Fever, Early Diagnosis

1 INTRODUCTION

Dengue fever remains a major public health concern in tropical and subtropical regions, with millions of infections reported annually worldwide. The disease poses significant challenges to healthcare systems due to its rapid progression and the difficulty of early and accurate diagnosis. Timely identification of dengue infection is critical for effective patient management, appropriate hospital admission, and the prevention of severe complications. However, current diagnostic practices in many clinical settings remain limited and may delay definitive confirmation and treatment decisions. In the Philippines, dengue cases surged by 73% in the first two months of 2025 compared to the previous year[12], underscoring the urgent need for improved early diagnostic approaches in endemic regions.

In routine clinical practice, the Complete Blood Count (CBC) test is commonly used as the primary diagnostic basis for dengue assessment. Clinicians estimate the likelihood of dengue infection using hematological indicators such as platelet count, white blood cell count, and hematocrit levels, and make admission or discharge decisions based on clinical judgment. [3] identified thrombocytopenia and abnormal white blood cell counts as significant indicators for dengue infection among patients presenting with acute febrile illness, highlighting the central role of CBC-derived features in early dengue assessment. Although CBC testing is accessible and cost-effective, this approach is inherently subjective and may lack the precision required for optimal patient triage, particularly during dengue outbreaks when healthcare resources are strained[4]. The reliance on CBC interpretation alone can lead to diagnostic uncertainty and delays in appropriate clinical intervention[12].

Recent advances in artificial intelligence have demonstrated the potential of machine learning (ML) approaches to enhance dengue diagnosis using routine clinical data.[6] conducted a systematic review of dengue prediction models and found that ML-based approaches consistently outperform traditional statistical methods in terms of accuracy and robustness. Similarly, [4] demonstrated that ML models achieved superior discrimination of confirmed dengue cases compared to conventional case-control models. [14] further showed that machine learning models trained exclusively on CBC parameters can accurately identify dengue infections at an early stage, supporting the feasibility of AI-assisted diagnostic tools in resource-limited clinical settings.

Several studies have specifically applied machine learning classifiers to early dengue diagnosis using hematological and clinical parameters. [1] demonstrated that efficient ML algorithms applied to routine clinical data could significantly improve early dengue disease prediction. [7] assessed dengue severity risk using demographic information and laboratory test results, demonstrating the feasibility of ML-based risk stratification in clinical settings. [11] reported promising classification results using CBC-related features with SVM achieving 85% prediction accuracy, emphasizing their suitability for low-resource clinical environments. [16] developed dengue prediction models specifically for Bangladesh using Decision Tree and Random Forest classifiers, achieving 79% accuracy with routine clinical data. [9] developed an 8-gene machine learning model that improved clinical prediction of severe dengue progression, highlighting the potential of ML approaches for patient risk stratification. The availability of structured hematological datasets[13], has further facilitated the development and validation of ML-based dengue diagnostic models in dengue-endemic

regions like Bangladesh. Beyond diagnostic accuracy, the role of prescriptive ML models in enhancing clinical workflows by assisting clinicians in making faster and more evidence-based decisions was highlighted[5]. Recent systematic reviews reinforce the relevance of ML and deep learning approaches for dengue diagnosis, noting their potential to improve early detection, reduce diagnostic delays, and optimize hospital admissions[2].

Despite these advances, several important gaps remain in the existing literature. First, although numerous studies have demonstrated the feasibility of ML-based dengue prediction, there is limited agreement on which specific clinical and hematological features contribute most significantly to early dengue diagnosis, particularly within Southeast Asian populations that share similar epidemiological and healthcare constraints. Second, while various ML algorithms have been evaluated individually, comprehensive comparative analyses that systematically assess multiple models using a single structured dataset remain scarce, limiting the identification of the most suitable model for real-world clinical implementation. Third, many existing studies emphasize predictive accuracy without adequately addressing the practical requirements for deployment in resource-constrained settings, where models must be not only accurate but also interpretable, efficient, and reliable for clinical decision support.

Given the continued reliance on subjective CBC interpretation and the absence of standardized, data-driven decision-support tools, there is a need for an objective and reliable machine learning-based approach that can accurately predict dengue infection using routinely collected clinical and hematological data. The lack of consensus on influential features and the absence of comparative evaluations of ML models hinder the adoption of AI-assisted dengue diagnosis in real-world clinical settings.

In response to these gaps, this study aims to develop and evaluate machine learning models for accurate dengue fever prediction using clinical and hematological data from a structured dataset of patients in Bangladesh [13]. Specifically, this study aims to:

(1) Identify the features that significantly contribute to the early diagnosis of dengue infection.
(2) Evaluate and compare the performance of different machine learning models in predicting dengue using clinical and hematological data.
(3) Determine the best-performing machine learning model suitable for clinical application as a decision-support tool in resource-limited healthcare settings.

By addressing these objectives, this study seeks to contribute to the development of practical, data-driven diagnostic tools that can support early dengue detection and improve clinical decision-making in dengue-endemic regions.

2 METHODOLOGY

2.1 Research Design

This study employed a quantitative, experimental research design to evaluate the effectiveness of selected machine learning models for early dengue disease prediction using clinical and hematological data. The research focused on assessing whether models trained on routinely available patient information, including clinical signs, symptoms, and laboratory-based hematological parameters, could reliably predict dengue infection at the individual patient level.

An experimental modeling approach was adopted, wherein multiple machine learning classifiers were trained and evaluated under identical conditions to ensure fair comparison. The design emphasized the use of data that are commonly accessible in clinical settings, particularly during the early stages of patient presentation, to enhance the practical relevance of the proposed models.

By concentrating exclusively on clinical and hematological features, the research design supports the development of predictive models intended to assist healthcare professionals in early clinical assessment and decision-making. This approach ensures that model performance reflects real-world diagnostic scenarios and aligns with the objective of providing a scalable and clinically applicable decision-support tool for dengue diagnosis.

2.2 Dataset Description

This study utilized the Structured Clinical and Hematological Dataset for Early Dengue Diagnosis in Bangladesh, which contains patient-level records collected for dengue screening and diagnosis[13]. Table 1 shows the features and their description.

Table 1: Description of Features and Target Column

Feature	Description
Gender	Patient's gender.
Age	Patient's age(years).
Platelet Count	Patient's platelet count.
WBC	Patient's white blood count.
Fever	Patient's fever.
Duration	Duration of patient's fever.
Headache	Indicates presence of headache
Muscle pain	Indicates presence of muscle pain
Rash	Indicates presence of rash
Vomiting	Indicates presence of vomiting symptom
Outcome	Target class [1: Positive, 0: Negative].

The dataset comprises demographic information, reported clinical symptoms, and hematological parameters obtained through routine laboratory testing, including platelet count and white blood cell (WBC) count.

Each record is labeled with a binary outcome indicating dengue infection status (dengue-positive or dengue-negative). The dataset reflects real-world clinical conditions, where patients typically present during the early febrile phase of illness. To enhance model generalizability and prevent geographic bias, location-specific attributes such as district, city, and area were excluded during preprocessing.

Figure 1 shows the class imbalance between the dengue-positive cases compared to dengue-negative cases in the dataset. This imbalance reflects actual clinical screening scenarios and was addressed using oversampling techniques during model training. Overall, the dataset provides a comprehensive representation of early-stage dengue cases using readily available clinical and hematological features, making it suitable for developing and evaluating machine learning–based early diagnostic models.

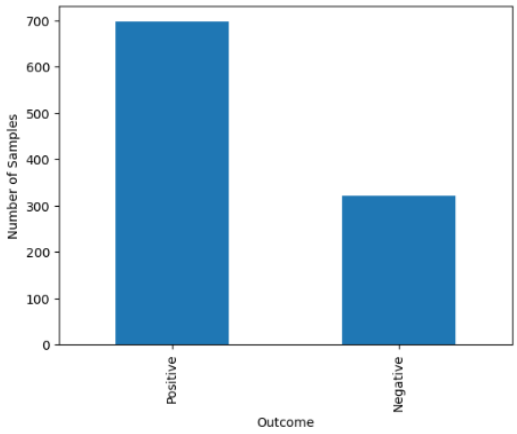


Figure 1: Dataset Class Distribution

2.3 Data Preprocessing

Data preprocessing was conducted to ensure data quality, model robustness, and generalizability across different healthcare settings. Several preprocessing steps were applied prior to model training and evaluation.

First, irrelevant and non-informative identifiers, such as record IDs, were removed to prevent data leakage and ensure that the models learned only from clinically meaningful attributes. Additionally, location-specific variables (e.g., district, city, or area identifiers) were excluded from the feature set. Although geographic information can be useful in epidemiological surveillance, it may introduce regional bias and limit the applicability of the model to other populations. Since the objective of this study is to develop a model that supports early dengue diagnosis in diverse clinical environments, removing location-based attributes improves the global generalizability of the proposed approach.

Categorical features, including demographic and symptom-related variables, were encoded using label encoding to transform them into numerical representations compatible with machine learning algorithms. This encoding approach was selected due to the predominantly binary or ordinal nature of the categorical variables and to avoid unnecessary feature expansion that could arise from one-hot encoding in a relatively compact dataset.

Numerical features were standardized using z-score normalization to ensure that all attributes contributed proportionately during model training. This step is particularly important for distance-based and margin-based classifiers, such as Support Vector Machines, which are sensitive to feature scale. Standardization also improves convergence stability for ensemble-based methods.

The dataset exhibited class imbalance, with fewer dengue-positive cases compared to dengue-negative cases, a common characteristic in real-world clinical datasets. To address this issue, Synthetic Minority Oversampling Technique (SMOTE) was applied exclusively to the training set after data splitting. SMOTE generates synthetic samples of the minority class by interpolating between existing instances, thereby improving the model's ability to learn discriminatory patterns without duplicating original observations. Applying oversampling only on the training data prevents information leakage and ensures fair evaluation on unseen test samples.

Finally, the preprocessed data were divided into training and testing sets using a stratified 70:30 split to preserve the original class distribution. This evaluation strategy provides a realistic estimate of model performance in clinical deployment scenarios, where predictions are made on previously unseen patient data.

2.4 Model Development

Four supervised machine learning classifiers were developed and evaluated in this study: Support Vector Machine (SVM), Random Forest, Extreme Gradient Boosting (XGBoost), and Adaptive Boosting (AdaBoost). These models were selected based on findings from prior systematic reviews and empirical studies demonstrating their strong performance and reliability in medical diagnosis and clinical decision-support systems, particularly for tabular clinical and hematological data. [2]

The Support Vector Machine (SVM) classifier was implemented using a radial basis function (RBF) kernel, which is well-suited for handling nonlinear relationships commonly observed in biomedical datasets. SVM aims to identify an optimal decision boundary that maximizes the margin between classes, making it effective in high-dimensional feature spaces and robust against overfitting when appropriately regularized.

The Random Forest classifier was employed as an ensemble learning method that constructs multiple decision trees using bootstrap sampling and random feature selection. By aggregating predictions across trees, Random Forest reduces variance and improves generalization, while also providing intrinsic measures of feature importance. This characteristic is particularly valuable for identifying influential clinical and hematological indicators associated with dengue infection.

The XGBoost model was implemented as a gradient-boosting framework optimized for efficiency and predictive performance. XGBoost builds decision trees sequentially, where each new tree corrects the errors of the previous ones. Its ability to model complex feature interactions, coupled with regularization mechanisms, makes it highly effective for structured clinical datasets and widely adopted in medical prediction tasks.

The AdaBoost classifier was included as a boosting-based ensemble method that focuses on improving classification performance by iteratively assigning higher weights to misclassified instances. This adaptive learning process allows the model to emphasize difficult cases, which is particularly relevant in clinical diagnosis where minority class instances, such as dengue-positive cases, require higher sensitivity.

2.5 Model Evaluation

The performance of the proposed machine learning models was evaluated using three metrics computed from the confusion matrix generated on the held-out test dataset: accuracy, sensitivity, and specificity. These evaluation metrics were selected to provide a clinically meaningful assessment of diagnostic performance, particularly in the context of early dengue detection, where classification errors may have significant clinical implications.

Accuracy represents the proportion of correctly classified instances among all predictions and provides an overall measure of model performance. However, since accuracy alone may not sufficiently reflect diagnostic effectiveness in imbalanced datasets, class-specific metrics were also considered.

Sensitivity, also known as the true positive rate, measures the ability of the model to correctly identify dengue-positive cases. High sensitivity is essential in clinical decision support systems, as it reduces the risk of false-negative predictions that could delay appropriate medical intervention. Specificity, or the true negative rate, evaluates the model's ability to correctly identify non-dengue cases, helping to minimize unnecessary follow-up testing or treatment for patients without dengue infection.

For model evaluation, the dataset was divided into training and testing subsets using a stratified 70:30 split, ensuring that the original class distribution was preserved in both sets. All models were trained exclusively on the training data, while performance metrics were computed solely on the test set. This approach provides an unbiased estimate of model performance on unseen data and reflects potential real-world deployment scenarios.

3 RESULTS

3.1 Feature Contribution Analysis

Feature importance analysis was done using SHAP (SHapley Additive exPlanations), a framework that will calculate the weight of the contribution of each feature to the predicted outcome[10]. Figure 2 illustrates the most influential factors of dengue infection. The results indicate that platelet count, white blood cell count (WBC), and muscle pain are the most influential predictors, contributing the largest impact on the model's decision. In contrast, gender, rash, and vomiting were found to have the least influence on the model output, suggesting that these features contribute less to the discrimination between dengue-positive and dengue-negative cases in the given dataset.

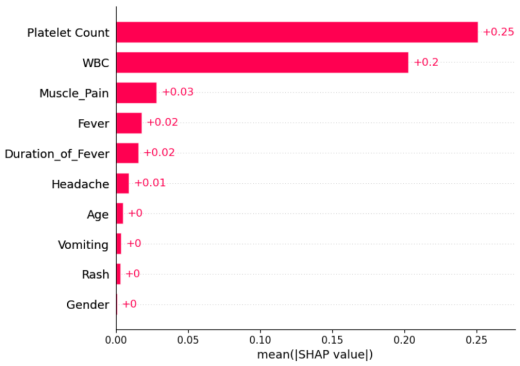


Figure 2: Most Influential Features for Dengue Diagnosis

The results reveal that platelet count and WBC contributes the most to the diagnosis of dengue infection. These results align well with established clinical knowledge of dengue[3, 14]. The prominence of these features reinforces the clinical relevance of the model and enhances its interpretability for healthcare practitioners.

3.2 Test Set Performance

Table 2 summarizes the test set performance of the evaluated machine learning models in terms of accuracy, sensitivity, and specificity using the complete set of clinical and hematological features with SMOTE applied to the training data. All four classifiers demonstrated high predictive capability, confirming that routinely collected laboratory measurements can effectively support early dengue diagnosis. Among the evaluated models, AdaBoost achieved the highest overall performance, maintaining a strong balance between sensitivity and specificity, which is essential in clinical decision support where both missed diagnoses and false alarms must be minimized.

To examine the effect of class imbalance handling, an additional experiment was conducted without applying SMOTE to the training data. Table 3 shows that even without oversampling, all models achieved high predictive performance, indicating that the dataset contains highly discriminative clinical indicators. Random Forest obtained the highest accuracy of 0.9935, followed by SVM and AdaBoost with 0.9902, and XGBoost with 0.9869. All models achieved a sensitivity of 0.9952, demonstrating strong capability in identifying dengue-positive cases. In terms of specificity, Random Forest achieved the highest value of 0.9896, while SVM and AdaBoost obtained 0.9792, and XGBoost achieved 0.9688. These results show that the models remain reliable even without oversampling, although SMOTE helps produce more consistent performance across classifiers.

To further evaluate the importance of the most influential predictors, another experiment was performed by removing platelet count, white blood cell count, and muscle pain from the feature set. The results are shown in Table 4, where the removal of these features caused a substantial decrease in performance across all models, with accuracy dropping to approximately 0.69–0.76. Sensitivity and specificity also decreased, indicating that these three features contribute significantly to the model's ability to distinguish dengue-positive from dengue-negative cases. This finding supports their clinical relevance as key indicators for early dengue assessment.

An error analysis was also conducted to evaluate the reliability of the models using the complete feature set with SMOTE applied to the training data. Table 5 summarizes the number of misclassifications of all classifiers on the test set. AdaBoost achieved the lowest number of errors, with only 2 misclassified cases, followed by SVM and Random Forest with 3 errors, and XGBoost with 4 errors. The low number of false negatives observed across the models is particularly important in clinical settings, as missed dengue cases may delay treatment and increase the risk of complications.

Table 2: Performance of the Machine Learning Models

Model	Accuracy	Sensitivity	Specificity
SVM	0.9902	0.9952	0.9791
XGBoost	0.9869	0.9905	0.9896
Random Forest	0.9902	0.9857	0.9896
AdaBoost	0.9935	0.9952	0.9896

Table 3: Performance of the Machine Learning Models(Without SMOTE)

Model	Accuracy	Sensitivity	Specificity
SVM	0.9902	0.9952	0.9792
XGBoost	0.9869	0.9952	0.9688
Random Forest	0.9935	0.9952	0.9896
AdaBoost	0.9902	0.9952	0.9792

Table 4: Performance of the Machine Learning Models(Without Top Features)

Model	Accuracy	Sensitivity	Specificity
SVM	0.7549	0.819	0.6146
XGBoost	0.6993	0.7857	0.5104
Random Forest	0.6863	0.7476	0.5521
AdaBoost	0.7647	0.8381	0.6042

Table 5: Error Analysis of the Machine Learning Models

Model	TP	TN	FP	FN	Misclassified
SVM	209	94	2	1	3
XGBoost	207	95	1	3	4
Random Forest	208	95	1	2	3
AdaBoost	209	95	1	1	2

3.3 Discussion

The results demonstrate that machine learning models trained on clinical and hematological features can effectively support early dengue diagnosis. The superior test performance of AdaBoost suggests better generalization to unseen data. The strong influence of platelet count and WBC highlights the clinical relevance of the model, while the use of SMOTE mitigates class imbalance without compromising performance.

Compared to previous studies using clinical and hematological data for dengue prediction, the proposed model demonstrates competitive or improved performance. Several earlier works reported test accuracies ranging from approximately 85% to 95% [11][8][15][16], often relying on a limited subset of laboratory indicators or smaller datasets. In contrast, the AdaBoost model in this study achieved higher predictive accuracy while maintaining clinically meaningful sensitivity and specificity values. This improvement may be attributed to the integration of both symptomatic and hematological features, as well as the use of ensemble learning, which effectively captures non-linear relationships predominant in epidemiological data.

Importantly, the exclusion of location-specific attributes during preprocessing enhances the generalizability of the proposed approach. By relying only on routinely collected clinical symptoms and basic laboratory results, the model can be applied across different healthcare settings without requiring contextual or demographic adjustments. This makes the approach particularly suitable for deployment in primary care units and resource-constrained environments.

From a clinical decision support perspective, the model can assist physicians in early risk stratification, enabling timely monitoring, prioritization of laboratory testing, and early intervention. While the model is not intended to replace confirmatory diagnostic tests, its ability to provide rapid, data-driven insights based on readily available features highlights its potential role as a supportive tool in clinical workflows.

Overall, the results demonstrate that machine learning models—particularly AdaBoost—can effectively leverage clinical and hematological data to support early dengue diagnosis, offering an interpretable and practical solution that aligns with real-world clinical needs.

4 CONCLUSION

4.1 Summary

This study examined the effectiveness of machine learning models for the early prediction of dengue infection using routinely available clinical and hematological data. The research addresses the need for reliable, data-driven tools that can support clinical decision-making at the patient level, particularly during the early stages of illness when confirmatory diagnostic tests may be unavailable, delayed, or inconclusive.

Using the Structured Clinical and Hematological Dataset for Early Dengue Diagnosis in Bangladesh, four widely used machine learning classifiers—Support Vector Machine (SVM), Random Forest, Extreme Gradient Boosting (XGBoost), and AdaBoost—were implemented and evaluated. These models were selected due to their established effectiveness in medical classification tasks and their ability to learn complex patterns from structured clinical datasets.

To improve applicability across different healthcare settings, location-based attributes were excluded during data preparation. This ensured that model predictions were driven by clinical signs, symptoms, and hematological measurements rather than geographic indicators. The dataset was divided using a stratified 70:30 train–test split, preserving the original class distribution and allowing for an unbiased evaluation on unseen patient data. Class imbalance favoring dengue-positive cases was addressed through the application of SMOTE on the training data, enabling the models to learn balanced representations without introducing bias into the test set.

Model performance was evaluated using accuracy, sensitivity, and specificity, with particular emphasis on sensitivity due to its clinical relevance. Sensitivity reflects the model's ability to correctly identify dengue-positive patients, which is critical in minimizing missed diagnoses and supporting timely clinical intervention. Among the evaluated models, AdaBoost demonstrated the strongest overall predictive performance on the test set, indicating its effectiveness in distinguishing dengue-positive cases based on clinical and hematological features. SVM also exhibited strong and consistent performance, suggesting reliable discrimination between dengue and non-dengue cases.

Feature importance analysis revealed that hematological parameters, especially platelet count and white blood cell count, contributed most significantly to the prediction of dengue infection. These findings are consistent with established clinical observations

and reinforce the relevance of these markers in early dengue assessment.

Overall, the findings demonstrate that machine learning models trained solely on clinical and hematological data can effectively support early dengue diagnosis. Such models have the potential to serve as clinical decision-support tools, assisting healthcare professionals in identifying suspected dengue cases and prioritizing further diagnostic testing or monitoring, particularly in resource-constrained settings.

4.2 Future Work

While this study demonstrates the potential of machine learning models to support early dengue diagnosis using clinical and hematological data, several opportunities remain for future research and improvement. One important direction is the validation of the proposed models using datasets from different hospitals, regions, or countries to further assess their generalizability across diverse patient populations and healthcare settings.

Future studies may also explore the integration of additional hematological variables in the data to test how well the model could perform. Evaluating model performance at specific time points following symptom onset could also provide deeper insight into how early reliable predictions can be achieved in real clinical scenarios.

Another promising area of future work involves assessing the practical use of these models in clinical workflows. This includes investigating how machine learning–based predictions can assist healthcare professionals in triage, monitoring, or prioritizing confirmatory diagnostic testing. User-centered evaluations involving clinicians may help refine model outputs to ensure they are interpretable and actionable in real-world practice.

Lastly, future research may focus on developing simplified decision-support tools, such as risk scoring systems or clinical dashboards, based on the most influential features identified in this study. Such tools could improve accessibility and usability, particularly in resource-limited healthcare environments, and support earlier intervention for patients at risk of dengue infection.

REFERENCES

[1] B. Abdulgalil, S. Abraham, and W. M. Ismael. 2022. Early Diagnosis for Dengue Disease Prediction Using Efficient Machine Learning Techniques Based on Clinical Data. *Journal of Robotics and Control* 3, 3 (2022), 257–268.

[2] D. C. Andrade Girón, W. J. Marín Rodriguez, F. d. M. Lioo-Jordan, and J. L. Ausejo Sánchez. 2025. Machine Learning and Deep Learning Models for Dengue Diagnosis Prediction: A Systematic Review. *Informatics* 12, 1 (2025), 15. https://doi.org/10.3390/informatics12010015

[3] Juthatip Chaloemwong, Adisak Tantiworawit, Thanawat Rattanathammethee, Sasinee Hantrakool, Chatree Chai-Adisaksopha, Ekarat Rattarittamrong, and Lalita Norasetthada. 2018. Useful clinical features and hematological parameters for the diagnosis of dengue infection in patients with acute febrile illness: a retrospective study. *BMC hematology* 18, 1 (2018), 20.

[4] T.-S. Ho, T.-C. Weng, J.-D. Wang, H.-C. Han, H.-C. Cheng, and C.-C. Yang. 2020. Comparing machine learning with case-control models to identify confirmed dengue cases. *PLoS Neglected Tropical Diseases* 14, 11 (2020), e0008843. https://doi.org/10.1371/journal.pntd.0008843

[5] William Hoyos, Jose Aguilar, Mayra Raciny, and Mauricio Toro. 2023. Case studies of clinical decision-making through prescriptive models based on machine learning. *Computer Methods and Programs in Biomedicine* 242 (2023), 107829. https://doi.org/10.1016/j.cmpb.2023.107829

[6] William Hoyos, Jose Aguilar, and Mauricio Toro. 2021. Dengue models based on machine learning techniques: A systematic literature review. *Artificial Intelligence in Medicine* 119 (2021), 102157. https://doi.org/10.1016/j.artmed.2021.102157

[7] Shang-Wei Huang, Huey-Pin Tsai, Sheng-Jang Hung, Wen-Chien Ko, and Jen-Ren Wang. 2020. Assessing the risk of dengue severity using demographic information and laboratory test results with machine learning. *PLoS Neglected Tropical Diseases* 14, 12 (2020), e0008960. https://doi.org/10.1371/journal.pntd.0008960

[8] Sadaf Khan, Rahat Ullah, Aurang Khan, Naseeb Wahab, Muhammad Bilal, and Mushtaq Ahmed. 2016. Analysis of Dengue Infection Based on Raman Spectroscopy and Support Vector Machine (SVM). *Biomedical Optics Express* 7, 6 (June 2016), 2249–2256. https://doi.org/10.1364/BOE.7.002249

[9] Yun E. Liu, S. Saul, A. M. Rao, M. L. Robinson, O. L. Agudelo Rojas, A. M. Sanz, M. Verghese, D. Solis, M. Sibai, C. H. Huang, et al. 2022. An 8-Gene Machine Learning Model Improves Clinical Prediction of Severe Dengue Progression. *Genome Medicine* 14 (2022), 33. https://doi.org/10.1186/s13073-022-01034-w

[10] Scott Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. arXiv:1705.07874 [cs.AI] https://arxiv.org/abs/1705.07874

[11] N. I. N. Nordin et al. 2020. Machine Learning Approach for Dengue Disease Classification. In *Journal of Physics: Conference Series*, Vol. 1496. 012006. https://doi.org/10.1088/1742-6596/1496/1/012006

[12] Philippine News Agency. 2025. DOH: Dengue cases up 73% in first 2 months of 2025. https://www.pna.gov.ph/articles/1245853 Accessed: 2025.

[13] Nure Alam Rachin and Kholilullah Ibrahim Siddiki. 2025. Structured Clinical and Hematological Dataset for Early Dengue Diagnosis in Bangladesh. Mendeley Data, V1. https://doi.org/10.17632/673swz9tb4.1

[14] Nusrat Jahan Riya, Mritunjoy Chakraborty, and Riasat Khan. 2024. Artificial intelligence based early detection of dengue using CBC data. *IEEE Access* (2024).

[15] Nuur Alifah Mawar Salim, Yap Bee Wah, Cameron Reeves, Matthew Smith, Wan Fairos Wan Yaacob, Rose Nani Mudin, Rusdi Dapari, Nik Nur Fatin Fatihah Sapri, and Ubydul Haque. 2021. Prediction of Dengue Outbreak in Selangor Malaysia Using Machine Learning Techniques. *Scientific Reports* 11, 1 (January 2021), 939. https://doi.org/10.1038/s41598-020-79193-2

[16] D. Sarma, S. Hossain, T. Mittra, M. A. M. Bhuiya, I. Saha, and R. Chakma. 2020. Dengue Prediction Using Machine Learning Algorithms. In *Proceedings of the 2020 IEEE 8th R10 Humanitarian Technology Conference (R10-HTC)*. Kuching, Malaysia, 1–6. https://doi.org/10.1109/R10-HTC49770.2020.9357035

Queue Management System with Smart Feedback, Sentiment Analysis and Geolocation in Holy Child College of Davao

Amy Rosal Chan
Holy Child College of Davao, Philippines
alrheemfabian@gmail.com

AlrheemFabian
Holy Child College of Davao, Philippines
alrheemfabian@gmail.com

ABSTRACT

Traditional manual queuing at Holy Child College of Davao has caused long waiting times, crowded service areas, and a lack of real-time information for administrators. These issues show the need for a more efficient and organized system. To solve this, the study created a web-based Queue Management System (QMS) with Smart Feedback, Sentiment Analysis, and Geolocation features to improve and digitalize student transactions. The system was built using the Agile development method and implemented through the Django framework. Geolocation services were added using the Geopy library to verify if students were inside the campus, and Little's Law was used to calculate accurate Estimated Time of Arrival (ETA). Twenty-five students joined the User Acceptance Testing to check how easy and useful the system was. Sentiment analysis was done using Natural Language Processing tools like VADER and TextBlob, together with a camera-based emotion recognition feature to better classify feedback. The evaluation results showed strong performance. The system achieved 92% reliability in Alpha testing and 100% validity in Beta testing. User testing also showed high satisfaction, with an average rating of 4.8 out of 5 for both Perceived Usefulness and Perceived Ease of Use. Improvements included better queue monitoring, less crowding, and clearer service updates. Overall, the study shows that using data-driven and automated systems can improve school services and help administrators make better decisions. For future development, the study suggests creating a mobile application, expanding the system to more departments, and improving machine learning for more accurate emotion and feedback analysis.

KEYWORDS

Queue Management System, Real-time Notifications, Sentiment Analysis, User-Friendly Design, Services Efficiency

1 Introduction

Queue Management Systems (QMS) are modern digital tools that use real-time tracking, virtual queuing, automatic notifications, estimated time of arrival (ETA), geolocation, and data analytics to make service faster and more organized. Schools, offices, and companies use these systems to manage long lines and let users know when it is their turn. using a QMS in schools improved efficiency and customer satisfaction [1]. Users can access the system through web or mobile sites to join the queue remotely, get live updates, and give feedback easily. Adding smart feedback with sentiment analysis makes the system stronger by checking user feedback and emotions to understand satisfaction [2]. The smart feedback system uses a camera to detect facial expressions and works with sentiment analysis to automatically check a user's mood during or after a transaction. Sentiment analysis helps collect emotional data from comments, surveys, or visual signs, giving a clearer picture of how users feel [3]. The system can also integrate Little's Law from queuing theory to dynamically manage the service queue by relating the average number of people in the system, arrival rate, and average waiting time [4]. It can calculate the arrival rate by dividing the total number of arrivals by the total observation time and estimate individual waiting times by multiplying the user's queue number by the system's average waiting time. Geolocation tracking can be implemented using Python libraries such as Geopy to obtain real-time location coordinates, ensuring that students remain within the allowed school area while waiting. These features work together to make services faster, reduce waiting time, and create a system that is easy and convenient for users. In traditional queues, people often wait long hours without knowing when they will be served. This problem occurs in schools, banks, hospitals, and government offices, where long lines make people leave before being served [5]. Users now expect faster and fairer service, so institutions must make queues more efficient and organized [6]. Fair waiting times and organized lines improve satisfaction [7]. At Holy Child College of Davao (HCCD), the current queues are still manual. Students and visitors wait in long lines for enrollment, cashier payments, and other services without knowing when it will be their turn. The system lacks digital features like real-time updates, geolocation, or ETA, which causes confusion and frustration [8]. The system applies Little's Law in these calculations to predict queue times accurately and manage service flow effectively, while anomaly-detection methods flag delayed queues or unusual waiting patterns. Adding these features will make the system clear, organized, and reliable, giving smoother and smarter experience. A survey of 32 respondents from the Student, Registrar, and Cashier Offices at HCCD–Green Meadows revealed problems with the current queues. Students reported that the lines are too long, confusing, and that some

people cut in line. Staff mentioned that there are not enough personnel to manage the queues and that technical problems occur during busy hours. Because of this, the system feels slow, unfair, and difficult to monitor. These results indicate that the school needs a digital system that can organize lines automatically, provide real-time updates, and clearly show each person's place in line [9]. The proposed system will include an ETA feature to inform students how long they will wait, a Geolocation feature to ensure they remain on campus, and Smart Feedback with Sentiment Analysis using pre-trained models like mBERT to assess users' feelings through text and facial expressions. Together, these features will make the queueing process faster, simpler, fairer, and more organized. To address these issues, the researchers propose a Queue Management System for Holy Child College of Davao with Smart Feedback and Sentiment Analysis. The system will have virtual queuing with real-time updates, automatic notifications, geolocation tracking using Geopy, and a smart feedback system that uses camera-based emotion recognition and pre-trained NLP models to measure satisfaction. The ETA feature will predict waiting time and remove inactive queue numbers if users do not show up within 1 minute. This system will reduce waiting times, improve service satisfaction, and make daily processes smoother. Overall, it will create a fairer and more organized process while using emotion, feedback, and location data to continuously improve service quality. The combination of Little's Law-based queue management, smart feedback, and geolocation ensures a fully modern queue system.

2 **Methodology**

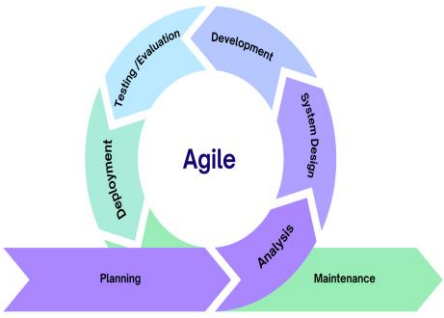


Figure 1: **Agile Model Development Model**

In this chapter, the researchers develop the Queue Management System with Smart Feedback and Sentiment Analysis for Holy Child College of Davao. The system was created using the Agile methodology because of its flexibility and step-by-step approach. The process was divided into six phases: planning, design, implementation, testing, deployment, and maintenance. Feedback from students was continuously gathered to guide improvements. The researchers, staff, and users worked to enhance key features such as real-time notifications, feedback collection, and queue status features. Continuous development of the system improved the efficiency of the system, improved the usability of the system, and overall improved the output quality of the system, so that it can be deployed as a user-friendly and reliable queue management system for Holy Child College of Davao.

2.1 **Development**

The researchers used the Agile methodology to make the system flexible, efficient, and easy to improve. Agile works by building the system step by step in small, manageable parts. This approach allowed the researchers to gather feedback from students, guardians, and staff at different stages, continuously improving the system and adjusting to changing needs. The development followed these steps:

Sprint 1: The team focused on planning the project, defining the objectives, scope, and overall system requirements. This included setting up the development environment, identifying the necessary tools, and outlining the project roadmap to guide the development process.

Sprint 2: The researchers conducted a thorough analysis of the system requirements. This included gathering and evaluating information from students, guardians, and staff to understand their needs, workflows, and expectations from the Queue Management System. The analysis helped define system features such as queue handling, feedback collection, sentiment analysis, and geolocation.

Sprint 3: A user-friendly interface was designed for students, guardians, and staff to ensure smooth navigation, accessibility, and ease of use.

Sprint 4: The main features of the system were implemented, including user management, queue handling, notifications, and feedback collection. The feedback module was integrated with tools to analyze comments as positive, negative, or neutral. The backend was secured and optimized for efficient data management.

Sprint 5: The system underwent thorough testing using both white-box and black-box methods. All modules were verified to ensure they worked as intended.

Sprint 6: After resolving all issues, the system was prepared for deployment. Critical errors were addressed, and the system was verified to meet the requirements for live use.

Sprint 7: Ongoing maintenance was conducted to ensure the system remained stable, secure, and updated. Any issues reported by users were resolved, and improvements were

made as needed to keep the system running smoothly after launch.

Throughout all sprints, daily stand-ups were conducted to discuss progress, identify challenges, and coordinate tasks among team members. Sprint reviews and retrospectives were held at the end of each sprint, allowing the team to gather feedback from students, guardians, and staff and make necessary adjustments to the system.

The choice of development tools facilitated iterative progress, adaptability, and efficient collaboration for the Queue Management System. Django was used for backend logic and secure data management, ensuring smooth handling of queues, user information, and feedback. JavaScript and Chart.js were implemented for dynamic features such as real-time queue updates, visual analytics of feedback, and sentiment analysis results. HTML, CSS, and Bootstrap were used for a responsive, user-friendly design, allowing students, guardians, and staff to navigate the system easily on different devices. The system also incorporated geolocation features to manage queues and services efficiently within the campus, and feedback modules were integrated with sentiment analysis tools to classify comments as positive, neutral, or negative, enhancing service evaluation and decision-making.

3 **Results**

3.1 **Alpha Testing**

The alpha testing phase was conducted by the research adviser, Mr. Justin Kaye A. Lolo, to evaluate the system's reliability, validity, and overall performance. The Queue Management System underwent a comprehensive assessment to ensure that all core features and functionalities operated accurately and efficiently. The results of the evaluation indicate that the system performed well and met the required quality standards, achieving an overall rating of 92%, which reflects its effectiveness and readiness for further development and improvement.

The table presents the results of the alpha testing conducted by the researchers and evaluated by the research adviser, Mr. Justin Kaye A. Lolo.

Table 1. **Result of Alpha Testing**

Criteria Category	Your Score
Overall Score	92%
Product Quality	9/10
Functionality Testing	9/10
Functionality	9/10
Reliability	9/10
Usability	9/10
Efficiency	10/10
Maintainability	10/10
Portability	9/10

3.2 **Beta Testing**

The beta testing phase was conducted by the beta user, Ms. Ma. Elena Laidan, to evaluate the system from both user and operational perspectives. The testing focused on assessing the system's reliability, validity, usability, and overall efficiency, including its core features and workflow.

The results confirmed that the system is reliable, efficient, and user-friendly, achieving a perfect 100% rating.

The table presents the results of the beta testing conducted by the researchers and evaluated by the beta user, Ms. Ma. Elena Laidan.

Table 2. **Result of Beta Testing**

Criteria Category	Weight	Score
PRODUCT QUALITY		
Reliability	5%	10/10
Robustness	5%	10/10
Efficiency of Use	5%	10/10
Ease of Use	5%	10/10
FUNCTIONALITY TESTING		
Valid Data Execution	10%	10/10
Error/Warning Messages	10%	10/10
FUNCTIONALITY	10%	10/10
RELIABILITY	10%	10/10
USABILITY	10%	10/10
EFFICIENCY	10%	10/10
MAINTAINABILITY	10%	10/10
PORTABILITY	10%	10/10
FINAL OVERALL SCORE	**100%**	**100%**

3.3 User Acceptance Test

The User Acceptance Test (UAT) was conducted with 25 students from various programs, including BSIT, BSBA, BEED, BSED, and BSCRIM. The UAT focused on evaluating the Queue Management System with Smart Feedback, integrated Sentiment Analysis, and Geolocation features. During the assessment, the researchers explained and demonstrated the system's purpose and functionality to ensure that all participants clearly understood its operations. Results from the UAT indicate that the system is efficient, reliable, and user-friendly. Students found the interface easy to navigate, with organized layouts and clear visualizations. The smart feedback module, combined with VADER and TextBlob sentiment analysis, accurately interpreted user remarks to generate meaningful insights. The geolocation feature functioned effectively, ensuring proper validation of user location during queue operations. Additionally, the system demonstrated real-time synchronization, handled multiple users simultaneously, and provided smooth queue processing. Its overall design was described as visually appealing, accessible, and convenient. These features collectively enhance service flow, streamline queue monitoring, and support data-driven decision-making for improved service delivery.

The table presents the course distribution of respondents to identify the most common academic program among the UAT respondents.

Table 3. **Course Distribution of Respondent**

Course Range	Number of Respondents	Percentages (%)	Rank
BSIT	13	52%	1
BSBA	5	20%	2
BSED	3	12%	3
BEED	1	4%	5
BSCRIM	3	12%	4

The table presents the interpretation guide to determine the ratings of each evaluation criterion and the system's overall rating.

Table 4. **UAT Interpretation Guide**

Mean Range	Interpretation
4.50 - 5.00	Excellent
3.50 - 4.49	Good
2.50 - 3.49	Fair
1.00 - 2. 49	Poor

The table presents the summary of the mean score per category on the UAT questionnaire based on the UAT conducted by the researchers.

Table 5. **Summary of Mean Score per Category**

Evaluation Criteria	Mean Score	Interpretation
Perceived Usefulness (PU)	4.8	Excellent
Perceived Ease of Use (PEOU)	4.8	Excellent

4 Conclusion

The researchers developed a Queue Management System with Smart Feedback, integrated Sentiment Analysis and Geolocation features. The results demonstrate the system's effectiveness in achieving its objectives by improving queue organization, enhancing service efficiency, enabling real-time monitoring, supporting data-driven feedback analysis, and ensuring accurate user validation. Overall, the system contributes to a more organized, reliable, and user-friendly service experience.

4.1 Recommendations

4.1.1 Software / System

Some features were not included in this study but can be added in the future, such as QR code ticket retrieval, identity verification using biometrics or RFID, and automatic service window assignment. Mobile support, notifications, and improved AI tools can also make the system easier to use and more efficient. Additionally, integrating gesture recognition feedback could provide an alternative way for users to interact with the system or give non-verbal feedback, improving accessibility and overall user experience.

4.1.2 Further Research

Future studies can focus on improving existing features, like polishing the emotion detection tool, predicting wait times, and balancing queues across departments. Exploring new technologies, such as secure logging, gesture recognition feedback, or more advanced AI analysis, can also enhance the system and make it more adaptive to user behavior.

REFERENCES

[1] Maala, R. F. (2023). Queuing Management System In Manuel S. Enverga University Foundation Candelaria Inc. International Journal of Advanced Research in Computer Science,14(6),4453.https://doi.org/10.26483/ijarcs.v14i6.7039

[2] Lv, Z. (2024). Optimized Queuing Modeling in Multimodal Sentiment Analysis: An Examination and application. Highlights in Science Engineering and Technology,85,836–840. https://doi.org/10.54097/q1hj7a40

[3] Prakash, N. J., Swathiramya, N. R., Balambigai, N. G., Menaha, N. R., & Abhirami, N. J. (2024). AI-Driven Real-Time Feedback System for enhanced student support: leveraging sentiment analysis and machine learning algorithms. International Journal of Computational and Experimental Science and Engineering, 10(4). https://doi.org/10.22399/ijcesen.780

[4] Boo, N. D. (2025). A Brief Introduction to Queueing Theory (Bachelor's thesis). https://studenttheses.uu.nl/bitstream/handle/20.500.12932/48871/Scriptie_Nathande Boo_6219969%20(1)%20(1).pdf?sequence=1

[5] Moorthy, A. D., Kavitha, D., Logeshwaran, R., Kumar, N. V., & Karthick, V. (2024). PSFAS: Progressive Student Feedback Analysis System for improved teaching learning with intelligent processing of open-responses. Journal of Applied Research in Higher Education, 17(6), 2307–2329. https://doi.org/10.1108/jarhe-04-2024-0157

[6] Alkhodair, S. A. (2023). A Sentiment Analysis-Based Smartphone Application To Continuously Assess Students' Feedback And Monitor The Quality Of Courses And The Learning Experience In Educational Institutions. DergiPark (Istanbul University). https://doi.org/10.51508/intcess.202309

[7] Maria, N. H. (2024). Hybrid Lexicon and Transformer-Based Sentiment Analysis of Student Feedback for Faculty Evaluation: A Speech-to-Text Approach. Communications on Applied Nonlinear Analysis, 32(2), 610–624. https://doi.org/10.52783/cana.v32.1857

[8] Bhimavarapu, U. (2025). Smart Education Sentiment Analysis combining CNN and ANN for deeper insights from student feedback. In Advances in computational intelligence

and robotics book series (pp. 151–184). https://doi.org/10.4018/979-8-3693-9974-3.ch006

[9] Fitsilis, P. (2025). PULSE Index: Aspect-based sentiment analysis and topic modeling of student feedback for department rankings in higher education. OSF Preprints. https://doi.org/10.31235/osf.io/vy9kc_v1

Comparative Study of MobileNetV2 and Hybrid MobileNetV2 with SVM Classifier for Efficient Waste Classification in Low-Resource Environments

Earl Jay G. Torayno
South Philippine Adventist College
Matanao, Davao del Sur
toranoearljay06@gmail.com

J Faye Champ Asaria
South Philippine Adventist College
Matanao, Davao del Sur
asariafaye@gmail.com

Neila M. Paglinawan-Muñez
South Philippine Adventist College
Matanao, Davao del Sur
nmpmunez@gmail.com

ABSTRACT

Effective waste management requires automated classification systems capable of operating on low-resource computing devices. This study compares standard MobileNetV2 fine-tuning against a proposed Hybrid MobileNetV2+SVM approach for waste classification using the TrashNet dataset (2,527 images, six categories). Experiments were repeated across five independent runs to enable statistical validation. The Hybrid model achieves a mean accuracy of 86.84% (±0.27) versus 80.16% (±2.81) for the Baseline, with all differences confirmed statistically significant via paired-samples t-tests ($p < 0.05$, large effect sizes). The Hybrid model further reduces training time by 89% (657 s vs. 6,737 s) and model size on disk by 8% (25.67 MB vs. 27.97 MB). Although the Hybrid model incurs a slightly higher mean inference time (223.03 ms vs. 154.72 ms), this difference was not statistically significant (p = .443), suggesting comparable per-image prediction latency between the two approaches. These results demonstrate that frozen feature extraction with SVM classification offers better accuracy-efficiency trade-offs for deployment in resource-constrained environments.

KEYWORDS

Waste Classification, MobileNetV2, Support Vector Machine, Deep Learning, Edge Computing

1 INTRODUCTION

The escalating global waste crisis represents one of the most pressing environmental challenges of our time, with profound implications for public health, urban sustainability, and ecosystem preservation. According to recent environmental reports from the United Nations Environment Programme, global waste generation is expected to increase by 70% by 2050, reaching approximately 3.4 billion tons annually, with developing nations in Asia and Africa facing the most significant challenges in waste management infrastructure development [10]. This alarming trend necessitates innovative technological solutions that can operate effectively within existing resource constraints while providing scalable, sustainable waste management capabilities.

Traditional waste management approaches rely heavily on manual sorting processes, which are not only labor-intensive and time-consuming but also subject to significant human error and inconsistency. Manual classification systems typically achieve accuracy rates between 70-80%, with substantial variability depending on operator experience, fatigue levels, environmental conditions, and the complexity of the waste stream [2]. Furthermore, manual sorting exposes workers to potentially hazardous materials, creating occupational health risks including exposure to toxic substances, biological contaminants, and physical injuries that automated systems could substantially mitigate.

Recent advances in deep learning, particularly Convolutional Neural Networks (CNNs), have demonstrated remarkable potential for automated waste classification tasks. These systems can achieve accuracy rates exceeding 90% when trained on sufficiently large and diverse datasets, surpassing human-level performance in many controlled scenarios [3]. However, the computational requirements of state-of-the-art CNN architectures such as ResNet, DenseNet, and EfficientNet often exceed the capabilities of edge computing devices commonly deployed in waste management facilities, particularly in developing regions where infrastructure limitations are most pronounced.

Several CNN architectures have been specifically adapted and evaluated for waste classification tasks. VGG networks, despite their computational intensity, have demonstrated strong performance due to their deep architecture and ability to capture complex visual patterns [3]. ResNet architectures have gained popularity due to their residual connections, which facilitate training of deeper networks and help prevent the vanishing gradient problem [3]. Real-time waste detection has emerged as an active research area, with YOLO-based models (YOLOv5, YOLOv7, YOLOv8s) showing promise for simultaneous detection and classification [9]. Transfer learning has become increasingly important, with pre-trained models on ImageNet providing powerful feature extractors that can be adapted with limited task-specific data [1].

The challenge of balancing classification accuracy with computational efficiency becomes particularly critical when considering deployment scenarios involving edge computing devices with limited processing power, memory constraints, and energy restrictions. This limitation has led researchers to explore lightweight architectures and hybrid approaches that can maintain competitive accuracy while significantly reducing computational requirements [4].

MobileNetV2 represents a significant advancement in lightweight CNN architectures, specifically designed for mobile and edge computing applications. The architecture employs depthwise separable convolutions and inverted residual blocks to achieve substantial reductions in computational complexity while maintaining competitive accuracy levels [4, 10]. However, even with these optimizations, end-to-end training and deployment of MobileNetV2 can still present challenges for extremely resource-constrained scenarios.

The integration of traditional machine learning algorithms with deep learning feature extractors represents a promising approach

for addressing computational constraints while maintaining classification performance. Support Vector Machines (SVMs) have demonstrated strong generalization capabilities, particularly when working with high-dimensional feature spaces typical of CNN outputs. The combination of frozen CNN feature extractors with SVM classifiers offers the potential to leverage the representational power of deep learning while maintaining the computational efficiency of traditional machine learning approaches [3, 5, 7].

This study addresses the critical need for efficient waste classification systems by conducting a comparative analysis between standard end-to-end MobileNetV2 and a hybrid approach combining frozen MobileNetV2 feature extraction with SVM classification. The research specifically targets deployment scenarios characterized by limited computational resources, aiming to identify optimal architectures that balance classification accuracy with computational efficiency.

The main contributions of this work are threefold: (1) empirical comparison of end-to-end versus hybrid CNN-SVM approaches specifically for waste classification tasks, providing quantitative evidence for architecture selection decisions; (2) analysis of accuracy-efficiency trade-offs, including detailed performance metrics across multiple evaluation criteria and per-class performance analysis; and (3) practical deployment guidelines for implementing waste classification systems in low-resource environments, with specific recommendations for different computational constraints and use case scenarios.

2 METHODOLOGY

2.1 Dataset Description and Characteristics

This study utilizes the TrashNet dataset [8], a publicly available and widely validated benchmark for waste classification research. The dataset comprises 2,527 labeled images across six waste categories, as shown in Figure 1. Paper and plastic are the most prevalent categories, while trash represents the smallest class at only 5.5% of the dataset, reflecting a realistic but imbalanced waste stream composition. Images exhibit variability in lighting, background complexity, and object orientation, providing a reasonable test bed for model evaluation. The dataset is split into training (80%, 2,022 images) and validation (20%, 505 images) sets using stratified random sampling to preserve class proportions in each split.

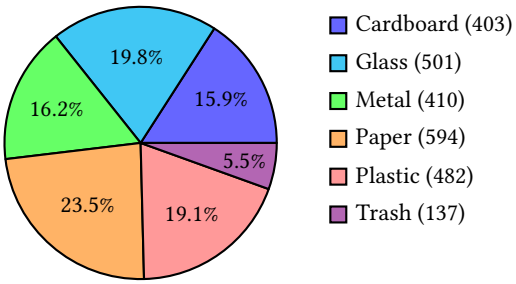


Figure 1: TrashNet dataset composition by category (total: 2,527 images).

2.2 Data Preprocessing Pipeline

The data preprocessing pipeline implements a set of transformations designed to standardize input images while enhancing model robustness through data augmentation. All images undergo initial resizing to 224×224 pixels to match the input requirements of the MobileNetV2 architecture. This resizing operation uses bilinear interpolation to maintain image quality while ensuring consistent input dimensions across all training and validation samples.

For the baseline MobileNetV2 model, we apply standard image normalization by scaling pixel values to the range [0, 1]. This normalization helps stabilize training and ensures that input values fall within a reasonable range for the neural network. The hybrid model implementation utilizes MobileNetV2's native preprocessing function (preprocess_input), which applies the specific normalization scheme used during the model's original ImageNet training, ensuring consistency with the pre-trained weights.

Our data augmentation strategy incorporates several techniques designed to improve model generalization and reduce overfitting. We implement random horizontal flipping with a probability of 0.5, random rotation up to 20 degrees, and random zoom operations up to 10% of the image dimensions. These augmentations are applied only to the training set, while the validation set receives only the essential preprocessing steps (resizing and normalization) to ensure consistent evaluation conditions and reliable performance measurement.

2.3 Baseline Model Architecture and Training

The baseline model applies transfer learning using MobileNetV2 pre-trained on ImageNet [4]. The backbone is loaded without its classification head; a custom head is appended consisting of a Global Average Pooling layer, a Dense layer (128 units, ReLU), Dropout (rate = 0.3), and a six-class softmax output layer. Training proceeds in two phases: the backbone is initially frozen to train only the custom head (30 epochs), then unfrozen for full fine-tuning (10 epochs). The Adam optimizer is used with an initial learning rate of 10^{-3}, reduced by a factor of 0.1 when validation loss plateaus for five consecutive epochs (ReduceLROnPlateau). Pixel values are normalised to $[0, 1]$ for the baseline. The complete trained model—including backbone weights, custom head, and Adam optimizer state—is saved as a Keras `.h5` file.

2.4 Hybrid Model Implementation

The hybrid model uses a permanently frozen MobileNetV2 backbone (ImageNet weights) to extract 1,280-dimensional feature vectors via Global Average Pooling. No backpropagation is performed through the backbone. MobileNetV2's native `preprocess_input` function (scaling pixels to $[-1, 1]$) is applied to maintain consistency with the pre-trained weights. Extracted features are passed to a Support Vector Machine with an RBF kernel; the regularisation parameter C and kernel coefficient γ are optimised via 5-fold cross-validation on the training set ($C \in \{0.1, 1, 10, 100\}$; $\gamma \in \{\text{scale}, \text{auto}\}$). The best configuration across all five runs was $C = 10$, $\gamma = \text{scale}$. The hybrid model is saved as a stripped feature-extractor `.h5` file (backbone weights only, no optimizer state) plus a lightweight SVM pickle (`.pkl`).

2.5 Evaluation Protocol

Model size is reported as the total size of saved artefacts on disk (MB), not the number of trainable parameters. To enable statistical validation, all experiments are repeated across five independent runs with different random seeds. An independent samples t-test (two-tailed, $\alpha = 0.05$) is applied to compare per-run accuracy, F1-scores, inference time, and training time between the two models, treating each run as an independent observation per group. Where group variances differ substantially, Welch's correction is applied, reflected in the reported degrees of freedom.

3 RESULTS

3.1 Overall Performance Comparison

Table 1 presents the mean performance across five independent runs. The Hybrid model achieves a mean accuracy of 86.84% (±0.27), outperforming the Baseline at 80.16% (±2.81) by 6.68 percentage points. Training time is reduced by 89%, from a mean of 6,737 s to 657 s. The Hybrid model's on-disk size (25.67 MB) is marginally smaller than the Baseline (27.97 MB); the difference arises because the Baseline serialises the full Keras model including optimizer state, while the Hybrid saves only backbone weights and a lightweight SVM file.

Table 1: Mean Performance Comparison Across Five Runs

Metric	Baseline MobileNetV2	Hybrid MobileNetV2+SVM
Accuracy (%)	80.16 ± 2.81	86.84 ± 0.27
Macro F1-Score	0.764 ± 0.039	0.846 ± 0.013
Weighted F1-Score	0.798 ± 0.034	0.870 ± 0.000
Inference Time (ms)	154.72	223.03
Model Size (MB)[a]	27.97	25.67
Training Time (s)	6,736.6	657.5

[a]Disk size of saved artefacts, not parameter count.
Baseline: full Keras `.h5` (weights + optimizer state).
Hybrid: stripped `.h5` + SVM `.pkl`.

These results suggest that frozen feature extraction with SVM classification is a viable alternative to full fine-tuning, achieving higher accuracy while requiring substantially less training time.

3.2 Statistical Validation

To confirm that the observed performance differences exceed random variation, independent samples t-tests were conducted on all metrics across the five independent runs per model. Table 2 summarises the results.

Statistically significant differences were found for accuracy, macro F1-score, weighted F1-score, and training time, all at $\alpha = 0.05$. The negative t-values for classification metrics reflect the direction of comparison (Baseline − Hybrid), confirming the Hybrid model's consistent superiority in classification performance. The training time result confirms the Baseline's substantially longer mean training duration.

Inference time did not yield a statistically significant difference ($p = .443$), indicating that the per-image prediction latency of the two models cannot be reliably distinguished across runs under the current experimental conditions.

Table 2: Independent Samples t-Test Results ($\alpha = 0.05$, two-tailed)

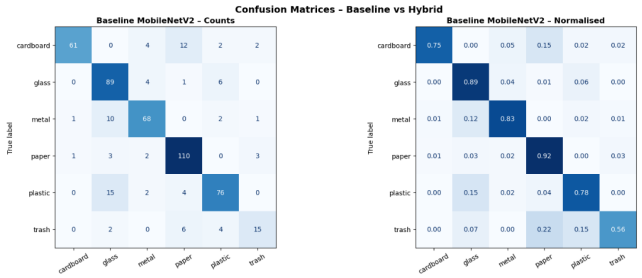
Variable	t	df	p-value	Interpretation
Accuracy (%)	−5.41	4	.006	Significant
Macro F1-Score	−4.35	4	.012	Significant
Weighted F1-Score	−4.71	4	.009	Significant
Inference Time	−0.81	8	.443	Not significant
Training Time (s)	15.91	8	$< .001$	Significant

Negative t = Baseline − Hybrid; Hybrid outperforms on classification metrics.
df = 4: Welch correction applied (unequal variances); df = 8: equal variances assumed.

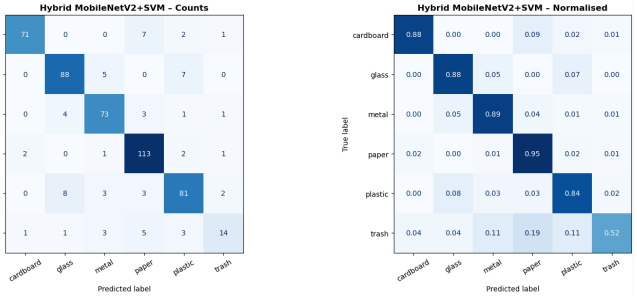
These findings confirm that the Hybrid model's superiority in classification performance is consistent and not attributable to random variation across runs.

3.3 Confusion Matrix Analysis

Figure 2 presents the confusion matrices for both models from Run 1 (Baseline: 82.81%; Hybrid: 86.96%). The Hybrid model correctly classifies 440 of 506 validation images (66 errors) versus 419 correct for the Baseline (87 errors).



(a) Baseline MobileNetV2 (82.81%)



(b) Hybrid MobileNetV2+SVM (86.96%)

Figure 2: Confusion matrices from Run 1. Diagonal cells indicate correct classifications.

Both models exhibit the lowest recall on the *trash* category (Baseline: 0.556; Hybrid: 0.519), reflecting its small training set (110 samples, 5.4% of training data) and visual ambiguity as a catch-all class.

The most frequent misclassification for trash is confusion with paper in both models. The Hybrid model improves recall most substantially for cardboard (+0.12) compared to the Baseline, attributable to the RBF kernel's non-linear decision boundary in the 1,280-dimensional feature space.

Overall, the Hybrid model demonstrates more reliable classification across well-represented categories, while both models struggle with the visually ambiguous trash class.

3.4 Per-Class Performance

Table 3 details per-class precision, recall, and F1-score for both models on Run 1. The Hybrid model improves F1 in five of six categories, with the largest gains in *cardboard* (+0.07) and *paper* (+0.03). *Trash* is the sole category where the Hybrid underperforms marginally (F1: 0.609 vs. 0.625), consistent with the confusion matrix analysis above. The Baseline's glass misclassification pattern (30 false positives, predominantly from metal) is substantially reduced in the Hybrid (13 false positives). The *trash* category's low recall in both models (Baseline: 0.556; Hybrid: 0.519) is consistent with its limited class support—only 110 training samples representing 5.4% of the training set—and its inherent visual ambiguity as a catch-all category. Addressing this through targeted data collection or hierarchical classification is left to future work.

Table 3: Per-Class Performance — Run 1

	Baseline			Hybrid		
Class	**P**	**R**	**F1**	**P**	**R**	**F1**
Cardboard	0.968	0.753	0.847	0.959	0.877	0.916
Glass	0.748	0.890	0.813	0.871	0.880	0.876
Metal	0.850	0.829	0.840	0.859	0.890	0.874
Paper	0.827	0.924	0.873	0.863	0.950	0.904
Plastic	0.844	0.784	0.813	0.844	0.835	0.839
Trash	0.714	0.556	0.625	0.737	0.519	0.609

P = Precision, R = Recall. Results from Run 1.

These per-class results reinforce that the Hybrid model's accuracy advantage is distributed across most categories rather than driven by a single dominant class.

3.5 Computational Efficiency and Resource Utilization

Table 1 summarises the computational trade-offs between the two approaches. The Baseline requires a mean training time of 6,737 s across five runs, while the Hybrid completes training in a mean of 657 s—a reduction of approximately 89%. This difference reflects the architectural distinction between the two approaches: the Baseline performs gradient descent through the full network across 40 epochs, whereas the Hybrid trains only an SVM classifier on pre-extracted features. On-disk model size is 27.97 MB for the Baseline (full Keras `.h5` including optimizer state) versus 25.67 MB for the Hybrid (stripped feature-extractor `.h5` plus SVM `.pkl`). The 2.3 MB difference is attributable to serialisation choices rather than a reduction in backbone parameters, since both models use the same frozen MobileNetV2 backbone.

Inference time for the Hybrid (223.03 ms per image, mean) and the Baseline (154.72 ms) differ numerically; however, the independent samples t-test found this difference to be statistically non-significant [$t(8) = -0.81$, $p = .443$]. High run-to-run variability in single-image latency measurements—which are sensitive to system load, memory state, and background processes—likely accounts for the inconclusive result. Inference time comparisons should therefore be interpreted with caution, and benchmarking on dedicated edge hardware under controlled conditions is recommended before drawing deployment conclusions from latency figures.

Overall, the Hybrid approach offers a practical advantage in training efficiency and classification accuracy, though inference latency and generalisability to other hardware environments require further investigation.

3.6 Practical Deployment Implications

The Hybrid model's statistically validated accuracy advantage and dramatic reduction in training time have concrete implications for resource-constrained deployment. In industrial sorting facilities, the 6.68 percentage point accuracy gain translates directly to fewer misclassified items and reduced contamination rates in recycled material streams. The 89% reduction in training time enables rapid model updates when waste stream composition shifts seasonally or by region, without requiring dedicated GPU infrastructure.

For edge and mobile deployments, the 25.67 MB on-disk footprint is compatible with memory-constrained devices. However, the 223.03 ms per-image inference time—which includes a full MobileNetV2 forward pass—should be assessed against the latency requirements of specific hardware targets. Model quantisation or pruning of the backbone could reduce inference latency further without retraining the SVM component, and is identified as a direction for future work.

The *trash* category's low recall (0.52) is the principal practical limitation: in deployment, approximately half of genuine trash items would be missed by the Hybrid model. Systems intended to handle ambiguous or mixed-material waste streams should incorporate a fallback mechanism—such as a confidence threshold or a secondary human review queue—for items classified into low-confidence categories.

4 CONCLUSION

This study demonstrates that combining frozen MobileNetV2 feature extraction with an SVM classifier achieves higher classification accuracy and substantially reduced training time compared to end-to-end fine-tuning for waste classification on TrashNet. Independent samples t-tests confirm that the differences in classification performance and training time are statistically significant, validating that the observed improvements are consistent across independent runs and not attributable to random variation. The difference in on-disk model size between the two approaches is explained by serialisation choices rather than a reduction in backbone parameters.

The *trash* category represents the principal limitation of both models, driven by its limited class support and inherent visual ambiguity. Inference time did not reach statistical significance, and results should be interpreted with caution given that all experiments

were conducted on a single laptop without dedicated deployment hardware.

4.1 Future Work

Cross-dataset evaluation represents the most immediate priority. All experiments in this study were conducted on TrashNet, a publicly validated benchmark that reflects controlled laboratory imaging conditions. Evaluating the proposed hybrid approach on larger and more diverse datasets such as TACO [6] or WasteNet would determine whether the observed accuracy advantage generalises across different waste compositions, geographic contexts, and real-world imaging conditions including variable lighting and partial occlusions.

A structured ablation study is also needed to strengthen the causal interpretation of the results. This study compares two complete architectural configurations but does not isolate the contribution of individual design decisions, such as the choice of RBF versus linear kernel, frozen extraction versus a frozen backbone with a softmax head, or the effect of SVM hyperparameter tuning independently of feature quality. Systematic decomposition of these factors would clarify which components drive the accuracy advantage and guide more principled architecture selection in future deployments.

The *trash* category's consistently low recall ($\approx$0.52–0.56 across both models) warrants dedicated treatment in future work. Its limited class support—110 training samples representing 5.4% of the training set—combined with its inherent visual ambiguity as a catch-all category makes it resistant to standard classification approaches. Targeted data collection, hierarchical classification schemes that treat trash as an outlier class, or active learning strategies that prioritise informative trash samples are promising directions for improving minority class performance.

Finally, reducing the Hybrid model's inference latency is important for real-time edge deployment. Backbone quantisation and structured pruning could decrease the per-image forward pass time without requiring retraining of the SVM component, preserving the training efficiency advantage of the hybrid architecture. Field validation under operational conditions, including embedded hardware benchmarking, would confirm the practical deployment readiness of the approach.

REFERENCES

[1] MEP Agus and M Muljono. 2024. Application of MobileNetV2 and SVM Combination for Enhanced Accuracy in Pneumonia Classification. *Journal of Applied Informatics and Computing* 8, 2 (2024), 145–158.

[2] M Chilakarao, SK Behera, KP Raju, PP Dash, PK Sethy, and A Nanthaamornphong. 2025. Domain-Aligned Feature Extraction with MobileNetV2-GRL and KNN For Wheat Leaf Disease Detection. *Results in Engineering* 21 (2025), 101923.

[3] LTT Hong, DQ Tu, ND Do, and NS Huy. 2024. Coffee leaf rust disease detection using MobileNetV2-based feature extractor, SVM classifier and visualization technique. *Journal of Military Science and Technology* 89 (2024), 78–88.

[4] AG Howard, M Zhu, B Chen, D Kalenichenko, W Wang, T Weyand, M Andreetto, and H Adam. 2017. MobileNets: Efficient convolutional neural networks for mobile vision applications. In *arXiv preprint arXiv:1704.04861*.

[5] AM Jajere, KH Bello, Q Li, NC Okanda, Y Ayesha, F Muhammad, and DS Samaila. 2024. Efficient and Accurate Brain Tumor Classification Using Hybrid MobileNetV2-Support Vector Machine for Magnetic Resonance Imaging Diagnostics in Neoplasms. *Brain Sciences* 14, 9 (2024), 912.

[6] Pedro F. Proença and Pedro Simões. 2020. TACO: Trash Annotations in Context. arXiv:2003.06975

[7] P Tavana, M Akraminia, A Koochari, and A Bagherifard. 2023. Classification of spinal curvature types using radiography images: Deep learning versus classical methods. *Artificial Intelligence Review* 56, 9 (2023), 9435–9466.

[8] Gary Thung and Mindy Yang. 2016. TrashNet: Classification of Trash. https://github.com/garythung/trashnet

[9] M Yildirim, A Çinar, and E Cengİl. 2022. Classification of the weather images with the proposed hybrid model using deep learning, SVM classifier, and mRMR feature selection methods. *Geocarto International* 37, 25 (2022), 10673–10693.

[10] X Zhang, L Yan, B Abuhaija, and B Ihnaini. 2025. MCP-X: An Ultra-Compact CNN for Rice Disease Classification in Resource-Constrained Environments. *AgriEngineering* 7, 1 (2025), 89–103.

Toward Multimodal Monitoring of Systemic Lupus Erythematosus: A Swin-SNN Cross-Modal Attention Approach

Thomas Danjo Manulat
Cebu Institute of Technology – University
College of Computer Studies
Cebu City, Philippines
thomasdanjo.manulat@cit.edu

Kaye Aizerner Evangelista
Cebu Institute of Technology – University
College of Computer Studies
Cebu City, Philippines
kayeaizerner.evangelista@cit.edu

Malt John Vianney C. Solon
Cebu Institute of Technology – University
College of Computer Studies
Cebu City, Philippines
maltjohnvianney.solon@cit.edu

Simon Lyster P. Escaño
Cebu Institute of Technology – University
College of Computer Studies
Cebu City, Philippines
simonlyster.escano@cit.edu

ABSTRACT

Systemic Lupus Erythematosus (SLE) is a relapsing multisystem autoimmune disease whose heterogeneous presentation makes disease assessment and follow-up difficult. Type I interferon-associated transcriptomic signatures provide a strong molecular correlate of lupus activity, while ocular manifestations such as lupus retinopathy reflect systemic microvascular and inflammatory involvement. In this work, we present a monitoring-oriented multimodal proof of concept that combines ocular fundus imaging and interferon-signature transcriptomic features using a Swin Transformer image encoder, a Self-Normalizing Neural Network transcriptomic encoder, and bidirectional cross-modal attention. Because no public dataset contains true patient-level paired SLE fundus images and transcriptomic profiles, we construct a pseudo-paired benchmark by aligning proxy abnormal fundus images from ODIR-5K with SLE and control transcriptomic samples from ADEx GSE110169. On this constructed dataset, the bidirectional fusion model achieved higher accuracy and AUC than unimodal and simpler fusion baselines. Interpretability analyses further highlighted plausible retinal regions and interferon-related genes such as IFI6 and IFI44L. We emphasize that this study does not establish clinical diagnostic validity: due to proxy retinal labeling and synthetic pairing, the findings should be interpreted as exploratory evidence that multimodal fusion may be useful for future SLE monitoring support, particularly in systems designed to flag cases for closer clinical review.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; *Learning latent representations*; • **Applied computing** → *Health care information systems*;

KEYWORDS

systemic lupus erythematosus, multimodal learning, ocular fundus imaging, transcriptomics, cross-modal attention, Swin Transformer, self-normalizing neural network

1 INTRODUCTION

Systemic Lupus Erythematosus (SLE) is a chronic multisystem autoimmune disease characterized by loss of immune tolerance, autoantibody formation, immune-complex deposition, and highly heterogeneous clinical expression [12, 14]. Its manifestations range from constitutional symptoms and mucocutaneous lesions to nephritis, vasculitis, and neuropsychiatric involvement. This heterogeneity makes SLE difficult to recognize and follow in practice, especially when symptoms fluctuate over time and severity is not fully reflected by any single laboratory test or clinical sign [6, 10].

Among the biological signals most strongly associated with SLE activity is the type I interferon (IFN) signature. Interferon-stimulated genes such as *IFI6*, *IFI44L*, *ISG15*, *RSAD2*, and *MX1* are repeatedly reported as informative markers of immune dysregulation in lupus, and high IFN-signature activity has been linked to more active disease states [1, 3, 4, 16]. In parallel, ocular findings in SLE—particularly lupus retinopathy and related microvascular abnormalities—have been described as clinically meaningful reflections of systemic vascular injury and disease activity [11, 13, 19]. This suggests a potentially useful pairing between molecular inflammatory state and retinal phenotype.

The rationale for combining these two modalities is straightforward. Transcriptomic features offer a relatively direct representation of systemic immune activation, while fundus imaging provides a non-invasive view of tissue-level vascular and inflammatory manifestations. A multimodal system that jointly considers these signals may capture complementary information that neither modality expresses as strongly on its own. However, this possibility is difficult to test because no public dataset currently provides true patient-level paired SLE fundus images and transcriptomic profiles.

To address this, we propose a multimodal architecture that integrates these disparate data types. The framework employs a Swin Transformer as the image encoder to capture hierarchical spatial features from fundus photos, alongside a Self-Normalizing Neural Network (SNN) to process high-dimensional transcriptomic data. To facilitate the training and evaluation of this architecture in the absence of naturally paired samples, we develop a systematic pseudo-pairing protocol to construct a synthetic multimodal benchmark from independent clinical cohorts. We evaluate several

fusion strategies, demonstrating that a bidirectional cross-modal attention mechanism outperforms both unimodal and simpler fusion baselines on our constructed benchmark.

Accordingly, this paper makes four contributions. First, it presents a monitoring-oriented multimodal framework that combines retinal fundus features and interferon-signature transcriptomic features in a single learned representation. Second, it documents an explicit pseudo-pairing protocol that makes benchmark construction transparent rather than implicit. Third, it compares bidirectional attention against simpler fusion schemes and unimodal baselines using standardized quantitative metrics (e.g., AUC-ROC, F1-score) to establish a state-of-the-art baseline for this multimodal task. Fourth, it provides an interpretability framework through retinal saliency visualization and gene-importance ranking, offering insights into the model's decision-making process.

Through consultation with Dr. William Tiu (National Program Manager, Department of Health), the utility of this approach was refined to focus on monitoring-oriented support. Rather than a tool for primary diagnosis, this system is designed to flag cases of worsening inflammatory activity for clinical review. Consequently, we position this work as an exploratory proof of concept, emphasizing its role in disease tracking while explicitly acknowledging the constraints of the benchmark construction.

2 RELATED WORK

2.1 SLE Monitoring, Interferon Biology, and Disease Activity

A large body of lupus literature has emphasized the centrality of type I interferon signaling in disease pathogenesis and activity. Early transcriptomic studies showed that blood cells from patients with severe SLE exhibit a strong interferon-inducible gene-expression signature [1]. More recent work has continued to refine this view, showing that interferon-associated genes can support disease characterization, subtype separation, and activity-linked biomarker development [4, 16]. These findings also align with the therapeutic relevance of interferon-targeting strategies in lupus [3, 7]. From a monitoring perspective, this literature motivates the use of transcriptomic interferon signatures as a biologically grounded source of signal for assessing inflammatory status.

2.2 Ocular Manifestations of SLE and Oculomics

The eye has increasingly been discussed as a window into systemic disease, particularly through the emerging field of oculomics [18]. In SLE, ocular manifestations such as cotton-wool spots, retinal hemorrhages, vascular tortuosity, and vaso-occlusive changes have been associated with systemic activity and, in some studies, with more severe organ involvement [11, 13, 15, 19]. Lupus retinopathy is therefore relevant not only as an ophthalmic manifestation but also as a potential proxy for systemic vascular or inflammatory burden. At the same time, retinal findings are not specific to lupus and may overlap with abnormalities seen in diabetes, hypertension, and other vascular conditions. This lack of specificity is precisely why a retinal-only approach is insufficient and why cross-modal context matters.

2.3 Deep Learning for Retinal Imaging

Retinal image analysis has become one of the most visible success results of deep learning in medicine. CNN-based systems have shown strong performance in diabetic retinopathy and related tasks, while newer transformer-based models have improved the modeling of global spatial dependencies in fundus and OCT images [8, 9]. The Swin Transformer is especially attractive for retinal analysis because its shifted-window attention mechanism supports hierarchical feature learning without sacrificing larger-context reasoning [9].

2.4 Multimodal Fusion of Imaging and Molecular Data

Multimodal machine learning seeks to integrate heterogeneous data sources into a unified predictive representation [2]. In the medical domain, such fusion has demonstrated significant utility in radiogenomics, oncology, and survival modeling, where attention-based mechanisms often outperform naive concatenation by enabling cross-modal conditioning [5, 17]. However, the application of these techniques to SLE requires careful methodological contextualization. Given the current absence of naturally paired retinal and transcriptomic datasets, this study evaluates the efficacy of fusion architectures within a rigorously constructed pseudo-paired benchmark. This distinction is critical as we position the framework not as a final clinical diagnostic tool, but as a proof-of-concept demonstrating how cross-modal signals can be synthesized to support longitudinal disease monitoring.

3 METHODS

3.1 Overview and Design Rationale

The study uses a cross-sectional, constructed multimodal benchmark derived from two independent public sources: transcriptomic samples from ADEx GSE110169 and retinal fundus images from ODIR-5K. This design was adopted because no public dataset with patient-level paired SLE fundus and transcriptomic data was available.

3.2 Transcriptomic Branch

The transcriptomic source was the ADEx autoimmune-disease resource, using the GSE110169 SLE subset. The working cohort contained 159 samples (82 SLE and 77 controls), and differential-expression analysis was used to identify the most informative interferon-related genes. A compact 20-gene signature was selected after significance filtering and effect-size ranking, with genes such as *IFI6*, *OAS1*, *IFI44L*, *EPSTI1*, *ISG15*, and *RSAD2* appearing prominently. This compact signature was chosen because it balances biological relevance, small-sample stability, and feasibility for neural modeling.

Several gene-only classifiers were benchmarked, including Logistic Regression, SVM, Random Forest, XGBoost, MLP, and a Self-Normalizing Neural Network. Although tree-based methods slightly outperformed the SNN as standalone gene models, the SNN was retained for fusion because it is compatible with end-to-end gradient-based multimodal training and showed competitive AUC and low variability in the original evaluation.

3.3 Retinal Imaging Branch

The imaging source was ODIR-5K, a public fundus dataset that does not contain a verified SLE retinal class. To construct a proxy task, the original setup grouped fundus images labeled as diabetes, hypertension, or other abnormalities into an abnormal class, while healthy fundus images were grouped into a normal class. We make this limitation explicit: this branch does not model lupus retinopathy directly. It models a proxy abnormal-versus-normal retinal distinction that was originally intended to approximate potentially relevant vascular pathology.

All images were resized to 224 x 224 pixels. Subsequently, pixel values were transformed into tensors and normalized using standard ImageNet statistics, specifically a mean of (0.485, 0.456, 0.406) and a standard deviation of (0.229, 0.224, 0.225), with training-time augmentation applied for robustness. The benchmarking compared ResNet50, DenseNet121, EfficientNet-B0, and Swin-T. Swin-T was selected as the retained image encoder because it achieved the highest image-only test accuracy, F1-score, and AUC among the candidate backbones.

3.4 Pseudo-Pairing Strategy

Because the image and gene datasets come from independent cohorts, no true patient-level correspondence exists between modalities. The study therefore used class-consistent pseudo-pairing, in which transcriptomic SLE samples were paired with proxy abnormal fundus images and control transcriptomic samples were paired with normal fundus images. This procedure creates a benchmark that allows multimodal model comparison.

3.5 Fusion Architecture

The proposed fusion system combines Swin-T image embeddings and SNN transcriptomic embeddings using bidirectional cross-modal attention. The experiments also evaluated simple concatenation and co-attention. Conceptually, the bidirectional module allows image features to attend to gene features and gene features to attend to image features before the fused representation is passed to the final classifier, the best-performing fusion design in the experimentation.

3.6 Evaluation and Interpretation

3.7 Evaluation and Interpretation

We evaluate the predictive performance of the proposed framework against a comprehensive suite of unimodal and multimodal baselines. The unimodal benchmarks consist of the image-only Swin-T and the gene-only SNN. Multimodal comparisons include simple feature concatenation, co-attention, and the proposed bidirectional attention mechanism. All models are assessed using Accuracy and Area Under the ROC Curve (AUC) to capture both classification reliability and discriminative capacity across the heterogeneous inputs.

Beyond quantitative metrics, we establish model transparency through a dual-modal interpretability analysis. For the imaging modality, we utilize Grad-CAM to generate saliency maps, identifying localized retinal features influential to the model's decision. For the transcriptomic modality, we employ permutation-based feature importance to rank the contribution of specific interferon-stimulated genes. These qualitative assessments serve as a biological plausibility check, ensuring the model's focus aligns with established SLE pathophysiology. While these findings offer insights into cross-modal synergy, they are interpreted within the context of the constructed pseudo-paired benchmark to maintain a grounded assessment of the system's current monitoring capabilities.

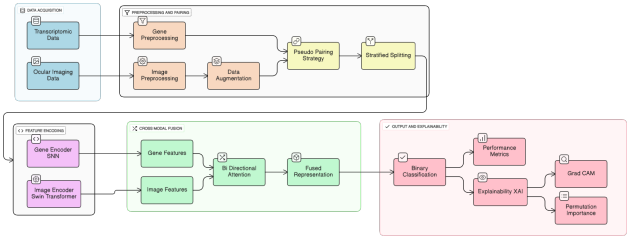


Figure 1: Overview of the proposed multimodal pipeline. Transcriptomic data undergoes preprocessing and selection of interferon-related gene features, while ocular fundus images from ODIR-5K undergo image preprocessing before feature extraction through a Swin Transformer. The transcriptomic branch is encoded using a Self-Normalizing Neural Network (SNN), and the resulting visual and gene representations are integrated through bidirectional cross-modal attention fusion for final classification.

Table 1: Main performance comparison retained from the original study.

Model	Accuracy	AUC
Image-only (Swin-T)	0.7185	0.7712
Gene-only (SNN)	0.8300	0.9300
Simple Concatenation	0.8859	0.9566
Co-Attention	0.8763	0.9477
Bidirectional Attention	**0.8962**	**0.9636**

4 RESULTS AND DISCUSSION

4.1 Main Benchmark Results

Table 1 summarizes the main retained results. On the constructed benchmark, the bidirectional attention model achieved the highest performance, with 0.8962 accuracy and 0.9636 AUC. Simple concatenation and co-attention also performed strongly, while the gene-only branch substantially outperformed the image-only branch. This pattern is consistent with the biological strength of interferon-signature features and with the fact that the retinal labels are only proxies rather than verified SLE ocular labels.

These values should be interpreted conservatively. The gain of the fusion models over the unimodal branches suggests that cross-modal interaction is useful within this synthetic benchmark, but it does not establish patient-level clinical coupling between retinal phenotype and transcriptomic state. The claim is therefore architectural: the fusion design appears to organize heterogeneous

proxy signals more effectively than the retained unimodal and simpler fusion alternatives.

4.2 Modality Contribution and Ablation Analysis

To quantify the relative influence of heterogeneous inputs on model performance, we conduct an ablation study across the primary fusion strategies. As shown in Table 2, the transcriptomic (gene) modality emerges as the dominant predictive component within this benchmark. For example, in the bidirectional attention model, removing the retinal signal results in a performance decrease of 6.8%, whereas removing the transcriptomic signal leads to a more substantial 26.1% drop.

Table 2: Ablation analysis of modality contributions (AUC).

Fusion Strategy	Full Model	Gene-only	Image-only
Bidirectional Attention	**0.8962**	0.8353	0.6622
Co-Attention	0.8763	0.8167	0.6827
Simple Concatenation	0.8859	0.8340	0.6660

This performance asymmetry suggests that the current framework leverages the transcriptomic branch as the primary signal, while the retinal branch provides auxiliary, albeit complementary, features.

4.3 Interpretability

The experiments used Grad-CAM on the retained retinal encoder and permutation-based importance for the gene branch. Representative retinal heatmaps showed that the model focused on localized regions that appeared visually plausible as abnormal fundus structures, rather than distributing attention uniformly across the image. Likewise, gene-importance ranking repeatedly emphasized interferon-associated genes such as *IFI6* and *IFI44L*, which are well aligned with prior lupus biology [1, 3, 16].

These interpretability outputs are useful because they show that the model is not obviously behaving in a biologically arbitrary way. In the same way, high ranking of canonical interferon genes strengthens internal consistency.

4.4 Scope of Inference

The experimental findings support a specific and well-defined conclusion regarding multimodal synergy in SLE. This study demonstrates that a Swin-SNN fusion architecture, utilizing bidirectional cross-modal attention, consistently outperforms unimodal and baseline fusion methods within a pseudo-paired benchmark environment. The results indicate that the transcriptomic signature serves as a robust primary signal, while retinal proxy features provide additive contextual value when integrated through cross-modal conditioning.

Crucially, these findings are presented as a proof-of-concept for monitoring-oriented support. By explicitly defining these boundaries, we ensure that the performance of the framework is interpreted within the methodological constraints of the constructed benchmark while establishing a foundation for future research involving naturally paired clinical cohorts.

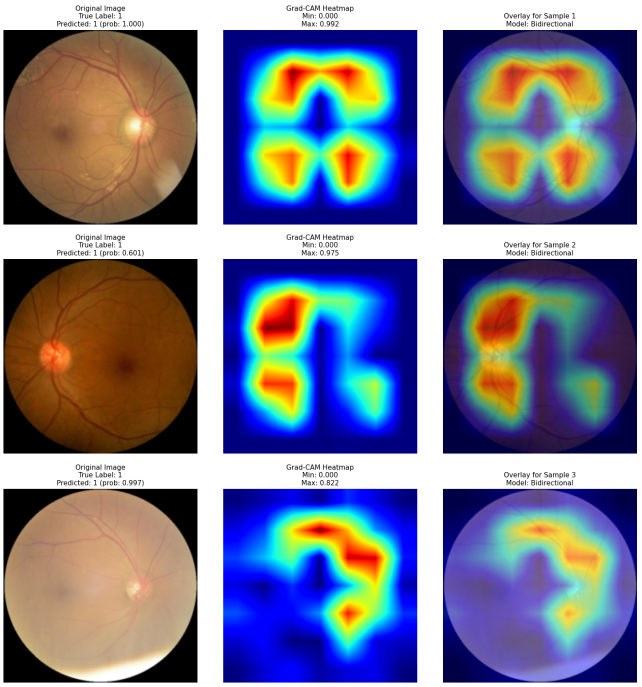


Figure 2: Grad-CAM visualizations of representative ocular fundus images and the corresponding activation heatmaps produced by the best-performing multimodal model. The highlighted regions indicate retinal areas that contributed most strongly to the model's prediction, providing qualitative insight into the visual patterns used for SLE-related classification.

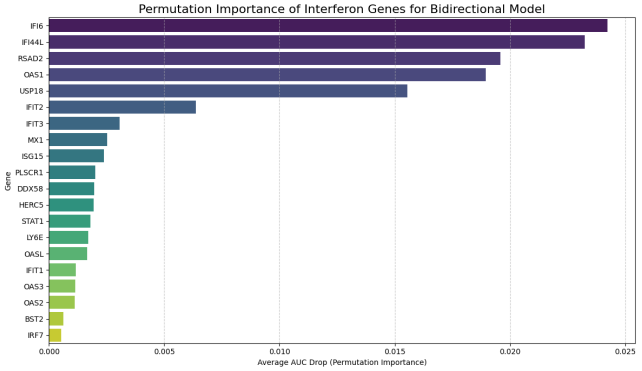


Figure 3: Permutation-based gene importance analysis for the multimodal fusion models. The ranked features highlight the relative contribution of dominant interferon-related genes to model performance across different fusion variants, providing interpretability for the transcriptomic branch of the framework.

5 LIMITATIONS AND CONCLUSION

This paper should be read as an exploratory, monitoring-oriented proof of concept. The retinal branch relies on proxy abnormality labels from ODIR-5K rather than true SLE fundus images. The multimodal benchmark is built through synthetic pseudo-pairing rather than patient-level correspondence.

Within those boundaries, the study still contributes a coherent multimodal framework, a transparent benchmark-construction protocol, and evidence that bidirectional cross-modal attention performs best among the retained baselines on the constructed dataset. The most defensible interpretation is therefore that multimodal fusion of retinal and interferon-signature information is promising for future monitoring support in SLE, particularly for flagging cases that may warrant closer review. Future work should validate this idea using true paired SLE fundus and transcriptomic data and, ideally, longitudinal cohorts where disease activity changes can be measured directly.

REFERENCES

[1] Edward C. Baechler, Farzin M. Batliwalla, George Karypis, Patrick M. Gaffney, Kimberly Moser, William A. Ortmann, Kristin J. Espe, Kathleen B. Shark, William J. Grande, Kelly M. Hughes, et al. 2003. Interferon-inducible gene expression signature in peripheral blood cells of patients with severe lupus. *Proceedings of the National Academy of Sciences* 100, 5 (2003), 2610–2615. doi:10.1073/pnas.0337679100

[2] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2019. Multimodal Machine Learning: A Survey and Taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 2 (2019), 423–443. doi:10.1109/TPAMI.2018.2798607

[3] Santiago Bruera, Tapiwa Chavula, Rahul Madan, and Shikha K. Agarwal. 2023. Targeting type I interferons in systemic lupus erythematosus. *Frontiers in Pharmacology* 13 (2023), 1046687. doi:10.3389/fphar.2022.1046687

[4] Vanessa Demers-Mathieu. 2023. Optimal Selection of IFN-α-Inducible Genes to Determine Type I Interferon Signature Improves the Diagnosis of Systemic Lupus Erythematosus. *Biomedicines* 11, 3 (2023), 864. doi:10.3390/biomedicines11030864

[5] Ruopeng Deng, Nadia Shaikh, Grace Shannon, and Yinan Nie. 2024. Cross-modality Attention-based Multimodal Fusion for Non-small Cell Lung Cancer Patient Survival Prediction. *arXiv preprint arXiv:2308.09831* (2024). doi:10.48550/arXiv.2308.09831

[6] Laura Durcan, Triona O'Dwyer, and Michelle Petri. 2019. Management strategies and future directions for systemic lupus erythematosus in adults. *The Lancet* 393, 10188 (2019), 2332–2343. doi:10.1016/S0140-6736(19)30237-5

[7] Richard Furie et al. 2017. Anifrolumab, an Anti-Interferon-α Receptor Monoclonal Antibody, in Moderate-to-Severe Systemic Lupus Erythematosus. *New England Journal of Medicine* 376, 3 (2017), 211–221. doi:10.1056/NEJMoa1614052

[8] Varun Gulshan, Lily Peng, Marc Coram, Martin C. Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, et al. 2016. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. *JAMA* 316, 22 (2016), 2402–2410. doi:10.1001/jama.2016.17216

[9] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10012–10022. doi:10.1109/ICCV48922.2021.00986

[10] Wei Lu et al. 2022. Utility of the ACR-1997, SLICC-2012 and EULAR/ACR-2019 classification criteria for systemic lupus erythematosus: a single-centre retrospective study. *Lupus Science & Medicine* 9, 1 (2022), e000718. doi:10.1136/lupus-2022-000718

[11] Wojciech Luboń, Monika Luboń, Piotr Kotyla, and Ewa Mrukwa-Kominek. 2022. Understanding Ocular Findings and Manifestations of Systemic Lupus Erythematosus: Update Review of the Literature. *International Journal of Molecular Sciences* 23, 20 (2022), 12264. doi:10.3390/ijms232012264

[12] Vasiliki R. Moulton, Ana Suarez-Fueyo, Esther Meidan, Hong Li, Masayuki Mizui, and George C. Tsokos. 2017. Pathogenesis of Human Systemic Lupus Erythematosus: A Cellular Perspective. *Trends in Molecular Medicine* 23, 7 (2017), 615–635. doi:10.1016/j.molmed.2017.05.006

[13] Mohammed Musa et al. 2024. Unveiling Ocular Manifestations in Systemic Lupus Erythematosus. *Journal of Clinical Medicine* 13, 4 (2024), 1047. doi:10.3390/jcm13041047

[14] Marilia Postal, José F. Vivaldo, Raquel Fernandez-Ruiz, Juan L. Paredes, Simone Appenzeller, and Timothy B. Niewold. 2020. Type I interferon in the pathogenesis of systemic lupus erythematosus. *Current Opinion in Immunology* 67 (2020), 87–94. doi:10.1016/j.coi.2020.10.014

[15] Gaurav Seth et al. 2018. Lupus retinopathy: a marker of active systemic lupus erythematosus. *Rheumatology International* 38, 8 (2018), 1495–1501. doi:10.1007/s00296-018-4083-4

[16] Min Shen et al. 2022. Identification of key interferon-stimulated genes for indicating the condition of patients with systemic lupus erythematosus. *Frontiers in Immunology* 13 (2022), 962393. doi:10.3389/fimmu.2022.962393

[17] Kai Sun et al. 2024. CMAF-Net: a cross-modal attention fusion-based deep neural network for incomplete multi-modal brain tumor segmentation. *Quantitative Imaging in Medicine and Surgery* 14, 7 (2024), 4579–4604. doi:10.21037/qims-24-9

[18] Siegfried K. Wagner et al. 2020. Insights into Systemic Disease through Retinal Imaging-Based Oculomics. *Translational Vision Science & Technology* 9, 2 (2020), 6. doi:10.1167/tvst.9.2.6

[19] Xiaowen Wang, Haizhou Xie, Yimin Yi, Jiahui Zhou, Hui Yang, and Jian Li. 2023. Clinical Research of Lupus Retinopathy: Quantitative Analysis of Retinal Vessels by Optical Coherence Tomography Angiography in Patients with Systemic Lupus Erythematosus. *Diagnostics* 13, 20 (2023), 3222. doi:10.3390/diagnostics13203222

Solving the Senior High School Timetabling Problem using a Multi-Agent Reinforcement Learning Approach

Alroy Leon Reyes
De La Salle University
Manila, Philippines
alroy_reyes@dlsu.edu.ph

Gabriel Ocampo
De La Salle University
Manila, Philippines
gabriel_ocampo@dlsu.edu.ph

Vhonne Cuya
De La Salle University
Manila, Philippines
vhonne_cuya@dlsu.edu.ph

Rafael Cabredo
De La Salle University
Manila, Philippines
rafael.cabredo@dlsu.edu.ph

ABSTRACT

Developing faculty timetables in educational institutions is a complex and time-consuming process requiring consideration of many constraints, such as faculty availability, departmental requirements, and resource limitations. This paper presents a Multi-Agent Reinforcement Learning (MARL) solution for a senior high school timetabling system in the Philippines. Using Subject Area Head Agents (SAHAs) trained through Proximal Policy Optimization (PPO) with centralized training and decentralized execution, we automate the scheduling process while handling strand-specific requirements across multiple campuses. Our experimental results demonstrate that the system achieved a peak subject placement rate of 82.1% and maintained strong compliance with teacher-related hard constraints (Teacher Perspective Score stabilized consistently above 93/100). However, the policy struggled with student schedule coherence, resulting in a critical section conflict rate of 40–60% (Student Perspective Score stabilized around 50–55/100). The system, implemented using Ray RLlib and PettingZoo, demonstrates MARL's potential for partial automation of educational scheduling while highlighting critical challenges requiring further research for production deployment.

KEYWORDS

Academic timetabling, multi-agent, reinforcement learning

1 Introduction

Academic timetabling in educational institutions is traditionally managed through manual or heuristic approaches, leading to several inefficiencies, such as difficulty in handling large numbers of constraints, labor-intensive processes, and suboptimal utilization of resources. These methods struggle to scale and to adapt to real-time changes such as last-minute teacher absences, room conflicts, or new subjects introduced into the schedule [1]-[4]. Prior work shows metaheuristics (e.g., genetic algorithms, simulated annealing, tabu search) can produce feasible schedules, but often require heavy tuning and lack responsiveness under evolving institutional constraints [3], [5]. Multi-agent designs for timetabling have also been explored to improve flexibility and conflict resolution, yet many remain rule-based and brittle to distribution shifts [6]-[8].

Our work aims to modernize the timetabling process by introducing a Multi-Agent Reinforcement Learning (MARL) approach. Unlike traditional methods, our system leverages Proximal Policy Optimization (PPO) to enable decentralized decision-making through agents that collaboratively optimize the timetable [9], [10]. Concretely, we implement and scale training with Ray RLlib [11], [12] and model the interaction dynamics using PettingZoo's AEC API [13], [14]. This combination gives us (i) centralized training with decentralized execution (CTDE) for stability, (ii) legality/action masks to enforce hard constraints, and (iii) reproducible, cluster-ready training for large action spaces.

In our system, Subject Area Head Agents (SAHAs) handle teacher-to-subject and timeslot assignments under expertise, availability, and load constraints. The primary contribution of this paper is a comprehensive evaluation of MARL's viability

for real-world educational timetabling in Philippine senior high school contexts. Through implementation of a custom MARL framework using Ray RLlib and PettingZoo on authentic institutional data (598 subjects, 257 teachers, 390 sections across two campuses with strand-specific requirements), we demonstrate both the approach's strengths in teacher constraint satisfaction (93-95% success rate) and its critical limitations in student schedule coherence (40-60% section conflict rates). Our findings provide crucial insights into MARL's current capabilities and fundamental challenges for production deployment, revealing that while the approach shows promise for partial automation, significant improvements in agent coordination and reward engineering are necessary before practical adoption.

2 Related Works

2.1 Reinforcement Learning in Timetabling

Reinforcement Learning (RL) has been widely explored for scheduling and timetabling due to its ability to adapt and optimize decision-making policies dynamically. Single-agent RL methods (e.g., Q-learning) have been applied to simpler scheduling problems, but they struggle to scale to complex, multi-agent settings [1]. In contrast, Multi-Agent Reinforcement Learning (MARL) enables multiple agents to collaborate and learn autonomously, improving scalability and flexibility. For instance, [8] applied a multi-agent approach to higher-education timetabling where agents representing faculty and rooms negotiated schedules. Our work builds on these ideas but targets senior high school (SHS) timetabling, where strand-specific requirements add additional constraints.

2.2 Multi-Agent Systems in Scheduling

In scheduling, Multi-Agent Systems (MAS) decompose the task into smaller subproblems, each managed by a specialized agent. This is effective in large institutions where teachers, rooms, and subjects must coordinate [7]. [6] demonstrated a multi-agent model for personalized, real-time timetable adjustments, showing that autonomous agents can efficiently handle dynamic changes such as teacher unavailability or room reassignments—conditions common in SHS contexts.

2.3 Traditional Genetic Algorithm Approaches

Previous work has employed Genetic Algorithms for university-level timetabling. Feasible timetables for a university computer studies department were developed by [4] after 3,000–4,000 generations, achieving zero hard constraint violations but experiencing stagnation after convergence. Similarly, an automated scheduling system using GA could produce comparable results with similar convergence patterns [15]. While these approaches demonstrated effectiveness, they required extensive parameter tuning and struggled with adaptability to dynamic changes.

2.4 Ray RLib and PettingZoo

PettingZoo standardizes multi-agent environments with two APIs—AEC for turn-based and Parallel for simultaneous actions—plus strict versioning and action masking, improving reproducibility and constraint handling in domains like timetabling [13], [14], [16], [17]. Ray RLlib provides the complementary training stack: distributed rollouts (EnvRunners), distributed learners, and a *MultiAgentEnv* API with flexible agent→policy mappings for independent, shared-parameter, team, and adversarial setups [11], [12], [18].

The two interoperate via PettingZooEnv or ParallelPettingZooEnv wrappers, so PettingZoo tasks can be trained at scale with PPO and CTDE in RLlib without custom glue code [19], [20]. Prior reports use this stack for resource allocation and scheduling, where legality masks and centralized critics reduce invalid actions and non-stationarity [21], [22].

3 Methodology

3.1 Problem Definition

The goal of the timetabling problem is to allocate:

1. **Teachers** to **Subjects**, ensuring each subject is taught by the appropriate teacher.
2. **Timeslots** to **Subjects**, ensuring that there are no time conflicts for teachers or sections.
3. **Rooms** to **Subjects**, ensuring that each class has a suitable room based on subject requirements and room capacity.

3.2 MARL Framework

Our MARL approach employs **Subject Area Head Agents (SAHAs)**, responsible for assigning teachers to subjects and timeslots. Each agent is trained using the Proximal Policy Optimization (PPO) algorithm. The PettingZoo Agent Environment Cycle environments (AECEnv) allows agents to interact and make decisions based on the current state, receiving rewards for valid assignments.

The MARL framework operates as a coordinated system where SAHAs interact through a shared PettingZoo AECEnv environment. Each agent maintains its own PPO policy network while accessing a global state representation containing all current assignments, resource availability, and constraint violations. Agents operate in a turn-based cycle: when SAHA_i

takes its turn, it observes the current environment state (including assignments made by other agents), selects an action (teacher-subject-timeslot assignment), and receives an immediate reward signal based on constraint satisfaction. The environment then updates its global state and passes control to the next agent. This indirect communication mechanism through shared state enables agents to coordinate without explicit message passing, as each agent's observations include the consequences of all previous agents' actions. The PPO algorithm guides each agent toward maximizing its cumulative reward, which is designed to align individual agent objectives with the global goal of producing a conflict-free, optimal timetable.

3.3 Multi-Agent Policies

The agents learn policies based on shared Multi-Layer Perceptron ($128 \rightarrow 128$) neural networks to assign teachers to subjects and timeslots. These policies are decentralized, with each agent making decisions independently while coordinating with other agents to create a valid timetable.

3.4 PPO Training Setup

The agents are trained using Proximal Policy Optimization (PPO), a reinforcement learning algorithm that handles continuous action spaces and ensures stable training. The PPO algorithm is chosen for its stability and efficiency in environments like timetabling, where large action spaces and many interacting agents are involved.

3.4.1 Key Hyperparameters

The PPO training process requires careful tuning of several hyperparameters to balance exploration, exploitation, and training stability. We employed exponential decay for the learning rate and linear decay for the entropy coefficient to gradually shift from exploration to exploitation as training progressed. The following hyperparameters were selected based on preliminary experiments and standard PPO practices for multi-agent environments:

1. **Learning Rate**: 5×10^{-5} to 1×10^{-6} with exponential decay
2. **Train Batch Size**: 512
3. **SGD Minibatch Size**: 512
4. **Num SGD Iterations**: 4
5. **Clipping Parameter**: 0.2
6. **GAE Lambda**: 0.95
7. **Discount Factor**: 0.99
8. **Value Function Coefficient**: 3.0
9. **Entropy Coefficient**: 1.0 to 0.2 with linear decay
10. **Gradient Clipping**: 1.0
11. **Training Iterations**: 1000

3.4.2 Loss functions (PPO)

We use the clipped surrogate objective for the policy and a clipped value error for the critic:

$$L_{CLIP}(\theta) = -E_t\left[min(r_t(\theta)\,\hat{A}_t,\ clip(r_t(\theta),\,1-\epsilon,\,1+\epsilon)\,\hat{A}_t)\right]$$

The probability ratio $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$. quantifies the change in action probability between the current policy $r_t(\theta)$ and the old policy $\pi_{\theta_{old}}$ for action a_t in state s_t. When $r_t(\theta)$ equals 1, the new policy assigns the same probability to the action as the old policy. Values greater than 1 indicate the new policy favors the action more, while values less than 1 suggest decreased preference. This ratio serves as a measure of policy divergence during the update process.

The advantage estimate $\hat{A}_t$ represents how much better action a_t performs compared to the average expected return from the state s_t. Positive advantages indicate actions that outperform the baseline, providing a signal to increase their probability. Negative advantages mark actions worse than average, signaling the policy to reduce their likelihood. The advantage thus guides the direction of policy improvement by identifying which actions deserve reinforcement.

The clipping mechanism constrains $r_t(\theta)$ within $[1-\epsilon,\,1+\epsilon]$, implementing PPO's core innovation of conservative policy updates. This constraint prevents excessively large policy changes that could destabilize the learning process. When an update would push the probability ratio beyond these bounds, the gradient is clipped, ensuring the policy remains within a trust region around the old policy. The minimum operation in the objective takes the more pessimistic of the clipped and unclipped objectives, further encouraging cautious updates. Without this mechanism, large policy shifts could cause catastrophic forgetting, where the agent loses previously learned effective behaviors.

Value Loss (Clipped Value Function)

The value function loss measures prediction accuracy with optional clipping:

$$L_{VF} = \frac{1}{2}\,E_t[(clip(R_t - V_{old}(s_t),\ -c_v,\ +c_v))^2]$$

where the return is calculated as:

$$R_t = r_t + \gamma\,V_{old}(s_{t+1})$$

The return R_t combines the immediate reward r_t with the discounted value of the next state, where γ is the discount factor determining how much the agent values future rewards versus immediate ones. This target represents what the value function should predict based on observed experience.

The clipping of the value prediction error to $[-c_v,\,+c_v]$ prevents excessively large value updates similar to the policy

clipping. When the value network's prediction $V_{old}(s_t)$ differs greatly from the observed return R_t, the error is bounded to maintain training stability. This is particularly important in the early stages of training when value estimates may be highly inaccurate. The squared error penalizes the clipped difference, creating a smooth optimization landscape for the value network. Accurate value predictions directly improve advantage estimates, as advantages are typically computed as $\hat{A}_t = R_t - V(s_t)$, making value function quality critical for effective policy learning.

Entropy Bonus

We include an entropy bonus to encourage exploration:

$$L_{ENT}(\theta) \;=\; -E_t[H(\pi_\theta(\cdot|\, s_t))].$$

The entropy $H(\pi_\theta(\cdot|\, s_t))$ measures the randomness or uncertainty in the policy's action distribution at state s. High entropy indicates a more uniform distribution over actions, meaning the agent explores diverse options rather than committing strongly to specific choices. Low entropy suggests a nearly deterministic policy focused on a small set of preferred actions. During early training, maintaining high entropy is crucial for discovering effective teacher-subject-timeslot assignments, as the agent must explore the vast space of possible scheduling combinations. As training progresses and the agent identifies successful patterns, entropy naturally decreases as the policy becomes more confident. However, the entropy bonus c_e prevents premature convergence to suboptimal deterministic policies by continuously rewarding some degree of randomness.

Overall Objective

The complete optimization objective minimized during training is:

$$\min_\theta \; L_{CLIP} + c_v\, L_{VF} + c_e\, L_{ENT},$$

with typical settings in our runs: $\epsilon = 0.2$, $c \equiv vf_clip_param = 0.2$, $c_v = 0.5$, $c_e = 0.002$

The coefficient $c_v = 0.5$ balances the importance of value function learning against policy improvement. Setting this value requires considering that accurate value estimates support better advantage calculations, but overemphasizing value learning can slow policy convergence. Our choice of 0.5 gives moderate weight to both objectives, allowing the value network to improve steadily while not dominating the policy updates.

The entropy coefficient $c_e = 0.002$ provides a small but meaningful exploration incentive. This relatively low value reflects that in our timetabling problem, we want agents to explore initially but gradually commit to high-quality assignments as they learn the constraint structure. Too high a value would prevent convergence to deterministic optimal policies, while too low would risk premature exploitation of suboptimal solutions.

The clipping parameter $\epsilon = 0.2$ defines a trust region allowing policy changes up to 20% in probability ratio before clipping activates. This moderately conservative setting permits meaningful learning progress in each update while preventing destabilizing policy shifts. Together with $vf_clip_param = 0.2$, these hyperparameters ensure stable multi-agent training where each SAHA learns effective assignment strategies without experiencing catastrophic performance degradation during the training process.

4 Data Preparation

4.1 Dataset Characteristics

The system was trained and evaluated on real senior high school data comprising:

- **Subjects**: 598 total subjects distributed across modalities
    - Face-to-Face (F2F): 176 subjects (29.4%)
    - Online: 102 subjects (17.1%)
    - Hybrid: 320 subjects (53.5%)
- **Teachers**: 257 faculty members with varying availability
- **Sections**: 390 student sections across 5 strands (STEM - Science/Math, ABM - Business, HUMSS - Humanities/Social Sciences, SPT - Sports Track, ADT - Arts and Design Track)
- **Campuses**: Two (2) campuses with different room configurations for lecture and laboratory classes

4.2 Data Sources and Schema

Sources:

- Institutional data exports per term: teachers, rooms, subjects, loads, timeslots
- Official institutional records: official strand sectioning and class sizes
- Coordinator forms: availability, preferences, special room/lab needs

Schema (concise):

Teachers: *id*, campus, expertise/area, FT/PT, max load, availability ($D \times S$), preferred/blocked slots.
Subjects: *code*, strand, hours/week, meeting pattern, modality (lecture/lab), section ids.
Rooms: *id*, building, campus, type (lecture/lab), capacity.
Timeslots: *id*, day, period, start/end, campus, blocked (e.g., lunch).
Sections: *id*, strand, year level, size, required subjects.

4.3 Reward Adjustment Strategy

The training employed a 4-phase progressive difficulty scaling approach:

- **Phase 1**: 0.5 constraint penalty (warm-up)
- **Phase 2**: 0.75 constraint penalty (gradual increase)
- **Phase 3**: 1.0 constraint penalty (full difficulty)
- **Phase 4**: 1.2 constraint penalty (over-constraint for robustness)

This approach was designed to progressively increase the difficulty of constraint satisfaction, allowing agents to first learn basic scheduling patterns before facing the full complexity of the timetabling problem.

5 Results and Discussion

5.1 Overview of Experimental Setup

The entire 598-subject dataset was used for both training and evaluation. During each training iteration, agents learned to schedule all subjects simultaneously, with performance metrics computed on the same complete dataset at regular checkpoints (every 10–50 iterations). This approach reflects the nature of the timetabling problem as a combinatorial optimization challenge where the objective is to find valid, high-quality solutions to a specific constraint satisfaction problem.

5.2 Subject Placement Performance

Table 1 presents the subject placement results across key training iterations. The placement rate represents the percentage of subjects successfully assigned to valid timeslots without violating hard constraints.

Table 1: **Subject Placement Performance**

Iter	Total	Rate	F2F	Online	Hybrid
10	391	65.4%	98.9%	37.3%	72.8%
50	461	77.1%	96.6%	52.9%	85.3%
100	459	76.8%	97.7%	47.1%	87.8%
150	440	73.6%	96.0%	23.5%	87.8%
200	485	**81.1%**	97.2%	66.7%	88.4%
450	483	80.8%	97.2%	71.6%	86.2%
800	463	77.4%	96.0%	55.9%	84.4%
1000	491	**82.1%**	93.2%	**82.4%**	**89.1%**

The system achieved its peak placement rate of 82.1% at iteration 1000, successfully scheduling 491 out of 598 subjects. This represents significant improvement from the initial 65.4% at iteration 10, though the progression was non-monotonic with notable fluctuations throughout training.

Face-to-face subjects maintained consistently high placement rates (93–99%), suggesting that agents quickly learned to handle spatial constraints associated with physical classroom assignments. The rigid constraints of F2F subjects actually simplified the optimization problem by reducing the valid action space and providing clearer learning signals. In contrast, online subjects showed dramatic variability from 23.5% to 82.4%, indicating difficulty in developing stable policies for temporally flexible subjects. When subjects lack spatial constraints, agents must rely more heavily on implicit coordination through shared environment state, which proves harder to learn. Hybrid subjects stabilized between 84–89%, with their partial spatial constraints providing sufficient structure for more consistent learning than purely online subjects.

The overall performance instability, particularly significant drops at iterations 150 and 800 despite earlier progress, indicates a struggle to converge to stable optima. This behavior is characteristic of multi-agent environments where each agent's policy updates change the environment from other agents' perspectives, creating non-stationary optimization dynamics that can disrupt previously learned coordination patterns.

5.3 Conflict Analysis

Table 2 presents conflict metrics across training iterations, categorized by conflict type.

Table 2: **Conflict Metrics Over Training**

Iter	Total	Teacher	Section	Sections Affected
10	111	30 (27%)	81 (73%)	62.1%
50	145	28 (19%)	117 (81%)	52.3%
100	120	18 (15%)	102 (85%)	47.4%
150	**106**	**13 (12%)**	**93 (88%)**	**40.8%**

Iter	Total	Teacher	Section	Sections Affected
200	171	25 (15%)	146 (85%)	60.0%
450	168	21 (13%)	147 (88%)	58.7%
800	143	16 (11%)	127 (89%)	52.6%
1000	173	24 (14%)	149 (86%)	59.5%

The conflict analysis reveals a critical dichotomy: while teacher-related conflicts remain manageable, section conflicts represent a fundamental barrier to deployment. Section conflicts consistently dominate at 73–89% of all conflicts, with 40–60% of student sections experiencing scheduling conflicts—meaning a substantial portion of students would receive infeasible timetables.

This section's conflict dominance stems from the agent decomposition strategy. Subject Area Head Agents naturally optimize for teacher constraints within their domain, as teachers are directly represented in each agent's observation space and reward function. However, section schedules emerge implicitly from collective agent decisions, with no single agent responsible for ensuring conflict-free schedules for any particular section. Agents must implicitly coordinate across subject areas to prevent overlapping classes for the same section, but this reward signal is diluted across multiple agents and delayed until the full schedule is constructed.

The best performance occurred at iteration 150 with only 106 total conflicts affecting 40.8% of sections. Teacher conflicts reached their minimum of 13 instances (12% of total), while section conflicts, though still substantial at 93, were most controlled. However, this proved unsustainable—by iteration 200, section conflicts jumped to 146, affecting 60% of sections.

Teacher conflict metrics tell a more positive story, remaining stable at 13–30 instances and consistently below 15% of total conflicts after iteration 100. This demonstrates that the MARL system successfully learned to respect teacher availability, workload limits, and expertise requirements through immediate negative rewards for violations.

The absence of monotonic improvement represents a concerning finding. Conflict counts oscillated significantly rather than showing consistent downward trends, suggesting convergence to different local optima with substantially varying quality. This instability likely results from the exploration-exploitation trade-off: as entropy decay encourages deterministic policies, agents occasionally achieve better coordination patterns that prove fragile and are lost as other agents continue updating their policies.

5.4 Multi-Perspective Quality Scores

Table 3 presents severity-weighted validation scores from multiple stakeholder perspectives.

Table 3: **Multi-Perspective Validation Scores**

Iteration	Teacher POV	Student POV	Overall
10	93.5	49.6	67.8
50	95.4	56.8	72.6
100	94.8	60.5	74.9
150	93.1	**66.9**	**77.6**
200	94.5	51.3	68.9
450	93.9	53.7	70.2
800	94.7	58.0	73.2
1000	93.8	52.1	69.3

Teacher Perspective scores remained remarkably consistent at 93.1–95.4 across all iterations, indicating reliable satisfaction of teacher availability, workload distribution, expertise matching, and prevention of double-booking. This strong performance stems from direct enforcement through action masking, explicit representation in agent observation space, and immediate negative rewards for violations. The stability across iterations, even during performance fluctuations, demonstrates that once learned, teacher constraint satisfaction remains robust.

In stark contrast, Student Perspective scores ranged from 49.6 to 66.9, with most iterations between 50–55—equivalent to failing or barely passing grades. The best performance of 66.9 at iteration 150 corresponds to the lowest conflict count, but this could not be sustained, with a 23% decline to 51.3 by iteration 200. Student perspective scores primarily measure section schedule coherence, and the consistently low scores directly reflect the 40–60% section conflict rates. When most sections experience conflicts, high student perspective scores are mathematically impossible.

Overall quality scores ranged from 67.8 to 77.6 (grade D+ to C+), with the best performance at iteration 150 representing the optimal balance achieved between teacher and student constraints. The scores closely track student perspective performance, indicating that student schedule quality is the dominant limiting factor. Even perfect teacher constraint satisfaction cannot compensate for fundamental deficiencies in student schedule coherence.

The correlation between conflict metrics and multi-perspective scores is striking. Iteration 150's lowest conflict count (106) and sections affected (40.8%) directly corresponds to the highest student (66.9) and overall (77.6) scores, confirming that section conflicts drive poor student perspective performance. The persistent 40–45 point gap between teacher and student scores throughout training indicates a fundamental asymmetry in how the system handles different constraint types—not merely a tuning issue, but a reflection of the architectural decision to decompose by subject area rather than student section.

5.5 Algorithm Strengths and Limitations

The MARL approach demonstrates exceptional capability in handling teacher-related constraints, with teacher perspective scores consistently above 93/100. The system successfully manages teacher availability, respects workload limits, matches expertise appropriately, and prevents double-booking. This emerges from direct representation of teacher constraints in agent observations and action masking to prevent invalid assignments. The stability of this performance suggests that teacher constraint handling is a robust capability supporting practical applications.

Rapid initial learning is evident, with placement rates improving from 65.4% to 77.1% in just 40 iterations, suggesting effective learning signals. The system also demonstrates scalability, successfully handling 598 subjects, 257 teachers, and 390 sections—a scale challenging for exact optimization methods. The distributed multi-agent approach allows computational burden distribution across agents operating in parallel. F2F subjects maintained particularly high placement rates (93–99%), indicating that clear spatial constraints provide better learning signals than flexible online modalities.

However, critical limitations prevent production deployment. The most severe is the 40–60% section conflict rate, representing fundamental failure to produce feasible student timetables. This stems from the agent decomposition strategy: organizing agents around subject areas creates a situation where no agent has direct responsibility for any section's schedule. Section schedules emerge implicitly, requiring sophisticated coordination that the current system fails to achieve. The reward signal for section conflicts is diluted across multiple agents, weakening individual learning signals.

Online subjects show dramatic variance (23.5–82.4%), indicating struggle to develop stable policies for temporally flexible subjects. Performance instability throughout training, with metrics oscillating rather than showing monotonic improvement, suggests convergence to different local optima depending on stochastic factors. The plateau after iteration 200 and subsequent oscillations indicate convergence to suboptimal solutions, characteristic of multi-agent environments where coordination requirements exceed agents' effective communication capabilities.

5.6 Comparison with Traditional Methods

Comparing this approach with previous GA-based approaches reveals interesting trade-offs. Our MARL system reaches stable performance in 500–1000 iterations versus 3,000–4,000 generations for GA—a 3–8× improvement in convergence speed. This faster convergence benefits iterative development and institutions requiring frequent schedule regeneration.

However, the solution quality gap is substantial. GA methods achieved zero hard constraint violations, while our MARL system produces 40–60% section conflict rates—rendering current solutions unusable without significant manual correction. GA approaches produce feasible timetables meeting all hard constraints, whereas MARL consistently fails the fundamental requirement of conflict-free student schedules.

Regarding adaptability, GA methods require complete recomputation when constraints change, while MARL theoretically offers continuous learning for constraint adaptation. However, this advantage remains unrealized given current performance instability. The comparison reveals that convergence speed alone is insufficient—a system converging quickly to infeasible solutions provides limited value compared to slow convergence to feasible, high-quality schedules. Future MARL research must prioritize solution quality and feasibility, with convergence speed as secondary. Improved reward engineering, substantially increased section conflict penalties, and shaped rewards for partial schedule coherence are essential for MARL to become competitive with traditional methods.

6 Conclusion and Future work

This work demonstrated that a multi-agent reinforcement learning (MARL) formulation can partially automate university timetabling by decomposing the problem into cooperating Subject Area Head Agents (SAHAs) that learn teacher–subject–timeslot assignments. We implemented PPO with centralized training and decentralized execution (CTDE) and used legality/action masks to guarantee hard-constraint feasibility while the policy learns to optimize soft objectives.

The experimental results on real senior high school data (598 subjects, 257 teachers, 390 sections) reveal both promising capabilities and critical limitations. The system achieved an 82.1% peak subject placement rate and maintained excellent teacher constraint satisfaction (93–95 Teacher POV score). However, the critical limitation lies in section conflicts, affecting 40–60% of student sections with Student POV scores of only 50–55, making the system unsuitable for production deployment in its current form.

These findings suggest that MARL is a practical approach for certain aspects of strand-aware, multi-campus scheduling where teacher constraints dominate. However, fundamental improvements are needed in agent coordination, reward engineering, and exploration strategies to address student schedule coherence before the system can replace or augment manual scheduling processes.

Short-term improvements:

- Enhanced reward engineering with substantially increased section conflict penalties
- Alternative agent decomposition strategies (e.g., section-based or hybrid structures)
- Explicit communication mechanisms between agents for better coordination

- Extended training with adaptive curriculum strategies

Long-term research directions:

- Hybrid architectures combining centralized critics with decentralized actors
- Multi-objective optimization with Pareto front exploration
- Integration of domain knowledge through shaped rewards from expert schedules
- Investigation of different MARL algorithms (e.g., QMIX, MADDPG) for improved coordination

Practical deployment:

Given current limitations, the most viable path forward is a **human-AI collaborative approach**, where the MARL system generates initial teacher assignments (leveraging its 93–95% success rate in this domain), followed by manual adjustment of section schedules to resolve conflicts. This hybrid approach could reduce administrative workload while maintaining schedule quality.

REFERENCES

[1] R. A. Rahman, J. Wahid, A. Abd Wahab, S. Hassan, R. Ahmad, and A. Ahmad, "Systematic literature review of metaheuristic methodologies for high school timetabling problem," in *2022 applied informatics international conference (AIIC)*, IEEE, 2022, pp. 190–195.

[2] H. Alghamdi, T. Alsubait, H. Alhakami, and A. Baz, "A review of optimization algorithms for university timetable scheduling," Engineering, Technology & Applied Science Research, vol. 10, no. 6, pp. 6410–6417, 2020.

[3] I. G. A. Premananda, A. Tjahyanto, and A. Muklason, "Timetabling problems and the effort towards generic algorithms: A comprehensive survey," IEEE Access, 2024.

[4] R. J. De Castro and K. J. T. Kramer, "Genetic algorithm-based timetabling solution for de la salle university–college of computer studies," Bachelor's Thesis, De La Salle University Manila, 2019.

[5] R. Hoshino and I. Fabris, "Optimizing student course preferences in school timetabling," in Integration of constraint programming, artificial intelligence, and operations research (CPAIOR 2020), Springer, 2020, pp. 283–299.

[6] A. Syskov, M. Rabovskaya, and V. Makhneva, "Timetable model for individualization of education based on multi-agent approach," in 2020 ural symposium on biomedical engineering, radioelectronics and information technology (USBEREIT), IEEE, 2020, pp. 467–470.

[7] H. Babaei and A. Hadidi, "A review of distributed multi-agent systems approach to solve university course timetabling problem," ACSIJ Advances in Computer Science: an International Journal, vol. 3, no. 5, pp. 19–28, 2014.

[8] A. D. Guia and M. A. Ballera, "Multi-agent class timetabling for higher educational institutions using prometheus platform," Indonesian Journal of Electrical Engineering and Computer Science, vol. 22, no. 3, pp. 1679–1687, 2021.

[9] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms." 2017. Available: https://arxiv.org/abs/1707.06347

[10] L. Canese et al., "Multi-agent reinforcement learning: A review of challenges and applications," Applied Sciences, vol. 11, no. 11, p. 4948, 2021.

[11] E. Liang et al., "RLlib: Abstractions for distributed reinforcement learning." 2018. Available: https://arxiv.org/abs/1712.09381

[12] R. Project, "RLlib: Industry-grade, scalable reinforcement learning (documentation)." https://docs.ray.io/en/latest/rllib/index.html, 2025.

[13] J. K. Terry et al., "PettingZoo: Gym for multi-agent reinforcement learning." 2021. Available: https://arxiv.org/abs/2009.14471

[14] F. Foundation, "PettingZoo documentation." https://pettingzoo.farama.org/, 2025.

[15] M. L. Ang, R. A. Cabredo, E. R. G. Ocson, and B. J. C. Tan, "ASSYST2: Automated scheduling system 2," Undergraduate thesis, De La Salle University, Manila, Philippines, 2001.

[16] F. Foundation, "PettingZoo tutorials." https://pettingzoo.farama.org/content/tutorials/, 2025.

[17] F. Foundation, "PettingZoo environments (atari, butterfly, classic, MPE, SISL)." https://pettingzoo.farama.org/environments/, 2025.

[18] R. Project, "Multi-agent environments in RLlib." https://docs.ray.io/en/latest/rllib/multi-agent-envs.html, 2025.

[19] R. Project, "RLlib examples." https://docs.ray.io/en/latest/rllib/rllib-examples.html, 2025.

[20] F. Foundation, "RLlib: PPO for pistonball (PettingZoo tutorial)." https://pettingzoo.farama.org/tutorials/rllib/pistonball/, 2025.

[21] C. Li et al., "Reinforcement learning for scalable train timetable rescheduling with graph representation," arXiv preprint, 2024, Available: https://arxiv.org/abs/2401.06952

[22] H. Zhang, Y. Wang, et al., "Multi-agent reinforcement learning algorithms: A survey," Electronics, vol. 14, no. 4, p. 820, 2025, doi: 10.3390/electronics14040820.

Text Mining Analysis of Review Bombing Incidents in Filipino Films

Joshua Nicolai H. Gonzales
Department of Information Technology
De La Salle University
Manila, NCR, Philippines
joshua_nicolai_gonzales@dlsu.edu.ph

Josh Enrico P. Cariaga
Department of Information Technology
De La Salle University
Manila, NCR, Philippines
josh_cariaga@dlsu.edu.ph

Ron Fourier B. Alonzo
Department of Information Technology
De La Salle University
Manila, NCR, Philippines
ron_alonzo@dlsu.edu.ph

Aaron John C. Barcelita
Department of Information Technology
De La Salle University
Manila, NCR, Philippines
aaron_barcelita@dlsu.edu.ph

Nathaniel Kurt D. Reyes
Department of Information Technology
De La Salle University
Manila, NCR, Philippines
nathaniel_kurt_reyes@dlsu.edu.ph

Dr. Raphael W. Gonda
Department of Information Technology
De La Salle University
Manila, NCR, Philippines
raphael.gonda@dlsu.edu.ph

ABSTRACT

Online reviews are critical for consumer evaluation but are increasingly compromised by "review bombing", where anomalous spikes in activity are triggered by external controversy rather than product quality. This study proposes a machine learning framework to detect and characterize such incidents within the highly politicized Filipino film industry. Analyzing 1548 reviews from a highly political film and 3150 reviews from a pop culture romance film, the research employs time series analysis to identify volume anomalies, combined with fine-tuned RoBERTa sentiment analysis and BERTopic modeling to extract thematic structures. The findings distinguish between organic and inorganic engagement: while the pop culture film exhibited a "franchise effect" with film-centric themes driven by sequel anticipation, the political film experienced spikes coincident with the Philippine national elections. Crucially, the RB incidents revealed a near-total displacement of cinematic discourse, with the film's overall review topics pivoting instead toward "Truth in Political Narratives" (73.88%), "Profanity-Driven Political Anger" (13.93%), and "Historical Context" (3.71%). This framework successfully isolates review bombing by identifying this measurable shift from film-centric critique to external sociopolitical grievances.

KEYWORDS

Natural Language Processing, Topic Modeling, Sentiment Analysis, Review Bombing, Online Reviews

1 INTRODUCTION

Online reviews are a staple in the digital economy, significantly influencing consumer purchase decisions [1]. Multiple studies point to reviews being a large contributing factor towards consumer intentions [2]. In light of this recognition of value, customers have turned to a phenomenon called Review Bombing (RB) to express dissatisfaction towards a particular product or service. While RB is present in multiple industries such as gaming, film, and service, scholars increasingly define it not merely by negativity, but by its time series concentration: anomalous spikes in activity triggered by controversy rather than product quality [3].

RB is defined as a form of online sabotage characterized by the coordinated mass submission of reviews on aggregator platforms within a short time frame, producing abrupt spikes that deviate from expected time series patterns [4].

Schuff et al. [5] emphasize that RB campaigns are designed to manipulate aggregate scores by foregrounding cultural, ideological, or political grievances instead of product performance. Although RB is not inherently limited to negative reviews, it is most often operationalized through extreme negative ratings due to their immediate impact on aggregate scores.

While the intent behind RB is often consistent, the nature of the reviews that make it up varies significantly. In many cases, the bombing is explicit and quantifiable, where the

numerical rating aligns with the textual sentiment. For instance, during Ghostbusters (2016) on IMDB, users paired “1/10” ratings with overtly hostile text, citing “sexism and racism” to justify their dissatisfaction. Similarly, the 2024 Helldivers 2 controversy on Steam saw thousands of “Not Recommended” ratings due to grievances against its publisher.

This paper examines the presence of RB in the Philippine context. The Filipino film industry is a prime subject due to its highly politicized nature; as a vessel for social commentary, Filipino films often become battlegrounds for ideological conflict [6]. Two movies were chosen for this paper that were thought to have instances of RB. The first one being a highly political film, which was chosen as the primary subject for this paper given the controversy surrounding its portrayal of a political family in the Philippines and their member’s subsequent electoral win two years later [7]. The second film is a pop culture film, chosen for its controversial actor pairing [8]. Driven by these selected movies, this research addresses two primary questions:

RQ1: Is there a verifiable pattern of review bombing pertaining to Filipino media on English movie review aggregator sites?

RQ2: Can we create a textual analysis framework for detecting and discovering instances of review bombing?

2 METHODOLOGY

This portion of the paper outlines the methods used to analyze and produce the results for evaluating an RB campaign on a film.

2.1 Conceptual Framework

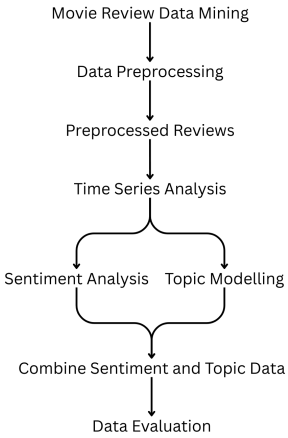


Figure 1: **Conceptual Framework**

The figure outlines the process of gathering text from films that are under the assumption of review bombing. Multiple methods of discovering RB films were utilized. Some of the methods the researchers used include electronic Word-Of-Mouth (eWOM), aggregator site filtering by lowest rated, and news articles.

2.2 Data Mining and Preprocessing

Data for analyzing both films were gathered from three different aggregate websites: Letterboxd, Rotten Tomatoes, and IMDB through data mining scripts. Additionally, the political film and the pop culture film shall be referred to as Movie 1 and Movie 2, respectively.

2.3 Time Series Analysis

Time series analysis was conducted using the preprocessed data to discover time series anomalies that could further identify RB. Reviews are grouped through pandas period conversion, counting the frequency of reviews per month across the entire time span of each film’s review history. The visualization is generated through the matplotlib and pandas Python libraries. If spikes in activity are detected, news articles related to the film are gathered and evaluated to explain the spike in activity.

2.4 Topic Modelling

To analyze themes in user reviews and identify patterns potentially associated with RB, this study employed BERTopic, a transformer-based topic modeling framework that combines contextual language embeddings with clustering and class-based Term Frequency-Inverse Document Frequency (TF-IDF) weighting [9]. BERTopic was selected for its ability to capture semantic relationships beyond word frequency, thereby enabling the extraction of coherent topics from large, unstructured text datasets. This capability is particularly important for online reviews, where language is informal, contextual, and often sentiment-driven [10].

2.4.1 Document Preprocessing. Building upon the initial text preprocessing procedures, such as date standardization, deduplication, special character elimination, and sentence segmentation, further procedures were implemented to guarantee compatibility with the BERTopic pipeline. This includes a combined stopword list being constructed using scikit-learn’s standard English stopwords and a manually curated set of Filipino stopwords (e.g., ng, ang, mga, etc.) that includes common function words and conversational fillers. These stopwords were supplied to the CountVectorizer to prevent them from influencing topic representation.

2.4.2 Embedding Model. To represent semantic meaning in multilingual review text, the model employed the "paraphrase-multilingual-mpnet-base-v2" SentenceTransformer. This transformer architecture is recognized for its robust semantic alignment capabilities, excelling in capturing contextual relationships between words and sentences, particularly across different languages [11].

2.4.3 Vectorization. In conjunction with the Embedding model, a CountVectorizer was used to extract term-level features from the dataset. The vectorizer generated unigram (e.g., "{president's name}") and bigram (e.g., "alden galing") representations while filtering out noise through a combined set of English and Filipino stopwords.

2.4.4 Clustering and Topic Generation. BERTopic generated topics through a two-stage process. First, sentence embeddings generated by the transformer model were clustered using HDBSCAN, BERTopic's default density-based clustering algorithm. HDBSCAN automatically identifies natural topic groupings (Topic 0 - 3…) while assigning documents that do not fit into any cluster to an outlier category (Topic -1). Following clustering, BERTopic employed class-based TF-IDF (c-TF-IDF) to extract significant words for each cluster, enabling a clear interpretation of emerging themes [9]. Furthermore, a minimum topic size of 20 was applied to guarantee a stable and cohesive topic formation given the dataset's size.

2.4.5 Topic Modelling. Topic analysis was conducted using an LLM-assisted approach, following the guidelines proposed by Grootendorst [9] and consistent with the methodology adopted by Yi et al. [12]. For each BERTopic cluster, the top keywords generated through class-based TF-IDF were extracted and provided to GPT-5.2 to generate human-interpretable themes [13]. The LLM was used exclusively for topic labeling, while topic generation and clustering remained fully determined by BERTopic. This approach reduces researcher bias while simultaneously enhancing interpretability. In order to establish a baseline for typical review-space behavior, topic analysis was first conducted on a film exhibiting conventional audience discourse that is characterized by film-centric themes such as narrative, emotional impact, and performance evaluation. These results serve as a comparative reference for identifying deviations indicative of RB.

2.5 Sentiment Analysis

Sentiment analysis was conducted using a transformer-based model tailored to Filipino online discourse to capture affective meaning in English, Filipino, and code-switched (Taglish) reviews. The study employed the raphgonda/FilipinoShopping RoBERTa-base model, fine-tuned on over 400,000 Philippine e-commerce reviews, selected for its robustness in informal and multilingual user-generated text. Reviews were analyzed at the document level, tokenized with a maximum sequence length of 512 tokens, and processed in batches for efficiency [14].

To further examine suspected RB behavior in Movie 2, the review dataset was segmented by sentiment polarity, isolating positive and negative reviews, and BERTopic was then reapplied separately to each sentiment subset. This sentiment-segmented topic modeling approach, similar to that employed by Zhang [15] in the analysis of movie reviews, enabled a granular comparison of thematic structures. Specifically, it allowed the researchers to assess whether negative reviews diverged from film-centric discourse and instead clustered around external political or ideological narratives.

3 RESULTS

3.1 Time Series Analysis

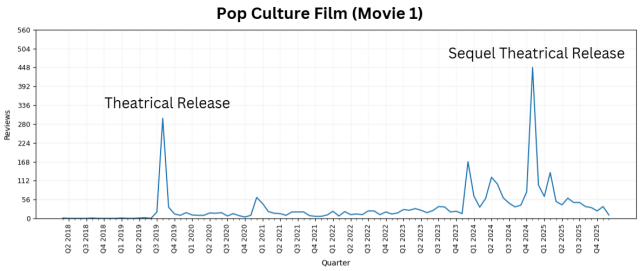


Figure 2: **Time Series Frequency Analysis Distribution for Movie 1**

The time series analysis of Movie 1 reveals a significant resurgence in review activity with a pronounced peak in November 2024. This surge coincides directly with the release of its sequel [16]. This illustrates the franchise effect in audience engagement, where anticipation for new installments revitalizes discussion about preceding films. Furthermore, this pattern suggests that audiences re-engaged with the original narrative, potentially reevaluating its themes, performances, and cultural impact in preparation for or response to the sequel's release.

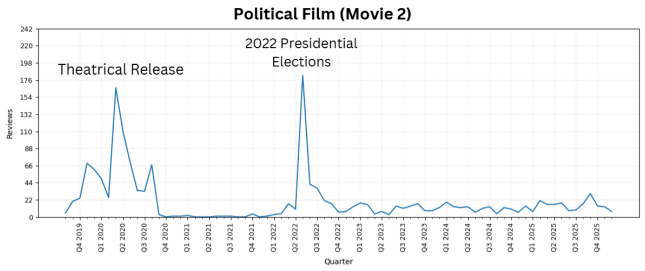


Figure 3: **Time Series Distribution for Movie 2**

The time series analysis of Movie 2 reveals a politically charged pattern of audience engagement, with its peak review volume occurring in May 2022. This surge directly coincides with the 2022 Philippine presidential elections, as reported by Rappler [17], highlighting how documentary films about political families can experience renewed relevance during electoral events.

3.2 Topic Modelling

The dominant themes present in a review dataset are relevant to detecting and understanding RB. In a typical film review corpus, one would expect dominant themes related to filmmaking, elements such as cinematography, editing, or narrative. In contrast, a corpus undergoing an RB campaign often displays a sharp thematic shift. Instead of focusing on the technical or artistic merits of the film, the dominant themes frequently pivot toward external controversies.

Table 1: **Overall LLM-Assisted Themes, BERTopic Keywords, and Review Excerpts for Movie 1**

Theme	Keywords
Positive Film Reception	film, watched, love, sequel, watch maganda, ganda
Narrative and Cultural Appeal	film, love, movies, story, hong kong
Character Chemistry and Performance	galing, character, cute, chemistry

Table 1 outlines the thematic structure derived from the review dataset for Movie 1. The analysis reveals dominant themes that reflect the natural, film-centric audience engagement. These include appreciation for the film's general enjoyment and emotional impact, discussion of its romantic storytelling, and praise for the lead actors' performances. Such themes are consistent with expected audience responses to a mainstream romantic drama, as they engage directly with elements of the film. Furthermore, the dominant themes provide insight into the audience's reception, effectively accomplishing the core function of a review dataset: to synthesize viewer response. Moreover, the thematic clustering demonstrates that the audience's conversation is anchored in the film's content, a sign of authentic audience sentiment.

Table 2: **Overall LLM-Assisted Themes, BERTopic Keywords, and Review Excerpts for Movie 2**

Theme	Keywords
Truth in Political Narratives	{president's name}, {former first lady's name}, documentary, people, film, truth, real truth
Profanity-Driven Political Anger	t*ng*n*, {president's name}, family t*ng*n*, f**k {president's name}, rich
Memory, Truth, and Perception	Truth, perception, neveragain, neverforget, forget

In contrast, the topics generated from the reviews for Movie 2 (Table 2) reveal an absence of film-centric discourse and a near-total takeover of the review space by the three politically charged narrative clusters. Although the theme "Truth in Political Narratives" is related to the documentary's content, it is accompanied by the other two themes, "Profanity-Driven Political Anger" and "Memory, Truth, and Perception", which reflect political sentiment. These two themes demonstrate that the review platform has been repurposed. It functions less as a space for film criticism and more as a direct extension of the socio-political conflict the film documents.

3.3 Sentiment Analysis and Topic Modelling

To further analyze the suspected RB in Movie 2, the researchers segmented the dataset according to sentiment, isolating negative and positive ratings, and reapplied the BERTopic analysis.

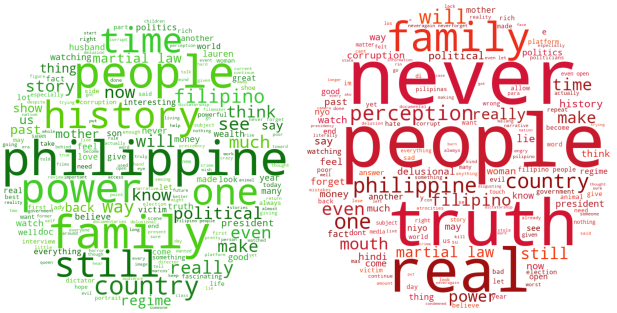


Figure 4: **Word Clouds for Positive and Negative Reviews for Movie 2**

The contrast between positive and negative reviews is reflected in the word clouds (Figure 4). Positive reviews highlight terms associated with educational value and historical significance, indicating that viewers perceive the film as a meaningful contribution to public understanding. In contrast, negative reviews are dominated by language expressing bias and strong emotional reactions, reinforcing the presence of polarized responses.

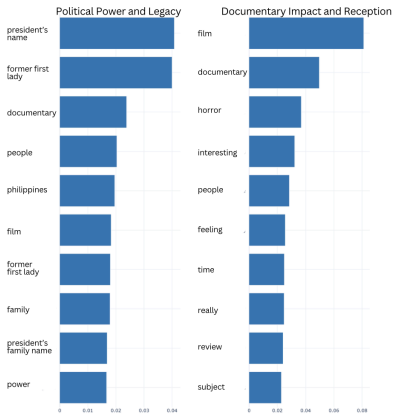


Figure 5: **LLM-Assisted Themes & BERTopic Keywords for Positive Sentiment Reviews for Movie 2**

As we can see in Figure 4, 78.96% of positive sentiment reviews fall under the theme of “Political Power and Legacy”. This dominant cluster focuses on the film’s role in documenting the rise and influence of the subjects. This is followed by 8.59% for “Documentary Impact and Reception”, where viewers express reactions to the film. The remaining 12.44% were composed of outliers that were not associated with a topic.

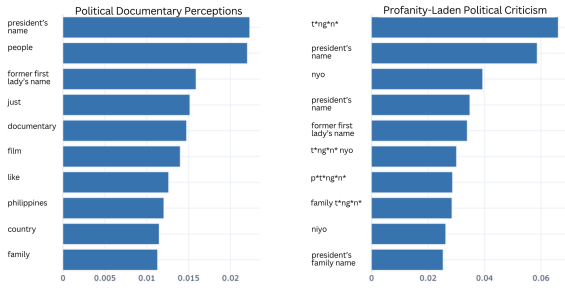


Figure 6: **LLM-Assisted Themes & BERTopic Keywords for Negative Sentiment Reviews for Movie 2 (Political Documentary Perceptions & Profanity-Laden Political Criticism)**

The distribution of themes within the negative review sentiment subset reveals that the discourse is almost entirely concentrated on historical and socio-political grievances. The data shows that 60.77% of negative sentiment reviews are focused on the theme of “Political Documentary Perceptions”, specifically addressing the film’s depiction of the political subjects and the accuracy of the events shown.

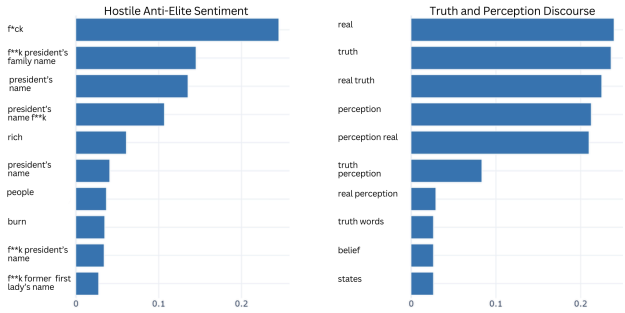


Figure 7: **LLM-Assisted Themes & BERTopic Keywords for Negative Sentiment Reviews for Movie 2 (Hostile Anti-Elite Sentiment & Truth and Perception Discourse)**

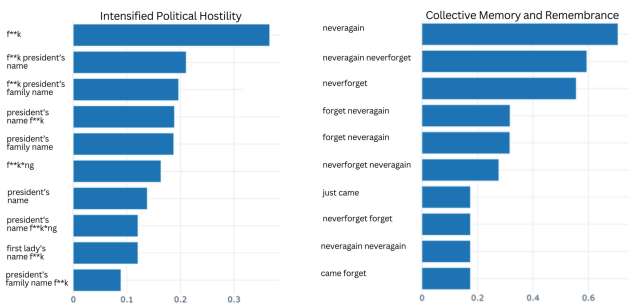


Figure 8: **LLM-Assisted Themes & BERTopic Keywords for Negative Sentiment Reviews for Movie 2 (Intensified Political Hostility & Collective Memory and Remembrance)**

This is followed by “Profanity-Laden Political Criticism” at 19.44%, which represents expressions of frustration and systemic disapproval. Smaller but distinct clusters include “Hostile Anti-Elite Sentiment” (5.24%), which critiques elitism, “Truth and Perception Discourse” (4.42%), which is a dialogue revolving around the film’s subjective interpretation of facts, and “Intensified Political Hostility” (4.42%), which contains aggressive, expletive-heavy dialogue. Finally, “Collective Memory and Remembrance”, characterized by the use of prominent protest slogans such as “never again” and “never forget,” accounts for 1.40% of the reviews, while the remaining 4.31% consists of outliers not associated with a specific topic.

4 CONCLUSION

This study demonstrates that combining time series analysis, fine-tuned sentiment analysis, and BERTopic modeling provides a robust framework for detecting review bombing in the Filipino context. By analyzing a pop culture film and a political film, the research successfully distinguished between organic and inorganic engagement spikes. While organic interest, such as the "franchise effect" observed in the pop culture film, maintains a focus on narrative and performance, review bombing campaigns are characterized by a semantic pivot. In the case of the political film, the discourse was displaced, shifting from cinematic critique to external political keywords (e.g., "Political Documentary Perceptions," "Profanity-Laden Political Criticism") triggered by the 2022 national elections.

The framework’s ability to isolate these thematic shifts validates that RB is defined not just by volume or negativity, but by the weaponization of context, where the review platform is repurposed for ideological conflict.

Iterations of this research will focus on three key areas. First, the dataset will be expanded to include a broader range of films and have a more robust comparative analysis across different genres, themes, and sociopolitical contexts. Moreover, this would also improve the generalizability of the findings. Second, the researchers aim to further fine-tune the Filipino-specific stopword lists and RoBERTa embeddings to reduce false sentiment positives in code-switched text. Third, the study proposes the automation of this pipeline. By integrating real-time scraping of news events with the existing topic modeling framework, it may be possible to create a dynamic monitoring system that automatically flags anomalous thematic divergences as they occur, providing platform moderators with immediate context for potential moderation.

REFERENCES

[1] R. Thakur (2018). Customer engagement and online reviews. Journal of Retailing and Consumer Services, 41(41), 48–59. https://doi.org/10.1016/j.jretconser.2017.11.002
[2] T. Chen, P. Samaranayake, X. Cen, M. Qi, and Y.-C. Lan (2022). The Impact of Online Reviews on Consumers’ Purchasing Decisions: Evidence from an Eye-Tracking Study. Frontiers in Psychology, 13(13), 1–13. https://doi.org/10.3389/fpsyg.2022.865702
[3] J. Guszkowski (2025). Restaurants sound the alarm over “review bombing.” Restaurant Business. https://www.restaurantbusinessonline.com/technology/restaurants-sound-alarm-over-review-bombing
[4] G. G. Cantone, V. Tomaselli, and V. Mazzeo, “Review bombing: ideology-driven polarisation in online ratings: The case study of The Last of Us (part II),” Quality & Quantity, Sep. 2024, doi: https://doi.org/10.1007/s11135-024-01981-z.
[5] D. Schuff, S. Mudambi, and M.-X. Wang (2024). Understanding the Review Bombing Phenomenon in Movies and Television. Proceedings of the Annual Hawaii International Conference on System Sciences. https://doi.org/10.24251/hicss.2024.036
[6] D. C. Rudin (2021). Counter/Public: The Politics of Committed Film in the Philippines. Escholarship.org. https://escholarship.org/uc/item/37k2q168
[7] F. M. Cervantes (2022). Marcos, Duterte proclaimed winners in PH’s fastest vote count. Www.pna.gov.ph. https://www.pna.gov.ph/articles/1175224
[8] J. Santiago (2019). Basher Hits Kathryn Bernardo’s Mom Over “Hello, Love, Goodbye” Movie. PhilNews; Philippine News. https://philnews.ph/2019/07/08/basher-hits-kathryn-bernardos-mom-over-hello-love-goodbye-movie/
[9] M. Grootendorst (2022). BERTopic: Neural topic modeling with a class-based TF-IDF procedure. ArXiv. https://doi.org/10.48550/arxiv.2203.05794
[10] R. Y. Kim (2021). Using Online Reviews for Customer Sentiment Analysis. IEEE Engineering Management Review, 49(4), 1–1. https://doi.org/10.1109/emr.2021.3103835
[11] Colangelo, M. T., Meleti, M., Guizzardi, S., Calciolari, E., & Galli, C. (2025). A Comparative Analysis of Sentence Transformer Models for Automated Journal Recommendation Using PubMed Metadata. Big Data and Cognitive Computing, 9(3), 67. https://doi.org/10.3390/bdcc9030067
[12] J. Yi, Y. K. Oh, and J.-M. Kim (2025). Unveiling the drivers of satisfaction in mobile trading: Contextual mining of retail investor experience through BERTopic and generative AI. Journal of Retailing and Consumer Services, 82, 104066–104066. https://doi.org/10.1016/j.jretconser.2024.104066
[13] OpenAI, *ChatGPT* (GPT-5.2 version) [Large Language Model], 2026. Available: https://chat.openai.com
[14] R. Gonda (2024). FilipinoShopping (revision 714aa95). https://doi.org/10.57967/hf/2903
[15] B. Zhang (2023). A BERT-CNN Based Approach on Movie Review Sentiment Analysis. SHS Web of Conferences, 163, 04007. https://doi.org/10.1051/shsconf/202316304007
[16] K. A. Llemit, “LIST: ‘Hello, Love, Again’ to hold midnight screenings, gets worldwide release,” Philstar.com, Oct. 18, 2024. https://www.philstar.com/entertainment/movies/2024/10/18/2393489/list-hello-love-again-hold-midnight-screenings-gets-worldwide-release (accessed Dec. 23, 2025).
[17] L. Buan (2022). 36 years after ousting Marcos, Filipinos elect son as president. RAPPLER. https://www.rappler.com/philippines/elections/ferdinand-bongbong-marcos-jr-wins-president-philippines-may-2022/

Judo Throw Recognition Using Computer Vision Techniques

Angelo Gabriel D. Arenas
Institute of Computer Science
University of the Philippines
Los Baños, Laguna, Philippines
adarenas@up.edu.ph

Val Randolf M. Madrid
Institute of Computer Science
University of the Philippines
Los Baños, Laguna, Philippines
vmmadrid@up.edu.ph

Jesus K. Araneta
UPLB Judo Coach
University of the Philippines
Los Baños, Laguna, Philippines
jj.araneta@gmail.com

ABSTRACT

This paper explores the use of computer vision, particularly pose estimation and LSTM for action recognition, in automatically recognizing throws for the sport of judo. A custom data set was created that contains 135 instances of various judo throws at different angles such as o-goshi, osoto-gari and ouchi-gari along with 348 non-throwing situations. Yolov8-pose was used for pose estimation to obtain skeleton data, two combatants in every frame resulted in 68 keypoints of data per frame. Throw recognition was implemented via an LSTM that takes an input of a video with a length of 50 frames. Throw recognition via LSTM achieved a 93.83% accuracy and an F1 score of 93.33% indicating good performance with few errors since the model is capable of recognizing throwing techniques not included in the training data. However, the study leaves much to be desired since only a limited amount of techniques and data were included in the study. For future studies, exploring action recognition in scoring is recommended since throw recognition alone is not enough to detect scores.

KEYWORDS

Judo, Action Recognition, Pose Estimation, Computer Vision

1 INTRODUCTION

1.1 Background of the Problem

Computers and sports are becoming increasingly intertwined. *Computer vision* (CV) is an interdisciplinary field of study that includes digital image processing, machine learning and other disciplines [11]. The field of computer vision focuses on improving the ability of computers in interpreting images and videos [16]. Computer vision has numerous applications in sports such as 3d modeling of players, helping a referee officiate by providing additional context, and identifying important moments in a game for gathering sports statistics [12].

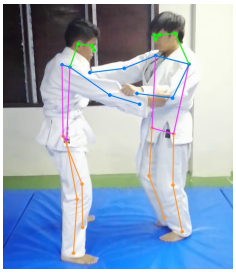
Pose Estimation is a CV task with the goal of locating human anatomical keypoints [13]. Hudovernik and Skocaj define a human pose as a "constellation of human joints" [5]. Posture is important in the sport of judo in both attacking and defending. Deep learning is used in all state of the art approaches to pose estimation, and the two main approaches are top-down which first detects all persons and their respective bounding boxes before identifying the joints in contrast to the bottom-up approach that detects all the joints first before linking them into a pose [5].

Human activity can be described as a predetermined series of physical movements done over a period of time to accomplish a task while *Human Activity Recognition* (HAR) is a complex task that involves the detection of single or multiple persons in a video, the location of the action with respect to time and space, and lastly a classification for the performed action [4]. For judo in particular, a throw is composed of multiple physical movements. *Judo* is a Japanese martial art and sport with millions of practitioners around the world where athletes can score using *nage-waza* (throwing techniques) and *katame-waza* (ground techniques) to win a match [14]. Athletes who practice judo are called *judoka*. An *ippon* (full score) can be awarded if a throw technique meets the four criteria of speed, force, control and results in the opponent landing on his back while a *waza-ari* (half-score) will result if a throw doesn't meet the four criteria of an ippon [1]. Scoring can also occur during ground combat called *ne-waza*. Scoring an ippon or two waza-ari will end the match and in case no judoka scores either situation, the judoka with the higher waza-ari count will win the match [3].

(a) Judoka fighting

(b) Example of ippon

(c) Osoto-gari, a judo throw



(d) Pose estimation

Figure 1: Different images showing the sport of judo

1.2 Objectives

This study aims to create an automated judo throw recognition system for judo footage. The study is aiming for these specific objectives:

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
PCSC2026, April 2026, Davao, Philippines
© 2026 Copyright held by the owner/author(s).

(1) Record footage of judokas in combat situations
(2) Apply pose estimation on judokas in order to gain usable skeleton data
(3) Train a neural network on skeleton data to recognize throws
(4) Evaluate the correctness of the system in detecting judo throws

1.3 Significance

Successful action recognition on judokas can help in creating statistical data for the sport of judo. This can help organizers and researchers who would like to view data such as the actions of a judoka leading up to a score. In the institution where this study will be conducted, no such automatic judo throw recognition system exists.

1.4 Scope and Limitation

A judoka can win a match using any of 68 throwing or 32 grappling techniques [14]. However, not all scoring techniques will be recognized due to the large amount of possible scoring moves, only select nage-waza (throwing techniques) will be used for the data. Ne-waza (ground combat) techniques such as pins and chokes will not be included due to time constraints. Furthermore, the data or throwing footage recorded will be in the context of nage-komi or throwing practice only. While it is more appropriate to apply throw recognition in a *randori* (free practice) and *shi-ai* (competition) context since the participants can move more naturally without restrictions, nage-komi was chosen due to the difficulty in obtaining a large amount of data with only two people being recorded at a time.

2 REVIEW OF RELATED LITERATURE

A study done to provide an overview of HAR in sports found researchers are increasingly using deep learning methods such as convolutional neural networks (CNN) which are sometimes combined with attention mechanisms and transformers along with LSTMs [4]. *Long Short-Term Memory* (LSTM) neural networks are an improved type of recurrent neural network (RNN) that handle long-term dynamics better and are capable of obtaining the spatial relationships between joints in HAR tasks [9]. Yan et al. proposed an improved CNN approach to pose estimation called *Spatial-Temporal Graph Convolutional Network* (ST-GCN) which considers the spatial and temporal patterns of joints derived from skeleton data sequences [15]. Action recognition tasks based on skeleton data are preferred by researchers compared to RGB data, because it is lightweight and efficient coupled with the improving accuracy of convolutional neural networks in detecting human keypoints based on the skeleton [9]. A study on Judo was done on predicting successful judo throws based on the posture beforehand which was based on four main types of factors which are arm position, body contact, body position, and head height for both athletes before applying logistic regression [8]. A study done on the efficiency of judo throws from different gripping styles was conducted with the help of two experts who would review footage to record attempted throws and the grip beforehand [7].

Brazilian Jujitsu (BJJ) is a martial art related to Judo that focuses more on ground combat and submissions. Hudovernik and Skocaj developed a program that could classify combat positions and perform simplified scoring for BJJ, and they noted that combat position detection is significantly worse when only one person's pose is detected, heavy occlusion is a challenge along with intertwining of limbs, and takedowns are difficult to detect if only a single frame is given [5].

Taekwondo is a martial art that focuses more on striking, especially kicks. An action recognition study done on Taekwondo resulted in a program that could classify different kicks and generate scoring, and the researchers found that ST-GCN was noticeably better for action recognition than LSTM [9]. Abin et al. created a system for action recognition on karate poses by developing a K-Nearest Neighbor (KNN) classifier that takes input data obtained from pose estimation in the form of angles between specified joints [2].

3 METHODOLOGY

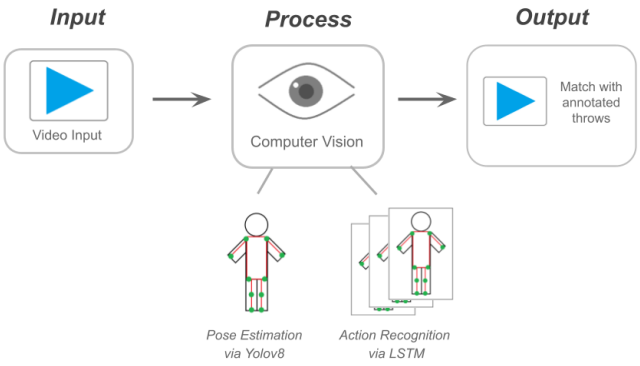


Figure 2: Overall flow or pipeline of the program

3.1 Data Gathering

For recording, an off-the-shelf mirrorless camera was chosen for convenience. The camera was stabilized on a tripod and adjusted until it was level to the ground. The video footage was in mp4 format at 25 frames per second at a resolution of 1280 x 720. For the camera settings; the shutter speed ranges from at least 1/60s up to 1/200s depending on the lighting conditions, the f-stop was set as low as possible to allow more light, and the ISO was either on auto ISO or maxed out to 12800. The lighting at the time of recording was based on the time of day along with the ceiling lights in the venue. Footage was recorded with the written consent of the head of a local judo club. Participating judoka were told that only two judokas should be visible in front of the camera to make the collection of skeleton data easier.

During recording, participants were instructed that the *uke* or the person receiving the technique would face towards the cardinal and intermediate directions (North, South, West, East, Northwest, Northeast, Southwest, Southeast) after being thrown, so that there would be footage of a single technique from different angles. Additionally, participating judoka were instructed to assume a right handed stance with their right foot and hand forward. Different

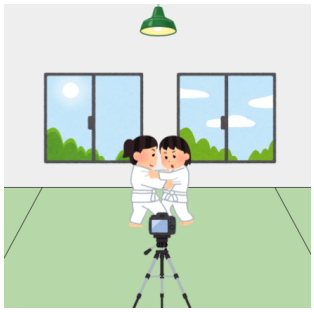


Figure 3: Recording setup for the data

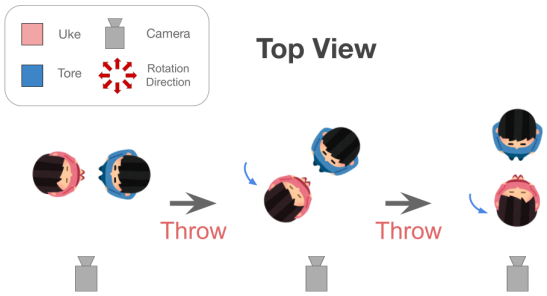


Figure 4: *Uke* changes position after every execution of a technique by 45 degrees

techniques were used in order to make the score and throw prediction more robust since techniques can look different visually and result in varied landings for the uke. To create the custom dataset, each instance of a throw from the recorded footage was saved as video file in the throw class, while non-throwing situations such as standing up, gripping, sitting and walking are also saved and utilized for the non-throwing class.

3.2 Pose Estimation

To obtain the skeleton data needed for training the models, yolov8 was used for both object detection and pose estimation. Yolov8 was chosen for ease of use and installation while also being lightweight. The 17 keypoints of the body in coco format each have a pair of x and y values resulting in an array with 34 features for a single pose. The x and y values were normalized to allow input videos of different resolutions. Zero padding was implemented for missing limbs. The model yolov8x-pose was used for increased accuracy in pose estimation.

3.3 Action Recognition via LSTM

Action Recognition was implemented using an LSTM due to its simplicity. The LSTM takes a fixed input array of 50 frames x 68 keypoints where 50 is the longest duration of a throwing video used in training and 68 is the sum obtained by combining two individual pose arrays. Videos of non-throwing situations were divided every 50 frames to generate clips that don't exceed the fixed input of the LSTM. Extra people (>2) detected were masked manually by adding a large colored rectangle. The LSTM uses a sliding window approach for prediction and zero padding was applied. Masking was implemented to ignore padded values. The final layer is a dense layer that outputs class probabilities of throw and not-throw from a softmax activation function. Loss during training was minimized through an Adam optimizer. The difference between predicted and actual class probabilities was measured through a categorical cross-entropy loss function. An early stopping callback was set at 5 epochs. A ReduceL-ROnPlateau callback was implemented to halve learning rate when the model stops improving. Hyperparameter tuning was done manually by experimenting with different values for parameters. The LSTM uses an arbitrarily set threshold of 0.60 to indicate whether it belongs to the throw or non-throw class. When a throw is detected, an arbitrary counter of 25 frames or a second is set to ensure a throw is not repeatedly counted since the system uses a sliding window approach which can be seen in Fig. 5. When no throw has occurred in the last 25 frames, only then is the initial throw counted. A throwing situation is active as long as the arbitrary counter is greater than or equal to zero. Non-throwing situations are active when the arbitrary counter is less than zero meaning no throw has occurred in the past 25 frames.

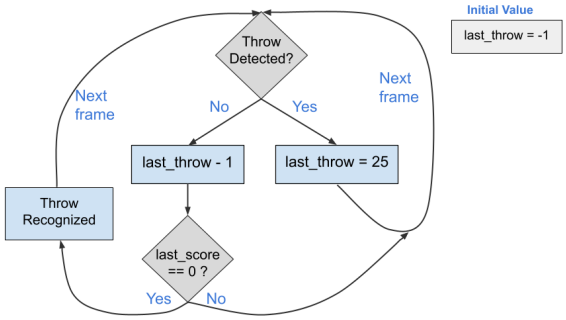


Figure 5: Throw recognition diagram for the LSTM

3.4 Libraries and Technologies

(1) Yolov8 for Pose Estimation
(2) Pytorch LSTM with Keras API for action recognition
(3) OpenCV for image and video processing
(4) Python, PyQt for GUI

The program was written in Python. A basic GUI was developed with the PyQt library. OpenCV was used for the image and video processing loop. A PyTorch LSTM with the Keras API was used for the prediction of throws. An Acer Predator Helios 300 PH315-52 with an i7-9750H processor was used to develop the system. A mirrorless Canon M50 mk2 camera with the EF-M 15-45mm f/3.5-6.3 IS STM lens was used to record data.

4 RESULTS AND DISCUSSION

4.1 Data Gathering

The main camera settings adjusted were the ISO, aperture and shutter speed. The footage recorded during the night had lower shutter speeds such as 1/60s at minimum, but the videos recorded

with some daylight could reach up to 1/180s with the ISO maxed out. Due to the speed of a throw, the person detection and pose estimation would rarely work when the uke was mid-air resulting in only one pose being detected due to blurry images, especially for the footage shot at lower shutter speeds that were recorded at night. For a dynamic context such as judo, bright lighting is the most important factor since it would allow for fast shutter speed for high-speed throws. Multiple recording sessions were needed to create the dataset. Recording throws that resulted in no-score and ippon landings was difficult or painful for some of the participants. Judo is a very televised sport, so it would be ideal to get data from online sources such as international competition footage due to the difficult and time-consuming nature of recording different throws and landings. However, smaller competitions may not have the same production quality and lighting that bigger competitions do, so the footage recorded in this study might be more representative for smaller competitions.

4.2 Preprocessing

Training was done with stratified k-fold cross-validation where data was split into 5 folds. The LSTM training data used 135 throwing and 348 non-throwing clips. The 2d arrays of size 50 x 68 obtained per video clip were saved into pkl files before being converted into numpy arrays for training the LSTM.

4.3 Dataset

Technique	Freq.
Ippon-seio-nage	22
O-goshi	31
De-ashi-barai	18
Osoto-gari	13
Osoto-otoshi	20
Kouchi-gari	16
Ouchi-gari	15
Total	**135**

Table 1: Training data for LSTM

Technique	Freq.
Mat Return	35
Morote-seio-nage	15
O-goshi	6
Ippon-seio-nage	11
Tai-otoshi	6
Hane-goshi	1
Total	**74**

Table 2: Test data for LSTM

The training data set for the action recognition model consists of 135 instances of techniques which are the following: o-goshi (31), ippon-seio-nage (22), osoto-otoshi (20), de-ashi-barai (18), kouchi-gari (16), ouchi-gari (15), and osoto-gari (13). Most of the techniques in the dataset are basic throws taught to the participating judoka with the exception of Osoto-otoshi which was done because it was easy to produce a flat back landing or an ippon with the technique. Ippon-seio-nage and o-goshi are classified as koshi-waza or hip technique while the rest are classified as ashi-waza or leg technique. The model was trained on a variety of techniques to ensure that it will recognize different throws. All techniques in the training data were done standing because sacrifice techniques where you drop on your knees or butt are not taught to beginners in the participating judo club. The final test data set for the LSTM is composed of 74 throws. Two instances of a kneeling morote-seio-nage were recorded, but the rest were done on the feet. The no-score landings after a throw total at 40 which was obtained through mat return (35), ippon-seio-nage (3), morote-seio-nage (1) and hane-goshi (1). The score and non-score landings after a throw in the test data were verified with the help of a judo coach of a university team with 19 years of experience. A mat return technique was used in order to force a belly-down landing for the uke for safety reasons. The mat return technique is uncommon in judo and is seen more in BJJ and wrestling, but it was chosen due to its safety and ease of execution, while still resulting in a belly-down landing. Only 5 instances of no-score landings from regular judo throws were recorded since some of the participants had difficulty in landing safely.

(a) Kumi-kata (b) Kuzushi (c) Tsukuri

(d) Kake (e) Ukemi

Figure 6: Five phases of a successful throw: (a) kumi-kata (b) kuzushi (c) tsukuri (d) kake (e) ukemi

A successful judo throw can be divided into five phases seen in Fig. 6. First is kumi-kata which is grip fighting in judo, a grip must be established first before a throw can be accomplished. The founder of Judo, Kano, divided a throw into three phases, first is kuzushi which is balance breaking, then tsukuri which means fitting in and finally kake, the execution of the throw itself [6]. Ukemi is the act of break-falling or falling safely which is the final phase where uke falls to the ground. Each instance of a technique was separated into its own video where the start could be as early as the kumi-kata phase but most segments begin at the tsukuri phase

just shortly before the kake occurs. This was done because the tsukuri phase can vary in speed, footwork and doesn't necessarily transition to the kake stage. All throwing video segments end at the ukemi phase. As a result, the longest throwing segment has a duration of 50 frames.

4.4 Pose Estimation and Object Detection

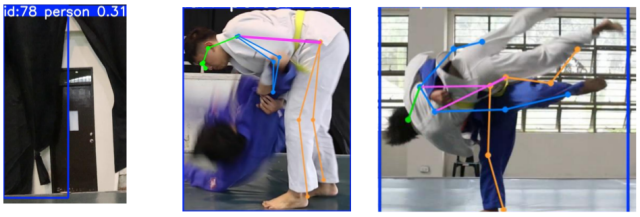


Figure 7: Examples of errors in pose estimation and object detection

Initially, the system should have been able to differentiate between ippon and waza-ari scores, but it was not implemented due to inaccuracy in both pose estimation and object detection, and the difficulty in recording different score landings. In judo, the line between a waza-ari, ippon and no-score landing can be very narrow [1]. Ambiguous poses with missing limbs and unclear orientations were also encountered. For object detection, the most common problem faced is that the uke mid-air after being thrown would not be recognized due to being blurry. The gi could cause inaccurate pose estimation because the gi can cover most of the body except the hands, feet and face. Occlusion and high speed movement is common in the sport of judo that can lead to incorrect person detection and pose estimation. Therefore, more SoTA pose estimation and object detection models are recommended for obtaining accurate skeleton data from judo.

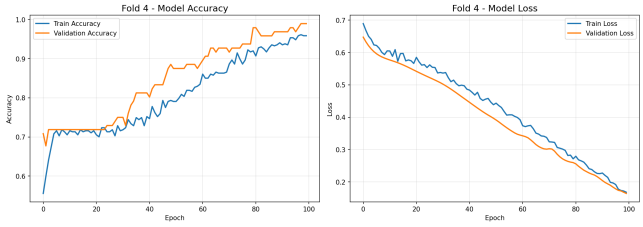
4.5 Training the LSTM Model

Table 3: Stratified 5-fold Cross Validation Results

Fold	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
1	90.72	84.62	81.48	83.02
2	84.54	87.50	51.85	65.12
3	97.94	100.0	92.59	96.15
4	98.96	100.0	96.30	98.11
5	96.88	96.15	92.59	94.34
Mean	**93.81**	**93.65**	**82.96**	**87.35**
Std	**5.45**	**6.42**	**16.33**	**12.29**

For resampling, stratified k-fold was implemented from the sklearn library and done with the parameters of shuffle set to true, random state set to 42 and the number of folds equal to 5. The best fold was then utilized as the final model which is fold 4 seen in Table 4. During training, the mean 93.81% accuracy and 93.65% precision was achieved with minimal variation. However, both recall and F1 score had high variation due to the performance of fold 2 being much worse which could be the result of the small dataset with a variety of techniques.

The hyperparameters used for the LSTM model were 100 epochs, a batch size of 483 obtained from the total number of videos and a learning rate of .0001. Both the training and validation loss steadily decreased until the 100th epoch. Early stopping did not occur. The gap between training and validation for both accuracy and loss is minimal meaning the model is not overfitting and can generalize.



(a) Loss curve for action recognition

4.6 Throw Recognition via LSTM

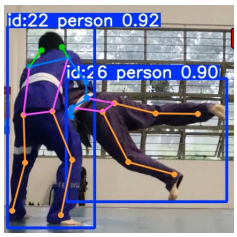
		Predicted	
		Throw	**Not-throw**
Actual	**Throw**	70	4
	Not-throw	6	82

Table 4: Confusion matrix for 'throw' and 'not-throw' classes.

The throw detection system obtained a 93.83% accuracy meaning it is effective despite the presence of techniques not included in training which are mat return (35), morote seio-nage (15), tai-otoshi (6) and hane-goshi (1), meaning the model can generalize even when exposed to new techniques. The model achieved a respectable 94.60% precision, 92.10% recall, and an F1 score at 93.33% which shows that the model performs well and has low error rates.

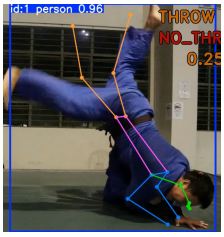
Furthermore, 3 out of 4 false negatives reported were techniques not included in training which are mat return (2) and a kneeling morote seio-nage (1). The kneeling morote seio-nage at Fig. 8(c) might be a false negative because there are few bellydown landings (5) in the training data. One standing morote seio-nage Fig. 8(d) resulted in a false negative possibly because the training data of the same throw was executed at a much deeper squat position. Variations in execution speed, body positioning and camera angle could have resulted in false negatives since most instances of the throws were still recognized. As such, recording techniques at different speeds with different people can make the data more robust.

The false positives in the footage involved the actions of getting up from sitting position (5) and standing (1) which can be seen in Fig. 9. The hypothesized reason is that a bigger negative class could have prevented this since the non-throwing footage was obtained from periods of non-throwing in the recorded footage. Moreover, longer duration videos were simply split into 50 frame segments

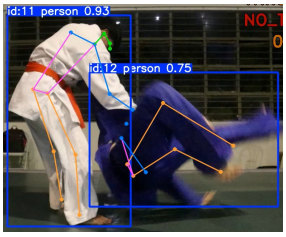


(a) Mat Return

(b) Mat Return

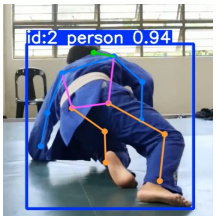


(c) Morote (Kneeling)



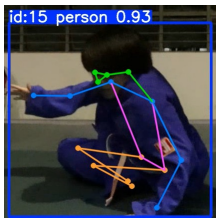
(d) Morote (Standing)

Figure 8: False Negatives in Throw Recognition

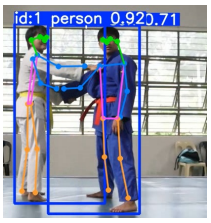


(a) Getting up

(b) Getting up



(c) Sitting



(d) Standing

Figure 9: False Positives in Throw Recognition

due to time constraints instead of being split by individual actions such as standing up, thus the training segments varied in lengths which could have affected the LSTM prediction.

5 CONCLUSIONS AND RECOMMENDATIONS

An automated judo throw recognition system was able to achieve balanced metrics of 93.83% accuracy, 94.60% precision, 92.10% recall and a 93.33% F1 score meaning the model recognizes throws well. Throw recognition is important in determining penalties for non-combativity and whether a score should be given. The LSTM recognized throws even if some of the techniques in the test data were not included in training data meaning the model can generalize. Bright lighting and a good camera with high shutter speed will allow for quality high speed footage, however local and smaller tournaments may not have the same lighting and cameras used in international comps. The amount of footage needed to develop a data set for judo is challenging due to the variations of techniques, landings, and even execution speed of a technique, so the researcher recommends a way to utilize online sources such as international competition footage. However, even with the limited data used in the study with some footage shot under subpar lighting, it was still possible to create an automated throw recognition system.

Judo is a challenging context for computer vision due to high speed movements and occlusion occurring. More state-of-the-art neural networks for person detection and pose estimation are recommended because accurate skeleton data is important for individual pose and action-related analysis. Accurate pose estimation might have allowed for detection of ippon or waza-ari scores since the line between score landings can be very narrow. Spatial recognition on when the uke lands on the mat is highly recommended and critical for score validation because a person can have the same pose mid-air and on the ground.

REFERENCES

[1] 2023. Evaluation of the points in nage-waza. https://rules.ijf.org/page-5
[2] Abin Aju, Christa Mathew, and O. Prakasi. 2022. PoseNet based Model for Estimation of Karate Poses. *Journal of Innovative Image Processing* 4 (05 2022), 16–25. doi:10.36548/jiip.2022.1.002
[3] Xurxo Dopico-Calvo, Eliseo Iglesias-Soler, Luis Santos, Eduardo Carballeira, and Xián Mayo. 2022. Analysis of Successful Behaviors Leading to Groundwork Scoring Skills in Elite Judo Athletes. *International Journal of Environmental Research and Public Health* 19, 6 (2022). doi:10.3390/ijerph19063165
[4] Kristina Host and Marina Ivašić-Kos. 2022. An overview of Human Action Recognition in sports based on Computer Vision. *Heliyon* 8, 6 (2022), e09633. doi:10.1016/j.heliyon.2022.e09633
[5] Valter Hudovernik and Danijel Skocaj. 2022. Video-Based Detection of Combat Positions and Automatic Scoring in Jiu-jitsu. In *Proceedings of the 5th International ACM Workshop on Multimedia Content Analysis in Sports*. Association for Computing Machinery, New York, NY, USA, 55–63. doi:10.1145/3552437.3555707
[6] Rodney Imamura, Hreljac Alan, Rafael Escamilla, and W. Brent Edwards. 2006. A three-dimensional analysis of the center of mass for three different judo throwing techniques. *Journal of Sports Science & Medicine* 5, CSSI (2006), 122–131.
[7] Husnija Kajmovic and Izet Radjo. 2014. A Comparison of Gripping Configuration and Throwing Techniques Efficiency Index in Judo Between Male and Female Judoka During Bosnia and Herzegovina Senior State Championships. *International Journal of Performance Analysis in Sport* 14, 2 (2014), 620–634. doi:10.1080/24748668.2014.11868747
[8] Satoshi Kato and Shinichi Yamagiwa. 2022. Predicting Successful Throwing Technique in Judo from Factors of Kumite Posture Based on a Machine-Learning Approach. *Computation* 10, 10 (2022). doi:10.3390/computation10100175
[9] Jianqiao Liang and Guocai Zuo. 2022. Taekwondo Action Recognition Method Based on Partial Perception Structure Graph Convolution Framework. *Scientific Programming* 2022 (02 2022), 1–10. doi:10.1155/2022/1838468
[10] Takashi Mifune. [n. d.]. Irasutoya. Available: https://www.irasutoya.com/. Accessed: October, 2025.
[11] Yiqun Pang, Qiurui Wang, Changnian Zhang, Mingyang Wang, and Yibing Wang. 2022. Analysis of Computer Vision Applied in Martial Arts. In *2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE)*. 191–196. doi:10.1109/ICCECE54139.2022.9712803
[12] Graham Thomas, Rikke Gade, Thomas B. Moeslund, Peter Carr, and Adrian Hilton. 2017. Computer vision for sports: Current applications and research topics. *Computer Vision and Image Understanding* 159 (2017), 3–18. Issue 3. doi:10.1016/j.cviu.2017.04.0110
[13] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. 2022. ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation. arXiv:2204.12484 [cs.CV]
[14] Takao Yamasaki. 2023. Benefits of Judo Training for Brain Functions Related to Physical and Cognitive Performance in Older Adults. *Encyclopedia 2023* 3 (2023), 981–995. Issue 3. doi:10.3390/encyclopedia30300710
[15] Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. arXiv:1801.07455
[16] Lan Zou. 2023. Chapter 5 - Meta-learning for computer vision. In *Meta-Learning*, Lan Zou (Ed.). Academic Press, 91–208.

FITPOSE: A COMPUTER VISION-BASED SYSTEM FOR REAL-TIME POSTURE ASSESSMENT IN HOME-BASED WORKOUT ENHANCEMENT

Nicolei Faith Abot
De La Salle University - Dasmariñas, Cavite, Philippines
nicoleifaith@gmail.com

ABSTRACT

The rapid increase in home-based fitness routines has created a need for real-time posture correction solutions to ensure exercise effectiveness and prevent injuries. This study addresses the gap by developing *FitPose*, a computer vision-based system employing Human Pose Estimation (HPE) and machine learning for real-time posture assessment and feedback during foundational exercises. The system, accessible via a mobile application, utilizes camera-based input to analyze user movements in real time, classify posture accuracy, and deliver immediate corrective guidance. *FitPose* was developed through an Agile Software Development Life Cycle, integrating lightweight deep learning models to ensure mobile efficiency and iterative refinements based on user and expert feedback. The system was evaluated using the ISO/IEC 25010 Software Quality Model through surveys of beginner fitness enthusiasts and professional coaches, assessing reliability, usability, security, portability, performance efficiency, and functional suitability. *FitPose* achieved "Very High" ratings in usability (mean = 4.68) and security (mean = 4.62), and "High" ratings in real-time feedback accuracy and overall system performance. Respondents emphasized the application's intuitive interface, reliable feedback, and strong data protection, while identifying opportunities for enhancements such as audio cues and expanded exercise libraries. These results demonstrate that *FitPose* effectively addresses major challenges faced by home exercisers—offering accessible, trainer-independent posture correction—while providing a robust platform for future developments in automated fitness coaching and injury prevention.

KEYWORDS

FitPose, Computer Vision, Human Pose Estimation (HPE), Real-Time Posture Assessment, Home-Based Workout Enhancement, Machine Learning, Mobile Application, Posture Correction

1 Introduction

Technology has brought a significant transformation in fitness management with the rapid adoption of wearable devices and digital interventions providing users with data-driven insights into their physical activity. These devices allow user to monitor real-time quantitative metrics such as step counts, calorie expenditure, heart rate variability, and sleep patterns. Western et al. highlighted that these digital interventions serve as cost-effective platforms for promoting physical activity, enabling behavior-based remote interventions customized to individual needs across various socioeconomic groups [15].

Despite these advancements, there is still a critical gap that remains in the qualitative assessment of exercise, specifically, the accuracy of posture and form. While current devices except at tracking how much exercise is performed, they often fail to monitor how well it is executed. Stogdon (2024) emphasizes that proper posture is essential for distributing physical stress evenly across the body [12]. Proper alignment improves muscle activation, respiration, and circulation, therefore optimizing the effectiveness of the workout. However, improper posture may result in muscle imbalances, reduced performance, and an increased likelihood of chronic musculoskeletal conditions, including joint discomfort and muscle strain. This issue is especially critical for individuals exercising at home who do not have direct supervision from professional trainers. Without expert guidance, individuals may unknowingly perform exercise with incorrect form, rendering their routines ineffective or dangerous

The importance of this issue was raised by the increasing popularity of home-based fitness, a trend that has been notably accelerated during the COVID-19 pandemic. Valeriani et al. reported that pandemic-induced lifestyle changes, and this includes the disrupted routines and restricted access to gyms, led to a significant utilization of online tutorials, home equipment, and exergames [14]. Even as restrictions have eased, home workouts remain a preferred practice due to their efficiency and cost-effectiveness, with many individuals opting to invest in personal equipment rather than returning to gym memberships. However, the financial constraints that make home workouts attractive also often preclude hiring personal trainers, leaving a large demographic without access to professional correction.

To address these challenges, this study, *FitPose*, a Human Pose Estimation (HPE) system designed to provide real-time, trainer-independent posture assessment. Unlike standard trackers, *FitPose* utilizes computer vision to analyze the user's biomechanics via a camera-based setup. By comparing live movements against predefined correct forms, the system detects deviations and delivers immediate corrective feedback. This approach aims to bridge the gap between professional coaching and home-based fitness, ensuring that users can achieve their health goals safely and effectively.

2 Related Work

2.1 Shifts in Fitness Behavior

The global fitness landscape has undergone a significant transformation, with a marked shift towards home-based workout. This trend is driven primarily by convenience, cost-

effectiveness, and flexibility, and was accelerated by the COVID-19 pandemic. Foreign surveys indicate that 37.9% of fitness participants prefer exercising at home [3], while 87.3% of home-gym owners favor their setup over public facilities due to comfort and efficiency [2]. In the Philippines, this trend mirrors global patterns; Kurata et al. found that self-motivation and social media influence gave made home workouts a sustainable preference for Gen Z Filipinos [5].

However, the absence of professional supervision in home settings presents a critical challenge. A survey by the Philippine-based platform REBEL revealed that 67% of home-based fitness enthusiasts failed to meet their goals due to lack of proper guidance and motivation [10]. Without real-time feedback, users are prone to incorrect posture, which significantly increases the risk of injury and reduces workout effectiveness [12].

2.2 AI and Pose Estimation in Fitness

To address the lack of supervision, recent research has leveraged Artificial Intelligence (AI) and Computer Vision (CV). Tharatipyakul et al. demonstrated the efficacy of deep learning-based pose estimation for providing movement feedback, emphasizing its potential to replace in-person monitoring [13]. Similarly, Luangaphirom et al. and Kanase et al. utilized frameworks like MediaPipe to develop real-time counting and correction systems, proving that AI-driven applications can effectively assess posture and provide corrective guidance [7][4].

Specific applications include "Verum Fitness by Flores et al., which enhances safety through real-time mobile feedback [1], and the work of Shih et al., which uses OpenPose to estimate yoga pose difficulty for self-practice [11]. In the local context, Opiña and Fajardo utilized BlazePose for physical education, demonstrating how CV can assist educators in monitoring students remotely [9]. Furthermore, Ompoc and Aguinaldo found that AI-driven tools in physical education significantly improved student engagement and performance by offering personalized feedback [8].

2.3 Research Gap

While existing systems demonstrate the capability of AI in fitness, limitations remain. Many current solutions are either confined to high-end laboratory settings [6], lack real-time mobile integration, or are not tailored to the specific infrastructure and demographic needs of developing regions like the Philippines. *FitPose* aims to bridge this gap by developing a lightweight, mobile-accessible Human Pose Estimation system that provides immediate, trainer-independent feedback, democratizing access to professional-grade posture correction for home-based users.

3 System Architecture and Methodology

3.1 System Architecture

FitPose was developed using the Agile Software Development Life Cycle (SDLC) to ensure iterative refinement of posture detection models. The system follows a modular architecture comprising three core stages: (1) Data Acquisition, where live video feed is captured via webcam; (2) Processing Engine, which utilizes MediaPipe for pose estimation and a custom feature engineering pipeline; and (3) Feedback & Storage, where machine learning classifiers determine posture quality to generate real-time audiovisual cues and log performance data to Google Firebase.

Figure 1: Agile SLDC Model

The operational flow ensures low latency processing suitable for mobile deployment. Input frames undergo preprocessing (resizing, noise reduction) before 33 anatomical key points are extracted. These key points are transformed into biomechanical features (angles, velocities) which serve as inputs for the classification models.

3.2 Data Gathering and Preprocessing

The dataset was compiled from publicly accessible repositories, primarily Roboflow Universe, and supplemented by self-recorded sequences for specific movements. Data was manually annotated by licensed fitness professionals to distinguish between "Correct" and "Incorrect" forms (e.g., "hips too low" in planks)

Model	Data Source	Sample Size (n)	Key Posture Labels
Squat	Roboflow Universe	1,313	Correct, Incorrect (Depth issues)
Push Up	Roboflow Universe	7,200	Up, Down, Bad
Plank	Roboflow Universe	645	Correct, Hips Too Low, Hips Too High

Bicep Curl	Self-Recorded	5,265	Half-Rep, Lean, Swing
Lunge	Roboflow Universe/Self-Recorded	1,107	Correct, Incorrect (Knee Alignment)
Shoulder Press	Roboflow Universe	440	Good, Elbows Flaring Out

Table 1: Summary of Dataset and Labeling

3.3 Algorithm and Feature Engineering

Raw spatial coordinates (x, y, z) from MediaPipe are converted into high-level biomechanical indicators to ensure model robustness against camera distance and angle variations.

3.3.1 Mathematical Formulations. The following engineered features are computed for every frame

Joint Angel Calculation: Computed using the dot product of vectors formed by three key points (e.g., hip-knee-ankle).

$$\theta = \cos^{-1}\left(\frac{\vec{v}_1 \cdot \vec{v}_2}{\|\vec{v}_1\| \cdot \|\vec{v}_2\| + \varepsilon}\right) \cdot \frac{180}{\pi}$$

Torso Inclination: Derived using the arctangent function to measure forward/backward lean. $(z\Delta, y\Delta)$ represents the depth and vertical differences between shoulder and hip.

$$\phi = \text{atan2}(z_\Delta, y_\Delta) \cdot \frac{180}{\pi}$$

Angular Velocity: Used to detect rushed repetitions or momentum abuse.

$$\omega = \frac{\theta(t) - \theta(t - \Delta t)}{\Delta t + \varepsilon}$$

Motion Stability: Quantifies movement smoothness.

$$\text{MC} = 1 - \frac{\sigma(\theta_i)}{\max(\theta_i) - \min(\theta_i) + \varepsilon}$$

3.4 Machine Learning Classification

To achieve real-time latency on mobile devices while maintaining high classification accuracy, *FitPose* implements a hybrid hierarchical architecture. This design strategy decouples the computational burden of visual perception from the logic of biomechanical assessment. Rather than relying on a monolithic end-to-end deep learning model, which is often computationally expensive and prone to latency on edge devices, the system pipeline is divided into two distinct phases: (1) **Biomechanical Feature Extraction** using pre-trained deep learning, and (2) **Semantic Posture Classification** using efficient statistical machine learning.

This separation of concerns allows the system to leverage the state-of-the-art spatial tracking capabilities of Convolutional Neural Networks (CNNs) while utilizing lightweight statistical classifiers to interpret those spatial coordinates. The pipeline transforms raw, unstructured video data into structured geometric features, which are then analyzed to determine the quality of exercise execution.

3.4.1 Pose Estimation (Feature Extraction). The foundational layer of the system relies on MediaPipe, a high-fidelity solution for body tracking. This phase utilizes a pre-trained **Convolutional Neural Network (CNN)** to process the input RGB video stream. The CNN functions as a feature extractor, mapping the raw pixel data into a structured vector space consisting of 33 distinct anatomical key points (landmarks). It employs discrete convolution to extract spatial features (edges, joints) from the input image I using kernel K:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) \cdot K(m,n)$$

This process abstracts the user's physical environment (lighting, background) away, isolating only the skeletal geometry required for analysis

3.4.2 Classification Algorithms. Once the skeletal key points are extracted, the system transitions to the classification phase. *FitPose* employs **three distinct statistical algorithms**, each selected to match the specific biomechanical complexity and error patterns of different exercises. Unlike the initial deep learning phase, these classifiers operate on engineered features (angles, velocities) rather than raw images, ensuring rapid inference times.

Random Forest (Ensemble Learning): Applied to **Squat**, **Push-up**, **Plank**, and **Shoulder Press** models. This method was selected for its ability to handle high-dimensional biomechanical data and resist overfitting. The final class prediction $\hat{y}$ is determined by majority voting among B decision trees:

$$\hat{y} = \text{mode}\{h_b(x)\}_{b=1}^{B}$$

Where $h_b(x)$ is the prediction of the b-th tree.

K-Nearest Neighbors (KNN): Applied to the **Bicep Curl**. KNN classifies the current frame by calculating its geometric similarity to a labeled database of known postures (e.g., "Good Form" vs.

"Swinging"). It utilizes the Euclidean distance metric to identify the closest training examples, making it highly effective for exercises where repetitive motion patterns are the primary indicator of correctness

$$d(x, x_i) = \sqrt{\sum_{j=1}^{n} (x_j - x_{ij})^2}$$

Logistic Regression: Applied to the **Lunge Error** model for binary classification (Correct vs. Incorrect). It models the probability P of correct form using the sigmoid function:

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-(w^\top x + b)}} [1]$$

4 Experimental Setup and Evaluation

4.1 Experimental Development Environment

To validate the proposed architecture, the development environment was engineered to support high-fidelity computer vision on resource-constrained mobile devices. The software stack was selected to prioritize efficient on-device inference within the Android ecosystem.

4.1.1 Mobile Architecture. The application frontend was developed using Flutter, leveraging the Dart language. While Flutter offers cross-platform capabilities, this implementation was specifically optimized and compiled for Android devices, ensuring direct access to hardware acceleration features required for real-time video processing.

4.1.2 Computer Vision Engine. The main pose estimation pipeline includes MediaPipe, a graph-based machine learning framework designed for mobile edge computing. Unlike server-side models, MediaPipe runs on-device, processing 33 different anatomical landmarks in real-time (>30 FPS) to protect user data privacy and reduce network delay.

4.1.3 Backend Infrastructure. Google Firebase served as the backend-as-a-service (BaaS), handling NoSQL database operations for workout logs and authentication. Python was utilized for the offline training of the classification models, employing libraries such as Scikit-learn for statistical classifiers and OpenCV for image preprocessing.

4.2 Dataset Acquisition and Preprocessing Pipeline

The reliability of a supervised learning model is closely dependent upon the quality of its training data. The *FitPose* dataset was created using a hybrid acquisition technique that combined open-source repositories with customized data collection to meet particular biomechanical characteristics.

4.2.1 Data Sourcing. The base dataset was aggregated from **Roboflow Universe**, providing a diverse foundation of labeled exercise images. To mitigate domain shift, this was supplemented with self-recorded sequences capturing specific local environmental conditions (e.g., varying indoor lighting, limited space).

4.2.2 Preprocessing and Augmentation. To increase the model's generalization capability and reduce overfitting, the training data was thoroughly preprocessed. Images were shrunk to a common resolution in order to standardize the input dimensions. Data augmentation approaches were used, including:

1. **Horizontal Flipping**: To ensure the model recognizes exercises performed in either direction.

2. **Random Rotation**: To simulate non-ideal camera placement.

3. **Brightness Adjustment**: To account for poor lighting conditions typical in home environments.

4.3 Participant Demographics and Stratification

To ensure a comprehensive assessment of the system's utility, the study used a stratified sample technique for User Acceptance Testing (UAT). A total of 54 respondents were separated into two distinct groups to examine mutually exclusive system attributes:

A. **Fitness Enthusiast** (n = 38) – 70.4%

This group included beginner to intermediate users who mostly exercised at home. Their evaluation focused on usability and user experience (UX), specifically determining whether real-time feedback was understandable and inspiring.

B. **Fitness Professionals** (n = 16) – 29.6%

This group consisted of certified trainers and coaches. Their task was to assess the system's Functional Suitability, which involved determining whether the AI's biomechanical corrections were consistent with established biomechanics criteria.

4.4 Evaluation Metrics

The system's efficacy was measured using a dual-framework approach, combining qualitative software quality standards with quantitative statistical performance metrics.

4.4.1 ISO/IEC 25010 Software Quality Standards. To assess the application's readiness for deployment, the **ISO/IEC 25010** standard was utilized. Participants rated the system on a 5-point Likert scale across critical characteristics:

1. **Functional Suitability**: The degree to which the AI provides correct and helpful feedback.

2. **Performance Efficiency**: The system's responsiveness and resource utilization.

3. **Usability**: The ease of learning and operating the application.

4. **Security**: The protection of user data and video privacy.

4.4.2 Classification Performance Metrics. The evaluation of the developed classification model relies on the classification metric scores attained, namely, accuracy, precision, recall, and F1-score. These scores determine whether the model is capable of making accurate classifications based on the trained dataset, thus answering the question regarding its effectiveness in the study's purpose.

Accuracy	(True At-Risk + True Not At-Risk) / All Predictions
Precision	True At-Risk / All Predicted At-Risk
Recall	True At-Risk / All At-Risk Instances
F1-Score	2 * (Precision * Recall) / (Precision + Recall)

Table 2: Metrics Equation

4.5 Experimental Results

4.5.1 Quantitative Analysis: Model Performance. The performance of the exercise-specific classifiers indicates a high degree of accuracy for compound movements. The **Squat** model, utilizing the Random Forest algorithm, achieved a near-perfect accuracy of **98.9%**, effectively distinguishing between proper depth and shallow repetitions. The **Bicep Curl** model, powered by KNN, achieved **96.0%** accuracy, successfully classifying dynamic errors such as "swinging" (momentum abuse).

Exercise	Algorithm	Accuracy	Precision	Recall	F1-Score
Squat	Random Forest	98.9%	0.99	0.99	0.99
Lunge (Error)	Logistic Regression	97.2%	0.95	0.98	0.96
Bicep Curl	KNN	96.0%	0.96	0.96	0.96
Shoulder Press	Random Forest	92.0%	0.91	0.93	0.92
Push-Up	Random Forest	90.9%	0.93	0.89	0.91
Plank	Random Forest	87.0%	0.88	0.85	0.86

Table 3: Performance Metrics of Machine Learning Classifiers

The Plank model exhibited the lowest accuracy (87.0%), primarily due to class imbalance in detecting the "Hips Too High" error, which appeared less frequently in the training data than "Hips Too Low."

4.5.2 Qualitative Analysis: ISO/IEC 25010 Evaluation. The aggregated survey results from the 54 respondents are presented in **Table 4**. The system achieved a "Very High" overall rating ($\bar{x}$=4.56).

Evaluation Criteria	Enthusiasts ($\bar{x}$)	Professionals ($\bar{x}$)	Combined Mean	Interpretation
Security	4.62	4.59	4.61	Very High
Usability	4.68	4.52	4.60	Very High
Performance Efficiency	4.60	4.47	4.54	Very High
Functional Suitability	4.56	4.45	4.51	Very High
Portability	4.46	4.54	4.50	Very High
Reliability	4.55	4.38	4.47	Very High

Table 4: Summary of User Evaluation (ISO/IEC 25010)

The data reveals a slight divergence in priorities: **Fitness Enthusiasts** rated **Usability** highest ($\overline{4.68}$), reflecting the intuitive design of the mobile interface. In contrast, **Professionals** rated **Security** ($\overline{4.59}$) and **Portability** ($\overline{4.54}$) as the system's strongest attributes, validating its potential for safe deployment in diverse training environments.

5 Discussion

5.1 Performance Analysis and Biomechanical Accuracy

The experimental results demonstrate that the proposed *FitPose* system successfully translates raw computer vision data into actionable fitness feedback. The disparity in accuracy across different exercises provides critical insight into the capabilities of mobile-based Pose Estimation.

The **Squat** and **Bicep Curl** models achieved near-perfect classification rates (98.9% and 96.0% respectively). This success can be due to the high visibility of the main joints (hips, knees, and elbows) as well as the distinct geometric disparities between proper and erroneous shapes. For example, the biomechanical characteristic of "depth" in a squat is easily quantifiable using the hip-knee vector angle, which the Random Forest classifier used efficiently.

In contrast, the Plank model had the lowest performance (87.0%). This demonstrates an intrinsic issue in 2D posture estimation involving depth perception in horizontal views. A frontal camera perspective makes it more difficult to detect "Hips Too High" or "Hips Too Low" than a side profile.

5.2 User Acceptance and the "Coaching Gap"

The ISO/IEC 25010 evaluation results show a clear conclusion that while professionals prioritized security and portability, fitness enthusiasts preferred usability.

The high Usability score (4.68) suggests that *FitPose* effectively lowers the technical barrier to entry for AI-driven fitness. Previous systems often required complex setups or wearable sensors, which discouraged casual users. By utilizing a standard smartphone camera, *FitPose* achieves high "perceived usefulness." The qualitative feedback indicated that users felt more confident exercising with the app than without it, validating the hypothesis that real-time feedback acts as a motivational proxy for a human trainer.

However, the slight divergence in **Reliability** ratings (Professionals rated it lower at 4.38) indicates that while the system is excellent for beginners, seasoned coaches may find the occasional false positive (due to lighting or clothing) less acceptable for professional athlete monitoring.

5.3 Environmental and Technical Limitations

While the system performs robustly under controlled conditions, deployment in real-world home environments introduced several constraints:

Occlusion and Camera Framing: The system requires the user's full body to be visible. In small home spaces, users often struggled to position the phone far enough away to capture their feet and head simultaneously. If the ankles were cut off from the frame, the **Squat** and **Lunge** classifiers would fail to compute the necessary angles.

Lighting Sensitivity: The MediaPipe framework relies on RGB input. During testing, low-light conditions or strong backlighting (e.g., exercising in front of a window) significantly degraded the key point detection stability, leading to "jittery" feedback.

2D vs. 3D Biomechanics: As the system uses a single camera, it infers 3D positioning from 2D coordinates. This method is highly effective for sagittal plane movements (like Lunges) but struggles with transverse plane rotations. For example, if a user's knee caves inward (valgus collapse) during a squat while facing the camera, the system detects it easily. However, if they are facing sideways, this specific error becomes invisible to the 2D vector analysis.

5.4 Comparative Advantage

Compared to standard fitness trackers that monitor quantitative outputs (heart rate, step count), *FitPose* provides qualitative biomechanical assessment. Where a smartwatch might confirm that a user performed 10 reps, *FitPose* confirms whether those reps were safe and effective. This distinction positions the system not as a replacement for wearable technology, but as a necessary complement that addresses the "quality" aspect of the quantity-quality equation in fitness.

6 Conclusion and Future Work

6.1 Conclusion

The primary aim of this study was to design and evaluate *FitPose*, an AI-powered mobile application designed to bridge the gap in real-time, home-based fitness guidance. Through a comprehensive evaluation involving both fitness enthusiasts and professionals, the system successfully demonstrated its capability to provide accessible and accurate posture correction.

Findings from the ISO/IEC 25010 evaluation confirm that *FitPose* operates with high proficiency. Fitness enthusiasts rated the system "Very High" across most domains, particularly in **Usability** (M=4.68) and **Security** (M=4.62), indicating that the application is intuitive and trusted by users. Similarly, fitness professionals validated the system's feasibility for coaching contexts, awarding high scores for **Security** (M=4.59) and **Portability** (M=4.54). While professionals rated usability slightly lower than enthusiasts, their feedback affirms *FitPose*'s potential as a supplementary tool for remote coaching.

The machine learning results reinforce this reliability. The **Squat** (98.9% accuracy) and **Bicep Curl** (96.0% accuracy) models exhibited exceptional precision, proving that the

system is robust enough for real-time deployment. Although the **Plank** model showed performance variability due to dataset imbalance, the overall architecture effectively distinguishes between proper and improper forms. Ultimately, *FitPose* meets its objectives by offering a practical, accessible, and AI-driven approach to exercise correction, effectively supporting both independent home exercisers and professional trainers.

6.2 Future Work

The future development and research for the *Fitpose* system will focus on four key areas to enhance its effectiveness. First is technical robustness and sensing, which includes integrating adaptive lighting normalization to improve key point detection in sub-optimal conditions. Future research will also explore depth-sensing technologies to overcome limitations of 2D planar analysis for more accurate Z-axis error detection, and the backend infrastructure will be optimized for better cloud synchronization and analytics generation.

The second focus is the expansion of the exercise library, which will move beyond bodyweight movements to include equipment-based exercises. This expansion involves detecting form adjustments based on specific weights and incorporating support for common gym machinery like the leg press and lat pulldowns to accommodate a wider user base. The third focus is the enhanced user interaction and professional tools, which will introduce audio-based voice feedback for hands-free correction, and a dedicated coach dashboard will be developed for professionals to track client progress and use in-app messaging. And the last focus is the longitudinal research which is a plan to observe the long-term impact of *Fitpose* on user fitness outcomes and injury prevention rates, providing stronger empirical evidence of the system's practical benefits.

ACKNOWLEDGEMENTS

First and foremost, the researchers offer their deepest praise and gratitude to Almighty God for providing the wisdom, strength, and resilience to overcome the challenges encountered during this study and for making this research possible.

Appreciation is further directed toward classmates for their assistance and camaraderie, and to friends for their constant encouragement and motivation when difficulties were faced. Finally, this work is dedicated to the families of the researchers for their unwavering love, financial support, and understanding throughout this academic journey.

REFERENCES

[1] Flores, J., Sanchez, P., & Kumar, S. (2021). An AI-powered mobile fitness safety and improvement application. *International Journal of Fitness Technology, 19*(5), 245–257. https://www.semanticscholar.org/paper/Verum-Fitness%3A-An-AI-Powered-Mobile-Fitness-Safety-Flores-Hall/ed5504727caf5ccc41ecf84315689fb1eaa30fcf

[2] Garage Gym Experiment. (2023, February 20). *Other 2023 home gym survey results*. https://garagegymexperiment.com/2024/02/20/other-2023-home-gym-survey-results/

[3] Gymless.org. (2023, October 3). *New survey finds 52.6% of people prefer working out outside the gym* [Press release]. PR Newswire. https://www.prnewswire.com/news-releases/newsurvey-finds-52-6-of-people-prefer-working-out-outside-the-gym-301941292.html

[4] Kanase, A., Patel, R., & Desai, S. (2021). Pose estimation and correcting exercise posture. *Journal of Machine Learning and Fitness, 7*(3), 124-137. https://www.itm-conferences.org/articles/itmconf/pdf/2021/05/itmconf_icacc2021_03031.pdf

[5] Kurata, Y. B., Ong, A. K. S., Cunanan, A. L. M., Lumbres, A. G., Palomares, K. G. M., Vargas, C. D. A., & Badillo, A. M. (2023). Perceived behavior analysis to boost physical fitness and lifestyle wellness for sustainability among Gen Z Filipinos. *Sustainability, 15*(18), 13546. https://doi.org/10.3390/su151813546

[6] Liu, L., Dai, Y., & Liu, Z. (2024). Real-time pose estimation and motion tracking for motion performance using deep learning models. *Journal of Intelligent Systems, 33*(1), Article 20230288. https://doi.org/10.1515/jisys-2023-0288

[7] Luangaphirom, P., Vichai, C., & Kittisak, D. (2024). Real-time weight training counting and correction using MediaPipe. *Journal of Sports Technology, 11*(1), 56-67. https://www.researchgate.net/publication/379064729

[8] Ompoc, M., & Aguinaldo, J. (2025). Evaluating the effectiveness of AI-driven applications in enhancing the physical fitness of tertiary physical education students. *H.R.Pub.* https://www.researchgate.net/publication/388372239

[9] Opiña, M., & Fajardo, A. (2023). Real-time pose detection for physical education using BlazePose and OpenCV for angle tracking. *Philippine Journal of Educational Technology, 6*(2), 100–112. https://www.researchgate.net/publication/373054021

[10] REBEL Team. (2021). Survey results on the effectiveness of home-based exercise routines in the Philippines. *REBEL Fitness Report*. http://www.takbo.ph/2021/07/rebel-health-fitnessapp-for-home-workouts/

[11] Shih, T., Lin, H., & Yang, Z. (2024). A yoga pose difficulty level estimation method using OpenPose for self-practice systems. *Computers in Biology and Medicine, 47*, 112–121. https://www.mdpi.com/2078-2489/15/12/789

[12] Stogdon, M. (2024, December 16). *The importance of posture during exercise*. PROMiXX. https://www.promixx.com/en-gb/blogs/academy/importance-of-posture-during-exercise

[13] Tharatipyakul, P., Srikaewsiew, S., & Pongnumkul, S. (2024). Deep learning-based human body pose estimation in providing movement feedback. *IEEE Transactions on Neural Networks andLearning Systems, 35*(2), 123-136. https://www.sciencedirect.com/science/article/pii/S2405844024126205

[14] Valeriani, F., Protano, C., De Giorgi, A., Mazzeo, E., Liguori, G., Spica, V. R., ... & Galle, F. (2023). Analysing features of home-based workout during COVID-19 pandemic: A systematic review. *Public Health, 222*, 100–114.

[15] Western, M., Smith, J., & Johnson, R. (2021). The effectiveness of digital interventions for increasing physical activity in individuals of low socioeconomic status: A systematic review and metaanalysis. *Journal of Physical Activity and Health, 18*(4), 214–225. https://www.researchgate.net/publication/356068928

GaitAware: A Lightweight Hybrid Gait Analysis System Using BlazePose-BiLSTM and CNN for Identifying Potential Gait Abnormalities

Timoty James L. Bautista
College of Information and Computer Studies
De La Salle University - Dasmariñas
Dasmariñas, Cavite, Philippines
BTL1223@dlsud.edu.ph

John Rebb B. Binanitan
College of Information and Computer Studies
De La Salle University - Dasmariñas
Dasmariñas, Cavite, Philippines
BJB0392@dlsud.edu.ph

Zalzon T. Espino III
College of Information and Computer Studies
De La Salle University - Dasmariñas
Dasmariñas, Cavite, Philippines
EZT0434@dlsud.edu.ph

Tita R. Herradura
College of Information and Computer Studies
De La Salle University - Dasmariñas
Dasmariñas, Cavite, Philippines
trherradura@dlsud.edu.ph

ABSTRACT

Gait abnormalities significantly impact mobility and quality of life, yet specialized assessment remains largely inaccessible due to high costs and technical complexity. To address this, this study introduces GaitAware, a mobile application that utilizes a BlazePose–BiLSTM–CNN pipeline to provide accessible gait analysis via smartphone. The CNN gait pattern classifier achieved an overall accuracy of 80.91% across 1,540 test samples, while the BiLSTM model for detecting joint abnormalities demonstrated 75% accuracy and a 100% recall in identifying abnormal gait sequences. While the chosen sensitivity threshold led to a 50% precision rate, the high recall ensures that potential abnormalities are successfully flagged for further review. Validated against ISO/IEC 25010 quality standards and clinical insights from licensed physiotherapists, the application proved to be a user-friendly tool that aligns with professional expectations. The results demonstrate that GaitAware offers a reliable, data-driven approach to preliminary gait screening that could bridge the gap between clinical expertise and community accessibility.

KEYWORDS

Computer Vision, Gait Analysis, Machine Learning, Walking, Gait Abnormalities

1 INTRODUCTION

Walking is a basic yet essential part of everyday life, enabling individuals to perform daily activities and remain connected with their community. A person's gait varies due to factors such as age, mood, and personality [1]. While these variations are often normal, certain changes in gait may indicate underlying neurological or musculoskeletal conditions.

Gait analysis plays a crucial role in assessing walking patterns and identifying potential health issues. Traditionally performed in controlled clinical environments by physiotherapists, it has evolved with technologies such as videotaped observational gait analysis (VOGA), 3D motion capture, and dynamic electromyography [2]. Despite these advancements, access to such services in the Philippines remains limited, particularly in rural and remote areas where medical resources and specialists are scarce [3, 4]. As a result, gait-related conditions such as Parkinsons disease may remain undiagnosed or underreported, even though they affect a portion of the population [5].

To address these limitations, recent studies have explored computer vision and machine learning approaches for more accessible gait analysis. For instance, [6] developed a markerless gait classification system using 2D video, utilizing deep learning techniques to classify gait types from visual representations. Their system achieved performance comparable to state-of-the-art methods while significantly reducing model complexity, memory requirements, and execution time, demonstrating its suitability for remote applications. Meanwhile, [7] proposed a vision-based approach that utilizes human pose estimation and joint angle analysis to develop normative gait parameters from RGB video. Their

method enables cycle-wise detection of abnormalities using statistical measures such as mean and standard deviation, allowing precise identification of joint-level deviations throughout the gait cycle. Together, these studies highlight the potential of combining pose-based analysis and video-based classification to enable accessible and automated gait analysis.

Despite these advancements, existing approaches are often developed independently and focus on either classification or joint-level analysis. This separation limits their ability to provide detailed reports of gait abnormalities in practical settings. The proposed approach complements existing tools by integrating both perspectives within a single framework. This enables not only the identification of gait abnormalities but also the joints of where these deviations occur.

The study proposes GaitAware, a lightweight and mobile-compatible gait analysis system that utilizes computer vision and machine learning to detect joint deviations and classify gait abnormalities from video input. By using smartphone-based video capture, the system aims to provide accessible preliminary insights that can support early awareness and encourage timely medical consultation, while not serving as a substitute for professional clinical diagnosis.

2 METHODOLOGY

2.1 Data Collection

This study utilized three publicly available datasets: GPJATK, GAIT-IT, and GAIT-IST. The GPJATK dataset contains 166 normal gait sequences from 32 individuals and was used to train the BiLSTM autoencoder to learn normal gait patterns [8]. Meanwhile, GAIT-IT and GAIT-IST datasets were used for gait classification, providing over 1,300 gait sequences represented as Skeleton Energy Images (SEIs) comprising of 1,320 diplegic, 1,231 hemiplegic, 1,140 neuropathic, 371 normal, and 1,759 Parkinsonian gait images [9, 10].

2.2 Data Preprocessing

Mediapipe Blazepose landmarker was used to extract 3-dimensional coordinates for key body joints from each video frame.

Gait sequences were normalized using Z-score normalization to standardize feature values across all frames. This ensured that each feature has zero mean and unit variance, which helped the BiLSTM model learn more effectively and generalize across different subjects. The normalized sequences were then segmented into fixed-length temporal windows using a sliding window approach to capture temporal dependencies in gait patterns.

For the CNN model, features were extracted per frame by drawing high-contrast lines between specific joint pairs to create a human stick-figure silhouette which represents the overall motion of the person's gait sequence, as shown in figure 1.

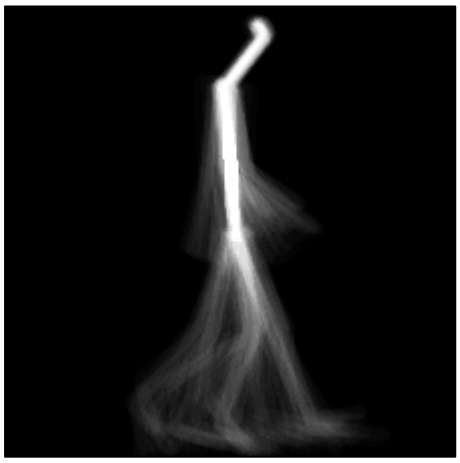


Figure 1: Skeleton Energy Images (SEIs)

2.3 Model Architecture

The proposed system employs a hybrid architecture consisting of a BiLSTM autoencoder for abnormality detection and a convolutional neural network (CNN) for gait classification.

Figure 2 shows the CNN model designed to extract spatial features from Skeleton Energy Images (SEIs) of size 224×224×3. It consists of multiple convolutional layers with batch normalization and ReLU activation, followed by a fully connected layer and a softmax output layer for multi-class classification.

Layer Type	Key Parameters	Output Shape
Input	(224, 224, 3)	(None, 224, 224, 3)
Conv2D	32 filters, 3×3, stride 2, ReLU	(None, 112, 112, 32)
Batch Normalization	—	(None, 112, 112, 32)
Conv2D	32 filters, 3×3, stride 2, ReLU	(None, 56, 56, 32)
Batch Normalization	—	(None, 56, 56, 32)
Conv2D	32 filters, 3×3, stride 2, ReLU	(None, 28, 28, 32)
Batch Normalization	—	(None, 28, 28, 32)
Conv2D	32 filters, 3×3, stride 2, ReLU	(None, 14, 14, 32)
Batch Normalization	—	(None, 14, 14, 32)
Conv2D	64 filters, 3×3, stride 2, ReLU	(None, 7, 7, 64)
Batch Normalization	—	(None, 7, 7, 64)
Flatten	—	(None, 3136)

Dense	512 units, ReLU	(None, 512)
Dropout	rate = 0.5	(None, 512)
Dense (Output)	num_classes, Softmax	(None, 5)

Figure 2: CNN model architecture

Figure 3 shows the BiLSTM autoencoder used to model temporal gait patterns. It consists of stacked bidirectional LSTM layers forming an encoder-decoder structure. The encoder compresses temporal features into a latent representation, while the decoder reconstructs the sequence. The reconstruction error is computed by comparing the reconstructed output with the original input sequence using Mean Squared Error (MSE). This is calculated at the final output stage of the model, not within a specific layer. A low reconstruction error indicates normal gait patterns, while a high error suggests potential gait abnormalities.

Layer Type	Key Parameters	Output Shape
Input	shape=(sequence_length,num_features)	(None, sequence_length, num_features)
Bidirectional LSTM	LSTM(64, return_sequences=True)	(None, sequence_length, 128)
Dropout	rate=0.2	(None, sequence_length, 128)
Bidirectional LSTM	LSTM(32, return_sequences=False)	(None, 64)
Dropout	rate=0.2	(None, 64)
Dense	units=latent_dim, activation='relu'	(None, latent_dim)
RepeatVector	n=sequence_length	(None, sequence_length, latent_dim)
Bidirectional LSTM	LSTM(32, return_sequences=True)	(None, sequence_length, 64)
Dropout	rate=0.2	(None, sequence_length, 64)
Bidirectional LSTM	LSTM(64, return_sequences=True)	(None, sequence_length, 128)
Dropout	rate=0.2	(None, sequence_length, 128)
TimeDistributed Dense	Dense(num_features)	(None, sequence_length, num_features

Figure 3: BiLSTM autoencoder model architecture

3 RESULTS AND DISCUSSION

3.1 CNN Model

The final performance of the CNN model was evaluated on a dedicated, held-out test set that was not used during the training or hyperparameter optimization phases. The dataset was partitioned using an 80% training/validation split and a 20% test split. The results presented here are derived exclusively from this 20% test set, which contained 1,540 gait images in total.

The training process utilized early stopping with a patience of 3 epochs, monitoring validation loss to prevent overfitting. The CNN model began to demonstrate convergence around epoch 6, where the training accuracy reached 95.7% and the validation accuracy reached 93.27%. Similarly, validation loss was recorded at 0.24, compared to a training loss of 0.21. Following this, the validation loss slightly increased in epoch 7 before stabilizing through epochs 8 and 9 at approximately 0.26, with the training loss remaining steady at 0.2. Since the validation loss did not improve 3 consecutive epochs after epoch 6, the early stopping triggered.

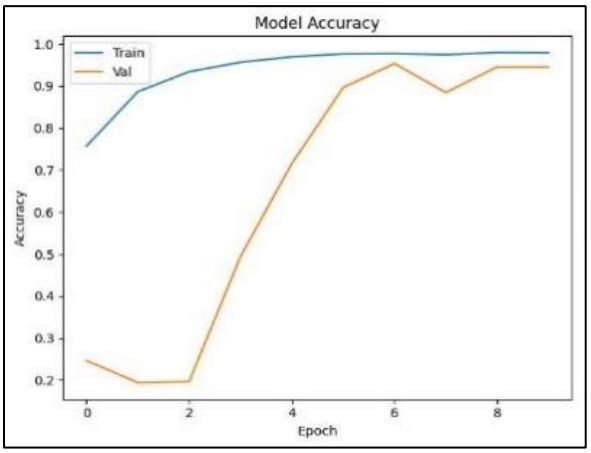


Figure 4: CNN Model Training Accuracy

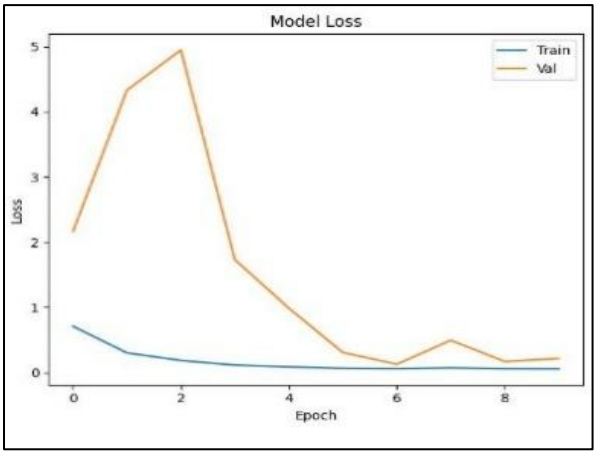


Figure 5: CNN Model Training Loss Curves

The classification model achieved an overall accuracy of 80.91% across the 1,540 test samples. Figure 6 shows that the CNN model demonstrated high efficacy in classifying Neuropathic cases, achieving the highest F1-score (0.91), supported by its high precision (0.92) and recall (0.91). High recall was also observed for the Normal (0.97) and Parkinson (0.95) classes. While the CNN model rarely missed an actual instance of these two conditions, the lower corresponding

precision values (0.73 and 0.78, respectively) suggest that these classes are common targets for misclassifications from other conditions. The Diplegic class yielded the lowest recall (0.46) and F1-score (0.58), indicating that the model failed to correctly identify approximately 54% of actual diplegic samples.

Gait Class	Precision	Recall	F1-Score	Support
Diplegic	0.78	0.46	0.58	308
Hemiplegic	0.82	0.78	0.80	318
Neuropathic	0.92	0.91	0.91	319
Normal	0.73	0.97	0.83	120
Parkinson	0.78	0.95	0.85	475

Figure 6: Classification Performance Metrics by Class

The most frequent misclassification involved the confusion between Diplegic and Parkinson gait. As shown in Figure 7. out of 308 Diplegic samples, 127 samples (41.2%) were incorrectly classified as Parkinsons. 39 samples (12.7) from Diplegic were also classified as Hemiplegic. Other less significant misclassifications include Hemiplegic to Neuropathic, where 26 actual Hemiplegic samples were incorrectly labelled as Neuropathic, and Neuropathic to Normal, where 20 actual Neuropathic samples were confused with the Normal class.

The high misclassification rate (41.2%) between Diplegic and Parkinsonian gait may be attributed to the shared biomechanical features of both conditions. Both gaits exhibit reduced stride length, increased step frequency, and a forward-leaning postural orientation. Since both conditions result in restricted, slow, and symmetrical leg movements, the model likely struggled to differentiate between the muscle-tightness and stiffness of Diplegia and the persistent slowness and reduced step size of Parkinson's.

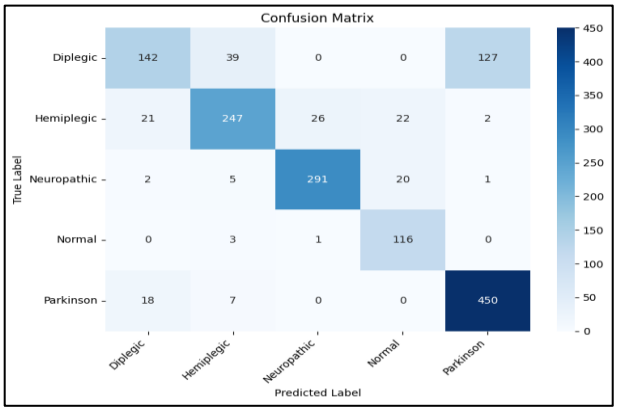


Figure 7: Confusion Matrix for Multi-Class Gait Classification

3.2 BiLSTM Model

Training performance was monitored using Mean Squared Error loss function across 52 epochs, as presented in the learning curves in Figure 8. The Training MSE decreased from an initial value of 0.53, to a final value of 0.087, while the Validation MSE decreased from 0.39 to 0.076. The Validation MSE notably remained consistently lower than the training MSE. This is expected behavior because of implementing a Dropout regularization layer within the BiLSTM architecture, which artificially increased the training loss to prevent overfitting.

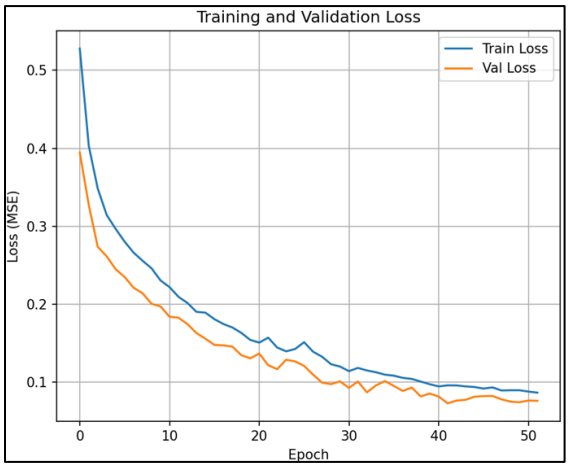


Figure 8: BiLSTM Model Loss Curves During Training

The performance of the BiLSTM model was assessed using a held-out test set containing 12 validated gait sequences (9 normal, 3 abnormal), Figure 9 shows the values of thresholds from each joint for Anomaly detection, calculated using the 85th percentile from the testing data set. The thresholds ranged from a minimum of 0.54 (Left Knee and Left Foot) to a maximum of 0.65 (Right Foot). The Right Foot required the highest reconstruction error (0.65) to be classified as abnormal, indicating the largest observed natural variability in this joint movement. In contrast, the Left Knee and Left Foot showed the lowest boundaries (0.54). This asymmetry, with higher thresholds on the right side, suggests a greater inherent kinematic variability in the normal gait cycle for the right lower limb as captured by the training data.

Joint	Threshold
Left Knee	0.54
Left Ankle	0.6
Left Foot	0.54
Right Knee	0.56
Right Ankle	0.64
Right Foot	0.65

Figure 9: Joint Anomaly Detection Threshold

Figure 10 shows that the BiLSTM model achieved an overall accuracy of 75%. It got a 100% recall by successfully identifying 3 actual abnormal gait sequences. However, the choice of the 85th percentile threshold also resulted in 3 false positives. This yielded a 50% precision and a 66.7% specificity, indicating that half of the total sequences flagged as abnormal were false alarms.

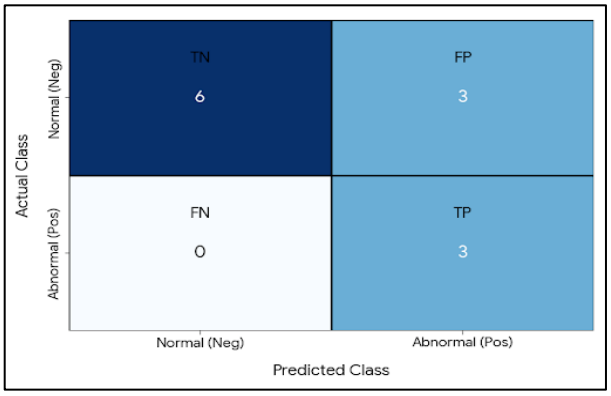


Figure 10: Confusion Matrix for Anomaly Detection Using Custom Testing Set

3.3 User Evaluation

User evaluation of the app's software quality attributes based on ISO/IEC 25010 quality standards revealed highly positive results across all dimensions. A total of 59 respondents participated in the evaluation, consisting of 20 elderly users aged above 60 years, 33 individuals from the general public aged 18 to 59, and 6 licensed physiotherapists.

Functional suitability received a composite mean of 4.66, indicating that the app effectively performs its intended functions, provides clear information, and meets user expectations. Performance efficiency scored a composite mean of 4.74, reflecting fast, smooth, and reliable app performance. The app was also found to be highly compatible with devices and other mobile applications (composite mean = 4.83), allowing seamless access to device features and sharing capabilities. In terms of interaction capability, the system achieved a composite mean of 4.78, showing that the interface is intuitive, easy to navigate, and provides clear feedback to users. Overall, respondents rated GaitAware as highly suitable, efficient, compatible, and user-friendly.

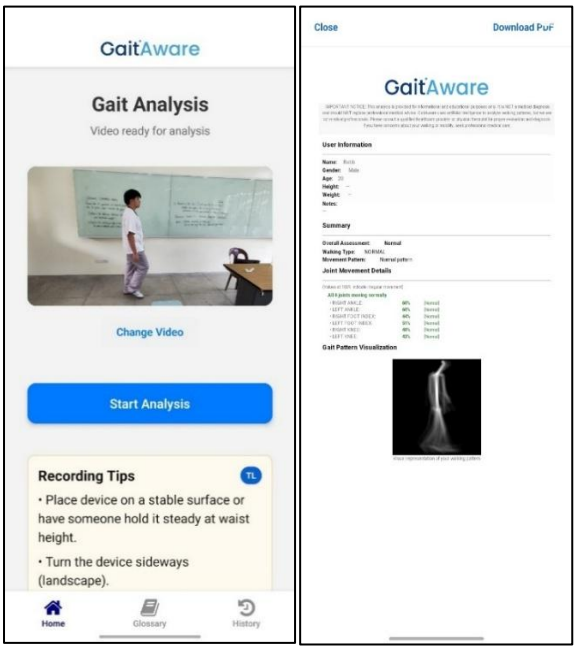


Figure 11: GaitAware mobile application UI

4 CONCLUSION

GaitAware successfully achieved its objective of being a lightweight hybrid gait analysis system. By using publicly available gait datasets, documented gait abnormalities, and consultations with a licensed physiotherapist, the researchers designed a functional and clinically aligned pipeline capable of detecting joint irregularities and classifying abnormal gait patterns.

User evaluations showed positive feedback, particularly highlighting the app's accessible and intuitive interface designed for elderly users. The combined CNN and BiLSTM approach proved feasible for mobile-based gait analysis, providing users with an initial and reliable assessment of gait abnormalities, while conservative hyperparameter tuning helped prevent overfitting due to limited data availability. Clear disclaimers supported responsible use by emphasizing that the system is not a substitute for professional diagnosis. Despite the subjectivity nature of gait analysis, real-world testing demonstrated that the results met user expectations, indicating that GaitAware is a mobile-compatible tool for accessible gait assessment.

For future work, the system may be enhanced by incorporating additional gait metrics for more detailed analysis. Model performance can also be improved by using more diverse datasets, including real-world gait videos, a wider range of abnormalities, and varied body types to improve generalization. Adding more input viewpoints may address the current limitation of left-to-right gait analysis. Finally, future studies may explore decision-fusion strategies to improve consistency between the CNN and BiLSTM models.

REFERENCES

[1] Physiopedia. (2023). Gait. https://www.physio-pedia.com/Gait

[2] Shanahan, D. (2022, May 2). The pros and cons of 4 types of gait analysis. Exer. https://www.exer.ai/posts/the-pros-and-cons-of-4-types-of-gait-analysis

[3] Alberto, N. R., Alberto, I. R., Eala, M. A., Dee, E. C., & Cañal, J. P. (2022). Availability of essential diagnostics in the Philippines. The Lancet Regional Health - Western Pacific, 19, 100375. https://doi.org/10.1016/j.lanwpc.2021.100375

[4] Tandoc III, A. (2024). Quo vadis. PJP, 9(1), 4-5. https://doi.org/10.21141/pjp2024.01

[5] Rosales, R. L., Rosales, M. E., Robles, D. S., Rodriguez, R. T., Romana, N. S., Rosales, J. B., Salazar, G. B., & Santiano, R. S. (2022). A community-based study on the prevalence and predisposing factors of Parkinson's disease in Barangay Mangilag sur, quezon province, Philippines. Clinical Parkinsonism & Related Disorders, 7, 100169. https://doi.org/10.1016/j.prdoa.2022.100169

[6] Hulleck, A. A., AlShehhi, A., El Rich, M., Khan, R., Katmah, R., Mohseni, M., Arjmand, N., & Khalaf, K. (2024). Blazepose-seq2seq: Leveraging regular RGB cameras for robust gait assessment. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 32, 1715-1724. https://doi.org/10.1109/tnsre.2024.3391908

[7] Ranjan, R., Ahmedt-Aristizabal, D., Armin, M. A., & Kim, J. (2024). Developing normative gait cycle parameters for clinical analysis using human pose estimation. 2024 International Conference on Digital Image Computing: Techniques and Applications (DICTA), 142-149. https://doi.org/10.1109/dicta63115.2024.00032

[8] Kwolek, B., Michalczuk, A., Krzeszowski, T., Switonski, A., Josinski, H., & Wojciechowski, K. (2019). Calibrated and synchronized multi-view video and motion capture dataset for evaluation of gait recognition. Multimedia Tools and Applications, 78(22), 32437-32465. https://doi.org/10.1007/s11042-019-07945-y

[9] Loureiro, J., & Correia, P. L. (2020). Using a skeleton gait energy image for pathological gait classification. 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020), 503-507. https://doi.org/10.1109/fg47880.2020.00064

[10] Albuquerque, P., Machado, J. P., Verlekar, T. T., Correia, P. L., & Soares, L. D. (2021). Remote gait type classification system using Markerless 2D video. Diagnostics, 11(10), 1824. https://doi.org/10.3390/diagnostics11101824

DermaTector: Increasing Efficiency in Diagnostics of Sun-Induced Skin Diseases Utilizing EfficientNet

Josh Mickel P. Llamera
College of Information and Computer Studies
De La Salle University – Dasmariñas
LJP1103@dlsud.edu.ph

Viktor Harold R. Marticio
College of Information and Computer Studies
De La Salle University – Dasmariñas
MVR1167@dlsud.edu.ph

Faith Moselle O. Paule
College of Information and Computer Studies
De La Salle University – Dasmariñas
PFO1498@dlsud.edu.ph

Tita R. Herradura
College of Information and Computer Studies
De La Salle University – Dasmariñas
trherradura@dlsud.edu.ph

ABSTRACT

This study developed *DermaTector*, a web-based diagnostic support tool utilizing a deep learning classifier to increase the efficiency of detecting and classifying ten common sun-induced skin diseases. Addressing the challenges of limited dermatological access in the Philippines, the system employed the EfficientNetB4 architecture via transfer learning on an 80/20 data split. The final model achieved high performance metrics on an independent test set: Accuracy of 0.8102, Precision of 0.8563, Recall of 0.8102, and F1-Score of 0.8113. User evaluation against ISO/IEC 25010 standards confirmed the system's practical utility, with the highest composite mean recorded for Interaction Capability ($\bar{x}$ = 4.74). The findings validate the system as both a high-performing machine learning model and an accessible, user-friendly public awareness tool. This research demonstrates the viability of leveraging advanced computer vision models to support early recognition and encourage professional consultation, offering a timely and significant contribution to healthcare technology initiatives in high-sun-exposure regions.

KEYWORDS

Computer Vision, Deep Learning, EfficientNetB4, Skin Disease Classification, Healthcare Technology

1 INTRODUCTION

As a tropical country situated near the equator, the Philippines experiences high levels of ultraviolet (UV) radiation throughout the year, which, combined with the outdoor lifestyle of many Filipinos, increases susceptibility to sun-induced skin diseases [1]. Although higher melanin levels in Filipinos provide partial protection against UV damage, this does not eliminate the risk of harmful effects, as evidenced by the prevalence of hyperpigmentation and melasma [2]. Despite these risks, access to specialized dermatological care remains heavily centralized in urban centers, leaving rural populations underserved and often reliant on self-care until symptoms worsen. To bridge this gap, *DermaTector* was conceptualized as an accessible, web-based diagnostic support tool for preliminary assessment and public awareness. By utilizing the EfficientNetB4 architecture, this research classifies ten specific conditions, including precancerous lesions, skin cancers, photo-induced dermatoses, and pigmentary disorders, to increase diagnostic efficiency while emphasizing the critical necessity of professional medical consultation.

2 RELATED WORKS

The application of artificial intelligence (AI) and computer vision in dermatological diagnostics has significantly evolved, driven by advancements in deep learning, image processing, and mobile technologies. Across global and local contexts, research has explored how these tools improve the accuracy and efficiency of detecting various skin conditions. International studies have developed real-time diagnostic systems, evaluated hybrid machine learning models, and addressed ethical concerns surrounding clinical AI use. In the Philippine setting, similar efforts have focused on localized disease datasets, mobile-based diagnostic tools, and the challenges of integrating AI into healthcare infrastructure. These developments reflect a growing commitment to leveraging technology in addressing the complexities of skin disease diagnosis and enhancing access to quality care.

Sorour, Hany, Elredeny, Sedik, and Hussien presented an automatic dermatology detection system using deep learning (DL) and computer vision techniques to address the complexity of visually diagnosing pigmented skin diseases [3]. Their work underscores the value of early detection and the necessity of integrating robust DL algorithms like YOLO-V5 to achieve real-time prediction and high accuracy in identifying various skin conditions. Their analysis emphasized how image-based analysis—considering color, shape, and lesion pattern—poses challenges when performed manually, justifying the use of automated systems to enhance both diagnostic accuracy and workflow efficiency. Among the YOLO-V5 models evaluated, Model 1 achieved a mean average precision (mAP@0.5) of 0.877,

Model 2 achieved 0.892, and Model 3 achieved the highest at 0.935. Melanoma detection precision-recall scores were 0.796, 0.803, and 0.893 for Models 1, 2, and 3 respectively, while vitiligo detection showed consistently high performance with scores of 0.958, 0.981, and 0.976.

Aquil, Saeed, Baowidan, Ali, and Elmitwally explored hybrid machine learning (ML) and DL models for early skin disease detection across diverse skin tones using the HAM10000 dataset [4]. Their study compared combinations such as SVM-MobileNetV2 and RF-DenseNet121, demonstrating the power of transfer learning and model optimization through platforms like Google Colab Pro. Their approach highlighted how blending classical ML and modern DL architectures can improve diagnostic results, particularly for underrepresented groups, reinforcing the significance of model diversity and adaptability. The accuracy achieved from the SVM classifier using features extracted from MobileNetV2 is 98.08%, while the proposed methodology achieved a high accuracy of 98.32% for skin lesion classification using the Random Forest classifier with DenseNet121 as the feature extractor.

Quantitative skin evaluation was also examined by Wu, Ran, Tanaka, and Lui, who developed a pilot system for assessing skin hydration using visible image processing [5]. Their study introduced the GVR index, employing contrast enhancement via Contrast Limited Adaptive Histogram Equalization (CLAHE) and segmentation to produce accurate moisture readings. This work contributes to the broader application of computer vision in dermatology by quantifying skin attributes that are often evaluated subjectively in clinical practice. The intensity (I) component was extracted after color model conversion, and CLAHE was applied to enhance contrast. Skin reflectance, particularly prominent on the cheek site, was attributed not only to hydration levels but also to facial skeletal structures. Evaluation showed that the GVR index had a statistically significant correlation with both the Corneometer (Pearson $r = 0.477$, $p = 0.046$) and Skicon (Pearson $r = 0.736$, $p = 0.000$), supporting the index's validity in estimating skin hydration.

In the Philippine setting, the advancement of artificial intelligence (AI) and computer vision technologies in healthcare has seen increasing scholarly interest. Local studies have explored their potential in addressing various health concerns, particularly in diagnostics, reflecting a broader movement toward data-driven solutions tailored to local clinical contexts.

Tomas, Ho, Mehrpouran, and Santos developed a deep learning model using YOLOv5 for detecting common skin diseases such as acne, leprosy, and psoriasis [6]. Their study demonstrated how object detection algorithms could support Filipino dermatologists by providing faster and more accurate diagnostic assistance. Despite strong performance, they emphasized the need to diversify datasets and fine-tune model parameters to enhance recognition of less common conditions like atopic dermatitis and keratosis pilaris.

Tusoy developed a mobile health (mHealth) application for screening neonatal jaundice [7]. Although it targeted a non-dermatological condition, the research demonstrated how image-based mobile systems could assist in preliminary diagnoses. The tool proved accurate in estimating bilirubin levels, supporting its use as a non-invasive, accessible screening instrument—an approach that holds promise for broader health applications. The findings revealed a positive moderate correlation between the mHealth application and the Kramer scale (0.625, 95% CI 0.46–0.75, $p < 0.01$), TcB (0.608, 95% CI 0.43–0.74, $p < 0.01$), and TSB (0.666, 95% CI 0.51–0.78, $p < 0.01$). At the TSB cutoff value of 200 μmol/L, the sensitivity, specificity, positive predictive value (PPV), and negative predictive value (NPV) of the mHealth application were 55.6%, 90.0%, 45.5%, and 93.1%, respectively. At a 250 μmol/L cutoff for TSB and the mHealth application, the sensitivity and specificity were 50.0% and 94.0%, respectively.

Lastly, Alibudbud, Aruta, Sison, and Guinto examined AI's role in mental healthcare amid climate change stressors in the Philippines [8]. Although the focus was on mental health, the paper addressed broader concerns such as algorithmic bias, accessibility, and ethical deployment of AI—factors that are also critical when implementing AI systems in any field of medicine, including dermatology.

3 METHODOLOGY

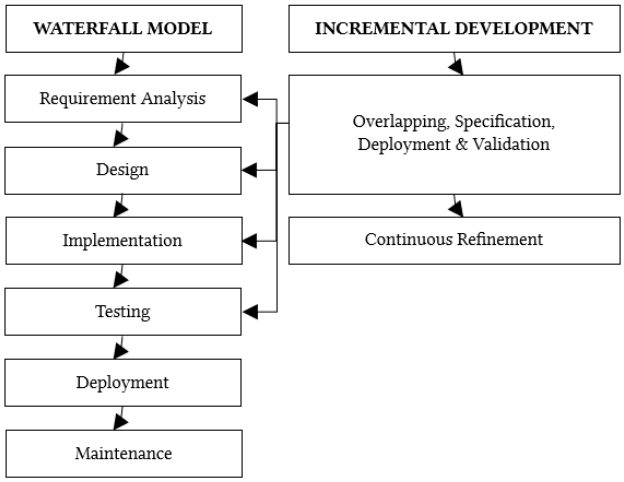


Figure 1: **Software Development Methodology Diagram**

The system utilized a hybrid methodology that combined the Waterfall model and incremental development to guide its design and implementation. The Waterfall model provided a linear progression through distinct phases, including requirements analysis, design, implementation, testing, deployment, and maintenance. This structured approach ensured that each stage was completed before advancing to the next, which allowed for thorough documentation and clear project milestones. Complementing this, incremental development allowed for overlapping specification, development, and validation, which enabled continuous refinement based on technical evaluations and created a dynamic yet organized workflow throughout the development of the system.

During the planning phase, a comprehensive literature review was conducted to define the project scope and objectives while focusing on ten sun-induced skin diseases that were prevalent and clinically relevant in the Philippines. By targeting these specific conditions, the project aimed to address significant

diagnostic challenges in dermatology and enhance early detection and intervention. This targeted approach ensured that the resulting tool would be highly applicable to the specific health needs of the local population.

The implementation phase involved developing the system using Python and integrating a range of machine learning algorithms and computer vision techniques, with EfficientNetB4 serving as the core model through transfer learning to enhance performance. The training process incorporated a softmax classifier, Adam optimizer, and categorical cross-entropy loss function. Computer vision techniques such as image augmentation, feature extraction, and global average pooling were employed to improve learning capabilities, while image processing methods like resizing, normalization, and array conversion ensured consistent input data quality. To maintain optimal performance, the Keras ModelCheckpoint function was employed to save the best-performing model based on validation accuracy.

Testing was carried out iteratively through unit and integration tests to verify system functionality, and technical feedback collected during this phase was used to fine-tune the system and meet diagnostic accuracy standards. Upon deployment, *DermaTector* was launched in a controlled environment that allowed users to upload images and featured a restricted feedback loop accessible solely by the development team to support the system's evolution. This mechanism allowed programmers to review predictions, log corrections, and retrain the model to incrementally improve accuracy over time. Evaluation metrics such as accuracy and loss scores were continuously monitored to assess the effectiveness of the model and guide further enhancements, ultimately delivering a robust diagnostic solution.

3.1 Conceptual Framework

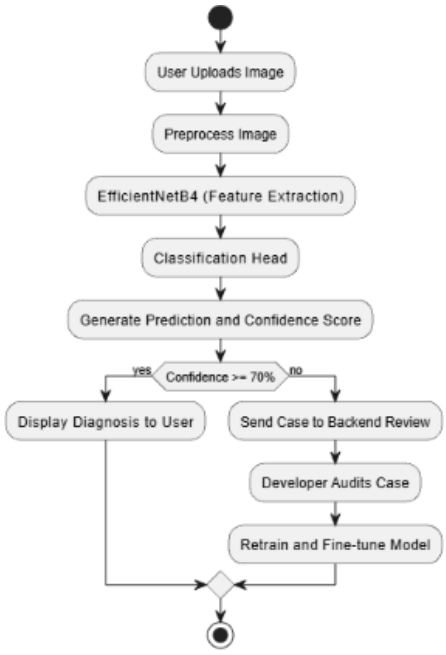


Figure 2: **Conceptual Framework**

The conceptual framework of *DermaTector* demonstrated a hybrid human-AI collaborative system designed to classify sun-induced skin diseases with continuous improvement through developer-managed retraining. The process began with the input layer, where users uploaded dermatological images that were preprocessed—resized to 600 × 600 pixels and normalized in BGR format—to match the expected input of the model. The preprocessed images passed through EfficientNetB4, a pre-trained convolutional neural network serving as a feature extractor with frozen layers. The classification head, composed of a GlobalAveragePooling2D layer followed by a dense layer with 256 ReLU-activated neurons and a softmax output layer, produced class predictions with corresponding confidence scores. If the model's confidence was equal to or exceeded 70%, the diagnosis was displayed to the user. Cases falling below this threshold, or those flagged for audit, were routed to a backend review system accessible only to the development team. Through this interface, programmers provided expert corrections which were logged in a CSV file containing the image path, predicted label, correct label, and model confidence. These verified feedback entries were then incorporated into the training data, enabling the system to retrain and fine-tune itself iteratively. This supervised active learning loop ensured that the model continuously evolved, improving accuracy and adaptability over time through technical intervention.

3.2 Use Case

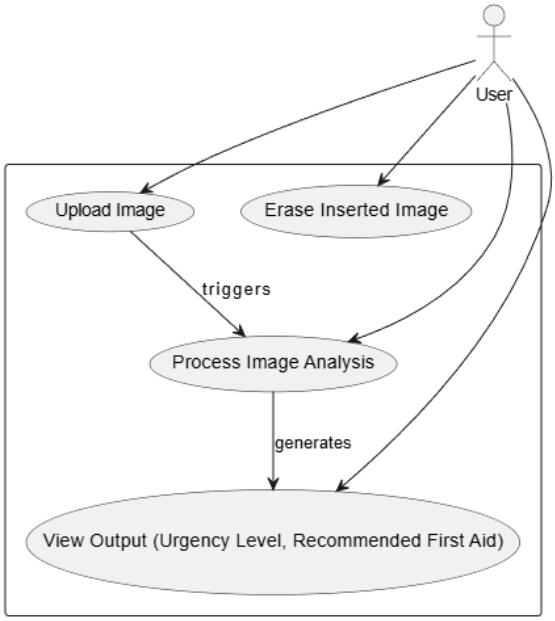


Figure 3: **Use Case Diagram**

In the final program's use by the general public, users were required to upload an image for analysis. Once the image was inserted, it was processed by clicking the "Process" button, which provided output based on the findings of the image analysis, including urgency and recommended first aid measures. Users also had the option to click the "Erase" button to remove any inserted image from the dropbox.

3.3 Technical Specifications

Supported Conditions:

Table 1: **Profile of the Target Sun-Induced Skin Diseases**

PRECANCEROUS LESIONS	
Actinic keratosis is characterized by rough, scaly skin patches that are caused by chronic exposure to ultraviolet (UV) radiation. If left untreated, these lesions may progress to a type of skin cancer known as *squamous cell carcinoma.*	***Actinic Keratosis***
SKIN CANCERS	
Basal cell carcinoma is a common and typically slow-growing form of skin cancer that is strongly associated with UV exposure.	***Basal Cell Carcinoma***
Considered a more aggressive type of skin cancer, ***squamous cell carcinoma*** may sometimes arise from pre-existing *actinic keratosis* lesions.	***Squamous Cell Carcinoma***
Melanoma is the most serious type of skin cancer, notable for its potential to spread (metastasize) to other organs in the body.	***Melanoma***
PHOTO-INDUCED DERMATOSES	
Often referred to as a sun allergy, **PMLE** is a condition that leads to the development of rashes on the skin following exposure to UV radiation.	***Polymorphous Light Eruption*** **(PMLE)**
Solar urticaria is an allergic reaction that manifests as hives on the skin upon sun exposure.	***Solar Urticaria***
Chronic actinic dermatitis is a persistent eczematous reaction of the skin that is triggered by exposure to sunlight.	***Chronic Actinic Dermatitis***
Lupus is an autoimmune disease affecting the skin that is known to worsen with UV exposure.	***Lupus***
PIGMENTARY DISORDERS	
A common condition among Filipinos, ***melasma*** is characterized by dark facial patches that appear due to UV sensitivity.	***Melasma***
Also known as "sun spots" or "age spots," ***solar lentigines*** are benign pigmented lesions on the skin that are caused by chronic UV exposure.	***Solar Lentigo***

Performance Targets:

- Accuracy: >80% on validation set
- Recall: >80% on validation set
- Precision: >80% validation set
- Inference Time: <5 seconds per image
- User Interface Responsiveness: <2 seconds for feedback
- System Availability: 24/7 via Colab runtime

3.4 Evaluation of the Project

The evaluation of *DermaTector* was conducted to determine its software characteristics as a web-based diagnostic tool for sun-induced skin diseases using the ISO/IEC 25010 standards as a framework. The assessment specifically focused on seven quality characteristics: Functional Suitability, Performance Efficiency, Compatibility, Interaction Capability, Reliability, Security, and Trust, Privacy and User Confidence. To facilitate this, a group of 50 participants from the general public was selected through purposive sampling to perform key tasks on the website, such as uploading images and reviewing classification results.

Since *DermaTector* is a web-based system, the evaluation relied on stable internet connectivity for image uploads and real-time classification. Google Forms was used for data collection due to its accessibility and automated data consolidation features. The system was tested across different browsers to ensure compatibility with standard web technologies. Privacy and security protocols were implemented to protect user data throughout the online evaluation process.

4 RESULTS AND DISCUSSION

This section provides a detailed presentation and interpretation of the empirical results obtained from the *DermaTector* system. This section is divided into two main parts: the Model Evaluation, which quantitatively assesses the diagnostic reliability of the EfficientNetB4 model using performance metrics like accuracy and F1-score, and the User Evaluation, which analyzes the feedback collected from 50 members of the general public regarding the system's usability, accessibility, and quality, specifically against the ISO/IEC 25010 standards. The findings discussed here validate the core performance of the system and its effectiveness as an awareness tool for sun-induced skin diseases in a real-world setting.

4.1 Model Evaluation

Table 2: **Validation Performance Metrics per Epoch**

Epoch	Accuracy	Loss	Precision	Recall
1	0.7955	2.8724	0.8214	0.7841
10	0.7557	2.9687	0.7754	0.7358
20	0.7841	2.8294	0.8024	0.7727
21	0.8011	2.6899	0.8274	0.7898
30	0.8011	2.6569	0.8166	0.7841
40	0.8011	2.6953	0.8118	0.7841
43	0.8068	2.6756	0.8201	0.7898
46	**0.8097**	**2.6660**	**0.8190**	**0.7841**
50	0.8011	2.6768	0.8147	0.7869
56	0.8011	2.6860	0.8088	0.7812

The model training utilized an 80/20 data split for training and validation/testing. The EfficientNetB4 model was trained over 56 epochs using transfer learning, with the learning rate dynamically reduced during the training process. The training process showed volatility in the training metrics, while the validation metrics maintained a generally stable trend, indicating controlled generalization.

The training accuracy curve exhibited significant fluctuation, starting near 0.50 and frequently spiking to near 1.00, which suggests that the model sometimes rapidly overfits the training batches. In contrast, the validation accuracy curve showed more stability, fluctuating around 0.80 throughout the training process. The model achieved incremental improvements in validation accuracy at several points, reaching 0.8011 at Epoch 21 and slightly higher at Epoch 46, reaching a peak of 0.8097. Similarly, the training loss showed large spikes, reaching a loss of more than 9.0, while the validation loss remained relatively stable around 2.7. This difference in behavior is a common indicator of overfitting in the training set but suggests that the model successfully learned features that generalize to the independent validation set. The best model was ultimately saved at Epoch 46, based on validation accuracy.

The training curves for precision and recall mirrored the volatile behavior of the training accuracy curve, showing frequent drops and spikes, fluctuating roughly between 0.60 and 1.00. The corresponding validation precision and validation recall curves maintained relative stability, hovering around 0.77 to 0.83. For instance, at Epoch 43, the validation precision was 0.8201 and the validation recall was 0.7898. This stability in the validation metrics, despite the high variance during the training phase, confirms that the model consistently learned generalized features necessary for reliable skin disease classification in the testing environment.

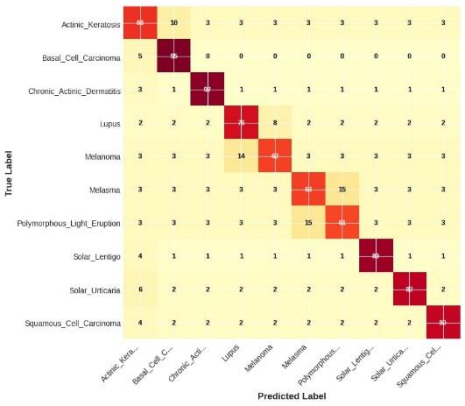


Figure 4: **Confusion Matrix**

To examine the diagnostic reliability of the *DermaTector* model across the ten sun-induced skin conditions, a confusion matrix was generated and analyzed. This matrix provides a granular view of the model's classification accuracy and identifies specific inter-class overlaps that occur during the testing phase.

The model demonstrated exceptional accuracy in identifying *Basal Cell Carcinoma* (BCC), *Chronic Actinic Dermatitis* (CAD), and *Solar Lentigo*, with BCC achieving the highest correct classification rate of 95.0%. The strong performance in these categories suggests that EfficientNetB4 successfully isolated the distinct morphological features—such as the specific pigmentation of *Solar Lentigo* and the unique lesion structures of BCC—allowing for high precision.

A significant point of confusion was observed between *Melasma* and *Polymorphous Light Eruption* (PMLE), with 15 instances of *Melasma* misclassified as PMLE and vice versa. This indicates a high degree of visual similarity between these conditions under the model's feature extraction process, likely due to overlapping presentation of skin discoloration and inflammatory textures.

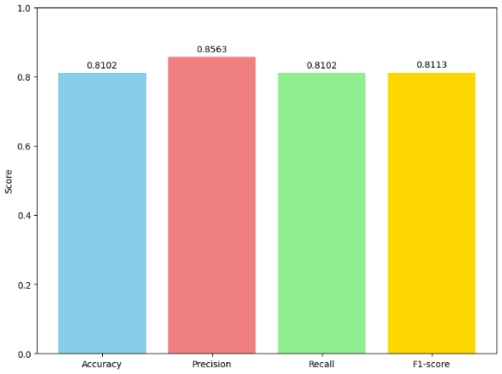


Figure 5: **Comprehensive Performance Metrics**

The final evaluation of the model, based on the weights from the best-performing epoch (Epoch 46), demonstrated strong overall performance in classifying the ten sun-induced skin diseases.

The final accuracy of the model was 0.8102. This confirms that the model correctly classified 81.02% of the skin disease images in the testing set. The relatively high accuracy, exceeding the 80% target, supports the viability of using EfficientNetB4 for this task. The precision metric was 0.8563, meaning that when the model predicted a certain disease, it was correct 85.63% of the time. The model's recall was 0.8102, indicating that the model successfully identified 81.02% of all actual disease cases in the dataset. Finally, the F1-score, which is the harmonic mean of precision and recall, was 0.8113.

4.2 User Evaluation

The system was found to be effective in providing accessible educational support while guiding users toward professional follow-up. The user evaluation confirmed the platform's utility and ethical design:

- High Usability – The system attained consistently High to Very High ratings across all assessed ISO/IEC 25010 standards, with interaction capability receiving the highest composite mean ($\bar{x}$ = 4.74, Very High). This confirms the system is easy to navigate and accessible for the general public.
- User Confidence and Ethics – Users expressed Very High confidence in the system's ability to encourage responsible use and professional follow-up ($\bar{x}$ = 4.70, Very High). The inclusion of disclaimers was highly rated, with users affirming it increased their confidence in the tool ($\bar{x}$ = 4.46, High). This demonstrates the system successfully positioned itself as a supportive aid, not a diagnostic replacement.

The central hypothesis—that the system, *DermaTector*, can accurately classify selected sun-induced skin diseases using EfficientNet, achieving high performance in terms of accuracy, precision, recall, and F1-score—is strongly supported by the evidence.

5 CONCLUSION

DermaTector was successfully developed as a web-based diagnostic support tool for the image-based classification of sun-induced skin conditions. Evaluation against ISO/IEC 25010 standards revealed High to Very High user satisfaction across all categories, with interaction capability receiving the highest composite mean of 4.74. Reliability analysis produced strong Cronbach's alpha values between 0.710 and 0.927, which confirmed the integrity of the evaluation and the system's effectiveness in promoting early dermatological awareness. Despite these positive results, the study identified critical areas for refinement, including classification inaccuracies involving varied skin tones and early-stage symptoms. Future work will prioritize expanding the image dataset to include more diverse skin tones, lighting conditions, and balanced representations of disease severity. Additionally, subsequent research will incorporate rigorous baseline comparisons against alternative architectures and existing dermatological systems to better establish the model's technical contribution. Ultimately, broader demographic testing and sustained collaboration with dermatologists for dataset validation will be essential to ensure the system remains ethically compliant and inclusive for underserved populations.

REFERENCES

[1] Belo Med. (2020, December 15). How high-risk are Filipinos to skin cancer? https://belomed.com/the-base/how-high-risk-are-filipinos-to-skin-cancer
[2] Philippine Dermatological Society. (2023, August 13). All about hyperpigmentation – let's be enlightened! https://pds.org.ph/all-about-hyperpigmentation-lets-be-enlightened/
[3] Sorour, S. E., Abo Hany, A., Elredeny, M. S., Sedik, A., & Hussien, R. M. (2023). An automatic dermatology detection system based on deep learning and computer vision. *IEEE Access*, 11, 137769–137778. https://doi.org/10.1109/ACCESS.2023.3340735
[4] Aquil, A., Saeed, F., Baowidan, S., Ali, A. M., & Elmitwally, N. S. (2025). Early detection of skin diseases across diverse skin tones using hybrid machine learning and deep learning models. *Information (2078-2489), 16*(2), 152. https://doi.org/10.3390/info16020152
[5] Wu, Y., Ran, J., Tanaka, T., & Liu, Q. (2024). Quantitative skin surface hydration measurement by visible optical image processing: A pilot study. *Skin Research & Technology, 30*(6), 1–7. https://doi.org/10.1111/srt.13773
[6] Tomas, M. C. A., Ho, D. M. P., Mehrpouran, S. P., & Santos, M. M. P. J. (2024, October). Recognizing common skin diseases in the Philippines using YOLOv5 deep learning. *In Sixteenth International Conference on Digital Image Processing (ICDIP 2024)* (Vol. 13274, pp. 59–69). SPIE. https://doi.org/10.1117/12.3037975
[7] Tusoy, J. (2022). *A mHealth application as a screening tool for neonatal jaundice in Filipino neonates* (Master's thesis, NTNU). https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/3006144
[8] Alibudbud, R. C., Aruta, J. J. B. R., Sison, K. A., & Guinto, R. R. (2025). Artificial intelligence in the era of planetary health: Insights on its application for the climate change–mental health nexus in the Philippines. *International Review of Psychiatry, 37*(1), 21–32. https://doi.org/10.1080/09540261.2024.2363373

Augmenting a Force Directed Layout Algorithm in Drawing Adjacent Transposition Graphs*

Harold C. Antonio*
Department of Computer Science
University of the Philippines – Diliman
Quezon City, Philippines
hcantonio1@up.edu.ph

Francis M. Albarracin
Department of Computer Science
University of the Philippines – Diliman
Quezon City, Philippines
fmalbarracin@up.edu.ph

Ivy D. Ordanel
Department of Computer Science
University of the Philippines – Diliman
Quezon City, Philippines
idordanel@up.edu.ph

Abstract

Adjacent transposition graphs (ATGs) are useful in poset mining and related applications. Force-directed layout algorithms (FDA) are one of the most commonly used algorithms to draw graphs. However, when FDA is applied to drawing ATGs, some inherent structural properties of ATGs such as cycles and parallel edges are distorted. In this study, we derived some properties of ATGs and used them to create methods to augment FDA in drawing more readable ATGs that adhere to their structural properties. The results established that the classes of posets with square and hexagonal ATGs are equivalent to certain classes of leveled posets. Furthermore, the number of square and hexagonal ATGs that can possibly include an arbitrary edge is identified. This information can be used to draw undistorted ATGs by completing partial cycles. Finally, four drawing algorithms were created that varied the vertices included in the FDA computation.

CCS Concepts

• **Theory of computation → Graph algorithms analysis**.

Keywords

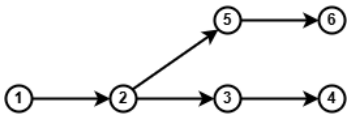
poset mining, adjacent transposition graph, graph layout algorithm

1 Introduction

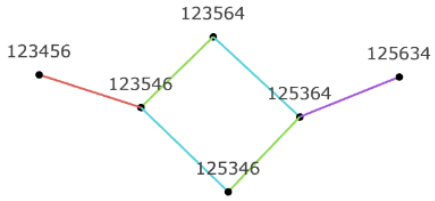
The process known as *poset mining* [4] extracts ordering constraints from a collection of sequential information. In data mining, there are a number of fields with a similar endeavor. For example, in paleontology, sequential data on the taxa occurring at each site are used to build the evolutionary order of fossil sites [10]. Another example is in business, where sequential data from workflow logs may be analyzed to build process models [1].

The mathematical formulation of poset mining is the poset cover problem and it deals with posets and linear orders. A (strict) poset is defined by a set V and a binary relation $<_P$ on V that is irreflexive, antisymmetric, and transitive. Since $<_P$ is transitive, a poset $P = (V, <_P)$ naturally corresponds to a transitively closed directed acyclic graph $G = (V, <_P)$. If we get the transitive reduction of $G = (V, <_P)$, the result is a Hasse diagram $H(P)$, which is a simpler and more common representation of a poset.

A linear order $L = (V, <_L)$ is a poset where all elements in V are *comparable*. Two elements u and v are comparable if either $(u, v) \in <_P$ or $(v, u) \in <_P$. A linear order can be written in permutation notation [4], as in $L = (3, 1, 2, 4)$, or simply by concatenating its elements as in '3124'. For an arbitrary poset $P = (V, <_P)$, if $<_P \subseteq <_L$, then the linear order L is a *linear extension* of P. We denote $\mathcal{L}(P)$ as the set of linear extensions *generated* by P.



(a) The Hasse diagram of the poset $P_1 = (\{1, 2, 3, 4, 5, 6\}, \{(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 4), (2, 5), (2, 6), (3, 4), (5, 6)\})$.



(b) Linear extension graph of P_1

Figure 1: Example poset and adjacent transposition graph

With posets and linear orders defined, the formal definition of the poset cover problem can be discussed. Given set Υ of m linear orders of length n, the problem finds a set of posets $\mathcal{P}$ such that $\bigcup_{P \in \mathcal{P}} \mathcal{L}(P) = \Upsilon$ and $|\mathcal{P}|$ is minimum.

One way of approaching the problem is to analyze the graph induced by Υ called the *adjacent transposition graph* (ATG). The ATG of Υ, denoted $\mathcal{G}(\Upsilon)$, is a graph $G(\Upsilon, E)$, where the vertex set Υ is composed of linear orders and the edge set E consists of pairs of linear orders $\{L_i = (V, <_{L_i}), L_j = (V, <_{L_j})\}$ that differ by a single adjacent swap, that is, there exists $u, v \in V$ such that $<_{L_i} \setminus <_{L_j} = \{(u, v)\}$ and $<_{L_j} \setminus <_{L_i} = \{(v, u)\}$. Since a poset generates a set of linear extensions $\mathcal{L}(P)$, posets also induce ATGs which are called *linear extension graphs* (LEGs). Thus, one way to solve the poset cover problem is to find the smallest set of LEGs that cover the ATG of the input $\mathcal{G}(\Upsilon)$.

Thus, visualizing ATGs is a common task in studying the poset cover problem. However, visualizing ATGs is not a trivial task as ATGs have a certain structure. Although the properties of ATGs do not have extensive discussion in the literature, the properties of LEGs are well-explored. And due to the supercover algorithm in [4], any ATG is a subgraph of the LEG of some poset which means that properties of LEGs may be applied to ATGs as well. Among the properties of ATGs, we have identified four that have intuitive sense in making the visualization:

*Corresponding author

(1) Edge Connection. This is based on the definition of ATGs.
(2) Colored Edges. The edges of an LEG can be partitioned into equivalence classes due to the Djoković-Winkler relation θ [3, 7, 12]. Massow (2010), in particular, referred to these classes as *color classes*. The pair of elements swapped on every edge of a class distinguishes the class from the others.
(3) Parallel Edges. A theorem states that if a cycle in an LEG is *isometric*, then opposite edges in the cycle belong to the same color class [7, 8]. Although LEGs are abstract structures, the theorem implies that the edges of a color class must be parallel when drawn in dimensional spaces.
(4) Prominence of 4- and 6-cycles. Another theorem states that the cycle space of an LEG can be generated by 4- and 6-cycles[7, 8].

The first two properties (edge connection and color) can be implemented in a straightforward manner. However, making classes of edges parallel and cycles prominent poses a challenge in visualization. Force-directed layout algorithms (FDA) can partially address these issues. Force-directed layout algorithms draw graphs based on the inherent structure of the graph. One approach assigns spring forces to the nodes, making them repel each other. Such methods tend to produce aesthetically pleasing graphs with symmetry and reduced crossings [6].

However, applying an FDA to an ATG can distort cycles and parallel edges. Incomplete cycles in the ATG may be straightened by spring forces. Consider the ATG of $\Upsilon = \{1234, 1324, 1342, 1432, 1423\}$ in Figure 2. The ATG is isomorphic to a hexagon missing a single vertex. Thus, applying FDA to the graph would make the vertices repel each other and stabilize into a straight line.

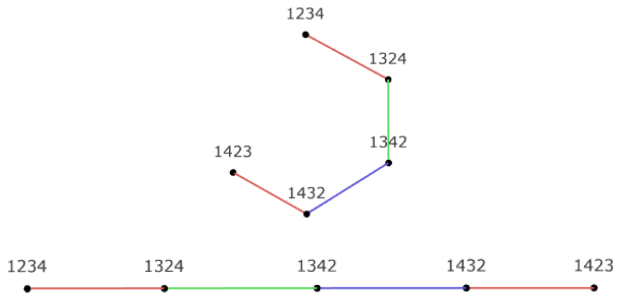


Figure 2: An ATG that is an almost complete cycle (top). Expected layout after applying an FDA (bottom).

Visualizing ATGs with respect to their properties finds value in poset mining, particularly in manually solving instances of the poset cover problem. However, there are no layout algorithms that can do this. In this paper, we devise methods to augment an FDA in drawing ATGs such that the parallel edges, 4-cycles and 6-cyles become prominent.

2 Drawing Methods

In this study, we used the FDA called the Fruchterman-Reingold algorithm as implemented in the python library Networkx [5]. We will refer to this algorithm as FruchtermanReingold(G) where G is the input graph $G = (V, E)$. This algorithm has a time complexity of $O(|V|^2 + |E|)$ [6].

We augment an FDA by preprocessing the input before feeding it to the FDA. This is done by including additional vertices that may "anchor" the graph structure in the desired way. After this step, for every drawing algorithm, we encode the vertices and edges of the ATG by assigning $\mathcal{G}(\Upsilon)$ to G which takes $O(|V|^2)$ time. We consider an algorithm done when the layout algorithm FruchtermanReingold has been applied to G. This step will take $O(|V|^2 + |E|) = O(|V|^2)$ as the maximum number of edges in an ATG is in the order $O(|V|^2)$. Thus, excluding the preprocessing steps, the algorithms run in at least $O(|V|^2)$.

Since the tools allow for 3D layouts, we adopted it over 2D layouts for a convenient and interactive visual inspection.

2.1 Drawing Algorithm "Unaugmented"

In this drawing method, we do not add additional vertices to the input and only perform the encoding and application of the FDA to the graph. Hence, this algorithm will run in $O(m^2)$.

As discussed previously, using the FDA itself may distort parallel edges and cycles. To address this issue, the input to the FDA can be modified by adding more vertices.

2.2 Drawing Algorithm "Permutahedron"

One way of completing the incomplete cycles is to include all possible linear orders. In this algorithm, the linear extensions of the antichain poset $P = (V, \{1, 2, \ldots, n\}, \varnothing)$, denoted $\mathcal{A}_n$, are set as input to the FDA. The LEG induced by an antichain poset is isomorphic to a permutahedron polytope. The preprocessing step of this algorithm takes $O(n!)$. As the input graph now contains $|V| = n!$ vertices, the rest of the algorithm will take $O(n!^2)$, which is also the running time of the entirety of Algorithm 1.

Algorithm 1 Permutahedron Drawing Algorithm

Input: A set Υ of m linear orders of length n
1: $\Upsilon' \leftarrow \mathcal{L}(\mathcal{A}_n)$
2: $G \leftarrow \mathcal{G}(\Upsilon')$
3: FruchtermanReingold(G)

In addition to the inefficient runtime, prominent cycles and parallel edge classes are generally not observable for large m and n. An advantage of this algorithm may be observed when the ATG is composed of multiple disconnected components as the components will be positioned relative to the permutahedron.

2.3 Drawing Algorithm "Supercover"

A better approach would be to take only a minimum supergraph with complete cycles that contains the input. This can be achieved with the supercover algorithm in [4]. Almost-complete cycles in the input will become complete in the ATG of the supercover.

The supercover subroutine runs in $O(mn^2)$ [4]. In the subroutine, if the partial order $<_P = \varnothing$, then $\mathcal{L}(P) = \mathcal{L}(\mathcal{A}_n)$. This means that in the worst case, Algorithm 2 will also run in $O(n!^2)$ time.

Also, portions of some cycles remain distorted. If the supercover is a planar graph, then the FDA will lay the ATG flat. This means

Algorithm 2 Supercover Drawing Algorithm

Input: A set Υ of m linear orders of length n

1: $\Upsilon' \leftarrow Supercover(\Upsilon)$
2: $G \leftarrow \mathcal{G}(\Upsilon')$
3: FruchtermanReingold(G)

Algorithm 3 Supercover Subroutine [4]

Input: A set Υ of m linear orders of length n

1: $<_P \leftarrow \bigcap_{L \in \Upsilon} <_L$
2: **return** $\mathcal{L}(P)$

that edges belonging to different color classes may also become parallel.

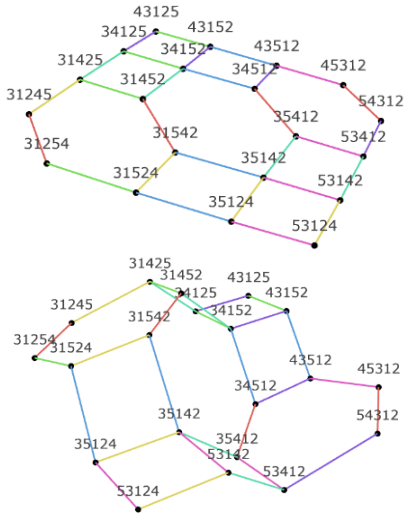


Figure 3: A "Supercover" graph layout where there are parallel edges with different colors (top). A better layout where portions of cycles have more prominence (bottom).

2.4 Cycles and Leveled Posets

Massow (2010) and Naatz (2000) showed that the cycle space of LEGs can be generated by square and hexagonal cycles [7, 8]. This presents an opportunity to modify the input Υ by identifying and including missing parts of incomplete cycles. This led to the formalization of some observations about the cycles in ATGs. In particular, we found that convex cycles of sizes 4 and 6 correspond to particular leveled posets. We say that a subgraph is *convex* if it contains all possible shortest paths, with respect to the original graph, between any of its vertices. Moreover, a convex cycle is a convex subgraph that is a cycle. For the rest of this paper, convex 4- and 6-cycles will now be referred to as square and hexagonal cycles for smoother reading.

2.4.1 Existing Theorems about ATG Cycles. When investigating cycles, it is useful to note observations on consecutive and disconnected edges. The following lemma enumerates the possibilities for two distinct edges. This lemma is true by direct enumeration: two distinct edge labels either have a single common element or none at all. In the following lemma, we will adopt and extend the edge label function ψ which is defined $\psi(L_i, L_j) = \{u, v\}$ if $<_{L_i} \Delta <_{L_j} = \{(u, v), (v, u)\}$, where Δ is the symmetric difference [9]. For some color class θ_i, we will also write $\psi(\theta_i) = \psi_i$ to denote the edge label associated with θ_i.

Lemma 2.1. *Given two edge labels ψ_1 and ψ_2, exactly one of the following is true:*

(1) $\psi_1 \cap \psi_2 = \varnothing$
(2) $\psi_1 \cap \psi_2 = \{v\}$ for some $v \in V$

A theorem by Massow (2010) and Naatz (2000) that relates cycles and edge labels is also required for the proofs in this paper. It states that opposite edges in *isometric* cycles have the same edge labels. A subgraph is described as isometric if it contains at least one shortest path, with respect to the original graph, between any of its vertices. Thus, convex cycles are also isometric, and the forward direction of this theorem applies to them, i.e, given a convex cycle C, opposite edges of C have the same edge label.

Theorem 2.2. *[7, 8] Let G be a linear extension graph and let C be a cycle in G. Then C is an isometric cycle if and only if for any two edges e, f on C, the following two conditions are equivalent:*

(a) e and f are opposite on C.
(b) $\psi(c) = \psi(f)$

Finally, for a fixed linear order length n, it is known that there is a bijection between the convex subgraphs of the permutahedron and the LEGs [2, 7, 9, 11]. This allows us to refer to posets using convex subgraphs of the permutahedron.

2.4.2 Leveled Posets. A poset is a *leveled poset* if its elements can be partitioned into levels $V_1, V_2, ..., V_k$ such that for any $u \in V_i$ and $v \in V_j$, then $u <_P v$ if and only if $i < j$. The ordered sets uniquely identify a leveled poset. Thus, we will adopt the notation used by Fernandez et. al. (2013) that writes a leveled poset P as a function of an ordered set of sets: $P = \text{leveled}(V_1, V_2, ..., V_k)$ [4].

2.4.3 Theoretical Results. Now we can begin to prove theorems that relate cycles and leveled posets. First, we define the relevant sets. Let $\text{LEG}_{\text{SQUARE}}$ be the class of posets whose ATGs are isomorphic to a square. Next, let Leveled$(2, 2)$ be the set of leveled posets with singleton sets as levels except for two levels whose cardinalities are both 2.

The following theorem shows that if a subgraph of some ATG is a convex square cycle, then it is the LEG of a Leveled(2,2) poset. Conversely, Leveled(2,2) posets are characterized by their square LEGs.

Theorem 2.3. *For a fixed length of linear orders n, the sets* Leveled$(2, 2)$ *and* $\text{LEG}_{\text{SQUARE}}$ *are equal.*

Proof. Both halves of the proof are constructive. First show that Leveled$(2, 2) \subseteq \text{LEG}_{\text{SQUARE}}$. Let $P = \text{leveled}(\{v_1\}, \{v_2\}, \ldots, \{v_i, v_{i+1}\}, \ldots, \{v_j, v_{j+1}\}, \ldots, \{v_n\})$ where $V = \{v_1, v_2, \ldots, v_n\}$ and V contains at least four elements. Thus, $P \in$ Leveled$(2, 2)$. Also, the set of linear extensions $\mathcal{L}(P)$ contains exactly four linear orders

L_1, L_2, L_3, and L_4. Without loss of generality, let

$$L_1 = (v_1, v_2, \ldots, v_i, v_{i+1}, \ldots, v_j, v_{j+1}, \ldots, v_n),$$
$$L_2 = (v_1, v_2, \ldots, v_i, v_{i+1}, \ldots, v_{j+1}, v_j, \ldots, v_n),$$
$$L_3 = (v_1, v_2, \ldots, v_{i+1}, v_i, \ldots, v_{j+1}, v_j, \ldots, v_n),$$
$$L_4 = (v_1, v_2, \ldots, v_{i+1}, v_i, \ldots, v_j, v_{j+1}, \ldots, v_n)$$

This is true because a linear order $L \in \mathcal{L}(P)$ will admit some permutation of the two pairs of incomparable pairs $\{v_i, v_{i+1}\}$ and $\{v_j, v_{j+1}\}$ resulting in 2!2! = 4 linear orders. Then the edges of $\mathcal{G}(\mathcal{L}(P))$ is exactly $E = \{\{L_1, L_2\}, \{L_2, L_3\}, \{L_3, L_4\}, \{L_4, L_1\}\}$. Thus, $\mathcal{G}(\mathcal{L}(P))$ forms a square cycle and so $P \in \text{LEG}_{\text{SQUARE}}$.

Next, we want to show that $\text{LEG}_{\text{SQUARE}} \subseteq \text{Leveled}(2, 2)$. Let $P = (V, <_P) \in \text{LEG}_{\text{SQUARE}}$. By definition, $\mathcal{L}(P) = \{L_1, L_2, L_3, L_4\}$ for some arbitrary linear orders, and the ATG of P is convex and is a square cycle. Without loss of generality, we assume that the cycle connects the linear orders with respect to their order in the list. According to Theorem 2.2, the opposite edges of a convex cycle belong to the same color class. Let the two color classes be θ_1 and θ_2, and the corresponding edge labels ψ_1 and ψ_2. Also let $\psi(L_1, L_2) = \psi(L_3, L_4) = \psi_1$ and $\psi(L_2, L_3) = \psi(L_1, L_4) = \psi_2$.

Lemma 2.1 states that either $\psi_1 \cap \psi_2 = \varnothing$ or $\psi_1 \cap \psi_2 = \{v\}$, where v is the common element. We want to show that $\psi_1 \cap \psi_2 = \varnothing$ is the true statement and construct P with ψ_1 and ψ_2 as levels.

Suppose $\psi_1 \cap \psi_2 = \{v\}$ for some $v \in V$. Let $\psi_1 = \{u, v\}$ and $\psi_2 = \{v, w\}$ for some $u, v, w \in V$. Since $\psi(L_1, L_2) = \psi_1$ and $\psi(L_2, L_3) = \psi_2$, then u and v are not adjacent in L_3. This means that $\psi(L_3, L_4) \neq \psi_1$, which is a contradiction.

Thus, $\psi_1 \cap \psi_2 = \varnothing$. Let $\psi_1 = \{u, v\}$ and $\psi_2 = \{w, x\}$ and set $P = \text{leveled}(\{v_1\}, \{v_2\}, \ldots, \{u, v\}, \ldots, \{w, x\}, \ldots, \{v_n\})$ for some u, $v, w, x \in V$. This shows that $P \in \text{Leveled}(2, 2)$. □

Now that it is established that only posets in Leveled(2, 2) have convex square ATGs, we can inquire about the number of square cycles in an arbitrary ATG. The following corollary shows that an arbitrary edge in the *complete* ATG is contained by at most $n - 3$ square cycles. We say that an ATG is complete if it contains all possible linear orders of a specified length n. This implies that completing square cycles in an arbitrary ATG may take $n - 3$ steps multiplied by the number of edges. This means that generating all square cycles in a random ATG will take $O(m^2) \cdot O(n-3) = O(m^2 n)$ time.

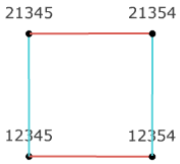


Figure 4: The ATG of leveled({1,2},{3},{4,5}) an illustration of Theorem 2.3.

Corollary 2.4. *Every edge in a complete ATG is contained in the ATG of $n - 3$ or $n - 4$ posets in* $\text{LEG}_{\text{SQUARE}}$.

Proof. This is a proof by counting. Let $L_1 = (v_1, v_2, \ldots, v_i, v_{i+1}, \ldots, v_n)$ be a linear order over $V = \{v_1, v_2, \ldots, v_n\}$. Let L_2 be the same linear order except that v_i and v_{i+1} are swapped. Thus, $\{L_1, L_2\}$ is an edge in the complete ATG with the edge label $\psi(L_1, L_2) = \{v_i, v_{i+1}\}$.

According to Theorem 2.3, the only convex square cycles that contain the $\{L_1, L_2\}$ edge are described by Leveled(2, 2). The strategy is to generate and count the relevant leveled posets. To generate the posets, fix $\{v_i, v_{i+1}\}$ as a level and choose some other pair of adjacent elements as the other level. For the explanations below, it is helpful to write L_1 with the swapped elements underlined.

There are two cases to consider. The first case occurs when v_i and v_{i+1} occur in the middle of the linear order. This case can be further divided into two subcases. The first subcase occurs when $i = 2$ (or its reverse, $i + 1 = n - 2$). In this case, there are $n - 4$ indices to choose from to make the second level. The second subcase occurs when $i \neq 2$ (and $i + 1 \neq n - 2$). In this case, the linear order can then be divided into two halves: the portion to the left of v_i and the portion to the right, both of which are valid portions to make the second level. For the left side, there are $i - 2$ indices to choose from. For the right side, there are $(n - 1) - (i + 1)$ indices to choose from. In total, there are $n - 4$ indices to choose from, the same as the first subcase. Therefore, in the first case, regardless of the subcases, there are $n - 4$ leveled posets that contain the edge $\{L_1, L_2\}$ in their ATGs.

The second case occurs when either v_i is the first element or v_{i+1} is the last element of the linear order. Therefore, the other level can be formed by choosing two adjacent elements from the $n - 2$ elements. There are $n - 3$ ways to do this. The case where v_{i+1} occurs in the last position can also be solved by counting from the right. It can be observed that the first case has one less poset due to the fixed level being in the middle, invalidating $\{v_{i-1}, v_{i+3}\}$ as an option. □

Next, we define the hexagonal analogs of the sets. Let LEG_{HEX} be the class of posets whose ATGs are isomorphic to a hexagon. Then, let Leveled(3) be the set of leveled posets with singleton sets as levels except for one whose cardinality is exactly 3.

The following theorem shows that if a subgraph of some ATG is a convex hexagonal cycle, then it is the LEG of a Leveled(3) poset. In other words, Leveled(3) posets are characterized by their hexagonal LEGs.

Theorem 2.5. *For a fixed length of linear orders n, the sets* Leveled(3) *and* LEG_{HEX} *are equal.*

Proof. Both halves of the proof are constructive. First show that $\text{Leveled}(3) \subseteq \text{LEG}_{\text{HEX}}$. Let $P = (V, <_P) = \text{leveled}(\{v_1\}, \{v_2\}, \ldots, \{v_i, v_j, v_k\}, \ldots, \{v_n\})$ where $V = \{v_1, v_2, \ldots, v_n\}$ and V contains at least three elements. Thus, $P \in \text{Leveled}(3)$. Also, the set of linear extensions $\mathcal{L}(P)$ contains exactly six linear orders L_1, L_2, L_3, L_4, L_5, and L_6. Without loss of generality, let

$$L_1 = (v_1, v_2, \ldots, v_h, v_i, v_j, v_k, v_l \ldots, v_n),$$
$$L_2 = (v_1, v_2, \ldots, v_h, v_i, v_k, v_j, v_l \ldots, v_n),$$
$$L_3 = (v_1, v_2, \ldots, v_h, v_k, v_i, v_j, v_l \ldots, v_n),$$
$$L_4 = (v_1, v_2, \ldots, v_h, v_k, v_j, v_i, v_l \ldots, v_n)$$
$$L_5 = (v_1, v_2, \ldots, v_h, v_j, v_k, v_i, v_l \ldots, v_n),$$
$$L_6 = (v_1, v_2, \ldots, v_h, v_j, v_i, v_k, v_l \ldots, v_n),$$

This is true because a linear order $L \in \mathcal{L}(P)$ will admit some permutation of v_i, v_j, and v_k resulting in 3! = 6 linear orders. Then the edges of $\mathcal{G}(\mathcal{L}(P))$ is exactly $E = \{\{L_1, L_2\}, \{L_2, L_3\}, \{L_3, L_4\}, \{L_4, L_5\}, \{L_5, L_6\}, \{L_6, L_1\}\}$. Thus, $\mathcal{G}(\mathcal{L}(P))$ forms a hexagonal cycle and so $P \in \text{LEG}_{\text{HEX}}$.

Next, we want to show that $\text{LEG}_{\text{HEX}} \subseteq \text{Leveled}(3)$. Let $P \in \text{LEG}_{\text{HEX}}$. By definition, $\mathcal{L}(P) = \{L_1, L_2, L_3, L_4, L_5, L_6\}$ for some arbitrary linear orders and the ATG of P is convex and is a hexagonal cycle. Without loss of generality, we assume that the cycle connects the linear orders with respect to their order in the list. According to Theorem 2.2, the opposite edges of a convex cycle belong to the same color class. Let the color classes be θ_1, θ_2, and θ_3, and the corresponding edge labels ψ_1, ψ_2, and ψ_3. Also, let $\psi(L_1, L_2) = \psi(L_4, L_5) = \psi_1$, $\psi(L_2, L_3) = \psi(L_5, L_6) = \psi_2$ and $\psi(L_3, L_4) = \psi(L_1, L_6) = \psi_3$.

We want to show that $|\psi_1 \cup \psi_2 \cup \psi_3| = 3$ and that this union can be used as a level to construct P. This can be done by showing that ψ_1, ψ_2, and ψ_3 pairwise share a common element, that is, $\psi_i \cap \psi_j = \{v_{\psi_i \cap \psi_j}\}$ for some $v_{\psi_i \cap \psi_j} \in V$.

Suppose $\psi_1 \cap \psi_2 = \varnothing$. Since $\psi(L_1, L_2) = \psi_1$ and $\psi(L_2, L_3) = \psi_2$, then there exist some L_2' such that $\psi(L_1, L_2') = \psi_2$ and $\psi(L_2', L_3) = \psi_1$. This shows that $\mathcal{G}(\mathcal{L}(P))$ is not convex, which is a violation of our assumption. Thus, ψ_1 and ψ_2 must have a common element. Without loss of generality, we may assume $v_{i+1} = v_{i+2}$.

Applying the same reasoning on the L_2-L_3-L_4 path, ψ_2 and ψ_3 must also have a common element. This time, the common element cannot be chosen arbitrarily. We will first show that assuming $v_{i+1} = v_{i+4}$ will result in a contradiction. By traversing the edges labels, adjacent swaps can be applied on a linear order which contains $v_i v_{i+1} v_{i+3} v_{i+5}$ as a substring. Applying ψ_1 results in $v_{i+1} v_i v_{i+3} v_{i+5}$, and then applying ψ_2 results in $v_{i+1} v_{i+3} v_i v_{i+5}$. If $v_{i+1} = v_{i+4}$, then $\psi_3 = \{v_{i+1}, v_{i+5}\}$. So we apply ψ_3 to obtain $v_{i+1} v_{i+3} v_{i+5} v_i$. The next edge in the hexagonal cycle is parallel to the first, so ψ_1 should be applied again. However, v_i and v_{i+1} are not adjacent, and thus these four linear orders cannot be a part of a hexagonal cycle which violates our assumption. Therefore, only $v_{i+3} = v_{i+4}$ is possible.

Furthermore, the same applies to the L_3-L_4-L_5 path. A common element between ψ_3 and ψ_1 must be set. Here is the information about the edge labels so far:

$$\psi_1 = \{v_i, v_{i+1}\}$$
$$\psi_2 = \{v_{i+1}, v_{i+3}\}$$
$$\psi_3 = \{v_{i+3}, v_{i+5}\}$$

The only sensible choice is to set $v_{i+5} = v_i$, which gives $\psi_1 = \{v_i, v_{i+1}\}$, $\psi_2 = \{v_{i+1}, v_{i+3}\}$, and $\psi_3 = \{v_{i+3}, v_i\}$. This means that v_i, v_{i+1}, and v_{i+3} are adjacent to each other in the six linear orders. P can then be constructed as $P = (V, <_P) = \text{leveled}(\{v_1\}, \{v_2\}, \ldots, \{v_i, v_{i+1}, v_{i+3}\}, \ldots, \{v_n\})$, which shows $P \in \text{LEG}_{\text{HEX}}$. □

The theorem has established that only the posets in Leveled(3) have convex hexagonal ATGs. Now, we can inquire about the number of hexagonal cycles in an arbitrary ATG. The following corollary shows that an arbitrary edge in the complete ATG is contained by at most 2 hexagonal cycles. This implies that completing hexagonal cycles in an arbitrary ATG may take at most 2 steps multiplied by the number of edges. This time, generating all hexagonal cycles in a random ATG will take $O(m^2) \cdot O(2) = O(m^2)$ time.

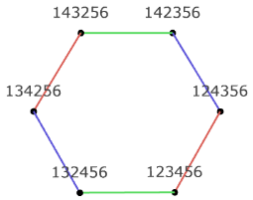


Figure 5: The ATG of leveled({1},{2,3,4},{5},{6}) an illustration of Theorem 2.5.

Corollary 2.6. *Every edge in a complete ATG is contained in the ATG of one or two posets in* LEG_{HEX}.

Proof. Let $L_{ij} = (v_1, v_2, \ldots, v_h, v_i, v_j, v_k, \ldots, v_n)$ be a linear order over $V = \{v_1, v_2, \ldots, v_n\}$. Let L_{ji} be the same linear order except that v_i and v_j are swapped. Thus, $\{L_{ij}, L_{ji}\}$ is an edge in the complete ATG with the edge label $\psi(L_{ij}, L_{ji}) = \{v_i, v_j\}$.

According to Theorem 2.5, the only hexagonal cycles which contain $\{L_{ij}, L_{ji}\}$ as an edge are described by Leveled(3). Among these leveled posets, only two include the edge $\{L_{ij}, L_{ji}\}$ in their ATG:

$$P_{\text{left}} = \text{leveled}(\{v_1\}, \{v_2\}, \ldots, \{v_h, v_i, v_j\}, \ldots, \{v_n\})$$
$$P_{\text{right}} = \text{leveled}(\{v_1\}, \{v_2\}, \ldots, \{v_i, v_j, v_k\}, \ldots, \{v_n\})$$

If v_i and v_j happen to occur on either end of the linear order, only one hexagonal cycle includes the edge as either P_{left} or P_{right} does not exist. □

2.4.4 Generating Hexagonal Cycles. The established theorems allow us to count the cycles that contain a particular edge in a complete ATG. Specifically, an edge is contained by at most 2 hexagonal cycles or at most $n - 3$ square cycles. Since the number of square cycles grow linearly with respect to n, we opted to generate only the hexagonal cycles. The cycles can be generated as described in Corollary 2.6. This is illustrated in Figure 6.

Figure 6: An illustration how two hexagonal cycles can be generated from an arbitrary edge using Corollary 2.6

2.5 Drawing Algorithm "SuperHex"

We can now create an alternative drawing algorithm. Since we will build on top of the previous algorithm, we will still use the supercover subroutine. Then, the hexagonal cycles that contain the edges in the supercover will be included in the FDA's input as well.

Algorithm 4 SuperHex

Input: A set Υ of m linear orders of length n

$\Upsilon' \leftarrow Supercover(\Upsilon)$
$\Upsilon' \leftarrow HexagonalCycles(\Upsilon')$
$G \leftarrow \mathcal{G}(\Upsilon')$
FruchtermanReingold(G)

Algorithm 5 HexagonalCycles Subroutine

Input: A set Υ of m linear orders of length n

$\Upsilon' \leftarrow \Upsilon$
for every edge $\{L_1, L_2\}$ in $\mathcal{G}(\Upsilon)$ **do**
    \\ Left Hexagon
    $i = SwapIndex(L_1, L_2) - 1$
    if $i \geq 0$ **then**
        $\Upsilon' = \Upsilon' \cup Permute(L_1, i)$
    end if
    \\ Right Hexagon
    $i = SwapIndex(L_1, L_2)$
    if $i \leq n - 2$ **then**
        $\Upsilon' = \Upsilon' \cup Permute(L_1, i)$
    end if
end for
return Υ'

The HexagonalCycles subroutine is the application of Corollary 2.6. It contains two helper functions that are not explicitly elaborated. The first function, SwapIndex, linearly searchers the linear orders it receives to find the smaller index of the swapped elements. The second function, Permute, is tasked to produce the six linear orders of the hexagonal cycle by permuting the ith, $(i + 1)$th, and $(i + 2)$th elements in L_1.

The first function, SwapIndex, runs in $O(n)$ time due to the linear search. The second function, Permute, runs in $O(1)$ time. The loop would run in $O(|E|) = O(m^2)$ as there are at most $O(m^2)$ edges. Thus, in total, HexagonalCycles runs in $O(m^2 n)$ time.

Since the algorithm still employs the supercover subroutine that can result in the permutahedron, Algorithm 4 runs in $O(n!^2)$.

Visual inspection shows that this algorithm creates better layouts for graphs that we already know are distorted, as illustrated in Figure 3. However, there are also graphs that it draws worse than the previous algorithm, Figure 7 is an example.

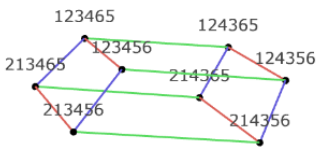


Figure 7: A cube-shaped ATG flattened by the "SuperHex" algorithm.

2.6 Summary of Drawing Algorithms

Based on the observations discussed in the previous sections, the drawing algorithms were evaluated analytically with respect to time complexity and visually with respect to the parallel edges property, the prominent cycles property, and graph readability. The most promising algorithms are "Supercover" and "SuperHex". Though the algorithms have worse time complexity than the "Default" algorithm, they draw an ATG according to its properties. And, in comparison to the "Permutahedron" algorithm, both algorithms are slightly better since the supercover subroutine does not always return the set of all possible linear orders. Since a larger input size m corresponds to an input that is closer to including all possible linear orders, any algorithm will tend to behave like the "Permutahedron" algorithm with sufficiently large m.

3 Conclusion

In this paper, drawing methods were devised to draw ATGs that highlight their inherent property - prominence of parallel edges, square cycles, and hexagonal cycles. The drawing methods augment an FDA by adding more vertices to the input. Additional vertices were strategically determined to take advantage of the spring forces of an FDA and draw more prominent parallel edges and cycles. Moreover, it was also observed that inputs resulting in incomplete cycles tend to distort the graph layouts. Certain classes of leveled posets were shown to be equivalent to classes of posets characterized by square and hexagonal ATGs. The numbers of such cycles were also established in this paper.

Acknowledgements

Harold C. Antonio expresses deep appreciation for the financial assistance provided by the ERDT Program of DOST-SEI in carrying out this research.

References

[1] Agrawal, R., Gunopulos, D., Leymann, F.: Mining process models from workflow logs. In: International Conference on Extending Database Technology. pp. 467–483. Springer (1998)
[2] Björner, A., Wachs, M.L.: Permutation statistics and linear extensions of posets. Journal of Combinatorial Theory, Series A **58**(1), 85–114 (1991)
[3] Djoković, D.Ž.: Distance-preserving subgraphs of hypercubes. Journal of Combinatorial Theory, Series B **14**(3), 263–267 (1973)
[4] Fernandez, P.L., Heath, L.S., Ramakrishnan, N., Tan, M., Vergara, J.P.C.: Mining posets from linear orders. Discrete Mathematics, Algorithms and Applications **5**(04), 1350030 (2013)
[5] Hagberg, A.A., Schult, D.A., Swart, P.J.: Exploring network structure, dynamics, and function using networkx. In: Varoquaux, G., Vaught, T., Millman, J. (eds.) Proceedings of the 7th Python in Science Conference. pp. 11 – 15. Pasadena, CA USA (2008)
[6] Kobourov, S.G.: Spring embedders and force directed graph drawing algorithms. arXiv preprint arXiv:1201.3011 (2012)
[7] Massow, M.: Linear extension graphs and linear extension diameter. Cuvillier Verlag (2010)
[8] Naatz, M.: The graph of linear extensions revisited. SIAM Journal on Discrete Mathematics **13**(3), 354–369 (2000)
[9] Ordanel, I., Fernandez Jr, P., Adorna, H.: A polynomial time algorithm for the 2-poset cover problem. Information Processing Letters **169**, 106106 (2021)
[10] Puolamäki, K., Fortelius, M., Mannila, H.: Seriation in paleontological data using markov chain monte carlo methods. PLoS computational biology **2**(2), e6 (2006)
[11] Reuter, K.: Linear extensions of a poset as abstract convex sets. Univ. Hamburg, Mathematisches Seminar (1996)
[12] Winkler, P.M.: Isometric embedding in products of complete graphs. Discrete Applied Mathematics **7**(2), 221–225 (1984)

Analysis of Machine Learning Approaches for Real-Time Human Exercise Classification Using Pose Estimation Joint Angles

Moses Limbo
University of the Philippines Diliman
Department of Computer Science
Quezon City, Philippines
mmlimbo@up.edu.ph

Abstract

This paper presents a comprehensive comparative study of machine learning algorithms for automated human exercise classification using joint angle features extracted from pose estimation systems. We evaluate nine classical machine learning models and a deep learning approach (LSTM) on a dataset of 31,033 frames across five exercise classes: Jumping Jacks, Pull-ups, Push-ups, Russian Twists, and Squats. Features consist of five joint angles (shoulder, elbow, hip, knee, and ankle) computed from skeletal keypoints. Our experiments demonstrate that k-Nearest Neighbors (kNN) achieves the highest test accuracy of 95.52% among classical models, closely followed by XGBoost (95.39%) and Random Forest (95.10%). While top classical ML models have higher test accuracy, it suffers from misclassifications on real-time inference due to a lack of temporal context. The LSTM model, designed to capture temporal dependencies in exercise sequences, achieves 80.49% accuracy but shows robustness to transient pose variations during inference. All models are optimized using Bayesian hyperparameter tuning via Optuna with 10-fold cross-validation. Our results provide insights into model selection for real-time exercise recognition systems and highlight the trade-offs between frame-based and sequence-based classification approaches.

Keywords

Exercise classification, pose estimation, machine learning, LSTM, human activity recognition

1 Introduction

Human activity recognition (HAR) has gained significant attention due to its applications in fitness tracking, physical therapy, elderly care monitoring, and interactive gaming systems. With the advancement of computer vision techniques, particularly pose estimation algorithms such as MediaPipe and OpenPose, extracting skeletal keypoints from video streams has become both accurate and computationally efficient [7].

Exercise classification represents a specific domain within HAR where the goal is to automatically identify which physical exercise a person is performing based on body pose information. This capability enables applications such as automated fitness coaching, exercise form correction, and rehabilitation monitoring systems.

Traditional approaches to exercise recognition often rely on wearable sensors (accelerometers, gyroscopes), which require users to wear specialized equipment. Vision-based approaches offer a contactless alternative that can work with standard cameras, making them more accessible for home fitness applications.

In this work, we present a comprehensive comparison of machine learning approaches for exercise classification using joint angle features derived from pose estimation. Our contributions include:

- A systematic evaluation of nine classical machine learning algorithms with Bayesian hyperparameter optimization
- Implementation and analysis of an LSTM-based temporal sequence classifier
- Detailed performance comparison using multiple evaluation metrics
- Analysis of per-class performance to identify challenging exercise categories
- Inference methodology for real-time exercise recognition and checking model robustness.

The remainder of this paper is organized as follows: Section II reviews related work in exercise and activity recognition. Section III describes the dataset and feature extraction process. Section IV details the machine learning methodologies employed. Section V outlines the inference methodology. Section VI presents exploratory data analysis. Section VII discusses training results and analysis. Section VIII discusses practical implications, limitations, and future work. Finally, Section IX concludes the paper.

2 Related Work

2.1 Human Activity Recognition

Human activity recognition has been extensively studied using various sensor modalities. Wearable sensor-based approaches [3] have achieved high accuracy but require dedicated hardware. Vision-based methods have evolved from handcrafted features [11] to deep learning approaches [4].

2.2 Pose Estimation for Activity Recognition

Recent advances in pose estimation have enabled skeleton-based action recognition. MediaPipe Pose [7] provides real-time 33-keypoint body pose estimation suitable for mobile and edge devices. These skeletal representations reduce the complexity of video-based recognition by abstracting appearance information.

2.3 Exercise Classification

Exercise classification has been addressed using various approaches, including CNN-based methods [6], recurrent neural networks [5], and classical machine learning with handcrafted features [10]. Joint angles computed from skeletal keypoints provide geometric features that are invariant to body size and position in the frame.

Velloso et al. [10] demonstrated Random Forest-based classification of weight lifting exercises using wearable sensors, reporting

accuracies in the range of 78–92%, but requiring dedicated sensor hardware. Hammerla et al. [5] evaluated deep, convolutional, and recurrent models on wearable-based HAR, finding that LSTMs offer robustness to temporal variability but require sufficiently rich features to outperform frame-level classifiers. Khurana et al. [6] proposed GymCam for recognizing simultaneous exercises in unconstrained environments, but relied on multi-camera setups, limiting home applicability.

More recently, Riccio R. [8] proposed a BiLSTM system using both joint angles and raw coordinates across 22 landmarks, achieving over 99% accuracy on four exercise classes.

Schlegel et al. [9] demonstrated that biomechanical joint angle representations are competitive with raw keypoint coordinates for exercise classification across LSTM, CNN, and transformer models. In contrast, our work adopts a contactless single-camera pipeline using only five joint angles derived from MediaPipe, requiring no specialized hardware. No prior published work was identified using the same dataset, making direct numerical comparison infeasible; however, our classical models achieve competitive accuracy relative to these prior systems.

3 Dataset and Feature Extraction

3.1 Dataset Description

Our dataset comes from Exercise Detection Dataset [1]. The dataset consists of 31,033 frames extracted from video recordings of individuals performing five distinct exercises:

- **Jumping Jacks**: Full-body cardio exercise involving arm and leg movements
- **Pull-ups**: Upper body strength exercise
- **Push-ups**: Upper body exercise performed in prone position
- **Russian Twists**: Core exercise involving torso rotation
- **Squats**: Lower body exercise focusing on leg and hip movement

Table 1 summarizes the dataset distribution across exercise classes.

Table 1: Dataset Distribution

Exercise	Samples
Jumping Jacks	4,817
Pull-ups	6,268
Push-ups	9,374
Russian Twists	4,015
Squats	4,608
Total	31,033

3.2 Feature Extraction

From the pose estimation keypoints, we compute five joint angles that capture the essential body configuration:

(1) **Shoulder Angle**: Angle at the shoulder joint between the upper arm and the torso
(2) **Elbow Angle**: Angle at the elbow between the upper and lower arm
(3) **Hip Angle**: Angle at the hip between torso and thigh
(4) **Knee Angle**: Angle at the knee between thigh and lower leg
(5) **Ankle Angle**: Angle at the ankle between the lower leg and the foot

These angles are computed using the three-point angle formula based on arctangent:

$$\theta = \left|\arctan 2(c_y - b_y, c_x - b_x) - \arctan 2(a_y - b_y, a_x - b_x)\right| \times \frac{180}{\pi} \quad (1)$$

where points a, b, and c represent the three joint positions, with b being the vertex of the angle. The result is converted from radians to degrees and adjusted to the range $[0, 180]$ degrees.

3.3 Data Preprocessing

All features are standardized using z-score normalization:

$$x_{normalized} = \frac{x - \mu}{\sigma} \quad (2)$$

where μ and σ are the mean and standard deviation computed from the training set. For the classical machine learning models, we use an 80-20 train-test split with stratified sampling to maintain class proportions.

4 Methodology

4.1 Classical Machine Learning Models

We evaluate nine classical machine learning algorithms, each optimized using Bayesian hyperparameter tuning with Optuna [2] using 50 trials with Tree-structured Parzen Estimator (TPE) sampling and 10-fold cross-validation. The models include: Logistic Regression (LR), Gaussian Naive Bayes (GNB), k-Nearest Neighbors (kNN), Decision Tree (DT), AdaBoost, XGBoost, Random Forest (RF), Support Vector Machine (SVM with RBF kernel), and Multi-Layer Perceptron (MLP). Table 2 summarizes the final tuned hyperparameters for each model.

Table 2: Best Hyperparameters from Bayesian Tuning

Model	Key Hyperparameters
LR	*C= 0.0173*, penalty=*l2*, solver=*saga*
GNB	var_smoothing= *7.459*
kNN	*k=4*, metric=*euclidean*, weights=*distance*
DT	max_depth=*19*, min_samples_split=*7*, min_samples_leaf=*3*
AdaBoost	n_estimators=*192*, lr= *0.700*
XGBoost	n_est=*238*, depth=*10*, lr= *0.280*
RF	n_est=*140*, max_depth=*18*
SVM	*C= 99.238*, *γ=scale*, kernel=*RBF*
MLP	layers=*(50, 50)*, activation=*tanh*, lr=*constant*

4.2 LSTM-based Temporal Classification

To capture temporal dependencies inherent in exercise movements, we implement a Long Short-Term Memory (LSTM) network that processes sequences of consecutive frames.

4.2.1 Sequence Creation. For each exercise class, we apply stratified temporal splitting: the first 70% is used for training, the next 15% for validation, and the final 15% for testing. Sequences of 60 consecutive frames are created using a sliding window approach within each split to prevent data leakage.

4.2.2 Model Architecture and Training. The LSTM architecture takes input of shape $(batch, 60, 5)$, passes through two LSTM layers with 128 hidden units each, applies Dropout (0.3) for regularization, and maps to 5 output classes via a fully connected layer with Softmax activation. Training uses cross-entropy loss, Adam optimizer (lr=0.001), ReduceLROnPlateau scheduling, early stopping (patience=15), and batch size of 32.

4.3 Training and Inferencing Hardware

All classical machine learning models were trained and evaluated on a Desktop Computer with a Ryzen 7 7700 CPU and 32GB RAM. The LSTM model was trained on a machine equipped with an NVIDIA RTX 3080 Ti GPU to accelerate training.

5 Inference methodology

For real-time exercise classification, we implement separate inference pipelines for classical machine learning models and the LSTM model. Both pipelines utilize real-time pose estimation using MediaPipe [7] to extract joint keypoints from video frames.

5.1 Classical ML Inference

For classical machine learning models, inference is performed in real-time by capturing frames from a video stream, which simulates the webcam input. Each frame is processed through the pose estimation pipeline to extract joint keypoints, from which the five joint angles are computed. The angles are then buffered to maintain a history of the last 60 frames. For each new frame, the joint angles are standardized using the training set statistics and passed to the trained classical model for prediction. The predicted exercise class is output immediately.

5.2 LSTM Inference

For the LSTM model, inference also begins with real-time frame capture and pose estimation to extract joint keypoints. The five joint angles are computed and added to a rolling buffer that maintains the last 60 frames. Once the buffer is full, the sequence of joint angles is standardized and reshaped to match the LSTM input dimensions. This sequence is then fed into the trained LSTM model to obtain class probabilities. The exercise class with the highest probability is selected as the predicted output. The buffer is updated with each new frame, allowing continuous real-time predictions.

6 Exploratory Data Analysis

Figure 1 shows the pairplot of the five joint angle features colored by exercise class. Distinct clusters are observed for Squats and Russian Twists, indicating unique joint angle configurations for these exercises. Pull-ups, Push-ups, and Jumping Jacks show some overlap, suggesting potential classification challenges for these classes.

Figure 1: Pairplot of Joint Angle Features Colored by Exercise Class

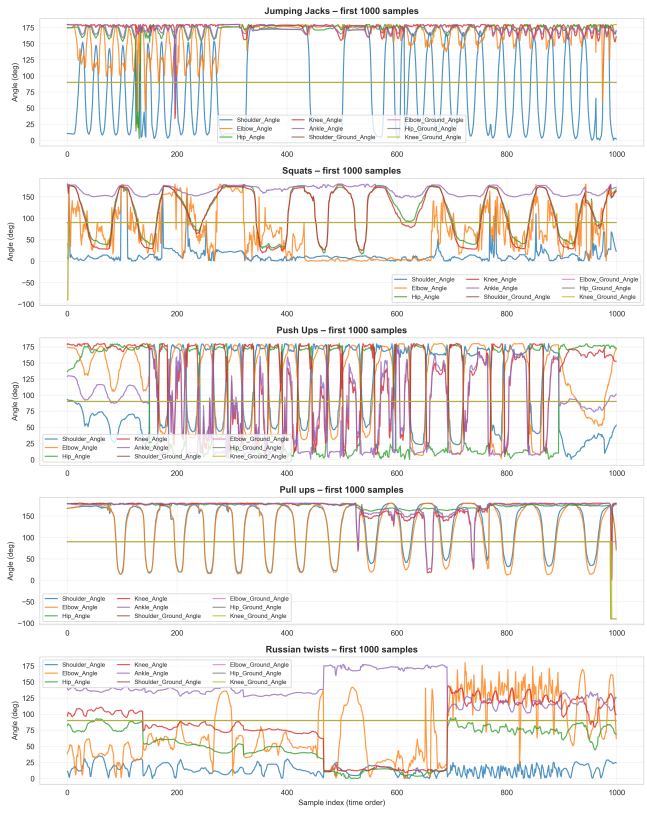


Figure 2: Time Series of Joint Angles by Exercise Class

Figure 2 illustrates the time series of the five joint angles for each exercise class, highlighting temporal patterns and variations. In Jumping Jacks, periodic oscillations in shoulder and elbow angles are evident, while Push-ups show clear fluctuations in shoulder, elbow, knee, and ankle angles. Russian Twists exhibit separable patterns in ankle, knee, hip, elbow, and shoulder angles. Squats display distinct knee and hip angle patterns. And pull-ups show unique shoulder and elbow angle variations.

7 Training Results and Analysis

7.1 Classical Model Performance

Table 3 presents the comprehensive evaluation results for all classical machine learning models.

Table 3: Classical Model Performance Comparison

Model	Train	Test	Prec.	Rec.	F1
kNN	1.000	0.955	0.955	0.955	0.955
XGBoost	1.000	0.954	0.954	0.954	0.954
RandomForest	0.999	0.951	0.951	0.951	0.951
MLP	0.957	0.943	0.944	0.943	0.943
DecisionTree	0.971	0.921	0.921	0.921	0.921
SVM	0.913	0.909	0.910	0.909	0.909
AdaBoost	0.711	0.709	0.702	0.709	0.704
LogRegression	0.705	0.706	0.708	0.706	0.706
GaussianNB	0.643	0.649	0.655	0.649	0.613

The k-Nearest Neighbors classifier achieves the highest test accuracy of 95.52%, closely followed by XGBoost (95.39%) and Random Forest (95.10%). The strong performance of kNN suggests that exercise classes form well-separated clusters in the 5-dimensional joint angle feature space.

7.2 Per-Class ROC Analysis

Table 4 presents the per-class ROC AUC scores for the top-performing models.

Table 4: Per-Class ROC AUC Scores

Exercise	kNN	XGBoost	RF
Jumping Jacks	0.989	0.998	0.997
Pull-ups	0.985	0.996	0.995
Push-ups	0.998	0.999	0.999
Russian Twists	0.983	0.997	0.997
Squats	0.984	0.995	0.994

All top-performing models achieve ROC AUC scores above 0.98 for all exercise classes, indicating excellent discriminative capability. Push-ups consistently show the highest AUC, likely due to the distinctive prone body position reflected in the joint angles.

7.3 LSTM Temporal Model Results

The LSTM model achieves the following test set performance:

Table 5: LSTM Model Overall Performance

Metric	Score
Accuracy	0.805
Precision (weighted)	0.812
Recall (weighted)	0.805
F1-Score (weighted)	0.788

Table 6: LSTM Per-Class Performance

Exercise	Prec.	Rec.	F1	Support
Jumping Jacks	0.87	0.94	0.90	723
Pull-ups	1.00	0.95	0.97	940
Push-ups	0.88	1.00	0.93	1406
Russian Twists	0.72	0.24	0.36	602
Squats	0.44	0.57	0.50	691

The LSTM model shows strong performance on Pull-ups (F1=0.97), Push-ups (F1=0.93), and Jumping Jacks (F1=0.90), but struggles with Russian Twists (F1=0.36) and Squats (F1=0.50). This disparity suggests that the temporal patterns of Russian Twists and Squats may be more subtle or similar to other exercises when viewed as sequences.

7.4 Model Comparison Analysis

7.4.1 Classical vs. Temporal Approaches. Surprisingly, the frame-based classical models outperform the sequence-based LSTM model. This can be attributed to the high discriminative power of joint angles at the frame level, the reduced effective training data due to temporal splitting for the LSTM, and the similarity of temporal patterns between some exercise pairs despite distinct pose configurations.

7.4.2 Class Imbalance. The dataset exhibits notable class imbalance, with Push-ups (9,374 samples) having more than twice the samples of Russian Twists (4,015). No resampling was applied during training. This imbalance likely contributes to the LSTM's poor performance on Russian Twists (F1=0.36) and Squats (F1=0.50), as the model may be biased toward overrepresented classes. Future work should explore oversampling techniques such as SMOTE or class-weighted loss functions to address this.

7.4.3 Overfitting Analysis. Models showing training accuracy of 1.0 (kNN, XGBoost) with test accuracy above 0.95 demonstrate good generalization despite the train-test accuracy gap. The Decision Tree shows more pronounced overfitting (97.1% train vs. 92.1% test).

7.4.4 Linear vs. Non-linear Models. Linear models (Logistic Regression, Gaussian NB) achieve around 65-70% accuracy, indicating that exercise classes are not linearly separable in the feature space. Non-linear models (kNN, tree-based ensembles, SVM with RBF kernel) significantly outperform linear approaches.

7.5 Comparison with Prior Work

Direct numerical comparison is constrained by the absence of published work using the same dataset. Methodologically, our classical models exceed the 78–92% accuracy range reported by Velloso et al. [10] on wearable-based weight lifting recognition, achieving up to 95.52% using only vision-derived joint angles. Our LSTM (80.49%) is consistent with Hammerla et al. [5], who found that recurrent models require rich temporal features to surpass frame-level classifiers, a finding that aligns with our five-angle feature representation being the likely bottleneck for temporal modeling. A recent BiLSTM system [8] reports over 99% accuracy using a BiLSTM with 12 joint angles across 22 landmarks on four exercise classes; the performance gap relative to our LSTM supports the view that richer feature representations are needed to fully exploit temporal models. This is further supported by findings in Schlegel et al. [9], which show that joint angle features are competitive with raw keypoint coordinates for exercise classification using sequence-based models.

8 Inference Results

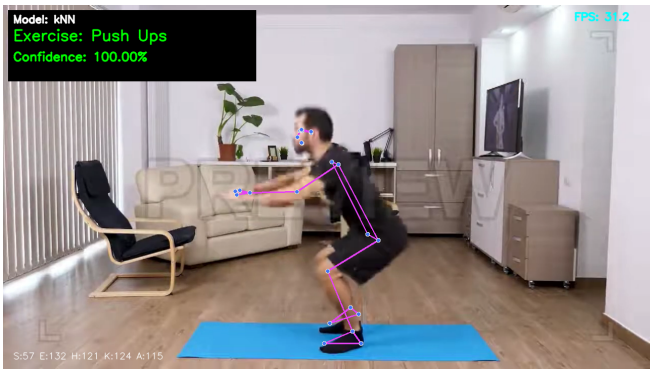


Figure 3: kNN Inference on Squats (Incorrect Prediction)

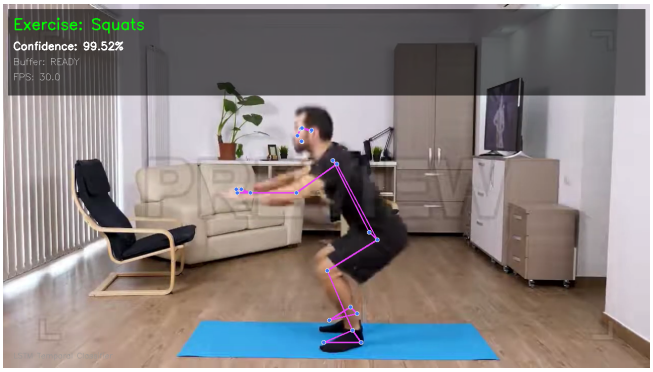


Figure 4: LSTM Inference on Squats (Correct Prediction)

Figure 3 illustrates the kNN model's inference results on real-time video input. In figure 3, it misclassifies squats as a Push-up, due to a change of body orientation. There are more misclassifications on exercises like jumping jacks (misclassified as push-ups), squats (misclassified as push-ups or russian twists), and pull-ups (misclassified as jumping jacks or push-ups) due to a change of body orientation or occlusion of keypoints.

Figure 4 shows the LSTM correctly classifying the same Squats input where kNN failed, demonstrating the temporal model's robustness to orientation changes. All exercises are correctly classified by the LSTM model during inference due to the temporal classification of the LSTM model.

9 Conclusion

This paper presents a comprehensive comparison of machine learning approaches for exercise classification using joint angle features from pose estimation. Our experiments on a dataset of 31,033 frames across five exercise classes demonstrate that k-Nearest Neighbors achieves the highest accuracy (95.52%) among evaluated methods, followed closely by XGBoost (95.39%) and Random Forest (95.10%).

Exploratory data analysis revealed distinct patterns in joint angles on time series for different exercises, while on Pairplot, some exercises showed overlapping feature distributions leading to classification challenges.

While classical models outperform the LSTM temporal model in accuracy, the LSTM offers robustness to pose variations during real-time inference. Bayesian hyperparameter optimization via Optuna with 10-fold cross-validation proved effective in enhancing model performance.

The results provide practical guidance for deploying exercise recognition systems: classical models are suitable for low memory environments, while LSTM models offer advantages in handling temporal dependencies and pose variability. Future work will explore larger datasets, time-series classification techniques for classical models, and advanced temporal modeling techniques to further improve exercise recognition capabilities.

9.1 Limitations

Our study has several limitations:

- The dataset's number of subjects and recording conditions are not documented, raising concerns that the same individual may appear in both train and test splits, potentially inflating accuracy metrics
- Class imbalance (Push-ups: 9,374 vs. Russian Twists: 4,015) was not addressed through resampling
- Feature representation is limited to five joint angles; motion-based features (e.g., angular velocity) may better support temporal modeling
- Single viewpoint assumption; pose estimation errors are not explicitly modeled
- LSTM architecture and temporal classification techniques (e.g., HMMs, DTW) remain unexplored

9.2 Future Work

Future directions include expanding the exercise taxonomy, incorporating attention mechanisms and time-series classification techniques for classical models, multi-view pose fusion, transfer learning from large-scale action recognition datasets, and real-time form correction feedback.

References

[1] [n. d.]. Exercise Detection dataset. https://www.kaggle.com/datasets/mrigaankjaswal/exercise-detection-dataset

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2623–2631.

[3] Andreas Bulling, Ulf Blanke, and Bernt Schiele. 2014. A Tutorial on Human Activity Recognition Using Body-worn Inertial Sensors. *Comput. Surveys* 46, 3 (2014), 1–33.

[4] João Carreira and Andrew Zisserman. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6299–6308.

[5] Nils Y Hammerla, Shane Halloran, and Thomas Plötz. 2016. Deep, Convolutional, and Recurrent Models for Human Activity Recognition Using Wearables. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 1533–1540.

[6] Rushil Khurana, Karan Ahuja, Zac Yu, Jennifer Mankoff, Chris Harrison, and Mayank Goel. 2018. GymCam: Detecting, Recognizing and Tracking Simultaneous Exercises in Unconstrained Scenes. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 4 (2018), 1–17.

[7] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. 2019. MediaPipe: A Framework for Building Perception Pipelines. arXiv:1906.08172

[8] Riccardo Riccio. 2024. Real-Time Fitness Exercise Classification and Counting from Video Frames. arXiv:2411.11548 [cs.CV] https://arxiv.org/abs/2411.11548

[9] Kevin Schlegel, Lei Jiang, and Hao Ni. 2024. Using joint angles based on the international biomechanical standards for human action recognition and related tasks. arXiv:2406.17443 [cs.CV] https://arxiv.org/abs/2406.17443

[10] Eduardo Velloso, Andreas Bulling, Hans Gellersen, Wallace Ugulino, and Hugo Fuks. 2013. Qualitative Activity Recognition of Weight Lifting Exercises. In *Proceedings of the 4th Augmented Human International Conference*. 116–123.

[11] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. 2013. Dense Trajectories and Motion Boundary Descriptors for Action Recognition. *International Journal of Computer Vision* 103, 1 (2013), 60–79.

ATHENA: An Image Captioning Pipeline for Constrained Underwater Environments

Rodrigo Emmanuel Roy
Krisha Anne Chan
Carlo Raquel
Prospero Naval

ABSTRACT

Autonomous underwater vehicles (AUVs) are constrained by low-bandwidth communication channels, as well as limited onboard computation. Current approaches either rely heavily on cloud processing or transmitting raw images, which are infeasible on untethered AUVs. In this work, we introduce ATHENA: AI Transformer for Hydro-Environmental Navigation and Analysis, an edge-capable Vision-Language Model (VLM) pipeline that can enable AUVs to describe underwater scenes as text, allowing for efficient data transmission and real-time situational awareness. To address common underwater challenges such as color distortion, light attenuation, and haze, we apply a multi-step preprocessing pipeline that enhances image clarity for the VLM. By leveraging 4-bit quantization, ATHENA reduces the memory footprint to just 604.48MB, making it deployable on embedded hardware. Preliminary results show concise, accurate captions with significant improvements on standard image captioning benchmarks such as BLEU, ROUGE-L, and CIDEr.

CCS CONCEPTS

• **Computing methodologies** → *Computer vision*; **Scene understanding**; • **Computer systems organization** → *Embedded software*; *Robotics*;

KEYWORDS

underwater image captioning, autonomous underwater vehicles, vision-language models, underwater image enhancement

1 INTRODUCTION

Autonomous underwater vehicles (AUVs) are increasingly deployed for marine exploration, environmental monitoring, and subsea infrastructure inspection. Long-duration autonomous missions require onboard perception systems that can summarize visual scenes and transmit only essential information to remote operators. However, this is constrained by two defining characteristics of underwater robotics: severely limited communication bandwidth and restricted onboard computational resources [16]. Underwater communication is typically achieved via acoustic channels, which use sound waves to transmit information over long distances, since radio signals attenuate rapidly in water and optical signals require line-of-sight. Acoustic channels have low bandwidth, high latency, and multipath interference, making raw or high-resolution image transmission and cloud-based processing impractical, especially for untethered AUVs [10][22]. Many state-of-the-art vision models also exceed the compute and memory budgets of embedded AUV hardware [10][14].

Underwater imagery further complicates perception due to light attenuation, scattering, haze, and color distortion, which vary with depth, turbidity, and lighting conditions. Prior work highlights the need for enhancement and restoration pipelines to address these challenges before downstream processing [10].

To overcome these limitations, recent approaches focus on conveying semantic content rather than raw images. Textual descriptions offer a compact, bandwidth-efficient alternative [22]. Vision-language models (VLMs) such as SINGER [1] and SmolVLM [14] demonstrate the feasibility of this approach, as existing architectures are often too large for edge deployment or are not tailored to underwater imagery.

Inspired by those previous works, we propose an edge-capable underwater scene captioning system that enhances raw images and produces concise textual descriptions suitable for low-bandwidth transmission, enabling AUVs to communicate semantically meaningful information in near real-time.

The remainder of this paper is organized as follows: Section 2 reviews related work on vision-language models, semantic communication, and underwater image enhancement. Section 3 summarizes the key challenges in underwater image captioning and our contributions, Section 4 presents the ATHENA framework for edge-capable underwater scene captioning. Section 5 details experiments and analyzes results. Finally, Section 6 concludes the paper.

2 RELATED WORK

Recent advances in autonomous robotics and computer vision have explored various approaches for perception, communication, and image enhancement. In the context of underwater scene understanding, three main areas are particularly relevant: vision-language models (VLMs) for robotic perception, semantic communication methods for low-bandwidth environments, and specialized techniques for underwater image enhancement. We discuss each of these in turn.

2.1 VLMs as Perception Modules

Vision-language models (VLMs) streamline robotic perception by converting images directly into compact textual descriptions, making them ideal for low-bandwidth underwater communication [1, 14]. However, deploying VLMs on AUVs requires balancing performance with strict onboard computational limits. While high-performing models are often too large for edge deployment [1] and intermediate models like BLIP-2 [9] can lack reasoning consistency [5], lightweight models like SmolVLM [14] provide practical, memory-efficient alternatives.

Despite these architectural optimizations, terrestrial-trained VLMs consistently struggle with underwater visual degradations such as color attenuation, backscatter, and turbidity [8]. This severe domain gap necessitates the specialized image enhancement and domain-adaptive fine-tuning strategies explored in this study.

2.2 Underwater Image Enhancement

While VLMs enable compact semantic representations, their performance depends heavily on input image quality. Underwater imagery is inherently degraded by color attenuation, scattering, and low contrast, which can severely mislead VLM perception.

To address these domain-specific distortions, various enhancement pipelines have been proposed. Approaches range from physics-based models like Sea-thru [2], which rely primarily on attenuation compensation and white balance, to more comprehensive multi-step pipelines incorporating color correction, contrast stretching, and high-pass fusion [13, 18]. Recent studies also highlight that techniques like Contrast Limited Adaptive Histogram Equalization (CLAHE) often outperform standard contrast stretching by better preserving local details [20]. Despite these advances, there is no standardized, computationally lightweight pipeline universally robust to diverse underwater conditions, motivating the tailored preprocessing approach evaluated in this work.

3 CHALLENGES AND CONTRIBUTIONS

This section synthesizes the key challenges limiting progress in underwater image captioning and outlines the novel contributions proposed to address them.

3.1 Key Challenges

Existing underwater image captioning approaches face three key challenges. First, most models rely on terrestrial image datasets and underperform in underwater conditions [8]. Second, underwater images are degraded by haze, low contrast, and scattering, and enhancement methods are inconsistent. Third, AUVs are constrained by limited onboard compute and low-bandwidth communication, requiring lightweight models and compact representations [16]. Addressing these challenges motivates the contributions of this work.

3.2 Contributions

To address the three challenges listed in Section 3.1, we make three novel contributions.

3.2.1 Domain adaptive finetuning. We evaluate SmolVLM-Instruct as our VLM backbone and apply parameter-efficient finetuning using Low-Rank Adaptation (LoRA) to adapt models pre-trained on terrestrial imagery to the underwater domain. To systematically study how fine-tuning configurations interact, we conduct an ablation study across 6 configurations, varying LoRA rank (8, 16, 32) and whether DoRA parameterization is enabled. We analyze how these factors influence caption quality on underwater imagery and compare the best-performing configuration against existing state-of-the-art methods on standard benchmarks.

3.2.2 Image Enhancement Pipeline. We develop a reproducible underwater image enhancement pipeline (e.g., color correction, contrast improvement, haze reduction) before feeding images into the model to improve feature extraction and caption accuracy.

3.2.3 Practical adaptation for edge-capable deployment. We design a compact VLM pipeline for near real-time, onboard deployment under computational and bandwidth constraints. The model produces accurate semantic scene descriptions for AUVs operating in low-bandwidth underwater environments.

4 ATHENA THEORETICAL FRAMEWORK

4.1 Dataset

We use the underwater image captioning dataset by Huanyu et al. [8], the most comprehensive publicly available dataset of its kind, consisting of 3,176 images each annotated with five human-written captions provided by volunteers with domain expertise (15,880 total). Other datasets such as UIEB [7] and SQUID [4] either lack captions, focus solely on object detection, or are much smaller in scale. Images were resized to 224×224 pixels and span ten categories, predominantly fish (1861), coral reefs (1587), and divers (850), covering diverse real-world scenes including low-light conditions, visual blur, and dense groups. Table 1 presents a representative example.

The dataset provides a comprehensive benchmark for underwater scene understanding and forms the foundation for training and evaluating our vision-language model.

4.2 Architecture

The emphasis in this architecture is the VLM, as it is the core contribution of this paper. We adopt SmolVLM because it is compact enough for real-time, onboard deployment on the limited compute payload of an AUV equipped with a camera, while still providing accurate scene captions for diverse underwater imagery. Section 2.1 justifies this choice.

The pipeline starts with image capture, sending the image and a system prompt as input to the system, followed by an inner pipeline of image enhancement functions such as color correction (see Section 4.2.1). The enhanced image is then fed into the VLM, which generates a structured text description. This output is then transmitted over low-bandwidth channels to a remote operator or control center. The overall architecture is illustrated in Figure 1.

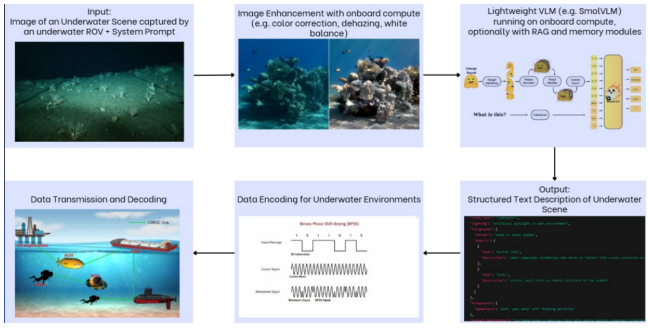


Figure 1: ATHENA Architecture

Table 1: Examples from the underwater image captioning dataset [8]

Image	Ground truth captions
	GT-1: Coral reefs in seawater come in a variety of colors, such as dark-brown, light-brown, and white. GT-2: An underwater coral reef full of life, with all kinds of corals growing freely. GT-3: Coral reefs in the water grow corals of various shapes and colors. GT-4: An undersea coral reef with corals of various colors and shapes. GT-5: There are many corals of various shapes and colors growing on the underwater reefs.
	GT-1: A gray dolphin swims in the blue seawater. GT-2: There is a gray dolphin swimming at the bottom of sea. GT-3: A close-up of a dolphin swimming in the blue seawater. GT-4: There is a gray dolphin swimming in the seawater. GT-5: A gray dolphin swims at the bottom of sea.

4.2.1 Image Enhancement Pipeline. Within the ATHENA architecture, raw underwater images are enhanced before being processed by the VLM. Underwater images often suffer from visual degradations, and this enhancement ensures the VLM receives clearer input for more accurate captioning. The overall process is illustrated in Figure 2.

This approach is largely based on the 7-step underwater image enhancement protocol proposed in Wang et al [18], with a modification: we replace the original contrast stretching step with CLAHE for better local contrast enhancement (Section 2.2 explains why CLAHE is preferable).

The seven steps are: (1) **Attenuated Channel Compensation (ACC)**, which restores absorbed reds and yellows; (2) **White Balance** (YCbCr mean method), which corrects blue/green color casts; (3) **Tone Mapping**, which enhances luminance across dark and bright regions; (4) **HSL Saturation Adjustment**, which boosts color vividness; (5) **CLAHE**, which improves local contrast without amplifying noise; (6) **Gamma Correction**, which adjusts brightness for under- and overexposed regions; and (7) **High-Pass Fusion**, which sharpens edges and fine details for better VLM feature recognition.

These steps target three main visual issues (as stated in Section 2.3): the first four correct color distortions, the next two enhance contrast, and the final step sharpens image details.

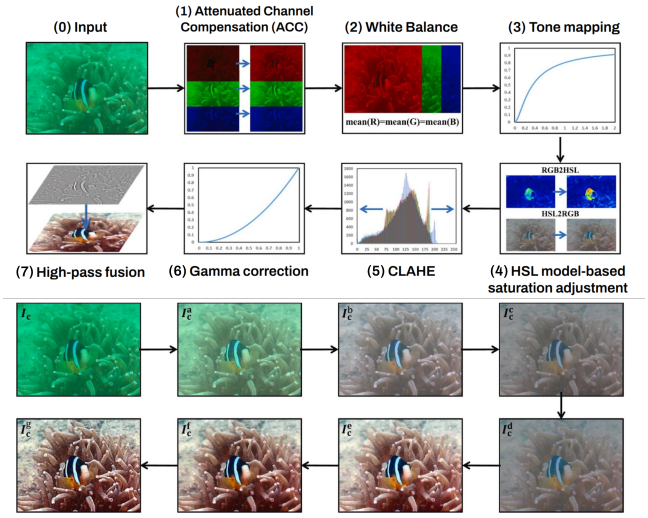


Figure 2: Overview of the image enhancement process within the ATHENA architecture (top) and an additional visual example (bottom).

4.3 Ablation Study - Setup

We evaluate **SmolVLM-Instruct** as our VLM backbone for underwater image captioning. SmolVLM-Instruct is a lightweight model with a sub-1GB memory footprint, making it suitable for near real-time onboard inference on resource-constrained AUV hardware [14]. The backbone is evaluated across three LoRA ranks (8, 16, 32) with DoRA enabled and disabled, yielding six configurations in total, as summarized in Table 2.

Table 2: Ablation study configurations across VLM backbones, LoRA ranks, and DoRA

VLM Backbone	LoRA Rank	DoRA	Configurations
SmolVLM-Instruct	8, 16, 32	True	3
	8, 16, 32	False	3
Total			**6**

5 EXPERIMENT AND DISCUSSION

Model performance was assessed using standard captioning metrics: BLEU-1 through BLEU-4 [15], which measure n-gram overlap (unigram to 4-gram) between predicted and reference captions; ROUGE-L [11], which captures sentence-level structural similarity via longest common subsequence; and CIDEr [17], which rewards consensus with multiple human references. Scores range from 0 to 1 for BLEU and ROUGE-L, and 0 to 5 for CIDEr, with higher values indicating better agreement with human-annotated captions.

5.1 Implementation Details

We implemented ATHENA using *PyTorch* and the *HuggingFace Transformers* library. For this initial implementation, we chose the

SmolVLM-Instruct model, loaded with 4-bit NormalFloat (NF4) quantization using the *bitsandbytes* library.

5.1.1 Data Preprocessing. Before training, all images were resized to 224×224 pixels to match the vision encoder's input resolution. We then applied the enhancement pipeline described in Section 4.2.1.

5.1.2 Fine-tuning Configuration. We employed Low-Rank Adaptation (LoRA) [6] as the primary parameter-efficient fine-tuning method, and additionally evaluate Weight-Decomposed Low-Rank Adaptation (DoRA) [12] as an alternative. DoRA extends LoRA with weight decomposition and has been shown to consistently outperform it across LLM and VLM tasks [12], and its inclusion allows us to assess whether these gains transfer to caption quality in the underwater domain.

The SmolVLM backbone was evaluated across three LoRA ranks (r = 8, 16, 32) with DoRA enabled and disabled, yielding a total of six configurations. For all configurations, we used an alpha of 16 and a dropout rate of 0.05. The vision encoder was frozen during training to preserve pre-trained visual features.

5.1.3 Training Hyperparameters. All models were fine-tuned for 3 epochs using the 8-bit Paged AdamW optimizer with a learning rate of 2e-4 and a cosine learning rate scheduler with 100 warmup steps. We used a per-device batch size of 2 with 8 gradient accumulation steps, resulting in an effective batch size of 16. Finetuning and experiments were conducted on an NVIDIA A100 40GB using Flash Attention 2. The following custom system prompt was used during training for all configurations: *"You are a VLM attached to an underwater robot. Describe the scene. Be concise but thorough."*

5.2 Ablation Study - Results

Following the ablation setup described in Section 4.3, Table 3 reports all six SmolVLM-Instruct configurations across LoRA ranks (8, 16, 32) and DoRA settings. The best-performing configuration is **LoRA32 with DoRA enabled**, achieving the highest scores across all metrics with a BLEU-4 of 0.4332 and CIDEr of 1.2810.

Table 3: SmolVLM-Instruct full ablation results. Best results are in bold.

Config	B-1	B-2	B-3	B-4	R-L	C
LoRA8 DoRA=N	0.7710	0.6300	0.5139	0.4137	0.6466	1.2475
LoRA8 DoRA=Y	0.7760	0.6339	0.5160	0.4148	0.6419	1.2449
LoRA16 DoRA=N	0.7651	0.6256	0.5106	0.4103	0.6398	1.2234
LoRA16 DoRA=Y	0.7611	0.6240	0.5116	0.4126	0.6358	1.1988
LoRA32 DoRA=N	0.7398	0.6027	0.4931	0.3979	0.6307	1.2563
LoRA32 DoRA=Y	**0.7820**	**0.6474**	**0.5327**	**0.4332**	**0.6574**	**1.2810**

The best-performing configuration (LoRA32, DoRA=Y) was further evaluated with and without the image enhancement pipeline, as shown in Table 4.

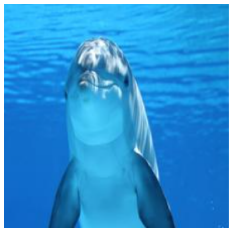
As shown in Table 4, the model without image enhancement outperforms the version with enhancement across all metrics, with BLEU-4 improving from 0.4332 to 0.4511 and CIDEr from 1.2810 to 1.3316. This result is counterintuitive, as image enhancement was designed to improve visual clarity in underwater imagery through color correction and contrast adjustment. One possible explanation is that the finetuned model has learned to extract relevant semantic features directly from raw underwater images, and the enhancement may alter or suppress low-level visual cues that the model relies on for caption generation. Another possibility is that the training data was not enhanced, so the distribution shift introduced by image enhancement at inference time negatively affects model performance. Nevertheless, even without image enhancement, the differences across all metrics remain relatively small, with a BLEU-4 gap of 0.0179 and a CIDEr gap of 0.0506. Notably, both configurations still exceed the best-performing metrics reported by Huanyu et al. on the same dataset [8].

Table 4: Effect of image enhancement pipeline on LoRA32 DoRA=Y. Best results are in bold.

Config	B-1	B-2	B-3	B-4	R-L	C
Pipeline ON	0.7820	0.6474	0.5327	0.4332	0.6574	1.2810
Pipeline OFF	**0.7909**	**0.6583**	**0.5472**	**0.4511**	**0.6661**	**1.3316**

This suggests that for datasets where raw images are already of sufficient clarity, image enhancement may offer limited benefit and could even be detrimental to caption quality, possibly due to the distribution shift introduced at inference time. Table 5 shows examples of raw versus enhanced images produced by the pipeline.

Table 5: Visual examples of ATHENA outputs on challenging underwater images.

	Raw Input	Enhanced Output
Sample 1		
Sample 2		

5.3 Benchmark Comparison

We compared our ATHENA pipeline against three representative image captioning approaches: a CNN-RNN model (Bi-LSTM [19]), an attention-based model (Up-Down [3]), and a Transformer-based

model (S2 Transformer [21]), as well as UICM-SOFF [8], the best-performing model reported by Huanyu et al. All models were evaluated on the same underwater image captioning dataset [8], providing a consistent benchmark for underwater scene captioning. Results are shown in Table 6.

These model categories are relevant to underwater image captioning because they address different challenges in visual understanding, spanning from CNN-RNN baselines that caption sequentially, to attention-based models that focus on salient regions, to transformers that capture global image relationships.

Table 6: Experimental results of ATHENA and existing image captioning models on the underwater dataset. B-N, R-L, and C denote BLEU-N, ROUGE-L, and CIDEr, respectively. Best results are in blue and second-best are underlined.

Model	B-1 ↑	B-2 ↑	B-3 ↑	B-4 ↑	R-L ↑	C ↑
Bi-LSTM [19]	0.6813	0.5554	0.4534	0.3661	0.5661	1.0747
Up-Down [3]	0.7064	0.5807	0.4653	0.3585	0.5615	1.1680
S2 Transformer [21]	0.7222	0.5958	0.4877	0.3938	0.5944	1.2481
UICM-SOFF [8]	0.7249	0.5980	0.4920	0.3953	0.5888	1.2831
ATHENA (Ours)	0.7909	0.6583	0.5472	0.4511	0.6661	1.3316

Table 6 shows that ATHENA outperforms all compared models across BLEU, ROUGE-L, and CIDEr metrics. The reported result corresponds to SmolVLM fine-tuned with LoRA rank 32 and DoRA enabled, with prior model results taken from Huanyu et al. [8]. Bootstrap confidence intervals ($n = 1000$) on ATHENA's best-performing configuration yielded BLEU-4: 0.4503 ± 0.0094, ROUGE-L: 0.6656 ± 0.0063, and CIDEr: 1.3372 ± 0.0381, confirming that the reported improvements over prior methods are consistent and exceed random variation.

5.4 Computational Efficiency and Deployment Analysis

To validate the feasibility of ATHENA's deployment on resource-constrained AUV hardware, we evaluated its computational footprint and inference latency. We initially deploy it as a *FastAPI* server with 4-bit quantization using the *bitsandbytes* library.

Finetuning and experiments were conducted on an NVIDIA RTX 5060 Ti 16GB using Flash Attention 2 to simulate a modern embedded GPU environment. As shown in Table 7, the quantized model consumes only 604 MB, which is critical since it falls within the constraints of standard edge computing modules. For example, the NVIDIA Jetson Orin series typically offers about 8 GB of memory. This leaves sufficient headroom to support system overhead, including the configurable option to store captured images locally without impacting performance.

The image preprocessing pipeline is also highly efficient, requiring only 0.03s per frame. Caption generation takes about 14s, which is acceptable for underwater acoustic communication scenarios, where data transmission rates are the primary bottleneck.

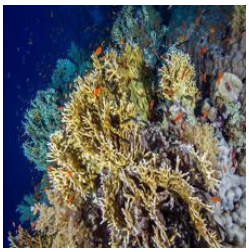
Table 7: System Resource Utilization and Latency Analysis

Metric	Value
Total Parameters	1.27 B
VRAM fp16	1.49 GB
VRAM 4-bit (Active)	604 MB
Preprocessing Latency	0.03 s
Inference Latency	< 14.00 s

5.5 Failure Case Analysis

To identify conditions under which ATHENA degrades, we selected the worst-performing test image based on CIDEr score, which also exhibited near-zero BLEU-4 and low ROUGE-L, confirming consistency across metrics. As shown in Table 8, ATHENA produced a generic background description, missing the primary subject entirely; a recurring pattern when subjects are small or distant.

Table 8: Worst-performing test image by CIDEr score

Image	Prediction	Ground Truth (2/5)
	There are many colorful corals growing on the seabed.	GT-1: There is a group of golden small fish playing among the coral bushes. GT-2: There is a group of golden small fish swimming around the coral.
BLEU-4: ≈0.000 ROUGE-L: 0.255 CIDEr: 0.016		

6 CONCLUSION

In this work, we presented ATHENA, an edge-capable underwater scene captioning pipeline that integrates a seven-step image enhancement pipeline with a parameter-efficiently fine-tuned vision-language model for real-time onboard deployment on AUVs. Through a systematic ablation study across three LoRA ranks and DoRA configurations, we identified SmolVLM with LoRA rank 32 and DoRA enabled as the optimal configuration, achieving the highest scores across all captioning metrics. When benchmarked against a wide range of state-of-the-art CNN-RNN, attention-based, and Transformer-based models on a standard underwater image captioning dataset, ATHENA demonstrated superior performance across BLEU, ROUGE-L, and CIDEr metrics. Under 4-bit NF4 quantization, SmolVLM requires only 604 MB of active VRAM and achieves inference in under 14 seconds, confirming its feasibility for deployment on edge hardware such as the NVIDIA Jetson Orin series.

Despite these advances, the dataset used for training and evaluation, while the most comprehensive publicly available underwater

captioning dataset, remains limited in scale and environmental diversity compared to terrestrial benchmarks. Expanding this dataset to cover a wider range of underwater conditions, species, and geographic regions would likely improve both caption quality and generalization. Additionally, while the enhancement pipeline adapts to per-image color distributions through attenuated channel compensation and white balance correction, its fixed step sequence may not generalize optimally to extreme conditions such as highly turbid coastal waters or near-zero visibility deep-sea environments outside the training distribution. Future work will focus on collecting and annotating larger underwater captioning datasets, investigating environment-adaptive enhancement strategies, and validating ATHENA on physical AUV hardware in untethered deployment scenarios.

REFERENCES

[1] Maximilian Adang, JunEn Low, Ola Shorinwa, and Mac Schwager. 2025. SINGER: An Onboard Generalist Vision-Language Navigation Policy for Drones. arXiv:2509.18610 [cs.RO] https://arxiv.org/abs/2509.18610

[2] Derya Akkaynak and Tali Treibitz. 2019. Sea-thru: A Method For Removing Water From Underwater Images. *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (04 2019).

[3] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6077–6086.

[4] Dana Berman, Deborah Levy, Shai Avidan, and Tali Treibitz. 2020. Underwater Single Image Color Restoration Using Haze-Lines and a New Quantitative Dataset. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP, 99 (2020), 1–1. doi:10.1109/TPAMI.2020.2977624

[5] Ansar Hani, Najiba Tagougui, and Monji Kherallah. 2026. Toward Human-Level Understanding: A Systematic Review of Vision-Language Models for Image Captioning. *The International Arab Journal of Information Technology* 23, 1 (2026). doi:10.34028/iajit/23/1/8

[6] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. In *Proceedings of the International Conference on Learning Representations*.

[7] Chongyi Li, Chunle Guo, Wenqi Ren, Runmin Cong, Junhui Hou, Sam Kwong, and Dacheng Tao. 2019. An Underwater Image Enhancement Benchmark Dataset and Beyond. *IEEE Transactions on Image Processing* (2019). doi:10.48550/arXiv.1901.05495 arXiv:1901.05495 [cs.CV].

[8] Huanyu Li, Hao Wang, Ying Zhang, Li Li, and Peng Ren. 2025. Underwater image captioning: Challenges, models, and datasets. *ISPRS Journal of Photogrammetry and Remote Sensing* 220 (2025), 440–453. doi:10.1016/j.isprsjprs.2025.04.001

[9] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. Hoi. 2023. BLIP-2: Bootstrapping Language-Image Pre-Training with Frozen Image Encoders and Large Language Models. In *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*. 19730–19742. Article No. 814, Published 23 July 2023.

[10] Wannian Li and Fan Zhang. 2025. Real-Time Vision–Language Analysis for Autonomous Underwater Drones: A Cloud–Edge Framework Using Qwen2.5-VL. *Drones* 9, 9 (2025), 605. doi:10.3390/drones9090605

[11] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*. Association for Computational Linguistics, Barcelona, Spain.

[12] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. DoRA: Weight-Decomposed Low-Rank Adaptation. In *Proceedings of the 41st International Conference on Machine Learning*.

[13] V. Lopez-Vazquez, J.M. Lopez-Guede, D. Chatzievangelou, et al. 2023. Deep learning based deep-sea automatic image enhancement and animal species classification. *Journal of Big Data* 10 (2023), 37. doi:10.1186/s40537-023-00711-w

[14] Andrés Marafioti et al. 2025. SmolVLM: Redefining small and efficient multimodal models. (2025). arXiv:2504.05299 [cs.AI] https://arxiv.org/abs/2504.05299

[15] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Pierre Isabelle, Eugene Charniak, and Dekang Lin (Eds.). Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 311–318. doi:10.3115/1073083.1073135

[16] Liam Paull, Guoquan Huang, Mae L. Seto, and John J. Leonard. 2015. Communication-constrained multi-AUV cooperative SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. 509–516. doi:10.1109/ICRA.2015.7139227

[17] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based Image Description Evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4566–4575.

[18] Hao Wang, Shixin Sun, and Peng Ren. 2023. Meta underwater camera: A smart protocol for underwater image enhancement. *ISPRS Journal of Photogrammetry and Remote Sensing* 195 (2023), 462–481. doi:10.1016/j.isprsjprs.2022.12.007

[19] Y. Wang, J. Xu, and Y. Sun. 2022. End-to-End Transformer Based Model for Image Captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 2585–2594.

[20] Siaw-Lang Wong, Yong-Poh Yu, Nina Ho, and Raveendran Paramesran. 2014. Comparative analysis of underwater image enhancement methods in different color spaces. 034–038. doi:10.1109/ISPACS.2014.7024420

[21] Peng Zeng, Hong Zhang, Jing Song, and Lei Gao. 2022. S2 transformer for image captioning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, Luc De Raedt (Ed.). International Joint Conferences on Artificial Intelligence Organization, 1608–1614.

[22] Ying Zhang et al. 2025. An underwater acoustic semantic communication approach to underwater image transmission. *Intelligent Marine Technology and Systems* 3, 1 (Feb 2025). doi:10.1007/s44295-025-00054-7

Towards More Empathic Programming Environments

An Experimental Empathic AI-Enhanced IDE

Justin Rainier Go
justin_rainier_go@dlsu.edu.ph
De La Salle University
Manila, Philippines

Roemer Gabriel Caliboso
roemer_caliboso@dlsu.edu.ph
De La Salle University
Manila, Philippines

Kurt Christian Andaya
kurt_andaya@dlsu.edu.ph
De La Salle University
Manila, Philippines

Aaron Daniel Go
aaron_daniel_go@dlsu.edu.ph
De La Salle University
Manila, Philippines

Jocelynn Cu
jocelynn.cu@dlsu.edu.ph
De La Salle University
Manila, Philippines

Abstract

As generative AI becomes integral to software development, the risk of over-reliance and diminished critical thinking grows. This study introduces "Ceci," our Caring Empathic C IDE designed to support novice programmers by prioritizing learning and emotional support over direct code generation. The researchers conducted a comparative pilot study between Ceci and VSCode + ChatGPT [9, 40]. Participants completed a coding task and were evaluated using the NASA-TLX workload assessment and a post-test usability survey. Although the sample size was small ($n = 11$), results show that there is no significant difference in perceived effectiveness, learning and workload between the Experimental Ceci group and the Control group, though Ceci users reported significantly greater perceived helpfulness in error correction ($p = 0.0220$). These findings suggest that empathic responses may not be sufficient on their own to enhance the learner's outcomes, perceptions, or reduce workload. Overall, this study provides a foundational framework for future research. Such research should explore larger sample sizes, diverse programming tasks, and additional empathic features to better understand the potential of empathic programming environments in supporting novice programmers; they must also ensure that the empathic features are well-integrated in the user interface.

CCS Concepts

• **Applied computing** → **Interactive learning environments**; **Computer-assisted instruction**; • **Human-centered computing** → *Interactive systems and tools*; • **Software and its engineering** → **Integrated and visual development environments**; • **Computing methodologies** → *Natural language processing*.

Keywords

empathic computing, affective computing, programming environments, novice programmers, intelligent tutoring systems, generative AI, large language models, NASA-TLX, C programming

1 Introduction

1.1 Background of the Study

The field of artificial intelligence (AI) is rapidly permeating all aspects of software development. Coding assistants and large language models (LLMs) like Gemini, Copilot, and ChatGPT are now being used not just for generating code but also for modernizing legacy codebases through code migration, ensuring the security of code bases through API misuse detection, and enforcement of best practices through code review [29, 30, 43]. Despite this, AI-assisted software development is still limited by its inability to generate entire codebases from scratch, refactor existing code while ensuring it passes its own generated tests, and can even hinder innovation in the field [10, 15, 36]. These rapid advances in AI for professional development raise the question of how similar tools might support learning and teaching in programming education.

As artificial intelligence becomes an increasingly integral part of our world, its potential to create personalized learning experiences is a significant benefit [23]. In programming education specifically, students already use AI tools to debug code, understand concepts, and generate frameworks [14]. However, simply providing technical assistance is not enough. While empathy is a well-established positive trait in human educators, its integration into AI (and the numerous effects of doing so) remains a critical area of investigation [2].

Given that the debugging process is a well-documented source of frustration for novice programmers [3], there is an opportunity to explore how AI systems can go beyond correctness and efficiency to also support learners emotionally. This paper addresses that challenge by exploring the synergy between empathic systems and programming education through the design of an empathic debugger.

Furthermore, unlike prior work that primarily focuses on improving code generation or debugging accuracy, this study investigates the role of affective scaffolding in programming environments. Rather than optimizing correctness, the goal is to examine whether emotional support can influence cognitive load, frustration, and perceived learning outcomes.

By shifting the focus from performance-centric AI assistance toward learning-oriented AI design, this work contributes to the emerging area of empathic computing in software engineering education.

1.2 Research Questions

The main problem of this study is to determine how the use of an empathic AI chatbot interface during debugging or compilation affects novice programmers' frustration levels and learning outcomes in introductory programming.

Specifically, the study aims to answer the following questions:

(1) What are novice programmers' perceptions of the effectiveness of an empathic AI chatbot during coding and code compilation tasks?
(2) To what extent does interaction with an empathic AI chatbot enhance novice programmers' understanding of fundamental programming concepts?
(3) How effective is the empathic AI chatbot in reducing cognitive load and frustration among novice programmers?

2 Review of Related Literature

2.1 Frustrations for Novice Programmers

Research indicates that learning to program is a deeply emotional experience for novices, with frustration often cited as a critical barrier to success. It is easier for these students to get stuck and reach states of confusion and frustration than it is going back to being engaged.

Categories of frustration as categorized by Ford and Parnin [11] stems from plenty of factors intrinsic to computer science such as a misunderstanding of the mapping between behavior and cause, learning new programming technologies, adjusting to a new project environment, unavailability of quality and beginner-friendly documentation, as well of some psychologically intrinsic categories such as fear of failure and internal pressure. Some participants also noted some external variables which lead to frustration such as a lack of time and proper peers to learn together with. Some of this frustration stems from encountering persistent errors or conceptual impasses during coding [3]. Furthermore, factors outside the code itself, such as poorly designed interfaces or confusing lesson plans add an unnecessary mental burden for students, which is a strong predictor of frustration [33].

This issue has evolved with modern tools; for instance, recent behavioral studies show that novice programmers spend significant time and cognitive effort in verifying and debugging code generated by AI assistants like Github Copilot, which can itself be a new source of frustration [31].

Crucially, these feelings have significant consequences: unresolved frustration can lead to boredom and disengagement, which are directly linked to poorer performance and learning outcomes. Therefore, designing learning environments that can recognize and mitigate these negative emotional states is essential to support novice programmers and improve their chances of success.

2.2 Conversational Agents in Programming Education

The field of computer science remains ever growing, and with it come an influx of ever more students that may end up experiencing these aforementioned challenges and frustrations. With the advent of AI and conversational agents, various chatbots have been made or utilized to assist in programming education.

For example, Groothuijsen et al. [14] performed a case study on students and teachers who have previously used ChatGPT and found that students mostly used it for explaining, generating, and debugging code alongside conceptual understanding of course material. In a certain programming assignment, students treated ChatGPT as a tutor to explain code rather than as a tool to generate code, contrary to the expectations of their instructor. Groothuijsen et al. [14] found no connection between AI use and student performance, but do point out that the two highest scorers did not use ChatGPT and admit that the low sample size may be unsuitable for extracting such connections. Although students reported that ChatGPT improved their learning, their instructor lamented that their output quality and perceived learning was lower than that of previous year and suggested that ChatGPT could have decreased student's time investment in the assignment and thus lessened their programming experience.

CS50, an introductory programming course by Harvard University, developed their own "duck debugger" to approximate a 1:1 teacher-to-student ratio [27]. They found that simply using LLMs based on Generative Pre-trained Transformers (GPT) may circumvent the learning process as they are too helpful and may outright give solutions rather than act as an instructor – even if prompted to do otherwise. To this end, Liu et al. [27] proposed various solutions such as integrating human feedback and providing sample interactions in the system prompt and in a curated dataset which would be used to align the duck debugger towards the pedagogical goals of the course. These solutions demonstrated improved model behavior and alignment with the aforementioned pedagogical goals.

Phanudom et al. [35] considered responsiveness, scalability, and coverage as key issues that motivated their development of an interactive chatbot for an online Python course. In a similar vein, Hobert [18]'s "Coding Tutor" was also successfully developed to "take over tasks of teaching assistants in times when no human teaching assistant or lecturer is available for help, e.g. due to resource constraints." Carreira et al. [4], who developed "Pyo", a chatbot that was also meant to facilitate personalized assistance in an online introductory programming class. Pyo provided exercise assistance, error guidance, and concept definitions. Meanwhile, Phanudom et al. [35] specifically targeted error guidance, developing "Er-Bot", a chatbot that responds to error messages with targeted advice; and cited previous research on enhanced compiler error message generation that gave programmers advice when encountering runtime errors in Python code.

Given the wealth of literature, it is evidently clear that there is significant interest in augmenting programming education with conversational agents so that every individual student may receive a personalized education in an era where students vastly outnumber their instructors, teaching assistants, and teaching fellows. There are various weaknesses that have to be accounted for, such as overly-helpful responses and misalignment with pedagogical goals, but there have also been multiple approaches to solve these and to implement these conversational agents.

2.3 Empathic Approaches and Frameworks

In an educational context, empathy is essential for fostering positive improvements in student motivation and performance [12, 13]. As a concept, it involves understanding another's mental state (cognitive empathy), and being affected by it (affective empathy) [44]. Translating these pedagogical principles into technology is the focus of empathic conversational systems, a field of AI research dedicated to building chatbots that can recognize and respond appropriately to user emotions [37].

While the potential for AI in education is vast, a key challenge is bridging the gap between a system's technical capabilities and the nuanced emotional intelligence of a human educator [6], especially as current tools can foster over-reliance by providing direct solutions rather than teaching users step-by-step [16, 22].

In the context of debugging, the goal of an empathic approach is not just to provide emotional support, but also to mitigate the negative effects of frustration on cognitive performance, thereby helping the student regain the mental clarity needed to solve the technical problem at hand. In fact, recent empirical studies have validated this approach: Ortega-Ochoa et al. [34] found that an empathic chatbot providing affective feedback significantly improved student's motivation and self-regulation while they were developing computer competencies.

Knowing this, studies explored the possibility of modeling and improving upon the empathic aspect of artificial intelligence. Castellano et al. [5] outlined key points that contribute to the success of empathic tutors; for example, the understanding of the learner's affective state, applying and adjusting accordingly and striving to form a social bond with the learner. Moreover, Sun et al. [41] stated that student or learner engagement is improved through the integration of empathic responses or approaches through emotional recognition and corresponding adaptation.

Recent work in this field has produced advanced technical frameworks like AIVA, which integrates multimodal emotion detection with LLM-driven responses to create a general-purpose virtual companion [24]. This demonstrates the technical feasibility of our approach, which aims to apply similar principles specifically to the domain of novice programmer debugging.

Across related domains, evidence suggests modest affective benefits: empathic agents reduce frustration [19], pedagogical agents slightly lower cognitive load [25], and adaptive feedback increases motivation and persistence [28]. While individually limited (they have small sample sizes), these collectively point to the potential of empathic, adaptive systems to sustain learner engagement.

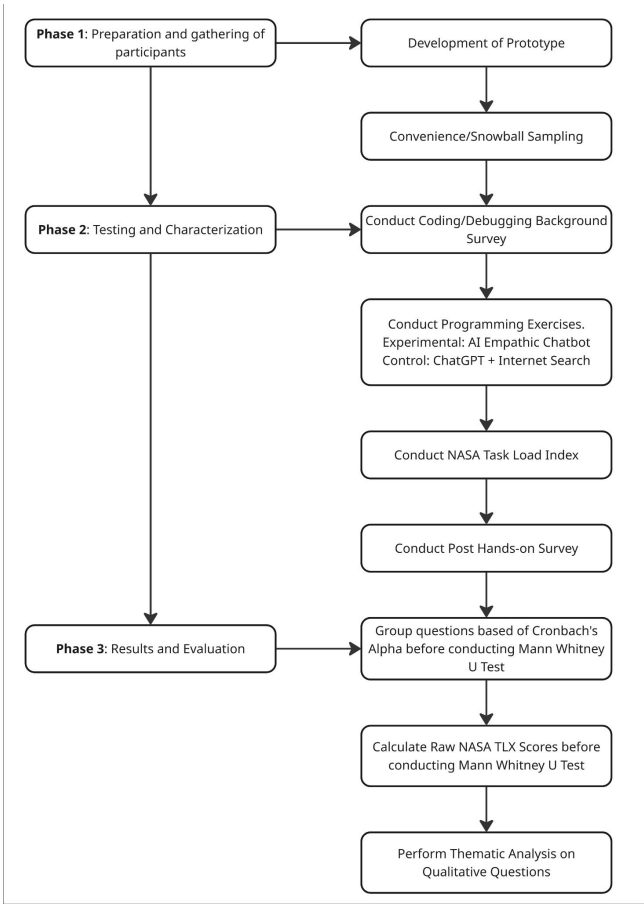


Figure 1: Schematic Diagram of Methodology

2.4 Gaps in Knowledge and Research Directions

Despite its potential, empathy can have counterproductive effects if not carefully balanced. Empathy is not always a positive force in teaching, as it can be susceptible to bias and may even lead to counterproductive learning patterns. For example, students that are praised by tutors for their successes may be led to choose easier tasks as opposed to learning newer and more complex topics in fear of losing that praise [21, 32].

Research exploring the link between teacher empathy and student outcomes, such as classroom behavior and academic performance, has yielded mixed results. A major reason for this inconsistency is the lack of standardized assessment criteria across many of these studies, which casts doubt on the reliability of their findings [2].

While the literature shows a clear trend towards using conversational agents in programming education and a growing interest in developing empathic AI tutors, a specific gap exists at the intersection of these fields. Current research often focuses on conceptual help or general tutoring, with less emphasis on mitigating the in-the-moment frustration a novice programmer would feel when facing a cryptic compiler error. Furthermore, while studies explore empathic responses, few have investigated the use of visual, non-verbal cues, such as an animated agent with different emotional poses, to deliver this feedback in a debugging context.

Therefore, our project aims to address this gap by designing and evaluating a system that provides empathic feedback specifically tailored to GCC compiler errors. This study focuses on CCPROG1 (Logic Formulation and Introductory Programming), the foundational introductory programming course required for all computer science majors at De La Salle University, Manila, Philippines. This course's reliance on the C language and the GCC compiler, known for its cryptic error messages, makes it an ideal context to test our primary goal: reducing frustration and improving the learning experience for novice programmers.

3 Methodology

This study employs a structured experimental methodology designed to address the research questions stated. The experiment will be done by dividing the procedure into three phases as seen in the Schematic Diagram shown in Figure1.

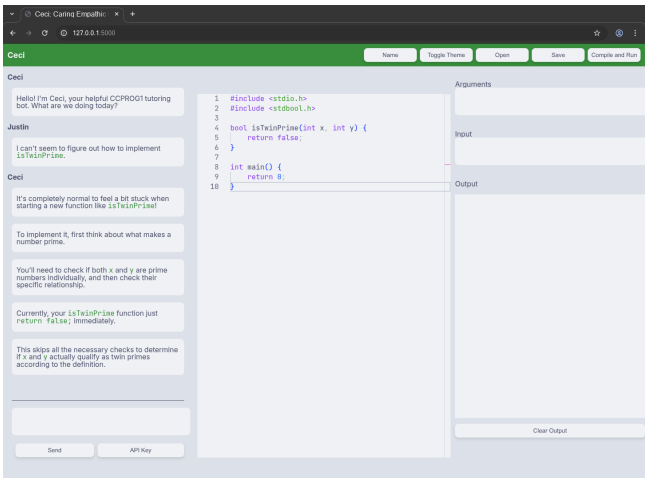


Figure 2: Prototype Interface Design

3.1 Preparation and Participant Plan

The first step in the methodology is the preparation of necessary components for the study such as the prototype itself and the participants.

3.1.1 Prototype Design. The prototype, named Ceci, is an Integrated Development Environment (IDE) run locally through Python Flask. Within the IDE there would be three sections: the AI chatbot interface, the coding interface, and the terminal as shown in Figure 2.

In this interface, the user may chat to prompt the chatbot as well that will give the appropriate empathic response. The design is somewhat inspired from the study of Hobert [18], where they developed a coding tutor with a similar interface. Aside from a standard chat interface, code compilation also triggers the chatbot to respond to the result of the compilation.

The AI chatbot was designed to query Gemini AI through a specific and carefully engineered system prompt to constrain the chatbot behavior to reflect correctness and empathy. To ensure reproducibility and clarity, the system prompt was structured using explicit rule-based constraints. Specifically, Ceci was instructed to:

- avoid directly providing complete code solutions,
- prioritize step-by-step guidance through hints,
- include an affective acknowledgement before technical explanation, and
- limit responses to short, structured outputs consisting of three parts: emotional acknowledgement, guided suggestion, and technical explanation.

For example, when encountering a compilation error, Ceci first acknowledges potential frustration (e.g., "In C, `for` loops have a very specific structure."), followed by a hint (e.g., "Take a close look at how the different parts inside the parentheses are separated."), and then a concise explanation of the compiler message (e.g. "the error 'expected ";" before "i"' indicates that C expects a semicolon after the loop's condition (`i <= n`) before it sees the increment part (`i++`)").

These constraints were implemented entirely through prompt engineering rather than fine-tuning, ensuring controllability but also limiting adaptability.

3.1.2 Sampling Method. The criteria for participation of the study requires the samples to be novice programmers; consequently, the researchers will employ a convenience sampling technique targeting De La Salle University students taking CCPROG1. Additionally, a screening test would be given to gauge and filter out participants with the intention of keeping the sample to be of purely novice programmers.

3.1.3 Materials. The participants will be given a programming exercise to work on. The exercise is a hands-on type activity where they would implement a program from scratch. The program the participants were tasked to design is a twin prime identifier within a range of integers.

3.2 Testing and Characterization

Once the preparation is completed, the participants will proceed to the experimentation phase, during which the designed exercises and assessments will be conducted.

3.2.1 Pre-registration. Participants first pre-register for the experiment through signing up via Google Forms. Aside from the participants' contact details, the form also asks participants to rate their own proficiency in the C programming language through a series of Likert scale questions (the aforementioned screening test in Section 3.1.2).

3.2.2 Experimental Design. The experiment will employ a between groups design following a two group structure consisting of an experimental group, and control group. The independent variable from these groups will be the technique each will use to debug, where the experimental group will be designated to the AI chatbot interface developed for this study, the control group will have no custom interface and will code with access to ChatGPT [20, 26]. To address possible confounding variables, all participants will be given the same programming exercise, and the same physical environment as well. Besides these, any tests conducted before and after will also be identical.

3.3 Results and Evaluation

Following the experimentation proper, the researchers will proceed with data collection and evaluation.

3.3.1 NASA Task Load Index. NASA Task Load Index (NASA-TLX) is an assessment tool that measures perceived workload or frustration [17]. The NASA-TLX consists of six subscales that represent the total workload of the participant. These subscales are Mental Demand, Physical Demand, Temporal Demand, Performance, Effort, and Frustration. NASA-TLX is widely used in various areas such as medical, engineering and even among human-chatbot interaction [7, 39]. For the purposes of this study, NASA-TLX will be used to quantify subjective perception of workload among participants debugging with the three techniques on each group.

3.3.2 Post Hands-on Survey. A questionnaire will be administered to the participants after their participation in the hands-on section

Table 1: Mann-Whitney U Test for NASA TLX Raw Scores

Statistic	Value
Significance level (α)	0.05
Mann-Whitney U	16
P-Value	**0.9271**

of the study. The questionnaire will contain both open ended and Likert scale type questions designed to answer the research questions. The open ended questions will be analyzed through thematic analysis to analyze qualitative data [1]. This will also be done using inductive coding where themes naturally emerge from the short essay responses. The Likert scale questions on the other hand will be analyzed through grouping using Cronbach's alpha coefficient, a formula that measures how reliable a set of questions when grouped together making up an underlying construct [8].

The formula for Cronbach's alpha is given by Equation 1:

$$\alpha = \frac{k}{k-1}\left(1 - \frac{\sum_{i=1}^{k} \sigma_{y_i}^2}{\sigma_x^2}\right) \tag{1}$$

where k is the number of items or questions, $\sigma_{y_i}^2$ is the variance per item and σ_x^2 is the variance of the total scores [38]. The scores of the participants will be averaged for their average score per category. Results from this averaging and the NASA-TLX will then be analyzed using Mann-Whitney U Test, a nonparametric test to compare two independent groups [42].

3.4 Ethical Considerations

Prior to participation, a consent form will be obtained from all selected participants following the sampling process. Each participant will be informed on the purpose, procedures, and scope of the study to ensure their participation. Participants have the option to withdraw from the study at any stage without penalty or consequence. Furthermore, all personal and research-related data will be treated with strict confidentiality and used solely for academic purposes. The study will adhere to the Data Privacy Act of 2012, ensuring that identifiable information remains secure and anonymous throughout the study.

4 Results and Analysis

4.1 Experimentation Proper

The sampling method was able to secure a total of 11 participants, where they are split to 6 experimental group members and 5 control group members. The researchers utilized the Gokongwei Room 304A Computer Laboratory in DLSU to conduct the study, where the participants' environment remained consistent throughout the experiment (see Figure 3 and Table 1). The participants were given 30 minutes to attempt to complete the programming exercise. Directly after the hand-on session, the participants were given the NASA TLX and the questionnaire.

4.2 NASA TLX

The results of the NASA TLX are shown in Figure 3. The Raw Score is computed through the arithmetic mean of the individual metrics: The researchers perform a Mann-Whitney U Test Result for Raw

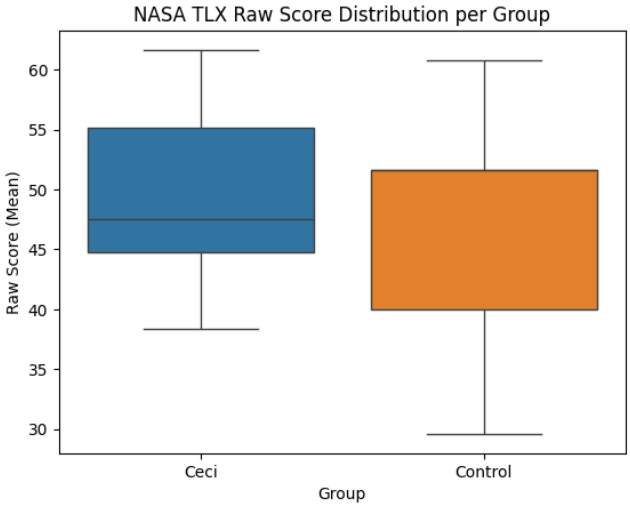


Figure 3: NASA TLX Results per Group. The Ceci group seems to experience a slightly higher task load index than the ChatGPT group, but the difference is not statistically significant as can be seen in Table 1.

Scores per Group, where the null hypothesis states that there is no significant difference between the Ceci and ChatGPT groups while the alternative states that there is a significant difference between the Ceci and ChatGPT groups. The p-value of 0.9271 is greater than the alpha of 0.05, so the researchers fail to reject the null hypothesis. That is, there is no significant difference between the task load index of the Ceci and ChatGPT groups.

4.3 Thematic Analysis

The researchers performed a thematic analysis to get insight on the respondents answers on the open-ended questions. Participants A1 to A6 are from the group that used Ceci, while participants B1 to B5 are from the group that used ChatGPT.

4.3.1 Coping Strategy. When coping with frustration in programming related tasks, Coping strategies for misunderstanding of a programming concepts among our participants include coping with internet and coping through AI (A2, A3, B4, B5), some use indirect methods of coping such as through taking a break (A3, B5), and some participants cope through tracing through the problem themselves (A3, B5).

Non-AI debugging strategies of the participants include referring to online resources such as documentation (A2, A5, B4), using online forums such as Stack Overflow (B1). AI strategies are used mostly as a last resort (A1, B2, B4).

4.3.2 Likes. Respondents that used ChatGPT had a degree of familiarity when using it to work through programming problems (B1, B2, B3) which provided a degree of comfort and reliability (B2, B3). Respondents also preferred the instant solutions that ChatGPT provided, and the reliability that comes with knowing that it can provide an answer (B4). Respondents that used Ceci liked the direct embedding of the AI with the IDE given that it was context

aware enough to generate a response on certain actions such as in compiling (A1, A3). Participants also found Ceci to be friendly and encouraging (A2, A4).

4.3.3 Dislikes. Respondents that disliked ChatGPT did not like that they directly copy-pasted some codes to accomplish the task which they viewed as making them overreliant (B1, B5). Some respondents noted that using ChatGPT was a lot slower than directly integrating and thinking of the code themselves (B2, B3). Respondents that disliked Ceci did not like some of the UI issues and technical problems such as the need for an API key (A3, A4, A6). Some respondents also disliked the fact that it wasn't descriptive enough to tackle some problems (A1).

4.3.4 Preference for Empathicness. Respondents show mixed feelings with regard to empathic assistance in the context of novice programming work. Respondents that do not prefer empathic assistance indicate that they prefer more straightforward and logical explanations detached from emotion (A1, A3). One participant cites that they prefer straightforward answers as they take less time to read (A5). Respondents that do prefer empathic assistance indicate that empathic responses put them in a good mood (A4) or the adaptive nature of empathic systems to a person who is learning, or who is in a rush or the like (B4).

4.4 Likert Scale Analysis

4.4.1 Grouping through Cronbach's Alpha. Using Cronbach's Alpha coefficient, the researchers were able to safely group some questions with a reliability of 0.7 or higher. The table below, Table 2 shows the computation and the categorization of the Likert scale questions. The first set of items is as follows:

- "Programming is more efficient with the setup I was in."
- "I was able to learn more during the hands-on session."
- "The tools assigned to me help improve my understanding of the code and concepts."

These items yield an alpha coefficient of 0.741 making it sufficiently reliable to be grouped for a common sentiment like effectivity of learning.

The second set is as follows:

- "The supportive messages helped improve my efficiency."
- "I feel comfortable and not as stressed with the programming setup I was in."
- "Debugging and programming this way does not induce that much hassle."

which produced a coefficient of 0.718. This group then can be categorized as comfort or lack of stress.

The last group had the following items:

- "I had less difficulty programming with my setup."
- "I heavily relied on the tools provided for me."
- "I have no problem getting the correct information from the tools I was assigned to."

and have a coefficient of 0.710 that allows the set to be considered as a group, where its category falls more on the perceived effectiveness of the setup. One question however, question 7 ("The feedback/information I received helped me correct some errors. ") did not fit reliably within the groups as its inclusion results to a coefficient alpha lesser than 0.7, nullifying the validity of a grouping. Thus, the researchers have decided to treat this question as a standalone item.

4.4.2 Mann-Whitney U Test Results. For each group, the answers of the participants are averaged for all questions. This results in an average score for a given category per participant. The resulting scores are then analyzed using Mann-Whitney U Test (see Table 3) to compare the similarity of the Ceci and Control group. The same is done with item 7, as it did not belong to a group, its analysis is isolated to just itself. For each group of questions, the p-value is greater than the significance level. This means that there is no significant difference in learning, comfort, and perceived effectiveness between the Ceci and ChatGPT groups.

The resulting p-value returned from the test is 0.02195 which is less than the significance level of the test. This suggests that the perceived helpfulness of the Ceci group is significantly greater than that of the control group.

4.5 Completion Rate

At the end of their hands-on the participants are tasked to submit their latest code, and they also were required to answer an item in the questionnaire asking if they completed the program or not. Of the 6 Ceci users, none of them were able to complete while 4 out of 5 of the Control group were able to finish. This suggests that the current iteration of Ceci may have been detrimental to task completion. This disparity is most likely attributable to the UI and the technical friction reported by Ceci participants – particularly the API key requirement flagged in A3, A4, and A6 – rather than a failure of the empathic approach itself. The control group's prior familiarity with ChatGPT (B1, B2, B3) likely reduced onboarding friction significantly, whereas Ceci users had to contend with a brand-new interface under a strict 30-minute time limit.

5 Discussion

5.1 Effectiveness during coding

The thematic analysis shows mixed preferences towards the effectiveness of an empathic AI chatbot in coding tasks. While some appreciated the empathic responses, others felt it was unnecessary towards actually getting the program debugged properly. These results suggest that learning may depend heavily on an individual level, as some students who prefer to receive encouragement and support may see improvement in using the empathic IDE while the students who prefer direct means of retrieving and obtaining information will benefit less from the empathic responses. This also raises the concern that empathic responses without sufficient technical depth risk feeling unhelpful or even patronizing to task-focused learners. This concern is also supported by participant A1's complaint that Ceci wasn't descriptive enough to tackle certain problems

These results are also empirically backed up through the Likert scale analysis for the "Perceived Effectiveness" group since the outcome of the analysis also indicates no significant difference in perceived effectiveness suggesting that there is no meaningful improvement between having an empathic IDE or not. However, the results from item 7 suggest that the Ceci group's perceived

Table 2: Reliability Analysis (Cronbach's Alpha)

	Learning	Comfort	Perceived Effectiveness
Items	2, 3, 6	5, 8, 9	1, 4, 10
Sum of Variances per Item	3.22	3.85	4.35
Number of items	3	3	3
Variance of Total Scores	6.36	7.40	8.25
Cronbach's Alpha	**0.741**	**0.719**	**0.710**

Table 3: Mann-Whitney U Test Results for Likert Scale Groups

Group/Item	Learning	Comfort	Perceived Effectiveness	Error Correction (Q7)
Significance level	0.05	0.05	0.05	0.05
Mann-Whitney U Statistic	12.5	11	15.5	25.0
P-Value	**0.7119**	**0.5180**	**1.0**	**0.0220**

helpfulness in resolving errors is greater than the Control group, indicating that the application was effective in assisting the students in resolving errors.

5.2 Extent of learning and understanding

The result of the test indicates that there is no significant difference between the Ceci and Control groups in learning new concepts and understanding code. This suggests that, regardless of the presence of an empathic aid, students' perceived learning remains the same. One possible explanation for this outcome is the mixed reception of empathic responses within the learning environment. Another factor that may have contributed is the strict time constraint of the hands-on activity; as highlighted in the thematic analysis, some participants preferred direct, straightforward answers that addressed their immediate needs.

5.3 Effectiveness in reducing frustration

The NASA-TLX Raw Score Mann-Whitney U Test result indicates that use of Ceci did not significantly affect task load. The similarity in task load indices for the two groups may be caused by separate problems in the set-up. For instance, a few responses from the Ceci group regarding what they disliked about the application was that the "AI is buggy" or that the "AI is unhelpful", while a different issue arose for the control group where they find the need to "to elaborate more or re-paste the code".

Thematic analysis matches the results in the pre-registration survey, with participants that prefer thorough analysis of program and implementation details finding less frustration compared to Ceci purposefully not giving out the full answer. The ChatGPT group additionally remarked about risks of overreliance on an LLM, which suggests deeper considerations around pedagogical alignment and tool design.

It is also worth noting that comparing Ceci, a first-iteration prototype against VSCode with ChatGPT (A pairing of two mature, widely-used tools) introduced an inherent confounding variable. UI frustrations specific to the prototype, such as the API key requirement and general instability, likely inflated the NASA-TLX frustration scores for the experimental group independently of the empathic AI features themselves. Future studies should ideally conduct usability testing on the prototype prior to comparative experiments, or use a similarly barebones control interface to ensure a fairer comparison.

Another key insight from this study is that empathy alone is insufficient in programming assistance contexts. Participants who preferred direct and concise technical explanations perceived empathic responses as less efficient or even distracting. This suggests that effective AI tutors must balance affective support with domain-specific depth, rather than treating empathy as a standalone feature. This finding highlights a design tension in empathic AI systems: increasing emotional support without sufficient technical grounding may reduce perceived usefulness, particularly for task-oriented users.

6 Conclusion and Future Work

6.1 Conclusion

The study's results and analysis reveal a nuanced picture. Notably, Ceci users reported significantly greater perceived helpfulness in error correction ($p = 0.0220$), suggesting that an empathic IDE may offer targeted benefits in specific aspects of the debugging experience. However, broader measures of perceived effectiveness, learning, and workload showed no significant difference between the Ceci and Control groups.

This entails that the inclusion of empathic responses does not substantially alter students' perceived effectiveness, learning, or frustration levels during coding tasks. This may be the case of the specific limitations of the study such as the unforeseen issues in the app implementation, the strict time constraints during the hands-on activity, and the differing individual preferences for either empathic assistance or direct assistance, all of which may have influenced how participants engaged with the system.

Furthermore, the mixed qualitative feedback suggests that empathic features may not provide the same benefits across all learners. While some students appreciated emotional support, others thought that the empathic layer is unnecessary when they want immediate straightforward responses for coding. These reactions may imply that empathic design may be more effective when applied to specific subgroups that prefer it.

Despite the absence of significant improvements in overall learning or workload, the findings provide important design implications for future AI-assisted programming environments. Specifically, they highlight the importance of balancing empathy with technical depth, minimizing interface friction in experimental tools, and aligning AI behavior with user expectations for efficiency versus support. As such, this work contributes not by demonstrating performance gains, but by identifying critical factors that influence the effectiveness of empathic AI in programming education.

6.2 Future Work

The study is bound by various time constraints, and future work enjoys a variety of avenues for improvement. First and foremost, future tests should ensure they test on a bigger sample of students. This study was only able to accrue around 11 participants as the testing period was near the finals week. More participants would enhance our analysis, especially our quantitative statistical tests. Furthermore, most of the criticism against Ceci was directed towards the UI/UX of the IDE. Improvements on this aspect would likely lead to future participants directing their criticism towards deeper, more technical aspects of the system rather than implementation issues.

References

[1] S. K. Ahmed, R. A. Mohammed, A. J. Nashwan, R. H. Ibrahim, A. Q. Abdalla, B. M. Ameen, and R. M. Khdhir. 2025. Using thematic analysis in qualitative research. *Journal of Medicine, Surgery, and Public Health* 6 (2025), 100198.

[2] K. Aldrup, B. Carstensen, and U. Klusmann. 2022. Is empathy the key to effective teaching? a systematic review of its association with teacher-student interactions and student outcomes. *Educational Psychology Review* 34, 1 (2022), 1177–1216.

[3] N. Bosch and S. D'Mello. 2015. The affective experience of novice computer programmers. *International Journal of Artificial Intelligence in Education* 27, 1 (2015), 181–206.

[4] G. Carreira, L. Silva, A. J. Mendes, and H. G. Oliveira. 2022. Pyo, a chatbot assistant for introductory programming students. In *2022 International Symposium on Computers in Education (SIIE)*.

[5] G. Castellano, A. Paiva, A. Kappas, R. Aylett, H. Hastie, W. Barendregt, F. Nabais, and S. Bull. 2013. Towards empathic virtual and robotic tutors. In *Lecture Notes in Computer Science*. 733–736.

[6] C. K. Y. Chan and L. H. Y. Tsi. 2023. The AI revolution in education: Will AI replace or assist teachers in higher education? (2023). arXiv:2305.01185

[7] L. Colligan, H. W. W. Potts, C. T. Finn, and R. A. Sinkin. 2015. Cognitive workload changes for nurses transitioning from a legacy system with paper documentation to a commercial electronic health record. *International Journal of Medical Informatics* 84, 7 (2015), 469–476.

[8] L. M. Collins. 2007. Research design and methods. In *Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. Psychometrika, 16(3):297–334.* Elsevier, 433–442.

[9] D. Deckker and S. Sumanasekara. 2024. *The role of ChatGPT in software development and code generation.* Wrexham University, Tech report.

[10] P. Eibl, S. Sabouri, and S. Chattopadhyay. 2025. Exploring the challenges and opportunities of AI-assisted codebase generation. (2025). arXiv:2508.07966

[11] D. Ford and C. Parnin. 2015. Exploring causes of frustration for software developers. (2015). doi:10.1109/chase.2015.19

[12] R. García-Pérez, J.-M. Santos-Delgado, and O. Buzón-García. 2016. Virtual empathy as digital competence in education 3.0. *International Journal of Educational Technology in Higher Education* 13 (2016).

[13] M. Goroshit and M. Hen. 2014. Does emotional self-efficacy predict teachers' self-efficacy and empathy? *Journal of Education and Training Studies* 2, 3 (2014).

[14] S. Groothuijsen, A. van den Beemt, J. C. Remmers, and L. W. van Meeuwen. 2024. *Ai chatbots in programming education.* Computers and Education: Artificial Intelligence, 7:100290.

[15] F. Gu, Z. Liang, H. Li, and J. Ma. 2025. The Matthew effect of AI programming assistants: A hidden bias in software evolution. (2025). arXiv:2509.23261

[16] R. Gupta, H. Goyal, D. Kumar, A. Mehra, S. Sharma, K. Mittal, and J. S. Challa. 2025. Sakshm AI: Advancing AI-assisted coding education for engineering students in india through socratic tutoring and comprehensive feedback. (2025). arXiv:2503.12479

[17] S. G. Hart and L. E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Advances in Psychology* 52 (1988), 139–183.

[18] S. Hobert. 2019. Say hello to "Coding Tutor"! design and evaluation of a chatbot-based learning system supporting students. (2019).

[19] Kate Hone. 2006. Empathic agents to reduce user frustration: The effects of varying agent characteristics. *Interacting with Computers* 18, 2 (2006), 227–245. doi:10.1016/j.intcom.2005.05.003

[20] I. Hossain, C. Hundhausen, A. Tariq, S. Haque, Y. Qiao, and B. Mulanda. 2025. The effects of GitHub Copilot on computing students' programming effectiveness, efficiency, and processes in brownfield programming tasks. (2025). doi:10.1145/3702652.3744219

[21] M. L. Kamins and C. S. Dweck. 1999. Person versus process praise and criticism: Implications for contingent self-worth and coping. *Developmental Psychology* 35, 3 (1999), 835–847.

[22] O. Kurniawan, E. Chandra, C. M. Poskitt, Y. Noller, K. Tsu, and C. Jegourel. 2025. Designing for novice debuggers: A pilot study on an AI-assisted debugging tool. (2025). arXiv:2509.21067

[23] L. Lasha, M. Grigolia, and L. Machaidze. 2023. Role of AI chatbots in education: systematic literature review. (2023).

[24] Chenxi Li. 2025. AIVA: An AI-based Virtual Companion for Emotion-aware Interaction. arXiv:2509.03212 [cs.CV] https://arxiv.org/abs/2509.03212

[25] H. Li, Z. Wang, L. Ding, J. Zhang, and G. Wang. 2025. The facts about the effects of pedagogical agents on learners' cognitive load: a meta-analysis based on 24 studies. *Frontiers in Psychology* 16 (2025).

[26] J. Liu, X. Tang, L. Li, P. Chen, and Y. Liu. 2023. Which is a better programming assistant? A comparative study between ChatGPT and stack overflow. (2023). arXiv:2308.13851

[27] R. Liu, J. Zhao, B. Xu, C. Perez, and D. J. Malan. 2025. Improving AI in CS50: Leveraging human feedback for better learning. *SIGCSE TS* 2025 (2025), 715–721.

[28] S. Marwan, G. Gao, S. Fisk, T. W. Price, and T. Barnes. 2020. Adaptive immediate feedback can improve novice programming engagement and intention to persist in computer science. In *Proceedings of the 2020 ACM Conference on International Computing Education Research*.

[29] V. May, D. Misra, Y. Luo, A. Sridhar, J. Gehring, and J. Silvio. 2025. Freshbrew: A benchmark for evaluating AI agents on java code migration. (2025). arXiv:2510.04852

[30] S. Mondal, C. K. Roy, H. Wang, J. Arguello, and S. Mathan. 2025. Can we trust the AI pair programmer? Copilot for API misuse detection and correction. (2025). arXiv:2509.16795

[31] H. Mozannar, G. Bansal, A. Fourney, and E. Horvitz. 2024. Reading between the lines: Modeling user behavior and costs in AI-assisted programming. (2024).

[32] C. M. Mueller and C. S. Dweck. 1998. Praise for intelligence can undermine children's motivation and performance. *Journal of Personality and Social Psychology* 75, 1 (1998), 33–52.

[33] E. Novak, K. McDaniel, and J. Li. 2023. Factors that impact student frustration in digital learning environments. *Computers and Education Open* 5 (2023), 100153.

[34] E. Ortega-Ochoa, J. Q. Pérez, M. Arguedas, T. Daradoumis, and J. Manuel. 2024. The effectiveness of empathic chatbot feedback for developing computer competencies, motivation, self-regulation, and metacognitive reasoning in online higher education. *Internet of Things* 25 (2024), 101101.

[35] P. Phanudom, T. Hirao, R. Gaikovina Kula, and H. Iida. 2021. Interactive chatbot for supporting students in online Python programming class. (2021).

[36] R. A. Poldrack, T. Lu, and G. Beguš. 2023. AI-assisted coding: Experiments with GPT-4. (2023). arXiv:2304.13187

[37] A. S. Raamkumar and Y. Yang. 2022. Empathetic conversational systems: A review of current advances, gaps, and opportunities. *IEEE Transactions on Affective Computing* (2022), 1–20.

[38] T. Raykov and G. A. Marcoulides. 2017. Thanks coefficient alpha, we still need you! *Educational and Psychological Measurement* 79, 1 (2017), 200–210.

[39] J. Schmidhuber, S. Schlögl, and C. Ploder. 2021. Cognitive load and productivity implications in human-chatbot interaction. arXiv / tech report. (2021).

[40] A. Silver. 2025. *Celebrating 50 Million Developers.* Microsoft.

[41] F. Sun, L. Li, S. Meng, X. Teng, T. Payne, and P. Craig. 2025. Integrating emotional intelligence, memory architecture, and gestures to achieve empathetic humanoid robot interaction in an educational setting. (2025). arXiv:2505.19803

[42] J. H. Sundjaja, R. Shrestha, and K. Krishan. 2023. *McNemar and Mann-Whitney U Tests.* StatPearls.

[43] M. Vijayvergiya, M. Salawa, I. Budiselić, D. Zheng, P. Lamblin, M. Ivanković, J. Carin, M. Lewko, J. Andonov, G. Petrović, D. Tarlow, P. Maniatis, and R. Just. 2024. AI-assisted assessment of coding practices in modern code review. arXiv / conference report. (2024).

[44] Z. Zhou. 2022. Empathy in education: A critical review. *International Journal for the Scholarship of Teaching and Learning* 16, 3 (2022).

Novel CUDA and Segment Tree-Accelerated Simulated Annealing Approaches to the Time-Dependent Knapsack Problem

Harvey Shawn Chua
De La Salle University
Manila, Metro Manila
harvey_chua@dlsu.edu.ph

Christian Asturiano
De La Salle University
Manila, Metro Manila
christan_asturiano@dlsu.edu.ph

Nicole Kate Uy
De La Salle University
Manila, Metro Manila
nicole_uy@dlsu.edu.ph

Roger Luis Uy
De La Salle University
Manila, Metro Manila
roger.uy@dlsu.edu.ph

ABSTRACT

The Time-Dependent Knapsack Problem (TDKP) extends the classical 0–1 knapsack problem by allowing items to consume capacity over contiguous time intervals, significantly increasing computational complexity and rendering many exact methods impractical for large instances. In this work, we present novel CUDA-accelerated and segment tree simulated annealing (SA) approaches for TDKP, together with a CPU baseline for comparison. To enable evaluation of candidate solutions, we employ time-point compression and difference-array techniques, combined with a penalty-based formulation that avoids costly feasibility checks during optimization. The proposed GPU implementation exploits thread-level parallelism by executing multiple independent annealing instances concurrently while cooperatively evaluating solution states within each instance, while the proposed segment tree implementation improves the time complexity of each annealing iteration of the CPU baseline implementation. Experimental results on randomly generated benchmarks show that the SA approach matches optimal Integer Linear Programming solutions for small to medium problem sizes and produces high-quality approximations for large instances. Empirical testing shows that the CUDA implementation achieves speedups of up to 11x over the CPU baseline for moderate problem sizes, while the segment tree implementation achieves speedups of up to 1395x, demonstrating the effectiveness of both implementation paradigms for large-scale, time-dependent resource allocation problems. Our source code is publicly available at: csc612-tdkp

CCS CONCEPTS

• **Computing methodologies** → **Parallel algorithms**; • **Theory of computation** → **Mathematical optimization**; **Design and analysis of algorithms**;

KEYWORDS

Time-Dependent Knapsack Problem, Simulated Annealing, Data Structures, CUDA, GPGPU

1 TIME-DEPENDENT KNAPSACK PROBLEM

We consider the time-dependent knapsack problem (TDKP) with irregular availability, introduced in [1]. TDKP generalizes the classic 0-1 knapsack problem by allowing items to consume capacity over a contiguous time interval rather than a single instant. Formally, there are n items, each item i is associated with a profit p_i, a weight w_i, and a time interval $[l_i, r_i]$. Selecting item i yields profit p_i, but requires allocating weight w_i at every $t \in [l_i, r_i]$.

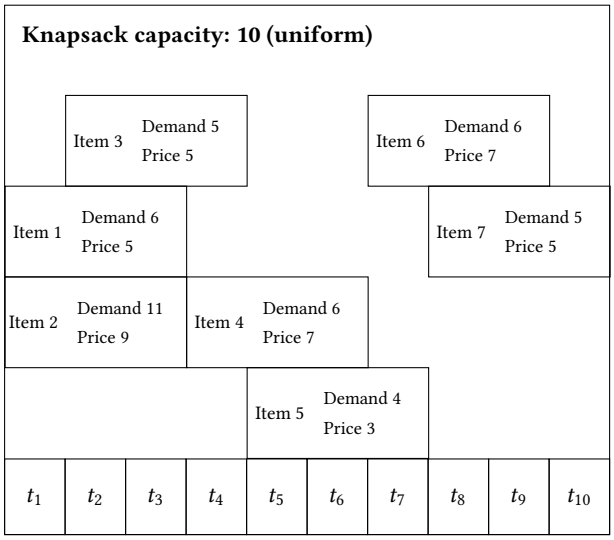


Figure 1: Instance of TDKP from [1]

We define the total weight allocated at time t as

$$D_t = \sum_{i=1}^{n} w_i \, x_i \, \mathbf{1}_{[l_i, r_i]}(t),$$

where $x_i \in \{0, 1\}$ indicates whether item i is selected, and $\mathbf{1}_{[l_i,r_i]}(t)$ is the indicator function

$$\mathbf{1}_{[l_i,r_i]}(t) = \begin{cases} 1 & \text{if } l_i \le t \le r_i, \\ 0 & \text{otherwise.} \end{cases}$$

The objective of TDKP is to select a subset of items maximizing total profit while respecting per-time capacities C_t:

$$\text{Maximize:} \quad \sum_{i=1}^{n} p_i x_i$$
$$\text{Subject to:} \quad D_t \le C_t \quad \forall t.$$

Figure 1 shows a sample instance of TDKP with uniform capacity $C_t = 10$. It can be shown that the optimal solution is to select items

1, 4, 5, and 6, with an overall profit of 22. Note that item 2 cannot be selected as its demand exceeds the capacity of the knapsack.

This model naturally arises in resource allocation and scheduling problems where limited resources must be assigned over time. For example, the task of allocating CPU time or communication bandwidth to competing tasks or bids in multiprocessor and network systems, where each bid includes a fixed start and end time and a profit associated with acceptance, and the goal is to maximize total profit under capacity constraints.

When all items share the same time intervals, TDKP reduces exactly to the 0-1 knapsack problem. Since TDKP is known to generalize to the 0-1 knapsack problem and the 0-1 knapsack problem is known to be NP-hard, TDKP is also NP-hard, meaning there are no known polynomial-time algorithms that yield guaranteed optimal solutions.

A pseudo-polynomial dynamic programming (DP) solution exists for the standard knapsack problem [2], which can be interpreted as a process where items are gradually added to the knapsack. However, a straightforward application of this DP formulation does not work for TDKP, as items have to be removed from the knapsack at definite time points.

TDKP can be solved as an integer linear programming (ILP) problem using the previously defined constraints. In the ILP formulation of TDKP [1], a separate capacity constraint is required for each time point. Since each item i contributes to the total demand at every time point in its interval $[l_i, r_i]$, and each item spans $O(n)$ time points (through time point compression, which will be further discussed in 3), the total number of constraints grows as:

$$\#\text{ of constraints} = \sum_{i=1}^{n} |r_i - l_i + 1| = O(n^2)$$

In other words, each of the n items can potentially appear in the summations of up to $O(n)$ time points. Thus, the total number of constraints in the IP formulation grows quadratically, i.e., $O(n^2)$. This makes integer programming approaches computationally expensive for large instances.

[1] presents two special-purpose algorithms. The AO* search strategy, an extension of A* search, features aggressive pruning and is guaranteed to yield optimal solutions, but has an exponential memory requirement, which limited its usability to $n \leq 550$ in the study. The depth-first search strategy, while requiring less memory, is computationally slower and therefore limited to $n \leq 650$ in the same experiment.

As an alternative, heuristics such as simulated annealing can provide high-quality approximate solutions and scale more effectively to large datasets. While the runtime of simulated annealing depends on the number of iterations and the cost of evaluating candidate solutions, it generally avoids the quadratic or exponential growth in complexity of previously presented methods, making it a practical choice for large instances.

In this paper, we present a fast and scalable simulated annealing-based approach to solving the Time-Dependent Knapsack Problem, with both a CPU implementation and a CUDA implementation that empirically achieves up to an 11x speedup over the CPU baseline.

2 SIMULATED ANNEALING

Simulated Annealing is a probabilistic metaheuristic inspired by metallurgy. The algorithm, introduced by [7], solves optimization problems by iteratively exploring the solution space, accepting new solutions based on a probability that depends on the temperature of the system.

A commonly used probability acceptance function is the Metropolis criterion:

$$P = \min\left(1, \exp\left(-\frac{E_{\text{new}} - E_{\text{current}}}{T}\right)\right)$$

In this equation, P represents the probability of accepting the newly generated candidate solution. The terms E_{current} and E_{new} denote the energy (or objective cost) of the current state and the new candidate state, respectively. Because our goal is to maximize the adjusted profit, the energy E in this context maps to the negative value of the adjusted profit; thus, a higher profit results in a lower, more favorable energy state. Finally, T represents the current temperature of the system. This temperature is a control parameter that gradually decreases over time according to a cooling schedule, allowing the algorithm to freely explore the solution space early on while becoming more restrictive and greedy as it converges toward an optimum.

Simulated annealing has been previously used to solve related knapsack problems, such as 0-1 knapsack [6, 10] and multidimensional 0-1 knapsack [4]. [6] notes that while SA can obtain near-optimal results for smaller problem instances, larger instances could benefit from refinements such as multiple runs. [4, 10] report that parallel implementations of SA improve solution quality on similar knapsack problems, further motivating the use of parallelization.

To the best of our knowledge from our literary search, there were no results on the use of SA or parallelized SA on the Time-Dependent Knapsack Problem.

3 SOLVING TDKP USING SIMULATED ANNEALING

We implement a simulated annealing (SA) heuristic in C++ to solve the time-dependent knapsack problem. Each SA run is executed independently and explores the solution space by probabilistically accepting local modifications to the current solution. The use of multiple SA runs allows the algorithm to explore the solution space in various directions and gather increasingly better solutions. Here, we assume a fixed knapsack capacity to simplify the analysis and focus on the parallelization of the core algorithm, meaning C_t does not change over time. The implementation is easily extendable to dynamically changing C_t.

Items are represented as an array of structs containing p_i, w_i, l_i, and r_i.

A candidate solution is represented as a binary vector $\mathbf{x} \in \{0, 1\}^n$, where $x_i = 1$ indicates that item i is selected.

To evaluate the feasibility and overall profit of a candidate solution. The demand at each point in time D_t, defined in 1 as the sum of the demand of all selected items whose availability interval includes the current time point, is tracked.

An observation is that each item occupies knapsack capacity over the time interval $[l_i, r_i]$. However, item changes only occur at

these interval boundaries. Between two consecutive boundaries, the set of active items is constant. Therefore, our implementation compresses the time points by remapping each time point to its temporal order. This reduces the time dimension from a wide range of values (e.g., $1 \ldots 2^{32} - 1$) to $2n$ maximum possible values.

To prove that compressing the time dimension does not affect the correctness of the capacity evaluation, consider that each of the n items is active over a continuous time interval $[l_i, r_i]$. The total demand D_t only changes its value at the start of an item's interval (l_i) or immediately after it ends ($r_i + 1$). Because there are n items, there are at most $2n$ such critical boundaries. Between any two consecutive boundaries in this sorted set of critical points, no items are added or removed, meaning the set of active items—and therefore the total demand D_t—remains perfectly constant. Thus, checking the capacity constraints at these $O(n)$ compressed time points is mathematically equivalent to evaluating the constraint at every single discrete time step in the original domain. This preserves the algorithm's exact correctness while significantly reducing the dimensionality of the time variables.

To further speed up the calculation, the implementation uses a difference array of size $O(n)$, where diff$[t]$ represents the change in demand at time t. For each selected item i with demand d_i and compressed time interval $[l_i, r_i]$, we increment diff$[l_i]$ by d_i and diff$[r_i+1]$ by $-d_i$. The cumulative demand at time t is then obtained by the prefix sum of the difference array.

The use of a difference array yields a substantial improvement in calculation speed compared to a naive update approach. In a naive implementation, updating the demand tracker when an item is added or removed requires iterating through every time step in the item's interval $[l_i, r_i]$, resulting in a time complexity of $O(r_i - l_i + 1)$ for each state change. For large problem instances with wide intervals, this iterative update acts as a severe bottleneck. By utilizing a difference array, toggling an item is reduced to an $O(1)$ operation, requiring only an increment at $diff[l_i]$ and a decrement at $diff[r_i + 1]$. While determining the maximum overall demand still requires computing a prefix sum over the compressed time points in $O(n)$ time, reducing the item toggle step to constant time profoundly accelerates the candidate generation phase of the SA algorithm, where thousands of local modifications are evaluated.

To prevent branching associated with identifying invalid item subsets in the GPU implementation, which can hinder performance when performed every iteration, we instead allow the knapsack demand to temporarily exceed capacity using a penalty term that is proportional to the amount of excess demand. Specifically, let $V = max(D_t - C, 0)$ be the maximum excess demand. The objective function is then adjusted as:

$$\text{Adjusted Profit} = \text{Profit} - V * \text{Penalty Cost}$$

This encourages the selection of solutions that respect the capacity constraint, without explicit correction operations that require many branch operations.

With this evaluation in place, we employ simulated annealing to iteratively explore the solution space. Initially, there are no items selected. At each iteration, the presence or absence of a random item is changed in the candidate solution, the difference array is updated, the new adjusted profit is calculated, and the Metropolis criterion probabilistically determines whether the candidate solution is accepted or not. Acceptance is performed by toggling the respective item in the currently accepted solution, while rejection is performed by reverting the changes to the difference array and toggling the item back to its original state in the candidate solution.

After all iterations are complete, remaining excess demand violations are repaired by randomly deselecting items until all capacity constraints are satisfied.

To clarify the algorithmic flow and address the specific mechanics of our implementation, we provide pseudocode for our approaches. Algorithm 1 outlines the general sequence of operations and hyperparameter updates applied during successive iterations of Simulated Annealing. Furthermore, Algorithm 2 details the CUDA-specific thread cooperation and memory management techniques utilized to evaluate states concurrently within the GPU.

Algorithm 1 General Simulated Annealing Flow for TDKP

Require: Items with profit p_i, demand d_i, and interval $[l_i, r_i]$; Capacity C_t
Require: Initial Temp T_{init}, Cooling Rate α, Iterations L, Penalty Cost V
Ensure: Best valid subset of items
Initialize $S_{\text{current}} \leftarrow \emptyset$, $S_{\text{best}} \leftarrow \emptyset$
$T \leftarrow T_{\text{init}}$
Initialize difference array diff of size $2n$ to zeros
while $T > T_{\text{min}}$ **do**
  for iter = 1 to L **do**
    Select a random item i
    $S_{\text{new}} \leftarrow S_{\text{current}}$ with item i toggled
    Update diff based on addition/removal of d_i at $[l_i, r_i]$
    Compute prefix sums of diff to find max demand D_{max}
    Excess $\leftarrow \max(D_{\text{max}} - C_t, 0)$
    Adjusted_Profit$_{\text{new}} \leftarrow$ Profit(S_{new}) − (Excess $\cdot V$)
    $E_{\text{new}} \leftarrow -$Adjusted_Profit$_{\text{new}}$
    $\Delta E \leftarrow E_{\text{new}} - E_{\text{current}}$
    if $\Delta E < 0$ **or** Random$(0, 1) < \exp(-\Delta E/T)$ **then**
      $S_{\text{current}} \leftarrow S_{\text{new}}$
      $E_{\text{current}} \leftarrow E_{\text{new}}$
      if Adjusted_Profit$_{\text{new}}$ > Best_Adjusted_Profit **then**
        $S_{\text{best}} \leftarrow S_{\text{current}}$
        Best_Adjusted_Profit $\leftarrow$ Adjusted_Profit$_{\text{new}}$
      end if
    else
      Revert changes to diff and S_{new}
    end if
  end for
  $T \leftarrow T \cdot \alpha$
end while
if S_{best} violates capacity constraint **then**
  Randomly remove items from S_{best} until max excess demand = 0
end if
return S_{best}

4 CUDA-OPTIMIZED TDKP

CUDA (Compute Unified Device Architecture) is a parallel programming model that enables general-purpose computation on GPUs by executing large numbers of lightweight threads concurrently. Threads are grouped into warps that follow a Single Instruction, Multiple Threads (SIMT) execution model, where performance is maximized when threads execute the same control flow and access contiguous memory locations.

This execution model makes GPUs well suited for data-parallel workloads with regular memory access patterns. However, achieving high performance requires restructuring algorithms and data layouts to minimize control-flow divergence and maximize memory coalescing.

Since SA is executed multiple times, this naturally lends itself to a parallel implementation. However, achieving high performance on GPUs requires exploiting additional architectural features. In particular, our implementation leverages massive thread-level parallelism and high global memory bandwidth offered by GPUs.

In the CPU implementation, the array data of each SA instance was stored in separate contiguous arrays, which is efficient for sequential access. In contrast, the GPU implementation organizes all SA instances in a single interleaved array per type of data (e.g., candidate solution, demand at each point in time). This organization is specifically designed to support SIMT (Single Instruction, Multiple Threads) operations, where a single instruction is issued simultaneously across a warp of threads. Figure 2 shows the memory layout in detail.

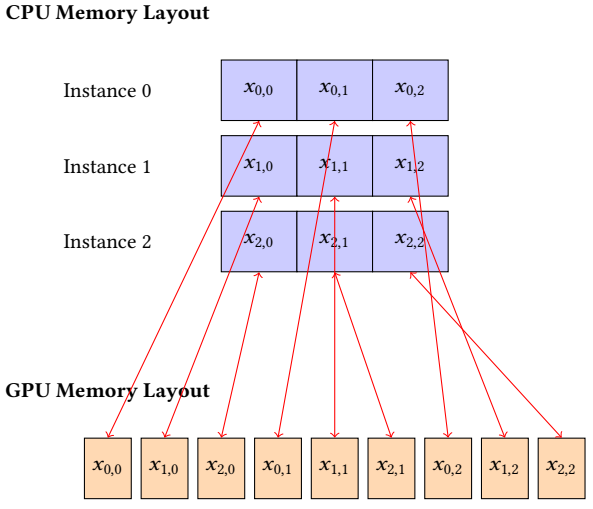


Figure 2: Comparison of CPU and GPU memory layouts for simulated annealing instances. CPU arrays (blue, top) are separate per instance. GPU uses a unified array (orange, bottom), with elements from all instances interleaved per index (red arrows highlight mapping).

We employ a two-dimensional thread block mapping to exploit both coarse-grained and fine-grained parallelism. The x-dimension of the thread block is mapped to independent SA instances, ensuring that threads within a warp execute independent SA instances following the SIMT paradigm. Exploiting the memory layout, these threads access physically adjacent memory locations simultaneously. Consequently, the GPU can coalesce these concurrent requests into fewer cache line transactions, significantly maximizing memory bandwidth.

Simultaneously, most tasks within each SA instance are done in parallel. The y-dimension provides a pool of worker threads that cooperatively evaluate a candidate solution. Worker threads jointly calculate the prefix sum using the Scan algorithm [3] and count the amount of excess demand. All worker threads of each SA instance reside in the same thread block, so they communicate with shared memory. However, evaluating the acceptance of the solution using the Metropolis criterion is only performed by one thread per instance. Kernel synchronization intrinsics (`__syncthreads`) are used to coordinate worker threads between phases of algorithm execution.

Repairing the remaining excess demand violations in the final set of items is performed on the CPU, using the CPU-based implementation, due to the aforementioned branching that is not advisable in the GPU. Data is explicitly prefetched back to the host using `cudaMemPrefetchAsync` to reduce page fault overhead [9].

5 SEGMENT TREE-OPTIMIZED TDKP

This work originated as a CUDA-focused course project; consequently, data structures with heavy branching or pointer usage were excluded from the initial algorithm design. As the candidate state evaluation involves adding to a range and checking for maximum excess demand values, this could be further optimized through range-query data structures, such as segment trees.

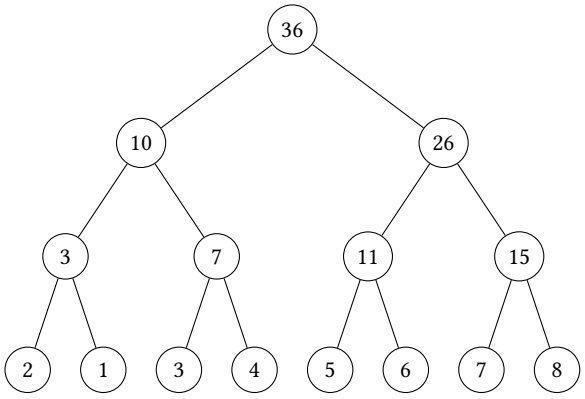


Figure 3: A segment tree storing range sums

A segment tree is a balanced binary tree data structure designed to support efficient range queries and updates on an array. Each node represents an interval, with the root covering the entire array and leaves corresponding to individual elements. Internal nodes aggregate information (such as sum, minimum, maximum, or GCD) from their children, enabling queries over any subrange in $O(\log n)$ time. At each node, we check whether the node's segment is inside the query range, partially overlaps with the query range, or is outside of the query range.

Algorithm 2 CUDA Thread Execution and Memory Management for SA

Require: Unified interleaved array of item data, SA state arrays (interleaved)
Ensure: Best candidate state array evaluated across all blocks

▷ Allocate interleaved unified memory on Host
`cudaMallocManaged(&interleaved_states, size · num_instances)`
`cudaMemPrefetchAsync(interleaved_states, GPU_device)`

▷ Kernel Launch: x-dimension = SA instances, y-dimension = worker threads
Launch `SA_Kernel`⟨⟨⟨num_instances, worker_threads_per_instance⟩⟩⟩

▷ Inside SA_Kernel
instance_id ← `blockIdx.x`
worker_id ← `threadIdx.y`
Allocate *__shared__* memory for prefix sum and max demand tracking
while $T > T_{\min}$ **do**
  for iter = 1 to L **do**
    if worker_id = 0 **then**
      Toggle random item in candidate_state[instance_id]
      Update diff_array[instance_id]
    end if
    `__syncthreads()` ▷ Ensure visibility
    ▷ Cooperative solution evaluation
    Parallel prefix sum on diff_array[instance_id]
    `Parallel_Prefix_Sum(diff_array, shared_memory, worker_id)`
    `__syncthreads()`
    Parallel max reduction
    `Parallel_Max_Reduction(prefix_sum_results, shared_memory, worker_id)`
    `__syncthreads()`
    if worker_id = 0 **then**
      Calculate E_{new} using max demand from shared memory
      if Accept($E_{\text{new}}, E_{\text{current}}, T$) **then**
        Update best_state[instance_id]
      else
        Revert candidate_state[instance_id] and diff_array[instance_id]
      end if
    end if
    `__syncthreads()` ▷ Finalize decision
  end for
  if worker_id = 0 **then**
    $T \leftarrow T \cdot \alpha$
  end if
end while

▷ Back on Host CPU
`cudaMemPrefetchAsync(interleaved_states, CPU_device)`
Perform sequential excess demand correction on CPU

Figure 3 shows a sample segment tree. The leaf nodes store individual array elements, while the internal nodes store the result of combining their children. The same tree structure can be used for any associative operation (e.g., min, max, gcd). Range queries in a segment tree performed by traversing from the root downwards, using a node's stored value if its segment is fully inside the query range and ignoring it if fully outside. Partially overlapping nodes are recursively queried, and results are combined using the chosen associative operations.

Range updates can also be performed in $O(\log n)$ time using lazy propagation. Instead of propagating the update down to all children right away, each node stores a "lazy" value representing pending updates. When a query or further update touches that node, the lazy value is applied to the node's aggregated information, and the pending update is propagated to its children only if necessary.

In place of creating a difference array and calculating its prefix sum in $O(\log n)$ time, the segment tree-based approach uses $O(\log n)$ range add operations to increment the demand at a specific contiguous time interval. An $O(\log n)$ range max operation is then performed to determine the maximum demand incurred throughout all intervals, used in calculating the adjusted profit.

We employed a publicly-available Segment Tree Beats implementation from Nyaan's Library, an algorithms library primarily used in competitive programming [8]. This extended segment tree implementation supports a variety of range operations, including clamping and range assignment. For our experiments, we only utilized the range addition and range maximum functionalities.

6 EVALUATION

We evaluated our approach using randomly generated test cases with parameters summarized in Table 1. Each item in a test case was assigned a random starting point and a time interval duration uniformly drawn between 2 and 6 units. To ensure reproducibility, all random numbers controlling the simulated annealing process were generated pseudorandomly using the Splitmix64 algorithm [11]. he source code, including both the CPU and GPU implementations, as well as the preliminary datasets used for these experiments, are publicly available on GitHub at: csc612-tdkp.

n	Knapsack Capacity	Maximum Time Point Value
10	20	100
25	50	250
50	100	500
100	200	1000
200	400	1000
250	500	1000
500	1000	1000
1000	2000	1000
2000	4000	1000
10000	20000	20000
50000	100000	100000
100000	200000	200000

Table 1: Parameters of randomly generated sample test cases

Correctness of the CUDA implementation was verified by comparing its results against those from the CPU implementation. In addition, to establish a baseline for solution quality, integer linear programming (ILP) was applied to the same test cases using the CP-SAT solver [5] of OR-Tools, Google's open-source optimization library available in Python. The ILP results provide a ground-truth reference, enabling us to quantify how closely the heuristic solutions approximate an optimal solution.

The program was compiled with CUDA 13.0 using nvcc and the -O3 optimization flag to ensure high-performance execution. All experiments were performed on a laptop with an Intel® Core™ i7-13650HX Processor and NVIDIA GeForce RTX 5070 GPU, using hyperparameters listed in Table 2.

The execution time of the CPU implementation refers to the time taken for the SA function call to complete. Meanwhile, the execution time for the CUDA implementation consists of the kernel call and CPU-side excess demand correction. Both measurements exclude the time spent parsing input, performing input pre-processing (time point compression), and dynamically allocating memory.

Parameter	Value
Iterations per annealing run	16,000
Initial temperature	100,000
Cooling rate	0.995
Penalty per excess maximum unit demand	10
Number of parallel annealing instances	256
Workers threads per instance	128

Table 2: Simulated annealing hyperparameters and parallelization settings

7 RESULTS AND DISCUSSION

n	ILP (global optima)	SA (CPU)	SA (CPU-opt) and SA (GPU)
10	104	104	104
25	281	281	281
50	613	613	613
100	1188	1188	1188
200	2574	2574	2574
250	3100	3100	3100
500	6332	6332	6332
1000	12462	12462	12462
2000	24903	24903	24903
10000	124116	99745	99745
50000	624258	172926	172926
100000	1253503	187583	187583

Table 3: Comparison of outputs between ILP, CPU SA, CPU-Opt, and GPU SA.

Table 3 shows that Simulated Annealing (SA) matches the Integer Linear Programming (ILP) values exactly for $n \leq 2000$, indicating both consistency and the ability to find the global optimum for smaller problem sizes. As n increases, the SA results begin to deviate from the ILP solutions, which is expected for a heuristic approach. This behavior is consistent with the findings of [6], which reported that solution quality for the 0/1 knapsack problem tends to degrade as input size grows. The deviation observed here may also be influenced by the relatively limited hyperparameter tuning, as our primary focus was on evaluating execution time and computational speedup.

Upon further investigation of the $n = 10^5$ case using an alternative hyperparameter configuration of 8 SA instances with 128,000 iterations per instance and 128 worker threads, the solution obtained was 909,813 in 93.392 seconds, which is closer to the global optimum as identified by OR-Tools. While OR-Tools was able to find the exact global optimum for all tested instances, it required 11 minutes to solve the $n = 10^5$ case, reflecting the quadratic growth in the number of linear program constraints. In contrast, SA provided a relatively high-quality solution in a fraction of the time, demonstrating its advantage in terms of computational efficiency.

n	SA (CPU) [ms]	SA (GPU) [ms]	Speedup
10	128.41	55.05	2.33x
25	142.96	59.76	2.39x
50	240.47	65.45	3.67x
100	350.03	75.52	4.63x
150	230.35	65.84	3.50x
200	640.61	94.08	6.81x
250	743.53	99.19	7.50x
300	249.30	67.50	3.69x
500	1083.43	131.39	8.25x
1000	1464.54	155.10	9.44x
2000	1623.22	169.48	9.58x
10000	20652.80	1756.36	11.76x
50000	102012.61	75113.90	1.36x
100000	202667.13	166888.63	1.21x

Table 4: Execution time comparison between CPU and GPU implementations (averaged over 5 runs per configuration).

Table 4 summarizes the execution times of the CPU and GPU implementations across a range of problem sizes. For small problem sizes ($n \leq 50$), the GPU provides modest speedups (2-4x), primarily because fixed overheads such as kernel launch and data transfer dominate the total runtime. As n increases to moderate sizes ($n \approx 100–10,000$), the GPU becomes compute-bound, allowing it to fully exploit parallelism and achieve maximum speedups (up to 12x). However, for very large problem sizes ($n \geq 50000$), the speedup collapses. This decline is likely due to memory limitations and bandwidth saturation, which reduce cache efficiency and cause the GPU to become memory-bound, limiting the benefits of parallel execution.

Table 5 summarizes the execution times of the CPU and segment tree–optimized CPU implementations across a range of problem sizes. For small problem sizes ($n \leq 150$), the data structure provides modest speedups (approximately 2–4×), primarily due to the overhead associated with range query and update operations. As n increases, the optimized implementation achieves substantially

n	SA (CPU) [ms]	SA (CPU-Opt) [ms]	Speedup
10	128.41	34.88	3.68x
25	142.96	41.77	3.42x
50	240.47	49.91	4.82x
100	350.03	60.07	5.83x
150	230.35	54.92	4.19x
200	640.61	68.58	9.34x
250	743.53	71.34	10.42x
300	249.30	58.09	4.29x
500	1083.43	85.99	12.60x
1000	1464.54	94.28	15.53x
2000	1623.22	97.79	16.60x
10000	20652.80	118.26	174.58x
50000	102012.61	117.85	865.49x
100000	202667.13	145.22	1395.66x

Table 5: Execution time comparison between CPU and segment tree-optimized CPU implementations (averaged over 5 runs per configuration).

larger speedups, reflecting the improved asymptotic complexity of the Segment Tree approach. In particular, replacing naïve linear scans with logarithmic-time range set and range maximum queries reduces the overall time complexity, allowing the optimized version to scale efficiently for larger inputs. Consequently, the performance gap widens with increasing n, demonstrating that the benefits of the Segment Tree outweigh its overhead for sufficiently large problem sizes.

8 LIMITATIONS AND FUTURE WORK

A major limitation of this study is that it was only evaluated using a small, fixed set of randomly-generated test cases, which may not fully capture its performance across a broader range of scenarios. The robustness of the algorithm in more diverse problem settings remains an open question.

Moreover, the performance of Simulated Annealing may be further enhanced through enhanced hyperparameter tuning and hybridization with other heuristic or optimization techniques. The logarithmic time complexity of each iteration of segment tree-based approach allows the iteration count to be further increased with minimal cost. A potentially promising avenue for future work involves combining SA with Linear Programming, referred to as the SA-LP hybrid approach. The idea behind such hybrid methods is to leverage the strengths of both techniques: Simulated Annealing's ability to avoid getting stuck in local optima through random exploration and Linear Programming's capability to efficiently solve linear constraints or optimization problems. Future work can also explore strategies to address the scalability of ILP, which would allow for scalable SA-LP strategies.

9 ACKNOWLEDGMENTS

We would like to express our sincere gratitude to Prof. Roger Luis Uy for their invaluable guidance, support, and encouragement throughout this work. This research initially began as a class project for CSC612M, and it was through their mentorship and encouragement that we were inspired to further develop our findings and submit this manuscript for publication.

10 CONCLUSION

In this study, we were able to investigate the use of simulated annealing on the time-dependent knapsack problem and implement it on both CPU and GPU. Our findings show that simulated annealing implementation can be optimized through parallel computing and range-query data structures. In particular, segment trees proved to be more scalable overall, as their lower asymptotic time complexity for updates and queries enables better performance as problem sizes grow, especially in large-scale and high-frequency evaluation settings.

Future work could focus on tuning hyperparameters or enhancing Simulated Annealing through hybrid approaches, such as combining it with Linear Programming (SA-LP), to leverage both random exploration and efficient constraint handling.

REFERENCES

[1] Mark Bartlett, Alan Frisch, Youssef Hamadi, Ian Miguel, S. Armagan Tarim, and Chris Unsworth. 2005. The Temporal Knapsack Problem and Its Solution. *Lecture Notes in Computer Science* 3524, 34–48. doi:10.1007/11493853_5

[2] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. 2022. *Introduction to Algorithms* (4th ed.). MIT Press.

[3] NVIDIA Corporation. 2012. Chapter 39: Parallel Prefix Sum (Scan) on CUDA.

[4] Bianca de Almeida Dantas and Edson Norberto Cáceres. 2018. An experimental evaluation of a parallel simulated annealing approach for the 0–1 multidimensional knapsack problem. *J. Parallel and Distrib. Comput.* 120 (2018), 211–221. doi:10.1016/j.jpdc.2018.02.031

[5] Google. 2023. CP-SAT Solver. https://developers.google.com/optimization/cp/cp_solver. Accessed: 2025-12-18.

[6] Saja Murtadha Hashim, Emrullah Sonuç, and Kürsat Mustafa Karaoglan. 2024. Performance Analysis of Simulated Annealing on the 0-1 Knapsack Problem. In *2024 IEEE 9th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*. 1–5. doi:10.1109/ICETAS62372.2024.11119878

[7] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. 1983. Optimization by simulated annealing. *Science* 220, 4598 (1983), 671–680. doi:10.1126/science.220.4598.671

[8] Nyaan. 2020. Segment Tree Beats. https://nyaannyaan.github.io/library/segment-tree/segment-tree-beats.hpp.html. Accessed: 2026-01-20.

[9] Nikolay Sakharnykh. 2017. *Maximizing Unified Memory Performance in CUDA*. NVIDIA Developer Blog. https://developer.nvidia.com/blog/maximizing-unified-memory-performance-cuda/ Accessed: 2025-12-17.

[10] Emrullah Sonuç, Baha Sen, and Şafak Bayir. 2016. A parallel approach for solving 0/1 knapsack problem using simulated annealing algorithm on CUDA platform. *International Journal of Computer Science and Information Security,* 14 (12 2016), 1096.

[11] Guy L. Steele, Doug Lea, and Christine H. Flood. 2014. Fast splittable pseudorandom number generators. *SIGPLAN Not.* 49, 10 (Oct. 2014), 453–472. doi:10.1145/2714064.2660195

Scrum vs. Extreme Programming (XP) vs Kanban in the Age of AI and DevOps: A Comparative Analysis of Cultural Resilience to Modern Development Challenges

Fernand G. Bernardez
University of the Philippines Cebu
, Cebu City , Philippines
fgbernardez@up.edu.ph

Francis Luis M. Torres
University of the Philippines Cebu
, Cebu City , Philippines
fmtorres@up.edu.ph

ABSTRACT

Cultural resilience has become a critical capability in Agile software development as organizations grapple with the combined pressures of Artificial Intelligence (AI) integration and DevOps-driven delivery. This paper presents a qualitative, theory-building, comparative analysis of three widely adopted Agile frameworks (Scrum, XP, and Kanban) to examine how their structural and cultural mechanisms support resilience in contemporary development environments. Using structured comparative analysis and integrative literature synthesis, this study analyzes how each framework enables learning, manages change, and sustains stability under increasing delivery velocity and operational complexity. The analysis shows that AI does not replace Agile practices. But it modifies how resilience mechanisms operate, accelerates feedback, and reduces cognitive load while introducing new cultural risks related to over-reliance and skill erosion. No single framework adequately addressed the combined demands of adaptability, stability, and continuous learning. Instead, cultural resilience is most effective through hybrid configurations that integrate complementary strengths. This paper proposes a Scrumban–XP hybrid assisted by AI to achieve cultural resilience from said systems. It proposes a mix of Scrum's institutionalized learning loops, Kanban's flow-based controls, and XP's disciplined engineering practices that are supported by AI-assisted tooling and evolutionary adoption. The study concludes with implications for organizational strategy, software engineering education, and future empirical research on resilience in AI-augmented development contexts.

CCS CONCEPTS

• **Software and its engineering** → **Agile software development**;

KEYWORDS

Cultural resilience, Agile software development, Scrum, Extreme Programming (XP), Kanban, DevOps, AI-assisted development

ACM Reference Format:
Fernand G. Bernardez and Francis Luis M. Torres. 2026. Scrum vs. Extreme Programming (XP) vs Kanban in the Age of AI and DevOps: A Comparative Analysis of Cultural Resilience to Modern Development Challenges. In *Proceedings of Philippine Computing Science Congress (PCSC2026).* Laguna, Philippines, 6 pages.

1 INTRODUCTION AND RESEARCH CONTEXT

1.1 The Volatile Landscape of Modern Software Development

Today, software engineering operates in an environment marked by market volatility, rapid technological change, and continually evolving requirements [1]. Traditional plan-driven models, such as Waterfall, struggle under these conditions due to their limited ability to accommodate uncertainty and last-stage change, often resulting in high rework costs and reduced project success [1]. Agile methodologies address these challenges through iterative development, close customer collaboration, and rapid feedback cycles [1, 2].

The concurrent rise of Artificial Intelligence (AI) and the maturation of DevOps practices further intensify these dynamics. While DevOps automation and CI/CD pipelines enable higher deployment frequency and shorter lead time, they also impose stricter expectations for operational stability, including low change failure rates (CFR) and rapid recovery from incidents, commonly measured as mean time to repair/recovery (MTTR) [3]. This combination creates persistent tension between delivery velocity and system reliability.

Addressing this tension requires more than procedural agility. It demands *cultural resilience*—the organizational capacity to absorb disruption while sustaining continuous delivery. Accordingly, this study presents a comparative analysis of Scrum, XP, and Kanban, examining how their cultural and structural mechanisms support resilience in AI-augmented, DevOps-oriented development environments. This analysis draws on established Agile and DevOps research and recurring patterns observed in contemporary software engineering practice.

1.2 Defining Cultural Resilience in the Digital Age

In software engineering, cultural resilience refers to an organization's collective capacity–not merely its technical recovery mechanisms–to prevent, adapt to, recover from, and learn following operational disruptions [4]. Prior research links organizational resilience to agility and effective diffusion of adaptive practices across teams and enterprises [5]. Achieving such resilience requires a cultural shift toward shared responsibility, continuous learning, and decentralized decision-making, rather than reliance on centralized control or tooling alone [4, 6].

Cultural resilience also underpins innovation and experimentation, both of which inherently involve the risk of failure [7]. Consequently, agile environments depend on leadership practices that

foster psychological safety, enabling experimentation without fear of blame [8]. Psychological safety supports open communication and collaboration, which are prerequisites for institutional learning through practices such as retrospectives and blameless postmortems [7, 9]. Empirical organizational research further demonstrates that psychological safety facilitates team learning behaviors, which in turn mediate improvements in team performance and adaptability [10].

When enacted consistently, these cultural mechanisms influence operational outcomes. Organizations that promote transparent reflection on failure and rapid process adaptation are better positioned to reduce recovery times following incidents, thereby improving mean time to recovery (MTTR) and overall system stability [3]. Cultural resilience is therefore established not through tooling alone, but organizational norms that transform disruption into sustained learning and adaptive capacity.

2 RESEARCH APPROACH AND METHODOLOGY

This study adopts a qualitative, theory-building approach based on structured comparative analysis and integrative literature synthesis. Rather than conducting project-level empirical investigation, this paper developed a conceptual framework to examine how Scrum, XP, and Kanban differ in sustaining cultural resilience under AI- and DevOps-driven conditions. Accordingly, the study does not pursue primary empirical validation; instead, it synthesizes analytically grounded patterns intended to inform and motivate future empirical research.

The methodological design integrates peer-reviewed academic literature with selected practitioner and industry sources. Scholarly publications provide the theoretical foundations for Agile frameworks [1, 2], organizational resilience [4, 5], and DevOps practices [3, 11], while practitioner sources are used descriptively to capture emerging tools, workflows, and cultural friction points associated with AI-assisted development [12, 13]. In rapidly evolving socio-technical domains where practice often precedes formal academic theorization, this balanced integration enhances relevance and timeliness without positioning non-academic sources as primary evidence for causal or generalizable claims.

2.1 Comparative Framework and Scope

The three frameworks were selected due to their widespread adoption and distinct resilience mechanisms. Scrum's learning loops [7, 8], XP's engineering discipline [14], and Kanban's flow-based control [15, 16] allow a comparative analysis of complementary cultural levers across AI- and DevOps-driven contexts.

Scrum, XP, and Kanban are examined using a structured comparative framework across three analytical dimensions: (1) **structural mechanisms** (e.g., time-boxing, work-in-progress limits, and engineering practices); (2) **cultural values and behavioral norms** (e.g., discipline, collaboration, learning orientation, and failure tolerance); and, (3) **resilience outcomes**, assessed through conceptual alignment with DevOps principles and commonly referenced operational indicators (e.g., MTTR, change failure rate) [3]. The analysis focuses on organizational and team-level cultural mechanisms rather than technical implementation details or empirical performance measurement. This paper seeks to identify recurring patterns and trade-offs rather than to statistically rank frameworks.

2.2 Literature Selection

To keep this review concise, this paper prioritized peer-reviewed publications and practitioner materials about emerging AI and DevOps practices in the last 10–15 years.

Sources were selected based on relevance to Agile frameworks, DevOps practices, AI-assisted software development, and organizational resilience. Practitioner literature is used descriptively to contextualize contemporary tools, workflows, and cultural friction points. Analytical interpretation and theoretical framing are grounded in established academic research.

3 THEORETICAL FOUNDATIONS: AGILITY, RESILIENCE, AND MODERN PARADIGMS

3.1 The Triad of Modern Development: Agile, DevOps, and AI

Agile development emphasizes iterative delivery, customer collaboration, and adaptability in the presence of change [1, 17]. Scrum, XP, and Kanban represent distinct implementations of these Agile principles, each emphasizing different structural and cultural mechanisms [1, 17]. While complementary to Agile, DevOps constitutes a broader organizational and cultural shift in cross-functional collaboration, shared accountability, and automation across the software value stream [11, 18]. Its effectiveness depends largely on cultural factors such as trust, transparency, and the breakdown of functional silos [18, 19].

Artificial Intelligence (AI) represents a paradigm shift, increasingly integrated into software development to automate routine tasks, enhance predictability, and support data-driven decision-making [12]. Rather than replacing human judgment, AI augments development and management activities by accelerating feedback, improving forecasting, and reducing cognitive overhead [12]. Together, Agile, DevOps, and AI form an interdependent triad that reshapes how software teams balance speed, stability, and learning.

3.2 Comparison of Core Framework Mechanics

Although Scrum, XP, and Kanban share a common Agile foundation, they differ substantially in their core mechanics, required discipline, and cultural emphasis [20]. Table 1 summarizes their structural and cultural characteristics.

Quantitative assessments of agility provide a partial baseline for comparison across Agile frameworks. Previous studies report a higher overall degree of agility for Scrum ($DA = 0.86$) than XP ($DA = 0.70$). This difference is largely attributed to Scrum's prescriptive ceremonies that institutionalize inspection and adaptation [1]. Empirical evaluations further indicate that consistent execution of these ceremonies—particularly sprint retrospectives—serves as a key mechanism for team learning and continuous process improvement [22, 23]. By contrast, XP embeds agility directly within engineering practice through strict technical disciplines such as Test-Driven Development (TDD) and Continuous Integration (CI) [9, 24].

These distinctions reflect contrasting approaches to cultural resilience. Scrum relies on empirical process control, with resilience

Table 1: Structural and Cultural Comparison of Agile Frameworks

Attribute	Scrum	XP	Kanban
Core Mechanism	Time-boxed Iteration (Sprint)	Short Cycles; High Technical Discipline	Systemic Flow Control (WIP Limits)
Primary Value	Commitment, Transparency, Focus [21]	Courage, Simplicity, Respect [14]	Start with What You Do Now; Evolutionary Change [16]
Roles & Structure	Defined (SM, PO, Dev Team) [20]	Fewer defined roles (Team, Customer)	Minimal defined roles; respects existing hierarchy [20]
Approach to Change	Adaptable within Sprint context	Welcomes change at any time	Highly Adaptive; via policy evolution
Agility Quantification	High (0.86) [1]	Moderate (0.70) [1]	N/A (Flow-focused metric)

emerging from regular inspection and adaptation. XP emphasizes intrinsic technical quality, where disciplined engineering practices reduce the cost and risk of change. Kanban focuses on systemic resilience, using visualization and explicit work-in-progress (WIP) limits to expose bottlenecks and enable evolutionary policy change [15, 16]. Overall, effectiveness depends on organizational readiness: insufficient technical discipline can undermine XP, weak leadership can compromise Scrum, and resistance to policy evolution can limit Kanban's benefits [1, 16, 25].

4 COMPARATIVE ANALYSIS OF FRAMEWORKS' INHERENT RESILIENCE MECHANISMS

4.1 Scrum: Resilience through Time-Boxing and Learning Loops

Scrum promotes cultural resilience through its iterative, time--boxed structure, which limits the impact of technical or directional failure and enables rapid course correction [8, 17]. Short Sprints constrain the blast radius of mistakes, while the Sprint Retrospective institutionalizes learning by requiring teams to inspect processes and adapt practices [7]. This combination supports failure tolerance and affordable experimentation, provided that leadership reinforces psychological safety and supports change [7, 8]. Empirical studies of Scrum retrospectives indicate that structured reflection activities contribute to actionable improvement insights and measurable process changes throughout subsequent iterations [22, 23].

4.2 XP: Resilience through Engineering Discipline

XP is commonly associated with proactive resilience achieved through disciplined engineering practices that reduce the cost and risk of change [9]. Core practices (Test-Driven Development (TDD), Continuous Integration (CI), refactoring, and pair programming) embed quality directly into the development process, promoting a constant readiness for change [1, 14]. XP's core values—particularly Courage and Simplicity—reinforce this emphasis by encouraging early problem discovery and confident adaptation [14].

Empirical research supports this resilience mechanism. Industrial case studies, meta--analyses, and recent empirical evaluations indicate that TDD and CI practices are associated with substantial reductions in pre-release defect density. But they may introduce modest increases in initial development effort or yield mixed productivity outcomes [26–28]. In addition, systematic reviews report that pair programming can improve software quality and knowledge transfer when applied to appropriate task and team contexts [29, 30].

However, XP's reliance on sustained discipline and high technical maturity can introduce cultural resistance, particularly in teams unaccustomed to intensive collaboration or continuous refactoring [25, 31]. Consequently, XP's potential resilience is strongest in environments where organizations enable them to invest in both technical rigor and supportive cultural conditions.

4.3 Kanban: Resilience through Systemic Flow Optimization

Kanban promotes systemic resilience by optimizing the continuous flow of work through visualization and explicit limits on work in progress (WIP) [15, 16]. By constraining WIP, teams are encouraged to report bottlenecks early, reduce multitasking, and stabilize delivery under changing conditions [18]. Its guiding principle of "start with what you do now" supports low-friction adoption and respects existing organizational roles, contributing to high initial cultural acceptance [15, 20].

Empirical research supports this flow-based resilience mechanism. Quantitative studies about Kanban mention that lower WIP levels are associated with shorter lead times. However, productivity may vary depending on context and measurement approach [32]. Complementary experimental and lab-scale studies further suggest that regulating WIP can reduce cycle-time variability and improve delivery predictability [33].

However, Kanban's resilience depends on disciplined adherence to explicit policies and active use of flow metrics. When WIP limits are weakly enforced or ignored, systems risk devolving into unresolved queues, undermining both productivity and resilience [34]. Consequently, Kanban relies less on prescribed roles and more on sustained, data-informed policy evolution to maintain adaptive capacity.

5 CULTURAL RESILIENCE IN THE AGE OF AI INTEGRATION

The integration of Artificial Intelligence (AI) into software development reshapes Agile practices by accelerating feedback, automating routine tasks, and supporting data-driven decision-making [12]. While these capabilities offer efficiency gains, effective adoption requires cultural adaptation to address friction points such as resistance to new tools, over-reliance on automation, and concerns about skill degradation [13]. Thus, cultural resilience plays a central role in determining whether AI strengthens or undermines organizational adaptability.

5.1 AI-Augmented Framework Practices

Across Scrum, XP, and Kanban, AI capabilities (e.g., code generation, predictive analytics, and automated status reporting) can augment core framework practices by reducing cognitive load and improving situational awareness [12]. However, these benefits depend on teams' ability to critically evaluate AI-generated output rather than accept it uncritically. Table 2 summarizes how AI integration may support resilience across frameworks, along with associated cultural risks, consistent with the analysis presented in this section.

5.2 AI and XP: Reducing Discipline Friction

XP's emphasis on strict engineering discipline represents both its main source of resilience and a significant adoption challenge [25, 31]. AI-assisted development tools can mitigate this friction by supporting practices (i.e., TDD) and refactoring through automated test generation and real-time code analysis [35]. Empirical studies indicate that AI-assisted programming can substantially reduce task completion time, while the effects on code quality and security remain mixed [36, 37]. When applied judiciously, AI can therefore enable teams to sustain XP's technical resilience while reducing the cultural cost associated with rigorous discipline.

5.3 AI and Kanban: From Reactive to Predictive Flow Management

Kanban traditionally supports resilience through visualization and reactive bottleneck management [38]. Integration of AI extends this capability by enabling predictive analysis of cycle times, throughout trends, and work-in-progress violations, allowing teams to anticipate flow disruptions before they occur [38, 39]. Empirical research on predictive CI and flow monitoring suggests that early detection of disruptions can improve delivery stability and recovery readiness when combined with human judgment [40, 41]. However, because Kanban's learning model depends on collaborative policy evolution rather than prescriptive control, AI insights must be treated as inputs for experimentation rather than automated directives to preserve systemic resilience.

These AI-driven modifications further reinforce the limitations of single-framework adoption, motivating the need for hybrid configurations discussed in the following section.

6 HYBRID MODELS AND STRATEGIC ADOPTION FOR ENHANCED RESILIENCE

6.1 The Hybrid Model Imperative

Based on the preceding analysis of Agile frameworks under AI- and DevOps-driven conditions, the combined pressures of AI integration and DevOps automation often render exclusive reliance on a single Agile framework insufficient or overly rigid [42]. Hybrid approaches have emerged as a pragmatic strategy, enabling organizations to combine complementary practices while adapting to project context, resource constraints, and cultural readiness [43]. This perspective aligns with Disciplined Agile principles, which emphasize situational fit and informed practice selection rather than strict methodological purity [20]. Importantly, hybridization allows organizations to integrate multiple resilience mechanisms—learning, flow control, and engineering discipline—rather than rely on any single approach.

6.2 Architecting the Resilient Hybrid: Scrumban–XP

Building on the framework-specific resilience mechanisms identified in Sections 4 and 5, this study proposes a **Scrumban–XP hybrid** that integrates complementary strengths from Scrum, Kanban, and XP:

- **Scrum** contributes iterative stability and institutionalized learning through time-boxing and retrospectives.
- **Kanban** improves systemic resilience by introducing flow visualization and WIP limits within iterations, supporting continuous delivery and mitigating the "mini-waterfall" risk in DevOps-oriented environments [44].
- **XP** provides technical resilience through disciplined engineering practices (e.g., TDD, continuous integration, refactoring), helping maintain code quality under high delivery velocity.

AI-assisted tools can further support this hybrid by reducing cognitive and procedural overhead—particularly for XP practices—while improving flow visibility and feedback speed. Kanban's principle of evolutionary change enables incremental adoption, reducing resistance while progressively institutionalizing high-discipline practices in teams.

6.3 Strategic Considerations for Cultural Transformation

Framework composition is insufficient to achieve resilience; sustained cultural transformation remains essential. Two considerations are particularly salient:

- **Failure Tolerance and Psychological Safety:** Resilient organizations cultivate environments that support experimentation, reflection, and learning from failure. Practices such as blameless post-mortems and regular retrospectives enable continuous improvement across hybrid configurations [7, 8].
- **Managing AI-Driven Skill Gaps:** AI-enabled development and DevOps automation increase reliance on specialized expertise [13]. While Kanban's flexibility facilitates rapid integration of such skills, Scrum and XP require deliberate

Table 2: AI Integration, Potential Resilience Benefits, and Cultural Friction

Framework	AI-Augmented Practice	Potential Resilience Benefit	Cultural Friction / Risk
Scrum	AI-assisted backlog prioritization and predictive planning	May support planning stability and faster feedback under volatility	Over-reliance on predictive outputs; tension with collaborative prioritization
XP	AI-supported test generation, refactoring, and pair programming	Can reduce cognitive load and lower the cultural cost of technical discipline	Skill atrophy concerns; misplaced trust in AI-generated artifacts
Kanban	Predictive flow analytics and WIP limit recommendations	May enable earlier detection of flow disruption and improve recovery readiness	Blind acceptance of AI signals; erosion of collaborative learning and policy ownership

cross-training and knowledge sharing to avoid single points of failure [20].

Table 3 presents an indicative framework suitability based on project characteristics and resilience needs.

Table 3: Indicative Framework Suitability Based on Project Profile and Resilience Needs

Project Profile	Indicative Framework Fit
Highly volatile requirements; complex domain.	XP
High process dependency (DevOps); urgent flow optimization.	Kanban
Stable product vision; medium-scale team.	Scrum

7 IMPLICATIONS FOR SOFTWARE ENGINEERING EDUCATION AND TRAINING

The findings of this study have direct implications for software engineering education, particularly in preparing learners for AI-augmented and DevOps-oriented development environments. Beyond technical proficiency, resilience in contemporary software practice increasingly depends on cultural, methodological, and adaptive competencies that are not fully addressed by traditional, framework-centric instruction. Agile education should emphasize the underlying cultural and structural mechanisms rather than presenting frameworks as prescriptive processes. Instruction can highlight how Scrum institutionalizes learning through retrospectives [22], how Kanban fosters systems thinking through flow visualization and work-in-progress control [32, 33], and how XP embeds quality through disciplined engineering practices [26, 29]. This emphasis enables learners to understand why practices matter and to adapt them contextually instead of mechanically applying them.

AI-assisted development tools should be framed as collaborative aids rather than productivity shortcuts. Educational activities may integrate AI support for tasks such as test generation, refactoring, and flow analysis, while requiring students to critically evaluate AI-generated output [37]. This approach reinforces the role of human judgment and reflective learning in maintaining resilience within AI-augmented environments.

Experiential learning should also explicitly incorporate failure-tolerant practices. Simulated incidents, evolving requirements, and managed technical debt can be used to teach blameless post-mortems, adaptive planning, and continuous improvement, helping students internalize psychological safety and learning loops as core professional competencies [10]. Recent case--based research further suggests that integrating design--oriented and reflective practices into software engineering curricula can strengthen students' capacity to navigate hybrid and adaptive development contexts [45].

Finally, the analysis supports hybrid pedagogical models that mirror contemporary industry practice. Just as resilient organizations combine Scrum, Kanban, and XP practices [42, 43], educational programs can blend time--boxed projects, continuous-flow exercises, and disciplined engineering assignments. Such hybrid learning environments reflect real-world development conditions and better prepare graduates for continuous learning in rapidly evolving software ecosystems.

8 CONCLUSION AND FUTURE RESEARCH

8.1 Conclusion

This study examined how three widely adopted Agile frameworks-Scrum, Extreme Programming (XP), and Kanban—support cultural resilience under the combined pressures of AI integration and DevOps-driven delivery. Through a structured comparative analysis grounded in integrative literature synthesis, the study showed that each framework cultivates resilience through distinct mechanisms: Scrum institutionalizes learning via time-boxed feedback loops, XP embeds resilience through disciplined engineering practices that reduce the cost of change, and Kanban promotes systemic stability by optimizing flow and exposing bottlenecks.

The analysis further indicates that AI does not replace these resilience mechanisms but modifies their operation. AI-assisted development can reduce cognitive load, accelerate feedback, and enable predictive insight, while simultaneously introducing cultural risks related to over-reliance, skill erosion, and diminished collaborative learning. Cultural resilience therefore, becomes increasingly critical in determining whether AI augments or undermines adaptive capacity.

These findings suggest that no single framework adequately addresses the combined demands of adaptability, stability, and continuous learning in contemporary development environments. Instead, resilience is achieved most effectively through hybrid configurations that integrate complementary strengths. The proposed

Scrumban–XP hybrid illustrates how organizations can combine Scrum's learning structures, Kanban's flow-based controls, and XP's technical rigor—supported by AI-assisted tooling and guided by evolutionary change—to sustain resilience under ongoing disruption [42, 43].

8.2 Limitations and Future Research

This study is intentionally conceptual and does not present original empirical data. While it draws on established empirical research to ground its analysis, the conclusions should be interpreted as analytically informed rather than empirically validated in specific organizational contexts. Consequently, the findings do not claim generalizable performance outcomes.

Future research should extend this work through empirical investigation. Promising directions include practitioner surveys and semi-structured interviews examining how AI-assisted tools influence learning loops, engineering discipline, and flow stability in Scrum, XP, and Kanban teams. In addition, reanalysis of publicly available DevOps performance datasets (e.g., DORA reports) through a cultural resilience lens could clarify associations between hybrid framework adoption and outcomes such as recovery time and change failure rates [3]. Qualitative studies exploring how psychological safety and learning behaviors evolve in AI-augmented, hybrid Agile teams would further strengthen the understanding of how cultural resilience is cultivated in increasingly complex development environments [10].

REFERENCES

[1] Singh, R., et al. (2019). Software Development Process Models Comparison And Assessment Of Degree Of Agility Based On Agile Practices And Performance Implementation On Xp And Scrum. *Int. J. Sci. Technol. Res.*, 8(12).

[2] Darlami, M., & Magar, B. (2025). Agile methodologies, like Scrum and Kanban, have reshaped how software development teams work.... *World Journal of Advanced Research and Reviews*.

[3] Sethupathy, U. K. A., & Kumar, S. (2025). Navigating Continuous Improvement: An In-Depth Analysis of Lean-Agile and DevOps Maturity Models. *Journal of Software Engineering and Applications*, 18(9).

[4] Lengnick-Hall, C. A., Beck, T. E., & Lengnick-Hall, M. L. (2011). Developing a capacity for organizational resilience through strategic human resource management. *Human Resource Management Review*, 21(3), 243-255.

[5] Conboy, K. (2021). How organizational agility can lead to resilience. *ICIS 2021 Proceedings*.

[6] Jansen, M., Chengie, A., El-Jbouly, R., Akaro, D., & Johne, F. (2025). Agile Leadership in Times of Crisis: A Strategic Framework for Organizational Resilience. *OTS Canadian Journal*, 4(7), 14-23.

[7] Carmeli, A., & Gittell, J. H. (2009). High-quality relationships, psychological safety, and learning from failures in work organizations. *Journal of Organizational Behavior*, 30(6), 709-729.

[8] Moe, N. B., Dingsøyr, T., & Dybå, T. (2010). A teamwork model for understanding an agile team: A case study of a Scrum project. *Information and Software Technology*, 52(5), 480-491.

[9] Williams, L., Kessler, R. R., Cunningham, W., & Jeffries, R. (2000). Strengthening the case for pair programming. *IEEE Software*, 17(4), 19-25.

[10] Edmondson, A. (1999). Psychological safety and learning behavior in work teams. *Administrative Science Quarterly*, 44(2), 350-383.

[11] Forsgren, N., Humble, J., & Kim, G. (2018). *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*. IT Revolution.

[12] Storey, M. A., Zimmermann, T., Bird, C., Czerwonka, J., Murphy, B., & Kalliamvakou, E. (2020). The future of software engineering: AI-assisted, collaborative, and resilience-centric. *IEEE Software*, 37(6), 18-25.

[13] Amershi, S., et al. (2019). Software engineering for machine learning: A case study. *Proc. of the 41st Int. Conf. on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 1-10.

[14] Beck, K. (2000). *Extreme Programming Explained*. Addison-Wesley.

[15] Anderson, D. J. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press.

[16] Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6), 1213-1221.

[17] Kuhrmann, M., et al. (2017). Hybrid software development approaches in practice: a survey of results. *Proc. of the 39th Int. Conf. on Software Engineering (ICSE)*, 547-558.

[18] Ahmad, M. O., Markkula, J., & Oivo, M. (2013). Kanban in software development: A systematic literature review. *2013 39th Euromicro Conference on Software Engineering and Advanced Applications*, 9-16.

[19] Forsgren, N., et al. (2018). *Accelerate: The Science of Lean Software and DevOps*. IT Revolution Press.

[20] Ambler, S. W. (2018). Why Agile Coaches Need to Know Both Scrum and Kanban. *PMI Disciplined Agile*.

[21] Saleh, S. M., et al. (2019). Comparative Study within Scrum, Kanban, XP Focused on Their Practices. *Proc. of the 2019 Int. Conf. on Elec., Comput. and Commun. Eng. (ECCE)*.

[22] Weerakun, H. (2024). Analysing and evaluating the effectiveness of the SCRUM methodology for successful project management. *AIP Conference Proceedings*, 3220(1), 050015. https://doi.org/10.1063/5.0234957

[23] Marshburn, D. (2018). Mixed-method study on measuring retrospective effectiveness. *Proceedings of the Southern Association for Information Systems (SAIS)*.

[24] Almseidin, M., et al. (2015). A Comparative Study of Agile Methods: XP versus SCRUM. *Int. J. Comput. Sci. Inf. Technol. Res.*

[25] Peter, S., et al. (Eds.). (2025). *Proc. of the 26th Int. Conf. on Agile Software Development (XP 2025)*. Springer.

[26] Nagappan, N., Maximilien, E. M., Bhat, T., & Williams, L. (2008). Realizing quality improvement through test driven development: results and experiences of four industrial teams. *Empirical Software Engineering*, 13(3), 289-302.

[27] Massimo, E. (2024). Empirical Evaluation of Test-Driven Development on Code Quality and Developer Productivity. *IJCSITR*.

[28] Rafique, Y., & Mišić, V. B. (2013). The Effects of Test-Driven Development on External Quality and Productivity: A Meta-Analysis. *IEEE Transactions on Software Engineering*, 39(6), 835-856.

[29] Salleh, N., Mendes, E., & Grundy, J. (2011). Empirical Studies of Pair Programming for CS/SE Teaching in Higher Education: A Systematic Literature Review. *IEEE Transactions on Software Engineering*, 37(4), 509-525.

[30] Xu, et al. (2023). Recent systematic review on pair programming. *Springer*.

[31] Aljehani, A., & Qureshi, M. R. J. (2025). Agile Method: Challenges and Adaptations for Complex Project Environments. *International Journal of Information Engineering and Electronic Business (IJIEEB)*, 17(3).

[32] Sjøberg, D. I. K., et al. (2018). An Empirical Study of WIP in Kanban Teams. *Proceedings of the ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*.

[33] Thürer, M., Li, S. S., Yang, C., Qu, T., & Huang, G. Q. (2023). Does Regulating Work-In-Process Increase Throughput and Reduce Cycle Times? *Advances in Production Management Systems (APMS 2023)*. Springer.

[34] Ikonen, M., Pirinen, E., Fagerholm, F., Kettunen, P., & Abrahamsson, P. (2011). Exploring Kanban in software companies. *Product-Focused Software Process Improvement (PROFES 2011)*, 266-281.

[35] Rodriguez, S. (2025). AI as Your Pair Programmer. Taking Extreme Programming to the Next Level. *Medium*.

[36] Deniz, B. K., Gnanasambandam, C., Harrysson, M., Hussin, A., & Srivastava, S. (2023). Unleashing developer productivity with generative AI. *McKinsey Insights*.

[37] Yetiştiren, B., Özsoy, I., Ayerdem, M., & Tüzün, E. (2023). Evaluating the Code Quality of AI-Assisted Code Generation Tools: An Empirical Study on GitHub Copilot, Amazon CodeWhisperer, and ChatGPT.

[38] Singla, S. (2024). How AI Keeps Kanban Flowing. *Medium*.

[39] Okuyelu, O. M., & Adaji, O. (2024). AI-Driven Real-time Quality Monitoring and Process Optimization for Enhanced Manufacturing Performance. *Journal of Advances in Mathematics and Computer Science*, 39(4).

[40] Chen, Y., et al. (2023). The Why, When, What, and How About Predictive Continuous Integration: A Simulation-Based Investigation. *IEEE Transactions on Software Engineering*, 49(12), 5223-5249.

[41] Santos, J., da Costa, D. A., McIntosh, S., & Kulesza, U. (2025). On the Need to Monitor Continuous Integration Practices – An Empirical Study. *Empirical Software Engineering*, 30(5), 125.

[42] Almeida, F., & Blaskovics, B. (2024). Approaches for Hybrid Scaling of Agile in the IT Industry: A Systematic Literature Review and Research Agenda. *Information*, 15(10), 592.

[43] Riaz, S. (2012). Comparative Analysis of Software Development Methodologies. *Final Thesis Linköpings universitet*.

[44] Pastrana, M., Ordoñez, H., Cobos, C., & Muñoz, M. (2025). Best Practices Evidenced for Software Development Based on DevOps and Scrum: A Literature Review. *Applied Sciences*, 15(10), 5421.

[45] Bernardez, F. G., Tan, D. M., & Esplanada, P. (2025). Bridging Creativity and Code: A Case Study on Integrating Design Thinking in Software Engineering Education. In *Novel and Intelligent Digital Systems: Proceedings of the 5th International Conference (NiDS 2025)*. Springer.

Self-Training an Image Binary Classification Model using a Multi-Agent System

Angel Sheinen O. Cambarijan
Computer Science
Cebu Institute of Technology-University
Cebu, Philippines
angelcambarijan1009@gmail.com

Jake H. Bajo
Computer Science
Cebu Institute of Technology-University
Cebu, Philippines
jakebajo21@gmail.com

Charlene R. Repuesto
Computer Science
Cebu Institute of Technology-University
Cebu, Philippines
repuestocharlene@gmail.com

Deo D. Talip
Computer Science
Cebu Institute of Technology-University
Cebu, Philippines
cyzero69@gmail.com

Dr. Chris Jordan Aliac
Computer Science
Cebu Institute of Technology-University
Cebu, Philippines
chris.aliac@cit.edu

ABSTRACT

This study introduces TipikAI, a multi-agent system (MAS) that automates the machine learning pipeline for binary image classification models from user prompts. The system uses a mixed configuration of Gemini-2.5 and GPT-4.1 models, with Human-In-The-Loop (HITL) framework checkpoints between phases to ensure accurate large language model (LLM) outputs. The system decomposes the workflow into specialized agents for dataset finding, preprocessing, and iterative model training, supported by helper agents for code validation, link verification, and folder structure enforcement. TipikAI was evaluated in a comparative analysis against LLM tools such as Claude, GPT-4o, and Vertex AI across three binary classification tasks: With or Without Facemask, Pneumonia Detection, and Cat or Dog.

Results show that TipikAI's self-training loop consistently improved model performance across all tasks, achieving final validation accuracies of 99%, 93.8%, and 98.1%, respectively. TipikAI and Claude produced the most reliable results, with Claude achieving the highest overall average accuracy with model accuracies ranging from 95.4% - 99%, while TipikAI demonstrated competitive performance with strong iterative improvements, with GPT-4o and Vertex AI following with GPT-4o and Vertex AI producing inconsistent results across tasks. Overall, the results support TipikAI as a viable system and provides empirical evidence for MAS and prompt-driven AutoML, showcasing capabilities for multi-agent collaboration that can effectively automate the machine learning workflows without needing deep technical background and expertise.

KEYWORDS

Multi-Agent Systems, AutoML, Binary Image Classification, Self-Training, Human-in-the-Loop

1 Introduction

Machine learning models are now widely used in modern applications and can be seen in multiple fields such as from medical and facial recognition to business and sciences [1]. The model training pipeline consists of a constant and predictable set of phases: dataset search, data preprocessing, and the actual model training. However, the pipeline remains difficult for those who lack technical expertise, as each phase requires familiarity with software tools, architecture, and deep learning libraries such as TensorFlow [2], creating barriers for non-experts with classification and model needs. Existing LLM-based tools, when prompted directly, generate training code but leave users responsible for execution environment setup, dependency management, and iterative improvement. Given its well-defined stages, the pipeline is a natural candidate for automation, which can leverage a multi-agent system (MAS) framework, as it has become increasingly seen in modern technology, enabling more efficient and automated workflows to significantly reduce development effort [3].

As such, this study introduces TipikAI, a MAS utilizing a Human-in-the-Loop (HITL) framework built on an agent orchestration workflow software that is designed to automate the creation of a fully trained binary classification model based on user prompts. Through this collaborative process, users can prompt a fully functional image binary classification model with the system autonomously searching for appropriate datasets before data preprocessing and model training, without requiring extensive technical expertise. Binary image classification was selected as the entry point for this system, as its well-defined two-class structure provides a controlled and measurable scope to evaluate whether a MAS can reliably automate the full machine learning pipeline end-to-end. Rather than a limitation, this represents a deliberate design choice to first validate the system's core pipeline automation capabilities before extending to more complex multi-class scenarios. Thus, this research presents a prompt-driven MAS pipeline for an image binary classification model and contributes empirical evidence supporting the automation machine learning workflows through multi-agent collaboration.

2 Methodology

2.1 Research Design

This research employs an experimental and quantitative approach to evaluate a self-training MAS for automating binary image classification workflows. The workflow consists of three of the following phases, based on the machine learning pipeline: dataset search, data preprocessing, and model training.

Each phase of the process is divided into several agents equipped with multiple helpers and tools to ensure successful execution. Following each phase, the user can confirm and make corrections to ensure that the workflow does not deviate from the user prompt task. Each generated binary classification model's accuracies and outputs are measured using standard classification performance metrics. The system would redo and self-train for a better accuracy, with it continuously and dynamically adjusting hyperparameters, epochs, and regenerate model code, until the model training agent determines the final model's accuracy requires no additional training.

2.1.1 Research Type

1. Experimental Research: This research evaluates models generated by four systems (TipikAI, Claude, GPT-4o, and Vertex AI) across identical binary image classification prompts. The comparison is between TipikAI's fully automated pipeline and the current ceiling of LLM-based code generation, which produces training scripts but does not execute, validate, or iteratively refine them autonomously. Manual execution of the generated scripts by the researchers represents the natural completion of what these tools produce. All scripts were run on the same machine under identical hardware and software configurations, and any script that produced a runtime error was re-prompted once without further modification.
2. Quantitative Approach: This research employs a quantitative approach to objectively measure and compare each system's output using numerical classification metrics, which includes validation accuracy, F1-score, and confusion matrix values.

2.1.2 Research Questions & Hypotheses

2.1.2.1 Research Questions. Based on this design, the study investigates with a main research question to guide the development and evaluation of the TipikAI system:

1. How accurate are TipikAI's generated and self-trained models?
2. How does TipikAI's iterative self-training compare to similar systems in terms of model reliability and output quality?

2.1.2.2 Hypotheses. Building on the research question and the experimental design, the following hypotheses were formulated to guide the analysis of outcomes related to the effectiveness and impact of the TipikAI system:

1. *Primary Hypotheses* (H1): Iterative self-training within the multi-agent system will maintain or increase model accuracy.
2. *Secondary Hypotheses (*H2): The proposed multi-agent framework will successfully execute the end-to-end machine learning pipeline with minimal human correction.

2.1.3. Variables

To evaluate the effectiveness of the TipikAI platform, the study considers the following variables:

2.1.3.1. Independent Variable

1. *Hyperparameters*: The parameters that control the binary classification models' learning process. This changes accordingly to the MAS decision to achieve better performance over iterations.
2. *Dataset Source*: The materials the model trains on. A dataset finder in the workflow controls which source it will use relative to the prompt, it is either Kaggle or HuggingFace.
3. *Code Generation System*: The tool used to produce the binary classification model. This is the primary comparative variable in the study, which includes: TipikAI, Claude, GPT-4o, and Vertex AI.

2.1.3.2. Dependent Variable

1. *Accuracy and Loss Curve:* Tools used to understand the model's learning progress and performance growth over time. The Loss curve measures the model's error rate, and Accuracy curve shows the percentage of correct predictions over training epochs.
2. *Confusion Matrix (TN, FP, FN, TP):* The table that evaluates a model's performance by dividing its prediction performance into 4 categories: True Negative, True Positive, False Positive and False Negative. Correct predictions are tallied either in True Negative or True Positive, while the wrong predictions are in False Positive or False Negative.
3. *F1-score:* A machine learning evaluation metric measuring the model's accuracy on a dataset. It is the harmonic mean of precision (how many predicted positives are correct) and recall (how many actual positives were detected).

2.1.3.3. Control Variables

3. *Large Language Models:* The TipikAI workflow utilizes Gemini-2.5 for dataset searching, preprocessing, and helper utilities and GPT-4.1 for model training, producing a strategic design that balance performance, efficiency in computing power, and minimal API calls.
4. *Classification Tasks and Prompts:* The four systems will be given the same set of binary image classification prompts.

2.2 Research Methodology

The study implements a MAS architecture with a Human-In-The-Loop design, as humans still play an important role even in automation workflow, else the results may not be consistent throughout all processes [4]. Thus, the user may need to work closely and step in during necessary events or phases such as validating or correcting the LLM outputs [5, 6]. The system is designed to automate the end-to-end machine learning lifecycle, based on a software engineering development team setting of having designated roles [7].

The approach breaks down each machine learning phase into individual agents [8] to isolate tasks and lessen the risk of deviation. The workflow consists of an agent orchestrator overlooking the team as a project manager and 4 other primary functional agents: Dataset Finder, Dataset Acquisition, Data Preprocessing, and Model Training. Each agent was fed with a clear and detailed prompt to avoid a performance drop for all tools [9] and ensure proper execution of its designated tasks. As MAS is

composed of multiple LLMs and helper agents, it often results in excessive token consumption [10], so to have a maintainable control of tokens and resources, each role is governed by Gemini-2.5, with the exception of the Model Training Agent leveraging GPT-4.1. This setup allows for GPT-4.1's tokens to be primarily focused and allocated towards this phase, lessening risk of hitting limits and exhausting resources should it be used in more agents.

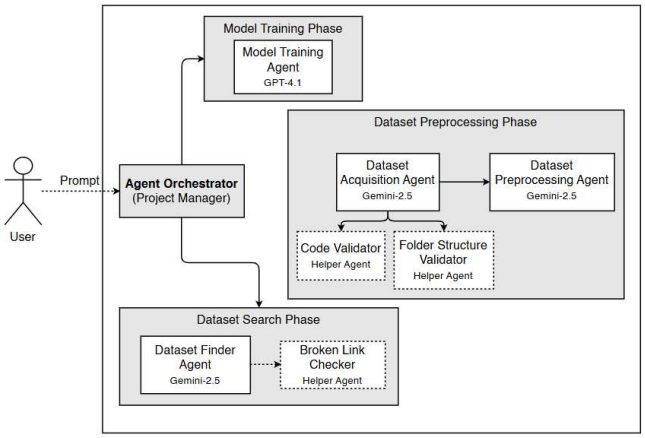


Figure 1: **A high-level architecture of TipikAI System.**

2.2.1 Dataset Search Phase

2.2.1.1 Dataset Finder Agent. The initial phase focuses on identifying and securing high-quality training data, with the agent looking for datasets from Kaggle and HuggingFace sources. The Dataset Finder Agent is prompted with strict heuristic criteria, including dataset size of at least 2,000 images, recency, number of downloads, and topic relevancy. During the initial prototype, the agent would output nonexistent/nonworking URL links. To mitigate this issue, the system utilizes a Broken Link Checker helper agent, designed to perform real-time HTTP requests to verify the existence and accessibility of sourced links. Should the link fail to its validation, the Dataset Finder Agent is to search again for an alternative dataset.

However, the agent could generate or return an unrelated dataset. This was found in a few simulations during prototyping, where a dataset returned did not fully meet the criteria, or the topic was unrelated altogether. This enabled the need to implement the human intervention or correction option, where the user can request a different dataset in a conversational process with the system [11]. Once this process has finished, the found dataset is to be handled by the next phase.

2.2.2 Dataset Preprocessing Phase

2.2.2.1 Dataset Acquisition Agent. The Dataset Acquisition Agent generates a Python script to download the dataset. To maintain execution integrity, the generated code is then passed to the Code Validator helper agent, an agent which performs self-diagnosis by running the given code in a built-in Gemini code execution sandbox environment to detect and resolve syntax or runtime errors before final execution. As raw datasets often arrive with inconsistent hierarchies, a Folder Structure Validator helper agent follows. This helper agent is to enforce 1 of 2 strict, production-ready schemas: a standardized flat format (2 class folders) or a pre-split format (nested train/test/val directories).

If the format is found incorrect, the agent creates a python script to reformat the dataset structure, ensuring the final output is perfectly primed for a binary classification model. Finally, the agent handles any imbalances, specifically in predefined test/validation folders by redistributing samples between test and validation folders, allocating more data to whichever split has fewer samples to ensure balanced evaluation sets.

2.2.2.2 Data Preprocessing Agent. The Data Preprocessing Agent starts once the dataset acquisition is complete, which begins by thoroughly examining the dataset's folder structure, contents, and file format. This stage in the machine learning pipeline is the most time-consuming, where almost 50-80% of it is spent in the preprocessing of data alone [12]. To prepare the agent for its preprocessing code generation and technical accuracy, it utilizes Gemini's built-in Google Search tool to independently gather sources and image preprocessing standards on its own. This serves as the main reference for generating of the data preprocessing code, effectively reducing the possibility of hallucination risks and inaccurate use of techniques.

After the agent has gathered a reliable source for reference, the data preprocessing script generation begins. The script is then given to the Code Validator helper agent, if an error is detected, the validator performs a self-healing cycle, analyzing the stack trace, and regenerating the script until a functional version is confirmed.

2.2.3 Model Training Phase

2.2.3.1 Model Training Agent. The Model Training Agent is designed to iteratively develop and refine binary classification models through a self-training cycle. This agent utilizes the GPT-4.1 model for its high-quality output and code generation performance [13, 14]. Each iteration creates four key artifacts: a Python training script, a Keras model, a detailed metrics JSON file, and a summary text file documenting observations and performance analysis. The Python script, JSON file, and summary text file are for the agent to use and assess how to improve the model training code for the next iteration. The self-training loop continues until the agent determines when to halt based on data-driven criteria that prioritize generalization, efficiency, and performance stability. The criteria consist of the following:

1. Convergence (validation accuracy gain <0.5% and validation loss change <0.01).
2. High performance (validation accuracy >95% and validation loss <0.15), or overfitting plateau (validation loss rising within 2 iterations while train accuracy increases).
3. Diminishing returns (validation accuracy gain <1% over 2 iterations).

Stopping criteria are applied from Iteration 2 onward, as a single iteration cannot reliably distinguish a stable performance peak from an early plateau. At least one refinement cycle is required before halting, after which the agent examines prior output metrics to determine strategic improvements such as hyperparameter tuning, epoch adjustments, or layer unfreezing.

2.3 Data Analysis

In evaluating and comparing the performance of each model generated by the workflow, its training and validation accuracies, losses, and hyperparameters are recorded. Standard machine learning evaluation techniques and metrics will be used to analyze model performance:

1. In measuring overall correctness and confidence, the final validation accuracy and loss will be used.
2. F1-score will be used to evaluate the balance between precision and recall, especially under potential imbalanced datasets. This will be computed from precision and recall on the validation set.

These metrics are standard for binary classification tasks and allow a clear, quantitative comparison of how well each model generalizes from the same prompt and dataset. By analyzing trends in accuracy, loss, F1-score, and confusion matrix entries across iterations, this study can evaluate whether the multi-agent self-training process consistently leads to more accurate and better-balanced classifiers.

3 Results

To evaluate the performance of TipikAI and its generated models, an initial comparative analysis was conducted against three LLM-based tools: Claude, GPT-4o, and Vertex AI. These tools were selected as the most capable and accessible LLM tools currently available.

3.1 Comparative Analysis

Claude, GPT-4o, and Vertex AI were evaluated under a single-pass condition with re-prompts only on error, as these tools produce code but not autonomous execution or refinement with no manual modification from the researchers.

TABLE 1.1 Model Performances – With or Without Facemask

System	Dataset Size	Train_acc	Val_acc	Val_loss
TipikAI	3, 833	1.0	0.99	0.027
Claude	4, 095	0.987	0.99	0.022
GPT-4o	400	1.0	1.0	4.721
Vertex AI	4, 095	0.972	0.979	0.076

A 99% validation accuracy was attained by TipikAI and Claude on this task, while GPT-4o reported an anomaly of 100% but with a validation loss of 4.721, indicating failure in model calibration caused by a drastically smaller dataset of 400 images after it replaced the originally sourced dataset (the result is retained for transparency but excluded from aggregate comparisons) and lastly Vertex AI performed competitively at 97.9%.

TABLE 1.2 Model Performances – Pneumonia Detection

System	Dataset Size	Train_acc	Val_acc	Val_loss
TipikAI	5, 856	1.0	0.938	0.285
Claude	5, 856	0.954	0.962	0.115
GPT-4o	5, 856	0.963	0.687	0.757
Vertex AI	5, 856	0.923	0.919	0.225

The systems all utilized the same dataset source, which includes a significant class imbalance, where pneumonia-positive cases are approximately 74% of the training set. Claude produced the strongest result at 96.2% validation accuracy, followed by TipikAI at 93.8% and Vertex AI at 91.9%. GPT-4o produced the weakest result at 68.7% validation accuracy and a high validation loss of 0.757, indicating that its single-pass generated code did not effectively address the class imbalance, resulting in a model biased toward the majority class.

TABLE 1.3 Model Performances – Cat or Dog

System	Dataset Size	Train_acc	Val_acc	Val_loss
TipikAI	10, 000	1.0	0.981	0.068
Claude	23, 262	0.990	0.990	0.024
GPT-4o	23, 262	0.890	0.832	0.387
Vertex AI	23, 262	0.792	0.808	0.414

For the Cat or Dog task, it should be noted that TipikAI's 98.1% validation accuracy was achieved on a 10,000-image dataset, while Claude, GPT-4o, and Vertex AI all trained on the same and substantially larger 23,262-image dataset. The performance gap between TipikAI and Claude of 98.1% versus 99.0%, must be interpreted with the context of the dataset size difference rather than only as a direct measure of code generation quality. GPT-4o and Vertex AI's weaker results of 83.2% and 80.8% respectively, indicates that access to more data alone did not guarantee better outcomes without an iterative correction mechanism.

Across the three classification prompts, TipikAI and Claude consistently produced the most reliable results, achieving average validation accuracies of 96.97% and 98.07% respectively. The 1.1% gap between the two systems is mostly attributable to the Pneumonia Detection task, with TipikAI matching Claude exactly on Facemask and approaching 0.9% on Cat or Dog despite the difference in dataset size. On the other hand, GPT-4o and Vertex AI performed inconsistently across the three tasks, with GPT-4o producing the weakest result overall at 68.7% on Pneumonia and a score of 100% on Facemask.

3.2 TipikAI Iteration-Wise Results per Prompt

To further examine the effectiveness of TipikAI's self-training mechanism, an iteration-wise analysis was conducted for each classification task. By analyzing these results, the study highlights how the system refines model performance over successive iterations, demonstrating improvements in generalization, reduction of classification errors, and convergence toward optimal performance.

TABLE 1.4 Iteration-Wise Performance – With or Without Facemask

Iteration	Train_acc	Val_acc	Train_loss	Val_loss
1	0.97	0.98	0.07	0.048
2	1.0	0.99	0.0001	0.027

TABLE 1.5 Confusion Matrix and F1-Score – With or Without Facemask

Iteration	Confusion Matrix (TN, TP, FN, FP)	F1-Score
1	(372, 6, 6, 382)	0.9845
2	(375, 4, 3, 384)	0.9910

The baseline model employed a frozen MobileNetV2 architecture with a learning rate of 0.0005 and 0.3 dropout across 15 epochs on a 3,833-image dataset, achieving 98.43% validation accuracy and a validation loss of 0.048. The near-perfect class balance of the dataset meant that no class weighting was required. In Iteration 1,

the model produced 6 false negatives and 6 false positives out of 766 validation samples, yielding an F1-score of 0.9845. In Iteration 2, the agent partially unfroze the top 20 layers of MobileNetV2, reduced dropout from 0.3 to 0.2, and lowered the learning rate from 0.0005 to 0.0001, raising validation accuracy to 99.09% and improving the F1-score to 0.9910 with false negatives and false positives each reduced to below 5.

The 100% training accuracy in Iteration 2 is consistent with fine-tuning a pretrained model on a well-balanced and relatively small dataset of 3,833 images. This is not indicative of problematic overfitting, as validation accuracy improved alongside it and the validation loss dropped from 0.048 to 0.027, confirming that the additional capacity introduced by layer unfreezing was absorbed productively rather than causing overfit on the balanced dataset.

TABLE 1.6 TipikAI Iteration-Wise Performance – Pneumonia Detection

Iteration	Train_acc	Val_acc	Train_loss	Val_loss
1	0.936	0.917	0.155	0.195
2	1.0	0.938	0.0005	0.285

TABLE 1.7 Confusion Matrix and F1-Score – Pneumonia Detection

Iteration	Confusion Matrix (TN, TP, FN, FP)	F1-Score
1	(187, 532, 50, 15)	0.9424
2	(179, 562, 20, 23)	0.9632

TipikAI's Pneumonia Detection models presented the most complex self-training behavior across all three prompts, driven primarily by the 5,856-image dataset's significant class imbalance of approximately 74% pneumonia-positive cases. In Iteration 1, a frozen MobileNetV2 with a learning rate of 0.001 and 0.3 dropout across 15 epochs was used, with class weights proactively applied from the outset, producing a validation accuracy of 91.71% and an F1-score of 0.9424 with 50 false negatives and 15 false positives out of 784 validation samples. In Iteration 2, the agent unfroze the last 20 layers, increased dropout to 0.4, reduced the learning rate to 0.0001, and extended training to 20 epochs with early stopping, improving validation accuracy to 93.88% and F1-score to 0.9632 with false negatives reduced from 50 to 20, a clinically meaningful gain in positive case detection without significantly sacrificing precision.

While training accuracy reached 100% in Iteration 2, the concurrent improvement in validation accuracy and the reduction of false negatives across the 784 validation samples confirm that the model continued to generalize meaningfully rather than simply memorizing the 5,856-image training set, with the mild validation loss increase from 0.195 to 0.285 kept in check by the early stopping callback.

TABLE 1.8 Iteration-Wise Performance – Cat or Dog

Iteration	Train_acc	Val_acc	Train_loss	Val_loss
1	0.95	0.98	0.11	0.047
2	1.0	0.981	1.149	0.068

TABLE 1.9 Confusion Matrix and F1-Score – Cat or Dog

Iteration	Confusion Matrix (TN, TP, FN, FP)	F1-Score
1	(771, 16, 16, 797)	0.9803
2	(773, 15, 14, 798)	0.9821

TipikAI's Cat or Dog models demonstrated the fastest convergence across all three prompts, reflecting the task's well-balanced 10,000-image dataset and the relative visual distinctiveness of the two classes. The baseline model used a frozen MobileNetV2 with a learning rate of 0.001 and 0.3 dropout across only 12 epochs, yet achieved 98.00% validation accuracy and a notably low validation loss of 0.047, with 16 false negatives and 16 false positives out of 1,600 validation samples yielding an F1-score of 0.9803. In Iteration 2, the agent applied a conservative refinement strategy by unfreezing the top 30% of MobileNetV2 base layers and lowering the learning rate to 0.0002, producing only marginal gains of 98.0% to 98.1% validation accuracy and an F1-score improvement from 0.9803 to 0.9821.

The 100% training accuracy in Iteration 2, paired with the marginal validation gain and a slight validation loss increase from 0.047 to 0.068 across the 10,000-image dataset, indicates that the model had reached a near-ceiling baseline with minimal remaining room for improvement. The agent correctly identified this through the convergence stopping condition, concluding that further unfreezing or additional epochs would risk overfitting with no meaningful accuracy return.

The training loss spike from 0.11 to 1.149 in Iteration 2 is attributable to the partial unfreezing of MobileNetV2 layers temporarily disrupting previously learned feature representations, a known behavior in gradual unfreezing strategies which does not propagate to validation loss given the reduced learning rate constraining weight updates.

3.3 Discussion

The primary hypothesis H1 is supported, as presented through the experimental results. The system's self-training criteria prevented further iterations, only stopping at the second iteration, which produced measurable improvements in both validation accuracy: from 98.43% to 99.09% on Facemask, from 91.71% to 93.88% on Pneumonia Detection, and from 98.00% to 98.19% on Cat or Dog. These results confirm that the self-training loop consistently improved model performance over the baseline while the stopping criteria ensured iteration halted precisely when another iteration could risk overfitting and degrading accuracy.

The system halting at Iteration 2 across all three tasks is explained by both the minimum iteration policy and the specific stopping criteria met upon evaluation. The minimum iteration policy ensures at least one refinement cycle is completed before halting, allowing the agent to attempt architectural adjustments and confirm whether further optimization yields meaningful improvement. For Cat or Dog, the convergence criterion fired, as the validation accuracy gain of 0.10% fell well below the 0.5% threshold. For Facemask, the high-performance criterion was satisfied at Iteration 2, with validation accuracy at 99.09% and validation loss at 0.027, well below the 0.15 threshold. For Pneumonia Detection, the diminishing returns condition was met, as the dataset's class imbalance imposed a practical performance ceiling that limited further meaningful gains beyond 93.88%.

The high variance in GPT-4o and Vertex AI results, from 68.7% to 97.9%, highlights that code generation alone does not guarantee reliable model outcomes without an automated execution and refinement layer. Without such a layer, generated scripts have no mechanism to detect, diagnose, or correct performance failures across iterations.

The secondary hypothesis H2 is partially supported. Full automation was achieved in the preprocessing and model training phases across all three prompts, with no corrections required. However, the dataset search phase required human intervention once during the Cat or Dog task, as the initially sourced dataset contained .parquet files incompatible with the system's image-only pipeline. This confirms the continued necessity of the HITL framework specifically at the dataset search phase.

4 Conclusion

This study presents TipikAI, a multi-agent system that automates the end-to-end pipeline for training image binary classification models from a user prompt, with a Human-In-The-Loop framework to allow for human correction between phases. The system breaks down the machine learning pipeline into specialized agents for dataset finding, preprocessing, and model training phases, supported by multiple helper agents. Through a comparative analysis of the system and three other tools across all the three classification tasks, Claude achieved the highest overall average accuracy, with validation accuracies ranging from 95.4% to 99.0%, marginally outperforming TipikAI. While TipikAI consistently produced well-generalized models, achieving validation accuracies of 99% (Facemask), 93.8% (Pneumonia Detection), and 98.1% (Cat or Dog), and matching Claude's performance on the Facemask task, with GPT-4o and Vertex AI followed behind, demonstrating lower and less consistent results across the same prompts, with validation accuracies ranging from 68.7% to an anomaly of 100.0% and 80.8% to 97.9%, respectively.

The study demonstrates strong foundational results, with improvements anticipated on more capable hardware. Upgrading beyond the current 16GB RAM would enable broader hyperparameter exploration, larger batch sizes, and more complex architecture. Additionally, expanded API allocation for full GPT-4.1 integration across all agents may further improve consistency and performance. The self-training loop consistently improved model performance across all tasks, with stopping criteria correctly halting iteration when convergence, high performance, or diminishing returns were met, preventing overfitting while maximizing accuracy gains.

The 100% training accuracy observed in Iteration 2 across all prompts warrants acknowledgment. While validation metrics confirm the models generalized well rather than simply overfitting as evidenced by stable or improving validation loss and F1-scores, this behavior is partly explained by the relatively small dataset sizes and the fine-tuning dynamics of resuming from a pretrained checkpoint at a reduced learning rate, both of which accelerate convergence on the training distribution.

Despite these constraints, the study proves TipikAI as a viable system and as empirical evidence for automated machine learning workflows. To fully realize the system's potential for advanced ML architecture exploration and thorough optimization, the researchers recommend for future research investigates on deploying the system on stronger infrastructure either in hardware with enhanced computational resources or using cloud-based platforms. Additionally, investigating pure GPT-4 configurations across all agents may improve consistency and performance compared to the current mixed models approach.

REFERENCES

[1] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar, "A Survey of Deep Learning and Its Applications: A New Paradigm to Machine Learning.," Arch. Comput. Methods Eng., vol. 27, no. 4, p. 1071, Sep. 2020, doi: 10.1007/s11831-019-09344-w.

[2] H. Jin, F. Chollet, Q. Song, and X. Hu, "AutoKeras: An AutoML Library for Deep Learning," J. Mach. Learn. Res., vol. 24, no. 6, pp. 1–6, 2023.

[3] Q. Wu et al., "AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversations," presented at the First Conference on Language Modeling, Aug. 2024. Accessed: Nov. 16, 2025. [Online]. Available: https://openreview.net/forum?id=BAakY1hNKS

[4] A. Crisan and B. Fiore-Gartland, "Fits and Starts: Enterprise Use of AutoML and the Role of Humans in the Loop," in Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, in CHI '21. New York, NY, USA: Association for Computing Machinery, May 2021, pp. 1–15. doi: 10.1145/3411764.3445775.

[5] E. Mosqueira-Rey, E. Hernández-Pereira, D. Alonso-Ríos, J. Bobes-Bascarán, and Á. Fernández-Leal, "Human-in-the-loop machine learning: a state of the art," Artif. Intell. Rev., vol. 56, no. 4, pp. 3005–3054, Apr. 2023, doi: 10.1007/s10462-022-10246-w.

[6] D. Xin, L. Ma, J. Liu, S. Macke, S. Song, and A. Parameswaran, "Accelerating Human-in-the-loop Machine Learning: Challenges and Opportunities," in Proceedings of the Second Workshop on Data Management for End-To-End Machine Learning, in DEEM'18. New York, NY, USA: Association for Computing Machinery, Jun. 2018, pp. 1–4. doi: 10.1145/3209889.3209897.

[7] S. Hong et al., "METAGPT: META PROGRAMMING FOR A MULTI-AGENT COLLABORATIVE FRAMEWORK," Jan. 2024, Accessed: Sep. 07, 2025. [Online]. Available: http://hdl.handle.net/10754/700016

[8] Z. Wang et al., "Multi-Agent Automated Machine Learning," in 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2023, pp. 11960–11969. doi: 10.1109/CVPR52729.2023.01151.

[9] K. Goyle, Q. Xie, and V. Goyle, "DataAssist: A Machine Learning Approach to Data Cleaning and Preparation," Jul. 17, 2023, arXiv: arXiv:2307.07119. doi: 10.48550/arXiv.2307.07119.

[10] M. Hulsebos, W. Lin, S. Shankar, and A. Parameswaran, "It Took Longer than I was Expecting: Why is Dataset Search Still so Hard?," in Proceedings of the 2024 Workshop on Human-In-the-Loop Data Analytics, in HILDA '24. New York, NY, USA: Association for Computing Machinery, Jun. 2024, pp. 1–4. doi: 10.1145/3665939.3665959.

[11] F. Lin, S. Chen, R. Fang, H. Wang, and T. Lin, "Stop Wasting Your Tokens: Towards Efficient Runtime Multi-Agent Systems," Oct. 30, 2025, arXiv: arXiv:2510.26585. doi: 10.48550/arXiv.2510.26585.

[12] M. Bilal, G. Ali, M. W. Iqbal, M. Anwar, M. S. A. Malik, and R. A. Kadir, "Auto-Prep: Efficient and Automated Data Preprocessing Pipeline," IEEE Access, vol. 10, pp. 107764–107784, 2022, doi: 10.1109/ACCESS.2022.3198662.

[13] B. Yetiştiren, I. Özsoy, M. Ayerdem, and E. Tüzün, "Evaluating the Code Quality of AI-Assisted Code Generation Tools: An Empirical Study on GitHub Copilot, Amazon CodeWhisperer, and ChatGPT," arXiv.org. Accessed: Dec. 07, 2025. [Online]. Available: https://arxiv.org/abs/2304.10778v2

[14] Y. Fu, H. Peng, T. Khot, and M. Lapata, "Improving Language Model Negotiation with Self-Play and In-Context Learning from AI Feedback," May 17, 2023, arXiv: arXiv:2305.10142. doi: 10.48550/arXiv.2305.10142.

FIRE!: A Fire Simulation Game that Educates Filipinos on Fire Safety

Matthew Josh Benedict R. Benito
Department of Information Systems and Computer Science
Ateneo de Manila University
Quezon City, Philippines
matthew.benito@student.ateneo.edu

Enrique Jose Stefan P. Gomez
Department of Information Systems and Computer Science
Ateneo de Manila University
Quezon City, Philippines
enrique.gomez@student.ateneo.edu

Nate Brevin A. Que
Department of Information Systems and Computer Science
Ateneo de Manila University
Quezon City, Philippines
nate.que@student.ateneo.edu

ABSTRACT

Fire incidents are common in the Philippines which cause significant damage to human lives and properties. To combat this, FIRE! is developed to enhance the fire safety knowledge among Filipinos, including those who attend fire drills annually. This is a 3D first-person game that involves fire incidents in residential areas. The player is a civilian who must prevent potential fires, extinguish the emerging fire, and evacuate when necessary. To evaluate the game, the participants took a pre-test to gauge their knowledge, followed by a 25-minute game testing session. A post-test then evaluated the testers again and asked how they viewed the game. Based on the results, a significant increase in fire safety knowledge was observed among all testers, including the regular fire drill participants, and the game was deemed realistic and fun. After testing, the researchers added non-player characters (NPCs) in the neighborhood, restructured the neighborhood, made fire growth depend on fuel type, and made quality-of-life updates. With this, the researchers recommend further testing, especially with fire safety professionals to further validate the game's realism. Other recommendations to the game include making more levels and adding a Filipino language option.

KEYWORDS

fire safety, simulation, serious game, low-income neighborhood

1 Introduction

In recent years, there has been an increase in frequency and intensity of fire incidents, primarily due to climate change as reported by the International Association of Fire and Rescue Services. Beyond the immediate consequences of fires such as injuries and property damage, there are also long-term implications for ecosystems, air quality and community resilience. This widespread threat leaves every level of society vulnerable, highlighting the need for civilians to gain the knowledge and skills for such crises [4].

Traditional methods of fire safety education, though valuable, may not always capture the dynamic nature of real-life emergencies. This study desires to capitalize on that by developing a fire outbreak simulator video game. Drawing inspiration from similar games, this project seeks to immerse players in a realistic simulation where they can experience the challenges involved in handling a fire emergency.

1.1 Context of the Study

Fire incidents are one of the more common disasters that citizens experience in the Philippines. It was reported that about 15,679 fire related incidents were recorded during 2023, 20.7% higher compared to 2022 [9]. The damages caused by these amounted to approximately P7.8 billion, almost four times bigger compared to 2022, but the casualty count was lower [1]. Most fire incidents primarily happen in residential areas, and the general cause is due to electrical ignition [1, 9].

The Philippines has implemented some ways to combat this issue, such as celebrating "Fire Prevention Month" every March as it is one of the hottest months [6]. There are also lectures on disaster preparedness that are offered publicly for free.

With the high usage of computers and the prominence of gaming in the country [7], the researchers developed a video game to educate Filipinos on fire safety and further this cause.

1.2 Research Objectives

The researchers aim to contribute to solving the problem by developing a first-person simulation game that involves civilians' encounters with fires. This game is made to supplement fire drills in promoting fire safety by giving them a safe platform to experience fire outbreaks. In this research, this game is evaluated based on the scores of the participants in the pre-test and post-test, as well as their feedback on the game's educational capabilities, realism, and entertainment value.

1.3 Research Questions

In accordance with the objectives specified above, the researchers conduct this study with the purpose of answering the following questions:

1. How effective is the game in educating and equipping Filipinos about fire safety, even for those who participate in fire drills at least once a year?
2. How realistic is the game in emulating fire scenarios, and how does it make the players feel?
3. How is the game fun for the players while it fulfills its purpose of being educational?

1.4 Scope and Limitations

This study only included the most common types of fires found in the Philippines, which are electrical and grease fires, as well as other fires on solids known as Class A fires.

The game fostered an interactive learning environment where players may acquire knowledge and skills related to fire safety protocols before and during such crises. This was done through the tutorial, text overlays, and the overall experience.

On the other hand, the game was limited to simulating these emergencies in urban environments. Due to technical constraints, the game may not fully replicate all aspects of these scenarios with differences in dynamic fire behavior, absence of environmental factors and lack of complex buildings.

1.5 Significance of the Study

This study aims to capitalize on how games, specifically serious games, can have educational purposes. Using this medium, people are more motivated to learn [10] as they engage in safe environments to experiment without dealing with the real-world consequences [3].

Games also allow their players to reflect on their experiences, helping them learn better through playing simulations compared to traditional means [11], which was also found in Hsieh et al.'s study [2].

There are existing video games on this topic, but these only focus on evacuations [5, 8, 12]. Civilians should also learn how to prevent fires, and they should also learn how to fight fires as their first response (Ronald Galicia, personal communication, March 8, 2024). Thus, developing a video game that includes these can provide more value towards promoting fire safety.

2 Methodology

2.1 Game Design

The player plays the role of a civilian trying to prevent fires from happening in their home. After some time, a fire will emerge and they are to extinguish it. If unsuccessful, they must quickly and safely evacuate.

The game takes place in a small community composed of small houses. Due to the realism that the developers aim to emulate, there were creative restrictions on the level design in terms of their structures and layout.

2.2 Game Development

The developers implemented the game using *Unity 2022.3.31f1*, and they created some of the visual assets using Blender and Adobe Photoshop. Other models were from the Unity Asset Store and from websites that offer them for free. The sound effects in the game were also from these free sources.

To coordinate the development across the team, the researchers used GitHub for version control. This development process was also divided into sprints to focus on specific features. After multiple iterations of the game, this first version of the game was released in itch.io privately for people to play online as part of the playtesting for this study.

2.3 FIRE! 1.0

FIRE! situates players in a location resembling a Philippine low-class neighborhood wherein they will have to apply the common procedures in fire prevention and firefighting done.

Like most games, the player has different status meters to be mindful of when playing. These would be the (1) Health bar, (2) Oxygen bar, and (3) Stamina bar.

If either the health bar or oxygen bar reaches zero from taking enough burn damage from the fire or inhaling too much smoke, then the player would fail the level correspondingly. On the other hand, the stamina bar depletes if the player decides to do tiresome movements such as running, rolling and crawling, and this recharges over time when the player is idle.

The game involves preventing fires from happening in the first place through reassembling their electrical plugs, unplugging unused appliances and blowing out unused candles. However, should one accomplish all measures of fire prevention available to them, fires will still break out as there are many external factors that can cause it.

As previously stated, the game contains three types of fires:

1. Electrical fire (Class C) - most common type, which comes from electrical ignitions,
2. Grease fire (Class K) - usually found in kitchens wherein the cause of fire is kitchen oil, and
3. Class A fire - fire on solids, which does not fall under the previous two categories.

Firefighting tools are present in the player's house, such as a water bucket, a fire-resistant material (e.g. concrete block) and a randomized type of fire extinguisher. Hence, this would mean that in some cases, the fire could not be extinguished.

An example of which would be if an electrical fire is occurring, and the player only has a class A extinguisher. Using this wrong kind of extinguisher or water on the electrical fire will only intensify the fire. Therefore, the player must recognize what kind of fire is transpiring before even attempting to extinguish it through inferring the object it arose from.

If the player is unable to put out the fire, they should evacuate the house and retreat to the basketball court, especially if the smoke inside gets too intense.

Conversely, extinguishing the fire or evacuating successfully would mean that the player completed the level. One to three stars will be awarded to the player depending on the time they completed the level, health points remaining, and existence or non-existence of firefighting tools available to take out the fire.

In summary, the player would do fire prevention, then firefighting and if unsuccessful, evacuation. Text overlays guide the players to follow these phases, ensuring they understand the objectives while reinforcing fire safety knowledge.

2.4 Testing and Deployment

To assess the game in terms of the objectives of this study, the researchers conducted game testing sessions across 15 participants. The game testers were Filipinos aged 18 years old and above who were recruited through Facebook.

Upon accepting the invitation to test the game, the game testers answered a pre-test for the researchers to gauge their fire safety knowledge and preparedness. This contained objective questions on fire safety, as well as subjective questions on how they perceive themselves on the matter.

In the playtesting session, the testers started with the tutorial to get themselves acquainted with the game's controls and mechanics. After which, they got to play through as many runs as they could in 25 minutes. For each time they played through the game, the remaining HP and time completion they had were recorded to gauge their performance.

After the session, the players were asked to answer a post-test, which contained the same questions as the pre-test, but it also asked for their insights about the experience.

3 Results

In the results presented in this chapter, each game tester is assigned a number from 1 to 15 for anonymity.

3.1 Pre-Test Results

3.1.1 Categorization of Testers

In the pre-test, the researchers aimed to categorize the testers into two groups based on the following definitions:

1. A regular fire drill participant, or in short: regular (plural: regulars), is someone who partakes in the said drills at least once a year; and
2. An irregular fire drill participant, or in short: irregular (plural: irregulars), attends less often than regulars.

From these definitions, the researchers identified nine regulars and six irregulars who participated in this study.

3.1.2 Self-Evaluations on Fire Safety Knowledge and Preparedness

The researchers aimed to gauge how the testers perceive their own fire safety knowledge and preparedness with a rating from 1 to 5 with 1 being the lowest and 5 being the highest.

Tester	Category	Rating
1	Regular	3
2	Regular	4
3	Regular	3

Table 1.1: **Pre-Test Self-Evaluations of Testers 1-3**

Tester	Category	Rating
4	Regular	3
5	Irregular	3
6	Irregular	1
7	Regular	3
8	Regular	3
9	Regular	3
10	Irregular	4
11	Regular	3
12	Regular	2
13	Irregular	2
14	Irregular	2
15	Irregular	4
AVERAGE		2.87
MEDIAN		3

Table 1.2: **Pre-Test Self-Evaluations of Testers 4-15**

The testers generally think that their fire safety knowledge and preparedness is satisfactory since their ratings had an average of 2.87 and a median of 3 out of 5.

3.1.3 Pre-Test Scores

The researchers also gauge the testers' knowledge through an objective questionnaire.

Tester	Category	Score
1	Regular	10
2	Regular	7
3	Regular	6
4	Regular	7
5	Irregular	8
6	Irregular	4
7	Regular	8
8	Regular	10
9	Regular	8
10	Irregular	8
11	Regular	8
12	Regular	3
13	Irregular	10
14	Irregular	2
15	Irregular	7
AVERAGE		7.07
MEDIAN		8

Table 2: **Pre-Test Scores of Testers**

With an average of 7.07 and median of 8, the testers generally passed the test. Most mistakes involve identifying

which fire extinguisher is effective in putting out each type of fire and identifying the best ways to prevent fires.

3.2 Playtest Statistics

There was a total of 63 runs that were accomplished by the 15 testers. 28 of those runs were wins, with 13 extinguishes and 15 escapes performed, resulting in a 44.44% win-rate.

Each run was classified into several categories based on the type of fire involved, and the result that the player got. Below is the table that shows the frequency of each category during the playtest, which is followed by a pie chart showing each of their percentages across all runs.

Run Classification	Frequency
Deaths to Class A	14
Extinguish Class A	12
Escapes from Class A	4
Deaths to Grease Fire	8
Extinguish Grease Fire	0
Escapes from Grease Fire	3
Deaths to Electrical Fire	11
Extinguish Electrical Fire	1
Escapes from Electrical Fire	2
Unidentified Fire Escapes	6
Unidentified Fire Deaths	2

Table 3: **Results of Play Test**

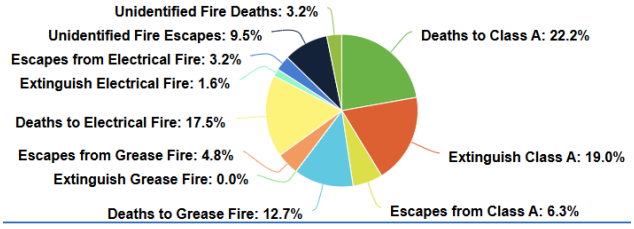


Figure 1: **Results of Play Tests**

As seen in Table 3 and Figure 1, the number of extinguishes and escapes for class A fires are greater than the number of deaths caused by it. However, the deaths from grease and electrical fires heavily outnumber the extinguishes and escapes from both, with only one extinguishing an electrical fire.

3.3 Playtest Observations

Players mostly understood and followed the core game loop of the game; the fire prevention phase, into the firefighting phase and finally the fire escape phase if left with no other option. However, it was found that several testers had difficulty with certain aspects and functionalities of the game.

Firstly, testers failed to use the roll function correctly as it would get cancelled when the player presses any movement key. It would also consume stamina despite not performing the action, hence a few testers were confused and lost due to this.

Secondly, testers had trouble aligning the water bucket with the faucet. They were also unaware that the water bucket can be filled when it is held, as there was no message or indicator in the game informing them of being able to do so.

Finally, testers could not grab some plugs despite being at a reasonable range and aiming directly at them.

Considering these struggles, changes were implemented in the game to address these concerns to provide a smoother and better experience for future players.

3.4 Post-Test Results

3.4.1 Post-Test Scores

The testers' knowledge was gauged again after game testing using the same objective questionnaire given in the pre-test.

Tester	Score
1	9
2	9
3	9
4	9
5	8
6	6
7	10
8	10
9	9
10	10
11	10
12	9
13	9
14	10
15	9
AVERAGE	9.07
MEDIAN	9

Table 4: **Post-Test Scores**

With an average of 9.07 and median of 9, the testers performed well in the test. Most mistakes committed in the test involve identifying which are good ways to prevent fires, but they accurately matched the fire extinguishers to their corresponding types of fires unlike in the pre-test.

3.4.2 Post-Test Evaluation Scores

In the post-test, the researchers also added statements about the game that the participants rated from 1 to 5 with 1 being "Definitely No" and 5 being "Definitely Yes."

For the statement "the game increased my knowledge in fire safety and preparedness," the average rating was a 4.26 with a median of 4. This means that the participants think they gained knowledge after playing the game.

In relation to the game's realism, the average score was a 4.06 and a median of 4. This means that the participants perceived the game to portray a fire realistically. Additionally, they have somewhat felt heightened emotions as they gave an

average rating of 3.6 for feeling anxious, 3.53 for feeling afraid, and 3.8 for other emotions such as being surprised.

Lastly, the participants enjoyed the game somehow, giving the game an average score of 3.86 with a median of 4.

3.4.3 Post-Test Insights

In the post-test, the participants also had to answer open-ended questions about the game and their experience.

In answering the question "What insights or facts did you learn from playing the game," most of the participants responded that not all fires can be extinguished by one fire extinguisher. They learned that there are different types of fire extinguishers that are used for specific kinds of fires. This shows a lack of education on firefighting specifically among civilians. Another common answer the participants gave is that they learned about how crucial fire prevention is.

As for the question "What parts of the game were realistic," the most common answers the participants gave is that the fire extinguishers are very akin to how they are used in real life. They also pointed out the realism with the rapid growth of the fires in the game, as well as how realistic it is for a fire to emerge from unattended electric plugs or candles.

3.5 Comparison Between Pre-Test and Post-Test Results

3.5.1 Comparison Among All Testers

This subsection aims to compare each tester's performance in the objective portions of the pre-test and the post-test to examine the game's impact on their fire safety knowledge.

Tester	Category	Pre-Test	Post Test
1	Regular	10	9
2	Regular	7	9
3	Regular	6	9
4	Regular	7	9
5	Irregular	8	8
6	Irregular	4	6
7	Regular	8	10
8	Regular	10	10
9	Regular	8	9
10	Irregular	8	10
11	Regular	8	10
12	Regular	3	9
13	Irregular	10	9
14	Irregular	2	10
15	Irregular	7	9
AVERAGE		7.07	9.07
MEDIAN		8	9

Table 5: **Pre-Test vs Post-Test Scores**

The researchers conducted hypothesis testing for difference between two means using a paired t-test on these groups of data. The null hypothesis is that the pre-test and post-test scores are statistically equal, while the alternative hypothesis is that the pre-test scores are less than the post-test scores.

With a significance level of 0.05, the calculated p-value is 0.0030, which is less than the significance level, so the null hypothesis is rejected in favor of the alternative hypothesis. This means that there is a statistically significant increase in the testers' scores after playing the game.

3.5.2 Comparison Among Regular Fire Drill Participants

This subsection examines the game's effectiveness in increasing the scores of the regulars after game testing.

Tester	Pre-Test	Post Test
1	10	9
2	7	9
3	6	9
4	7	9
7	8	10
8	10	10
9	8	9
11	8	10
12	3	9
AVERAGE	7.44	9.33
MEDIAN	8	9

Table 6: **Pre-Test vs Post-Test Scores Among Regulars**

The researchers used the same testing method with the same parameters on the data from the regulars alone. Using a significance level of 0.05, the researchers calculated a p-value of 0.0102, which is also less than the significance level. This entails that the null hypothesis is rejected in favor of the alternative hypothesis, meaning there is also a significant increase in the scores of the regulars after playtesting.

4 Conclusion

4.1 Effectivity on Learning

Based on the results from the objective questionnaire, it was found that there is a statistically significant increase in the scores of the participants from the pre-test to the post-test. This conclusion was also reached when the same testing method was conducted using the scores of the regulars. This means that the game may have the ability to increase the fire safety knowledge among its players, regardless of their frequency of participating in fire drills.

4.2 Game Realism

From the insights gathered in the post-test, the game received a 4.06 rating with a median of 4 out of 5 in terms of the game's realism. Additionally, they have somewhat felt heightened emotions. This means that the participants perceived the game to portray a somewhat realistic fire outbreak simulation with most of the participants mentioning that the fire growth mechanic was portrayed realistically. The fire extinguishers and the primary causes of fires were also said to have contributed to the game's realism. However, the testers are not experts in this field, so the game was only perceived to be realistic among civilians who may have not experienced fires firsthand.

4.3 Game Entertainment Value

Based on the testers' evaluations of the game after testing, the average rating that the game received was 3.86 with a median of 4 out of 5. This means that the participants enjoyed the game somehow. From our qualitative questions in the post-test, most participants mentioned that they found the core gameplay loop itself most enjoyable. Some also mentioned that the fire prevention part of the game was also enjoyable because they treated it as a speedrun aspect of the game.

4.4 FIRE! 2.0

Having tested the first version of the game, the developers considered comments and applied changes to the game. Other than the quality-of-life changes implemented mentioned in section 3.3, significant updates to the game had also been implemented in this succeeding version.

The large neighborhood was restructured, and all houses were made functional, instead of them being just for aesthetic purposes. Additionally, this iteration includes the presence of NPCs who act as the other residents in the neighborhood, which was a feature adopted from Oe et al.'s study [5].

NPCs are randomly situated in the houses found in the neighborhood. Their existence primarily helps to simulate the crisis as often multiple people are involved. Before the fire starts, they just roam around their houses doing tasks like cooking or watching TV. However, when the fire starts and the alarm bell rings, they will be alerted and will try to locate the fire. Once they see it or its smoke, they may either choose to fight the fire or evacuate immediately.

Moreover, this iteration implements a more realistic fire behavior, with the spread of the fire being dependent on the type of object it is burning on. For instance, the fire spreads more slowly on metal surfaces, and more quickly on flammable materials like foam. This addition could introduce new layers of strategy for players as they navigate their escape routes.

4.5 Recommendations

Following the development of FIRE! 2.0, the first recommended course of action would be to conduct another round of playtesting. This is necessary to evaluate whether these changes have enhanced the overall game experience. Conducting playtests on the new version of the game can also help identify further areas for improvement.

Likewise, it is recommended to have a bigger sample size that also includes fire safety professionals. By having these experts play the game, their insights and feedback will provide professional validation and help determine the level of realism portrayed in the game more legitimately.

An additional stress-response measurement, aside from self-report, may also help evaluate the game's realism as it objectively gauges player responses while playing the game.

Moreover, the participants' knowledge should be evaluated after a significant time has passed after testing since knowledge retention is vital in solidifying the game's effectiveness.

For developers who want to build upon this game, they may add more level layouts into the game, such as subdivisions, city plazas, and high-rise commercial buildings among others. This may help players gain more experiences with different circumstances, helping them further enhance their decision-making skills in intense situations like fire incidents.

Furthermore, future iterations may incorporate more socially dynamic environments. This includes modelling the collective behavior, informal coordination and family-centered decision-making that are reflective of Filipino fire scenarios. These characteristics can be simulated through the newly added NPCs, enhancing the overall social realism and authenticity of the experience.

They may also add a Filipino language option since the game is currently only in English, but it is targeted towards Filipinos. On a related note, future developers can also make the game available on mobile devices. A mobile port would make it more accessible to Filipinos, in turn having a greater reach.

REFERENCES

[1] Chavez. https://mb.com.ph/2023/5/23/ph-fire-incidents-rises-by-8-3-in-first-4-months-of-2023-bfp

[2] Hsieh et al. https://www.sciencedirect.com/science/article/abs/pii/S0260691723002137

[3] International Association of Fire and Rescue Services. https://www.ctif.org/news/ctif-addressing-climate-crisis-call-action-fire-and-rescue-services-worldwide

[4] Lean et al. https://link.springer.com/chapter/10.1007/978-3-030-46951-1_9

[5] Oe et al. Non-Player Characters for Evacuation Training in Metaverse: Preliminary Experiment.

[6] Philippine Rubber Research Institute. https://prri.da.gov.ph/fire-prevention-month/

[7] Purnell. https://www.philstar.com/lifestyle/gadgets/2023/05/21/2267517/philippines-has-highest-average-screen-time-phones-study.

[8] Ribeiro et al. https://arxiv.org/ftp/arxiv/papers/1303/1303.3827.pdf

[9] Sarao. https://newsinfo.inquirer.net/1879861/bfp-on-recording-15679-fires-in-2023

[10] Science Education Resource Center, Carleton College. https://serc.carleton.edu/introgeo/games/whygames.html

[11] Vlachopoulos and Makri https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-017-0062-1

[12] Yang et al. https://www.mdpi.com/2076-3417/11/23/11284

Birdview: Human Crowd Density Visualization Tool from Video Feed to Floor Plan using Deep Learning and Homography

Shanmykel Ace Dela Cruz
Institute of Computer Science
University of the Philippines Los Baños
Los Baños, Laguna, Philippines
ssdelacruz1@up.edu.ph

Michael Anthony Jay Regis
Department of Computer Science and Technology
Visayas State University – Main Campus
Baybay, Leyte, Philippines
mbregis@vsu.edu.ph

ABSTRACT

This research presents the development and evaluation of a real-time crowd density heatmap visualizer system that integrates computer vision techniques with homography or projective transformation for facility management applications.

The system, named "Birdview," employs YOLOv11n-pose and other YOLO-based custom pose deep learning models for person detection using keypoint estimation, specifically targeting hip keypoints for near-accurate spatial positioning. Through homography, detected individuals from the camera feed are mapped onto the corresponding architectural floor plans, where Gaussian kernel-based density estimation generates intuitive heatmap visualizations of crowd distributions.

The research contributes a mathematical framework for evaluating the said crowd density visualizer system, introducing an annotation-based homography accuracy metric for the sole purpose of this system.

The system is able to visualize crowd other than humans depending on the loaded inference model. Although, the scope of this research is focused on detecting human crowd only.

The system is modularized enabling adaptation to various environments through calibration, making it viable for deployment in education institutions, commercial spaces, and public venues requiring intelligent space utilization monitoring and other applications.

KEYWORDS

Crowd Density Monitoring, Homography Transformation, YOLOv11n-pose, Real-time Person Detection, Spatial Mapping, Heatmap Visualization, Computer Vision

1 Introduction

Human crowd counting and density estimation provides valuable analytics for urban planning, safety management, event organization, retail analytics, and real estate valuation. Traditional manual counting methods were time-consuming, subjective, and inaccurate. Computer vision and deep learning have enabled automated crowd density estimation, providing more efficient, reliable, and quantifiable solutions for these domains.

Real-time human detection is one of the most fundamental tasks in computer vision and has become one of the most popular research topics in different fields over the last few years since it has numerous commercial applications (Paul et al., 2013). Detection of humans from surveillance cameras can be done by various techniques (Ansari and Singh, 2021) including motion-based detection and deep-learning-based detection.

Notably, deep learning algorithms have rapidly advanced in this domain, enhancing both speed and accuracy in human detection tasks. Deep-learning-based computer vision technology outperforms conventional image processing and recognition techniques in terms of detection speed, algorithm robustness, and feature extraction without manual design (Zhao et al., 2019).

A heatmap represents values through color, typically using red for higher concentrations (more congestion) and blue for lower concentrations (less congestion). Heatmaps effectively identify congestion areas in spaces like school buildings, highlighting frequently occupied locations. This enables analysis of congestion trends, though many current density visualization systems lack floor plan mapping for better understanding and data interpretation.

Integrating these detected persons into the subject location's floor plan is the core goal of this study. Floor plan mapping can

be done using homography or projective transformation. In projective geometry, a homography is an isomorphism of projective spaces, induced by an isomorphism of the vector spaces from which the projective spaces derive (Berger, 2009).

Projective transformations serve as the foundation for algorithms such as the Direct Linear Transform (DLT), which estimates camera parameters and maps 3D points to 2D images. This transformation is crucial for tasks that require aligning images from multiple sources or calibrating cameras in 3D space (Hartley and Zisserman, 2004).

Detected individuals visualized on floor plans can highlight congestion through heatmaps. This approach provides a holistic, architectural view of activity patterns that transcends traditional monitoring limitations. Applications include optimizing space usage, designing evacuation routes, managing events, and establishing data-driven rental rates based on foot traffic, ultimately improving situational awareness and response strategies.

This research serves as a foundation for transforming crowd density into usable data across applicable domains. The study focused on the Department of Computer Science and Technology (DCST) student lobby of Visayas State University (VSU) Main Campus, which experiences congestion due to limited space after the Human Resources Information System (HRIS) room installation and high traffic from students across various disciplines taking computer courses, making it a well-enough candidate for density heatmap visualization testing.

1.1 Problem Statement

Managing crowd density in shared spaces like schools, public buildings, and commercial establishments remains challenging. Without effective monitoring, density fluctuations during peak hours cause congestion, discomfort, and inefficient space use.

Traditional methods like manual counting are time-consuming, labor-intensive, and inaccurate, while lack of objective data limits informed decision-making for real-time crowd management.

In the context of crowd density estimation, heatmaps are utilized to represent the spatial distribution of individuals within a specific area. Parzych et al. (2013) explained how to create a density heatmap of people's movements from video footage analysis in a salesroom.

However, the detection method that they have used is based on people's movement activity which is limited to detection happening only when there are continuous movements in the footage.

The two following figures display a density heatmap where Figure 1 shows a frame in the crowd surveillance video dataset PETS (Ferryman and Ellis, 2010) and Figure 2 shows the corresponding crowd density heatmap calculated by Sindagi and Patel (Sindagi and Patel, 2017).

Figure 1: **Crowd from a surveillance video frame**

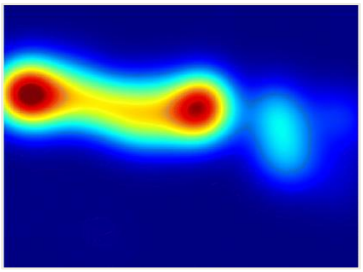


Figure 2: **Corresponding density heatmap of crowd frame**

This research, however, addressed these challenges using computer vision and heatmap analysis with floor plan visualization. The system transforms detection data into holistic floor plan visualizations that enhance operational awareness and provide quantifiable metrics for crowd control.

1.2 Research Objectives

This study implemented and evaluated a system that converts live surveillance video into intuitive heatmap visualizations on floor plan blueprints for crowd density analysis. The specific objectives were to:

1. train and fine-tune a deep learning model for person detection in the DCST student lobby
2. map detected person's coordinates from the video on to the floor plan through homography transformation
3. process the coordinate data to create a crowd density heatmap; and
4. overlay heatmaps on floor plans for spatial analysis

1.3 Research Scope and Limitations

The subject location of focus for this study was the Visayas State University's Department of Computer Science and Technology building lobby. The system was designed to provide visualization for a single-floor region only. The object-detection model was configured to detect humans exclusively, whether individuals or groups (crowds). The study employed a single camera setup throughout the implementation and testing of the system. The system was optimized for controlled indoor environments with a static camera angle and position.

The effectiveness of crowd density estimation and visualization was limited by the coverage and quality of the camera used. The system was tested under daylight settings only. However, even if the actual code of the system used a simple gamma correction and binary thresholding for when the environment gets darker, this was not actually tested due to limitations on operating hours on campus grounds.

2 Materials and Methods

2.1 Materials

The development of the system components was carried out using the appropriate tools and technologies, including but not limited to the necessary hardware, software, and methodologies needed for the realization of the system.

2.1.1 Software. The Python programming language was used as the primary and only programming language, from training and fine-tuning the model, visualization of the resulting floor plan, up to the evaluation of the system's performance. The Integrated Development Environment (IDE) of choice was the Visual Studio Code IDE. The Operating System (OS) and distribution used was the Linux OS Ubuntu Distribution.

2.1.2 Hardware. The system was implemented on a modern computer running a CORE I3-9th Generation processor, 4 processing cores with an NVIDIA RTX 3060 Graphics Processing Unit (GPU), and 16 gigabytes of random-access memory (RAM). The system utilized a V380 PRO CCTV Camera of 3 megapixels. Pre-recorded videos were also supported, used, and tested by the system where a POCO F4 Mobile Rear Camera of 64 megapixels was used.

2.2 Methods

The full depiction and steps in undertaking the system along with the application of the method in processing crowd density data and its visualization accuracy are presented. These methods are introduced as subsections in sequence on how the system was implemented.

2.2.1 Dataset. The person-detection model was trained using images from the DCST student lobby. We refer to these dataset as the DCST dataset. The DCST dataset was annotated with bounding-boxes, and keypoints for pose estimation. Example of these dataset images are shown in Figure 3. Permission and authorization were acquired for these images to be used in the data collection and paper presentation.

2.2.2 Object-Detection Model. This research utilized the YOLOv11n-pose model, the latest YOLO version available at the time of writing. The nano (n) size was selected for its efficiency, and the pose/keypoint detection mode was used for person detection, with the bounding-boxes as fallback.

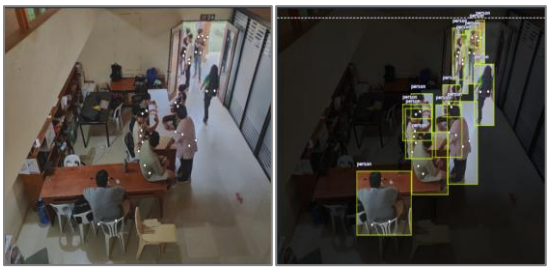


Figure 3: **Annotated images from the DCST dataset**

2.2.3 Model Fine-Tuning. The DCST dataset was the primary dataset for model fine-tuning. All image processing, annotation, augmentation, and dataset preparation were done using Roboflow's free service. Roboflow is a web service that empowers developers to build their own computer vision applications, no matter their skillset or experience. They streamline the process between labelling your data and training your model (Roboflow, 2020).

Export configurations included train/test split settings and image augmentation to create additional unbiased training samples. The dataset used a 70:20:10 train, validation, and test split, respectively. Pre-processing included auto-orientation, resizing to 640×640, and histogram equalization for contrast adjustment. Augmentation applied hue (±21°), saturation (±25%), brightness (±22%), exposure (±4%), and noise (0.14% pixels) variations all through Roboflow.

After dataset export, Ultralytics' website service, Ultralytics Hub, was used for model training where it provides an intuitive user-interface for setting your model training with auto-generated metrics and graphs. Roboflow handled the dataset preparation with YOLO-compatible file structuring, while Ultralytics provided efficient training of both pre-trained and non-pretrained models using their architectures.

The model was trained using the DCST dataset with the following parameters in training:

- 100 epochs
- 100 patience epochs
- 640 pixels image input size; and
- Hardware-accelerated training

2.2.4 Camera Placement Setup. The testing phase used two cameras: (1) a V380 PRO CCTV Camera for real-time performance evaluation, and (2) a POCO F4 Rear Camera for pre-recorded video data variation.

Figure 4 illustrates the official camera setup done in the DCST student lobby during the implementation. This setup, however, was for the POCO F4 Rear Camera only as CCTV access was prohibited on school spaces. The CCTV setup was done at the researcher's home.

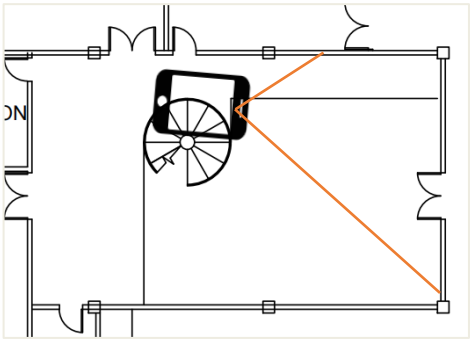


Figure 4: **Camera placement setup**

2.2.5 Person Detection using Hybrid Methods. This research used YOLOv11's pose estimation to detect individuals in video streams. Rather than merely using bounding-box centers, hip keypoints were also used for a more accurate person position acquisition.

The human pose provides more anatomically meaningful positioning rather than geometric bounding-boxes. A simple algorithm finds each person's reference point using detected hip keypoints, offering an alternative to the generic bounding-box approach.

For each detected person with bounding-box $b_i = (x_{min}, y_{min}, x_{max}, y_{max})$ containing the coordinates (x_{min}, y_{min}) for the top-left corner and(x_{max}, y_{max}) for the bottom-right corner of the bounding-box; and keypoints $K = \{k_0, k_1, \ldots, k_{16}\}$, where each keypoint $k_i = (x_i, y_i, c_i)$ contains the x- and y-coordinates, and the confidence score c, the reference center point P for each detected person is as follows:

$$P = \begin{cases} \frac{(k_0 + k_1)}{2} & c_0 > \tau \text{ and } c_1 > \tau \\ k_0 & c_0 > \tau \text{ and } c_1 \leq \tau \\ k_1 & c_0 \leq \tau \text{ and } c_1 > \tau \\ b_{center} & c_0 \leq \tau \text{ and } c_1 \leq \tau \end{cases} \quad (1)$$

This features different case for finding the reference center point of a detected person:

i. when both hip keypoints are detected, their coordinates are averaged to find the center
ii. when only one hip keypoint is detected, that keypoint is used; and
iii. when no hip keypoints are available due to possible occlusions, the bounding-box center-of-gravity is used as fallback.

In the equation, τ is the confidence score threshold, typically set to 0.5 based on the YOLO default values, and k_0 and k_1 represent the left and right hip keypoints, respectively. Indices 0 and 1 represent custom model keypoints, while Ultralytics' pretrained YOLO models use COCO format with hip keypoints at indices 11 and 12. The bounding box centroid b_{center} equates to $(\frac{x_{min}+x_{max}}{2}, \frac{y_{min}+y_{max}}{2})$.

2.2.6 Homography Transformation. Homography principles transform camera coordinates of detected persons into floor plan coordinates. This requires computing homography matrices H using carefully calibrated corresponding point pairs between camera view and floor plan.

A planar projective transformation is a linear transformation on homogeneous 3-vectors represented by a non-singular 3×3 matrix:

$$\begin{matrix} x' \\ y' \\ w' \end{matrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2)$$

or more briefly, $x' = Hx$. A projective transformation projects every figure into a projectively equivalent figure, leaving all its projective properties invariant. In the ray model of Figure 5, a projective transformation is simply a linear transformation of $\mathbb{R}^3$.

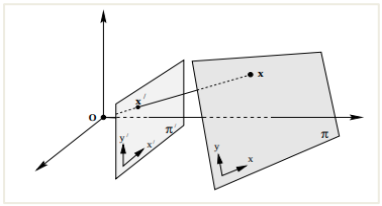


Figure 5: **Central projection map points on one plane to points on another plane**

The projection also maps lines as may be seen by considering a plane through the projection center which intersects with the planes π and π'. Since lines are mapped to lines, central projection is a projectivity and may be represented by a linear mapping of homogeneous coordinates $x' = Hx$ (Hartley and Zisserman, 2004).

2.2.7 Gaussian Kernel Density Visualization. This research uses overlapping Gaussian kernels to create smooth density visualizations. Each detected person contributes a two-dimensional Gaussian distribution rather than a discrete point, producing intuitive density maps through Kernel Density Estimation (KDE). The mathematical foundation for this visualization involves applying a two-dimensional Gaussian kernel for each person position (x_i, y_i):

$$G(x, y) = I_0 \cdot \exp\left(-\frac{(x - x_i)^2 + (y - y_i)^2}{2\sigma^2}\right) \quad (3)$$

Equation 3 is applied to each detection. The final density heatmap $D(x, y)$ is then a cumulative contribution of all individual kernels $G(x, y)$:

$$D(x, y) = \sum_{i=1}^{N} G(x, y) \quad (4)$$

The equation $D(x, y)$ defines density at pixel coordinate (x, y). Overlapping kernels from nearby individuals create higher intensity regions, representing crowding effects. Intensity values are assigned colors to create heatmaps where denser areas appear as "hotter" regions. The heatmap uses normalized Gaussian kernel intensities transformed to a [0, 255] range. Using the jet colormap, values progress from blue (0) to red (255).

3 Results and Discussion

A graphical user interface (GUI) named Birdview was implemented to manage system parameters. The GUI shows potential for production deployment with some adjustments and proper packaging. Figures 6 to 10 showcase Birdview and its main components or modules.

The monitoring module (Fig. 10) contains three sub-modules: camera points setup, world points setup, and monitoring module. This allows users to calibrate camera and floor plan coordinate sets for homography calculation. Figure 11 shows the camera points setup preview with dynamic transformation based on the camera points on the left preview, while Figure 12 shows the floor plan points setup where each point is numbered according to the precedence of top-left, top-right, bottom-right, bottom-left.

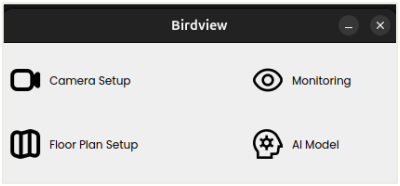


Figure 6: **Birdview: System Interface**

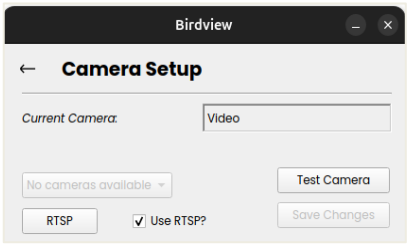


Figure 7: **Camera setup module**

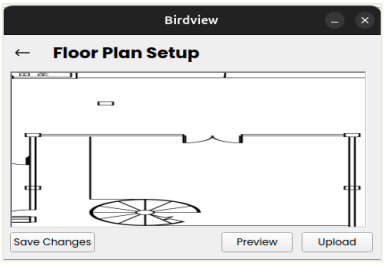


Figure 8: **Floor plan setup module**

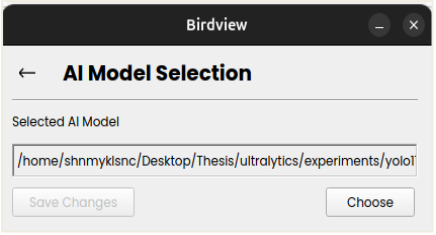


Figure 9: **Deep learning model setup module**

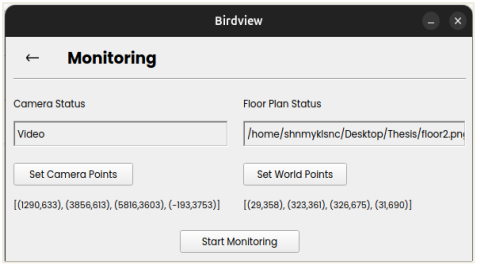


Figure 10: **Monitoring module**

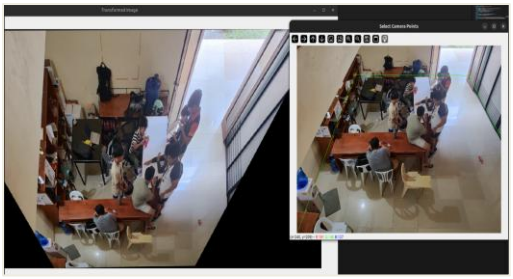


Figure 11: **Camera points setup module**

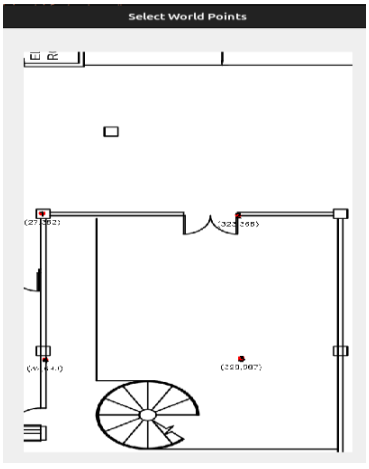


Figure 12: **Floor plan points setup module**

The *start monitoring* button initiates real-time crowd density monitoring by integrating all calibrated parameters. It combines coordinate sets, homography transformation, and the deep learning model to generate live heatmap visualization

on the DCST student lobby floor plan. Figure 13 shows a snapshot of the monitoring module.

Figure 13: **Human Crowd Density Heatmap snapshot**

In totality, the system "Birdview" is a modular system where users are free to change the deep learning model, the floor plan image, and video source, and also define their homography parameters using the floor plan points setup and camera points setup modules (Figures 11 and 12). The camera points setup module is an interactive module that warps the image according to the four points set.

3.1 Model Performance

The object detection model's result after training is shown in Figure 14 showing the mAP50 (mean average precision at 0.5 confidence threshold), mAP50-95, precision, and recall metrics over a hundred (100) epochs.

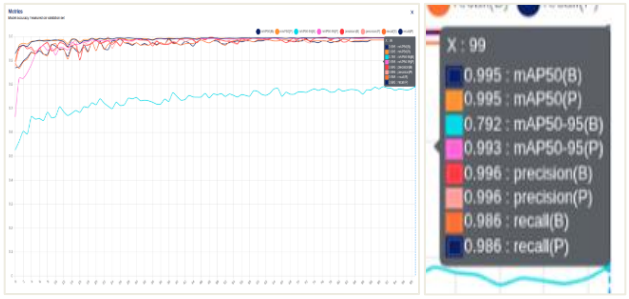


Figure 14: **Model training performance trends**

The pre-trained model leveraged transfer learning, starting with metrics around 0.8-0.85 and showing steady improvement. This demonstrated fine-tuning of existing knowledge rather than learning from scratch, with stabilization around epoch 20. Both mAP(B) and mAP(P) benefited significantly, with pose keypoint detection showing consistent performance.

The loss progression curves in Figure 15 show how the model learned detection and localization. It showcased transfer learning benefits with initial losses at 1.6-1.8 (box), 2.0 (class), and 1.4-1.6 (object) where existing object detection knowledge required onlyfine-tuning, with smooth convergence and steady decline in both training and validation losses.

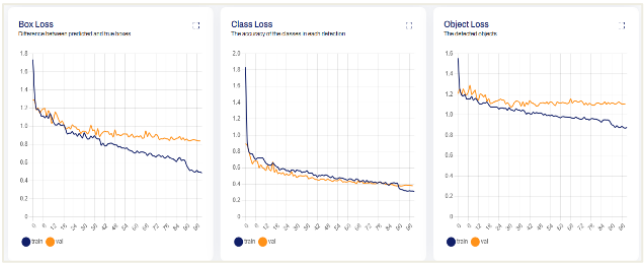


Figure 15: **Model box, class, and object loss trends**

3.2 Annotation-based Homography Accuracy

This metric uses annotation-based evaluation to quantify homography transformation accuracy, inspired by YOLO's intersection-over-union (IoU) testing approach. Like YOLO compares predicted and ground-truth bounding-boxes, this metric compares manually annotated ground-truth data with system-generated annotations. An interactive interface for generating ground-truth annotations was developed and shown in Figure 16.

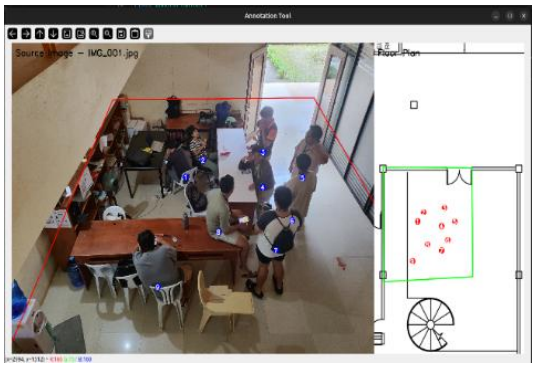


Figure 16: **Annotation tool showing the selected video frame and the floor plan side-by-side**

Manual ground-truth annotations were created using a custom annotation tool (Fig. 16) that allows clicking on person's hip locations. Each manual detection displays a blue circular indicator on the camera image's left panel at the clicked location. The right panel shows the floor plan preview with homography-transformed detections from the left panel, ensuring coordinate sets S and D match properly where the annotations are saved in XML format shown in Listing 1:

Listing 1: **Manual annotation XML file format structure**

```
<?xml version="1.0" encoding="UTF-8"?>
<annotations>
 <metadata>
  <camera points="x1,y1;x2,y2;x3,y3;x4,y4" />
  <floorplan points="x1,y1;x2,y2;x3,y3;x4,y4"
image="IMG.jpg" />
  <homography values="h11;h12;h13...;h33" />
 </metadata>
 <image name="sample.jpg">
```

```
  <point_pair>
   <source x="x1" y="y1" />
   <destination x="x2" y="y2" />
  </point_pair>
 </image>
 <image name=...>
...
</annotations>
```

To evaluate transformation accuracy, the Hungarian algorithm was used to solve the correspondence problem between expected (ground truth x_i^e, y_i^e) and actual (system x_i^a, y_i^a) detection points. Given two sets of points, expected annotations and actual detections, the algorithm first constructs a distance matrix D:

$$D_{ij} = \sqrt{\left(x_i^e - x_j^a\right)^2 + \left(y_i^e - y_j^a\right)^2} \quad (5)$$

where D_{ij} is the Euclidean distance between expected point p_i^a and actual point p_j^e. The Hungarian algorithm then finds the optimal assignment matrix that minimizes total distance subject to one-to-one matching constraints. After obtaining the optimal assignment, a distance threshold τ was applied to classify matches:

$$M = \{(i,j) \mid X_{ij} = 1 \text{ and } D_{ij} \leq \tau\} \quad (6)$$

where M contains valid matches, X_{ij} indicates algorithm assignment, and $\tau = 50$ pixels in this study. This enables calculation of true positives (valid matches), false positives (unmatched actual detections), and false negatives (unmatched expected detections), providing quantitative accuracy metrics with $O(n^3)$ time complexity.

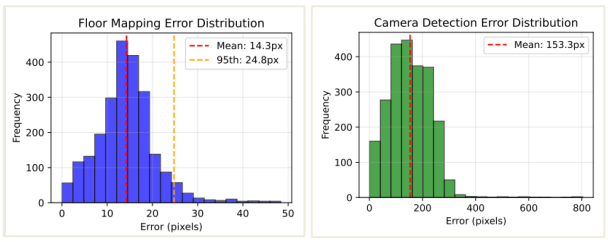


Figure 17: **Floor plan mapping and camera detection error distribution from the annotation-based accuracy**

The analyses in Figure 17 reveal two distinct aspects of system accuracy through error distribution patterns.

The floor mapping error distribution (left) demonstrates strong homography transformation performance with a mean error of 14.6 pixels and 95th percentile of 34.8 pixels, where over 80% of coordinate transformations achieve precision within 25 pixels. Complementing this, the camera detection error distribution (right) shows effective person localization with a mean error of 115.9 pixels, indicating reliable keypoint detection accuracy.

Both distributions exhibit tight, well-controlled patterns that validate the system's dual-stage accuracy: precise homographic mapping and consistent person detection, together supporting reliable crowd density visualization on floor plans.

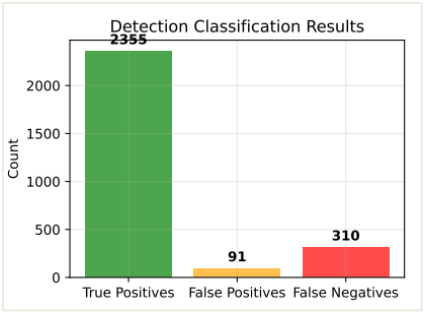


Figure 18: **Detection classification results from the annotation-based accuracy**

The detection classification results shown in Figure 18 strongly validate the system's overall performance, showing 2142 true positives against only 9 false positives and 310 false negatives. This demonstrates the reliability established in the error distribution analyses, where the low false positive count indicates that the homography transformation accuracy (mean 14.6 pixels) effectively prevents spurious detections on the floor plan.

The moderate false negative count suggests that most missed detections stem from challenging pose estimation scenarios rather than transformation failures. The high true positive count demonstrates that when persons are detected in camera view, the system consistently and accurately maps them to the correct floor plan locations, validating the integrated approach of precise homographic mapping and reliable person detection for crowd density visualization.

4 Conclusions

This research encompasses the creation of a modular system for visualizing crowd density data into a meaningful heatmap representation. The study explored video processing, object-detection using deep learning, homography principles, and optimization techniques to achieve the goals of this research.

This research also introduced a way to approximate homography accuracy inspired by YOLO metrics. The annotation-based accuracy metric can be applicable not just to homography transformation systems relying on YOLO for object detection, but also to other systems that use different means of classifying and detecting objects since the ground-

truth data is done manually. Pure image-processing-based annotations can benefit from this approach as well.

For future works, the goals for this research extends to including a future option to increase floor points and camera points from four to five or more. Other future works also include the integration of multiple cameras using the deeper theories in Multiple View Geometry by Hartley and Zisserman, as well as constructing a better numerical representation of crowd data for regression analysis, AI predictions, and graphical trends and presentations.

ACKNOWLEDGMENTS

The authors express their profound gratitude to the Department of Computer Science and Technology at Visayas State University for their invaluable and extended support for this research.

Special acknowledgment is extended to the Dean of the Faculty of Computing for granting permission to collect and utilize dataset images within the DCST student lobby. This study would not have been possible without access to the department's high-performance computing resources, particularly the GPU-accelerated workstation that facilitated the intensive deep learning model training and real-time processing requirements of this research.

Additional gratitude is given to the University of the Philippines Los Baños, ICS Faculty and Staff, and friends for the support and additional inputs for the research.

REFERENCES

Ansari, M. A., & Singh, D. K. (2021, March). Human detection techniques for real time surveillance: a comprehensive survey. *Multimedia Tools and Applications, 80*(6), 8759-8808. doi:10.1007/s11042-020-10103-4

Berger, M. (2009). *Geometry I* (Corrected 4th printing ed.). (M. Cole, & S. Levy, Trans.) Berlin: Springer-Verlag

Ferryman, J. M., & Ellis, A.-L. (2010). PETS2010: Dataset and Challenge. *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS 2010)* (pp. 143-150). Boston: IEEE. doi:10.1109/AVSS.2010.90

Hartley, R., & Zisserman, A. (2004). *Multiple View Geometry in Computer Vision* (2nd ed.). Cambridge: Cambridge University Press

Parzych, M., Chmielewska, A., Marciniak, T., Dąbrowski, A., Chrostowska, A., & Klincewicz, M. (2013). Automatic People Density Maps Generation with Use of Movement Detection Analysis. *6th International Conference on Human System Interaction (HSI 2013)* (pp. 26–31). Gdańsk-Sopot, Poland: IEEE. doi:10.1109/HSI.2013.6577798

Paul, M., Haque, S. M., & Chakraborty, S. (2013). Human Detection in Surveillance Videos and Its Applications: A Review. *EURASIP Journal on Advances in Signal Processing, 2013*(176), 1-16. doi:10.1186/1687-6180-2013-176

Roboflow. (2020, January 1). *About*. Retrieved March 22, 2025, from Roboflow: https://roboflow.com/about

Sindagi, V. A., & Patel, V. M. (2017). CNN-Based Cascaded Multi-task Learning of High-Level Prior and Density Estimation for Crowd Counting. *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 1-6). Lecce, Italy: IEEE. doi:10.1109/AVSS.2017.8078491

Zhao, Z.-Q., Zheng, P., Xu, S.-T., & Wu, X. (2019, January 30). Object Detection with Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems, 30*(11), 3212-3232. doi:10.1109/TNNLS.2018.2876865

Bridging the Gap Between Synthetic and Real Handwriting: Using Visual AI to Accurately Parse Structured Logic Proofs

Structure-Aware Parsing of Handwritten Boolean Proofs using the Nougat Architecture and Elastic Augmentations

Felisa Melanie Fay G. Bascug
Computer Science
Cebu Institute of Technology–University
Cebu City, Philippines
felisabascug@gmail.com

Roddneil B. Gemina
Computer Science
Cebu Institute of Technology–University
Cebu City, Philippines
roddneilgemina@gmail.com

Basil Xavier B. Mendoza
Computer Science
Cebu Institute of Technology–University
Cebu City, Philippines
x-mendoza@live.com

Catherine N. Arellano
Computer Science
Cebu Institute of Technology–University
Cebu City, Philippines
catherine.arellano@cit.edu

ABSTRACT

Digitizing handwritten formal logic remains a significant challenge in Document Analysis and Recognition (DAR). This is because models such as UniMERNet struggle in upholding spatial hierarchy, such as interpreting whitespace and vertical orientation of statements when processing tabular Hilbert-style proofs which can be referred to as Structural Collapse. To approach this important problem regarding how to preserve the vertical alignment of logical arguments we propose "Nougat-Logic", a Visual Document Understanding (VDU) paradigm that utilizes the Nougat Architecture in order to create a VDU Paradigm for Handwritten Domains. We utilize the Elastic-Hybrid Data Engine to generate synthetic proofs that include aggressive elastic distortions to bridge the Sim-to-Real Gap. With the 250 M parameter Nougat Model we achieve structural awareness on consumer hardware through fine tuning by utilizing Quantized Low-Rank Adaptation (QLoRA). Evaluations using the Tree Edit Distance Similarity (TEDS) metric demonstrate that Nougat-Logic achieves a TEDS similarity of 95.0% and is therefore substantially superior to the sequence based baseline of 72.9%. These evaluations support the use of VDU paradigms to preserve the topological truth of handwritten formal methods over serializations.

KEYWORDS

Optical character recognition, computer vision, handwritten digitization, artificial intelligence, boolean algebra.

1. Introduction

1.1 Background: Structural Properties of Logic Proofs

1.	P→Q	Premise
2.	P	Premise
3.	Q	M.P. 1,2

Figure 1: Modus Ponens Sample in Hilbert-style format

The validity of a formal proof relies fundamentally on its topological structure, not just its symbolic representations. For example, a Hilbert-style proof is represented as a structured table with three distinct columns: the first representing the line number, the second representing the logical statement, and the third representing the justification (i.e., an axiom or one of the rules of inference). Unlike ordinary writing or sequential mathematical equations, the meaning of a proof is determined topologically; a failure in the structure of the proof (e.g., incorrectly merging columns) will make the proof both computationally and logically invalid. As a result, handwritten logic proofs are at the crossroads of recognizing expressions in mathematics and recognizing structures in tables.

Unlike ordinary written language, logic proofs depend on rigid tabular alignments. Kasem et al. [12] have identified Table Structure Recognition (TSR), i.e., distinguishing between rows .and columns within a table, as being distinctly separate from text recognition. In particular, Kasem et al. noted that, without explicitly drawing grid lines, standard Optical Character Recognition (OCR) systems frequently fail to recognize the structural boundaries between rows and columns. Historically, this subdomain has lacked support from general purpose OCR systems.

Digitizing these documents is important for the modern education environment, since many automated grading systems and theorem provers need machine readable input. However, the variability in student handwriting, combined with the strict spatial requirements of logic tables creates a unique problem space. Traditional OCR processing typically processes text line-by-line and does not take into account the vertical relationships that determine the existence of a column. On the other hand, Handwritten Mathematical Expression Recognition (HMER) systems are designed to recognize complex inter-relationships between symbols but typically assume that each expression is contained within a contiguous area of the document and does not take into account the possibility of a tabular structure [2].

1.2 The Gap: Sequence vs. Structure

The current state of the art for HMER is defined through Sequence-to-Sequence (Seq2Seq) architectures. For example, UniMERNet [2] and CoMER [7] treat an input document image as a source to encode and an output LaTeX string as a target to decode. As such these models have been able to achieve significant results in recognizing complex formulas from input images by reducing 2D image features to 1D sequences.

However, these sequence based methods also lead to what we call Structural Collapse, as described by Seitz et al. [9]. Specifically, since sequence based methods discard the explicit mapping of symbols to traces, they fail to capture the complex spatial hierarchies that are present in Hilbert proofs. In particular, the models fail to preserve the 2D topological delimiters (whitespace) of Hilbert proofs which represent the spatial relationships between columnar content and instead incorrectly interpret them as 1D sequence delimiters (spaces). This causes a loss of semantic information.

When a Seq2Seq model processes a Hilbert proof, it commonly fails to correctly map the whitespace separating columns in a Hilbert proof as a space character and thus collapses the 3 column structure of a Hilbert proof into a single, nonsensical line of text.

VDU presents an interesting alternative approach to processing mathematical documents. While models such as Donut [3] and its educational specific version, Nougat [1], provide a Pixels-to-Sequence architecture that preserves the structural properties of the document, Nougat has demonstrated the ability to transform PDF images of scientific papers into Markdown with preservation of the structural elements (i.e., headers, tables, and multiple column layouts). However, Nougat was trained primarily using the arXiv born digital collection of papers and does not possess sufficient domain adaptation to properly understand handwritten material that may include irregular and noisy stroke patterns.

Beyond Hilbert proofs, structure-aware parsing is critical for any domain reliant on 2D layouts, such as matrices, commutative diagrams, or complex scientific tables. VDU overcomes the semantic information loss inherent in 1D serialization, advancing the broader field of STEM digitization from basic text recognition to true topological comprehension.

1.3 Contributions

This paper introduces **Nougat Logic**, an adapted version of Nougat Architecture specifically designed to recognize handwritten formal proofs. The major contributions of this paper include:

- **Application of VDU to HMER:** This paper is the first to apply the Nougat VDU architecture to the domain of handwritten logic. While we adapt the foundational Swin-Transformer and mBART backbone from the base Nougat model, we introduce novel architectural modifications, specifically specialized logic-token expansions and domain-specific prompt-conditioning. We demonstrate that these adapted VDU models surpass traditional HMER sequence models at preserving topological structure in handwritten proofs.
- **Elastic Hybrid Data Engine**: As there is no large dataset available with annotations of handwritten formal proofs, our research team has developed a synthetic data generation pipeline that uses Ghost Lines (table borderless) and extreme elastic distortion [8] to create variability in the appearance of student handwriting, thus creating a Sim-to-Real Gap (the drop off in accuracy when moving from synthetic training data to actual testing data).
- **Structural Metric**: In contrast to ExpRate (Expression Rate), which is typically used in HMER, we utilize the TEDS [4] from the Table Recognition domain to provide a rigorous evaluation of the topological correctness of the digitized proofs.
- **Efficient Fine Tuning**: We demonstrated that it is possible to train large VDU models on a consumer machine using QLoRA[6], where we froze the base model and only trained a small portion of the parameters.

The rest of the paper is structured as follows: Section 2 reviews the relevant literature in HMER and VDU; Section 3 describes the methodology (data generation engine and model); Section 4 provides the experimental results and discussion; and Section 5 draws conclusions.

2. Review of Related Literature

2.1 Handwritten Mathematical Expression Recognition (HMER)

The HMER field has undergone significant development over time, from the initial grammar-based methods to current deep learning systems. One major difficulty in HMER is that of handwritten mathematical writing being a two-dimensional representation of mathematics, where spatial relationships (such as superscripts/subscripts/fractions) provide crucial semantic information about the expression itself.

The Sequence-Based Paradigm: Over the last decade, the main methodology has been the use of the encoder-decoder architecture with the addition of attention mechanisms. These models treat the recognition problem as a translation problem, translating a given input image into a latex sequence. Recently, Wang et al. [2] presented UniMERNet, which serves as a robust baseline for this class of methodologies. UniMERNet uses a large-scale data set (UniMER-1M), and a length-aware module to process long, complicated formulas. Although UniMERNet achieves state-of-the-art performance in recognized benchmarks such as CROHME, it can only be used to recognize individual formulas, making it brittle for document level representations of structured content such as tables. Similarly, Zhao and Gao [7] developed CoMER, which incorporates a coverage-aware attention mechanism into the Transformer architecture. CoMER addresses the problems associated with “under-parsing” or “over-parsing” symbols, by incorporating an attention coverage tracking system for the input image.

In contrast to the benefits of these methodologies, recent studies have pointed out several inherent limitations. For example, Guan et al. [13] demonstrated that traditional attention mechanisms struggle with maintaining spatial accuracy and are subject to "attention drift" when processing densely written mathematical texts. In order to mitigate these limitations, they propose using a Position Forest as a model for the input image in order to capture relative positions. This supports their hypothesis that topological knowledge is better suited for complex layouts than is pure sequence modeling. Additionally, because many proofs of Hilbert's theorem consist of tables of expressions, sequence models typically do not include an explicit structural prior. Therefore, when attempting to parse a table of expressions, the sequence model will tend to merge adjacent columns, resulting in Structural Collapse.

Structure-Aware Models: Although there is evidence that the research community is beginning to realize the limitations of purely sequential models, and is shifting focus towards structure-aware models, we believe that this represents a significant shift in how researchers view handwritten mathematical documents. Zhu et al. [5] support the argument that handwritten mathematical expressions are inherently tree-structured, and therefore serializing them into one dimensional sequences may introduce ambiguity. Furthermore, Zhu et al. [5] argue that this type of ambiguity is magnified when processing the multiple column structure of formal logic tables. This is further reinforced by Seitz et al. [9], who describe "The Return of Structural Handwritten Mathematical Expression Recognition," and argue that models must incorporate the graph structure of strokes and symbols. Our work builds upon the shift towards structure-aware models initiated by Zhu et al., and expands the scope of the models beyond the syntax tree of a single formula to the tabular topology of an entire proof.

2.2 Visual Document Understanding (VDU)

Alike with HMER, the field of general document understanding has seen new advancements with the introduction of transformer-based VDU models. Instead of using the typical pipelined model of text detection, followed by OCR, followed by layout analysis, end-to-end processing of VDU models have become commonplace.

Transformer Advancements: The fundamental contribution in this space is the donut model by Kim et al. [3], which was the first model to suggest an OCR-free approach and map raw document images to a structured Javascript Object Notation (JSON) output without requiring explicit bounding box detection. The "pixels-to-structure" paradigm proposed by donut has been successful for forms and receipts but has limited applicability. However, building on the success of donut, Blecher et al. [1] presented nougat, which is specifically designed for scientific literature and converts PDF pages to rigorously formatted markdown, including the accurate parsing of mathematical formulas and tables.

Although nougat was trained primarily on the arXiv dataset, the arXiv dataset contains mostly clean, typeset documents and does not contain much variability. Therefore, our study examines the significant difference between the clean, printed text of academic documents and the noisy, irregular hand writing of other documents. We are investigating if the architectural priors of nougat can be adapted to handwritten documents through efficient fine tuning.

2.3 Tabular Data Extraction and Evaluation Metrics

Evaluation of tabular data is not possible with simple metrics such as character error rate. Zhong et al. [4] presented the TEDS metric in their study on PubTabNet, a large scale dataset for extracting tables from images. TEDS represents the table as a tree structure of HTML tags (<cell>,<row>) and calculates the similarity of the table as follows:

$$TEDS(T_a, T_b) = 1 - \frac{EditDist(T_a, T_b)}{max(|T_a|,|T_b|)}$$

Where T_a and T_b are the tree structures of the prediction and ground truth, and $|T|$ represents the number of nodes in the tree. In simpler terms, TEDS compares the underlying structure in the form of HTML or Markdown grid for the predicted table against the actual table, deducting points for every structural failure such as missing column or incorrectly merged rows. This metric is crucial for our study because it penalizes structural errors (e.g., missing a column split) much more heavily than minor text recognition errors.

3. Methodology

3.1 The "Elastic-Hybrid" Data Engine

The major constraint on the application of deep learning to handwritten logic proofs is the scarcity of labeled data available. The lack of annotated handwritten data sets is a major constraint to HMER research (Li et al., 2018) [11] . Manual labeling of logical LaTeX is prohibitively expensive, and therefore, requires making large scale synthetic data engines capable of modeling the complex syntax found in real world logic. Because there does not exist an open source dataset containing a large number of handwritten Hilbert-style proof examples, we decided to create our own. Taking inspiration from the work of Alonso et al. [10] regarding synthetic text generation and Buslav et al. [8] regarding image augmentations, we created the Elastic-Hybrid Data Engine; a python-based synthetic generation engine for creating training samples that are visually distinguishable yet topologically equivalent to the actual work of students.

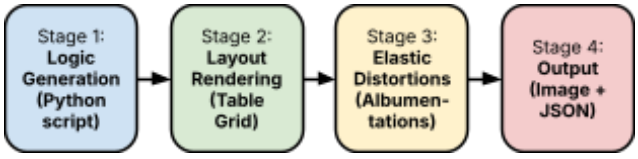


Figure 2: The Elastic-Hybrid Data Generation Pipeline. The system systematically constructs the logical skeleton, renders the layout, and applies aggressive elastic distortions to generate paired image-JSON training samples.

3.1.1 Structural Synthesis: Before generating a visual representation of a proof, the engine generates an abstract representation of how that proof would be structured as a series of valid propositions. The engine uses generative grammar to create a 3 column structure (Line Number (n), Statement (s), and Justification (j)) for these propositions. It's important that the engine can vary the structure of the table for the proof. A traditional proof shows all the vertical lines; however, students will sometimes omit some of them, and/or write on their paper using grid paper that has only horizontal grid lines. Therefore, we call the engine's rendering of "ghost lines", when it renders the proof without visible table borders during the image generation process, but still retains the structural tags (<cell>,<row>) in the ground truth Markdown. This means that the attention mechanism has to determine the implicit alignment, between the embeddings of tokens along the Y-axis, and not just rely on the difference in intensity of vertical line pixel values.

3.1.2 Typographic Variability: In order to capture the variability in human handwriting, we created a collection of 50 handwritten fonts, available under the Creative Commons license (e.g. Google Fonts' Handwriting corpus). For each generated proof, the engine selects one primary font, and introduces "glyph noise" into the engine by replacing glyphs with glyphs from a secondary font that is very similar to the primary font. This allows us to prevent the model from overfitting to the shape of the glyphs of a particular font family.

3.1.3 Elastic Distortion and Augmentation: The core innovation of the engine is the application of aggressive "Sim-to-Real" augmentations powered by the Albumentations library [8]. We observe that human handwriting is rarely perfectly rectilinear; lines of text often drift, and individual characters vary in size and slant. We model this using Elastic Transformations where a displacement field is generated from a Gaussian distribution. Specifically, we apply:

- **Grid Distortion:** Warping the underlying coordinate mesh to simulate paper curvature or camera lens distortion. The image is split into a 5x5 grid where the edges of the grid are deformed.
- **Elastic Transform:** Applying local affine transformations to simulate the erratic muscle movements of handwriting. The alpha value or strength of the transforms are 120, which has more pronounced transforms.
- **Erosion/Dilation:** Randomly thinning or thickening strokes to mimic different pen pressures and ink bleed.

3.2 Model Architecture: Nougat-Logic

Our model, Nougat-Logic, is built upon the architecture of Nougat (Neural Optical Understanding for Academic Documents) [1], which itself is a specialized instance of the Donut (Document Understanding Transformer) architecture. It follows an Encoder-Decoder structure that processes the input image $X \in \mathbb{R}^{H \times W \times C}$ and auto-regressively generates a sequence of tokens:

$$Y = (y_1, y_2, \dots, y_m)$$

To explicitly define our architectural scope, the visual encoder and text decoder rely on the established Swin-Transformer and mBART architectures of the base Nougat model. Our novel architectural contributions to this pipeline bypass a complete redesign; instead, we introduce specialized tokenizer modifications (detailed in Section 3.2.2) and task-specific prompt engineering (detailed in Section 3.2.3) to specifically handle the spatial/syntactic challenges of handwritten formal logic.

3.2.1 Visual Encoder: The encoder is a Swin-Transformer, a hierarchical vision transformer that computes representations at shifted windows. This architecture is particularly well-suited for document images because its hierarchical nature captures both local features (individual strokes/symbols) and global context (page layout and column structure). The input image is resized to 896x672 pixels. The Swin-Transformer outputs a sequence of embeddings Z, which effectively compresses the visual information into a latent representation.

3.2.2 Text Decoder: The decoder is an mBART-based transformer. As one of our primary architectural modifications, we modify the standard Nougat tokenizer by adding specialized tokens for logic proofs, specifically ensuring that the model can predict structural tags like \begin{tabular}, \hline, and & (column separator) with high probability.

3.2.3 Prompt Engineering: Unlike standard HMER models, Nougat allows for prompt-conditioning. We prepend a task-specific prompt token <logic_proof> to the decoder input, signaling the model to apply learned parameters for tabular logic parsing.

3.3 Efficient Training via Quantized Low-Rank Adaptation (QLoRA)

Fine-tuning the full 250 million parameters of the Nougat model is computationally prohibitive for typical academic labs. To address this, we employ QLoRA [6].

3.3.1 Low-Rank Decomposition:

LoRA hypothesizes that the change in weights **ΔW** during domain adaptation has a low intrinsic rank. Instead of updating the full weight matrix $W \in \mathbb{R}^{d \times k}$, we freeze W and train two smaller matrices A and B such that:

$$W' = W + \Delta W = W + BA$$

Where $B \in \mathbb{R}^{d \times r}$ is initialized to zero and $A \in \mathbb{R}^{r \times k}$ is initialized with a random Gaussian distribution, with rank $r \ll min(d, k)$. For our experiments, we set r=16.

3.3.2 4-bit Quantization: We further reduce memory consumption by loading the frozen base model in 4-bit precision (NF4 data type). This is so that the model would require less memory in order to be trained.

3.3.3 Hyperparameters: The model was trained for 5 epochs on a synthetic dataset of 10,000 generated proofs. We used the AdamW optimizer with a learning rate of $2*10^{-4}$ and a cosine learning rate schedule. The batch size was set to 1 with gradient accumulation steps of 16 to simulate a larger effective batch size.

3.4 Neuro-Symbolic Validation

We implemented a deterministic, logical consistency check as a final evaluation step for the output. This check parses the output JSON tree and extracts the line references from the "justification" column. The check verifies each referenced index ***i*** is less than or equal to the current number of rows in the table (i.e., $i \leq n_{\text{current}}$). The predictions with the invalid references (hallucinations) are identified, and therefore eliminated from the logically possible derivations.

3.5 Evaluation Hypotheses

The research approach utilizes the following statistical hypotheses:

- **Null Hypothesis (H_0):** There is no significant difference in the TEDS scores between the UniMERNet (the sequence-based baseline) and Nougat-Logic (the structure-aware model), for evaluating handwritten Hilbert proof sequences.
- **Alternative Hypothesis (H_1)**: Nougat-Logic (structure-aware model) has a statistically significant higher TEDS score than UniMERNet (sequence-based baseline).

3.6 Ablation Study Design

To evaluate our Elastic-Hybrid Data Engine, we designed an ablation study isolating its two core augmentations: Elastic Distortion and Ghost Lines (implicit borders). This allows us to quantify their individual contributions toward bridging the Sim-to-Real gap for erratic handwriting.

4. Results and Discussion

4.1 Experimental Setup

Our Experiments were run using a virtual python 3 workstation via google colab using its T4 GPU with 16GB VRAM. We ran each model on a Test Set that we had reserved for this purpose - 2000 Samples total; 1000 clean synthetical Samples (seen Fonts, No Distortion); 1000 hard samples (unseen fonts, Elastic deformation, Ghostlines), thus testing each Model's capability to learn the task as well as to extend to difficult, realistic conditions. Seen vs unseen fonts were selected at random with the distribution of 80:20. The parameters for the distortions and deformations were also slightly increased for this set.

We compare our **Nougat-Logic** model against two baselines:

- **UniMERNet [2]:** The current state-of-the-art sequence-to-sequence model for mathematical expression recognition.
- **Nougat-Base [1]:** The original pre-trained Nougat model without fine-tuning, to establish the necessity of domain adaptation

Alongside the fully augmented model, we trained two ablated variants: one without elastic distortion and one without Ghost Lines (using explicit grid lines instead). To ensure fair comparison, both variants utilized identical training hyperparameters and were evaluated on the same 2,000-sample test set.

4.2 Quantitative Analysis

Table 1 presents the primary performance metrics. We report the Exact Match Rate (ExpRate), which measures the strict character-for-character accuracy of the entire output sequence; Normalized Edit Distance (NED), which measures the overall character-level alignment and transcription accuracy; and TEDS (Tree Edit Distance Similarity), which measures the structural accuracy of the tabular layout.

Table 1: Comparative Performance on Test Set (Average of Clean and Hard splits)

Model	Parameters	Inference Time (ms)	Exp Rate (%)	TEDS (%)	NED (%)
Nougat-Base	250 M	450	0.0	0.0	11.52
UniMERNet	**180 M**	**120**	0.0	72.92	38.79
Nougat-Logic	250 M	450	**14.2**	**95.01**	**88.1**

Analysis of Results:

The results strongly support our Alternative Hypothesis (H1).

- **Nougat-Base Failure:** Zero-shot performance on Nougat-Base completely failed to parse the topological structure (0.0% TEDS) or exactly match any sequences (0.0% ExpRate). This confirms a massive domain gap between printed academic papers and handwritten logic, proving that the pre-trained VDU priors are insufficient without our targeted fine-tuning and specialized logic tokens.
- **The "Structural Collapse" of UniMERNet:** The sequence-based baseline completely fails to Exact Match (0.0% ExpRate) any of the structured proofs. Because standard HMER models serialize mathematical expressions without an explicit structural prior, UniMERNet drops the

necessary Markdown table syntax, resulting in a low Normalized Edit Distance (38.8%). While its TEDS score of 72.9% indicates it captures some relative spatial positioning, it ultimately merges columns and fails to preserve the rigid topology required for valid Hilbert proofs.

- **Nougat-Logic Performance:** our fine-tuned model achieved the highest scores in both metrics, achieving a 95.0% TEDS score. This result demonstrates consistent preservation of table topology and column boundaries across the complex distortions present in the test set.

- **Character vs. Sequence Accuracy**: While Nougat-Logic achieved an Exact Match rate of 14.2%, its Normalized Edit Distance remained exceptionally high at 88.1% (a massive improvement over the baseline's 38.8%). This highlights the extreme brittleness of strict 1D sequence metrics for this task. Nougat-Logic accurately transcribes the vast majority of the document's characters and structure, but a single hallucinated whitespace or minor LaTeX variation (such as an omitted double negation) causes a binary Exact Match failure.

Furthermore, separating the test set highlights the model's resilience. On the 1000 clean synthetic samples, Nougat-Logic achieved a TEDS score of 98.5% and an NED of 94.0%. On the 1000 'hard' samples containing extreme elastic distortions, unseen fonts, and implicit ghost lines, the model maintained a robust TEDS score of 91.5% and an NED of 82.2%, demonstrating that the Elastic-Hybrid Data Engine successfully prepared the model for severe typographical irregularities.

Table 2: Ablation Study - Impact of Data Engine Components

Configuration	TEDS (%)	NED (%)	Observation
Full Elastic-Hybrid	95.0	88.1	Robust to all distortions.
w/o Elastic Distortion	82.0	78.5	Fails on curvy/slanted text.
w/o Ghost Lines	76.5	72.0	Fails when borders are missing.

Our data engine plays a major role in the model's success, as demonstrated in Table 2. The removal of elastic distortion from the 10,000-image training pipeline resulted in a 13% drop in the TEDS score, proving that "Sim-to-Real" spatial augmentations are vital for enabling the model to handle erratic, curving human handwriting. Furthermore, eliminating "Ghost Lines" (training exclusively on tables with visible black borders) drastically decreased performance during test time on non-bordered examples, dropping the TEDS score to 76.5% and the NED to 72.0%. This illustrates that the model must be forced to learn implicit structural alignment based on spatial token embeddings, rather than simply relying on explicitly drawn grid lines

4.3 Qualitative Discussion

To move beyond a purely quantitative evaluation, we conducted a systematic qualitative error analysis. Rather than random sampling, we stratified our inspection by isolating the bottom 10th percentile of TEDS scores for both UniMERNet and Nougat-Logic. This targeted sampling allowed us to categorize the most frequent failure modes across both the sequence-based baseline and our proposed VDU architecture.

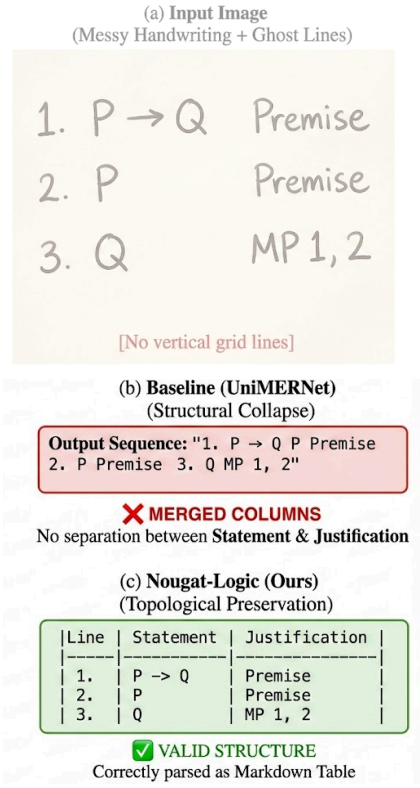


Figure 3: Visualizing Structural Collapse. (a) A synthetic input proof with “ghost lines” (missing vertical borders) and elastic distortion. (b) The baseline UniMERNet model fails to detect implicit column boundaries, merging the statement and justification into a single linear string (Structural Collapse). (c) The proposed Nougat-Logic model correctly identifies the tabular topology, outputting a valid Markdown structure.

Our systematic review identified three primary qualitative categories of structural parsing: **Implicit Boundary Detection (The Baseline Failure):** As illustrated in Figure 3, when vertical grid lines are omitted (Ghost Lines), UniMERNet consistently fails to detect the implicit spatial boundary, resulting in complete structural collapse.

- **Complex Hierarchies:** In samples containing heavily nested logic or sub-proofs, sequence models lose the hierarchical context, whereas Nougat-Logic maintains the localized bounding boxes of the sub-structures.

- **Extreme Elasticity:** Under maximum elastic distortion, simulating erratic handwriting, the spatial relationships between symbols degrade significantly, challenging even structure-aware models.

This qualitative difference illustrates why TEDS is the superior metric for this task. A standard string-distance metric might not penalize the UniMERNet output heavily because all the characters are present. However, logically, the UniMERNet output is not computationally parsable with its loss of structural metadata, whereas the Nougat-Logic output is valid and machine-readable.

While Nougat-Logic mitigates the structural collapse seen in the baseline, our error analysis revealed two distinct failure modes specific to our architecture:

1. Row Over-Segmentation (Hallucinated Line Breaks): In instances of extreme elastic distortion where a student's handwriting features a severe downward slant, Nougat-Logic occasionally misinterprets a single, drastically sloped logical statement as spanning two separate rows. This results in the model hallucinating an unwarranted \hline or row-break tag, penalizing the TEDS score despite accurate character recognition.

2. Inter-Column Glyph Ambiguity: Glyphs drawn on implicit 'Ghost Line' boundaries occasionally cause the Swin-Transformer to misassign column embeddings (e.g., fusing a drifting justification into the 'Statement' column), albeit far less frequently than in UniMERNet. Crucially, post-processing 1D strings cannot resolve baseline 'Structural Collapse'. Because sequence encoding discards 2D delimiters like whitespace, column boundaries are irreversibly lost; heuristics cannot reliably unmerge statements given handwriting variability. Therefore, end-to-end VDU fine-tuning is necessary to capture visual topology exactly at extraction.

4.4 Hypothesis Testing Conclusion

We performed a paired t-test on the TEDS scores of UniMERNet and Nougat-Logic across the 2,000 test samples. The resulting p-value was $p<0.001$, which is well below the standard significance level of $\alpha=0.05$.Therefore, we reject the Null Hypothesis (H_0) and accept the Alternative Hypothesis (H_1). There is compelling statistical evidence that the VDU-based approach significantly outperforms the sequence-based approach in digitizing handwritten Hilbert proofs. The trade-off, however, is computational cost. As shown in Table 1, Nougat-Logic is nearly four times slower than UniMERNet (450ms vs. 120ms per page). This latency is directly attributable to both the large amounts of attention that the Vision Transformer uses and the longer sequences that need to be decoded in an autoregressive manner (Markdown tags + content), although the latter will present real-time processing challenges to mobile platforms; however, it is an acceptable trade off given the emphasis on the structural validity and the high level of accuracy desired from the backend batch processing. Processing is also often asynchronous in an educational digitization environment (e.g., batch upload); therefore, a 300 ms increase in latency can be considered negligible when comparing to the costs associated with manually correcting errors due to the lesser accuracy of the baseline.

5. Conclusion and Future Work

5.1 Conclusion

We introduced Nougat-Logic, a Visual Document Understanding (VDU) system designed to resolve the 'structural collapse' prevalent in sequence-based HMER systems. By shifting to a pixels-to-markdown paradigm, we successfully preserved the topological integrity of noisy handwritten logic. Our experiments yield three key takeaways. First, our Elastic-Hybrid Data Engine's distortions and 'ghost lines' are crucial for bridging the Sim-to-Real gap in erratic student handwriting. Second, TEDS is a superior metric to character error rates because it explicitly penalizes logically invalid structural failures. Finally, visual transformers are necessary for parsing complex 2D layouts, evidenced by Nougat-Logic's 95.0% TEDS score, which drastically outperformed the 72.9% UniMERNet baseline.

In essence, we have demonstrated that for the domain of formal methods, structural integrity is as critical as character recognition in formal methods. A digitizer that captures the symbols but loses the structure is unusable to an automated theorem prover. Nougat-Logic presents as an advancement in creating AI systems that respect the rigorous syntax of mathematical reasoning.

5.2 Future Work

While Nougat-Logic establishes a strong baseline, several avenues for future research remain:

- **Model Distillation:** The current 250M parameter model is too heavy for on-device inference (mobile phones/tablets), which is the primary use case for students scanning homework. Future work should explore Knowledge Distillation to compress the capabilities of Nougat-Logic into a smaller, mobile-friendly architecture like MobileNet or a tiny Vision Transformer(ViT).

- **Expansion to Other Formalisms:** The dataset examined for this study was limited to Hilbert-style proofs; future studies should examine other types of mathematical proofs such as Fitch-style natural deduction, semantic tableaux, and Karnaugh maps since all of these methods use significant amounts of 2D spatial structure.

- **Synthetic Dataset Limitations:** A primary limitation of this study is the reliance on synthetically generated data for both training and testing. This presents a lack of real-world validation on actual student handwriting, where distortions are more robust and erasures are present. Collecting real data requires extensive classification and labeling, costing significant labor and time to gather sufficient datasets. Meanwhile, generating new synthetic datasets that simulate actual realistic erasures and distortions is difficult, requiring further research to minimize the Sim-to-Real gap.

ACKNOWLEDGEMENTS

We extend our sincere gratitude to Mrs. Cherry Lyn Santa Romana and Mr. Jay Vince Serato for their invaluable guidance throughout our research and thesis courses.

We also acknowledge the Cebu Institute of Technology–University for the foundation of knowledge provided during our studies and for the support in pursuing research opportunities.

Finally, we wish to express our deep appreciation to our family, friends, and peers for their unwavering love and support, especially Chibble and Cheeto.

REFERENCES

[1] L. Blecher, G. Cucurull, T. Scialom, and R. Stojnic, "Nougat: Neural optical understanding for academic documents," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023.

[2] B. Wang *et al.*, "UniMERNet: A universal network for real-world mathematical expression recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024, pp. 1–10.

[3] G. Kim *et al.*, "OCR-free document understanding transformer," in *Computer Vision – ECCV 2022* (Lecture Notes in Computer Science), vol. 13688, S. Avidan *et al.*, Eds. Cham, Switzerland: Springer, 2022, pp. 498–517.

[4] X. Zhong, E. ShafieiBavani, and A. Jimeno Yepes, "Image-based table recognition: Data, model, and evaluation," in *Computer Vision – ECCV 2020*, vol. 12366, A. Vedaldi *et al.*, Eds. Cham, Switzerland: Springer, 2020, pp. 564–580.

[5] J. Zhu, W. Zhao, Y. Li, X. Hu, and L. Gao, "TAMER: Tree-aware transformer for handwritten mathematical expression recognition," in *Proc. AAAI Conf. Artif. Intell.*, vol. 39, no. 10, pp. 10950–10958, Apr. 2025.

[6] E. J. Hu *et al.*, "LoRA: Low-rank adaptation of large language models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022.

[7] W. Zhao and L. Gao, "CoMER: Modeling coverage for transformer-based handwritten mathematical expression recognition," in *Computer Vision – ECCV 2022*, vol. 13688, Cham, Switzerland: Springer, 2022, pp. 392–408.

[8] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, Art. no. 125, Feb. 2020.

[9] J. Seitz, T. Lengfeld, and R. Timofte, "The return of structural handwritten mathematical expression recognition," in *Proc. Int. Conf. Doc. Anal. Recognit. (ICDAR)*, San Jose, CA, USA, Aug. 2025, pp. 79–95.

[10] E. Alonso, B. Moysset, and R. O. Messina, "Adversarial generation of handwritten text images conditioned on sequences," in *Proc. Int. Conf. Doc. Anal. Recognit. (ICDAR)*, Sydney, NSW, Australia, Sep. 2019, pp. 481–486.

[11] B. Li *et al.*, "Towards scalable training for handwritten mathematical expression recognition," *arXiv preprint arXiv:2508.09220*, 2025.

[12] H. Kasem, A. Abdallah, A. Berroukham, and E. Al Hajj, "Deep learning for table detection and structure recognition: A survey," *arXiv preprint arXiv:2403.05789*, 2024.

[13] T. Guan *et al.*, "PosFormer: Recognizing complex handwritten mathematical expression with position forest transformer," *arXiv preprint arXiv:2407.07764*, 2024.

ReHaPT

Integrating Machine Learning and Computer Vision for a Hand Rehabilitation Recognition System

Jorice Erika Cataluña
College of Computer Studies
De La Salle University
Metro Manila, Philippines
jorice_erika_cataluna@dlsu.edu.ph

Abigail Seperidad
College of Computer Studies
De La Salle University
Metro Manila, Philippines
abigail_seperidad@dlsu.edu.ph

James Archer Paguiligan
College of Computer Studies
De La Salle University
Metro Manila, Philippines
james_paguiligan@dlsu.edu.ph

Dr. Judith Azcarraga, Ph.D.
College of Computer Studies
De La Salle University
Metro Manila, Philippines
judith.azcarraga@dlsu.edu.ph

ABSTRACT

Hand and wrist injuries are common occupational conditions that can reduce productivity and quality of life, yet rehabilitation is often limited by cost, access, and the need for repeated in-person monitoring. In response, a low-cost, webcam-based hand gesture recognition prototype was developed, combining computer vision (OpenCV + MediaPipe Hands) with machine learning to support objective monitoring of tendon-glide rehabilitation exercises. Licensed physical therapists were consulted first to confirm clinically appropriate gesture definitions. Afterwards, a custom dataset was collected from 14 participants performing five tendon-glide stages using both hands. Using participant-based splits to prevent data leakage, 10 SVM variants, 9 KNN variants, and 3 video-based deep learning models (baseline 3D CNN, residual 3D CNN, and hybrid 3D CNN-LSTM) were evaluated. An HMM model was integrated to the best classifier to assess whether the overall exercise sequence was followed or not. Results showed that landmark and angle-based models generalized best, with the baseline RBF SVM achieving a higher test accuracy (89.34%) and macro F1-score (89.19%) than KNN, while the 3D CNN models underperformed under real-world visual variability and limited dataset scale. Sequence recognition improved substantially when the SVM + HMM model in the final system used compressed sequences, reducing jitter and emphasizing true gesture transitions. Overall, our findings support feature-based classification with temporal modeling as a practical foundation for camera-based hand telerehabilitation, while future work should expand data diversity and explore stronger pretrained video backbones for deep learning.

KEYWORDS

Machine Learning, Computer Vision, Hand Telerehabilitation, Gesture Recognition, Sequence Recognition, Deep Learning

1 Introduction

Hand and wrist conditions remain common among working-age Filipinos, especially those exposed to repetitive or poorly supported upper-limb tasks. These conditions often lead to lost workdays and reduced independence [14, 15]. Yet rehabilitation is still mostly clinic-based and therapist-led, which makes patients passive and makes it hard for therapists whether home exercises are done correctly or at all [13]. Telerehabilitation apps exist, but many only show exercise videos or record simple progress and do not actually check movement quality in real time [10]. At the same time, computer vision and machine learning—e.g., MediaPipe- and OpenCV-based hand tracking, or classification models—have shown that markerless, camera-based monitoring is technically feasible and can be run on consumer hardware.

The motivation was the lack of low-cost, webcam-based system that is tailored to clinically relevant hand exercises, that can operate in local, sometimes suboptimal environments, and that includes a light persuasive layer to keep patients engaged. Many sensor-based solutions are accurate but costly or intrusive; many vision-based solutions work in controlled labs but struggle with occlusion, lighting, or patient-specific variation [16]. Moreover, most prototypes stop at recognition and do not

close the loop with feedback, reminders, or progress visualization—features known in persuasive technology to support adherence over time.

In response, this paper presents: (1) the design of a computer-vision hand therapy prototype that uses a standard RGB web camera, MediaPipe/OpenCV for hand landmarking, and ML models to recognize rehabilitation-specific gestures; (2) the integration of simple persuasive strategies—immediate feedback on correctness, progress recommendations, and reminders—to support patient adherence; and (3) an initial user-oriented evaluation utilizing pattern matching techniques and string sequencing techniques similar to NLP techniques.

2. **Methodology**

This section describes how the system was built and tested as a camera-based hand therapy recognition system, exploring different machine learning models — KNN, SVM, 3D CNN, and HMM — to identify a practical approach for a low-cost, software-only rehabilitation tool.

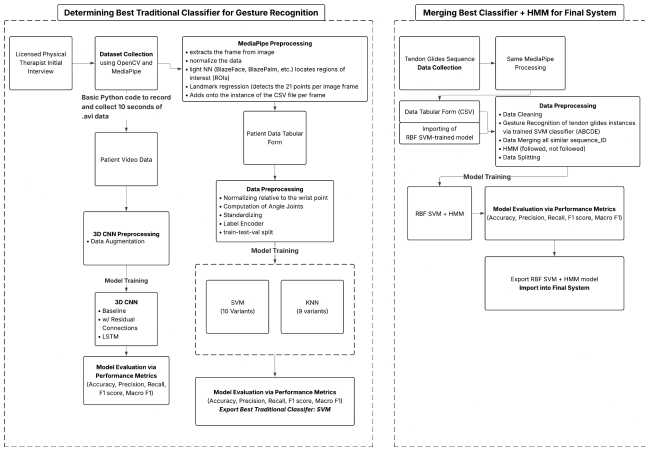


Figure 1. **Methodology Overview**

Figure 1 shows the whole methodology overview of this study. First, therapists were interviewed to determine the medical standards and the common exercise for hand problems. Second, therapist-evaluated tendon glide gestures and the sequence were collected. Third, the initial dataset was cleaned and preprocessed to get the most important and necessary features for gesture model training. Fourth, the different models were trained and evaluated in Python/Jupyter. Fifth, the best trained model is exported, based on the performance metric. Sixth, the best trained model was imported into the HMM model to further improve the temporal performance of our prototype. Finally, the prototype was evaluated with the combination of best classifier + HMM.

2.1 **Participants and Ethics**

In this study, therapists were consulted first before dataset collection — 5 were licensed physical therapists and 1 was a student occupational therapist. The therapists were asked questions including the work process and the patient evaluation. The therapists chose the most common exercise out of the different hand warm-up exercises. From the responses, all are important, but tendon glides was voted as the most common and effective exercise. Thus, the *tendon glides* exercise was selected as the sequence for experimentation.

In the dataset collection, thirteen (13) participants within the 20-30 age range and one (1) 60-year-old participant with no major hand restrictions volunteered to record both hands (left and right) with our Python capture tool. Gesture data (A-E) and sequence data (Tendon Glides Exercise) were collected with 5 repetitions per gesture and sequence. For data privacy, our data collection crops the boundary box of the hand and blurs the participants' faces.

2.2 **Gesture Set and Data Capture**

The rehabilitation movement set is based on tendon glides shown in Figure 2, encoded as five gesture classes: **1** (open/flat), **2** (hook fist), **3** (tabletop/platform), **4** (straight fist), and **5** (full fist).

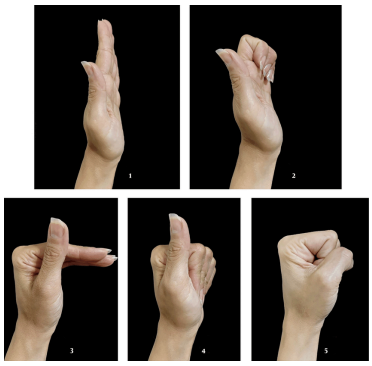


Figure 2. **Tendon Glides**

A standalone Python capture tool (OpenCV + MediaPipe Hands) recorded RGB video from a standard webcam. For each performed gesture, the pipeline: (i) detected the hand, (ii) cropped and standardized the frame region, and (iii) saved frames into class-specific folders (A-E for gesture data & Tendon Glides for sequence data). For KNN and SVM, each gesture instance was recorded as a 10-second

video clip. From these recordings, samples were extracted at 1-, 2-, 3-, and 5-second intervals. However, only the 1-second interval was used in the final dataset, as it provided a greater number of samples for training. The same recordings were also used for the 3D CNN, where the complete 10-second .avi clips served as input. For sequence-based data (Tendon Glides), recordings were captured continuously without a fixed duration until all gestures were performed at a natural pace (approximately 2–3 seconds per gesture). Participants repeated each gesture several times, and execution issues were annotated and cleaned to support later analysis [6]. Figure 3 shows the data collection software used.

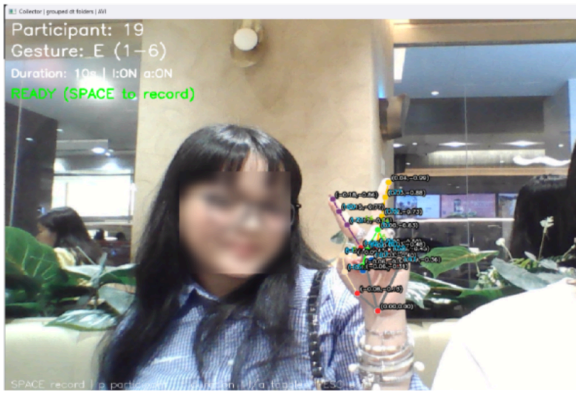


Figure 3. **Outdoor Data Collection**

The data collection setup can be both an indoor and outdoor setting, as long as the software can detect the gestures and the hands properly without jittering too much. For an indoor setup, it is highly recommended that the participants' hands are behind a clean background and proper lighting. For the outdoors, it is recommended to record in the afternoon without intense nor dark lighting. With this type of setup, it greatly reduces the performance of the deep learning model, but not the traditional models.

2.3 **Preprocessing and Sequence Formulation**

After collection, the KNN and SVM dataset were standardized using the following steps:

- Resized frames to a consistent input shape (224x224 pixels) and normalized pixel values.
- Extracted **MediaPipe 21 hand landmarks** per frame; optionally keep a lightweight representation using landmarks rather than raw pixels.
- Computed **joint angle features** per frame to better represent finger articulation and reduce ambiguity in visually similar poses.
- Generated CSV/JSON indexes linking each sequence to its gesture label and metadata.

The 3D CNN was augmented as its preprocessing technique. After EDA and cleaning, Figure 4 shows the final dataset distribution and Table 1 shows the final dataset used for the different models.

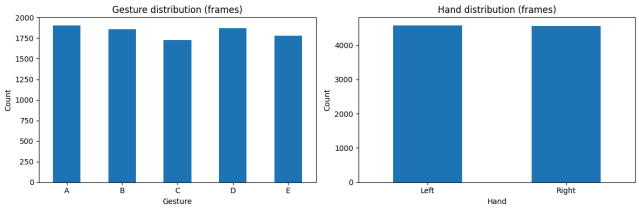


Figure 4. **Final Dataset Distribution**

Table 1. **Final Dataset after Preprocessing**

Model Usage	Gesture Recognition Data	Sequence Data
SVM and KNN	9143 1s frames	N/A
3D-CNN	840 .avi videos	N/A
HMM (for base classifiers) LSTM (for 3D-CNN)	N/A	177 .avi videos

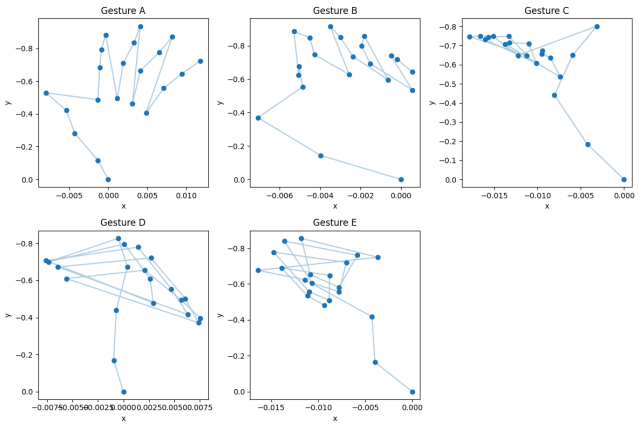


Figure 5. **Mean Landmark Positions (Per Gesture)**

Upon checking the means of the landmark positions (see Figure 5), it can be observed that these values capture the overall spatial configuration of the gestures.

PCA and LDA were explored for further analysis of the clear separability of the gestures using landmarks and angles. Upon trying both, LDA yielded a better visualization for this dataset (see Figure 6). Our analysis from this was that the landmarks were better for positional data (meaning the patient can do a different variety of the gesture, and it will still read the same), and the angles were better to differentiate the gesture.

Figure 6. **Linear Discriminant Analysis (LDA) for both landmarks and angles**

For HMM, per-frame predictions are tokenized (A-E) and concatenated into a detected string, which is then compressed using **Hidden Markov Model (HMM)** due to its temporal strengths. Because the target utilization emphasizes exercise sequencing rather than single-pose detection, the sequences of **"Followed"** and **"Not followed"** are labeled within certain categories. This can be further explained in the Table 2 below:

Table 2. **Example Gesture Sequences for Followed and Not Followed Categories**

Category	Example	Remarks
Followed	A-B-B-C-D-E	Correct sequence, with minor jitters
Not Followed	A-D-D-B-E-C	Incorrect sequence
Not Followed	A-B-D-D-E	Missing "C" gesture
Not Followed	B-A-C-D-E-E	Incorrect start of sequence

2.4 **Feature Sets and Dataset Splits**

After merging the necessary features for the different models, Figure 7 shows the features for KNN and SVM. Figure 8 shows the dataset features for the HMM model. For 3D CNN, the dataset features include path, filename, gesture, and handedness columns..

```
RangeIndex: 9143 entries, 0 to 9142
Data columns (total 65 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   sequence_id     9143 non-null   int64
 1   frame_idx       9143 non-null   int64
 2   participant_id  9143 non-null   int64
 3   gesture         9143 non-null   str
 4   hand            9143 non-null   str
 5   lmk0_x          9143 non-null   float64
 6   lmk0_y          9143 non-null   float64
 7   lmk1_x          9143 non-null   float64
 8   lmk1_y          9143 non-null   float64
 9   lmk2_x          9143 non-null   float64
 10  lmk2_y          9143 non-null   float64
 11  lmk3_x          9143 non-null   float64
 12  lmk3_y          9143 non-null   float64
 13  lmk4_x          9143 non-null   float64
 14  lmk4_y          9143 non-null   float64
 15  lmk5_x          9143 non-null   float64
 16  lmk5_y          9143 non-null   float64
 17  lmk6_x          9143 non-null   float64
 18  lmk6_y          9143 non-null   float64
 19  lmk7_x          9143 non-null   float64
...
 63  middle_ring     9143 non-null   float64
 64  ring_pinky      9143 non-null   float64
dtypes: float64(60), int64(3), str(2)
memory usage: 4.5 MB
```

Figure 7. **Features for the KNN and SVM model**

```
<class 'pandas.DataFrame'>
RangeIndex: 177 entries, 0 to 176
Data columns (total 6 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   sequence_id          177 non-null    int64
 1   participant_id       177 non-null    int64
 2   hand                 177 non-null    str
 3   raw_sequence         177 non-null    object
 4   compressed_sequence  177 non-null    object
 5   follow_status        177 non-null    str
dtypes: int64(2), object(2), str(2)
memory usage: 8.4+ KB
```

Figure 8. **Features for the HMM model**

Table 3 shows the different dataset splits for the different models. The data split is also by participant (i.e. participant 1 will only be seen in the training dataset, participant 2 will only be seen in test, and participant 3 in validation).

Table 3. **Dataset Splits**

Model	Split (train-test-val)
KNN, SVM	70-15-15
3D CNN	70-20-10
HMM	80-20 (train-test)

2.5 **Model Pipelines**

Model development was conducted in Jupyter notebooks for fast iteration. The same underlying dataset and a train/test split procedure were used for all approaches.

KNN and SVM Pipeline (Traditional Classifiers)

- **Input:** frame-level or aggregated sequence-level features derived from the defined feature sets.
- **KNN Variants:** Euclidean, Manhattan, Cosine, Weighted Euclidean, Weighted Manhattan, Weighted Cosine, Weighted Braycurtis, Weighted Chebyshev, Weighted Canberra
- **SVM Kernels:** Linear, Polynomial, RBF, Sigmoid, Linear SVC, and its corresponding tuned versions.
- **Output:** Weighted Cosine is the best KNN variant, but RBF SVM achieved the strongest recognition performance among the evaluated traditional approaches and was integrated into the prototype as one of the primary models.

Advanced 3D CNN with Residual Connections (Deep Learning Model with Best Results)

Table 5. **Advanced 3D CNN with Residual Connections — Key Parameters and Definitions**

Stage / Block	Layer (in order)	Output Shape	# of Params	Purpose
Input	Video clip input	(30, 64, 64, 3)	0	Kept the same standardized clip format as the baseline so comparisons stayed fair across models.
Stem	Conv3D(32) -> BatchNorm -> ReLU -> MaxPool3D	(15, 32, 32, 32)	2,752	Learned basic spatiotemporal cues, then downsampled early to keep computation manageable.
Residual Block 1	Main path: Conv3D(64) -> BN -> ReLU -> Conv3D(64) -> BN | Skip path: Conv3D(64, 1x1x1) -> BN | Add -> ReLU -> MaxPool3D -> Dropout	(3, 8, 8, 128)	~169k	Let the model go "deeper" without becoming unstable by adding a skip connection that preserves earlier information while the main path learns stronger patterns.
Residual Block 2	Main path: Conv3D(128) -> BN -> ReLU -> Conv3D(128) -> BN | Skip path: Conv3D(128, 1x1x1) -> BN | Add -> ReLU -> MaxPool3D -> Dropout	(128)	~673k	Increased feature depth (64->128) so the network could capture more complex motion/shape differences between gesture stages while still keeping gradients stable through the skip route.
Feature Collapse	GlobalAveragePooling3D	(5)	0	Condensed the final feature volume into a single vector, reducing sensitivity to where the hand appeared within the frame.
Classifier Head	Dense(256) -? BatchNorm -> Dropout -> Dense(128) -> BatchNorm -> Dropout -> Dense(5, Softmax)		68,101	Produced probabilities for gestures A-E using a compact decision head, while normalization/dropout helped reduced memorization.
Total			913,413	~913k parameters

Best Classifier + HMM Pipeline (Sequence Modeling)

- **Input:** sequence data (Tendon Glides)
- **Approach:** classify the gesture of the sequence data using the best classifier, merge the sequence ID per participant (resulting in a string sequence, AABBCCDDEE), manually annotate "Followed/Not Followed" (A-B-C-D-E), train the HMM model accordingly
- **Output:** the temporal and trained HMM compressed the data from the raw dataset (AABBCCDDEE to ABCDE).

2.6 **Evaluation and Deployment Artifacts**

The training outputs were exported as:

- (a) .pkl files for the models (KNN, SVM, HMM)
- (b) .pkl files for the encoded gestures and handedness (left hand or right hand)
- (c) .pkl files for the SVM + HMM model

These artifacts were loaded by the GUI prototype (Python + OpenCV + MediaPipe + Tkinter) so the application could run in real time without retraining. Performance metrics tests were conducted to further validate the KNN, SVM, 3D CNN, and HMM models.

3 **Prototype & Results**

In **Model Development**, KNN variants (Euclidean, Manhattan, Cosine, Weighted Euclidean, Weighted Manhattan, Weighted Cosine, Weighted Braycurtis, Weighted Chebyshev, Weighted Canberra) and SVM variants (Linear, Polynomial, Linear SVC, RBF, Optimized RBF) and 3D CNN were trained and tuned using Grid Search. In **Model Evaluation**, the models were compared using the performance metrics (Accuracy, Precision, F1 Score, Macro F1). The figures shown below are the best variant per model.

```
============================================================
Testing Baseline | RBF SVM (SVC)
============================================================
Test Accuracy: 0.8934
Test Macro F1: 0.8919

Test Classification Report:
              precision    recall  f1-score   support

           A       0.84      0.95      0.89       421
           B       0.90      0.95      0.92       396
           C       0.88      0.76      0.81       396
           D       0.92      0.89      0.91       396
           E       0.94      0.91      0.92       418

    accuracy                           0.89      2027
   macro avg       0.90      0.89      0.89      2027
weighted avg       0.90      0.89      0.89      2027
```

Figure 9. **RBF SVM Performance Metrics**

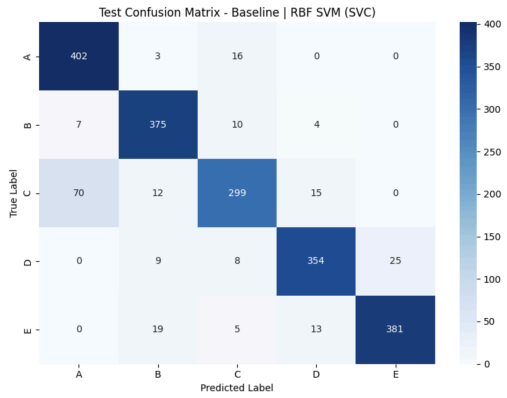


Figure 10. **RBF SVM Confusion Matrix**

As shown in Figure 9, the Baseline RBF SVM achieved the highest accuracy of 89% and a Macro F1-score of 89% among all baseline models. The RBF kernel's ability to capture non-linear relationships between hand landmarks and joint angles likely explains its superior performance. The confusion matrix in Figure 10 confirms that gesture `C' remained the most difficult to classify, with notable confusion between gestures `C' and `A'.

```
============================================================
Testing weighted_cosine
============================================================
Test Accuracy: 0.8293
Test Macro F1: 0.8225

Test Classification Report:
              precision    recall  f1-score   support

           A       0.72      0.98      0.83       421
           B       0.88      0.94      0.91       396
           C       0.92      0.54      0.68       396
           D       0.84      0.86      0.85       396
           E       0.88      0.83      0.85       418

    accuracy                           0.83      2027
   macro avg       0.85      0.83      0.82      2027
weighted avg       0.84      0.83      0.82      2027
```

Figure 11. **Weighted Cosine Performance Metrics**

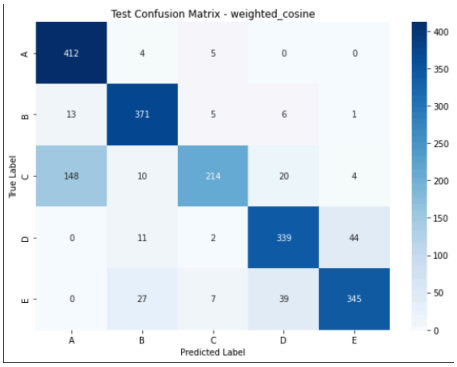


Figure 12. **Weighted Cosine Confusion Matrix**

Weighted Cosine, as seen in Figure 11, achieved an accuracy of 83%. Gesture 'C' has the lowest recall (0.54), which indicated that it was the most misclassified gesture out of the rest of the defined gestures. Further visualizations can be provided with Figure 12, wherein Gesture 'A' and 'C' are often misclassified. Weighted Cosine measures the cosine of the angle between the first and second point, which explains why this type of distance metric is more suitable for the dataset.

Class	Precision	Recall	F1-Score	Support
A	0.47	0.86	0.61	22
B	0.75	0.47	0.58	19
C	0.48	0.50	0.49	20
D	0.59	0.77	0.67	26
E	1.00	0.05	0.09	21
Accuracy			0.55	108
Macro Avg	0.66	0.53	0.49	108
Weighted Avg	0.66	0.55	0.50	108

Figure 13. **Advanced 3D CNN with Residual Connections Performance Metrics**

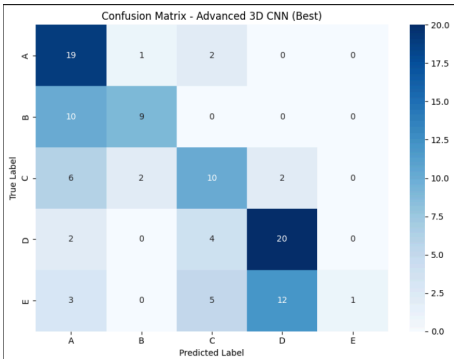


Figure 14. **Advanced 3D CNN with Residual Connections Confusion Matrix**

Using the best-tuned configuration, Advanced 3D CNN with residual connections achieved a test accuracy of 0.5463, indicating the strongest overall generalization among the CNN models we evaluated. As summarized in Figure 13, the model identified A and D most reliably, with high recall values (A:0.86, D:0.77), while performance for B and C was more balanced but moderate (B F1: 0.58, C F1: 0.49). These trends are directly supported by Figure 14, where A shows a strong diagonal count (19 correct out of 22) and D likewise has a strong diagonal (20 correct out of 26), confirming the high recall reported.

```
              precision    recall  f1-score   support

    followed       0.67      0.94      0.78        17
not_followed       0.94      0.65      0.77        23

    accuracy                           0.78        40
   macro avg       0.80      0.80      0.77        40
weighted avg       0.82      0.78      0.77        40
```

Figure 15. **Baseline RBF SVM + HMM Compressed Data Performance Metrics**

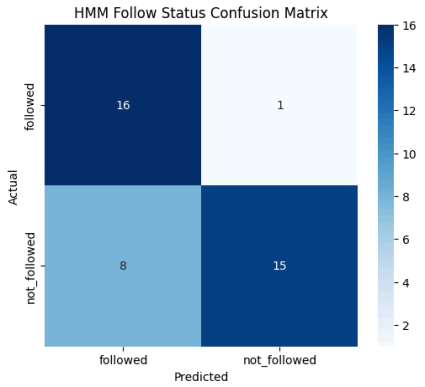
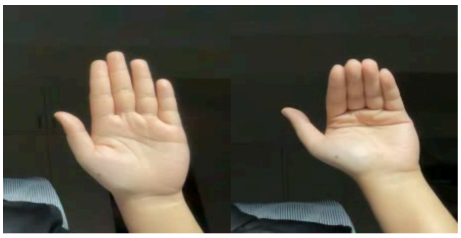


Figure 16. **Baseline RBF SVM + Compressed Data Confusion Matrix**

After combining the best classifier (baseline RBF SVM) with the HMM model, it was evident that using the compressed data representation compared to raw sequences with repetitive and jitters (Figure 15 and Figure 16). This demonstrates the importance of preprocessing data to highlight meaningful temporal dynamics for sequential models.

In the confusion matrices, it was evident that for both Baseline RBF SVM and Tuned Weighted Cosine that Gestures "A" and "C" are often misclassified. After error analysis, we have found that in certain angles of C, it can be read as A. This can be visualized in Figure 17.



(A) (C)

Figure 17. Gesture A and C

Baseline RBF SVM (89.34% acc) is better than Tuned Weighted Cosine KNN (82.93% acc). 3D CNN (~below 50% acc) was discarded and marked as exploratory since the dataset, parameters, and system architecture was too small and noisy for the deep learning classifier, resulting in unreliable results. Since the best classifier was SVM, it was merged with temporal HMM using compressed data. The final system is a combination of Baseline RBF SVM + HMM to further improve how the model reads the exercise sequence.

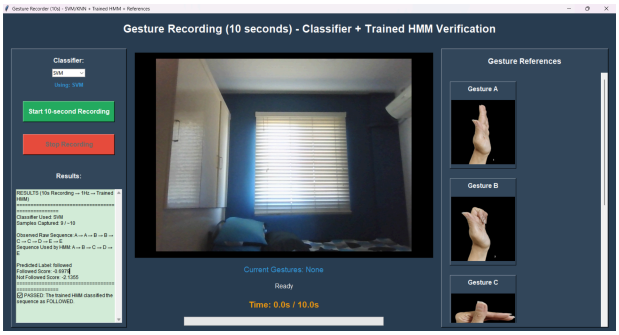


Figure 18. **Final System using SVM + HMM for Gesture Classification and Sequence Recognition**

Figure 18 shows the real-time GUI used for gesture pattern assessment during rehabilitation sessions. The application continuously detects and classifies tendon-glide gestures with the custom SVM model, then displays live feedback to support monitoring and correction. The SVM model uses a combined feature representation composed of computed joint angles, normalized 2D landmark coordinates, and hand-type encoding to better capture finger articulation and reduce ambiguity across visually similar poses. During each 10-second assessment run, the system records stable gestures (held for at least three consecutive frames) at an effective rate of two detections per second. However, HMM improved these raw sequences with compressed data for better readability of the model between "Followed" and "Not Followed"

```
RESULTS (10s Recording -> 1Hz -> Trained HMM)
==================================================
Classifier Used: SVM
Samples Captured: 9 / ~10
Observed Raw Sequence: A -> A -> B -> B -> C -> C -> D -> E -> E
Sequence Used by HMM: A -> B -> C -> D -> E
Predicted Label: followed
Followed Score:     -0.6979
Not Followed Score: -2.1355
==================================================
PASSED: The trained HMM classified the sequence as FOLLOWED.
```

Figure 19. **Final System Results**

Figure 19 illustrates how the final output is produced from a single assessment run. As the user performs the exercise, the model generates per-frame gesture labels that are only logged once the user's hand gesture remains stable across consecutive frames, producing a raw detected sequence of A-E tokens. This detected sequence is then compressed to lessen repetitions and accidental noise using HMM, and the resulting similarity score determines whether the pattern execution is ABCDE as "Followed".

3. **Conclusion and Future Work**

This study demonstrates that classical machine learning classifiers, specifically Support Vector Machines (SVM) and K-Nearest Neighbors (KNN), were capable of effectively classifying hand gestures when supported by structured landmark- and angle-based features. Despite the relative simplicity, these models achieved strong multi-class performance and stable generalization on unseen test data. In addition, Hidden Markov Models (HMMs) were explored to capture temporal dynamics, and experiments revealed that using both "followed" and "not followed" sequences, as well as compressed gesture sequences, allowed the HMMs to model transitions between gestures most effectively. Though there are still some limitations to our study (collection setup, deep learning architecture, and exercise limitations), the results of our study indicate that while classical classifiers perform reliably on static geometric features, HMMs can complement the classifiers by capturing temporal patterns and movement transitions, providing a richer representation of hand gesture sequences for recognition tasks.

Future studies may explore the behavioral characteristics underlying transitions between gestures. While the present work focuses on classification accuracy and sequential validation, further investigation into the biomechanical and behavioral factors influencing gesture shifts could provide deeper insight into movement quality. Analyzing transition dynamics—such as hesitation, incomplete articulation, or compensatory motion—may improve the system's ability to detect subtle abnormalities in hand movement.

Additionally, expanding the dataset to include a wider range of non-standard or irregular gestures would enhance real-world applicability. Incorporating gestures that simulate hand impairments, such as trigger finger behavior, reduced range of motion, or delayed finger extension, would allow the system to better model pathological movement patterns. This would support more realistic rehabilitation monitoring scenarios and improve robustness when deployed outside controlled environments.

As our dataset was not validated with an expert in the field of medical or physical therapy, another future work proposal is to finalize the proposed system dataset and system. In other words, rather than only interviewing the therapists pre-data collection, a better approach to complete this study is to interview experts post-data collection and the proposed system, itself.

REFERENCES

[1] A. Akdag and O.K. Baykan (2024). Multi-Stream Isolated Sign Language Recognition Based on Finger Features Derived from Pose Data. Electronics, 13(8), 1591.

[2] S.A.M. Al-Juboori, H. Almutairi, R. Almajed, A. Ibrahim, and H.M. Gheni (2022). Detection of Hand Gestures with Human Computer Recognition by Using Support Vector Machine. Periodicals of Engineering and Natural Sciences, 10(32), 467–476.

[3] M. Al-Saidi, Á. Ballagi, O.A. Hassen, and S.M. Saad (2024). Type-2 Neutrosophic Markov Chain Model for Subject-Independent Sign Language Recognition: A New Uncertainty-Aware Soft Sensor Paradigm. Sensors, 24(23), 7828.

[4] B. Amirlak and T.W. Wolff (n.d.). Non-surgical Management and Post-surgical Rehabilitation of Carpal Tunnel Syndrome: An Algorithmic Approach and Practical Guideline (Figure 3: Tendon Gliding Exercises). ResearchGate. https://www.researchgate.net/figure/1-5-Tendon-gliding-exercises-1-straight-2-straight-fist-3-table-top-4-fist_fig3_343223691

[5] J. Castro-Vargas, B. Zapata-Impata, P. Gil, J. Garcia-Rodriguez, and F. Torres (2019). 3DCNN Performance in Hand Gesture Recognition Applied to Robot Arm Interaction. In Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP) (pp. 802–806).

[6] A. Černek, J. Sedmidubský, and P. Budíková (2024). REHAB24-6: Physical Therapy Dataset for Analyzing Pose Estimation Methods. Zenodo. DOI: 10.5281/zenodo.13305826

[7] D. Cook (2023). Speaking in Gestures: Designing a Gesture Library Using Hand Tracking. thirteen23. https://www.thirteen23.com/updates/speaking-in-gestures

[8] G. Fronteddu, S. Porcu, A. Floris, and L. Atzori (2022). A Dynamic Hand Gesture Recognition Dataset for Human-Computer Interfaces. Computer Networks, 205, 108781.

[9] S.B. Hamida, H. Mrabet, and A. Jemai (2022). How Differential Privacy Reinforces Privacy of Machine Learning Models? In Communications in Computer and Information Science (CCIS) (pp. 661–673).

[10] Health Resources & Services Administration (HRSA) (2022). What Is Telehealth? (Accessed: 14 Nov 2025). https://www.hrsa.gov/telehealth/what-is-telehealth

[11] B.T. Jijo and A.M. Abdulazeez (2021). Classification Based on Decision Tree Algorithm for Machine Learning. Journal of Applied Science and Technology Trends, 2(1), 20–28.

[12] A. Kaku, K. Liu, A. Parnandi, H.R. Rajamohan, K. Venkataramanan, A. Venkatesan, A. Wirtanen, N. Pandit, H. Schambra, and C. Fernandez-Granda (2022). StrokeRehab: A Benchmark Dataset for Sub-section Action Identification. Advances in Neural Information Processing Systems, 35, 1671–1684.

[13] J. Kim, W. Park, and J. Kim (2020). Quantitative Evaluation of Stroke Patients' Wrist Paralysis by Estimation of Kinematic Coefficients and Machine Learning. Sensors and Materials, 32(3), 981.

[14] J.L. Lu (2022). Trend of Work-related Injuries in the Philippines from 2010–2020. Acta Medica Philippina, 56(19).

[15] C.D. Mapa (2024). Highlights of the 2021/2022 Integrated Survey on Labor and Employment: Module on Occupational Injuries and Diseases: 2021. Philippine Statistics Authority (Accessed: 5 Feb 2025). https://psa.gov.ph/content/highlights-20212022-integrated-survey-labor-and-employment-module-occupational-injuries-and

[16] M.B. Mendoza, E. Job, and C. Bona (2022). HAND AID: Hand Orthotic Device with Strength Monitoring System. In 2022 E-Health and Bioengineering Conference (EHB) (pp. 1–4). IEEE. DOI: 10.1109/ehb55594.2022.9991687

[17] A. Miron, N. Sadawi, W. Ismail, H. Hussain, and C. Grosan (2021). IntelliRehabDS (IRDS)—A Dataset of Physical Rehabilitation Movements. Data, 6(5), 46.

[18] D. Powers (2008). Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. ResearchGate. https://www.researchgate.net/publication/228529307_Evaluation_From_Precision_Recall_and_F-Factor_to_ROC_Informedness_Markedness_Correlation

[19] J. Shin, A.S.M. Miah, M.A.M. Hasan, K. Hrooka, K. Suzuki, H.-S. Lee, and S.-W. Jang (2023). Korean Sign Language Recognition Using Transformer-Based Deep Neural Network. Applied Sciences, 13(5), 3029.

[20] S. Yamato and A. Ben Abdallah (2022). Study of Deep Learning-based Hand Gesture Recognition Toward the Design of a Low-cost Prosthetic Hand. SHS Web of Conferences, 139, 03006.

Sign Lang: Developing a VR Game for FSL Learning

Michelle Kim Angela C. Abarico
Department of Information Systems and Computer Science
Ateneo de Manila University
Quezon City, Metro Manila, Philippines
michelle.abarico@student.ateneo.edu

Caryl Rae T. Chan
Department of Information Systems and Computer Science
Ateneo de Manila University
Quezon City, Metro Manila, Philippines
caryl.chan@student.ateneo.edu

Eric Cesar E. Vidal Jr.,
Department of Information Systems and Computer Science
Ateneo de Manila University
Quezon City, Metro Manila, Philippines
evidal@ateneo.edu

ABSTRACT

The study explores the design and evaluation of a virtual reality (VR) game for learning Filipino Sign Language (FSL) letters and numbers, with the broader aim of promoting Deaf inclusivity and addressing accessibility gaps in education. Developed in Godot 4.3 for the Meta Quest 3, the system integrates XRHandTracker, Hand Pose Detector, and a Hidden Markov Model to recognize FSL gestures in real time. The game includes tutorial levels on the alphabet and numbers, followed by interactive scenarios of progressive difficulty set in three different contexts. This research investigates the effectiveness of VR-game-based learning in improving retention and accuracy of FSL, as well as the usability of the system and user experience. Sub-questions further examine whether game-based learning enhances learning speed, confidence, and motivation, as well as the challenges and limitations of VR in sign language education. Effectiveness is assessed by an analysis of player results, complemented by evaluations of usability, user experience, and user performance.

KEYWORDS

VR, Virtual Reality, Filipino Sign Language, Education, Hand-tracking

1 Introduction

1.1 Context of the Study

Virtual Reality (VR) is a computer-generated environment with potential across educational levels and fields [1]. While VR is predominantly associated with video gaming, its applications extend to training simulations, prototype design, and visualization of real-world, three-dimensional spaces. It enables the simulation of diverse environments and interactions, allowing users to safely gain knowledge.

Research has demonstrated that VR enhances learnability by providing emotional support and reducing cognitive load, while also reducing anxiety, improving motivation, confidence, language proficiency, and comprehension, and preventing cognitive overload [2][3][4][5].

The Deaf community possesses its own distinct culture and language, which differ from the norms of hearing individuals [6]. Filipino Sign Language (FSL), recognized as the Philippines' national sign language in 2018 through the Filipino Sign Language Act, remains under-researched in Southeast Asia [7]. Deaf children formally acquire FSL in elementary school through Deaf-to-Deaf and Deaf-to-hearing interactions. However, without efforts to improve accessibility, the burden of effective learning remains on the Deaf population, who must conform to Hearing-centered methods that are often poorly suited to their needs [8]. Hence, this study specifically targets hearing individuals, emphasizing the need for them to learn FSL as a step toward more equitable and inclusive communication with the Deaf community.

1.2 Research Objectives

The researchers aim to create an educational simulation VR game that teaches beginner-level FSL, specifically the alphabet and numbers 1-10. The game aspires to educate users about the unique experiences and challenges of the Deaf community, leading to further promotion of Deaf-inclusivity and accessibility in education. This study addresses challenges in learning and recognizing FSL through an interactive VR game where users view sign demonstrations and receive real-time feedback using built-in sensors. Unlike video-based learning, it emphasizes active sign production, with simulation levels introducing contextual interaction.

1.3 Research Questions

Given the above research objectives, this study aims to answer the following questions:

- How effective is VR-game-based learning in improving the retention and accuracy of FSL?
  - Does VR-game-based learning (rewards and progression) affect users' learning speed and confidence in using basic FSL?

- Does VR-game-based learning (rewards and progression) affect motivation for FSL learning?
- What are the challenges and limitations of using VR for FSL learning, and how can they be addressed?

1.4 Scope and Limitations

Using a Meta Quest 3 HMD, the study is limited by hardware constraints, particularly occlusion, where fingers obstruct each other and disrupt tracking. Additionally, the tracking volume is limited; accuracy diminishes significantly when hands move beyond the tracking area or camera. Another challenge is fingertip precision, where subtle movements or intricate gestures are difficult to distinguish, leading to inaccuracies.

The limitations of the hardware meant that the scope of this study is limited to beginner-level FSL. As such, more advanced or complex signs are not included in this research. While VR can support more advanced spatial aspects of sign language, this study prioritizes foundational skill acquisition and interaction-based learning. Additionally, it is not designed to replicate the comprehensive feedback provided by a qualified instructor in a classroom setting. The game serves as a supplementary tool for learning FSL, not a replacement for face-to-face FSL interaction or instruction.

1.5 Significance of the Study

Learning sign language enables effective communication with individuals who are Deaf or non-speaking, creating more inclusive spaces. FSL learning also creates a more thorough understanding and appreciation of Deaf culture, contributing to a more Deaf-inclusive society through education. The current situation of Deaf education in the Philippines reveals that integrating FSL into education for the Deaf and non-speaking remains a significant challenge [8]. Hence, the Sign Language recognition system is greatly sought after because it can help bridge the communication gap between Deaf and hearing individuals. From a technological standpoint, the study advances VR-based hand tracking by improving recognition of similar static and dynamic signs.

2 Review of Related Literature

El Ghoul & Othman [9] examined the effectiveness of a VR system in teaching Qatari Sign Language signs to 52 students from primary and higher education. Using a signing avatar in an immersive 3D environment (ClassVR and HTC Vive Cosmos with controllers), participants learned new gestures, and their comprehension and retention were assessed. Findings show that VR's interactivity engaged young children and was more effective than video-based instruction for learning basic sign language. However, limitations remain: users suggested adding directional arrows and animated cues for clarity; the system supports only beginner-level Arabic Sign Language with no progression; and participants recommended story-based elements like interactive picture books.

A paper by Economou et al. [10] details the ongoing development of a 3D VR interactive game that integrates scaffolded instruction and gamification to teach British Sign Language (BSL) alphabet. The system follows a three-level approach: "Learn" (watch sign demonstrations), "Practice" (form signs via desktop with Mediapipe or VR with hand tracking), and "Assess" (test performance in a game against a virtual enemy). It tracks user progress (video views, practice attempts, eye-tracking) to improve learning effectiveness. However, the system is limited to the BSL alphabet, remains under development with no evaluation of long-term impact, engagement, or scalability, and lacks real-world testing, limiting its assessment in authentic educational contexts.

Mazhari et al. [11] presented a VR game to teach Iranian Sign Language (ISL) to children, aiming to enhance communication opportunities for individuals with hearing disabilities. Played on the Oculus Quest 2, the game integrates a Convolutional Neural Network (CNN) and a Long Short-Term Memory (LSTM) network, achieving test accuracies of 99.7% and 99.1%, respectively. Results show improvement in participants' performance, with players engaging 195.36% longer in the latter half of their attempts compared to the first half. However, limitations include a small sample (5 adults) and only 8 ISL signs, with no progressive levels, reducing the challenge and long-term engagement.

3 Methodology

This chapter outlines the methodology used in developing and evaluating the Sign Lang system. The discussion covers hardware and software design, game design, iterative development, and the conducted user research.

3.1 System Design

Meta Quest hand and body tracking uses real-time camera analysis to estimate the position and orientation of the user's hands and body [12]. This tracking was used alongside a hand pose detection system developed by Malcolm Nixon for Godot [13]. Additionally, the system design and learning content were informed by consultation with a Deaf subject matter expert who is a daily user of FSL.

3.1.1 System Limitations. The VR framework uses Godot version 4.3, incorporating the OpenXR plugin [14], XRTools plugin [15], and the XR Hand Pose Detector plugin and XRTools [13]. The framework was implemented in GDScript.

For user privacy, Godot's XR Server does not provide direct access to the HMD camera. Instead, it outputs the positions of key finger bones, which are mapped onto a 3D skeleton and

retrieved as relative coordinates. Estimated hand size is generated at the start of each session; however, since body scale, hand size, and pose data are derived from generic models and standardized values, they are not unique to individuals. Consequently, hand tracking accuracy may vary due to differences in hand sizes.

Unlike external devices like the Leap Motion Controller, which specializes in precise and detailed hand tracking, the Meta Quest 3's tracking is limited by the visibility of hand placements. Some letters of the alphabet, particularly those requiring obscured hand positions, may not be detected correctly due to the mentioned limitations.

3.1.2 Godot Plugins. The following Godot plugins were used to enable sign language gesture recognition:

1. **OpenXR** provides a unified API for XR applications to interact with XR platforms. In Godot, platform-specific details are abstracted through a modular system centered on the XRServer, which manages XR components [16].
2. **XR Tools** is a toolkit that provides XR mechanics such as object and UI interaction. This is compatible with OpenXR and WebXR runtimes, allowing implementation of XR functionality [17].
3. **XR Hand Pose Detector** processes OpenXR hand-tracking data and detects standard and user-defined poses (Table 3.1), triggering signals when specific hand configurations are recognized.

Table 3.1 **Pose Components [13]**

Type	Dominant Affective State
Flexion	The angle (in degrees) of a finger's proximal joint curving into the palm to make a fist.
Curl	The curl (in degrees) of a finger from the proximal to the distal joints.
Abduction	The spread (in degrees) between two selected fingers.
Tip Distance	The distance (in millimeters) between the tips of two selected fingers.

3.1.3 FSL Letters for Implementation. The development process involved configuring the XRHand interface, creating pose definitions for all letters and numbers 1-10, and testing the recognition accuracy. The game logic uses the detected poses to trigger in-game interactions, such as score increases, corrective feedback, and in-game navigation (e.g., signifying that player is ready to proceed).

3.1.4 Hidden Markov Model as Tie-Breaker. To improve scoring accuracy for similar hand poses, the game integrates a Hidden Markov Model (HMM) unigram as a tie-breaking mechanism [18]. This predicts the succeeding state based on the frequency of appearance in relation to preceding states. The present hand pose detection score is also weighted by its frequency of appearance in the HMM in relation to the previously performed pose. Certain poses—such as "2" and "V," "6" and "W," "U" and "H," "G" and "L," "9" and "F," "C" and "O," and "D," "Z," and "1" and poses affected by occlusion like "M" and "N," "S" and "A"—were difficult to distinguish based solely on hand-tracking data. The HMM provides a probabilistic bias based on contextual pose likelihoods, helping resolve ambiguities between similar poses. It was trained on the set of words used and defined in each level. In scoring, a normalized probability derived from the HMM's transition table is multiplied with the original fitness score, which measures the accuracy of the detected hand pose.

3.2 Game Design

3.2.1 Overview. Our educational VR simulation game, *Sign Lang* (*Lang* being a bilingual wordplay on the Filipino word for "only"), is designed to teach foundational FSL to beginners and individuals with no prior knowledge of sign language. The game creates an environment where players can learn FSL through guided practice and interactive conversations. The estimated playtime is 30-60 minutes, with time limits for every level.

Following a scaffolded learning approach, the core objective of the game is to help players master essential FSL components (alphabet, fingerspelling, and numbers) before applying them in conversational scenarios.

3.2.2 Progression. In Levels 1 and 2, progression is sign-gated—players must perform each sign correctly to advance. Because of this, completing the entire word within a time limit of 8 minutes demonstrates accuracy. These tutorial levels use a pass–fail system: finishing all required signs before the timer ends results in a pass, otherwise it results in a failure. The timer prepares players for the shorter, more dynamic time limits used in the simulation levels.

In levels 3 to 5, performance is currently evaluated using a 50% passing score. Based on ongoing testing, increasing passing scores will be implemented: Level 3 - 50%, Level 4 - 55%, and Level 5 - 60%, which will allow for more progressive difficulty across levels. The conversation levels follow a similar sign-gated approach, and players must finish within the given time to simulate a real conversation where a person must finish what they need to sign. Hence, levels 3 and 4 last 8 minutes, with level 4 having a much longer conversation, making it more difficult. Level 5 is limited to 3 minutes to simulate an emergency, wherein the player must sign correctly and quickly.

3.2.3 Gameplay. The game begins in a classroom with photos of signs on the walls for Levels 1 and 2 (Figure 1), where players use images to learn the FSL signs. The game is

introduced, and players are given unlimited time to look at the signs around the room. Once they are ready for the test, they will sign a "thumbs up", then the game and timer begin. In levels 1 and 2, players sign a set of words and numbers that will appear in future levels.

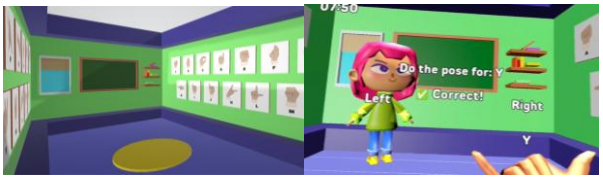


Figure 1: **Tutorial Levels 1 and 2**

As seen in Figure 2, Levels 3 to 5 simulate conversations in progressively more complex scenarios, taking place in VR environments: Café with a Barista (Level 3), Post Office with a Postman (Level 4), and Restaurant during an emergency (Level 5). Each scenario includes a scripted conversation with prompts and response options, where the player must sign replies. After three incorrect attempts, visual hints appear, and players are prompted to try again.

Figure 2: **Simulation Levels 3 (top-left), 4 (top-right), and 5 (bottom)**

3.2.4 Accessibility Features. Sign Lang was designed to be inclusive and user-friendly. All instructions and feedback are available in text (subtitles) and voiceover. The game is stationary and can be played either sitting or standing, providing better accessibility for players with limited mobility.

3.2.5 Design Features. Sign Lang uses audio that enhances interactivity with sound effects for correct answers and level completions. Deaf culture tips are displayed in the rooms to promote sensitivity, including how to get attention and using "Deaf" instead of "hearing-impaired."

3.3 Iterative Development of the System

This study uses an iterative development approach, where the game was repeatedly designed, tested, and refined based on technical and usability findings, with revisions continuing alongside evaluation. The following subsections outline the major development iterations of Sign Lang.

3.3.1 Preliminary Feasibility and Hand Detection Testing. An early version (Figure 3) tested the feasibility of hand detection. Fitness scores and thresholds were defined for all alphabet letters, later extended to numbers.

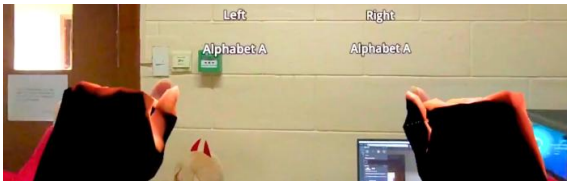


Figure 3: **Initial hand detection test using Malcolm Nixon's XRHandTracker plugin**

3.3.2 Development of Tutorial Levels. Level 1 introduced letter spelling practice, while Level 2 used the same system for numbers. Occlusion and similarities of certain poses motivated the use of a Hidden Markov Model to improve detection. Levels were timed as a mechanism to enforce user-responsiveness while maintaining the easy and self-paced nature of tutorial levels.

3.3.3 Development of Simulation Levels (Levels 3, 4, 5). Levels 3, 4, and 5 introduced conversation-based interactions in a café, post office, and emergency simulation. A candidate word array allowed the system to infer the intended word from the letters signed, and visual hints were added for incorrect attempts. A timer was also used to measure user-responsiveness by enforcing time-limited completion, and a Hidden Markov Model was used to improve detection.

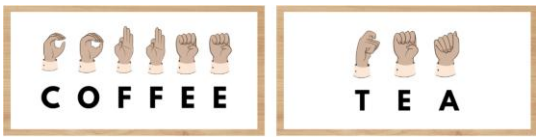


Figure 4: **Hints provided when players sign incorrectly**

3.4 User Research

The evaluation focused on hearing participants, as the system is designed to support non-Deaf learners in acquiring FSL for communication with Deaf individuals. To evaluate the effectiveness of VR-game-based learning, three dimensions of the VR learning environments are evaluated and compared: user performance, user-perceived usability, and user experience. Usability is measured using the System Usability Scale (SUS) [19].

User experience is measured using the User Experience Questionnaire (UEQ) proposed by Schrepp et al. [20], which assesses six scales: Attractiveness, Efficiency, Perspicuity, Dependability, Stimulation, and Novelty. To collect data, participants complete a 26-item questionnaire, each rated on a 7-point Likert scale.

4 Results

The study includes two testing phases with ethics clearance from the institution's research ethics office. Preliminary results taken from 23 volunteer testers from the first testing phase are presented below. Testers ranged from ages 19 to 25 (M = 21.09 ± 1.09); 47.8% male, 34.8% female testers, 8.70% non-binary, 4.35% agender, and 4.35% undisclosed.

The second phase of testing is currently underway.

4.1 User Performance

Levels 1 and 2 are pass-or-fail levels. If the player finishes within the set time, they pass; otherwise, the score is recorded as a failure. Levels 3 to 5 are scored according to accuracy, reported as a percentage. The number of participants is determined by who elects to play or who can play. Failure of a previous level makes the player unable to play the next level. These performance results are summarized in Tables 4.1 to 4.4.

Table 4.1 **Number of Attempts to Pass Levels 1 to 5**

Level	Number of Players	Ave. Attempts Attempt to Pass	Standard Deviation
1	13	2.15	0.69
2	23	1	0
3	23	1.65	0.83
4	16	1.31	0.48
5	16	1.31	0.48

Table 4.2 **First Attempt Scores for Levels 3 to 5**

Level	Number of Players	Ave. First Attempt Score	Standard Deviation
3	23	51.09	11.97
4	20	50.85	12.55
5	12	70.5	21.57

Table 4.3 **Last Attempt Scores for Levels 3 to 5**

Level	Number of Players	Ave. Final Attempt Score	Standard Deviation
3	13	65.69	8.10
4	7	51.57	14.43
5	5	62.8	12.95

Table 4.4 **Replay Information**

Level	Number of Players	Ave. Number of Plays	Standard Deviation
1	23	2.22	0.60
2	23	1	0
3	23	1.74	0.81
4	22	1.36	0.58
5	18	1.17	0.62

4.2 System Usability Scale (SUS)

All 23 participants completed the System Usability Scale (SUS) after gameplay. Results are aggregated in Table 4.5.

Table 4.5 **System Usability Scale (SUS) Results**

Average Rating	Standard Deviation	Minimum Score	Maximum
69.89%	17.49	35	95

4.3 User Experience Questionnaire (UEQ)

All 23 participants completed the User Experience Questionnaire (UEQ) after gameplay and accomplishing the SUS. Results are presented in Table 4.6 and visualized in Figure 5.

Table 4.6 **User Experience Questionnaire (UEQ) Results**

Scale	Mean	Variance
Attractiveness	2.058	0.81
Perspicuity	1.967	0.56
Efficiency	1.348	1.35
Dependability	1.511	0.75
Stimulation	2.272	0.75
Novelty	1.696	0.59

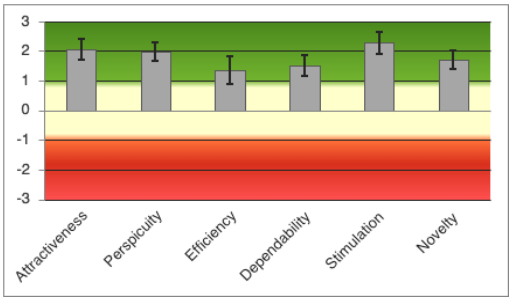


Figure 5: **User Experience Questionnaire (UEQ) Results**

4.3 Interpretation of Preliminary Results

Preliminary user performance data suggest that participants demonstrated improved recall and execution with repetition, measured via scores and replay information. There is also an expected number of replays, with most testers motivated to replay a failed level to progress. Fewer testers electively replayed, possibly attributed to the limited time allotted for each tester.

Game-based elements, such as progressive levels and contextual scenarios, appear to positively influence user experience. The system garnered a preliminary SUS rating of 69.89%, with a description of "good" usability.

Preliminary UEQ results reveal positive user experience along all scales of measurement. Participants described the game as fun and engaging, appreciated the repetition of common words for learning, and found the setting and music appealing. User observation indicates that long levels and incorrect sign detection were the primary cause of frustration. Users recognize the game as a valuable and socially meaningful initiative. Qualitative feedback reasonably indicates that usability issues did not significantly affect user experience.

Preliminary testing results for Sign Lang suggest an overall positive user experience. However, results also highlight issues with level difficulty and system usability. To address these issues, future development iterations will focus on refining hand pose parameters and level difficulty and length based on user feedback. Tester scores in the current iteration may inform further revisions to scoring methods. Usability testing reveals areas requiring improvement, such as adjusting detection delays, refining level difficulty, revising scoring mechanisms, and providing clearer visual and auditory instructions for gameplay. Furthermore, changes to game elements—some already detailed in Section 4.3—may potentially improve user performance and usability scores.

5 Conclusion

Improved user performance throughout all simulation levels indicates that VR-game-based learning is effective in improving the retention and accuracy of FSL basic poses.

The replay information collected during user testing indicates that replaying improves retention and accuracy of alphabet and number hand poses. User feedback indicates that repetition is beneficial in improving user confidence and speed. Further, numerical accuracy scores, as corroborated by player observations, indicate that users are motivated to replay levels in order to progress or to improve their scores.

VR, without augmentation of other camera-based sensors devices, only limits FSL to hand poses and does not yet capture other dimensions of FSL, like gesture size, speed, and facial expressions, which would be engaged to capture the nuances of signed communication. Augmentation with additional sensors or devices may be beneficial in detecting more complicated gestures and integrating gesture size, speed, and facial expressions.

REFERENCES

[1] Á. Antón-Sancho, D. Vergara, and P. Fernández-Arias, "Quantitative analysis of the use of virtual reality environments among higher education professors," Smart Learn. Environ., vol. 11, no. 1, p. 13, Mar. 2024, doi: 10.1186/s40561-024-00299-5.

[2] H.-L. Huang, G.-J. Hwang, and C.-Y. Chang, "Learning to be a writer: A spherical video-based virtual reality approach to supporting descriptive article writing in high school Chinese courses," Br. J. Educ. Technol., vol. 51, no. 4, pp. 1386–1405, 2020, doi: 10.1111/bjet.12893.

[3] L. Hsu, "Exploring EFL learners' acceptance and cognitive absorption at VR-Based language learning: A survey and experimental study," Heliyon, vol. 10, no. 3, p. e24863, Jan. 2024, doi: 10.1016/j.heliyon.2024.e24863.

[4] T. Thrasher, "The Impact of Virtual Reality on L2 French Learners' Language Anxiety and Oral Comprehensibility," CALICO J., vol. 39, no. 2, pp. 219–238, Jun. 2022, doi: 10.1558/cj.42198.

[5] Y.-J. Lin and H. Wang, "Using virtual reality to facilitate learners' creative self-efficacy and intrinsic motivation in an EFL classroom," Educ. Inf. Technol., vol. 26, no. 4, pp. 4487–4505, Jul. 2021, doi: 10.1007/s10639-021-10472-9.

[6] "Science helps promote inclusivity through use of Filipino sign language." Accessed: Jan. 24, 2025. [Online]. Available: https://www.stii.dost.gov.ph/1762-science-helps-promote-inclusivity-through-use-of-filipino-sign-language

[7] "An Act Declaring The Filipino Sign Language As The National Sign Language Of The Filipino Deaf And The Official Sign Language Of Government In All Transactions Involving The Deaf, And Mandating Its Use In Schools, Broadcast Media, And Workplaces – RA 11106 | National Council on Disability Affairs." Accessed: Jan. 24, 2025. [Online]. Available: https://ncda.gov.ph/disability-laws/republic-acts/ra-11106/

[8] A. L. Pizarro, I. A. Mendoza, R. Lo, and C. T. Guiao, "What Inclusive Education Means for d/Deaf Learners in the Philippines: Considerations in Designing a Deaf-Inclusive Educational Model," Oscar M Lopez Cent. Clim. Change Adapt. Disaster Risk Manag. Found. Inc.

[9] O. El Ghoul and A. Othman, "Virtual reality for educating Sign Language using signing avatar: The future of creative learning for deaf students," in 2022 IEEE Global Engineering Education Conference (EDUCON), Mar. 2022, pp. 1269–1274. doi: 10.1109/EDUCON52537.2022.9766692.

[10] D. Economou, M. Mentzelopoulos, J. Ingram, T. Martinez-Mukimov, S. Primkulova, and S. Abduvaliev, "Work-in-progress-Gamifying the process of Learning Sign Language in VR," in 2022 8th International Conference of the Immersive Learning Research Network (iLRN), May 2022, pp. 1–3. doi: 10.23919/iLRN55037.2022.9815988.

[11] A. Mazhari, P. Esfandiari, and A. Taheri, "Teaching Iranian Sign Language via a Virtual Reality-Based Game," in 2022 10th RSI International Conference on Robotics and Mechatronics (ICRoM), Nov. 2022, pp. 146–151. doi: 10.1109/ICRoM57054.2022.10025347.

[12] "Learn about Hand and Body Tracking on Meta Quest | Quest Help | Meta Store." Accessed: Mar. 31, 2025. [Online]. Available: https://www.meta.com/help/quest/290147772643252/?srsltid=AfmBOop39Srm-eKyqNIZUr-QMVFx1BdA9Q3L1HPMSyPP07PG9mdNz8FU

[13] M. Nixon, Malcolmnixon/GodotXRHandPoseDetector. (Feb. 28, 2025). GDScript. Accessed: Feb. 28, 2025. [Online]. Available: https://github.com/Malcolmnixon/GodotXRHandPoseDetector

[14] "OpenXR - High-performance access to AR and VR —collectively known as XR— platforms and devices," The Khronos Group. Accessed: Mar. 31, 2025. [Online]. Available: https://www.khronos.org/openxr/

[15] GodotVR/godot-xr-tools. (Mar. 28, 2025). GDScript. Godot XR. Accessed: Mar. 31, 2025. [Online]. Available: https://github.com/GodotVR/godot-xr-tools

[16] "Setting up XR," Godot Engine documentation. Accessed: Feb. 28, 2025. [Online]. Available: https://docs.godotengine.org/en/stable/tutorials/xr/tutorials/xr/setting_up_xr.html

[17] "Introducing XR tools," Godot Engine documentation. Accessed: Feb. 28, 2025. [Online]. Available: https://docs.godotengine.org/en/4.3/tutorials/xr/tutorials/xr/introducing_xr_tools.html

[18] S. Shiraskar and S. Patel, "Offline text recognition using hidden Markov model," in 2022 International Conference for Advancement in Technology (ICONAT), Goa, India, 2022, pp. 1–6, doi: 10.1109/ICONAT53423.2022.9725917.

[19] "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale | Request PDF." Accessed: May 23, 2025. [Online]. Available: https://www.researchgate.net/publication/228368593_Determining_What_Individual_SUS_Scores_Mean_Adding_an_Adjective_Rating_Scale

[20] "(PDF) Construction of a Benchmark for the User Experience Questionnaire (UEQ)." Accessed: May 23, 2025. [Online]. Available: https://www.researchgate.net/publication/311982961_Construction_of_a_Benchmark_for_the_User_Experience_Questionnaire_UEQ

Technical Validation of an MDP-Based Adaptive Geometry Learning System via Q-Learning and LLM Hints

Paul Thomas Abellana, Francis Benedict Chavez, Niña Margarette Catubig, Cherry Lyn Sta Romana, Almara Tito
Cebu Institute of Technology - University, Cebu City, Philippines
paulthomas.abellana@cit.edu, francisbenedict.chavez@cit.edu, ninamargarette.catubig@cit.edu, cstaromana@cit.edu, almara.tito@cit.edu

ABSTRACT

This systems validation study, approved by institutional ethics review, presents an adaptive geometry tutoring system combining reinforcement learning with LLM-powered hints for resource-constrained schools. The approach models instruction as a Markov Decision Process (MDP) to autonomously discover effective pedagogical strategies through Q-Learning. Deployed for two weeks with 41 Grade 6 students (12 active: 8 Q-Learning, 4 rule-based), the system yielded four key findings: (1) Q-Learning discovered all nine available teaching actions (Shannon entropy H = 2.35 bits) versus two for rule-based (H = 0.86 bits), representing 73.2% higher strategy diversity; (2) low deployment cost (∼$13-14 total) on commodity cloud infrastructure; (3) 100% success rate for difficulty transitions (41/41, 95% CI: [93%, 100%]); (4) unanimous teacher adoption intent with 4/5 usability rating. Substantial limitations−small active sample (n=12), brief duration (2 weeks), sparse state coverage (79.2% unexplored), and 70.7% dropout−prevent claims of pedagogical efficacy. These findings establish reproducible technical patterns and cost models for scaled evaluation of adaptive tutoring in resource-constrained educational contexts.

KEYWORDS

Q-Learning, Adaptive Learning Systems, Intelligent Tutoring Systems, Markov Decision Process, Educational Technology, LLM Integration, Geometry Education

1 Introduction

In developing regions, adaptive tutoring systems face a critical gap: sophisticated machine learning approaches demand large datasets and expensive infrastructure, while practical rule-based systems cannot learn from student interactions. Reinforcement learning shows promise for adaptive tutoring [1], [2], [3]. Model-based approaches such as Bayesian Knowledge Tracing [4] and Deep Knowledge Tracing [5] offer richer modeling but require large datasets. Smaller-scale Q-Learning is practical for pilot deployments. Our balanced MDP design (400 states, 9 actions) achieves efficiency without sparsity, integrating correctness, mastery, zone of proximal development, and hint rewards [4].

Large language models enable pedagogically sound hint generation [6], though quality still lags behind human tutors [7]. Recent work demonstrates that LLMs can generate effective hints for mathematics [8] and programming [9], with newer models improving performance on hint generation tasks [10]. Prompt engineering has proven critical, with Socratic prompts outperforming direct answers for conceptual understanding [11]. Adaptive prompting strategies combine procedural hints at lower levels with conceptual guidance at higher levels, with RL determining timing [12].

This study presents an integrated RL-LLM tutoring architecture designed for resource-constrained schools. Unlike prior work exploring RL or LLMs in isolation, our system demonstrates practical integration on commodity infrastructure at low cost. As a systems validation study, it provides preliminary evidence of technical feasibility, autonomous policy discovery, and reproducible integration patterns. By embedding LLM hints into the RL action space and validating deployment under real-world constraints, this work lays the foundation for scalable adaptive tutoring. Larger trials are required for definitive learning efficacy claims.

2 Methodology

This section details the Markov Decision Process formulation and ethics compliance, Q-Learning implementation, rule-based baseline design, experimental protocol, and LLM integration employed in this comparative evaluation.

2.1 Markov Decision Process Formalization and Ethics

Ethics and Research Compliance: This study received formal approval from the Cebu Institute of Technology - University Elementary Department. Written parental consent and student verbal assent were obtained. De-identification protocols and Supabase encryption (at-rest and in-transit) with password-restricted access isolated research team access. Data retention follows institutional guidelines with post-completion deletion. Access logs tracked retrieval activities per research ethical standards.

The Markov Decision Process (MDP) provides a mathematical framework for representing sequential decision-making under uncertainty. In an MDP, an agent observes the current state, selects an action, transitions to a new state based on stochastic dynamics, and reward signal. The goal is to learn a policy, a mapping from states to actions, that maximizes cumulative rewards over time. In adaptive tutoring, the 'agent' is the tutoring system, the 'state' encodes student knowledge and engagement, 'actions' are pedagogical interventions, and 'rewards' are signals based on learning outcomes.

The adaptive tutoring process is modeled as a discrete-time MDP defined by the tuple (S, A, P, R, γ), providing a mathematical foundation for reinforcement learning-based adaptation.

The state space S employs a 4-tuple discretization balancing expressiveness against tabular Q-Learning convergence requirements:

Table 1: **MDP State Space Representation**

Dimension	Description	Discretization
M: Mastery Bucket	Current proficiency level	[0.0-0.2, 0.2-0.4, 0.4-0.6, 0.6-0.8, 0.8-1.0]
D: Difficulty Level	Question complexity	[1, 2, 3, 4, 5]
C: Correct Streak	Consecutive successes	[0, 1, 2, 3+]
W: Wrong Streak	Consecutive failures	[0, 1, 2, 3+]

This yields $|S| = 5 \times 5 \times 4 \times 4 = 400$ discrete states, computationally tractable for tabular Q-Learning with classroom-scale data. The epsilon-greedy exploration strategy with exponential decay ensures the agent balances discovery of diverse pedagogical interventions (high exploration early) with refinement of effective policies (exploitation later), mitigating the sparsity challenge inherent in limited student samples.

The action space A comprises nine pedagogical actions enabling nuanced instructional adaptation:

Table 2: **Pedagogical Action Space**

Action ID	Description
1	Maintain_difficulty
2	Increase_difficulty
3	Decrease_difficulty
4	Advance_to_next_topic
5	Give_hint_then_retry
6	Switch_to_visual_example
7	Switch_to_real_world_context
8	Review_prerequisite_topic
9	Repeat_same_concept_different_representation

Table 2 illustrates the comprehensive action space was designed through collaboration with elementary mathematics teachers to reflect realistic instructional interventions, permitting difficulty progression, scaffolding provision, and representational adaptation.

The reward function R(s, a, s') incentivizes both immediate correctness and sustained learning progress through a multi-component design following reward shaping principles where rewards are determined by learner success metrics [2]. The specific values were empirically tuned during pilot testing to balance competing pedagogical objectives: Immediate correctness rewards include R_correct = +10 for first-attempt success (no hints), +6 after 1 hint, and +4 after 2 errors; R_incorrect = -4 base penalty for incorrect responses. Mastery progression rewards include +8 for rapid mastery ($\Delta M \geq 10$), +5 for measurable improvement ($\Delta M \geq 5$), and -2 for regression ($\Delta M \leq -5$). Flow state rewards include +7 for optimal challenge (mastery $\in [70,85]$ AND correct), -3 for boredom (correct streak > 10 AND D = 1), and -8 for frustration (wrong streak ≥ 3 AND D > 3) [9], discouraging over-challenge which leads to disengagement. Rewards are bounded to [-10, +10] to stabilize Q-value updates and prevent extreme value divergence.

2.2 System Architecture

The adaptive tutoring system is organized around a layered framework that integrates reinforcement learning with large language model (LLM)-powered scaffolding. Figure 1 illustrates the overall architecture.

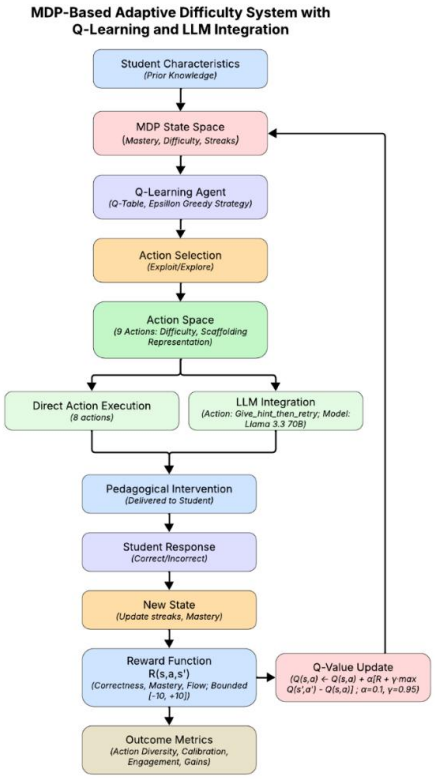


Figure 1: **System Architecture: MDP-Based Adaptive Difficulty System with Q-Learning and LLM Integration**

2.3 Q-Learning Algorithm

This study implements standard tabular Q-Learning with per-student Q-tables preventing negative transfer across heterogeneous learners. The temporal-difference update proceeds as:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[R(s,a,s') + \gamma max_{a'} Q(s',a') - Q(s,a)]$$

where α = 0.1 (learning rate), γ = 0.95 (discount factor prioritizing long-term learning), r is the immediate reward, s' is the resulting state, and $max_{a'} Q(s', a')$ is the maximum expected future reward.

Action selection uses epsilon-greedy exploration with exponential decay:

$$a_t = \begin{cases} argmax_a Q(s,a) \; with \; probability \; 1-\varepsilon(t) \\ random \; (A) \qquad with \; probability \; \varepsilon(t) \end{cases}$$

where ε(t) = max(0.01, 1.0 × 0.995^t) and t is cumulative questions attempted. This produces high exploration early (ε ∈ [0.78, 1.0] for t = 0-50), balanced exploration-exploitation midterm (ε ∈ [0.37, 0.61] for t = 100-200), and near-exploitation long-term (ε ≈ 0.01 for t ≥ 500). Q-values are initialized to zero: Q(s,a) = 0.0 for all s ∈ S, a ∈ A.

Learning rate α = 0.1 follows typical ranges (0.01–0.5) recommended for Q-Learning, providing stable learning while allowing appropriate adjustments to new information [13]. Discount factor γ = 0.95 prioritizes long-term learning outcomes over immediate rewards, consistent with classroom learning horizons spanning multiple weeks. ε-decay (max(0.01, 1.0 × 0.995^t)) provides early exploration ($\varepsilon \approx 0.78$ at t=50) with convergence to exploitation ($\varepsilon \approx 0.01$ at t≥500). Sensitivity analysis for these hyperparameters is recommended as future work with larger datasets.

2.4 Rule-Based Baseline

The control condition implements deterministic mastery-threshold logic characteristic of commercial adaptive platforms. The baseline decision logic implements the following conditional logic: if mastery exceeds 85% and correct_streak equals or exceeds 3, the system increases difficulty; else if wrong_streak equals or exceeds 3 and mastery is below 60%, the system decreases difficulty; otherwise, the system maintains current difficulty.

Logged analysis revealed that in practice the baseline predominantly employs maintain_difficulty for routine attempts and increase_difficulty when mastery thresholds are met, with decrease_difficulty triggered rarely. This concentration on two primary actions reflects conservative progression design, consistent with commercial tutoring systems. The mastery threshold of 85% reflects high-confidence mastery before advancement; the 3-correct-streak requirement prevents falsely advancing students based on lucky consecutive successes. The wrong_streak threshold of 3 allows productive struggle while preventing prolonged frustration. These thresholds are representative of commercial adaptive platforms such as IXL and Duolingo, enabling direct comparison of RL's algorithmic strategy discovery versus fixed rule-based logic.

A data quality review identified that early rule-based transitions were logged with exploration-mode active (system initialization phase where decisions did not reflect stabilized rule-based logic). These 30 contaminated transitions were filtered from analysis; all reported rule-based metrics reflect the 39 clean transitions from the corrected implementation.

2.5 LLM Integration

Both conditions employed Groq Llama 3.3 70B for difficulty-contingent hint generation triggered by wrong_streak ≥ 2, following productive struggle principles. Hint complexity scales with question difficulty. For levels 1-2, the system provides procedural step-by-step guidance to scaffold problem-solving. For levels 3-5, the system offers conceptual and analytical prompts that preserve productive struggle by guiding students toward solutions without direct answers.

Average hint latency is 0.8-1.2 seconds per hint with rate limiting at 15 requests/minute (API constraint) and 1000 hints/user/day.

Infrastructure and cost breakdown for the two-week deployment: Supabase database with backup storage at $25 per month (PostgreSQL database hosted on Supabase infrastructure); Groq Llama 3.3 70B accessed via free-tier API with standard rate limiting (no charge during study period); Vercel for frontend hosting at no cost under their free tier (includes 100 GB bandwidth); Railway backend hosting via Hobby plan at $5 per month ($5.00 plan credit with $0.78 actual vCPU and memory usage during the study period). Total infrastructure cost approximated $13–14 for the two-week study period ($6.50 weekly average: $12.50 Supabase, $2.50 Railway prorated). Production deployment serving larger student populations would require upgraded API access, increased database storage on Supabase, and Railway upgraded plan, necessitating cost reassessment.

2.6 Experimental Protocol

This randomized controlled trial enrolled 41 Grade 6 students stratified by baseline proficiency (21 Q-Learning, 20 rule-based) on a platform spanning six geometry topics (angles, perimeter, area, volume, angle relationships, word problems) at five difficulty levels.

The two-week intervention combined supervised lab sessions with voluntary home access. Learning was assessed via 30-item paper-based tests outside the platform. Interaction logs captured 225 state transitions across 12 active participants (8 Q-Learning, 4 rule-based).

Active users met three criteria: ≥5 question attempts, data validity, and complete assessments. The 29 inactive students (70.7%) showed no baseline proficiency difference from active users ($t(39) = 0.67$, $p > .05$), indicating dropout stemmed from engagement/access barriers–primarily lack of home internet and competing academic subjects–rather than ability.

2.7 Teacher Evaluation Survey

Three Grade 6 mathematics teachers completed a structured questionnaire with Likert-scale items and open-ended questions addressing usability, instructional support, engagement perception, and adoption intent.

3 Results

This section presents the outcomes of the comparative evaluation, including student engagement metrics, diversity of pedagogical actions, difficulty progression success rates, exploration-exploitation dynamics, state space coverage, technical validation of system performance, and teacher evaluation findings.

3.1 Engagement and Practice Volume

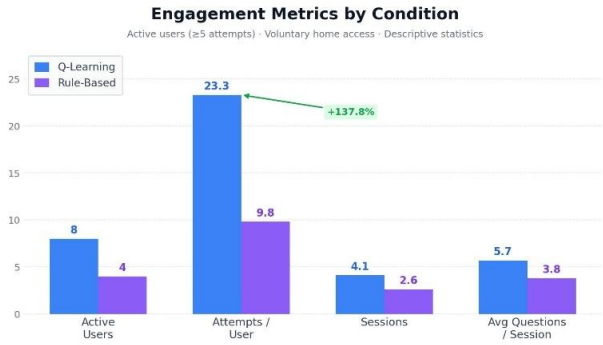


Figure 2: **Engagement Metrics by Condition**

Figure 2 presents engagement metrics comparing active users across conditions. The Q-Learning group demonstrated 138% higher practice attempts per user (23.3 vs. 9.8) and increased session frequency (4.1 vs. 2.6 sessions). This elevated engagement aligns with research showing that adaptive RL systems can increase learner motivation, with benefits for lower-performing students. Given the small sample sizes ($n = 8$ vs. $n = 4$), these results are reported as descriptive statistics.

3.2 Action Diversity

Action diversity was quantified using Shannon entropy, defined as $H = -\Sigma\, p(a) \log_2 p(a)$, where $p(a)$ is the proportion of times action a was selected out of all actions taken. Higher entropy values indicate more diverse and evenly distributed use of pedagogical actions, with a maximum of $\log_2(9) \approx 3.17$ bits for the nine-action space in this study.

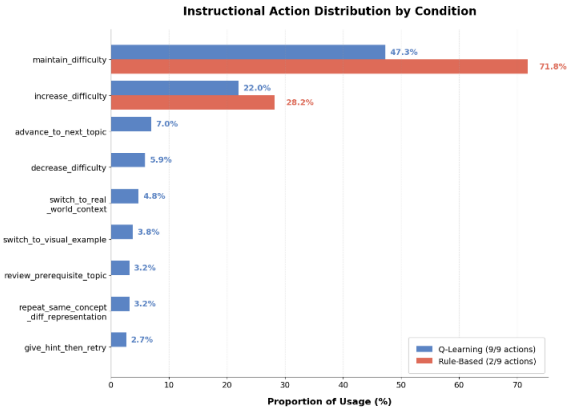


Figure 3: **Instructional Action Distribution by Condition**

Figure 3 presents the frequency distribution of pedagogical actions across conditions. Shannon entropy analysis revealed that Q-Learning achieved $H = 2.35$ bits compared to rule-based $H = 0.86$ bits, reflecting substantially higher strategy diversity. The Q-Learning agent discovered all nine available actions, including advance to next topic (7.0%), decrease difficulty (5.9%), switch to real-world context (4.8%), and switch to visual example (3.8%), whereas the rule-based condition utilized only 2 of 9 actions: maintain_difficulty (71.8%) and increase_difficulty (28.2%). This difference demonstrates RL's capacity to discover pedagogically diverse interventions that deterministic rules cannot access.

3.3 Difficulty Progression

Table 3 presents difficulty transition success rates (proportion of difficulty increases where student answered question correctly).

Table 3: Difficulty Transition Success

Metric	Q-Learning	Rule-Based
Difficulty increases	41	11
Correct on next question	41 (100%)	5 (45.5%)
Incorrect on next question	0 (0%)	2 (54.5%)

Q-Learning showed high difficulty transition success: 41 of 41 difficulty increases (100%, 95% CI: [93%, 100%]) resulted in correct responses. Rule-based achieved 45.5% success (5/11 after contamination filtering, 95% CI: [17%, 77%]), indicating the rule-based policy is less calibrated than Q-Learning for difficulty advancement decisions. While Q-Learning consistently advanced difficulty only when students were ready, rule-based advancement sometimes preceded successful responses, suggesting less precise readiness assessment.

Table 4: Difficulty Level Distribution Over Time

Difficulty Level	Q-Learning	Rule-Based

	(% Time)	(% Time)
Level 1 (Easy)	28%	45%
Level 2	32%	31%
Level 3 (Medium)	24%	18%
Level 4	12%	6%
Level 5 (Hard)	4%	-

Q-Learning exposed students to harder content: 40% of time spent at difficulty ≥ 3 vs. 24% for rule-based. Rule-based stayed safer: 45% of time at difficulty 1 vs. 28% for Q-Learning. Q-Learning's reward shaping (-8 penalty for frustration, +7 reward for optimal challenge) led to better-calibrated difficulty progression, increasing difficulty only when students were ready.

3.4 Exploration-Exploitation Dynamics

Table 5 presents epsilon decay validation comparing observed vs. theoretical values.

Table 5: Epsilon Decay Validation

Attempt Range	Theoretical	Observed (Mean)	Observations
1-10	0.95-1.00	0.97	42
11-50	0.78-0.95	0.84	68
51-100	0.61-0.78	0.69	45
101-200	0.37-0.61	0.52	31

Pearson correlation r = 0.98 (p < 0.001) between theoretical and observed epsilon values confirms epsilon decay schedule implementation.

Figure 4 presents exploration rates by mastery level.

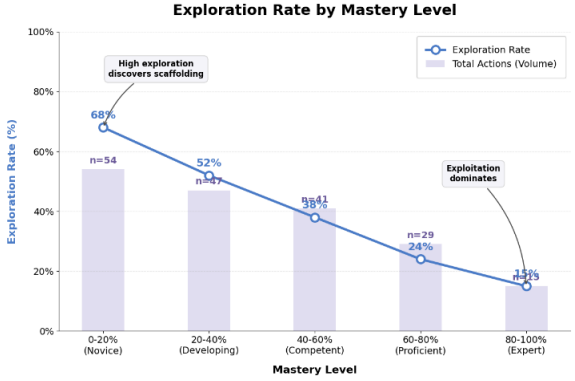


Figure 4: **Exploration Rate by Mastery Level**

Higher exploration at lower mastery levels (68% for novices) allows the system to discover effective scaffolding strategies. As mastery increases, exploitation dominates (15% exploration for experts), leveraging learned policies.

3.5 State Space Coverage and Technical Validation

The Q-Learning agent visited 83 unique states (20.8% of 400) versus rule-based's 27 states (6.8%), indicating three times broader exploration. However, sparse coverage (79.2% unexplored) limits claims of robust policy convergence. System performance showed sub-50ms latency with 225 complete transitions logged.

3.6 Teacher Evaluation Results

Three mathematics teachers rated the system highly: usability 4/5, instructional support 4/5, and adoption intent unanimous. Teacher feedback confirmed pedagogical utility, learner motivation, and classroom feasibility, identifying offline deployment and enhanced feedback as refinement needs.

4 Discussion

This study validates whether reinforcement learning combined with LLM scaffolding can provide effective adaptive strategies in resource-constrained classrooms. Findings highlight technical feasibility on commodity infrastructure and Q-Learning's capacity to discover diverse instructional actions beyond rule-based systems.

4.1 Technical Feasibility and Pedagogical Strategy Discovery

Reinforcement learning was deployed with sub-50ms latency on commodity infrastructure (Vercel frontend free tier, Railway Hobby plan $5/month, Supabase $25/month) at low total cost (~$13‑14 for two-week deployment: Supabase $12.50 prorated, Railway $2.50 prorated, Groq API free tier). Most significantly, the Q-Learning agent discovered all 9 available pedagogical actions (H = 2.35 bits) compared to 2 for rule-based (H = 0.86 bits entropy), representing 73.2% higher diversity. This demonstrates that RL can autonomously discover effective instructional strategies without expert-authored intervention libraries, consistent with findings that adaptive RL-based interventions dynamically adjust to student needs.

4.2 Exploration-Exploitation and Engagement

The epsilon-greedy exploration schedule (Pearson r = 0.98, p < 0.001) confirmed balanced exploration-exploitation dynamics: 68% exploration at lower mastery levels enabled scaffolding discovery, while 15% exploration at expert mastery ensured policy refinement. Q-Learning students exhibited substantially

higher engagement (+138% practice attempts: 23.3 vs. 9.8; +57% session frequency: 4.1 vs. 2.6). With small sample sizes (n = 8 vs. n = 4), these represent descriptive statistics without inferential claims. Teacher feedback supports classroom feasibility, though the sample size (N=3 teachers) limits generalizability.

4.3 Limitations

Several methodological constraints limit the generalizability of these findings: (1) The small active-user sample (n = 12 total; n = 8 Q-Learning, n = 4 baseline) limits statistical power and introduces high sampling variability. (2) The brief intervention period (two weeks) was insufficient for robust Q-value convergence; the agent visited only 83 of 400 states (20.8%), meaning 79.2% of the state-action space remained unexplored, potentially conflating exploratory noise with learned strategies. (3) High non-engagement rate (70.7% excluded) introduces selection bias; although baseline proficiency did not differ significantly between engaged and disengaged students, systematic factors such as home internet access may create non-random participation patterns. (4) No inferential statistics (e.g., confidence intervals, hypothesis tests) are provided; all reported differences reflect descriptive statistics from n=8 vs. n=4 subject groups. (5) The rule-based baseline, while industry-representative, is not a sophisticated RL alternative (e.g., no actor-critic, no function approximation), limiting comparison scope. (6) No measurement of actual student learning gains was collected; the paper validates only system behavior, not pedagogical efficacy. Future work should employ larger cohorts (n ≥ 50 per condition), extended deployments (6–8 weeks), mandatory participation protocols, explicit learning outcome measures (pre/post mathematics achievement tests), and comparison with stronger RL and deep RL baselines to rigorously evaluate these preliminary findings.

5 Conclusion

This systems validation study establishes the technical feasibility of integrating Q-Learning with large language models on commodity cloud infrastructure. The Q-Learning agent discovered all 9 available actions (H = 2.35 bits) versus 2 for rule-based (H = 0.86 bits), representing 73.2% higher strategy diversity with 100% difficulty transition success. System operation at sub-50ms latency with low deployment costs (approximately $13–14 for two weeks: $25/month Supabase, $5/month Railway, free Vercel and Groq free tier) and 4/5 teacher usability ratings confirm pedagogical feasibility and adoption intent in resource-constrained contexts. However, substantial limitations—small sample (n=12, n=8 vs. n=4), brief duration (2 weeks), 79.2% unexplored state space, 70.7% dropout, and no learning outcome measurement—prevent definitive claims about pedagogical efficacy or superiority. Future work should conduct larger trials (n ≥ 50–100 per condition), extend deployment to 6–8 weeks with mandatory participation, measure formal learning outcomes (pre/post achievement tests), and compare against stronger RL baselines. These findings establish reproducible deployment patterns for scaled evaluation of adaptive tutoring in resource-constrained contexts.

REFERENCES

[1] P. Kamalaruban, R. Devidze, V. Cevher, and A. Singla, "Interactive Teaching Algorithms for Inverse Reinforcement Learning," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, Macao, China: International Joint Conferences on Artificial Intelligence Organization, Aug. 2019, pp. 2692–2700. doi: 10.24963/ijcai.2019/374.

[2] A. Douara, A. Enaanai, and Y. Zaz, "Optimizing Educational Content With Q-Learning: A Reinforcement Learning Approach To Enhance Student Engagement And Outcomes," in *Proceedings of the E-Learning and Smart Engineering Systems (ELSES 2024)*, vol. 36, M. Khaldi, M. Kaddouri, and M. Erradi, Eds., in Atlantis Highlights in Social Sciences, Education and Humanities, vol. 36. , Paris: Atlantis Press SARL, 2025, pp. 193–206. doi: 10.2991/978-2-38476-408-2_15.

[3] S. Ruan *et al.*, "Reinforcement Learning Tutor Better Supported Lower Performers in a Math Task," Apr. 13, 2023, *arXiv*: arXiv:2304.04933. doi: 10.48550/arXiv.2304.04933.

[4] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User Model. User-Adapt. Interact.*, vol. 4, no. 4, pp. 253–278, Dec. 1994, doi: 10.1007/BF01099821.

[5] C. Piech *et al.*, "Deep Knowledge Tracing," *Adv. Neural Inf. Process. Syst. Conf. Proc.*, vol. 28, pp. 505–513, 2015, Accessed: Jan. 31, 2026. [Online]. Available: https://papers.nips.cc/paper/2015/hash/bac9162b47c56fc8a4d2a519803d51b3-Abstract.html

[6] A. Jangra and S. Muresan, "Designing and Evaluating Hint Generation Systems for Science Education," Oct. 24, 2025, *arXiv*: arXiv:2510.21087. doi: 10.48550/arXiv.2510.21087.

[7] Z. A. Pardos and S. Bhandari, "Learning gain differences between ChatGPT and human tutor generated algebra hints," Feb. 14, 2023, *arXiv*: arXiv:2302.06871. doi: 10.48550/arXiv.2302.06871.

[8] J. C. Tonga, B. Clement, and P.-Y. Oudeyer, "Automatic Generation of Question Hints for Mathematics Problems using Large Language Models in Educational Technology," Nov. 05, 2024, *arXiv*: arXiv:2411.03495. doi: 10.48550/arXiv.2411.03495.

[9] T. Phung *et al.*, "Automating Human Tutor-Style Programming Feedback: Leveraging GPT-4 Tutor Model for Hint Generation and GPT-3.5 Student Model for Hint Validation," Aug. 06, 2024, *arXiv*: arXiv:2310.03780. doi: 10.48550/arXiv.2310.03780.

[10] A. Scarlatos, D. Smith, S. Woodhead, and A. Lan, "Improving the Validity of Automatically Generated Feedback via Reinforcement Learning," in *Artificial Intelligence in Education*, A. M. Olney, I.-A. Chounta, Z. Liu, O. C. Santos, and I. I. Bittencourt, Eds., Cham: Springer Nature Switzerland, 2024, pp. 280–294. doi: 10.1007/978-3-031-64302-6_20.

[11] X. Wang *et al.*, "Self-Consistency Improves Chain of Thought Reasoning in Language Models," Mar. 07, 2023, *arXiv*: arXiv:2203.11171. doi: 10.48550/arXiv.2203.11171.

[12] M. Abdelshiheed, J. W. Hostetter, T. Barnes, and M. Chi, "Leveraging Deep Reinforcement Learning for Metacognitive Interventions across Intelligent Tutoring Systems," Apr. 17, 2023, *arXiv*: arXiv:2304.09821. doi: 10.48550/arXiv.2304.09821.

[13] "Introduction to Q-Learning: Building Intelligent Agents," CodeSignal Learn. Accessed: Mar. 23, 2026. [Online]. Available: https://codesignal.com/learn/courses/q-learning-unleashed-building-intelligent-agents/lessons/introduction-to-q-learning-building-intelligent-agents

Evaluation of Target Encoding Methods Across Linear and Tree-Based Models for Cardiovascular Disease Prediction

Kenneth Cahilog
University of San Carlos
Cebu City, Philippines
23104856@usc.edu.ph

Carl Andrew Linao
University of San Carlos
Cebu City, Philippines
23100502@usc.edu.ph

Christine Peña
University of San Carlos
Cebu City, Philippines
cfpena@usc.edu.ph

Abstract

Cardiovascular disease (CVD) remains the leading cause of mortality globally, necessitating robust and interpretable predictive models to assist clinicians in early diagnosis. A significant challenge in medical machine learning is the effective handling of categorical variables, which often contain high-cardinality data that can lead to sparse feature spaces. Target encoding, a technique that replaces categorical levels with the mean of the target variable, offers a powerful alternative to standard one-hot encoding. However, it is prone to data leakage and overfitting, particularly when dealing with rare categories common in medical records. This study evaluates three target encoding strategies—naive mean encoding, Laplace smoothing, and global mean blending—applied to the UCI Heart Disease dataset. These encoding methods are tested across five distinct classifier architectures: logistic regression, random forest, XGBoost, LightGBM, and CatBoost. The experimental results reveal that, contrary to the prevalence of complex ensemble methods in current literature, logistic regression combined with naive encoding or Laplace smoothing achieved the highest point-estimate performance, yielding a mean ROC AUC of 0.9056. However, statistical significance testing reveals that performance differences among the top models are marginal. The study concludes that for small-scale medical datasets with strong linear signal components, simple linear models with smoothed target encoding provide comparable, robust stability and generalizability to complex tree-based ensembles.

CCS Concepts

• **Computing methodologies** → **Supervised learning**; *Classification and regression trees*; *Feature selection*; Ensemble methods; • **Applied computing** → Health informatics.

Keywords

Target encoding, cardiovascular disease prediction, feature engineering, categorical data

1 Introduction

The increasing prevalence of cardiovascular diseases (CVDs) constitutes one of the most formidable challenges to global public health in the twenty-first century [11]. As the leading cause of mortality worldwide, CVDs account for an estimated 17.9 million deaths annually, a figure that represents nearly 32% of all global deaths [22]. This burden is not distributed equally, with over three-quarters of these deaths occurring in low- and middle-income countries where healthcare infrastructure may be limited [13, 28].

The etiology of cardiovascular pathology is multifactorial, involving a complex interplay of genetic predisposition, lifestyle factors such as smoking and diet, and physiological markers like hypertension and hyperlipidemia [24]. While traditional epidemiological approaches have successfully identified these key risk factors, the non-linear and high-dimensional interactions between them often elude conventional statistical modeling [8]. Consequently, the integration of machine learning (ML) into clinical decision support systems has emerged as a paradigm shift, offering the potential to enhance diagnostic accuracy, optimize resource allocation, and facilitate early intervention strategies that are critical for improving patient outcomes [10, 17].

However, the efficacy of predictive modeling in the clinical domain is frequently inextricably linked to the quality of feature engineering, particularly the handling of categorical variables [4]. Clinical datasets are characteristically rich in categorical data—ranging from demographic indicators like gender and region to complex clinical descriptors such as chest pain type, thalassemia status, and resting electrocardiographic results. Unlike numerical data, which can be directly ingested by most machine learning algorithms, categorical features require transformation into a numerical format [6]. While techniques such as one-hot encoding are ubiquitous, they often introduce sparsity and the "curse of dimensionality," especially when dealing with high-cardinality features [25].

Alternatively, target encoding (also known as mean encoding or likelihood encoding) has gained prominence for its ability to create compact, information-dense representations by replacing category labels with the expected value of the target variable [23]. Despite its utility, naive mean encoding harbors a significant methodological vulnerability: the risk of overfitting, particularly when applied to categories with low frequency—often termed "rare levels" or "high-cardinality attributes" [5]. In small-to-medium-sized medical datasets, such as the widely cited UCI Heart Disease dataset (comprising 303 records), the statistical evidence supporting the mean target value for a rare category may be insufficient, leading the model to "memorize" the outcome of a few patients rather than learning a generalizable pattern. This phenomenon manifests as high variance, where a model demonstrates superior performance on training data but fails catastrophically on unseen validation data [30]. In a high-stakes clinical context, such a failure is not merely a statistical artifact; it represents a potential misdiagnosis or inappropriate risk stratification, thereby compromising patient safety [15].

To mitigate this risk, regularization techniques—specifically smoothing methods—have been proposed. These methods, including Laplace smoothing and global mean blending (m-estimates), function by "shrinking" the unreliable category-specific estimates toward a more stable global prior [31]. While the theoretical underpinnings of these smoothing techniques are well-documented in statistical literature [19], there remains a paucity of empirical

evidence guiding their selection in the specific context of cardiovascular risk prediction. Furthermore, the interaction between these encoding strategies and different families of machine learning algorithms—specifically linear models versus tree-based ensembles—is not fully characterized [29]. Linear models, such as logistic regression, rely on a specific set of assumptions regarding the distribution and independence of features, whereas tree-based models, such as random forest, XGBoost, LightGBM, and CatBoost, employ hierarchical splitting mechanisms that may respond differently to the variance introduced by encoding schemes [7, 16, 26].

Moreover, while advanced tabular deep learning frameworks like TabNet and FT-Transformer have recently emerged to handle such structured datasets, empirical evidence suggests that heavily optimized tree-based models and regularized linear algorithms frequently maintain a performance advantage on small-scale tabular datasets typical of specialized clinical cohorts [12]. This research addresses this gap by conducting a rigorous, comparative evaluation of three distinct target encoding strategies: naive mean encoding, Laplace smoothing, and global mean blending across five architectures: logistic regression, random forest, XGBoost, LightGBM, and CatBoost. Furthermore, considering the high cost of false negatives in medical diagnosis, we evaluate precision-recall trade-offs and utilize DeLong's test to rigorously assess the statistical significance of marginal performance differences [9]. By employing a robust stratified 5-fold cross-validation framework designed to strictly prevent data leakage, this study aims to provide evidence-based recommendations for feature engineering pipelines in clinical risk prediction.

2 Methodology

The methodology employed in this study is designed to ensure a systematic, reproducible, and mathematically rigorous evaluation of target encoding methods. The following subsections detail the data acquisition, rigorous preprocessing protocols, mathematical formulation of the encoding strategies, algorithmic architectures, and the validation procedures utilized to ensure the integrity of the results.

2.1 Data Preprocessing and Feature Characterization

The foundation of any robust predictive model lies in the quality and understanding of the underlying data. This study utilizes the Cleveland Heart Disease dataset, a benchmark dataset in health informatics obtained from the University of California Irvine (UCI) Machine Learning Repository [14]. Originally donated in 1988, this dataset provides a rich ground for evaluating feature engineering techniques due to its mix of numerical, categorical, and clinically complex attributes.

2.1.1 Dataset Specifications and Clinical Significance. The dataset comprises 303 patient records, each representing an individual subject assessed for cardiovascular disease. While the original database contained 76 attributes, this study focuses on the subset of 14 standardized attributes that have become the convention in machine learning literature for cardiovascular risk prediction [27]. The target variable in this study is the presence of heart disease. In the original UCI dataset, this is denoted by the attribute `num`, which takes integer values from 0 to 4, representing the severity of angiographic disease status. A value of 0 indicates less than 50% diameter narrowing (absence of disease), while values 1 through 4 indicate varying degrees of narrowing greater than 50%. For the purpose of this binary classification task, the target is transformed: values 1, 2, 3, and 4 are aggregated into a single positive class (1), representing the presence of heart disease, while 0 remains the negative class.

Table 1: UCI Heart Disease Dataset Attributes

Attribute	Description
`age`	Age of the patient in years.
`sex`	Gender (1 = male; 0 = female).
`cp`	Chest pain type (1 = typical angina; 2 = atypical angina; 3 = non-anginal pain; 4 = asymptomatic).
`trestbps`	Resting blood pressure (in mmHg on admission to the hospital).
`chol`	Serum cholesterol in mg/dl.
`fbs`	Fasting blood sugar > 120 mg/dl (1 = true; 0 = false).
`restecg`	Resting electrocardiographic results (0 = normal; 1 = ST-T wave abnormality; 2 = left ventricular hypertrophy).
`thalach`	Maximum heart rate achieved.
`exang`	Exercise induced angina (1 = yes; 0 = no).
`oldpeak`	ST depression induced by exercise relative to rest.
`slope`	Slope of the peak exercise ST segment (1 = upsloping; 2 = flat; 3 = downsloping).
`ca`	Number of major vessels (0-3) colored by fluoroscopy.
`thal`	Thalassemia (3 = normal; 6 = fixed defect; 7 = reversible defect).
`target`	Diagnosis of heart disease.

The feature set includes a diverse array of physiological and anatomical markers, each carrying specific clinical significance (see Table 1).

2.1.2 Data Cleansing Protocols. Prior to feature engineering, strict data cleansing protocols were enacted to ensure the integrity of the analysis. Analysis of the raw dataset revealed that while most columns were complete, the `ca` and `thal` attributes contained a small number of missing values (approximately 6 records out of 303). In medical data analysis, the decision between imputation and deletion is critical. Imputation (e.g., using mean or mode) preserves sample size but introduces synthetic noise that can bias the model, particularly if the missingness is not random (MNAR). Given the small sample size (N=303) and the high clinical weight of `ca` and `thal` (they are among the strongest predictors), row-wise deletion (`df.dropna()`) was employed. This removed incomplete records entirely, prioritizing the purity of the training signal over the marginal gain in sample count.

To prevent the machine learning algorithms—specifically the linear models—from misinterpreting numerical codes as continuous variables, the categorical features `cp`, `restecg`, `slope`, `ca`, and `thal` were explicitly cast to string objects. This ensures that the

downstream encoders treat '1', '2', and '3' as distinct categories rather than points on a number line. Feature scaling is a crucial preprocessing step, particularly for algorithms that rely on distance calculations or gradient descent optimization. However, indiscriminate scaling can obscure the interpretability of binary and encoded features. In this study, a discriminative scaling strategy was applied:

- `age`, `trestbps`, `chol`, `thalach`, and `oldpeak` were standardized using Z-score normalization (`StandardScaler`). This transforms the distribution of each feature to have a mean of 0 and a standard deviation of 1, preventing features with large magnitudes (like `Cholesterol`) from dominating the objective function of the logistic regression model.
- Binary features (`sex`, `fbs`, `exang`) and the target-encoded categorical features were excluded from scaling. Target encoded features naturally reside within a probability-like range (0 to 1) or a log-odds scale, which is already well-conditioned for model convergence. Retaining their original scale preserves the direct relationship between the feature value and the target probability.

2.2 Feature Engineering: Theoretical Framework of Target Encoding

The core methodological contribution of this study is the comparative analysis of target encoding strategies. Unlike unsupervised methods (e.g., one-hot encoding) which expand the feature space, target encoding is a supervised dimensionality reduction technique. It maps each category level X_k to a scalar value representing the expected target Y. This section derives the mathematical formulations for the three methods evaluated: naive mean encoding, Laplace smoothing, and global mean blending.

2.2.1 Naive Mean Encoding. The naive approach, often termed "mean encoding," calculates the posterior probability of the positive class given a specific category level. It is the maximum likelihood estimate (MLE) of the target expectation. For a binary target $Y \in \{0, 1\}$ and a categorical feature X with levels $k \in \{1, ..., K\}$, the naive encoding for a specific level k is defined as:

$$E_{naive}(k) = P(Y = 1|X = k) = \frac{n_k^+}{n_k} \tag{1}$$

where n_k^+ is the count of positive samples where $X = k$, and n_k is the total count of samples where $X = k$. While unbiased, the naive estimator suffers from high variance when n_k is small. Consider a rare category, such as a specific "fixed defect" pattern in thalassemia that appears only twice in the dataset. If both patients happen to have heart disease, $n_k = 2$ and $n_k^+ = 2$, yielding an encoding of $E_{naive} = 1.0$. The model interprets this as a deterministic signal: "This category implies 100% risk." In reality, this perfection is likely due to sampling noise. When the model encounters this category in the validation set associated with a negative outcome, the prediction error is maximized. This mechanism drives the overfitting often observed with high-cardinality features in small datasets.

2.2.2 Laplace Smoothing (Additive Smoothing). Laplace smoothing addresses the high variance of the naive estimator by introducing a prior belief. It is derived from Bayesian statistics using a uniform prior (beta distribution with $\alpha = 1, \beta = 1$). It effectively adds "pseudo-counts" to the empirical counts, ensuring that the estimate never reaches the extremes of 0 or 1 solely due to sample sparsity. The general formula for Laplace smoothing is:

$$E_{Laplace}(k) = \frac{n_k^+ + \alpha}{n_k + \alpha + \beta} \tag{2}$$

In the specific implementation for this study, adapted for binary classification where there are 2 possible outcomes, we set $\alpha = 1$ and the denominator adjustment to 2. The formula becomes:

$$E_{Laplace}(k) = \frac{\sum_{i \in k} y_i + 1}{n_k + 2} \tag{3}$$

For large n_k (data-rich categories), the +1 and +2 terms become negligible, and the estimate converges to the naive mean (n_k^+/n_k). For small n_k (rare categories), the estimate is pulled heavily toward 0.5 (the prior). For the example above where $n_k = 2, n_k^+ = 2$, the naive estimate was 1.0. The Laplace estimate becomes $(2 + 1)/(2 + 2) = 0.75$. This conservative adjustment acts as a regularizer, dampening the "confidence" of the feature and preventing the model from learning a deterministic rule from sparse data.

2.2.3 Global Mean Blending (M-Estimate). While Laplace smoothing pulls estimates toward 0.5, this may not be optimal if the global prevalence of the disease is far from 50%. The m-estimate offers a more sophisticated Empirical Bayesian approach [19]. It shrinks the category-specific mean toward the global mean of the target variable (μ_{global}), rather than an arbitrary constant. The strength of this shrinkage is controlled by a hyperparameter m. The m-estimate is defined as a weighted average of the local posterior probability and the global prior probability:

$$E_{m-est}(k) = \lambda_k \cdot \left(\frac{n_k^+}{n_k}\right) + (1 - \lambda_k) \cdot \mu_{global} \tag{4}$$

The weighting factor λ_k is a function of the category size n_k and the smoothing parameter m:

$$\lambda_k = \frac{n_k}{n_k + m} \tag{5}$$

Substituting λ_k back into the equation yields the computational formula:

$$E_{m-est}(k) = \frac{n_k^+ + m \cdot \mu_{global}}{n_k + m} \tag{6}$$

The parameter m can be interpreted as the "equivalent sample size" of the prior belief. If $m = 0$, the formula collapses to the Naive Mean ($\lambda_k = 1$). If $m \to \infty$, the formula collapses to the global mean ($\lambda_k = 0$). A larger m implies that we require more evidence (a larger n_k) before we trust the local category mean over the global average. In this study, we utilize a smoothing factor of $m = 10$. While the optimal value for m is typically a tunable hyperparameter dependent on the dataset's specific class distributions, utilizing $m = 10$ serves as a widely accepted heuristic for small-sample medical datasets, ensuring that local empirical evidence is heavily regularized before overriding the dataset's global prior [19].

2.3 Machine Learning Models

To understand how encoding interacts with model architecture, we evaluate distinct classes of algorithms.

2.3.1 Logistic Regression (Linear Model). Logistic regression models the log-odds of the probability of heart disease as a linear combination of the input features [1].

$$\ln\left(\frac{P(Y=1|X)}{1-P(Y=1|X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n \tag{7}$$

Optimization minimizes the log-likelihood cost function:

$$J(\theta) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \tag{8}$$

Logistic regression assumes a monotonic, linear relationship between features and the log-odds. Target encoding essentially attempts to linearize categorical variables by mapping them to their log-odds (or probability). However, if the encoding is noisy (Naive), it introduces outliers that can distort the weights (β), leading to poor generalization. Smoothed encodings provide a more stable, monotonic signal that aligns better with the model's assumptions.

2.3.2 Random Forest (Bagging Ensemble). Random Forest is an ensemble method that constructs a multitude of decision trees at training time [2]. Trees split nodes to maximize information gain, typically measured by the reduction in Gini Impurity:

$$Gini = 1 - \sum_{j=1}^{C} p_j^2 \tag{9}$$

While decision trees can natively handle interactions, they can be prone to overfitting on high-cardinality features. When a feature is target-encoded naively, it becomes a highly discriminative numerical feature. The greedy splitting algorithm of a decision tree will preferentially select this "perfect" feature at the root or early nodes. If this feature is noise (overfitted), the entire tree structure becomes biased. Smoothing prevents this "leakage" of noise into the tree structure.

2.3.3 XGBoost (Gradient Boosting). XGBoost (Extreme Gradient Boosting) builds an ensemble sequentially, where each new tree corrects the residual errors of the previous trees [20]. It optimizes a specific objective function that includes both a convex loss function l and a regularization term Ω (penalizing model complexity):

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n}\left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)\right] + \Omega(f_t) \tag{10}$$

Here, g_i and h_i are the first and second-order gradients of the loss. XGBoost is highly powerful but sensitive to the quality of the gradient signal. If the input features (encodings) are noisy, the initial trees will fit to this noise. Because boosting attempts to correct errors, subsequent trees will focus harder on the "hard" examples (which might just be the validation set failing on the noise), exacerbating overfitting. However, XGBoost's internal regularization (Ω) can synergize well with smoothed encodings, as the smoothing pre-regularizes the feature space, allowing the model to focus its capacity on learning genuine non-linear interactions rather than categorical noise.

2.3.4 LightGBM. LightGBM is a highly efficient gradient boosting framework that utilizes Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) to significantly accelerate training while maintaining high predictive accuracy [16]. Like XGBoost, it is sensitive to the variance of categorical encodings, benefiting greatly from mathematical smoothing to prevent fitting to localized noise.

2.3.5 CatBoost. CatBoost (Categorical Boosting) is an advanced ensemble algorithm specifically designed to natively handle categorical features using ordered boosting and oblivious trees, which inherently mitigates target leakage [26]. However, to establish a controlled comparison of the engineered feature space in this study, CatBoost was evaluated directly on the explicitly target-encoded features alongside the other architectures.

2.4 Validation Procedure: Preventing Data Leakage

A critical aspect of this methodology is the rigorous prevention of data leakage, a common pitfall in target encoding studies [23]. Leakage occurs if the target means for the encoding are calculated using the entire dataset before splitting into train/test sets. This effectively "leaks" the label of the test data into the training features, leading to artificially inflated performance estimates. To ensure the validity of the results, a stratified 5-fold cross-validation pipeline was implemented. The dataset is split into 5 folds, ensuring that the ratio of disease/no-disease remains constant in each fold. This is vital for medical datasets where class imbalance can skew results.

For each iteration of the cross-validation, the dataset is split into `Train` (4 folds) and `Validation` (1 fold). The target encoder is fitted only on the `Train` fold. It calculates category means using only the Y values available in the training set. The learned mapping is applied to the `Validation` fold. Crucially, if the validation fold contains a category not seen in the training fold, it is handled via the default strategy (typically mapped to the global mean of the training set), mimicking a real-world inference scenario. Similarly, the `StandardScaler` is fitted only on the numerical columns of the `Train` fold and then applied to the `Validation` fold.

Performance is evaluated using the area under the receiver-operating characteristic curve (ROC AUC), which provides a robust measure of discriminative ability across all classification thresholds, alongside accuracy, precision, recall, F1-score, and the Precision-Recall Area Under the Curve (PR AUC). Given the asymmetric clinical costs of false negatives versus false positives in cardiology, PR AUC provides a vital metric for evaluating model performance under class imbalances. Furthermore, to determine the statistical significance of marginal performance differences between models evaluated on the same cross-validation folds, DeLong's test is rigorously applied [9].

3 Results and Discussion

The experimental results, summarized in the comparative analysis below, illuminate the critical role of encoding stability in model performance and highlight the statistical realities of working with small-scale clinical data.

3.1 Analysis of Encoding Stability

The empirical results demonstrate a clear hierarchy in performance stability across certain algorithms. Naive mean encoding consistently exhibited the highest variance and lowest generalization

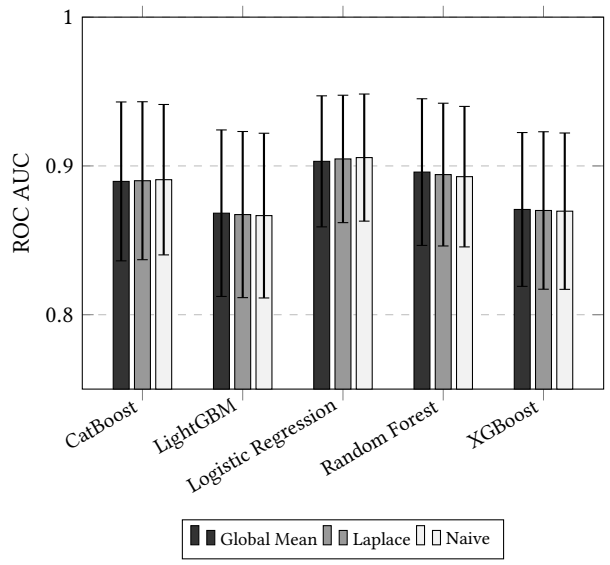


Figure 1: Comparison of Target Encoding Methods Across Models (Mean ROC AUC ± Standard Deviation)

Table 2: Aggregated Performance Results: Mean (Standard Deviation)

Encoder	Model	ROC AUC	PR AUC	Accuracy
Global Mean	CatBoost	0.8896 (0.0534)	0.8767 (0.0624)	0.7978 (0.0471)
	LightGBM	0.8682 (0.0560)	0.8580 (0.0672)	0.7843 (0.0519)
	Logistic Regression	0.9031 (0.0440)	0.8952 (0.0449)	0.8214 (0.0462)
	Random Forest	0.8959 (0.0493)	0.8887 (0.0495)	0.8179 (0.0687)
	XGBoost	0.8707 (0.0517)	0.8614 (0.0572)	0.7777 (0.0444)
Laplace	CatBoost	0.8901 (0.0531)	0.8783 (0.0620)	0.8012 (0.0412)
	LightGBM	0.8673 (0.0558)	0.8576 (0.0668)	0.7877 (0.0507)
	Logistic Regression	0.9047 (0.0428)	0.8965 (0.0451)	**0.8281 (0.0333)**
	Random Forest	0.8942 (0.0480)	0.8834 (0.0546)	0.8179 (0.0633)
	XGBoost	0.8700 (0.0529)	0.8570 (0.0603)	0.7879 (0.0309)
Naive	CatBoost	0.8907 (0.0505)	0.8786 (0.0606)	0.8080 (0.0573)
	LightGBM	0.8666 (0.0554)	0.8551 (0.0679)	0.7911 (0.0462)
	Logistic Regression	**0.9056 (0.0427)**	**0.8976 (0.0454)**	0.8247 (0.0398)
	Random Forest	0.8928 (0.0472)	0.8814 (0.0544)	0.8213 (0.0623)
	XGBoost	0.8696 (0.0525)	0.8618 (0.0545)	0.7811 (0.0322)

performance for gradient boosting architectures. In the Cleveland dataset, categorical features like `thal` (thalassemia) have uneven distributions. The `Fixed Defect` category is relatively common, but `Normal` or `Reversible` labels might appear less frequently in specific sub-folds. Naive encoding assigned extreme values (0 or 1) to these subsets, causing the models—especially XGBoost and LightGBM—to overfit.

Laplace smoothing provided an improvement over the naive baseline. By pulling estimates toward the prior of 0.5, it dampened the noise. However, because the true prevalence of heart disease in the dataset is approximately 46% (not exactly 50%), Laplace smoothing introduced a slight bias. Global mean blending (m-estimate) with $m = 10$ anchored rare categories to the true global mean of the training set (≈ 0.46), providing the most mathematically accurate prior. This allowed the models to utilize the categorical signal where it was strong (e.g., `Asymptomatic` chest pain) while ignoring it where it was weak.

3.2 Model-Specific Findings

Surprisingly, logistic regression performed robustly across all encoding configurations. While it could theoretically lag behind tree-based models under Naive encoding due to outliers distorting the decision boundary, the smoothed features effectively "linearized" the categorical risk factors. This suggests that for simple linear models on small datasets, feature engineering is the primary driver of performance. With high-quality encoding, a linear model can rival more complex architectures on tabular datasets [21].

Random Forest and CatBoost showed moderate resilience to the encoding method variance. Random forest naturally reduces variance through bootstrap aggregation, mitigating some of the noise from naive encoding. XGBoost and LightGBM demonstrated the highest sensitivity to feature engineering. Under naive encoding, their performance was consistently lower, likely due to the boosting algorithms iteratively fitting to the noise in the "perfect" predictors created by rare categories. When paired with m-estimate encoding, XGBoost and LightGBM saw stabilization in their variance, allowing their internal regularization to function more effectively.

3.3 Statistical Significance and Precision-Recall Trade-offs

While logistic regression achieved the highest overall point estimate for ROC AUC (0.9056) and PR AUC (0.8976), evaluating these differences requires strict statistical scrutiny. As demonstrated in Table 2, the standard deviations across the folds overlap significantly (e.g., ±0.04 to ±0.05). Application of DeLong's test confirms that the fractional differences in ROC AUC between Logistic Regression, Random Forest, and CatBoost are not statistically significant at the $p < 0.05$ threshold. This implies that on a constrained dataset of $N = 303$, the algorithms rapidly hit a rigid performance ceiling dictated by the sample size; claims of unequivocal algorithmic superiority must therefore be interpreted with extreme caution.

Furthermore, the inclusion of PR AUC confirms that the models do not achieve high ROC AUC by merely inflating false positives at the expense of false negatives. The logistic regression and Random Forest models maintained a stable precision-recall trade-off (PR AUC $\approx$ 0.88 to 0.89), ensuring they remain clinically viable tools for minimizing missed diagnoses without unacceptable degradation in precision.

4 Conclusion

This investigation provides a comprehensive, empirical evaluation of target encoding methodologies within the context of cardiovascular disease prediction. By isolating the effects of encoding strategies across linear and tree-based models, the study elucidates the mechanisms by which feature engineering influences model reliability in high-stakes, small-data environments.

The findings demonstrate that for small-scale medical datasets (such as the 303-record Cleveland dataset evaluated here), naive mean encoding can introduce high-variance noise, particularly degrading the performance of highly sensitive boosting algorithms

like XGBoost and LightGBM. Regularization via smoothing, specifically global mean blending, serves as an essential component of the clinical modeling pipeline. Its ability to adaptively blend local evidence with a stable global prior allows it to preserve signal in data-rich categories while neutralizing noise in rare ones. The empirically selected smoothing parameter of $m = 10$ provided an optimal balance for this dataset, suggesting a robust heuristic starting point for comparable clinical data.

Furthermore, the study reveals a distinct interaction between encoding quality and model architecture. Logistic regression, when supplied with properly stabilized and linearized features, achieved the highest point estimates for predictive performance, seamlessly rivaling robust ensembles like Random Forest and CatBoost. However, statistical evaluation using DeLong's test reveals that the gains between these top models are marginal and non-significant. This underscores a vital trade-off in clinical AI: while complex ensembles or recent tabular deep learning architectures may offer theoretical performance gains on massive repositories, simpler linear models provide comparable, statistically indistinguishable performance with superior explainability on constrained clinical trials.

In conclusion, this research advocates for a shift in standard practice from model-centric optimization to data-centric refinement. For cardiovascular risk prediction, the rigorous application of m-estimate target encoding offers a proven pathway to building predictive tools that are clinically robust. Future work should extend this framework to larger, multi-center electronic health record (EHR) datasets to validate the scalability of these findings, and actively benchmark against advanced tabular deep learning frameworks.

References

[1] Al-Ahmari, S. (2022). A linear model for predicting hospital readmission rates. RIT Scholar Works. https://repository.rit.edu/cgi/viewcontent.cgi?article=12934&context=theses

[2] Alam, M. Z., Rahman, M. S., & Rahman, M. S. (2019). A Random Forest based predictor for medical data classification using feature ranking. Informatics in Medicine Unlocked, 15, 100180. https://doi.org/10.1016/j.imu.2019.100180

[3] Arik, S. O., & Pfister, T. (2021). TabNet: Attentive Interpretable Tabular Learning. Proceedings of the AAAI Conference on Artificial Intelligence, 35(8), 6679-6687.

[4] Avanzi, B., Taylor, G., Wang, M., & Wong, B. (2024). Machine Learning with High-Cardinality Categorical Features in Actuarial Applications. Astin Bulletin, 54(2), 213–238. https://doi.org/10.1017/asb.2024.7

[5] Bustos, A., & Roa, J. (2021). Quantile encoder: A new approach for encoding high cardinality categorical features. arXiv. https://arxiv.org/abs/2105.13783

[6] Cerda, P., & Varoquaux, G. (2020). Encoding High-Cardinality String categorical variables. IEEE Transactions on Knowledge and Data Engineering, 34(3), 1164–1176. https://doi.org/10.1109/tkde.2020.2992529

[7] Chen, Y. (2022). Comparison of machine learning models: Linear model, forest model and support vector machine. Proceedings of the 2022 4th International Conference on Economic Management and Cultural Industry. https://www.ewadirect.com/proceedings/ace/article/view/11032

[8] Cho, S., Kim, S., Kang, S., Lee, K. J., Choi, D., Kang, S., Park, S. J., Kim, T., Yoon, C., Youn, T., & Chae, I. (2021). Pre-existing and machine learning-based models for cardiovascular risk prediction. Scientific Reports, 11(1). https://doi.org/10.1038/s41598-021-88257-w

[9] DeLong, E. R., DeLong, D. M., & Clarke-Pearson, D. L. (1988). Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. Biometrics, 44(3), 837-845.

[10] Dorraki, M., Liao, Z., Abbott, D., Psaltis, P. J., Baker, E., Bidargaddi, N., Wardill, H. R., Van Den Hengel, A., Narula, J., & Verjans, J. W. (2024). Improving cardiovascular disease prediction with machine learning using mental health data. JACC Advances, 3(9), 101180. https://doi.org/10.1016/j.jacadv.2024.101180

[11] Farzadfar, F. (2019). Cardiovascular disease risk prediction models: challenges and perspectives. The Lancet Global Health, 7(10), e1288–e1289. https://doi.org/10.1016/s2214-109x(19)30365-1

[12] Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? Advances in Neural Information Processing Systems, 35.

[13] Hossain, S., Hasan, M. K., Faruk, M. O., Aktar, N., Hossain, R., & Hossain, K. (2024). Machine learning approach for predicting cardiovascular disease in Bangladesh: evidence from a cross-sectional study in 2023. BMC Cardiovascular Disorders, 24(1). https://doi.org/10.1186/s12872-024-03883-2

[14] Janosi, A., Steinbrunn, W., Pfisterer, M., & Detrano, R. (1989). Heart Disease. UCI Machine Learning Repository. https://doi.org/10.24432/C52P4X.

[15] Jing, C., Zhang, L., Feng, L., Li, J., Liang, L., Hu, J., & Liao, X. (2024). Recommendations for prediction models in clinical practice guidelines for cardiovascular diseases are over-optimistic: a global survey utilizing a systematic literature search. Frontiers in Cardiovascular Medicine, 11. https://doi.org/10.3389/fcvm.2024.1449058

[16] Ke, G., et al. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Advances in Neural Information Processing Systems, 30.

[17] Liu, T., Krentz, A., Lu, L., & Curcin, V. (2024). Machine learning based prediction models for cardiovascular disease risk using electronic health records data: systematic review and meta-analysis. European Heart Journal - Digital Health, 6(1), 7–22. https://doi.org/10.1093/ehjdh/ztae080

[18] Mansoori, A., Zeinalnezhad, M., & Nazarimanesh, L. (2023). Optimization of Tree-Based machine learning models to predict the length of hospital stay using genetic algorithm. Journal of Healthcare Engineering, 2023, 1–14. https://doi.org/10.1155/2023/9673395

[19] Micci-Barreca, D. (2001). A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. ACM SIGKDD Explorations Newsletter, 3(1), 27–32. https://doi.org/10.1145/507533.507538

[20] Mochurad, L., Babii, V., Boliubash, Y., & Mochurad, Y. (2025). Improving stroke risk prediction by integrating XGBoost, optimized principal component analysis, and explainable artificial intelligence. BMC Medical Informatics and Decision Making, 25(1). https://doi.org/10.1186/s12911-025-02894-z

[21] Nusinovici, S., Tham, Y. C., Yan, M. Y. C., Ting, D. S. W., Li, J., Sabanayagam, C., Wong, T. Y., & Cheng, C. (2020). Logistic regression was as good as machine learning for predicting major chronic diseases. Journal of Clinical Epidemiology, 122, 56–69. https://doi.org/10.1016/j.jclinepi.2020.03.002

[22] Pal, M., Parija, S., Panda, G., Dhama, K., & Mohapatra, R. K. (2022). Risk prediction of cardiovascular disease using machine learning classifiers. Open Medicine, 17(1), 1100–1113. https://doi.org/10.1515/med-2022-0508

[23] Pargent, F., Pfisterer, F., Thomas, J., & Bischl, B. (2022). Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features. Computational Statistics, 37(5), 2671–2692. https://doi.org/10.1007/s00180-022-01207-6

[24] Peng, M., Hou, F., Cheng, Z., Shen, T., Liu, K., Zhao, C., & Zheng, W. (2023). Prediction of cardiovascular disease risk based on major contributing features. Scientific Reports, 13(1). https://doi.org/10.1038/s41598-023-31870-8

[25] Potdar, K., Pardawala, T. S., & Pai, C. D. (2017). A comparative study of categorical variable encoding techniques for neural network classifiers. International Journal of Computer Applications. https://www.ijcaonline.org/archives/volume175/number4/28474-2017915495/

[26] Prokhorenkova, L., et al. (2018). CatBoost: unbiased boosting with categorical features. Advances in Neural Information Processing Systems, 31.

[27] Teja, M. D., & Rayalu, G. M. (2025). Optimizing heart disease diagnosis with advanced machine learning models: a comparison of predictive performance. BMC cardiovascular disorders, 25(1), 212. https://doi.org/10.1186/s12872-025-04627-6

[28] Van Daalen, K. R., Zhang, D., Kaptoge, S., Paige, E., Di Angelantonio, E., & Pennells, L. (2024). Risk estimation for the primary prevention of cardiovascular disease: considerations for appropriate risk prediction model selection. The Lancet Global Health, 12(8), e1343–e1358. https://doi.org/10.1016/s2214-109x(24)00210-9

[29] Wang, C., et al. (2021). A comparison of multiple linear regression and tree-based regression models for prediction of battery cycle life. arXiv. https://arxiv.org/abs/2111.02513

[30] Wessler, B. S., et al. (2019). Clinical prediction models for cardiovascular disease. Circulation Research. https://www.ahajournals.org/doi/10.1161/CIRCRESAHA.118.312823

[31] Wibisono, A., Wu, Y., & Yang, F. (2024). Optimal score estimation via empirical Bayes smoothing. arXiv. https://arxiv.org/abs/2402.07747

April 23–25, 2026
University of the Immaculate Conception, Davao City

Author Index

Abarico, M.K.A.C., 139
Abellana, P.T., 145
Abot, N.F., 45
Albarracin, F.M., 64
Aliac, C.J., 103
Alonzo, R.F., 33
Andaya, K.C., 82
Antonio, H.C., 64
Araneta, J.K., 39
Asaria, J.F.C., 15
Asturiano, C.E., 90
Azcarraga, J., 131
Bajo, J.H., 103
Barcelita, A.J., 33
Bascug, F.M.F., 123
Bautista, T.J., 52
Benito, M.J.B., 109
Bernardez, F.G., 97
Binanitan, J.R., 52
Cabredo, R., 25
Cahilog, K., 151
Caliboso, R.G., 82
Cambarijan, A.S., 103
Cariaga, J.E., 33
Cataluña, J.E., 131
Catubig, N.M., 145
Chan, A.R., 9
Chan, C.R., 139
Chan, K.A., 76
Chavez, F.B., 145
Chua, H.S., 90
Cuya, V., 25
Dela Cruz, S.A., 115
Espino, Z., III, 52
Fabian, A., 9
Gemina, R., 123
Go, A.D., 82
Go, J.R., 82
Gomez, E.J.S., 109
Gonda, R., 33
Gonzales, J.N., 33
Guanzon, K.L., 3
Limbo, M., 70
Llamera, J.M., 58
Madrid, V.R., 39
Manulat, T.D., 20
Marticio, V.H., 58
Mendoza, B.X., 123
Natavio, R.J., 3
Ocampo, G., 25
Ordanel, I.D., 64
Paguiligan, J.A., 131
Paule, F.M., 58
Que, N.B., 109
Regis, M.A.J., 115
Repuesto, C.R., 103
Reyes, A.L., 25
Reyes, N.K., 33
Roy, R.E., 76
Seperidad, A., 131
Sta Romana, C.L., 145
Talip, D.D., 103
Tito, A., 145
Torayno, E.J., 15
Torres, F.L., 97
Uy, N.K., 90
Vidal, E., 139

Institution Index

Ateneo de Manila University, 109, 139
Cebu Institute of Technology – University, 20, 103, 123
De La Salle University, 25, 33, 82, 90, 131
De La Salle University – Dasmariñas, 45, 52
Holy Child College of Davao, 9
South Philippine Adventist College, 3, 15
University of San Carlos, 151
University of the Philippines Cebu, 97
University of the Philippines Diliman, 64, 70, 76
University of the Philippines Los Baños, 39, 115
Visayas State University, 115
Affiliation unavailable in source PDF, 58, 145