Proceedings of the 24th Philippine Computing Science Congress Volume II: Student Research Workshop

Editors: Dr. Judith J. Azcarraga, Dr. Ethel Chua Joy Ong & Mark Edward M. Gonzales

ISSN 1908-1146

Computing Society of the Philippines
csp.org.ph

Contents

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.

About the Student Research Workshop

Introduced in this year's PCSC, the **Student Research Workshop** invites students at various stages of their research to present their work and receive mentorship and feedback from the research community. This is also an opportunity to present exciting contributions that showcase innovative technologies and detail preliminary experiments that foster discussions to provoke new ideas to emerge.

Paper Submission

The Student Research Workshop welcomes papers in two different categories:

- **Thesis Proposals.** This category is appropriate for students who have already decided on a thesis topic and wish to receive feedback on their proposal and broader ideas for their ongoing work.
- **Research Papers.** This category accepts work-in-progress, experiments with preliminary results, and late-breaking work. We encourage submissions describing experiments with positive and negative results that can provide relevant insights to the computing community. The first author must be a current student.

Submissions in both categories may be archival or non-archival, based on the decision of the authors. All archival papers are published in the SRW proceedings. Extended versions of archival papers and non-archival papers may be submitted to any venue in the future.

Paper Review

Papers submitted to the Student Research Workshop are reviewed through a Reviewed process as defined by the Association for Computing Machinery (ACM), and receive light feedback from reviewers. Evaluation criteria are as follows:

- **Contribution.** Does this work present (potential) research contributions or ideas that will stimulate interesting conversations among PCSC attendees?
- **Originality.** How does the work build on, expand, or innovate from existing work in the area?
- **Validity.** How well are the chosen methods described and justified within the submission?
- **Clarity.** How clear, understandable, and targeted is the writing?

Benefits of Submitting

The benefits of submitting to the Student Research Workshop are as follows:

- **Mentorship.** The Student Research Workshop provides an opportunity for students to receive constructive feedback from the computing research community.
- **Publication Track Record.** Should an author opt for an archival publication, your paper can help your research gain visibility while also improving your publication record. This is particularly beneficial when applying for advanced graduate studies here and abroad.
- **Exploratory Studies.** Submissions with preliminary results can provide insights on why and in which scenarios a particular method succeeds or fails, thus helping both the students and the computing community to work together in advancing the field.

Post-Presentation Mentorship

The Computing Society of the Philippines offers students the opportunity to receive mentorship and guidance in preparing their research papers, should they opt for archival publication. The goal of the mentorship program is to help students improve the quality of writing for publication.

Each paper is assigned to a mentor who will review and provide feedback in the form of guidelines and suggestions to improve the overall writing. This mentor is not the same person who reviewed the final submission for publication.

Organizing Committee

Program Chair and Local Host Chair

Judith J. Azcarraga, Ph.D.
De La Salle University

Student Research Workshop Chair

Ethel Chua Joy Ong, Ph.D.
De La Salle University

Organizing Committee Members

Ann Franchesca B. Laguna, Ph.D.
De La Salle University

Neil Justin V. Romblon
De La Salle University

Michelle Renee D. Ching, DIT
De La Salle University

Candy Joyce H. Espulgar
De La Salle University

Organizers

Computing Society of the Philippines

Association for Computing Machinery Women in Computing


College of Computer Studies, De La Salle University

Predicting Philippine Undergraduate Employability in Mock Interviews using XGBoost, CatBoost, and LightGBM

Gary Louis P. Garcia
South Philippine Adventist College
Matanao, Davao del Sur, Philippines
garylouis.p.garcia@gmail.com

Jane T. Antoque
South Philippine Adventist College
Matanao, Davao del Sur, Philippines
janeantoque7@gmail.com

ABSTRACT

Employability encompasses well-rounded talents beyond technical skills. Mock interviews offer insights into employability, and we propose a predictive model using mock interview ratings and gradient boosting algorithms such as LightGBM, XGBoost, and CatBoost. We employed a rigorous validation process, using 5-times repeated 10-fold cross-validation on 70% of the dataset and a separate 30% for testing. After hyperparameter optimization, LightGBM and XGBoost attained an accuracy of 91.5%. All models demonstrated a precision of 93.7%. Both LightGBM and XGBoost achieved a recall rate of 91.2%. Notably, XGBoost exhibited the highest AUC of 98.1%. Feature importance analysis reveals a combination of key factors enhancing employability including cognitive abilities, physical presentation, communication skills, practical experience, and self-confidence.

KEYWORDS

Undergraduate, employability, interview, LightGBM, XGBoost, CatBoost

1 INTRODUCTION

Employability is the ability of an individual to secure the right job that matches their education [8]. Despite the importance of technical skills and experience, true employability goes beyond by cultivating a well-rounded set of talents and achievements that make graduates stand out. Not only does this increase your chances of landing a good job, but it also paves the way for long-term fulfillment and success, benefiting not only yourself, but also your future employers, the local community, and even the national economy [12].

Mock interviews are simulated job interviews that provide individuals the opportunity to practice answering common interview questions and gain experience interacting with potential employers in a formal setting [7].

Recent studies have been conducted to predict the employability of undergraduate students using mock interview ratings. A study by Casuat and Festijo in 2019 [2] predicted the overall employability of undergraduate students where classifiers employed are decision tree (DT), random forest (RF), and support vector machines (SVM). Among the three classifiers, SVM had the highest score in all classification metrics, namely accuracy, recall, precision, and F1-score, of 91 2%. In 2020, they have extended their study [3]. They identified the most predictive attributes among the employability signals of undergraduate students using the scores generated by three feature reduction techniques with SVM with SMOTE such as recursive feature elimination (RFE), univariate selection (US), and principal component analysis (PCA).

In the present effort, we utilized a dataset of mock interview ratings collected from the Kaggle website for undergraduate students across various disciplines. We employed gradient boosting algorithms, namely XGBoost, LightGBM, and CatBoost, to build predictive models of student employability based on mock interview performance. Our analysis also includes key findings about employability indicators, which confirm some, but also provide novel insights compared to previous studies [2, 4].

The general objective of this study is to predict the employability of undergraduate students based on mock interview ratings using LightGBM, XGBoost, and CatBoost. Specific objectives of this study are as follows.

- Evaluate the accuracy, precision, recall, and f1-score of LightGBM, XGBoost, and CatBoost in predicting the employability of undergraduate students.
- Compare the performance of LightGBM, XGBoost and CatBoost in predicting student employability, measured by the key metric categories of accuracy, precision, recall, f1-score, and area under the curve (AUC).
- Generate feature importance plot for LightGBM, XGBoost, and CatBoost to identify the most predictive features for undergraduate students' employability.

The remainder of this paper is organized as follows. Section 2, the Methodology, presents the experimental setup. Section 3, the Results and Discussions presents our findings and their interpretations. Finally, Section 4, Recommendations and Future Works summarize our key insights and pave the way for future research directions.

2 METHODOLOGY

2.1 Dataset Collection

This study uses the publicly available data set titled 'Students' Employability Dataset-Philippines' hosted on Kaggle https://www.kaggle.com/datasets/anashamoutni/students-employability-dataset. The dataset comprises mock job interview results for 2,982 students from various university agencies across the Philippines. It adheres to the Data Privacy Act, ensuring participant anonymity and confidentiality. However, biases inherent in mock interview evaluations may limit its representativeness of the entire student population. Table 1 shows included features and descriptions, based on Casuat and Festijo's work [3], except for 'Self-confidence,' which is described by the authors of this study.

Table 1: Students' Employability Dataset

Feature Name	Description
General Appearance	The way a person looks in general
Manner of Speaking	Appropriate style of expressing oneself
Physical Condition	The condition or state of the body
Mental Alertness	The state of active attention of the mind
Self-confidence	Manifests as poise, conviction, and clear, persuasive communication.
Ability to Present Ideas	The ability to present the ideas clearly
Communication Skills	The ability to convey ideas to others effectively and efficiently
Internship Student Performance Rating	The performance assessment conducted by the immediate superior of OJT

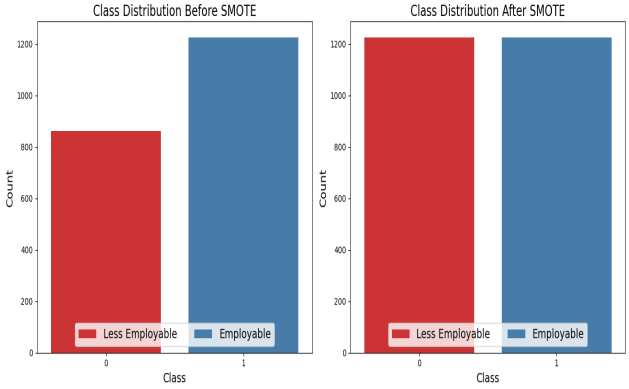


Figure 1: Class distribution before and after SMOTE.

2.2 Dataset Pre-Processing

The dataset contained no missing values. The 'Student number' feature was deemed irrelevant and removed. The 'Class' feature labels were transformed to numerical values: 'employable' as 1 and 'less employable' as 0 for machine learning algorithms. A 70/30 split was used to divide the dataset into training and testing sets, allowing for training three different algorithms while ensuring a fair evaluation on unseen data.

2.3 Handling Class Imbalance

The dataset had a class imbalance, with 42% labeled 'less employable' and 58% 'employable'. To address this without reducing overall observations, Synthetic Minority Oversampling Technique (SMOTE) was employed. SMOTE generated new data for 'less employable', balancing classes to a 50/50 ratio as shown in Figure 1. This mitigated bias towards the majority class, enhancing model performance.

2.4 Model Selection

Previous study [2] predicting student employability using similar features have primarily focused on none-boosting machine learning algorithms. This study explores the potential of boosting algorithms for this task. Three prominent boosting algorithms were chosen for analysis: LightGBM [6], XGBoost [5], and CatBoost [9]. This selection of models allows for a comprehensive evaluation of boosting algorithms compared to none-boosting methods for predicting student employability.

2.5 Hyperparameter Tuning

We have implemented grid search hyperparameter optimization to improve the performance of LightGBM, XGBoost and CatBoost models. Our goal was to find the best parameter values for each model and assess its impact on classification metrics including AUC, accuracy, precision, recall and F1 scores. The optimized hyperparameters for each model are shown in Table 2.

Table 2: LightGBM, XGBoost, and CatBoost Hyperparameters

Models	*Hyperparameter*	*Value*
LightGBM	subsample	0.8
	reg_lambda	0
	reg_alpha	0.5
	num_leaves	36
	n_estimators	200
	min_child_samples	20
	max_depth	7
	learning_rate	0.05
	colsample_bytree	1.0
XGBoost	max_depth	6
	alpha	1
	learning_rate	0.1
	n_estimators	500
	subsample	0.6
	colsample_bytree	0.8
	min_child_weight	1
	gamma	0.1
	reg_alpha	1
	reg_lambda	1
CatBoost	depth	6
	iterations	150
	l2_leaf_reg	3
	learning_rate	0.5

2.6 Model Evaluation

We employed repeated k-fold cross-validation by splitting the 70% training data into 10 folds. Each boosting model was trained on nine folds and evaluated on the held-out fold which is repeated five times for robust performance estimation. The remaining 30% served as the test set to gauge model generalization. Performance was assessed using the following performance metrics.

- **Accuracy:** It computes the ratio of correctly classified instances to the total number of instances [10].

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

- **Precision:** It is the ratio of true positive instances divided by the total number of instances predicted as positive [11].

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

- **Recall:** Given as the ratio of relevant instances that are recovered [11].

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

- **F1-Score:** It is the combination of both precision and recall used to get the average value of them [1].

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

Additionally, ROC curves and associated AUC were used to evaluate the model's class discrimination ability across various thresholds, offering a comprehensive performance view.

2.7 Predictive Features Identification

To understand which factors or features drive decision-making among the models, we analyzed feature importance scores. Visualizing these scores as column charts helped us identify key predictors for each model. Comparing the results across models revealed features consistently highlighted as top predictors by multiple models.

3 RESULTS AND DISCUSSIONS

3.1 Evaluation and Comparison of Models' Performance

LightGBM, XGBoost, and CatBoost all have very similar performance metrics with respect to accuracy, precision, recall, and F1-score. Their accuracy rates are around 91.5%, 91.5%, and 91.4%, respectively, which indicates that the three models are highly effective in predicting the employability status of individuals, as shown in Table 3.

The precision of 93.7% for all models suggests that when they predict an individual as employable, there is a high chance of this prediction being accurate. This is crucial for avoiding false positives in employability predictions, where predicting someone as employable when they are not could lead to inefficient use of resources.

Recall values are slightly different, with LightGBM and XGBoost at 91.2% and CatBoost at 91.0%, showing that LightGBM and XGBoost are marginally better at identifying all actual employable cases. However, this difference is minimal. The F1-scores are also very close, with LightGBM and XGBoost at 92.4% and CatBoost at 92.3%, indicating a balanced performance between precision and recall across the models.

The AUC (Area Under the ROC Curve) values are impressive across all models, ranging from 0.979 to 0.981. This suggests that the three models have excellent capability in distinguishing between the employable and less employable individuals. The slight differences in AUC values (LightGBM at 0.979, CatBoost at 0.980, and XGBoost at 0.981) may not be practically significant, considering the overall high performance.

Furthermore, in the previous study of Casuat and Festijo [2] , an SVM model was used, achieving a 91.22% accuracy rate. Although the experimental setups differ, the current models' performance remains comparable and on par with the SVM's performance. The models' performance are collectively shown in Table 3.

Table 3: Models' Performance Comparison

Model	*Accuracy*	*Precision*	*Recall*	*F1-Score*	*AUC*
SVM[2]	0.9122	0.9115	0.910	0.910	–
LightGBM	0.915	0.937	0.912	0.924	0.979
XGBoost	0.915	0.937	0.912	0.924	0.981
CatBoost	0.914	0.937	0.910	0.923	0.980

3.2 Predictive Features Analysis

In the quest to better understand factors influencing student employability, the feature importance rankings provide insights into the aspects that each machine learning model (i.e., LightGBM, XGBoost, and CatBoost) considers most influential for predicting a student's likelihood of securing employment, in the context of job interviews.

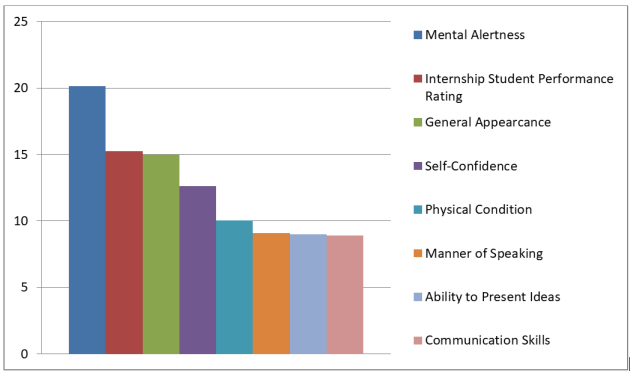


Figure 2: CatBoost Feature Importance Chart

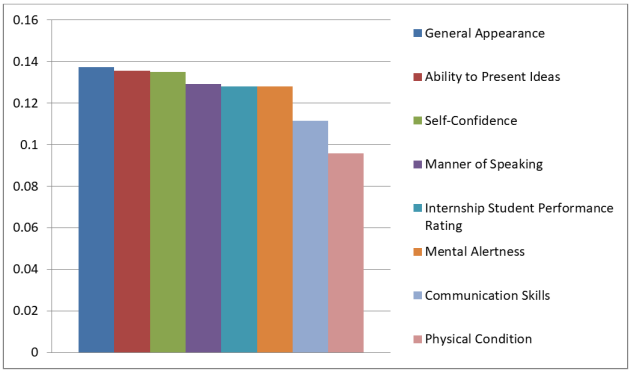


Figure 3: XGBoost Feature Importance Chart

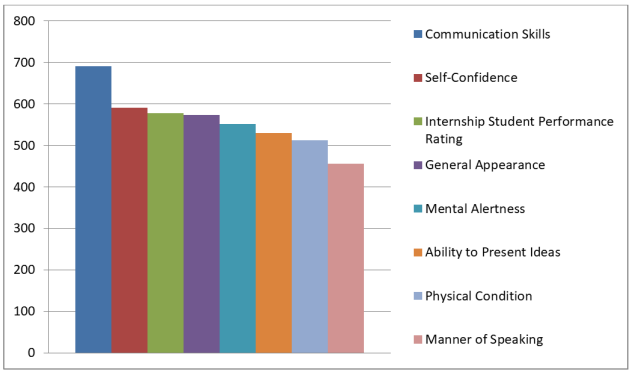


Figure 4: LightGBM Feature Importance Chart

In the CatBoost model, as shown in Figure 2, Mental Alertness is identified as the most crucial feature according to CatBoost. This suggests that the model places high importance on cognitive abilities, attentiveness, and quick thinking. The internship student performance rating is also important, indicating that the model

considers past OJT performance as a significant factor in predicting employability. General Appearance and Self-Confidence are also ranked high, suggesting that the model gives importance to how students present themselves, both physically and in terms of confidence. Soft skills such as communication skills and the ability to present ideas are also considered important, but are ranked lower than the aforementioned factors.

On the other hand, XGBoost, as shown in Figure 3, places the highest importance on General Appearance, suggesting that the overall appearance of a student plays a significant role in predicting employability. The ability to Present Ideas and Self-Confidence follow closely in importance, indicating that the model values students' abilities to articulate and express themselves. Mental Alertness and Student Performance Ratings are also considered important but are ranked slightly lower. Communication Skills and Physical Condition are relatively lower in importance according to XGBoost.

Furthermore, LightGBM, as shown in Figure 4, places the highest importance on Communication Skills, suggesting that effective communication is a key factor in predicting student employability. Self-confidence and Internship Student Performance Ratings follow closely, indicating the significance of confidence and OJT performance. General Appearance and Mental Alertness are also considered important. The ability to Present Ideas and Physical Condition are ranked lower in importance according to LightGBM.

Comparing across models, we observe that Mental Alertness, General Appearance, and Communication skills appeared as top-ranked qualities. Moreover, employability qualities such as Self-Confidence, Internship Performance Rating, and General Appearance are common in at least two models. This suggests their potential universal importance in employability prediction. Additionally, each model prioritizes unique features. CatBoost focuses on cognitive abilities, XGBoost emphasizes physical presentation and communication skills, while LightGBM highlights communication skills and confidence.

In the previous study by Casuat and Festijo [3] , the identified key predictors of student employability are Manner of Speaking, Mental Alertness, and Ability to Present Ideas. We can observe an overlap between the current study and previous work regarding Mental Alertness being a crucial factor in both. This reinforces its potential universal importance in employability prediction. Interestingly, both studies also identify communication skills as significant, although emphasized differently by the chosen models in this study and directly mentioned as a key predictor in the previous work. Exploring further, the current study identifies a broader range of influential features, including General Appearance, Communication Skills, and Self-Confidence, which are not explicitly mentioned in the previous work.

In light of this, we can emphasize the qualities undergraduates should need to be employable such as Mental Alertness, Communication Skills, General Appearance, Self-Confidence, Internship Student Performance Rating, and Ability to Present Ideas as shown in Figure 5. This means that a combination of cognitive abilities, physical presentation, communication skills, practical experience, and self-confidence are key determinants of student employability, reflecting the multifaceted nature of readiness for the workplace. Candidates who possess and demonstrate these qualities are more likely to stand out to employers and succeed in securing employment opportunities.

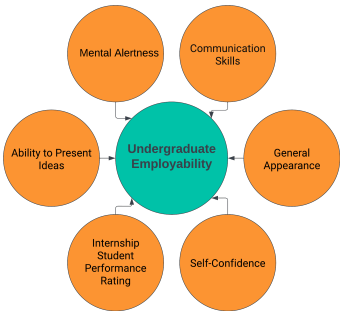


Figure 5: Employability Qualities

4 RECOMMENDATIONS AND FURTHER WORK

All models such as LightGBM, XGBoost, and CatBoost perform similarly well in predicting employability status, with high accuracy, precision, recall, and AUC values. Given their comparable performance, the choice between them can be based on factors like ease of implementation, computational efficiency, or specific project requirements. Feature importance may influence the choice of a model for the task at hand. Feature importance rankings can validate or challenge existing domain knowledge about crucial employability factors. For example, if educators prioritize fostering cognitive abilities, CatBoost's emphasis on Mental Alertness might make it a preferred choice for training programs. Further efforts can focus on continuously collecting data and refining models to better account for evolving job market dynamics and educational trends. These enhancements have the potential to improve the accuracy and relevance of employability predictions.

ACKNOWLEDGMENTS

We are grateful to Ms. Neila M. Paglinawan-Muñez for her insightful comments and helpful suggestions, which significantly improved this work.

REFERENCES

[1] A. Alhassan, B. Zafar, and A. Mueen. 2020. Predict students' academic performance based on their assessment grades and online activity data. *International Journal of Advanced Computer Science and Applications* 11, 4 (2020).
[2] C. D. Casuat and E. D. Festijo. 2019. Predicting students' employability using machine learning approach. In *2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*. 1–5.
[3] Cherry D Casuat and Enrique D Festijo. 2020. Identifying the most predictive attributes among employability signals of undergraduate students. In *2020 16th IEEE international colloquium on signal processing & its applications (CSPA)*. IEEE, 203–206.
[4] C. D. Casuat, E. D. Festijo, and A. S. Alon. 2020. Predicting students' employability using support vector machine: a smote-optimized machine learning system. *International Journal* 8, 5 (2020).
[5] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.
[6] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017).
[7] Rhiannon Lord, Ross Lorimer, John Babraj, and Ashley Richardson. 2019. The role of mock job interviews in enhancing sport students' employability skills: An

example from the UK. *Journal of Hospitality, Leisure, Sport & Tourism Education* 25 (2019), 100195.

[8] National Committee of Inquiry into Higher Education (Great Britain), Ron Dearing, and Sir Ron Garrick. 1997. *Higher education in the learning society.* The Committee.

[9] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. CatBoost: unbiased boosting with categorical features. *Advances in neural information processing systems* 31 (2018).

[10] O. Saidani, L. J. Menzli, A. Ksibi, N. Alturki, and A. S. Alluhaidan. 2022. Predicting student employability through the internship context using gradient boosting models. *IEEE Access* 10 (2022), 46472–46489.

[11] P. Thakar and A. Mehta. 2017. Role of Secondary Attributes to Boost the Prediction Accuracy of Students Employability Via Data Mining. *arXiv preprint* (2017). arXiv:1708.02940

[12] Mantz Yorke. 2006. *Employability in higher education: what it is-what it is not.* Vol. 1. Higher Education Academy York.

Comparative Analysis of Machine Learning Techniques in the Classification of Pili *(Canarium ovatum Engl.)* Fruit Varieties

Leo Constantine S. Bello
Ateneo de Naga University
Naga City, Camarines Sur
lbello@gbox.adnu.edu.ph

Joshua C. Martinez
Ateneo de Naga University
Naga City, Camarines Sur
joshuamartinez@gbox.adnu.edu.ph

ABSTRACT

Pili is one of many Philippine fruit trees that are endemic in the country and are predominantly found in the Bicol Region. Existing means of classifying Pili are available and highly accurate but are known to be invasive, destructive, costly, and require a highly technical background. Non-invasive means through Computer Vision techniques have been applied to some fruits, but no study has applied it yet in classifying pili. This study aimed to provide a comparative analysis of four (4) machine learning algorithms: LDA, k-NN, SVM, and CNN, commonly used in image classification, and assess which is best suited for further study and application.

KEYWORDS

Pili, Canarium ovatum Engl., Comparative Analysis, Image Processing, Variety Classification

Figure 1: Visual feature differences among pili fruit varieties

1 INTRODUCTION

The Philippines is home to abundant fruit and nut trees. It is said to be the center of diversity for several fruit trees that bear edible nuts. *Canarium ovatum Engl*, also known as Pili, is one of the most important edible nut-bearing trees. This species of the Canarium genus is endemic in the country, and its high density of population distribution and growth is restricted to areas relatively close to its center of origin, the Bicol Region. Its fruit kernel distinguishes Pili. Pili has been designated as a top priority, high-value, and high-impact fruit crop due to the government's research and development efforts, joining the ranks of mango, durian, lanzones, rambutan, banana, papaya, and citrus.

The Pili tree is known as the "Tree of Hope" because of its many uses, from sap to roots. The most important part of the fruit, the kernel, is processed into pastries and confectioneries, and the whole kernel is now emerging in the local and international markets. Its pulp is considered a delicacy in some areas outside the region, and its shells can be used to make charcoal fuel or handicrafts. Its nut oil has potential applications in the food and pharmaceutical industries and is in high demand for domestic and international exports.

Pili has been identified as a highly variable species. The trees differed in growth habits, fruiting season, yield, response to asexual propagation, stem diameter, leaf size, number of flower clusters/shoot, and flowering period. One of the species' most noticeable features is its fruits, which vary in shape, color, weight, thickness (pulp and shell), flavor, kernel size, and content. The fruit is 4-6 cm in size and comes in elliptical, oblong, oval, and obovate shapes. Its pulp turns from green to dark purple to nearly black as it ripens. Saturated and unsaturated fatty acid percentages in the oil vary, as do the percentages of filled nuts and kernel recovery in seedling trees. Calcium, potassium, zinc, and phytic acid concentrations vary from variety to variety [6]. When identifying fruit varieties, someone with sufficient knowledge and expertise could determine and distinguish variations based on their visual characteristics, as shown in Fig.1.

Fruit variety classification is critical for producers and farmers to ensure the purity and crop output of the variety. Furthermore, because of their potential to increase productivity and profitability, saplings and fruits from the recommended variety for production were deemed more expensive. However, the morphological and color characteristics of pili fruit from various types are strikingly similar, with significant overlap. A person with little to no knowledge of or experience with different types may have difficulty identifying one. These circumstances may present an ideal opportunity for any enterprising individual to offer low-quality pili variants illegally, as high-quality increases profit margins. Furthermore, some orchards grow a diverse range of pili varieties to adapt to changing climates, harvest seasons, and other agronomic factors. Many small orchards grow diverse fruits with varying demand and market value, which can complicate local post-harvesting due to unintentional and fraudulent mixing of different fruit varieties. This problem is not new for other crop groups such as rice [8] and corn [4].

Historically, Pili variations were identified solely by visual inspection. Botanists use plant traits as identification keys, then conduct sequential and adaptive research to identify varied plant kinds. The technique focuses on answering questions about pili fruit qualities, including form, color, length, and width. Consistently focusing on unique traits refines the species pool. Accurately responding to inquiries leads to the desired diversity. Additionally, farmers

and producers rely on manual inspection and sorting due to cost-effectiveness and limited professional availability. The classification outcome may be affected by investigator competence and subjectivity, which can lead to discrepancies, workload, and tiredness. Identifying traditional varieties is difficult for the general public, but considerably more difficult for botanical experts such as conservationists, farmers, foresters, and product designers who face frequent problems. Identifying a certain type might be challenging even for professionals. Drawbacks of this method include high error rates, low precision, and significant processing time, particularly for specific types.

High-performance analytical procedures that are commonly used include liquid chromatography, gas chromatography-mass spectrometry [9], seed protein electrophoresis [11], and DNA molecular markers [12]. Despite their high accuracy, most of these technologies are invasive, harmful, hazardous to human health, time-consuming, sophisticated, and expensive, with little chance of being repeatable in the future. As a result, it is critical to use a safe, non-destructive, and accurate automated system for variety classification. These non-destructive automation processes can save money by increasing efficiency while reducing subjectiveness caused by human experts. Numerous studies in the field of agronomy have demonstrated the use of non-destructive plant species classification techniques such as magnetic resonance imaging [16], electronic tongue [10], acoustic method [15], electronic nose [1], and computer vision. Computer vision and image processing are two methods for classifying crops that are both low-cost and provide significant analytical and computational power. Image pre-processing, segmentation, feature extraction, and classification are the steps in computer vision-based classification, with feature extraction significantly impacting both classification accuracy and classification precision.

Several studies already implemented the use of computer vision techniques in the classification of crops and plants. In this study, the researchers would like to explore some notable machine learning techniques such as Linear Discriminant Analysis (LDA), k-Nearest Neighbor (k-NN), Support Vector Machine (SVM), and Convolutional Neural Networks (CNN) wherein they performed well in classifying fruit subjects [2–5, 7, 13, 14], and assess its viability in accurately classifying the pili fruit according to its variety.

2 METHODOLOGY

2.1 Data Gathering

In this study, a total of 1,400 pieces of ripe pili fruit samples from seven (7) varieties were harvested at the Albay Research and Development Center, Department of Agriculture Field Office V in Buang, Tabaco City. The sample collection comprised two hundred (200) fruits from each variety, namely, Lanuza, Laysa, Magayon, Magnaye, Mayon #1, M. Orolfo and Orbase. All fruits collected were from the varieties labeled parent trees with the assistance of a resident agriculturist. The harvest was done one tree at a time. It was to ensure no occurrences of mixing up of fruit samples.The collection was done in August 2023 during one of the fruit's maturing months.

To augment the existing dataset, the researcher gathered another set of two hundred (200) fruit samples from pili cultivars from four (4) different trees. These species of pili are a product of selective propagation and were not considered as varieties. Selection of the cultivar trees were assisted by the resident agriculturist. These samples are were comprised to form another group of data called 'Cultivars'. This was done to ensure that the model would recognize distinctions and discrimination between varieties and cultivars.

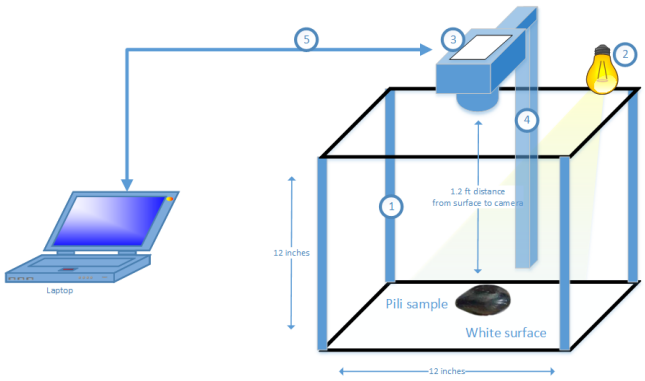


Figure 2: Image Acquisition Setup

2.2 Image Acquisition: Materials and Methods

To capture the images of the fruits, an image acquisition system (Fig. 2) was assembled. The aim was not to implement a final version of the system but a prototype mimicking conditions or environment where samples were to be captured. The system was conceived to acquire images with high contrast between the subject and the background with the aim of capturing image with minimal shadow cast. This system underwent multiple experimentation on the configuration of the camera as well as the lighting before capturing pili samples for further image analysis.

2.3 Image Pre-processing

Captured original images underwent background subtraction to remove unnecessary objects. Afterwards the images were cropped to have equal height and width of 2912 x 2912 pixels centered to the subject. To lessen the storage and memory space during the image analysis,image-set underwent to a process of resizing images to 700 x 700 pixels. This is done through batch processing employing python 3.11.0 and OpenCV's image processing capabilities.

Images were then subjected to background cleaning to remove unnecessary objects such as dirt and foreign objects found in the background. This is followed by replacement to plain white background to provide ease in further image processing activities such as thresholding and binarization which were performed during feature extraction.

To augment the dataset, images were subjected to horizontal flip followed by vertical flip. Each of the flip were stored to a separate folder named according to their respective varieties. To further augment the dataset, images were rotated 90 degrees, 180 degrees and 270 degrees. Considering the original and the augmented images resulted to total of 57,600 images, having 7200 for each variety. Due to computational cost for having 700 x 700-pixel images in feature extraction and other process in image analysis, resizing images to 256x256 pixels were considered in this study.

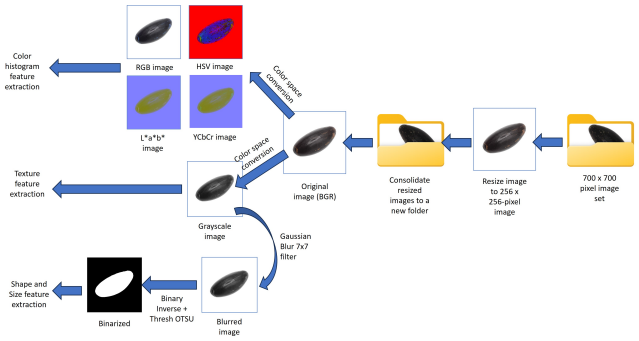


Figure 3: Image pre-processing activities prior to feature extraction.

2.4 Feature Extraction

The next step in developing a classifier model after the image pre-processing is feature extraction. Main and important visual external features for a fruit in general are its color, texture, shape and size. Fig. 3 illustrates further the image pre-processing activities before feature extraction. Extracted features were saved in Comma Separated Values (.csv) files to be loaded during the classifier modelling. It should be noted that features under color, texture, shape and size were hand-crafted and is intended for modelling traditional machine learning algorithms such as, in this study, LDA, SVM and K-NN. Higher level feature extraction from a transfer learning technique was used for the CNN model.

2.4.1 Color feature extraction. Color histogram analysis was performed on different color spaces such as RGB, HSV, L*a*b*, and YCbCr. This is one of the most important tasks in image processing and computer vision tasks. Each of the mentioned color spaces provide valuable insights on the development of the pili fruit. In this study, the features were extracted accordingly and were summarized using Histogram Mean and Standard Deviation.

2.4.2 Texture feature extraction. Three (3) methods were used to extract texture features: GLCM (Grey-level Co-occurrence Matrix), GLRLM (Grey-level Run Length Matrix) and the DWT (Discrete Wavelet Transform). The GLCM extracts the features Contrast, Energy, Homogeneity and Correlation. GLRLM extract the features Short Run Emphasis (SRE), Long Run Emphasis (LRE), and Gray-level Non-Uniformity (GLN). DWT was used to extract Approximate and Detailed Energy, and Shannon Entropy. features

2.4.3 Shape feature extraction. Shape features were extracted after defining the image contours. Shape feature extracted were: Area, Perimeter, Compactness, Roundness, Aspect Ratio, Eccentricity, Solidity, and Hu Moments.

2.4.4 Size feature extraction. To extract the size of the fruit samples, the researcher used the features bounding box dimension (width and height), and the diameter.

2.4.5 High-level feature extraction. To extract the High-level visual features, the weights of ImageNet[1] coupled with VGG168[2] CNN Model was used. Using this pre-trained transfer learning technique, the researcher aimed to extract features that cannot be extracted by hand-crafted way.

2.5 Classifier Modelling

In modeling a non-invasive image-based classifier for pili varieties, three (3) traditional Machine Learning models – Linear Discriminant Analysis (LDA), Support Vector Machine (SVM) and k-Nearest Neighbor (k-NN), and a Deep Learning model – Convolutional Neural Network (CNN) was considered.

2.5.1 Traditional Machine Learning. In loading the dataset in the model, the traditional machine learning used hand-crafted features: Color, Texture, Shape and Size feature vectors in .csv format. These features were loaded individually, and in combination of all four features to test the effectivity of the designed traditional machine learning model.

The feature sets were split into the ratio of 70:15:15, wherein 70 percent of the dataset is the training, 15 percent for the testing, and another 15 percent for the validation. Selection of the data samples were done in random. Due to values inside the datasets were diverse in range, the training, testing, and validation sets were subjected to normalization before loading them into the model. To reduce the dimensionality of the datset, Principal Component Analysis (PCA) was used. In this study, the number of components to be retained was 10. This was to set to balance the trade-off between dimesionality and accuracy (overfitting or underfitting) of the models.

(1) Linear Discriminant Analysis (LDA) - The LDA was set with 'lsqr' solver or Least Squares solution. This was considered due to the 'lsqr' solver can be applied to high-dimesional but sparse data because it leverages sparsity to speed up computation. The shrinkage or the regularization of the model was set to 'auto' where it used the Ledoit-Wolf Lemma. This helped the model in improving its robustness and stability especially with the study's scenario of high dimensional data. Lastly, the number of components were into 'None' this is to preserve the original values of feature values.
(2) Support Vector Machine (SVM) – The models' kernel is set to 'poly' or Polynomial. This is due to the data not linearly separable. The C which is the Regularization parameter is set default value of 1. This is because larger values tend to overfit and lesser values of C tends to underfit the resulting model. The gamma was set to 'scale'. The decision function shape was set to 'ovr' or one-vs-rest strategy of SVM to accommodate multi-class classification. 'ovr' create a total comparison of N X number_of_binary_classifiers, where N is the number of classes. In comparison with 'ovo' (one-vs-one), this strategy has faster training time and works well with imbalances in classes.

[1]Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009.
[2]Tammina, Srikanth. "Transfer learning using vgg-16 with deep convolutional neural network for classifying images." International Journal of Scientific and Research Publications (IJSRP) 9.10 (2019): 143-150.

(3) k-Nearest Neighbor (k-NN) – The k-NN model set the number of neighbors to 30. This is to ensure that the result of the model is a balanced due to lesser value tend to overfit the model. The value of the power parameter p is set to 2 so that the Minkowski distance would be transformed to Euclidean. The algorithm was set to 'auto' to let the classifier decide the most appropriate algorithms from BallTree, KDTree and Brute Force depending on the values passed to the train function. Lastly, the weights were set to 'uniform'. This sets all the neighbors to be weighted equally.

2.5.2 Convolutional Neural Network. This work utilized the embedded data generator of keras to create a dataset from actual images, instead than depending on feature vectors. The original dataset photos were separated into 70:15:15 ratios: 70% for training, 15% for test, and 15% for validation. This yielded 420 original photos per class folder for training, 90 for testing, and 90 for evaluation. This value is multiplied by eight (8) recognized classes. The CNN model was defined with 8 classes, 32 batches, and 100 epochs. Augmentations included rotation, width, height, shear, zoom, horizontal flip, vertical flip, and fill nearest. This increases the randomness and variation of the original dataset.

The CNN base model combined transfer learning from VGG16 with ImageNet weights. Input images were resized to 224 x 224 pixels for VGG16 design. Using sequential design, the CNN model can be developed from input through output, with layers stacked early. The CNN model was augmented with a pre-trained base model. Flattened, this becomes a 1-dimensional vector. A Dense or Fully linked layer with 512 neurons with 'ReLU' activation was added to the original model after flattening. A Dropout of 0.5 was introduced to the model to prevent overfitting. To penalize the loss function, L2 regularization was added to the dense layer with a value of 0.01. In the last Dense layer, eight neurons reflect the number of classes in the classification challenge. Softmax, often used for multi-class classification, was utilized in this layer. The output layer showed anticipated class probabilities for each class.

2.6 Performance Evaluation

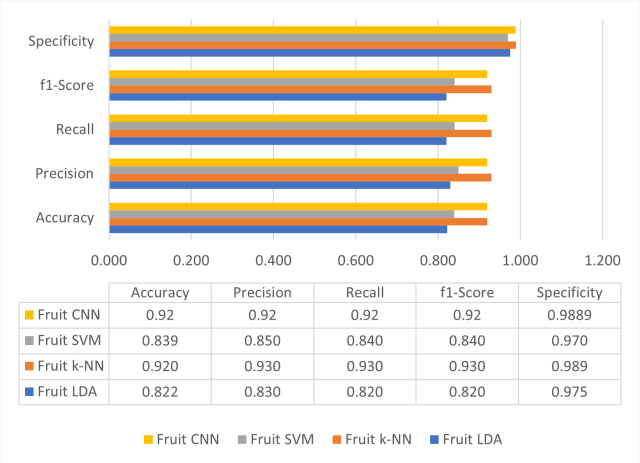
Models produced from LDA, k-NN, SVM, and CNN will be evaluated in terms of Accuracy, Precision, Recall, Specificity, and F-measure. To determine the classifier's performance, the study used k-fold cross validation coupled with kappa statistics. The results of k-fold cross validation will be subjected to confusion matrix to provide detailed picture of the performance of the classifiers.

3 PRELIMINARY RESULTS

3.1 Model-based performance

After the models were trained, they were subjected to an evaluation of their performance. The evaluation involved using the testing and validation portions of the dataset in order to determine their rate of recognition and discrimination of unknown objects. The models were evaluated based on their accuracy, precision, recall, specificity, and F1-score. To be able to determine how well the model performed, k-fold cross validation and kappa statistics performed.

Fig. 4 shows the performance of the models LDA, SVM, and k-NN using the criteria of accuracy, precision, recall, F1-score, and



	Accuracy	Precision	Recall	f1-Score	Specificity
Fruit CNN	0.92	0.92	0.92	0.92	0.9889
Fruit SVM	0.839	0.850	0.840	0.840	0.970
Fruit k-NN	0.920	0.930	0.930	0.930	0.989
Fruit LDA	0.822	0.830	0.820	0.820	0.975

Figure 4: Performance Evaluation results for LDA, k-NN, SVM and CNN in terms of Accuracy, Precision, Recall, f1-Score, and Specificity

specificity. The highest score for accuracy was gained by CNN with a score of 0.92, and LDA gained the least accuracy score of 0.822. For precision of the models, k-NN gained the highest score of 0.930, and LDA got the least score of 0.830. The same ranking and score were gained for recall and F1-Score. For specificity, the k-NN model gained the highest score of 0.989, and the least was gained by SVM with a score of 0.970.

4 FURTHER WORKS

The exhaustive analysis of the classifiers conducted in this study will continue to focus on class-based performance, in which the varieties will be evaluated using the same metrics. This is done to understand the dynamics of each fruit's visual features and the factors that influence the analysis's outcome. The classifiers will also undergo k-fold cross-validation and kappa statistics to determine their overall performance. We will implement application software to test the viability and usability of the best-performing classifier on real-world pili fruit samples.

REFERENCES

[1] Takahiro Arakawa, Kenta Iitani, Koji Toma, and Kohji Mitsubayashi. 2021. Biosensors: Gas Sensors. (2021).
[2] Dhiya Mahdi Asriny, Septia Rani, and Ahmad Fathan Hidayatullah. 2020. Orange Fruit Images Classification using Convolutional Neural Networks. In *IOP Conference Series: Materials Science and Engineering*, Vol. 803. IOP Publishing, 012020.
[3] Sumaira Ghazal, Waqar S Qureshi, Umar S Khan, Javaid Iqbal, Nasir Rashid, and Mohsin I Tiwana. 2021. Analysis of visual features and classifiers for Fruit classification problem. *Computers and Electronics in Agriculture* 187 (2021), 106267.
[4] Shima Javanmardi, Seyed-Hassan Miraei Ashtiani, Fons J Verbeek, and Alex Martynenko. 2021. Computer-vision classification of corn seed varieties using deep convolutional neural network. *Journal of Stored Products Research* 92 (2021), 101800.
[5] Lazhar Khriji, Ahmed Chiheb Ammari, and Medhat Awadalla. 2020. Artificial Intelligent Techniques for Palm Date Varieties Classification. *International Journal of Advanced Computer Science and Applications* 11, 9 (2020), 489–495.
[6] Cristopher G Millena, Bernardo A Altavano, and Rosario S Sagum. 2021. Dietary Fiber and Fermentability Characteristics of Different Pili (Canarium ovatum, Engl.) Varieties in the Philippines. *Philippine Journal of Science* 150, 4 (2021), 845–855.

[7] Juan M Ponce, Arturo Aquino, and José M Andújar. 2019. Olive-fruit variety classification by means of image processing and convolutional neural networks. *IEEE Access* 7 (2019), 147629–147641.

[8] Salman Qadri, Syed Furqan Qadri, Abdul Razzaq, Muzammil Ul Rehman, Nazir Ahmad, Syed Ali Nawaz, Najia Saher, Nadeem Akhtar, and Dost Muhammad Khan. 2021. Classification of canola seed varieties based on multi-feature analysis using computer vision approach. *International Journal of Food Properties* 24, 1 (2021), 493–504.

[9] Zhengjun Qiu, Jian Chen, Yiying Zhao, Susu Zhu, Yong He, and Chu Zhang. 2018. Variety identification of single rice seed using hyperspectral imaging combined with convolutional neural network. *Applied Sciences* 8, 2 (2018), 212.

[10] María L Rodríguez-Méndez, C Apetrei, and José A De Saja. 2010. Electronic tongues purposely designed for the organoleptic characterization of olive oils. In *Olives and olive oil in health and disease prevention*. Elsevier, 525–532.

[11] Simona Rogl and Branka Javornik. 1996. Seed protein variation for identification of common buckwheat (Fagopyrum esculentum Moench) cultivars. *Euphytica* 87, 2 (1996), 111–117.

[12] Carlo Miguel C Sandoval, Evelyn Mae Tecson-Mendoza, and Roberta N Garcia. 2017. Genetic diversity analysis and DNA fingerprinting of pili (Canarium ovatum Engl.) using microsatellite markers. *Philippine Agricultural Scientist* 100, 1 (2017).

[13] Sean Huey Tan, Chee Kiang Lam, Kamarulzaman Kamarudin, Abdul Halim Ismail, Norasmadi Abdul Rahim, Muhamad Safwan Muhamad Azmi, Wan Mohd Nooriman Wan Yahya, Goh Kheng Sneah, Moey Lip Seng, Teoh Phaik Hai, et al. 2021. Vision-Based Edge Detection System for Fruit Recognition. In *Journal of Physics: Conference Series*, Vol. 2107. IOP Publishing, 012066.

[14] Alper Taner, Yeşim Benal Öztekin, and Hüseyin Duran. 2021. Performance analysis of deep learning CNN models for variety classification in hazelnut. *Sustainability* 13, 12 (2021), 6527.

[15] Yossi Yovel, Matthias Otto Franz, Peter Stilz, and Hans-Ulrich Schnitzler. 2008. Plant classification from bat-like echolocation signals. *PLoS Computational Biology* 4, 3 (2008), e1000032.

[16] Yifan Zhou, Raphaël Maître, Mélanie Hupel, Gwenn Trotoux, Damien Penguilly, François Mariette, Lydia Bousset, Anne-Marie Chèvre, and Nicolas Parisey. 2021. An automatic non-invasive classification for plant phenotyping by MRI images: An application for quality control on cauliflower at primary meristem stage. *Computers and Electronics in Agriculture* 187 (2021), 106303.

Mendo: An AI Symptom-Based Medicine Recommender for an Over-The-Counter Drug Dispenser

Wincyr Jan P. Dela Calzada
University of the Immaculate Conception
Davao City, Philippines
wdelacalzada_200000000970@uic.edu.ph

Johann Gabriel Sebastian B. Telesforo
University of the Immaculate Conception
Davao City, Philippines
jtelesforo_2000000000102@uic.edu.ph

Clyde Chester Balaman
University of the Immaculate Conception
Davao City, Philippines
cbalaman@uic.edu.ph

Jenn Leana P. Fernandez
University of the Immaculate Conception
Davao City, Philippines
jlfernandez@uic.edu.ph

ABSTRACT

In the era of healthcare digitization, the integration of cutting-edge technologies presents an opportunity to elevate medical services and improve patient well-being. This study focuses on a comprehensive solution—the AI-powered over-the-counter (OTC) drug dispensing system—designed to enhance accessibility to affordable and essential medications in local pharmacies. The system includes an intuitive user interface, multifunctional features, and a machine dispensing mechanism, all empowered by advanced AI algorithms. This innovative approach tailors precise medication recommendations and dispenses pre-packaged doses. Continuously evolving and adaptable to changing user needs, this system has been rigorously evaluated for accuracy, showcasing its efficacy in medication decisions. Positioned to enhance healthcare outcomes and address disparities in access, the AI-enabled OTC drug dispenser signifies a new era of efficiency, personalization, and technological empowerment within community pharmacies.

KEYWORDS

Machine learning model, artificial intelligence, software management system

1 INTRODUCTION

In recent years, the healthcare sector has embraced technological advancements to enhance patient care [16]. However, traditional pharmacy stores continue to face challenges in ensuring patient safety and healthcare efficiency [2]. Medication errors, often caused by illegible prescriptions and communication breakdowns, remain a global concern [15]. In Davao City, Philippines, these challenges persist, affecting medication management and patient safety [16]. The limited accessibility of pharmacy services outside regular hours raises concerns about potential delays in accessing vital medications, particularly over-the-counter (OTC) drugs, leading to inconvenience and health risks [15].

The rapid growth of the IT industry has introduced new infrastructures and devices, offering opportunities to improve healthcare delivery [11]. Despite these innovations, drug dispensers, especially for OTC medications, are not widely available in specific areas [3]. This research focuses on implementing a recommendation system within an OTC drug dispenser to address existing gaps in healthcare technology [11, 14]. These gaps include the need for more accessible OTC medication provision, the accuracy and completeness of recommendation systems, ongoing maintenance and updates, diversity in medication options, and legal constraints on machine commercialization [1, 13, 14].

By reducing pharmacists' time spent on customer interactions, drug dispensers can enhance efficiency [4]. However, integrating recommendation systems into healthcare, especially for OTC medications, remains understudied [6]. This study aims to develop a comprehensive dataset and train a model to recommend drugs based on symptoms, illnesses, or health discomforts [7].

The objectives include gathering diverse patient data, training an accurate recommendation model, developing a user-friendly system, and evaluating system accuracy and usability.

(1) **Gather dataset** - Gather an extensive dataset with diverse patient demographics, medical histories, and treatment outcomes.
(2) **Train a model for drug recommendation** - Train a model to accurately suggest medications based on user symptoms or conditions.
(3) **Develop system** - Develop a user-friendly system integrating the trained drug recommendation model.
(4) **Evaluate the system using Intrinsic Evaluation** - Evaluate the system's accuracy, precision, recall, dispensing accuracy, machine usability, and turnaround time using Intrinsic Evaluation.

Addressing the rising demand for pharmaceutical vending machines, the Anytime Medicine Vending Machine proposes 24/7 access to OTC drugs, especially in areas with limited medical store access [13]. Its architecture, controlled by an advanced ARM processor, includes features like RFID client identification and GSM-based stock management [13]. The machine complements automated pharmacies by offering non-prescription items [8], showcasing advancements in pharmacy intelligence and predictive analysis [8].

In the Philippines, AI-enabled pharmaceutical vending machines like OTC Express are emerging to enhance medication dispensing and inventory management [5]. These systems, supported by government initiatives, promise improved patient outcomes, although ethical and regulatory considerations are crucial [5].

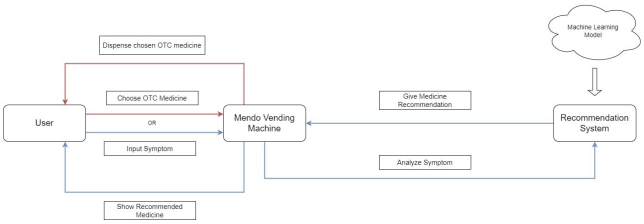


Figure 1: Conceptual Framework of Mendo Machine

Machine learning models are also employed in drug recommendation systems, leveraging datasets like the UCI repository to empower informed decision-making [10]. The investigation assesses content-based recommendation approaches, achieving a 90% accuracy rate in cloud-deployed models [10].

Advancements in administrative functions and patient management are evident, with studies showcasing reduced error rates and improved transparency in clinical scenarios [12]. Innovations in localized product recommendation systems for vending machines, as seen in Lin et al.'s work, further enhancing customer experiences and profitability [9].

2 METHODOLOGY

This section shows and discusses the procedures and methods used in this study to identify the most important characteristics and features of an AI-enabled recommender for an OTC drug dispenser.

2.1 Conceptual Framework

Figure 1 illustrates the comprehensive functionality of the Mendo drug dispenser. The process begins with user interaction, providing users with the choice to either directly purchase from the drug dispenser or opt for the machine's recommendation based on their symptoms. The purchasing option empowers users to select the over-the-counter drug they wish to acquire from the Mendo drug dispenser. On the other hand, the AI assessment option allows users to utilize the recommendation system within the Mendo drug dispenser, which suggests medicines tailored to alleviate their symptoms.

2.1.1 Data Collection. The data collection process entails systematically gathering pertinent information from diverse sources, including pharmacists and database websites such as Kaggle. The compiled data encompasses various criteria, including symptoms, over-the-counter medicines, reviews, on/off label classification, ease of use, effectiveness, and satisfaction. The researchers predominantly constructed their dataset through interviews with duly-registered and licensed pharmacists, simultaneously supplementing it by exploring existing datasets on Kaggle. Consequently, the researchers identified an extant dataset that aligns with the specified criteria and integrated the information obtained through interviews into this existing dataset.

2.1.2 Data Pre-Processing. The researchers initiated data preprocessing by cleaning and preparing the dataset for AI or Machine Learning algorithms, removing inconsistencies, errors, and redundant information. Methods employed included handling missing values using Pandas fillna() and dropna(), train-test splitting, and summarizing the preprocessed data. For data transformation, factorization, text normalization, tokenization, label encoding, and TF-IDF vectorization were utilized to enhance usability and prepare the dataset for analysis. In feature extraction, the researchers used dimensionality reduction to eliminate irrelevant features from the dataset not needed for model training, effectively modifying the dataframe.

2.2 Training the Model

The study revolves around developing an AI-enabled OTC drug dispensing machine, where users interact with the system through a user interface by inputting their specific symptoms. SVM excels in classifying suitable medicines for various health conditions by leveraging labeled data and handling high-dimensional, non-linear relationships effectively. K-Nearest Neighbors (KNN), a machine learning algorithm classifying data points based on proximity, contributes to accurate medication recommendations by considering user input and aligning it with appropriate medications in the training dataset. Together, SVM and KNN offer a comprehensive approach, ensuring user safety and satisfaction in improving the OTC drug dispenser's recommendation capabilities.

2.3 System Development

The development of the Mendo Recommendation System involved a streamlined approach to system architecture. The Front-End was crafted using HTML and Tailwind Flowbite for CSS styling, ensuring a visually appealing and user-friendly interface. Python Flask served as the Back-End framework providing robust functionality, it is also where the researchers integrated the AI Model. The management system, catering to admin tasks such as monitoring dispensed medicines, tracking remaining quantities, and editing prices, was implemented using PHP and MySQL for efficient data management. This cohesive integration of technologies ensures a seamless and effective user experience for both customers and administrators.

2.4 Evaluation

Intrinsic evaluation plays a pivotal role in assessing the trained model and system, with a primary focus on accuracy, precision, and recall for the AI model. Additionally, the evaluation extends to the system's dispensing accuracy, machine usability, and turnaround time, providing a comprehensive analysis of both the model's predictive capabilities and the operational efficiency of the dispensing system.

3 PRELIMINARY RESULTS

In order to provide initial insights on the usability of ErrgoEngine, the following objectives were established:

3.1 Machine Learning Model Results

In Table 1 machine learning model, the researchers opted for a "linear" kernel configuration. The test data portion constituted 20% of the total, utilizing a random state parameter set at 47 for consistent data splitting. These specific settings resulted in an accuracy and recall rate of 40.79%, coupled with a precision rate of 34.20%. It's

Kernel	Accuracy	Precision	Recall
Linear	40.79%	34.20%	40.79%
Sigmoid	40.79%	34.20%	40.79%
Gaussian	40.79%	34.20%	40.79%
Poly	40.79%	34.20%	40.79%

Table 1: SVM Kernel Results

noteworthy that the dataset predominantly consisted of prescription (RX) drugs, with only a limited selection of over-the-counter (OTC) medications available for analysis.

n_neighbors	Accuracy	Precision	Recall
3	40.79%	30.04%	40.79%
5	31.58%	31.25%	31.58%
7	30.26%	34.24%	30.26%
10	30.26%	29.28%	30.26%

Table 2: kNN Kernel Results

In Table 2, it becomes clear that an escalation in the number of neighbors (n_neighbors) aligns with improvements in accuracy, precision, and recall metrics for the k-Nearest Neighbors (KNN) model, maintaining a constant random state of 47 and a test size of 0.2. However, a considerable decline in these performance measures is noted upon reaching 13 neighbors. Consequently, within the scope of this machine learning model, an optimal count of n_neighbors is determined to be 10, underscoring the importance of parameter tuning in optimizing the model's predictive performance.

3.2 System Results

Inputs	Machine Learning Model (SVM)
fever	Paracetamol (Biogesic)
headache	Ibuprofen (Medicol)
cough with phlegm	Carbocisteine (Solmux)
cough without phlegm	Butamirate (Sinecod)
cold	Phenylephrine HCL (Neozep)

Table 3: OTC Medicine Recommendation Results

Table 3, presenting the OTC Medicine Recommendation Results, reflects the recommendation accuracy of the Mendo system in providing appropriate medication recommendations based on inputted symptoms/conditions. The correlation between symptoms and recommended medicines—such as Paracetamol for fever, Ibuprofen for headache, Carbocisteine for cough with phlegm, Butamirate for cough without phlegm, and Phenylephrine HCL for cold—is consistent with both the dataset and pharmacist validations. These findings underscore the reliability of the recommendation system in accurately identifying and addressing common health conditions, enhancing its utility in providing accessible and effective over-the-counter medication guidance.

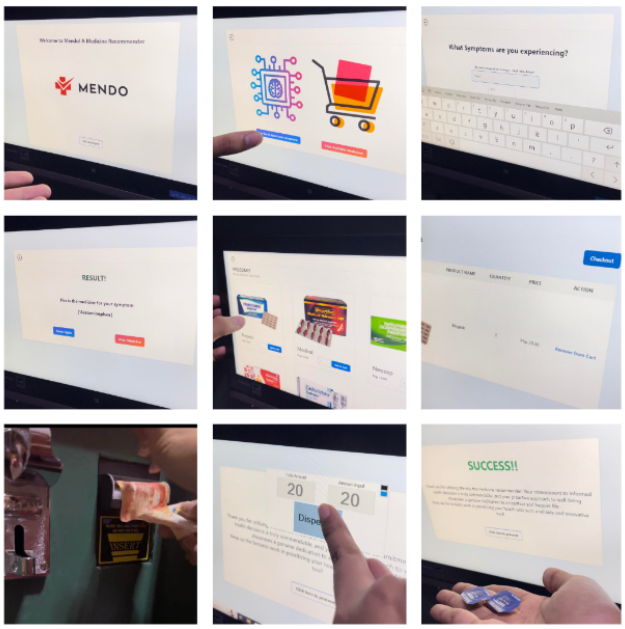


Figure 2: The whole process of interacting and ordering medicine with Mendo

The Mendo Machine evaluation covered dispensing accuracy, system usability, and turnaround time. Accuracy was ensured by comparing checked-out and dispensed medicines. Mendo's user-friendly interface streamlines transactions, requiring upfront payment and updating the admin dashboard for inventory management. Figure 2 shows the user's interaction with Mendo from the AI recommender, confirming Paracetamol for fever, to the dispensing of the actual medicine. Turnaround time analysis revealed efficient processing, ranging from one minute and thirty seconds to two minutes, highlighting Mendo's ability to improve user experience and prompt medication access in healthcare settings.

4 FURTHER WORKS

This section discusses the works to be included in this study.

4.1 Gathering More Data for the Dataset

For future work, gathering additional data, specifically focusing on Over-the-Counter (OTC) drugs, is imperative as the current dataset contains only 756 OTC data points, which may not be sufficient for implementing deep learning models effectively. Increasing the dataset size will enhance the model's accuracy and robustness, enabling more comprehensive analysis and reliable recommendations for OTC medications within the Mendo system.

4.2 Implementation of Deep Learning

After gathering sufficient data, the researchers intend to incorporate advanced deep learning techniques, focusing specifically on Natural Language Processing (NLP) and Named Entity Recognition (NER), into the AI-enabled OTC drug dispensing machine. This integration will enhance the AI's capabilities by enabling it

to extract symptoms like "fever" and "cough" from user inputs for accurate medication recommendations using NLP. Additionally, NER will identify and extract symptom details, medication names, current regimens, and allergies, ensuring personalized and precise non-prescription medication recommendations while considering potential drug interactions and allergic reactions, thus significantly enhancing user safety and efficacy.

The researchers also aim to enhance the AI's capabilities by integrating Natural Language Processing (NLP) into the system not just to detect input errors but to understand diverse languages including Tagalog or Cebuano (Bisaya), as suggested by professional pharmacists. This adaptation is crucial for ensuring seamless communication and user interaction, particularly in culturally diverse settings like the Philippines, where users may prefer to interact in their native languages. Incorporating NLP will not only improve the system's accuracy but also contribute to a more inclusive and user-friendly experience for all potential users.

REFERENCES

[1] 2016. REPUBLIC ACT NO. 10918 - AN ACT REGULATING AND MODERNIZING THE PRACTICE OF PHARMACY IN THE PHILIPPINES, REPEALING FOR THE PURPOSE REPUBLIC ACT NUMBERED FIVE THOUSAND NINE HUNDRED TWENTY-ONE (R.A. NO. 5921), OTHERWISE KNOWN AS THE PHARMACY LAW - Supreme Court E-Library. (2016). https://elibrary.judiciary.gov.ph/thebookshelf/showdocs/2/70354#:~:text=a%20professional%20pharmacist.-,SEC.,duly%20licensed%20by%20the%20FDA

[2] Yasser K Alotaibi and Frank Federico. 2017. The impact of health information technology on patient safety. *Saudi medical journal* 38, 12 (2017), 1173.

[3] Aries Buenaventura, Sunday Marc Joseph Padua, Lester Sarmiento, Engr Jose Marie, B Dipay, and Ryan S Evangelista. 2022. Vendi Medic: A Geographic Information System Based Non-Prescription Medicine Vending Machine. *Journal of Optoelectronics Laser* 41, 8 (2022), 2022.

[4] Chegg. 2022. Solved Big-Box Pharmacy operates out of the back of Big-Box. https://www.chegg.com/homework-help/questions-and-answers/big-box-pharmacy-operates-back-big-box-retailer-caters-customers-typical-pharmacy-scripts--q101508621/

[5] Fred Dabu. 2021. Tech, plant-based products shown at NIH Conference. https://up.edu.ph/tech-plant-based-products-shown-at-nih-conference/

[6] Philstar Global. 2022. Government limits purchases of fever and flu meds as retailers run out of stock. https://www.philstar.com/headlines/2022/01/11/2153203/government-limits-purchases-fever-and-flu-meds-retailers-run-out-stock

[7] Aineena Hani, Yen Ocampo, Samaya Dharmaraj, and Eka Santhika. 2021. More AI Being Deployed in the Philippine Healthcare Sector. https://opengovasia.com/more-ai-being-deployed-in-the-philippine-healthcare-sector/

[8] John Van Horn and Saravana. 2022. Pharmaceutical vending machine blog: Medicine Vending Machines. https://www.idsvending.com/blog/category/pharmaceutical-vending-machine/

[9] Feng-Cheng Lin, Hsin-Wen Yu, Chih-Hao Hsu, and Tzu-Chun Weng. 2011. Recommendation system for localized products in vending machines. *Expert systems with applications* 38, 8 (2011), 9129–9138.

[10] Utkarsh Mathur. 2022. *A Content Based Recommender System for Medicine using Machine Learning Algorithm.* Ph. D. Dissertation. Dublin, National College of Ireland.

[11] Solomon Negash, Philip Musa, Doug Vogel, and Sundeep Sahay. 2018. Healthcare information technology for development: improvements in people's lives through innovations in the uses of technologies. , 189–197 pages.

[12] Juan G Diaz Ochoa, Orsolya Csiszár, and Thomas Schimper. 2021. Medical recommender systems based on continuous-valued logic and multi-criteria decision operators, using interpretable neural networks. *BMC medical informatics and decision making* 21 (2021), 1–15.

[13] K N Raviraja, R Pavan, P Sanjay, D Srikanth, and T Kiran Kumar. 2019. Anytime medicine vending machine. https://zenodo.org/record/2667053?fbclid=IwAR3D09C04vKD--VLPmjYclWAs2Q91trFPO0hmG_BQChQinhKAYq9G2QMQz8#.ZEaoi3YRWUl

[14] Benjamin Stark, Constanze Knahl, Mert Aydin, and Karim Elish. 2019. A literature review on medicine recommender systems. *International journal of advanced computer science and applications* 10, 8 (2019).

[15] Rayhan A Tariq, Rishik Vashisht, Ankur Sinha, and Yevgeniya Scherbak. 2023. Medication dispensing errors and prevention. https://www.ncbi.nlm.nih.gov/books/NBK519065/

[16] Harold Thimbleby. 2013. Technology and the future of healthcare. *Journal of public health research* 2, 3 (2013), jphr–2013.

Well-Being Assessment Using ChatGPT-4: A Zero-Shot Learning Approach

Julianne Vizmanos
De La Salle University
Manila, Philippines
julianne_vizmanos@dlsu.edu.ph

Ethel Ong
De La Salle University
Manila, Philippines
ethel.ong@dlsu.edu.ph

Jackylyn Beredo
De La Salle University
Manila, Philippines
jackylyn.beredo@dlsu.edu.ph

Remedios Moog
De La Salle University
Manila, Philippines
remedios.moog@dlsu.edu.ph

ABSTRACT

Traditional approaches of using self-report questionnaires and emotion-based lexicon pose limitations in assessing the well-being states from dialogue utterances which consequently impact the generation of appropriate empathetic responses. The development of ChatGPT unveiled the potential of applying large language models in various domain of text understanding tasks, including well-being assessment. In this paper, we present our investigation of using ChatGPT-4 to measure well-being based on Seligman's PERMA model. The International Survey on Emotion Antecedents and Reactions (ISEAR) dataset was manually annotated with the elements of PERMA. Zero-shot ChatGPT-4 is then employed to label the dataset across five well-being states: *excelling, thriving, surviving, struggling,* and *in crisis*. In the absence of a reference gold standard to serve as baseline, we compared our results with those produced from using the PERMA Lexicon. Because there is ambiguity in the boundary between neighboring well-being states, we reduced the labels to three with *excelling, thriving and surviving* comprising one state, while *struggling* and *in crisis* remain as separate states. Results from applying the intercoder agreement yielded 37.22%, 39.61%, and 50.11%, respectively. Our findings highlight the challenges of automating this inherently subjective task without a PERMA-labeled dataset to serve as a basis for ground truth and lays the groundwork in employing ChatGPT-4 for psychological well-being assessment.

KEYWORDS

Well-being assessment, PERMA model, PERMA lexicon, ChatGPT, zero-shot learning

1 INTRODUCTION

Psychological well-being plays an important role in a person's mental health which can impact our emotions, relationships, productivity, and overall satisfaction with life. Seligman's PERMA model [15] is one of the most influential models in psychological well-being that measures its five core elements, namely positive emotion (P), engagement (E), relationships (R), meaning (M), and accomplishments (A). Enhancing each of these five (5) elements can enable an individual to achieve a more fulfilled and satisfied life.

Psychological well-being assessment is usually measured using self-report questionnaires such as the PERMA Profiler [3]. This instrument contains 15 questions that covers the five (5) elements of PERMA and 8 questions that focus on overall health, negative emotion, happiness, and loneliness. Prior studies have reported its reliability in measuring well-being [4, 7]. However, these tools are resource-intensive and pose challenges in scalability.

With the availability of conversational agents or chatbots, researchers began exploring the use of these technologies in delivering mental health and well-being support services [6, 8–10]. Most of these chatbots are designed primarily to generate appropriate empathetic responses, but very few works have focused on measuring the support seeker's well-being. The PERMA Lexicon is an attempt to automate well-being measurement [2, 14] but faced some shortcomings. These lexicon-based approaches heavily rely on a pre-defined set of words with their corresponding PERMA score and may lead to inaccuracies if a word to be processed is not found in the dictionary. Moreover, lexicons are limited with their inability to understand context and handle figurative languages such as irony and sarcasm [1].

The emergence of large language models (LLMs) such as GPT-3.5, GPT-4 [11], and LLaMa [16] offered potential benefits in healthcare applications through the generation of human-like responses that are coherent and textually relevant to the user's prompts pose. Developed by OpenAI, ChatGPT is trained on GPT-3.5 using the Reinforcement Learning from Human Feedback (RLHF) technique to align its responses to humans [12]. Research work are also exploring its application in sentiment analysis [1, 18, 19] and emotion detection [17]. To perform such NLP tasks, models are trained with annotated datasets. The release of ChatGPT has opened a new area of research that focuses on employing these models in zero-shot learning - that is, without additional task-specific training.

In this paper, we describe our investigation of using ChatGPT-4 in measuring well-being based on Seligman's PERMA Model [15]. We manually annotated the International Survey on Emotion Antecedents and Reactions (ISEAR) dataset [13] with the five (5) elements of PERMA. Zero-shot ChatGPT-4 is then employed to label the dataset across five well-being states: *excelling, thriving, surviving, struggling,* and *in crisis*. Our main contribution in this paper is to provide an evaluation of ChatGPT-4's performance in well-being assessment by comparing it with the results of the PERMA Lexicon [2]. Our findings highlight the challenges of automating an inherently subjective task without a PERMA-labeled dataset to serve as a basis for ground truth, and lays the groundwork in employing ChatGPT-4 for psychological well-being assessment.

2 TASK DESCRIPTION

In this section, we provide a computational definition of the well-being assessment task, followed by the process of computing the PERMA scores using the PERMA Lexicon. We include a discussion of how we formulated the prompts to instruct ChatGPT to perform the required task.

2.1 Well-being Assessment

We formulate the well-being assessment task as a text classification problem similar to the approach by MHBot and VHope [2, 10]. Given an input utterance **u**, the well-being assessment annotates the utterance **u** to one of the predefined set **L** of well-being states where **L = {excelling, thriving, surviving, struggling, in crisis}**. Thus, the well-being assessment is a function $f : U \rightarrow L$ that annotates each utterance $u \in U$ with a label $l \in L$ that best represents the well-being state of the utterance, where **U** is all the input utterances and **L** is all the possible well-being states.

2.2 PERMA Lexicon

The PERMA Lexicon is a collection of words with positive and negative scores representing the elements of PERMA. This dataset was used by the MHBot [10] and the VHope [2] chatbots to measure the well-being of an individual based on the lexical choices found in their messages, social media posts, or utterances. The formula shown in Equation 1 computes the total positive and negative PERMA score of a given input text.

Given an input text or utterance $\boldsymbol{u}$, the average *PERMA weight* is first computed from the sum of the *PERMA_weight(w_i)* for each token w_i in $\boldsymbol{u}$ divided by the total number of tokens $\boldsymbol{n}$. For each PERMA element, denoted by category $\boldsymbol{c}$, the lowest score min_c is subtracted from the resulting average *PERMA_weight* and then multiplied by 10, which corresponds to the total number of categories - 5 positive and 5 negative PERMA elements. The product is then divided with the difference of the category's max_c and min_c values as indicated in Table 1. From this process, each of the 10 categories yields a score that is totalled by group (*pos* and *neg*), leading to the final positive and negative PERMA score.

$$PERMA\_score(pos, neg) = \sum_{c=p}^{a} \frac{\left(\frac{\sum_{i=1}^{n} PERMA\_weight(w_i)}{n} - \min_c\right) * 10}{max_c - min_c} \tag{1}$$

where,

$PERMA\_score$ = total well-being score
pos = positive score
neg = negative score
w = tokens in the input
c = p, e, r, m, a (positive and negative)
n = total number of tokens in the input
min = minimum score of the category
max = maximum score of the category

The *PERMA_score* is then interpreted using the labels from the PERMA Profiler [3]: *very high functioning*, *high functioning*, *normal functioning*, *sub-optimal functioning*, and *languishing*. These labels may be vague to users of VHope, thus, they were renamed following the Mental Health Continuum [5] phases to support the belief that an individual's mental health is not binary but a continuously changing state. The revised labels are indicated in Table 2.

Table 1: Minimum and maximum scores of each Category representing the positive and negative PERMA elements

	Minimum Score	Maximum Score	Mean
POS_P	-0.36639	0.76549	0.04172
POS_E	-0.30074	0.34065	0.03234
POS_R	-0.28884	0.78376	0.03824
POS_M	-0.16748	0.77167	0.02517
POS_A	-0.19784	0.55031	0.03990
NEG_P	-0.32731	0.70697	0.04705
NEG_E	-0.15230	0.84017	0.04354
NEG_R	-0.28648	0.62033	0.04045
NEG_M	-0.14987	0.31674	0.03416
NEG_A	-0.15369	0.24760	0.03426

Table 2: PERMA Score Interpreter

Label	Positive Score	Negative Score
Excelling	7 and above	0 to 1
Thriving	6 to 6.9	1.1 to 2.5
Surviving	4.5 to 5.9	2.6 to 3.9
Struggling	3 to 4.4	4 to 4.9
In crisis	below 3	above 5

Initial testing of VHope [2] using the interpretation suggested by [3] showed inaccuracy in labeling the utterances based on the computed well-being score. The values were adjusted as shown in Table 2 and used during VHope's user testing phase. The accuracy was again re-assessed by comparing the user's computed well-being score from Equation 1 and their score derived manually from the self-report questionnaire. Out of the 21 well-being levels, the computed and the manually derived well-being scores agree on 12 well-being levels, or 57% accuracy.

Guidance counselors also reviewed the PERMA labels assigned to utterances. Of the 97 PERMA labels extracted from 43 conversation logs, only 57 labels or 59% were noted as appropriate. But even with the low accuracy, the counselors noted that the PERMA labels assigned by VHope were able to dynamically adapt to the user's changing well-being state throughout a conversation. This makes the PERMA Lexicon a sufficient basis for comparing the well-being assessment generated by ChatGPT.

2.3 Prompt Formulation

We adapted the prescribed prompt formulation template defined by OpenAI[1] for the well-being assessment task to indicate the *role* to be portrayed by the LLM, the *task* or instruction to be performed, the *input* text, and the target *labels*.

[1]https://platform.openai.com/docs/guides/prompt-engineering/six-strategies-for-getting-better-results

Following the prompt engineering strategies, the formulated prompt designates a role for the LLM:

> *You are a PERMA expert.*

specifies the task to be performed:

> *Your task is to categorize the sentences into the five zones of The Mental Health Continuum by Delphis: Excelling, Thriving, Surviving, Struggling, and In Crisis.*

and defines the target labels from [5]. The desired length of the output is also specified in the prompt:

> *Given the explanation for each category, output the category only. Number each row, but NO explanation.*

3 METHOD

We cleaned the International Survey on Emotion Antecedents and Reactions (ISEAR) dataset by removing special characters and duplicate entries. Nuisance rows such as those containing variants of *no*, *not applicable*, *no description*, and *nothing* were also removed. The cleaned dataset yielded 7,475 rows. The emotional responses recorded in the *content* column of the data is utilized in this study.

The PERMA Lexicon is employed to annotate the ISEAR dataset with the labels: *excelling*, *thriving*, *surviving*, *struggling*, and *in crisis*. Each entry as tokenized to derive the corresponding PERMA weights. Tokens without associated weights in the lexicon are assigned with a value of zero. The overall PERMA well-being score is then computed for each entry and assigned the corresponding label indicated in Table 2.

Zero-shot ChatGPT-4 is employed to annotate the ISEAR dataset. In the zero-shot setting, the pre-trained model is not provided with any additional task-specific training. There was a noticeable significant variance in the execution time between the web interface and API of ChatGPT-4. Because of this, the web interface of zero-shot ChatGPT-4 is employed to annotate the ISEAR dataset.

Without a reference gold standard to serve as the ground truth, we utilized the Intercoder Agreement to measure the consistency of annotation labels between the PERMA Lexicon and ChatGPT-4. We deemed it inappropriate to use the annotations derived from the PERMA Lexicon as the ground truth since VHope reported achieving only 57% accuracy when using this approach [2].

4 PRELIMINARY RESULTS

In consultation with a guidance counselor specializing in PERMA, we performed three (3) comparative analyses of our results by clustering the PERMA labels as shown in Table 3. The **5-Label Analysis** compares the performance of ChatGPT with that of the PERMA Lexicon by looking at each of the labels independently. The **4-Label Analysis** explores the influence of aggregating the neighboring labels *excelling* and *thriving* to the performance of the models in well-being assessment. Lastly, the **3-Label Analysis** extends the aggregation of neighboring labels to include *surviving* with *excelling* and *thriving*. Because well-being assessment is highly subjective, our approach addresses the ambiguity in the boundary between neighboring well-being states.

Table 4 presents the annotation results derived using the PERMA Lexicon and ChatGPT-4. The matrix highlights the common annotation labels along the main diagonal line. This serves as the basis for comparing the performance of the two approaches using the 5-Label, 4-Label, and 3-Label Analyses.

Table 3: Comparative Analyses

	Labels				
5-Label	excelling	thriving	surviving	struggling	in crisis
4-Label	excelling + thriving		surviving	struggling	in crisis
3-Label	excelling + thriving + surviving			struggling	in crisis

4.1 5-Label Analysis

As shown in Figure 1, *Struggling* has the highest percent agreement at 43.99%, followed by *surviving* at 41.47% and *excelling* at 34.07%. The *thriving* and *in-crisis* labels have the lowest, at 12.77% and 1.67% agreement, respectively. Overall, there is a 37.22% intercoder agreement in the 5-label analysis.

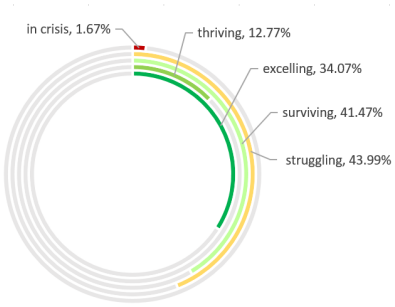


Figure 1: Intercoder Agreement for each well-being label in the 5-Label Analysis.

Surviving and *struggling* have the highest contribution to the overall intercoder agreement at 55.10% and 39.22%, respectively. These values suggest that the labels are not only prevalent in the dataset but that both the PERMA Lexicon and ChatGPT-4 performed well in labeling instances of these states. In the case of *excelling*, although the intercoder agreement is significant, its contribution to the overall percentage agreement is only 2.23%, indicating that both approaches tend to agree in labeling this state despite its low occurrence in the dataset.

The high instances of intercoder disagreements in general, and for the labels *excelling*, *thriving*, and *surviving* in particular, can be attributed to the subjective nature of well-being assessments and the fuzzy boundary between neighboring labels. Looking at Table 4,

Table 4: Annotation matrix for the Well-being labels derived using the PERMA Lexicon and ChatGPT-4.

		ChatGPT-4				
		excelling	thriving	surviving	struggling	in crisis
Lexicon	excelling	62	54	33	32	1
	thriving	125	89	230	232	21
	surviving	258	264	1533	1519	123
	struggling	83	156	1052	1091	98
	in crisis	7	24	205	176	7

there is a 63.20% disagreement rate between the neighboring labels. The occurrence of classification biases is similar to how different psychologists may give different labels to an individual's well-being due to how they interpret a situation.

It is observed that intercoder disagreements typically occur in utterances with contradicting statements. The utterance *"I had lied to a person because I had thought that I could not tell him the truth. When he found out he was not angry but understanding. We talked the whole thing over"* presents a situation wherein an individual lied to a person because he thought that the truth cannot be shared. The person did not explode in fits of anger upon discovery of the truth, but was rather understanding in discussing and resolving this issue. This complex narrative is a challenge for the labeling task as the initial deceit could suggest a *crisis* while the resolution leans towards *survival*. As such, an utterance encompasses a wide range of emotions and states of well-being - ultimately relying on the coders to decide on a label that best represents the overall state of well-being.

4.2 The 4-Label Analysis

As observed in Figure 2, *Struggling* still has the highest agreement rate at 43.99%, followed by *surviving* at 41.47%. Meanwhile, the combined *excelling* and *thriving* label has achieved a 37.54% agreement, and *in crisis* label remained the lowest at 1.67%. Overall, there is a 39.61% intercoder agreement for the 4-label analysis.

Surviving and *struggling* have a significant impact to the overall intercoder agreement at 51.77% and 36.85%, respectively. This suggests a strong consensus between PERMA Lexicon and ChatGPT-4 in assigning these labels because of clearer distinction between these well-being states. The consolidation of *excelling* and *thriving* achieved 37.54% intercoder agreement and contributed 11.14% to the overall agreement rate, implying that closely-related positive states may be easily agreed upon by both annotators as opposed to its finer states. Its consolidation also lowered the disagreement rate between the neighboring labels to 59.29%.

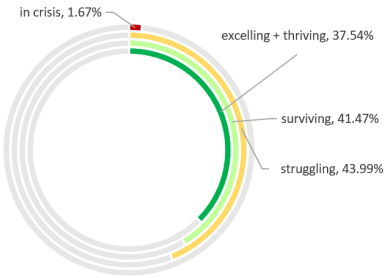


Figure 2: Intercoder Agreement for each well-being label in the 4-Label Analysis

4.3 The 3-Label Analysis

The aggregation of *excelling*, *thriving*, and *surviving* label achieved the highest percent agreement at 57.87%. This is followed by *struggling* at 43.99%. *In crisis* remains at 1.67% as exemplified in Figure 3. Overall, there is a 50.11% agreement rate for the 3-label analysis.

The combination of *excelling*, *thriving*, and *surviving* label recorded a 57.87% intercoder agreement and significantly influenced the overall agreement at 70.69%. This could be attributed to the more evident boundaries between the labels and reinforces that positive states of well-being is distinct as opposed to its negative counterpart. However, the high disagreement rate for *in crisis* at 98.33% suggests that the PERMA Lexicon and the language model might need further refinement, or the threshold employed by the PERMA Lexicon should be re-examined.

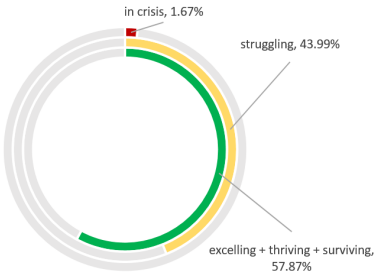


Figure 3: Intercoder Agreement for each well-being label in the 3-Label Analysis.

4.4 Overall Analysis

The 5-Label Analysis reported the lowest intercoder agreement at 37.22%. However, an increase of 2.39% is observed for the 4-Label Analysis, resulting to 39.61%. Ultimately, the 3-Label Analysis yielded the highest intercoder agreement of 50.11% with a significant boost of 12.89%. This suggests that the subtle nature of well-being states are more challenging to classify because of their overlapping characteristics, but the aggregation of labels exemplified that the states of well-being are more distinct when considered in a wider scope.

Collectively, the analyses highlighted the complexity in classifying the continuous nature of well-being - reinforcing its subjectivity. As such, it is observed that classifying general well-being states gained significant results as opposed to distinguishing between its varying levels. This alludes to the blurred boundaries between its varying levels as opposed to its clearer limits when consolidated with states sharing similar characteristics. It also implies that the PERMA Lexicon or the ChatGPT-4 model may need further development, and the threshold for PERMA Lexicon's classification may need to be re-assessed.

5 FURTHER WORK

The insights gained from this study can contribute to the expanding knowledge of psychology in Natural Language Processing (NLP); as well as contribute to the exploration of automated well-being assessment tools through LLMs. Future works should examine ChatGPT-4 in one-shot and few-shot settings to determine the influence of the setting on the intercoder agreement rate. Balancing of the distribution of instances for each well-being state may also improve the performance of LLMs. Alternative metrics that consider the subjective nature of well-being assessment can also be explored to provide additional insights to the model's efficacy in this classification task.

REFERENCES

[1] Mohammad Belal, James She, and Simon Wong. 2023. Leveraging ChatGPT As Text Annotation Tool For Sentiment Analysis. arXiv:2306.17177 [cs.CL]

[2] Jackylyn L. Beredo and Ethel Ong. 2022. Analyzing the Capabilities of a Hybrid Response Generation Model for an Empathetic Conversational Agent. *International Journal of Asian Language Processing* 32, 4 (2022). https://doi.org/10.1142/S271755452350008X

[3] Julie Butler and Margaret L. Kern. 2016. The PERMA-Profiler: A Brief Multidimensional Measure of Flourishing. *International Journal of Wellbeing* 6, 3 (2016).

[4] Thainá Ferraz de Carvalho, Sibele Dias de Aquino, and Jean Carlo Natividade. 2021. Flourishing in the Brazilian Context: Evidence of the Validity of the PERMA-Profiler Scale. *Current Psychology* 42 (2021), 1828—-1840.

[5] Delphis. 2020. The Mental Health Continuum is a Better Model for Mental Health. https://delphis.org.uk/mental-health/continuum-mental-health/.

[6] Vanshika Gupta, Varun Joshi, Akshat Jain, and Inakshi Garg. 2023. Chatbot for Mental health support using NLP. In *Proceedings of the 2023 4th International Conference for Emerging Technology (INCET)*. https://doi.org/10.1109/INCET57972.2023.10170573

[7] Margaret L. Kern, Lea E. Waters, Alejandro Adler, and Mathew A. White. 2015. A Multidimensional Approach to Measuring Well-being in Students: Application of the PERMA Framework. *Journal of Positive Psychology* 10, 3 (2015), 262–271.

[8] Junhan Kim, Yoojung Kim, Byungjoon Kim, Sukyung Yun, Minjoon Kim, and Joongseek Lee. 2018. Can a Machine Tend to Teenagers' Emotional Needs? A Study with Conversational Agents. In *Proceedings (ACM) Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems (CHI EA '18)*. Montreal, Canada, 1–6.

[9] Yi-Chieh Lee, Naomi Yamashita, Yun Huang, and Wai Fu. 2020. "I Hear You, I Feel You": Encouraging Deep Self-disclosure Through a Chatbot. In *Proc, (ACM) 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Hawaii, USA, 1–12.

[10] Ethel Ong, Melody Joy Go, Rebecalyn Lao, Jaime Pastor, and Lenard Balwin To. 2024. Investigating Shared Storytelling with a Chatbot as an Approach in Assessing and Maintaining Positive Mental Well-Being among Students. *International Journal of Asian Language Processing* 33, 3 (2024). https://doi.org/10.1142/S2717554523500170

[11] OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL]

[12] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. 2022. Training Language Models to Follow Instructions with Human Feedback. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.

[13] Klaus R. Scherer and Harald G. Wallbott. 1994. Evidence for Universality and Cultural Variation of Differential Emotion Response Patterning. *Journal of Personality and Social Psychology* 66, 2 (1994), 310.

[14] H. Andrew Schwartz, Maarten Sap, Margaret L. Kern, Johannes C. Eichstaedt, Adam Kapelner, Megha Agrawal, Eduardo Blanco, Lukasz Dziurzynski, Gregory Park, David Stillwell, Michal Kosinski, Martin E.P. Seligman, and Lyle H. Ungar. 2016. Predicting Individual Well-being through the Language of Social Media. In *Biocomputing 2016: Proceedings of the pacific symposium*. World Scientific, 516–527.

[15] Martin Seligman. 2010. Flourish: Positive Psychology and Positive Interventions. *The Tanner Lectures on Human Values* 31, 4 (2010), 1–56.

[16] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL]

[17] Naoki Wake, Atsushi Kanehira, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. 2023. Bias in Emotion Recognition with ChatGPT. arXiv:2310.11753 [cs.RO]

[18] Zengzhi Wang, Qiming Xie, Yi Feng, Zixiang Ding, Zinong Yang, and Rui Xia. 2024. Is ChatGPT a Good Sentiment Analyzer? A Preliminary Study. arXiv:2304.04339 [cs.CL]

[19] Yiming Zhu, Peixian Zhang, Ehsan-Ul Haq, Pan Hui, and Gareth Tyson. 2023. Can ChatGPT Reproduce Human-Generated Labels? A Study of Social Computing Tasks. arXiv:2304.10145 [cs.AI]

ConcreteGuard: A YOLOv8-based Web Application for Early Detection of Concrete Cracks

Mary Josephine R. Gamit
De La Salle University - Dasmariñas
Dasmariñas, Cavite
gmr0078@dlsud.edu.ph

Miguel Antonio D. Chavez
De La Salle University - Dasmariñas
Dasmariñas, Cavite
cmd0935@dlsud.edu.ph

John Carlo M. Gayoso
De La Salle University - Dasmariñas
Dasmariñas, Cavite
gjm0934@dlsud.edu.ph

Sheryl D. Kamantigue
De La Salle University - Dasmariñas
Dasmariñas, Cavite
sdkamantigue@dlsud.edu.ph

ABSTRACT

Concrete, a widely used building material, plays a crucial role in global construction. Detecting and repairing concrete damage is vital to maintain structural integrity and safety. Current detection methods involve manual inspection and non-destructive testing, which can be time-consuming and labor-intensive. To address these challenges, this project introduces a web application utilizing Computer Vision and Machine Learning to aid civil engineers in concrete crack detection and mapping.

ConcreteGuard employs the YOLOv8 model, trained on a diverse dataset with data augmentation. All built-in hyperparameters provided by YOLOv8 were used in their default setting. Evaluation metrics demonstrate a precision of 0.882, recall of 0.772, mAP50 of 0.868, and an F1 score of 0.823. OpenCV is used for contour detection and distance transform for crack width measurement, achieving an average difference of 0.765 mm. Challenges persist for cracks less than 2 millimeters wide.

Additionally, the system facilitates crack mapping in construction, offering a systematic approach to identify, locate, and document cracks in structures. In situations where rapid preliminary assessments are crucial, such as seismic-prone regions, ConcreteGuard's crack mapping functionality can prove indispensable for professionals in civil engineering and construction.

KEYWORDS

Crack mapping, YOLOv8, web application

1 INTRODUCTION

Concrete is a fundamental material in the construction industry due to its versatility and widespread use. Comprising cement, water, and aggregates like sand, gravel, and crushed stone, it forms the backbone of many structures. The global cement market was valued at USD 326.81 billion in 2021, with projections indicating growth to USD 481.73 billion by 2029 [10]. Despite a temporary decline of 3.6% in 2020 due to the COVID-19 pandemic, the market is rebounding, fueled by increased demand for residential and public infrastructure projects, including hospitals and healthcare centers.

However, despite its durability, concrete is susceptible to deterioration from various factors such as temperature changes, moisture, chemical reactions, and structural [1]. Detecting and addressing these issues promptly is crucial for maintaining structural integrity and ensuring the safety of building occupants.

Traditionally, professionals rely on manual visual inspections and non-destructive testing methods like ultrasound and ground-penetrating radar to identify concrete damage [11]. One visual concrete inspection method used is crack mapping. It allows engineers to identify the extent and severity of cracks within a structure [3]. By systematically documenting the location and size of cracks, engineers can assess the structural health and prioritize repairs accordingly. This information is crucial for preventive maintenance and ensuring the long-term durability of concrete infrastructure. While effective, this method is labor-intensive, time-consuming, and requires specialized equipment and expertise, adding complexity and cost to the inspection process.

In recent years, computer vision and Convolutional Neural Networks (CNNs) have emerged as promising tools for concrete crack detection. Unlike traditional methods, CNNs can automatically extract features from images without manual intervention, leading to more accurate and efficient detection [13].

Additionally, OpenCV, short for Open-Source Computer Vision Library, is a free and powerful software library for computer vision tasks and is used in this study [8].

Past research has demonstrated the efficacy of employing computer vision and deep learning techniques, such as convolutional neural networks (CNN), for concrete crack detection. Despite their success, these methods often grapple with small datasets, limiting their generalizability. This study builds upon this foundation by leveraging CNNs to develop a concrete crack detection web application, with an edge lying in its ability to generate comprehensive crack mapping reports. While previous approaches have achieved accuracies ranging from 84% to significant improvements in crack quantification, the focus of this study extends to practical application through an accessible digital platform, promising a new dimension in crack detection methodology [13, 2].

This study aims to develop "ConcreteGuard," a web application designed to enhance the accessibility and user-friendliness of concrete crack detection and documentation. ConcreteGuard facilitates the detection, classification, and measurement of cracks in concrete structures, allowing users to capture images and process them using the CNN model. Additionally, the application generates detailed reports to aid in comprehensive evaluation.

To achieve these objectives, the study utilized a dataset of images from Roboflow, a framework for creating computer vision models without hand-labeling images, supplemented by images captured by researchers from four public schools in Dasmarinas [7]. The dataset was used for training using the YOLOv8 model for crack detection and measurement, which was integrated into the web-based application for easy access and report generation.

However, it's important to note the study's limitations, including the exclusion of crack length measurement, internal flaw detection, and exploration of mobile applications. Furthermore, the application does not incorporate safety measures, and users are advised to adhere to safety guidelines during inspections and repairs. Moreover, YOLOv8 was chosen as the model due to its exceptional detection speed and its suitability for training with limited hardware resources [7]. This choice was reinforced by the procurement of 100 compute units from Google Colab, enabling uninterrupted training sessions despite the absence of GPU or similar high-performance hardware.

Despite these limitations, the study has the potential to significantly improve efficiency and accuracy in concrete structure inspection, thereby enhancing public safety and infrastructure longevity.

2 METHODOLOGY

2.1 Data Collection

The objective of this method is to compile a diverse dataset of concrete images, including examples of concrete cracks with various patterns. A total of 10,711 images were collected, with 711 sourced from four specific public schools in Dasmariñas, Cavite, and an additional 10,000 obtained from various public datasets within Roboflow. The primary dataset captures a broad spectrum of real life environmental and structural conditions, acquired using a Digital Single-Lens Reflex (DSLR) camera. Researchers took photographs from various angles and distances to ensure comprehensive coverage. These primary images are then uploaded to the project repository on Roboflow. Conversely, the secondary dataset was acquired by duplicating public dataset repositories into the project repository.

2.2 Data Processing

The primary dataset underwent annotation by researchers using Roboflow's annotation tool to segment concrete cracks. In contrast, the secondary dataset was pre-annotated, yet inaccuracies persist, prompting researchers to manually edit annotations as well. The images were resized to 640x640 and had their contrast adjusted using histogram equalization.

Moreover, data augmentation techniques were implemented to boost diversity and avoid underfitting. Calibration for varying lighting conditions was also conducted to maintain dataset consistency. Each training image generates three outputs through augmentation methods such as flipping, rotating by 90 degrees, shearing, adjusting saturation, brightness, exposure, applying blur, noise, and creating mosaics. An example of an original image is demonstrated by Figure 1, and its augmentations on Figure 2.

Figure 1: Original image of concrete crack.

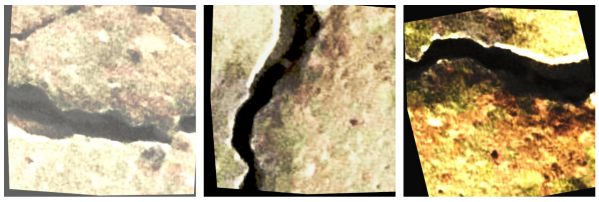


Figure 2: Augmented images of concrete crack.

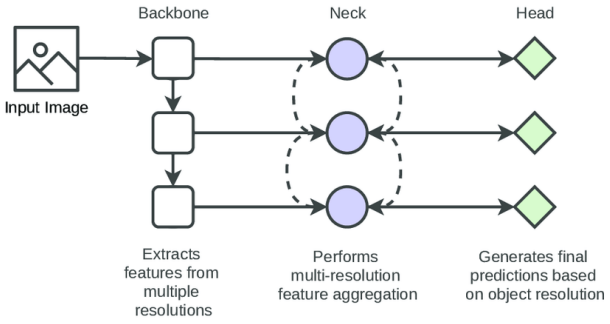


Figure 3: The YOLOv8 simplified model architecture.

2.3 Data Processing

The primary dataset underwent annotation by researchers using Roboflow's annotation tool to segment concrete cracks. In contrast, the secondary dataset was pre-annotated, yet inaccuracies persist, prompting researchers to manually edit annotations as well. The images were resized to 640x640 and had their contrast adjusted using histogram equalization.

Moreover, data augmentation techniques were implemented to boost diversity and avoid underfitting. Calibration for varying lighting conditions was also conducted to maintain dataset consistency. Each training image generates three outputs through augmentation methods such as flipping, rotating by 90 degrees, shearing, adjusting saturation, brightness, exposure, applying blur, noise, and creating mosaics. An example of an original image is demonstrated by Figure 1, and its augmentations on Figure 2.

2.4 Language, Model, Hosting Platforms

The web application was built using Gradio, a Python library that streamlines user interaction with machine learning models through intuitive web interfaces [14]. It simplifies the process of inputting data and presenting results, ensuring accessibility for users of varying technical backgrounds. On the other hand, Hugging Face serves

as a central hub for the machine learning community, offering pre-trained models and hosting capabilities through "Spaces." These Spaces enable effortless deployment of the model, allowing users to interact with it via a web browser without the need for software installation [15].

Furthermore, YOLOv8 is a cutting-edge object detection and segmentation algorithm that utilizes a single convolutional neural network (CNN) to simultaneously predict bounding boxes, class probabilities, and pixel-level masks for multiple objects in an image. It is known for its fast inference speed, high accuracy, and ability to segment objects at the pixel level, making it suitable for real-time applications [12].

As seen on Figure 3, YOLOv8 breaks down an image into its key features using a pre-trained CNN backbone. It then combines these features from different resolutions to create a comprehensive understanding of the image. Finally, a specialized head predicts bounding boxes and class probabilities for objects directly, without needing pre-defined boxes, making it faster and more efficient.

2.5 Crack Width Measurement

$$W_r = \alpha W_p D C \tag{1}$$

The crack width measurement utilizes the OpenCV library, which is capable of measuring objects in images by employing contours and distance transform [9]. Contour detection serves the purpose of identifying shapes within the image, while the distance transform computes the distance of each pixel to the nearest edge. By measuring the distance and angle of the object relative to the camera lens, the real-world dimensions can be inferred from the number of pixels with the following equation, where W_r is width in millimeters, α is the angle relative to the lens, W_p is width in number of pixels across, D is distance to the lens, and C is the calibration factor. The calibration factor is the average ratio of widths measured by calipers and by the application over varying crack widths taken from 15 centimeters.

3 PRELIMINARY RESULTS

3.1 Model Performance

This study employed a YOLOv8 neural network to detect cracks in concrete images. The model's performance was evaluated on the combined primary and secondary dataset comprising 21259 training images, 2136 validation images, and 1688 test images.

Table 1: Model Training Experiments

Test	Images	Augmented	Epoch	Precision	Recall	mAP
1	10000	23376	50/50	0.875	0.775	0.879
2	711	170	35/50	0.591	0.354	0.389
3	200000	46752	22/50	0.782	0.691	0.718
4	10711	16710	45/100	0.862	0.752	0.827
5	10711	25083	52/150	0.882	0.772	0.868

Table 1 shows different experiments involved training the YOLOv8 model with varying amounts of images, augmentation, epochs, and achieved different precision, recall, and mAP scores for concrete crack segmentation. Experiment 1 had the highest mAP but it only consisted of secondary images. Experiment 5 had the second highest mAP, while also including the primary images gathered by the researchers, indicating better overall performance. The model produced was used for the web application. The metrics are as follows: Precision – 0.882; Recall – 0.772; mAP50 – 0.868; F1 Score – 0.8233.

Precision measures the accuracy of the positive predictions made by the model, whereas recall assesses the model's ability to identify all actual positive cases. mAP, or mean Average Precision, is a metric used to assess the accuracy of object detectors like YOLOv8 across different classes and threshold levels. It calculates the average precision for each class and then averages these scores, offering a comprehensive measure of model performance. The F1 score, which is 0.8233 combines both precision and recall into a single metric. It is particularly useful as it balances the trade-offs between precision and recall, providing a single score to measure the overall efficacy of the model at a set confidence threshold, in this case, at 0.459.

The results highlight the extensive training process of the YOLOv8 model, incorporating crucial aspects such as batch size selection and early stopping to prevent overfitting [5, 4].Conversely, to avoid underfitting, data collection strategies and augmentation techniques were employed to diversify the training set [6]. With these methods, the model's performance plateaus at epoch 32, suggesting saturation of its learning capacity [5, 4, 6]. However, evaluation metrics demonstrate the effectiveness of the mentioned methods. Overall, YOLOv8 consistently achieves high precision and recall rates for object detection tasks, along with exceptional detection speeds. These factors make YOLOv8 a viable choice for concrete crack detection and mapping.

3.2 Measurement Validation

Table 2: Measured and Analyzed Crack Widths

Image No.	Measured Width (mm)	Analyzed Width (mm)	Difference (mm)	Error (%)
1	2.7	3.02	0.32	11.85
2	3.1	3.09	0.01	0.32
3	2	1.96	0.04	2
4	1.3	3.7	2.4	184.62
5	1	1.7	0.7	70
6	5.6	6.64	1.04	18.57
7	5.3	5.76	0.48	9.06
8	8.5	7.16	1.34	15.76
9	9.8	9.1	0.7	7.14
10	3.4	2.78	0.62	18.24
Average Difference				0.765
Average Error (%)				33.756

Table 2 shows the difference between the measured crack width and the width analyzed by OpenCV contour detection and distance transform. The average difference is 0.765 mm, with an average error of 33.75%. Overall, the results show that the system is a promising tool for concrete crack segmentation and width measurement.

The above data shows a coefficient of -0.445 between width and percentage error, and the largest errors are from images of cracks with widths under 2 millimeters. It can be surmised that the thinner

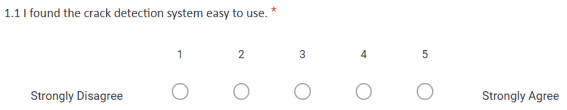


Figure 4: Example of user evaluation question.

the crack on the image, the more excess surface area the machine learning model segments from the image, contributing to the high percentage error for cracks with images under a certain threshold.

3.3 User Evaluation

3.3.1 Demographics. The study employed judgement sampling and concentrated on engaging participants comprising structural engineers, homeowners seeking to identify structural cracks in their residences, building proprietors, building managers, and engineering students. This deliberate selection of these distinct groups is intended to facilitate the acquisition of valuable insights and perspectives from individuals possessing pertinent expertise and hands-on experience in the realm of structural crack assessment. Of the 58 respondents, 25% of them state that they have working experience where their profession allows them to interact with concrete structures.

3.3.2 User Evaluation Survey. The quality of the web application was assessed by using a 5 – Point Likert scale. The ratings presented to respondents were qualitative and in increasing order with "Strongly Disagree" assigned to 1 point, and "Strongly Agree" assigned to 5 points, where 3 points indicates neutrality.

The system was assessed with questions fitting under the categories usability, functionality, and user satisfaction. The mean opinion score for questions under the usability category was 4.4, indicating that users found the system intuitive and easy to use. The mean opinion score for questions under the functionality category was 4.6, indicating that the users found that the features available to the users generally worked as intended. The mean opinion score for questions under the reliability category was 4.7, indicating that users experienced a low failure rate for the system's features. User Satisfaction was scored at a mean opinion score of 4.6, with an average score of 4.7 for the item "I would highly recommend this crack detection to others" indicating that users were generally pleased with their experience with the application. In summary, the model's performance and the application's design generally led users to be generally satisfied with the system.

4 RECOMMENDATIONS FOR FURTHER WORK

In moving forward, the researchers recommend expanding the dataset used for YOLOv8 training by incorporating a larger volume of images to further enhance the model's adaptability to diverse real-world scenarios. Specifically, a focus on including more instances of hairline cracks in the dataset will contribute to improving the model's sensitivity to subtle variations in concrete surface conditions.

Additionally, future efforts should explore the incorporation of data that factors in varying angles and perspectives during image capture. Considering the impact of angle on crack measurements, this adjustment in the dataset will contribute to refining the model's accuracy in assessing crack dimensions across different viewpoints. By addressing such nuances in data representation, we can further optimize the model's performance in accurately detecting and measuring cracks in real-world applications. This comprehensive approach to dataset enrichment, with a specific emphasis on diverse crack types and angles, will undoubtedly contribute to the continued improvement of the YOLOv8-based concrete crack detection system.

REFERENCES

[1] Juan Camilo Avendaño. *Identification and quantification of concrete cracks using image analysis and machine learning*. 2020.
[2] Hyunjin Bae and Yun-Kyu An. "Computer vision-based statistical crack quantification for concrete structures". In: *Measurement* 211 (Apr. 2023), p. 112632. DOI: http://dx.doi.org/10.1016/j.measurement.2023.112632.
[3] Kim Basham. *How to evaluate and troubleshoot concrete cracks - recommendations*. Aug. 2022. URL: https://www.forconstructionpros.com/concrete/equipment-products/repair-rehabilitation-products/article/21232603/kb-engineering-llc-how-to-evaluate-and-troubleshoot-concrete-cracks-recommendations.
[4] Jason Brownlee. *Difference between a batch and an epoch in a neural network*. Aug. 2022. URL: https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/.
[5] Jason Brownlee. *Use early stopping to halt the training of neural networks at the Right Time*. Aug. 2020. URL: https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/.
[6] Muhammad Faizan. *Apply data augmentation on Yolov5/yolov8 dataset*. June 2023. URL: https://medium.com/red-buffer/apply-data-augmentation-on-yolov5-yolov8-dataset-958e89d4bc5d.
[7] Muhammad Faizan. *Roboflow*. Apr. 2022. URL: https://medium.com/red-buffer/roboflow-d4e8c4b52515.
[8] GeeksforGeeks. *Distance transformation in image - python opencv*. Jan. 2023. URL: https://www.geeksforgeeks.org/distance-transformation-in-image-python-opencv/.
[9] GeeksforGeeks. *Measure size of an object using python opencv*. Apr. 2023. URL: https://www.geeksforgeeks.org/measure-size-of-an-object-using-python-opencv/.
[10] Fortune Business Insights. *Cement market size, share, growth: Emerging trends [2032]*. Apr. 2022. URL: https://www.fortunebusinessinsights.com/industry-reports/cement-market-101825.
[11] Byunghyun Kim and Soojin Cho. "Automated vision-based detection of cracks on concrete surfaces using a deep learning technique". In: *Sensors* 18.10 (Oct. 2018), p. 3452. DOI: http://dx.doi.org/10.3390/s18103452.
[12] Alon Lekhtman. *Train yolov8 instance segmentation on your data*. Feb. 2023. URL: https://towardsdatascience.com/trian-yolov8-instance-segmentation-on-your-data-6ffa04b2debd.
[13] Majid Mirbod and Maryam Shoar. "Intelligent concrete surface cracks detection using computer vision, pattern recognition, and Artificial Neural Networks". In: *Procedia Computer Science*. Vol. 217. Sept. 2020, pp. 52–61. DOI: http://dx.doi.org/10.1016/j.procs.2022.12.201.
[14] Ibrahim Abayomi Ogunbiyi. *How to deploy a machine learning model as a web app using gradio*. June 2022. URL: https://www.freecodecamp.org/news/how-to-deploy-your-machine-learning-model-as-a-web-app-using-gradio/.
[15] Miguel Rebelo. *What is hugging face?* May 2023. URL: https://zapier.com/blog/hugging-face/.

Towards a Memory-Efficient Filipino Sign Language Recognition Model for Low-Resource Devices

Shuan Noel Co
De La Salle University
Manila, Metro Manila
shuan_co@dlsu.edu.ph

Darius Ardales
De La Salle University
Manila, Metro Manila
darius_ardales@dlsu.edu.ph

Miguel Gonzales
De La Salle University
Manila, Metro Manila
miguel_gonzales@dlsu.edu.ph

Stephanie Joy Suzada
De La Salle University
Manila, Metro Manila
stephanie_susada@dlsu.edu.ph

Waynes Weyner Wu
De La Salle University
Manila, Metro Manila
waynes_wu@dlsu.edu.ph

Thomas James Tiam-Lee
De La Salle University
Manila, Metro Manila
thomas.tiam-lee@dlsu.edu.ph

Ann Franchesca Laguna
De La Salle University
Manila, Metro Manila
ann.laguna@dlsu.edu.ph

ABSTRACT

In this paper, we present a preliminary LSTM-based model for recognizing Filipino sign language words in videos using hand landmarks extracted from MediaPipe. Furthermore, we show that post-quantization can significantly reduce its size without sacrificing its performance, showing the potential for practical use. Despite being trained on only a small number of instances per class, results show that the model was able to achieve an accuracy of 93.29%, while a 90% reduction in model size.

KEYWORDS

Filipino sign language, sign language recognition, tinyML

1 INTRODUCTION

Nowadays, the world is becoming more interconnected and inclusive. Bridging the communication gap for all groups of people has emerged as a pressing goal for researchers and practitioners alike. In the Philippines, the deaf community represents a significant part of society that faces challenges in communication, often leading to discrimination and marginalization [7, 18, 19]. The Philippine Statistics Authority (PSA) estimates that there are 1,784,690 individuals with hearing difficulty in 2020, comprising around 1.6% of the population [14]. In 2018, the Philippine government signed into law Republic Act No. 11106, also known as the "Filipino Sign Language Act", which designates Filipino Sign Language as the national sign language of the Filipino deaf, mandating its use in schools, workplaces, and broadcast media [2].

First, we develop a preliminary LSTM-based deep neural network for recognizing a small subset of words that are specifically unique to FSL. We show that training a model for this small subset is possible with only a small size of training data. Second, we show that quantization can be used to substantially reduce the size of the model without sacrificing its performance.

This paper is structured as follows. Section 2 discusses the related literature. Section 3 discusses the methodology we used for developing the sign language recognition model. Section 4 discusses the evaluation of the model and the results of the model. Finally, Section 5 provides conclusions and directions for future work.

2 RELATED WORKS

This section discusses the related literature of this study and situates the position of this work in the existing body of knowledge.

2.1 Sign Language Recognition

There has been a wealth of studies done on sign language recognition. For a period, hidden Markov models (HMM) and recurrent neural networks (RNN) were the most common approaches to classify sign language [15]. However, with the recent developments in machine learning, most deep learning approaches such as CNNs and LSTMs have shown superior performance in this domain [16]. Despite its seemingly straightforward presentation, the problem of sign language recognition is a complex task with many considerations and challenges.

While most studies focus on the hand gestures only, there are studies that focus on recognizing the body gestures [9] and facial expressions [5, 17] as well. Another challenge in sign language recognition is that some of these features may be occluded at certain points in time [16]. One way to alleviate this is to perform feature fusion, considering all the features in the prediction to make the model more robust to such cases [8].

2.2 Filipino Sign Language Recognition

One major challenge is the small number of datasets available for FSL. The largest dataset to our knowledge for FSL is the FSL-105 dataset, which contains around 20 labelled instances for 105 introductory words and phrases [21]. While helpful, it does not compare to the amount of data available for other sign languages.

In recent years, a few researchers have made attempts to incorporate various sign language recognition approaches to FSL. In the works of [3, 4], a model for recognizing letters of the alphabet was developed. In the former, it was successfully deployed in a Raspberry Pi, achieving an accuracy of 93.29%. However, these works

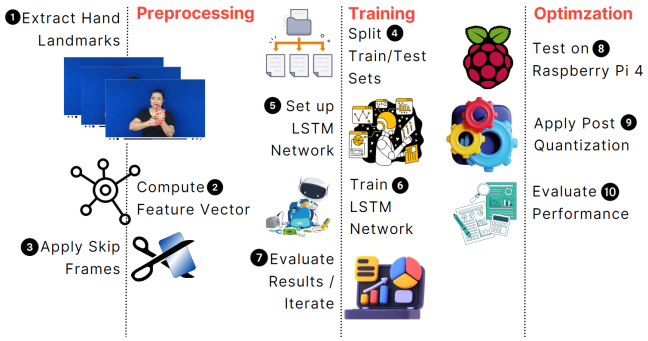


Figure 1: ML Pipeline

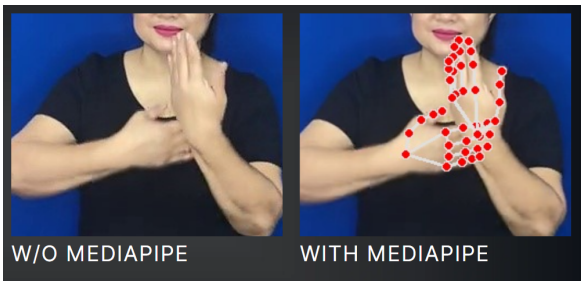
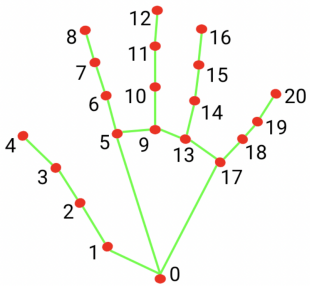


Figure 2: Before and After MediaPipe Processing

Table 1: List of Features Considered for the Model



are limited to alphabet signs only, which are all static gestures that do not incorporate movement. Similarly, the work of [12] trained a CNN model for static images showing numbers. Other works have applied various deep learning approaches to the task of FSL recognition, such as LSTM [6, 13], ResNet [13], and gated recurrent units [20], with good results on low label count manually collected datasets.

3 METHODOLOGY

This section discusses the methodology we used in training the sign language recognition model for FSL.

3.1 Dataset

In this study, we used data from the FSL-105 dataset. The FSL-105 dataset is a labelled dataset comprising 105 introductory words and phrases in FSL [21]. Each instance in the dataset contains the word or phrase, and a video of a person showing the sign language movement corresponding to that word. In this study, we only considered four words: “bread”, “egg”, “chicken”, and “crab”. These words were chosen because they have unique signs in FSL compared to other sign languages, and they are commonly used words. Each word has a total of 20 instances.

3.2 Training Pipeline

Figure 1 shows the pipeline for the training process. The process can be divided into three main phases. In the preprocessing phase, the videos are converted as a sequence of frames (images), which are then pre-processed using computer vision tools to extract key features in preparation for training. In the training phase, an LSTM deep neural network is trained from the input features. Finally, in the optimization phase, a post-quantization method is applied to the resulting model to reduce its size while maintaining its performance.

3.3 Preprocessing

In this phase, we perform preprocessing steps on the data in preparation for training. First, we extracted the instances belonging to the four target classes from the FSL-105 dataset. Next, we preprocess each video to extract the desired features for training.

Each video can be represented as a sequence of frames or images. For each frame, we extract key landmarks showing the inferred position and orientation of the hand on the frame. First, OpenCV was used to extract the image data from the individual frames of the video, then the image data was fed to MediaPipe to extract the landmarks. This preprocessing allows us to isolate the hands from the other parts of the video to eliminate background noise. This helps the model focus on the relevant information. For example, factors such as the skin color of the person doing the gesture can be ignored as they are not relevant to the task.

Our preprocessing and training process closely follows the framework outlined by [10]. Figure 2 shows some examples of video frames and the corresponding landmarks extracted by MediaPipe [11]. MediaPipe extracts 21 landmarks representing key joints in a person’s hand. Each landmark is represented as a normalized point in 3-D space with three numerical values corresponding to the x, y, and z components. From these raw landmarks, we compute a feature vector by engineering a set of features that are more meaningful for gesture representation. Specifically, we compute the distances between a set of landmark pairs. Table 1 shows the distances that were computed for each hand.

Each instance is represented as a sequence of frames, where each frame is embedded as the feature vectors, we only considered the frames where the hand has been detected by MediaPipe. We also applied a skip frame operation to standardize all sequences to 10 frames.

3.4 Training

We posed the sign language recognition problem as a problem of sequential gesture recognition. While there are certain words and phrases that don’t require much movement or changes in the gesture, there are also words and phrases that rely on the movement

of the hand gesture over time. To accommodate this, we chose LSTM as the model architecture for training the sign language recognition model. The "sequence" pertains to the sequence of embeddings per frame of a single sign language gesture instance.

LSTM (Long Short-Term Memory) neural networks excel in capturing long-range dependencies within sequential data. Unlike simple neural networks that struggle with learning relationships over extended sequences due to vanishing or exploding gradient problems, LSTMs are specifically designed to address this issue. The key innovation lies in their gated architecture, allowing the network to selectively remember or forget information over time. Each LSTM unit possesses a memory cell that serves as a persistent storage, and three gates (input, forget, and output) regulate the flow of information. The input gate controls which information to update, the forget gate decides what to discard from the cell's memory, and the output gate determines the information to be passed to the next layer. This intricate mechanism enables LSTMs to capture nuanced temporal dependencies.

We split our dataset into a training and test set with an 80-20 split. This resulted to 64 instances for training and 16 instances for testing. We then define the architecture of our model based on [10], defined as follows. In order: (1) an LSTM layer with 256 neurons, (2) a dropout layer, (3) another LSTM layer with 256 neurons, (4) another dropout layer, (5) an LSTM layer with 128 neurons, (6) a dense layer, (7) a batch normalization layer, (8) a RelU activation function, and finally (9) an output layer with 4 output neurons and softmax activation function. We used categorical cross entropy as the loss function, a decaying learning rate starting from 0.001, and ADAM as the optimizing algorithm. We trained the model for 300 epochs. Finally, we test the performance of the model by evaluating its predictions of the test set. The model training was implemented through TensorFlow. [1].

3.5 Optimization

After training, we used optimization techniques to reduce the memory requirements of the resulting model. The main technique we used to optimize the model was post-quantization. Post-quantization is a technique that reduces the precision of numerical representations such as the weights and activations of the neurons from a floating point to a lower-bit fixed-point of numbers, thereby compressing the model and reducing its memory storage requirement and computational complexity. We also converted the model from TensorFlow to TensorFlow Lite, which streamlined the deployment process for lower-end devices like mobile devices and resource constrained environments.

4 RESULTS AND FINDINGS

This section discusses the performance of the resulting sign language recognition model and compares it against alternative approaches.

4.1 Performance of the Model

Even prior to quantization, the LSTM model trained on the hand landmarks achieved a 100% precision, recall, and accuracy on the validation data. Figure 2 shows the confusion matrix of the predictions, while Figure 3 shows the training and validation set accuracy throughout the training process. The optimal accuracy was already achieved in less than 50 epochs of training.

Table 2: Confusion Matrix for Model Performance

Actual\Pred	Egg	Chicken	Crab	Bread
Egg	3	0	0	0
Chicken	0	5	0	0
Crab	0	0	2	0
Bread	0	0	0	6

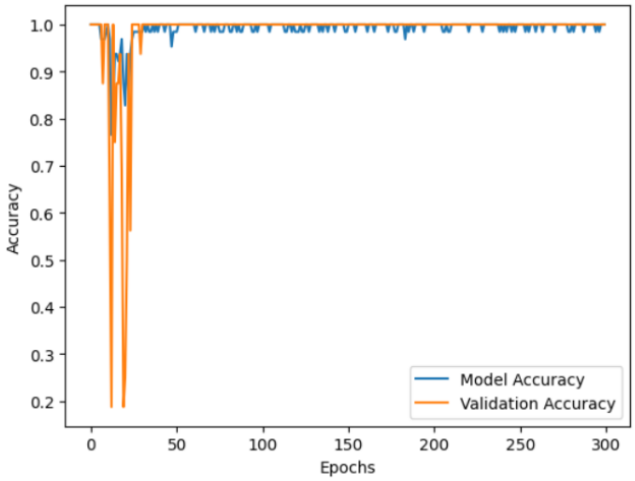


Figure 3: Train and Validation Accuracy in Model Training Process

4.2 Effect of Post-Quantization

After performing post-quantization, we were able to reduce the model size from 11.68MB to 1.01MB, corresponding to a 91.35% reduction in size. Despite this substantial reduction, the model maintained a high accuracy of 93.75%, with only one misclassification in the validation set. Table 3 shows the confusion matrix of the model after post-quantization. These results show the immense potential of post-quantization in the context of FSL recognition.

Table 3: Confusion Matrix for Model Performance After Post-Quantization

Actual\Pred	Egg	Chicken	Crab	Bread
Egg	1	0	0	0
Chicken	0	4	0	1
Crab	0	0	8	0
Bread	0	0	0	2

When testing the model, significant improvements in usability can be observed. For instance, prior to quantization the model would take a while to load when run on lower end systems. It would also

suffer from delays when deployed as a real-time application. However, post-quantization significantly boosted performance speeds, decreased loading times, and removed delays.

Nonetheless, there are still some problems when deploying the model on microcontrollers like Raspberry Pi due to the bottleneck of the MediaPipe processing. This causes performance issues when used in a real-time setup. Nonetheless, the model can run efficiently when used in an offline setting, or if the hand landmarks can be pre-processed. These results show that there are still issues to be resolved before the technology can be deployed in a real-world setting.

4.3 Comparison with Alternative Approaches

To highlight the advantages of the approach discussed in this paper, we compared the performance of the model against more straightforward approaches.

4.3.1 CNN without MediaPipe. The poor performance can of course be attributed to the fact that motion information was not being considered in this condition. CNN does not consider past movements or gestures, it may have difficulties in differentiating the classes, especially given the lack of data.

4.3.2 LSTM Without MediaPipe. We also attempted to train an LSTM model but without using MediaPipe by feeding in the individual frames of the RGB video sequence. For this model, the accuracy improved to 31.25%. Upon closer inspection of the confusion matrix in Table 4, it becomes clear why this is the case – all the validation instances were being predicted under "crab".

Table 4: Confusion Matrix for LSTM Performance Without MediaPipe

Actual\Pred	Egg	Chicken	Crab	Bread
Egg	0	0	5	0
Chicken	0	0	5	0
Crab	0	0	5	0
Bread	0	0	1	0

These results show that while LSTM can theoretically handle image sequence data, the use of raw RGB frames as LSTM input is not enough to successfully train an effective model with such a small dataset. Furthermore, the resulting size of the LSTM model is large at 1.73GB. These results highlight the advantage of adding a pre-processsing step to detect hand landmarks, as it significantly reduces the input size of the model, allowing for faster learning and smaller model size.

4.4 Summary of Results

Table 5 shows the summary of the results. From here, we can see that the introduced model for sign language recognition achieved good results on FSL by capturing hand movements as well as limiting the feature space to only the hand landmarks. Furthermore, post-quantization was able to significantly reduce the model size, showing potential for it to be deployed on low-resource devices.

Table 5: Summary of Results

Model	Input	Quantization	Accuracy	Model Size
CNN	Raw single video frame	no	25.49%	42MB
LSTM	Raw video sequence of frames	no	31.25%	1.73GB
LSTM	**MediaPipe hand landmarks on sequence of frames**	**no**	**100%**	**11.68MB**
LSTM	**MediaPipe hand landmarks on sequence of frames**	**yes**	**93.75%**	**1.01MB**

5 CONCLUSION AND FUTURE WORK

There is still a lot of future work in the field of FSL recognition. First, the models can be scaled up to handle a wider set of vocabulary. In this aspect, it would be interesting to explore whether current models would struggle with a larger set of classes, some of which may have similarities with one another. In this regard, one consideration is the development of techniques that do not require large amounts of data, as FSL datasets are currently limited. Second, the facial expressions and body gestures can be incorporated into the models, handling challenges such as occlusions to improve performance. Third, we can explore optimization techniques such as post-quantization for the development of real-time FSL recognition systems that can work on smartphones or similar devices so that they could be democratized to the Philippine deaf community. We believe these results can serve as a foundation for more FSL research and pave the way for the development of larger-scale recognition systems for the language.

REFERENCES

[1] [n. d.]. TensorFlow. https://www.tensorflow.org

[2] 2018. Republic Act No. 11106. https://www.officialgazette.gov.ph/2018/10/30/republic-act-no-11106/

[3] Mark Christian Ang, Karl Richmond C Taguibao, and Cyrel O Manlises. 2022. Hand Gesture Recognition for Filipino Sign Language Under Different Backgrounds. In *2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*. IEEE, 1–6.

[4] Mark Allen Cabutaje, Kenneth Ang Brondial, Alyssa Franchesca Obillo, Mideth Abisado, Shekinah Lor Huyo-a, and Gabriel Avelino Sampedro. 2023. Ano Raw: A Deep Learning Based Approach to Transliterating the Filipino Sign Language. In *2023 International Conference on Electronics, Information, and Communication (ICEIC)*. IEEE, 1–6.

[5] Siddhartha Pratim Das, Anjan Kumar Talukdar, and Kandarpa Kumar Sarma. 2015. Sign language recognition using facial expression. *Procedia Computer Science* 58 (2015), 210–216.

[6] Carmela Louise L Evangelista, Criss Jericho R Geli, Marc Marion V Castillo, and Carol Biklin G Macabagdal. 2023. Long Short-Term Memory-based Static and Dynamic Filipino Sign Language Recognition. In *2023 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*. IEEE, 235–240.

[7] Jasmine DC Hidalgo, Kayla Joy R Pantanilla, Almira E Castro, and Mickaela R Alfon. 2023. Employability of Persons With Disabilities. *International Journal of Academic Management Science Research* 7, 4 (2023), 29–36.

[8] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. 2018. Multiposenet: Fast multi-person pose estimation using pose residual network. In *Proceedings of the European conference on computer vision (ECCV)*. 417–433.

[9] Dimitrios Konstantinidis, Kosmas Dimitropoulos, and Petros Daras. 2018. Sign language recognition based on hand and body skeletal data. In *2018-3DTV-conference: The true vision-capture, transmission and display of 3D video (3DTV-Con)*. IEEE, 1–4.

[10] Wee Kiat Lim. 2021. Hand Gesture Detection and Sequence Recognition. https://weekiat-lim.medium.com/hand-gesture-detection-sequence-recognition-7f3215f88dde.

[11] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Yong, Juhyun Lee, et al. 2019. Mediapipe: A framework for perceiving and processing reality. In *Third workshop on computer vision for AR/VR at IEEE computer vision and pattern recognition (CVPR)*, Vol. 2019.

[12] Myron Darrel Montefalcon, Jay Rhald Padilla, and Ramon Llabanes Rodriguez. 2021. Filipino sign language recognition using deep learning. In *2021 5th International Conference on E-Society, E-Education and E-Technology*. 219–225.

[13] Myron Darrel Montefalcon, Jay Rhald Padilla, and Ramon Rodriguez. 2022. Filipino sign language recognition using long short-term memory and residual network architecture. In *Proceedings of Seventh International Congress on Information and Communication Technology: ICICT 2022, London, Volume 4*. Springer, 489–497.

[14] Liberty Notarte-Balanquit. 2023. Filipino Sign Language: Filipino Sign Language Numerals and the Expansion of Deaf Linguistic Repertoire (online lecture). https://www.youtube.com/watch?v=4vBEN00ecGw

[15] Sylvie CW Ong and Surendra Ranganath. 2005. Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 27, 06 (2005), 873–891.

[16] Razieh Rastgoo, Kourosh Kiani, and Sergio Escalera. 2021. Sign language recognition: A deep survey. *Expert Systems with Applications* 164 (2021), 113794.

[17] Joanna Pauline Rivera and Clement Ong. 2018. Facial expression recognition in filipino sign language: Classification using 3D Animation units. In *Proc. the 18th Philippine Computing Science Congress (PCSC 2018)*. 1–8.

[18] Freya Silva-Dela Cruz and Estrella Calimpusan. 2018. Status and challenges of the deaf in one city in the philippines: towards the development of support systems and socio-economic opportunities. *Asia Pacific Journal of Multidisciplinary Research* 6, 2 (2018), 33–47.

[19] Marcella L Sintos. 2020. Psychological Distress of Filipino Deaf: Role of Environmental Vulnerabilities, Self-Efficacy, and Perceived Functional Social Support. *Asia-Pacific Social Science Review* 20, 3 (2020).

[20] Isaiah Tupal, Melvin Cabatuan, and Michael Manguerra. 2022. Recognizing Filipino Sign Language with InceptionV3, LSTM, and GRU. In *2022 IEEE 14th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*. IEEE, 1–5.

[21] Isaiah Jassen Lizaso Tupal and Cabatuan K Melvin. [n. d.]. FSL105: The Video Filipino Sign Language Sign Database of Introductory 105 FSL Signs. *Available at SSRN 4476867* ([n. d.]).

AI-Assisted Chest X-ray Annotation Tool for Abnormality Classification and Localization

Kyla Joy P. Shitan
University of the Immaculate Conception
Davao, Davao del Sur, Philippines
kshitan_200000000995@uic.edu.ph

Julieza Jane Bella A. Raper
University of the Immaculate Conception
Davao, Davao del Sur, Philippines
jraper_200000000052@uic.edu.ph

Karl Vincent F. Bersamin
University of the Immaculate Conception
Davao, Davao del Sur, Philippines
kbersamin_200000000668@uic.edu.ph

Kristine Mae M. Adlaon
University of the Immaculate Conception
Davao, Davao del Sur, Philippines
kadlaon@uic.edu.ph

ABSTRACT

Accurate interpretation of Chest X-ray (CXR) images presents challenges within the medical field, prompting the integration of Artificial Intelligence (AI) to support radiologists. This study introduces a comprehensive system crafted to facilitate the annotation process for radiologists. The primary objectives entail the development of an advanced annotation model utilizing EfficientDet for precise abnormality detection and bounding box placement. Furthermore, the system incorporates a customizable radiography tool aimed at refining annotations. Developed using EfficientDet and Django frameworks, the system underscores areas for enhancing model accuracy. Future endeavors will concentrate on gathering feedback from radiologists to further optimize the system's efficacy and utility in clinical settings.

KEYWORDS

Deep learning, chest-Xray, annotation, abnormality detection

1 INTRODUCTION

Approximately one billion radio imaging examinations are performed annually worldwide. The prevalence of radiologist errors has been estimated at 4% in a typical sample of cases encountered in practice (perceptual error). However, errors may be as high as 30% when test cases all show abnormalities (perceptual and cognitive errors) [3].

Even with our current medical annotation tools, there are still problems that require better improvements. Such challenges are: the Lack of Standardization, Time Constraints, Inter-observer Variability, and Complexity of Medical Images.

This gap and leave room for more research, and with this, the researchers will conduct further study which aims to be able to optimize the annotations that exist in a Chest-X Ray. The study will utilize two publicly available datasets; The NIH CheXpert Chest X-Ray dataset, and the Vin-DR CXR Dataset.

The datasets have 14 abnormalities overall, with 9 in common; Atelectasis, Cardiomegaly, Consolidation, Infiltration, Nodule/Mass, Pleural Thickening, Pneumothorax, Pulmonary Fibrosis, and Pleural Effusion.

A study on the effects of model augmentation in the diagnosis of CXR was conducted and it was found that machine learning tools created to simplify CXR interpretation perform well, enhance clinicians' detection abilities, and boost the effectiveness of radiology workflow. Clinical engagement and expertise will be crucial to secure the adoption of high-quality CXR machine learning systems [1]. This form of integration still needs further study after its initial usage during the COVID-19 pandemic [2].

A study was conducted for HITL (Human-in-the-loop) solution to improve chest radio-graph diagnosis. The extensive implications for future clinical AI deployment and implementation strategies arise from the superior diagnostic accuracy exhibited by the combined HITL AI solution when compared to both radiologists and AI functioning independently [6].

A study focused on enhancing ICU chest X-ray classification for diagnosing pathology in critical patients. The research utilized approaches such as using manual annotations, automatically generated silver labels, or a combination of both to evaluate their impact on classification performance [8].

With limited manual annotation, models trained on silver labels notably enhanced performance. The MS model, trained solely on silver labels, achieved a 75.3% AUC score, increasing to 75.5% with transfer learning (MC+S). However, the quality of silver labels was crucial; if too erroneous, transfer learning proved a useful alternative. Combining silver labels with transfer learning and additional training on gold labels yielded optimal results.

Another study focuses on the challenging task of diagnosing chest-related diseases through chest X-ray (CXR) radiography.

With this dataset, the researchers employed an ensemble approach. Leveraging an ensemble of deep learning models including EfficientNet-B5, Xception, and DenseNet-201. The model first classified diseases based on infected organs (heart or lung), achieving an impressive AUC of 0.9489 for multi-classification. In the subsequent binary classification phase for specific diseases, the model demonstrated outstanding average AUC values of 0.9926 for heart diseases and 0.9957 for lung diseases. The study's innovative augmentation techniques and careful hyperparameter tuning, the research achieved superior results, surpassing previous models. Rigorous testing on various diseases, including pneumonia, edema, and consolidation, consistently demonstrated high accuracy (e.g., 0.9954 accuracy and 0.9956 AUC for pneumonia), underscoring the model's robustness and reliability across different disease categories [5]. The study has a limitation, primarily the absence of a detailed discussion on potential challenges faced during the implementation process.

Another study utilized the VinDr-CXR dataset where the researchers proposed a two-step approach; To employ the use of YOLOv5 to pinpoint abnormalities' locations, and, a binary CNN classifier, ResNet50, to classify these abnormalities [7].

The findings showed an enhancement with the two-step method, achieving a notable 77% F1 score and an mAP@0.5 score of 81.2% when YOLOv5 and ResNet50 were combined. This surpassed single-step approaches like YOLOv5, Faster R-CNN, and CheXNet.

The next study proposed a novel two-step approach for classifying chest X-ray (CXR) images. The first step involved multi-class classification, categorizing images into normal, lung disease, and heart disease. The second step focused on binary classification, identifying specific diseases within the lungs and heart. To implement the two-step classification approach, they developed two deep learning methods: DC-ChestNet, an ensemble learning of three deep convolutional neural network (DCNN) models, and VT-ChestNet, based on a modified Swin transformer architecture [13].

In the first phase, VT-ChestNet outperformed competitors with an AUC of 95.13%, followed by the average AUCs of 99.26% for heart diseases and 99.57% for lung diseases. DC-ChestNet also yielded promising results, starting with a 94.89 AUC and demonstrating notable accuracy in binary classification, achieving 99.26% AUC for heart diseases and 99.57% AUC for lung diseases.

The study by [9], which leveraged the NIH CheXpert dataset, focuses on comparing radiologists and a convolutional neural network-based AI algorithm in interpreting chest X-ray images.

Using a clinician-guided approach, the researchers categorized potential findings in chest X-rays systematically. Results show the AI algorithm achieved an AUC of 0.807 for labels and a weighted mean AUC of 0.841 after training. However, it excelled in high-prevalence findings and performed slightly less for rarer conditions. The comparative accuracy, measured using the Kappa statistic, was 0.543 for the AI algorithm and 0.585 for radiologists.

The study also details a method for labeling images based on radiological reports, achieving high precision (99.2%) and recall (92.6%). Limitations include an initially unbalanced dataset and a small number of radiology residents in the comparison.

In [4], an innovative approach using Weighted Boxes Fusion (WBF) to combine annotations from multiple radiologists is introduced. This method enhances abnormality detection in chest X-ray images by leveraging the expertise of multiple radiologists to improve deep neural network performance.

The proposed approach achieved better mean average precision (mAP) scores, indicating its effectiveness in training image detectors from labels provided by multiple radiologists.

Objectives:

Building upon relevant studies, the researchers have identified the following objectives to guide the investigation and development process:

(1) Develop an annotation model for abnormality detection and bounding box placement on chest X-ray images using EfficientDet
(2) Customize a radiography tool that allows radiologists to review, edit, and refine the annotations generated by the model, ensuring accuracy and incorporating domain expertise.
(3) Implement a method to save annotations by the radiologist and the model.
(4) Assess model performance in abnormality detection and annotation accuracy using metrics like mAP, IoU, accuracy, F1 score, confusion matrix, and inference speed. Evaluate accuracy in identifying abnormalities.

2 METHODOLOGY

In this section, researchers will detail the methodology for developing the annotation tool and detection model.

2.1 Conceptual Framework

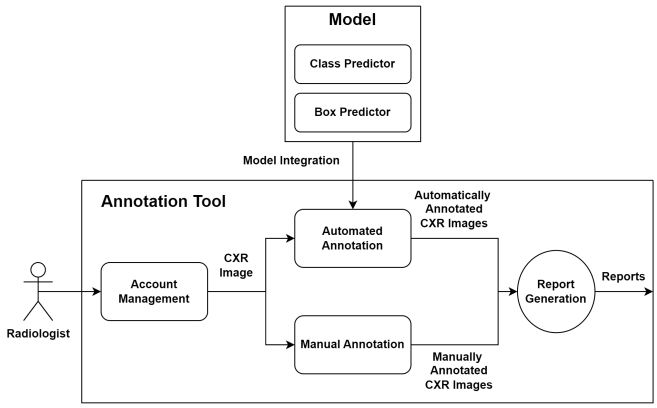


Figure 1: Conceptual Framework

As shown in the figure, this framework outlines a systematic process for abnormality detection in Chest X-ray images, combining human expertise and computational algorithms for accuracy and efficiency. The Annotation Model, EfficientDet, locates and classifies abnormalities, while radiologists refine initial annotations using the Annotation Tool.

Annotation and Classification Model:
The Annotation Model, built on EfficientDet, efficiently processes Chest X-ray images, identifying abnormalities. Integrated into the platform, it speeds up annotation by providing initial automated annotations, which radiologists can review and improve.

Annotation Tool:
The annotation tool supports radiologists in diagnostic tasks. Radiologists log in securely to upload chest X-ray images, which undergo two annotation methods: automated annotation by an EfficientDet model and manual annotation by radiologists. After annotation, the tool generates comprehensive reports, ensuring efficient workflow from image upload to report generation.

2.2 Model Preparation and Integration

This section outlines preparing, training, and integrating the pre-trained EfficientDet D0 model for annotating chest X-ray images from the VinBig and NIH datasets.

Data Preprocessing: Data preprocessing included converting images from PNG to JPG for consistency and resizing them to 512x512 pixels for uniformity and accuracy. The model focused on 9 common classes for streamlined training. Annotations for the same image were merged using Weighted Box Fusion (WBF) to improve accuracy. Out of 1908 images, 1527 were for training, and 382 for validation, with an 80% training and 20% validation split.

Model Training For model training, TensorFlow and Keras were used. The EfficientDet D0 model had pre-trained weights and was configured with 9 classes. Hyperparameters were carefully chosen, and regularization reduced errors. Data augmentation improved adaptability, and performance evaluation used COCO detection metrics.

2.3 Annotation Tool Features Implementation

This part discusses the features of the annotation tool. It goes into detail about the major and minor features that the system offers.

User Account Management User account management involves a simple registration form for new users to sign up and a secure logout function to end sessions.

Image Management Image management allows users to upload and change images. Image annotation and processing involve automatic annotation with the model and manual tools for radiologists. Users can draw boxes, add labels, and zoom for detailed inspection, aiding in categorization and reporting.

Reporting and Data Management This include generating reports summarizing key findings from annotated images and saving both images and reports for future reference.

2.4 Materials Used

In this part, the discussion shifts to the tools used to craft the system.

Software Tools The project used XAMPP with phpMyAdmin for local server development and database management. Django, Konva, and Bootstrap were also utilized . MySQL handled structured data storage.

Datasets The VinBig and NIH Chest X-ray datasets, obtained from Kaggle, supplied chest X-ray images with annotations for training the EfficientDet model.

Hardware and Computational Resources Google Colab, a cloud-based platform, was used to train the EfficientDet model. The annotation tool was developed and hosted in a local server environment provided by XAMPP.

2.5 Testing and Validation

The model will undergo thorough testing to assess capabilities and identify improvements.

Average Precision and Average Recall: These metrics provided insights into how accurately the model could identify and localize objects of interest across different levels of detection strictness and object sizes.

Area Under the Curve (AUC) Scores: These scores provided a quantitative measure of the model's ability to distinguish between different types of abnormalities.

3 PRELIMENARY RESULTS

This study applies object detection to chest X-ray analysis. The aim is to identify and localize key features and abnormalities in chest X-rays. The model was trained for 1300 steps on a merged dataset of chest X-ray images.

Figure 2: Total Loss over Training Steps

After analysis, it was found that the model reached its peak at the 1200th step with the lowest training loss. However, at the 1300th step, there was a noticeable increase in total loss, indicating suboptimal learning. Therefore, training was stopped at the 1200th step for optimal model performance.

At the 1200th step, the model's loss metrics were:

- Classification Loss: 0.43003213
- Localization Loss: 0.010713379
- Regularization Loss: 0.032603912
- Total Loss: 0.47334942

During training, the loss steadily decreased, indicating progress. However, there is still room for improvement.

3.1 Performance

The object detection model's performance was evaluated using Average Precision (AP) and Average Recall (AR) metrics across various IoU thresholds and image area sizes. Higher AP and AR values indicate better performance. IoU measures overlap between predicted and ground truth bounding boxes, while 'maxDets' sets the maximum number of detections per image.

3.2 Analysis of Model Performance

1. Overall Precision and Recall:

AP @[IoU=0.50:0.95 | area=all | maxDets=100 = 0.104:]
This suggests that, on average, the model correctly identifies relevant objects with moderate accuracy when considering various IoU thresholds. It implies there's significant room for improvement in the model's precision.

2. Precision at Specific IoU Thresholds:

AP @[IoU=0.50 | area=all | maxDets=100 = 0.275:]
At an IoU threshold of 0.50, the model performs considerably better, indicating it can detect objects with a reasonable overlap with the ground truth.

AP @[IoU=0.75 | area=all | maxDets=100 = 0.050:]
At a higher IoU threshold of 0.75, the precision drops significantly, suggesting the model struggles with very accurate localization of objects.

3. Precision by Area Size:

The model shows varying performance based on the size of the detected objects. It performs best for large objects (AP = 0.146) compared to small (AP = 0.008) and medium-sized objects. (AP = 0.095)

4. Recall by Area Size:

Small Areas [IoU=0.50:0.95 | maxDets=100 = 0.057:]
The model's significantly lower recall for small areas suggests that it struggles to correctly identify smaller objects in the chest X-rays.

Medium Areas [IoU=0.50:0.95 | maxDets=100 = 0.245:]
There's a need for enhancement of the model's ability to detect medium-sized features. Considering that abnormalities usually fall into this size range.

Large Areas [IoU=0.50:0.95 | maxDets=100 = 0.349:]
This higher recall rate for large objects suggests that the model is more effective in identifying larger anomalies.

5. Recall Analysis:

The model's recall scores (AR) range from 0.174 to 0.349, showing improved detection as it makes more detections, especially for larger objects.

The model can identify chest X-ray features to some extent, but its accuracy in precisely localizing objects (higher IoU thresholds) and detecting smaller objects needs improvement. This moderate performance could be due to dataset complexity, model limitations, or the need for more training or advanced augmentation techniques.

3.3 AUC Scores

The table below displays the Area Under the Curve (AUC) scores for each class.

Table 1: AUC Scores

Abnormalities	AUC Scores
Cardiomegaly	0.54
Pleural Thickening	0.51
Pulmonary Fibrosis	0.56
Pleural Effusion	0.63
Nodule/Mass	0.54
Infiltration	0.58
Atelectasis	0.46
Consolidation	0.47
Pneumothorax	0.5

The AUC score evaluates how well a model can distinguish between classes, derived from the ROC curve. While the model shows potential in spotting some chest X-ray issues, its AUC scores aren't optimal, particularly for certain conditions. This highlights the need for further model refinement and investigation into areas of poor performance.

3.4 Overall Performance

The model demonstrates a moderate level of accuracy in detecting various chest abnormalities, as indicated by the AUC scores. However, there is room for improvement, especially in classes with AUC scores closer to 0.5, such as Pneumothorax and Consolidation. The model shows relatively better performance in detecting Pleural Effusion and Infiltration, as evidenced by higher AUC scores. While the model shows promise, its current level of accuracy necessitates further refinement before it can be reliably used in clinical settings. Improvements could include additional training data, further hyperparameter tuning, or exploring more complex model architectures.

4 FURTHER WORK

The evaluation of the AI-assisted chest X-ray abnormality classification model reveals promising yet moderate performance across various metrics. Integrated into the annotation system, the model can offer valuable assistance to radiologists by simplifying the detection and annotation process. While demonstrating an ability to identify abnormalities within chest X-ray images, there are notable areas for improvement, particularly in achieving higher precision and recall rates, especially for smaller abnormalities and precise localization tasks.

Currently, our efforts are focused on refining the model, representing just the initial step in this endeavor. Addressing data preparation and quality issues is important to enhancing model performance. Achieving acceptable metric results is crucial for future use, as there is still ample room for improvement.

In the future, we aim to assess the annotation tool's performance in clinical settings. Upon achieving satisfactory results, we'll integrate radiologists' feedback through the tool. Their input is crucial for correcting model errors and improving workflow efficiency.

Ongoing refinement and optimization efforts, including additional training data and fine-tuning of hyperparameters, are essential to enhance the model's capabilities and meet the rigorous standards required for clinical deployment. Despite current limitations, the integration of the model represents a significant advancement in AI-assisted radiology, with the potential to improve diagnostic accuracy and efficiency in clinical practice. Looking ahead, future iterations will focus on further improving the system and allowing radiologists to actively test and provide feedback, ensuring continuous enhancement and adaptation to clinical needs.

REFERENCES

[1] Hassan K Ahmad, Michael R Milne, Quinlan D Buchlak, Nalan Ektas, Georgina Sanderson, Hadi Chamtie, Sajith Karunasena, Jason Chiang, Xavier Holt, Cyril HM Tang, et al. 2023. Machine learning augmented interpretation of chest X-rays: a systematic review. *Diagnostics* 13, 4 (2023), 743.

[2] Harrison X Bai, Robin Wang, Zeng Xiong, Ben Hsieh, Ken Chang, Kasey Halsey, Thi My Linh Tran, Ji Whae Choi, Dong-Cui Wang, Lin-Bo Shi, et al. 2020. Artificial intelligence augmentation of radiologist performance in distinguishing COVID-19 from pneumonia of other origin at chest CT. *Radiology* 296, 3 (2020), E156–E165.

[3] Warren B Gefter, Benjamin A Post, and Hiroto Hatabu. 2023. Commonly missed findings on chest radiographs: causes and consequences. *Chest* 163, 3 (2023), 650–661.

[4] Khiem H Le, Tuan V Tran, Hieu H Pham, Hieu T Nguyen, Tung T Le, and Ha Q Nguyen. 2023. Learning from multiple expert annotators for enhancing anomaly detection in medical image analysis. *IEEE Access* 11 (2023), 14105–14114.

[5] Adnane Ait Nasser and Moulay A Akhloufi. 2022. Classification of CXR chest diseases by ensembling deep learning models. In *2022 IEEE 23rd International Conference on Information Reuse and Integration for Data Science (IRI)*. IEEE, 250–255.

[6] Bhavik N Patel, Louis Rosenberg, Gregg Willcox, David Baltaxe, Mimi Lyons, Jeremy Irvin, Pranav Rajpurkar, Timothy Amrhein, Rajan Gupta, Safwan Halabi, et al. 2019. Human–machine partnership with artificial intelligence for chest radiograph diagnosis. *NPJ digital medicine* 2, 1 (2019), 111.

[7] Vu-Thu-Nguyet Pham, Quang-Chung Nguyen, and Quang-Vu Nguyen. 2023. Chest x-rays abnormalities localization and classification using an ensemble framework of deep convolutional neural networks. *Vietnam Journal of Computer Science* 10, 01 (2023), 55–73.

[8] Helen Schneider, David Biesner, Sebastian Nowak, Yannik C Layer, Maike Theis, Wolfgang Block, Benjamin Wulff, Alois M Sprinkart, Ulrike I Attenberger, Rafet Sifa, et al. 2022. Improving Intensive Care Chest X-Ray Classification by Transfer Learning and Automatic Label Generation.. In *ESANN*.

[9] Joy T Wu, Ken CL Wong, Yaniv Gur, Nadeem Ansari, Alexandros Karargyris, Arjun Sharma, Michael Morris, Babak Saboury, Hassan Ahmad, Orest Boyko, et al. 2020. Comparison of chest radiograph interpretations by artificial intelligence algorithm vs radiology residents. *JAMA network open* 3, 10 (2020), e2022779–e2022779.

Open Law Philippines: Legal Document Retrieval Analysis

Andres Clemente
College of Computer Studies
De La Salle University Manila
Manila, Philippines
andres_clemente@dlsu.edu.ph

Priscilla Licup
College of Computer Studies
De La Salle University Manila
Manila, Philippines
priscilla_licup@dlsu.edu.ph

Kenn Villarama
College of Computer Studies
De La Salle University Manila
Manila, Philippines
kenn_michael_villarama@dlsu.edu.ph

Joshua Permito
College of Computer Studies
De La Salle University Manila
Manila, Philippines
joshua_permito@dlsu.edu.ph

Ann Laguna
College of Computer Studies
De La Salle University Manila
Manila, Philippines
ann.laguna@dlsu.edu.ph

Donnald Miguel Robles
College of Computer Studies
De La Salle University Manila
Manila, Philippines
donnald_robles@dlsu.edu.ph

ABSTRACT

Due to the vast amount of legal document data, understanding and organizing law-related documents can be challenging, particularly for those outside the legal field. Initiatives such as Harvard's Case Law project have made strides in simplifying this process through a web application and relevant data visualization tools. This study proposes developing a novel document retrieval system tailored to the nuances of Philippine law, leveraging advanced deep learning techniques. Recent advancements in natural language processing, particularly models like Juris2vec, have shown promise in legal text analysis. Alongside these developments, the emergence of transformer models such as LEGAL-BERT, specifically pre-trained and fine-tuned for the legal domain, further enhances our ability to process legal documents accurately. By integrating BERTopic for thematic organization/filtering and Sentence-BERT (SBERT) for semantic document search, and in collaboration with legal experts, our project aims to significantly enhance the accessibility and comprehension of legal documents for a diverse range of users, including legal practitioners, scholars and the public. This endeavor not only promises to bridge the gap in legal document retrieval but also to pioneer the application of these sophisticated models in the context of Philippine law, setting a precedent for future legal informatics research.

KEYWORDS

Legal document retrieval, legal informatics, topic modeling, document similarity, Sentence-BERT (SBERT), BERTopic

1 INTRODUCTION

1.1 Overview

Legal documents, including court cases, are considered public property; however, not all are readily accessible to the public. Some of these documents are only physically available, requiring a visit to an office and incurring reproduction expenses. In this digital era, the importance of making public documents accessible online and free of charge cannot be overstated. While efforts to standardize legal documents have begun on a global scale, Philippine law repositories still lack the necessary tools for efficient document search and analysis. Locating relevant documents typically involves sifting through numerous files to identify those pertinent to a specific case or topic. To address these challenges, implementing Natural Language Processing (NLP) techniques such as topic modeling and document similarity can significantly enhance the relevance assessment of each document.

There is a current lack of efficiency when it comes to the retrieval of Philippine legal documents. In order to provide a way to harness the complexity of these documents, NLP tasks such as topic modeling and document similarity, as well as the involvement of Large Language Models like Sentence-BERT (SBERT) and topic modeling techniques like BERTopic, should be implemented to properly organize the contents of legal documents.

1.2 Objectives

The study aims to create a document retrieval system for Philippine legal documents using the transformer models based on topic and semantic similarity. More specifically, the study's objectives tackle on the following:

(1) To cluster Philippine legal documents into relevant topics using BERTopic and implement topic-based filtering for a document retrieval system.
(2) To implement a semantic retrieval system for Philippine legal documents using Sentence-BERT (SBERT).
(3) To implement a user-interface that would allow users to visualize similarities as well as retrieve relevant Philippine legal documents given a specific query.

1.3 Scope & Limitations

This study focuses on nationally recognized Philippine legal documents such as Philippine Supreme Court Decisions, Republic Acts, Senate Bills, Presidential Proclamations, and other nationally recognized legal materials. Regional documents may be considered for inclusion in later phases of the project. This initial constraint is considered sufficient for the preliminary analysis because regional cases typically do not establish jurisprudence at the national level. While regional cases are important for understanding the application of laws in specific geographic areas, they typically do not carry the same weight in terms of setting precedent and shaping the legal landscape on a national scale. Focusing on Philippine legal documents will allow for a more comprehensive and focused initial

analysis, helping to understand the overarching legal framework and key legal principles that apply to the entire country. Due to the significant advancements of specific NLP techniques with regard to transformers, a focus on SBERT embeddings would be implemented in the study. This study will be especially beneficial for future researchers who are interested in the legal domain of the Philippines. However, several challenges that must be acknowledged include the computational requirements of advanced NLP models like SBERT and BERTopic as they demand computational power, especially with large datasets. The system will also handle sensitive legal documents so data privacy measures and protocols must be implemented to protect data integrity and confidentiality.

1.4 Significance

The study is essential because it can potentially spur numerous implications concerning the retrieval and interpretation of Philippine legal documents through Large Language Models. Moreover, it could also dictate the evolution and current condition of the Philippine legal system based on the data gathered by the researchers. Overall, the study holds significance for the NLP community by advancing techniques tailored for the legal domain in the Philippines, offering a system for enhanced legal research and document analysis within its unique context.

2 RELATED WORK

We present the related studies that discuss document similarity and topic modeling.

2.1 Document Similarity

The evolution of document similarity analysis in legal documents has seen significant progress through the adoption of advanced NLP technologies, transitioning from traditional techniques such as TF-IDF to more sophisticated methods including Word2Vec and transformer models like BERT and JurisBERT. The advent of sentence transformer models, exemplified by Sentence-BERT (SBERT), marks a substantial leap in overcoming previous computational hurdles, offering notable improvements in processing speed and accuracy for semantic textual similarity tasks. SBERT, in particular, demonstrates a considerable enhancement in computational efficiency over traditional methods, facilitating more practical applications in demanding NLP tasks [1]. JurisBERT represents a notable advance in domain-specific modeling, achieving significant gains in precision and training efficiency over multilingual BERT and BERTimbau for legal text analysis [2]. This model's development emphasizes the growing focus on tailoring NLP technologies to specific fields, enhancing their applicability and effectiveness in real-world scenarios. Further investigations into the comparative performance of BERT-like models and traditional methods in semantic similarity highlight the potential of tailored embedding strategies. A study delving into Russian news similarity detection emphasizes the advantages of in-domain pre-training and fine-tuning on specialized datasets [3]. Despite the computational demands, such as the slow GPU performance noted with SBERT-WK's QR matrix decomposition, these advancements contribute critical insights into optimizing NLP models for specific content analysis, pointing towards the ongoing refinement of document similarity approaches within the legal domain and beyond. Additionally, a noteworthy development in retrieval tasks is the introduction of the Relevance Score Proportional to Relevance (RPRS) [4]. This efficiently leverages SBERT bi-encoders for comprehensive text coverage without memory constraints, showing significant potential in legal domain retrieval tasks as they were also tested using the COLIEE 2021 dataset which is drawn from an existing collection of predominantly Federal Court of Canada case law. Consequently, empirical studies demonstrate that SBERT significantly outperforms traditional methods like TF-IDF, GloVe embeddings, InferSent, and Universal Sentence Encoder, and baseline BERT models in terms of accuracy, efficiency, and scalability [1] [5]. Not only does SBERT achieve higher MAP scores and better Spearman correlation values in semantic similarity tasks, but it also drastically reduces computational overhead, enabling the processing of large datasets in seconds rather than hours [1] [6].

2.2 Topic Modeling

Topic modeling has evolved significantly with the shift from classical models like LDA and NMF to advanced text embedding techniques using BERT variants, improving thematic discovery within documents by capturing contextual nuances. The introduction of BERTopic marked a significant advancement, utilizing BERT for embeddings, HDBSCAN for clustering, and class-based TF-IDF for topic prediction, enhancing interpretability and relevance of identified topics [7]. Despite its assumption of a single topic per document, BERTopic's approach to topic modeling has shown promise, especially within the legal domain, as seen with LEGAL-BERT [8]. This model, tailored for legal texts, demonstrated high accuracy in capturing the essence of complex legal documents, suggesting potential for automating legal document summarization. Additionally, BERTopic's effectiveness across domains was highlighted in a study, showing its adaptability and robustness, including in multilingual contexts, compared to other models like Top2Vec and classical approaches, which struggle with context and relationships between topics [9]. This evolution in topic modeling techniques represents a leap forward in extracting meaningful insights from vast text corpora, offering a more nuanced understanding of document themes.

3 OPEN LAW PHILIPPINES

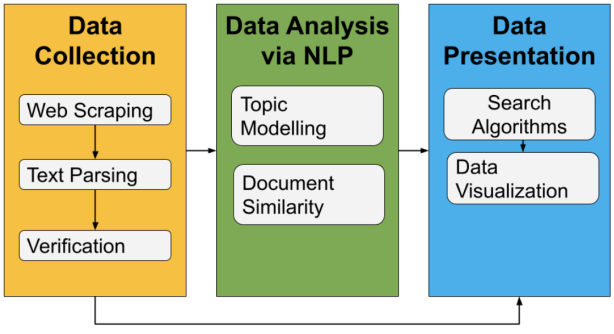


Figure 1: Conceptual Framework of the Study.

Open Law Philippines is the web application which would incorporate the document retrieval system for legal documents. The development of the study follows a 3-stage pipeline namely data collection, data analysis via NLP, and data presentation (Figure 1).

Each aspect of the framework will be discussed thoroughly in the following sections. As of this moment, the planned output of the Open Law Philippines study is a document retrieval system that incorporates SBERT embeddings for semantic search as well as assigning topics for finding and filtering relevant documents through a BERTopic model.

3.1 Data Collection

For this study, the data will be sourced from government websites in the Philippines, focusing on documents from the Official Gazette, Supreme Court Rulings, and House and Senate Bills, among others. The official gazette encompasses a variety of documents, including Executive Issuances, Presidential Speeches, Proclamations, and more. Legislative data includes senate bills, house bills, resolutions, journals, committee reports, republic acts, and treaties. Judicial data comprises Supreme Court Decisions, Resolutions, Rules of Court, and other related documents. This collection will be initially constrained to national documents, such as Legislative Acts, Republic Acts, Commonwealth Acts, Batas Pambata and Philippine Supreme Court Decisions. Regional documents will be excluded from this initial analysis as regional cases do not establish jurisprudence. Since web scraping may still have errors and inaccuracies, human verification will still be conducted after the study's methodology to ensure the veracity of data. Every data that has been verified by a human transcriber shall be noted. Due to the large amount of data that has to be collected, this system shall be implemented randomly. A legal expert would also provide their feedback on the effectiveness of the User Interface.

3.2 System Architecture

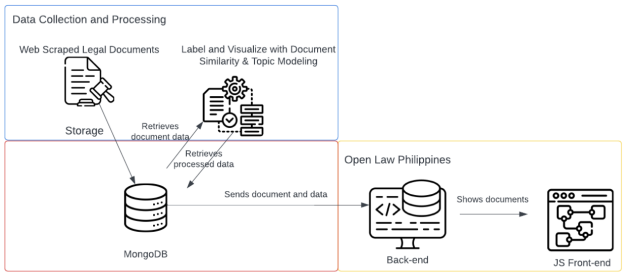


Figure 2: Diagram of Open Law Philippines Architecture.

Figure 2 shows the system architecture which encompasses data collection, processing, and storage of processed web-scraped data. Additionally, it involves document retrieval through the Open Law Philippines web application. MongoDB is chosen to store pre-labeled document data and processed documents for document similarity. For the Open Law Philippines back-end, Express will serve as the API responsible for querying and saving legal expert inputs and user corrections to the database. The back-end server primarily connects the frontend application to the MongoDB database, facilitating the retrieval of pre-labeled document data and supporting data visualizations in the study. In the frontend, ReactJS, known for its component-based UI, will be used for convenient programming. Additionally, ReactJS will be utilized together with NextJS, simplifying programming with built-in functions instead of custom ones.

3.3 Target Visualization

For this study, the focus of target visualizations has been narrowed down to dimensionality reduction scatter plots, BERTopic-specific visualizations, and word clouds. These chosen visualizations will serve as essential references for conveying the NLP techniques conducted in the study. Figure 3 shows a word cloud sample from using BERTopic. Also, the deliberate selection of dimensionality reduction scatter plots (see left of Figure 4), and BERTopic-specific visualizations such as distance maps (see right of Figure 4) and bar charts (see Figure 5) aim to provide a highly relevant visualization for the included processes in the system. Additionally, the incorporation of word clouds adds a textual dimension, highlighting key terms and patterns within the collection of legal documents. Conversely, scatter plots of dimensionality reduction as an implemented visualization for document similarity allow for better interpretation due to the reduction of high-dimensional document feature vectors through spatial distribution that represents similarities between the corpus of documents. These visualizations collectively contribute to enhancing the clarity of findings in the study.

Figure 3: BERTopic's Most Relevant Words for a Specific Topic used in Legal Documents.

3.4 Document Similarity & Topic Modeling

For tasks related to document similarity and topic modeling, SentenceTransformers (SBERT), which is fine-tuned for semantic search, will be used as the model, while the class-based c-TF-IDF will be utilized for topic selection in topic modeling. A variation of techniques such as tokenizing, dimensional reduction and clustering will also be implemented whenever applicable in order to handle the large amount of data to be assessed for topic modeling. For document similarity, a semantic search function will be used using Approximate Nearest Neighbors (ANN) because it offers a fast and efficient way to find the most relevant documents within a large

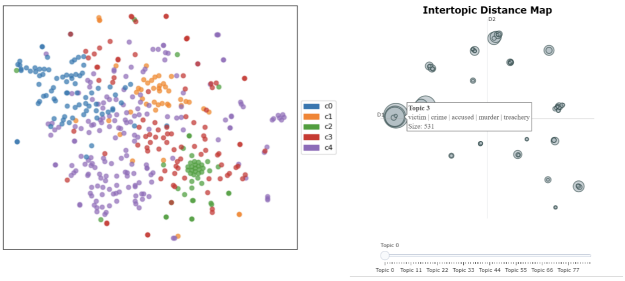


Figure 4: t-SNE Projection and Intertopic Distance Map.

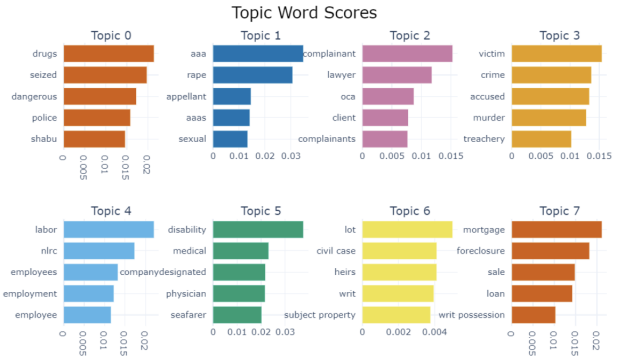


Figure 5: Topic Word Scores Bar Chart in BERTopic using a Subset of the Dataset.

dataset by approximating the nearest neighbors rather than computing the exact distances to all points in the dataset. Additionally, ANN's ability to work with high-dimensional data aligns well with the nature of document embeddings produced by SentenceTransformers, facilitating a more nuanced and accurate semantic search. Meanwhile, a list of top-assigned topics will be generated for topic modeling to provide insights into the prevalent themes within the data, allowing for effective organization, summary, and analysis of large text corpora.

4 FURTHER WORK

In an effort to address the challenges of legal document retrieval and analysis within the Philippine context, our proposal aims to enhance the retrieval and analysis of legal documents in the Philippines by integrating SBERT for semantic search and BERTopic for thematic organization. This combination is expected to improve the accessibility and usability of legal texts for professionals, scholars, and the public, using advanced natural language processing techniques to address gaps in legal informatics. Future plans include evaluating the BERTopic model with tools like OCTIS, integrating Elasticsearch with SBERT for better search capabilities, expanding the document repository to include regional texts, and supporting multiple local languages to cater to the Philippines' linguistic diversity. This ongoing work demonstrates our commitment to using cutting-edge strategies to make legal information more widely accessible.

REFERENCES

[1] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, nov 2019. Association for Computational Linguistics.

[2] Charles F. O. Viegas, Bruno C. Costa, and Renato P. Ishii. Jurisbert: A new approach that converts a classification corpus into an sts one. In Osvaldo Gervasi, Beniamino Murgante, David Taniar, Bernady O. Apduhan, Ana Cristina Braga, Chiara Garau, and Anastasia Stratigea, editors, *Computational Science and Its Applications – ICCSA 2023*, pages 349–365, Cham, 2023. Springer Nature Switzerland.

[3] A. Vatolin, E. Smirnova, and S. Shkarin. Russian news similarity detection with sbert: pre-training and fine-tuning. pages 692–697, 06 2021.

[4] Arian Askari, Suzan Verberne, Amin Abolghasemi, Wessel Kraaij, and Gabriella Pasi. Retrieval for extremely long queries and documents with rprs: A highly efficient and effective transformer-based re-ranker. *ACM Trans. Inf. Syst.*, 42(5), apr 2024.

[5] Jacob Malmberg. *Evaluating semantic similarity using sentence embeddings.* PhD thesis, 2021.

[6] Khushboo Taneja, Jyoti Vashishtha, and Saroj Ratnoo. Efficient deep pre-trained sentence embedding model for similarity search. pages 605–615, 09 2023.

[7] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure, 2022.

[8] Raquel Silveira, Carlos Gustavo Fernandes, Joao Araujo Monteiro Neto, Vasco Furtado, and J. Ernesto Pimentel Filho. Topic modelling of legal documents via legal-bert, Aug 2023.

[9] Roman Egger and Joanne Yu. A topic modeling comparison between lda, nmf, top2vec, and bertopic to demystify twitter posts. *Frontiers in Sociology*, 7, 2022.

Improved Restaurant Review Analysis using VADER-based Sentiment Analysis and Automatic Rating Matching

Alphonsus Joseph Bihag
College of Science and Computer Studies
De La Salle University - Dasmariñas
bad1246@dlsud.edu.ph

Justin Brian Abus
College of Science and Computer Studies
De La Salle University - Dasmariñas
aja0666@dlsud.edu.ph

Richard Tyrese Michio Uy
College of Science and Computer Studies
De La Salle University - Dasmariñas
urm0144@dlsud.edu.ph

ABSTRACT

This study presents an automatic review rating model for restaurant reviews using a rule-based sentimental analysis tool, VADER. The study aims to predict the rating of restaurant reviews based on their underlying sentiment. Sentiment analysis, a subfield of natural language processing, was used to determine the overall sentiment of a review, whether it is positive, negative, or neutral. This study demonstrates the effectiveness of using VADER for sentiment analysis to predict the actual rating of restaurant reviews as the findings of the study indicate. Utilizing LIME the researchers also explain the words that were most considered for (1) Highest Rated Reviews (2) Middle-rated reviews (3) Lowest rated reviews. The study also explores a theory in rule-based sentiment analysis of using language translation in order to make possible changes in accuracy. This study can be useful for businesses that rely on customer reviews, such as restaurants and food delivery services to gain insights into customer satisfaction and make data-driven decisions.

CCS CONCEPTS

• Artificial intelligence ~ Natural language processing • Machine learning ~ Supervised learning ~ Supervised learning by classification

KEYWORDS

Sentimental Analysis, Valence Aware Dictionary for Sentiment Reasoning (VADER), Artificial Intelligence, LIME, Language Translation

1 INTRODUCTION

Online reviews provide a valuable evaluation of a product or service's quality, serving as a trustworthy source of insight for both consumers and businesses. These reviews demonstrate their immense helpfulness in various commercial and social areas, influencing customer attitudes and choices regarding a company's goods and services. The restaurant industry, in particular, relies heavily on word-of-mouth from consumers. Statistics show that 76% of consumers consider online reviews to be highly important in ‘food and drink’ restaurant businesses, highlighting the dynamic role these reviews play in their improvement [9]. However, a critical challenge lies in these reviews, inconsistency between the actual content of the review and the provided rating by their author [12]. This discrepancy undermines the trustworthiness of reviews and hinders accurate interpretation. Sentiment analysis, a technique utilizing Natural Language Processing (NLP) to analyze textual sentiment, emerges as a potential solution to bridge the inconsistency gap [7].

Advancements in Natural Language Processing (NLP) have enabled computers to analyze and understand human language with increasing accuracy. This study utilizes VADER (Valence Aware Dictionary for Sentiment Reasoning), as a tool to address inconsistencies between review content and ratings provided. It is a rule-based sentiment analysis tool that follows grammatical and syntactical conventions for translating sentiment intensity. Most sentiment analysis models that use supervised learning algorithms these days consume loads of labeled data in the training phase to give satisfactory results which is usually expensive and leads to high labor costs in real-world applications [3]. However, VADER comes with its sentiment analysis lexicon, disregarding most of these costs. It is also a gold standard list of lexical features suitable for finding semantics in micro-blog text [1].

This study aims to classify various restaurant reviews using VADER-based sentiment analysis to provide matching ratings with restaurant reviews found online and determine the performance of the model.

2 METHODOLOGIES

2.1 Area of Study

The internet revolutionized how people interact with information and services, including the way they discover and share restaurant experiences. This research focuses on online restaurant reviews, specifically those found on social media platforms like Facebook and dedicated review websites like Zomato. Facebook, with its vast user base and ingrained social features, provides a unique platform for food reviews. Users can share their dining experiences with friends and followers, offering valuable insights and influencing the restaurant choices of others. Additionally, Facebook's search functionality allows users to discover reviews from a wide range of individuals, creating a comprehensive information pool on various restaurants and cuisines. Meanwhile, platforms like Zomato offer a wealth of restaurant-specific information, including menus, user reviews, and star ratings. This data allows researchers to delve deeper into consumer trends and conduct market research within the food industry. By analyzing Zomato reviews, we can gain valuable insights into consumer preferences, identify top-performing restaurants, and understand how factors like location and pricing influence a restaurant's success.

2.2 Data Gathering Procedure

Reviews amounting to 1150 were manually obtained by the proponents from Zomato and Facebook, listing all reviews from various restaurants which were compiled into an Excel (xlsx) file. The researchers provided the following information for each review: textual feedback, true rating, source, year written, and the restaurant for which the reviews were written. The 'true ratings' are derived from the evaluation of outside evaluators that were independent of both the original author and the proponents to label each review based on text connotations for model evaluation. For the selection of the reviews, the proponents focused on reviews of restaurants with a physical presence or local branch in the Philippines. Only reviews within a five-year range of the study's date of conduct were counted among the data for sentiment analysis. Also English, Filipino, and Taglish reviews were collected for this study. Further descriptions of the variables considered by the study in gathering data and other variables during the sentiment analysis procedure are provided in Table 1 below:

Variables	Description
Review	These are the feedback provided by customers of restaurants for their products and service either for the purpose of praise, suggestion, or expression of negativity.
True Rating	A label provided by external evaluators that classifies the reviews as either Positive, Neutral, or Negative for model evaluation.
Source	The website from which the reviews were obtained.
Year Written	The year in which the reviews were posted by their author in their respective source.
Recipient Restaurant	The restaurant for which the review was written by their author.
VADER Compound	The compound score produced by VADER that attributes the degree or score in negative, neutral, or positive altogether.
Star - Rating	The output of the model constructed in this study represents the scale of a review in its degree of negativity or positivity.

Table 1: Description of the Variables Considered and Used in the Study

2.3 Data Processing

Since VADER sentiment analysis operates primarily in English, reviews written in Filipino or Taglish required translation. To ensure consistency and efficiency, a batch translation approach was employed utilizing Google Translate. This involved grouping the Filipino and Taglish reviews together and submitting them for translation at once. While Google Translate offers a valuable tool for basic comprehension, it is important to acknowledge that nuances and cultural references within the reviews might not be perfectly captured in the translation process. The newly translated reviews were then combined with the reviews written in English.

Following the translation process, VADER-based sentiment analysis was applied to obtain a vader compound for each review. The vader compound was translated into matching ratings following a balanced distribution of scores that VADER could output from a range of -1 to +1 as discussed in Table 2 below:

Ratings and Sentiment	Compound Score Range
5 – Stars (★★★★★) (Very Positive)	0.60 to 1.0
4 – Stars (★★★★☆) (Positive)	0.21 to 0.59
3 – Stars (★★★☆☆) (Neutral)	-0.20 to 0.20
2 – Stars (★★☆☆☆) (Negative)	-0.59 to -0.21
1 – Star (★☆☆☆☆) (Very Negative)	-1.0 to -0.6

Table 2: VADER Compound Score to Matching Rating System

This system of assigning ratings based on the compound score is derived from their equivalent sentiment middle-ground where in the context of restaurant reviews, more stars depicted greater positivity [8].

2.4 Language and Model

Valence Aware Dictionary for Sentiment Reasoning or VADER relies on a dictionary that maps lexical features to emotion intensities called sentiment scores [2]. These scores are appropriately categorized into three categories that include neg (negative), neu (neutral), and pos (positive) to produce a compound score that factors in the previous categories based on the analysis of a given text. It computes the compound score using the formula below:

$$x = \frac{x}{\sqrt{x^2 + \alpha}}$$

Figure 1: Compound Score Formula for VADER

Where x = sum of valence scores of constituent words, and α = Normalization constant in which the default value is fifteen (15).

The compound score is the sum of the valence scores, adjusted according to the rules of the Sentiment Reasoning dictionary that is VADER, normalized to be between -1 for 'most extreme negative' and +1 for 'most extreme positive' [13].

However, in the case of the data produced by VADER, as it focuses on calculating and producing scores, it lacks proper explainability in the analysis process for humans to be able to understand. For this, an

algorithm called Local Interpretable Model-agnostic Explanations (LIME) may be used to help explain the prediction process of VADER [11]. It works by constructing a local interpretable model by finding the most important features [4][11] based on a set of calculated probabilities that are separated into provided classes based on a given sample of text. As a post-hoc method, it performs its processes after a prediction is made, meanwhile, LIME is able to present a visual model or aid of the probability calculated with the method separated into classes, top features, and textual evidence by highlighting the features from the text sample.

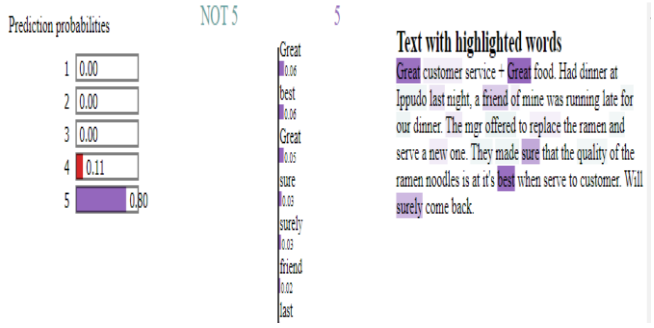


Figure 2: Visual Model produced by LIME

As exhibited by Figure 2, LIME utilizes the calculated probabilities and categorizes them into their appropriate class based on the method that has been factored in from the prediction of the sentiment classifier (VADER). It also shows a sorted graph of the features based on their relevance to the text sample fed to LIME while also providing a visual representation of the sample with highlights on the words shown in the sorted graph.

For the proponents of the study to interpret the data produced by their chosen method of sentiment analysis, LIME is used to explain certain samples of data. However, as rule-based methods such as VADER do not output class probabilities as in VADER's case that only outputs a single score (compound score), in order to utilize LIME to explain the results, it is needed to artificially generate the class probabilities.

```
def prediction(text):
    probs = []
    x = 0

    # First, offset the float score from the range [-1, 1] to a range [0, 1]
    offset = (vadar_sentiment(text) + 1) / 2.
    # Convert offset float score in [0, 1] to an integer value in the range [1, 5]
    binned = np.digitize(5 * offset, np.array([1, 2, 3, 4, 5])) + 1
    # Similate probabilities of each class based on a normal distribution
    simulated_probs = scipy.stats.norm.pdf(np.array([1, 2, 3, 4, 5]), binned, scale=0.5)

    while x < len(simulated_probs):
        probs.append(simulated_probs[x])
        x = x + 1
    this = np.array(probs)
    this

    return this
```

Figure 3: Artificial Class Probability Procedure

The procedure shown in Figure 3 utilizes a simple workaround to simulate the class probabilities using a continuous-valued sentiment score from the original range of '-1' to '1' by VADER to a normalized float score within the range of '0' to '1' that is scaled to five times in magnitude for each class in which for this case is based on the star-based rating system for reviews [10].

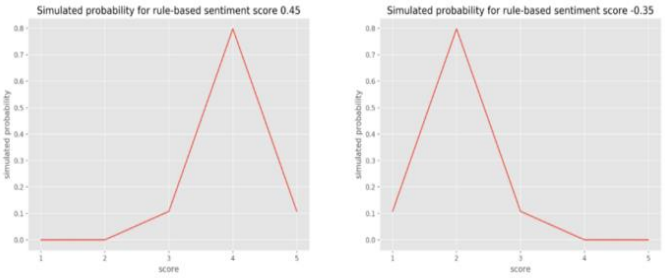


Figure 4: Examples of Simulated Probabilities Using the Work-around Procedure of Rao (2019): Artificial Class Probability Procedure

As shown in Figure 4, using the workaround procedure can adjust the compound score produced by a rule-based sentiment analyzer like VADER to the appropriate scale for the study, 0.45 was properly scaled into 4 in the graph from the left while -0.35 was appropriately scaled into 2 as found in the graph from the right.

2.4.1 Model Evaluation

Sentiment analysis relies on accurate results to ensure effectiveness. Metrics like precision, recall, and F1-score are calculated to assess this, considering how well the system classifies texts. These scores depend on correctly identifying positive, negative, and neutral sentiment, with this study expanding on existing metrics to include true Neutral (TN) and false Neutral (FN) as derived from Kanstren.

From this, it is possible to compute the following scores or metrics using the formulas that are summarized in the given Table 3:

Score/ Metric	Formula
Accuracy	$\frac{\text{True Positives + True Negatives}}{\text{True Positives + True Negatives + False Positives + False Negatives}} = \frac{\text{N. of Correct Predictions}}{\text{N. of all Predictions}} = \frac{\text{N. of Correct Predictions}}{\text{Size of Dataset}}$
Precision	$\frac{\text{True Positives}}{\text{True Positives + False Positives}} = \frac{\text{N. of Correctly Predicted Positive Instances}}{\text{N. of Total Positive Predictions you Made}}$:
Recall	$\frac{\text{True Positives}}{\text{True Positives + False Negatives + False Neutrals}}$
F1-Score	$2 * \frac{\text{Precision * Recall}}{\text{Precision + Recall}}$

Table 3: Summary of Metric and their Formulas

Sentiment analysis relies on several metrics to evaluate its effectiveness. Accuracy, the most fundamental metric, measures the overall proportion of correct predictions. Precision focuses on the exactness of positive predictions, while recall emphasizes the model's ability to identify all actual positive cases. Finally, the F1-score

combines both precision and recall for a more balanced assessment [5][6].

This study incorporates "Neutral" as a sentiment category. To account for this, an adjusted recall score will be calculated, penalizing the model for both false negatives (missing positive cases) and false neutrals (missing neutral cases). This adjustment ensures a more comprehensive evaluation of the model's performance.

3 RESULTS AND DISCUSSION

Actual vs Predicted Results			
TARGET / OUTPUT	Actual	Predicted	SUM
Actual	247 23.73%	188 18.06%	435 56.78% 43.22%
Predicted	16 1.54%	590 56.68%	606 97.36% 2.64%
SUM	263 93.92% 6.08%	778 75.84% 24.16%	837 / 1041 80.40% 19.60%

Figure 5: Checking Overall VADER Accuracy after translation

Figure 5 shows the accuracy of the sentiment analysis model in classifying the reviews. The model in classifying positive and negative reviews shows that it has an 80.40% accuracy.

```
print(f"True Positive count is {truePositiveCT}")
print(f"True Negative count is {trueNegativeCT}")
print(f"True Neutral count is {trueNeutralCT}")
print(f"False Positive count is {falsePositiveCT}")
print(f"False Negative count is {falseNegativeCT}")
print(f"False Neutral count is {falseNeutralCT}")
```

```
True Positive count is 590
True Negative count is 247
True Neutral count is 14
False Positive count is 188
False Negative count is 16
False Neutral count is 95
```

Figure 6: Identifying the True and False Prediction for the Translated and Combined Dataset

Figure 6 shows the total number of true or false positives, true or false negatives, and true or false neutrals after language translation. It was able to count a total of 590 for True Positive, 247 for True Negative, 14 for True Neutral, 188 for False Positive, 16 for False Negative, and 95 for False Neutral classifications. It can be observed that the majority of the predictions made fell under the True classification, showing that the model is significantly effective. After evaluation, the model performed with an F1-score of 0.85, a precision of 0.75, a recall of 0.97, and an overall accuracy of 74% when including neutrality.

Figure 7: LIME on Highest-Rated Reviews

By using the LIME Explainer model, ratings made by the model were easier to understand. Figure 7 shows the highest-rated review and how it was analyzed by VADER through LIME. It can be observed that is filled with positive words including 'pleasing' 'terrific' and 'better' leading to the review being the highest-rated. It shows the vast number of words that VADER considered to classify reviews.

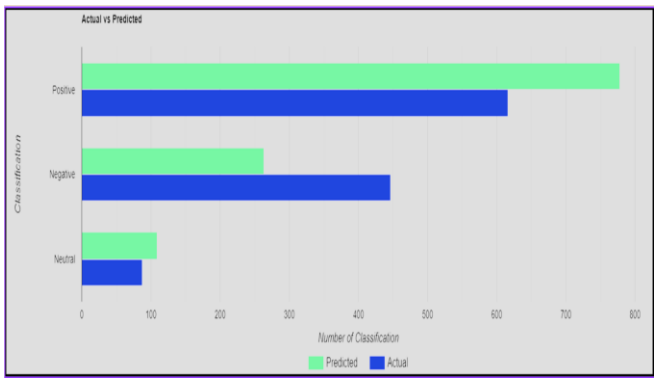


Figure 8: *Comparison of Actual and Predicted Classifications*

Figure 8 shows the comparison between the number of actual and predicted classifications. At the end of the sentiment analysis, the model classified 773 reviews as positive, 267 as negative, and 109 as neutral. In comparison to the actual classification of the data where 616 were positive, 446 negative, and 87 were neutral, the model appears to overestimate the positivity in the sentiment. There is a discrepancy of 157 classifications between positive and negative categories, with the model classifying 157 more reviews as positive than the actual data, classified by outside evaluator.

4 FUTURE WORK

This study acknowledges limitations due to VADER's English-centric nature. For multilingual data, translating reviews in English or using alternative NLP methods trained on the specific languages is recommended. Additionally, exploring state-of-the-art neural networks for sentiment analysis is suggested for potentially higher accuracy. Finally, the importance of a larger, balanced dataset with diverse sources is emphasized to enhance the overall analysis.

REFERENCES

[1] Bonta, V., Kumaresh, N., & Janardhan, N. (2019). A comprehensive study on lexicon based approaches for sentiment analysis. *Asian Journal of Computer Science and Technology*, *8*(S2), 1–6. https://doi.org/10.51983/ajcst-2019.8.s2.2037

[2] Calderon, P. (2018, March 31). *Vader sentiment analysis explained*. Medium. https://medium.com/@piocalderon/vader-sentiment-analysis-explained-f1c4f9101cd9

[3] Chiny, M., Chihab, M., Bencharef, O., & Chihab, Y. (2021). LSTM, Vader and TF-IDF based hybrid sentiment analysis model. *International Journal of Advanced Computer Science and Applications*, *12*(7). https://doi.org/10.14569/ijacsa.2021.0120730

[4] De Sousa Silveira, T., Uszkoreit, H., & Ai, R. (2019). Using aspect-based analysis for explainable sentiment predictions. *Natural Language Processing and Chinese Computing*, 617–627. https://doi.org/10.1007/978-3-030-32236-6_56

[5] Johnson, J. (2020, July 22). Precision, recall & confusion matrices in Machine Learning. BMC Blogs. https://www.bmc.com/blogs/confusion-precision-recall/

[6] Kanstren, T. (2020, September 12). A look at precision, recall, and F1-score. Medium. https://towardsdatascience.com/a-look-at-precision-recall-and-f1-score-36b5fd0dd3ec

[7] Kouadri, W. M., Ouziri, M., Benbernou, S., Echihabi, K., Palpanas, T., & Amor, I. B. (2020). Quality of sentiment analysis tools. *Proceedings of the VLDB Endowment*, *14*(4), 668–681. https://doi.org/10.14778/3436905.3436924

[8] Nielsen, N. (2024, February 29). *Restaurant rating system: Your guide to understanding reviews and stars*. Limepack Restaurant Rating System Your Guide to Understanding Reviews and Stars Comments. https://www.limepack.eu/blog/restaurant-rating-system-your-guide-to-understanding-reviews-and-stars

[9] Pitman, J. (2023, September 7). *Local consumer review survey 2022: Customer reviews and behavior*. BrightLocal. https://www.brightlocal.com/research/local-consumer-review-survey-2022/

[10] Rao, P. (2019, September 9). Fine-grained sentiment analysis in Python (part 2). Medium. https://towardsdatascience.com/fine-grained-sentiment-analysis-in-python-part-2-2a92fdc0160d

[11] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "why should I trust you?" *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2939672.2939778

[12] Shan, G., Zhou, L., & Zhang, D. (2021). From conflicts and confusion to doubts: Examining review inconsistency for fake review detection. *Decision Support Systems*, *144*, 113513. https://doi.org/10.1016/j.dss.2021.113513

[13] Swarnkar, N. (2020, May 21). *Vader sentiment analysis: A complete guide, algo trading and more*. Quantitative Finance & Algo Trading Blog by QuantInsti. https://blog.quantinsti.com/vader-sentiment/#:~:text=that%20hot.%E2%80%9D.-,Compound%20VADER%20scores%20for%20analyzing%20sentiment,1%20(most%20extreme%20positive

Enhancing Audio Data Processing: Insights from the Development and Evaluation of a Transcriber tool

Carlo A. Castro
University of the Immaculate Conception
Davao, Davao del Sur, Philippines
ccastro_200000000215@uic.edu.ph

Muslimin B. Ontong
University of the Immaculate Conception
Davao, Davao del Sur, Philippines
montong_200000000677@uic.edu.ph

Aurora Cristina Manseras
University of the Immaculate Conception
Davao, Davao del Sur, Philippines
amanseras@uic.edu.ph

Kristine Mae M. Adlaon
University of the Immaculate Conception
Davao, Davao del Sur, Philippines
kadlaon@uic.edu.ph

ABSTRACT

Language classification models play a crucial role in various natural language processing applications, including machine translation. While significant research has been conducted on text-based language classification, relatively less attention has been given to audio data. This paper aims to bridge this gap by exploring the development of a tool specifically designed for classifying audio inputs, with a particular focus on indigenous languages. The Minna Transcriber Tool is a solution tailored to preserve audio files and extract features by generating metadata for each audio segment. Additionally, the paper delves into the creation of a language classification algorithm capable of accurately identifying indigenous languages from audio recordings. Through a combination of classical machine learning techniques and deep learning algorithms.

KEYWORDS

Datasets, neural networks, natural language processing, translation

1 INTRODUCTION

Languages represent a cornerstone of human diversity, serving as conduits through which we perceive, interact with, and interpret the world in distinct ways. They encapsulate our cultures, collective memories, and values, forming an integral part of our identities [1]. In today's interconnected world, improving communication through technology is crucial [14]. By using technology to enhance communication, individuals, businesses, and organizations can overcome barriers and build stronger relationships [11]. One of the significant endeavors in language preservation in the Philippines is the work of Department of Science and Technology, which primarily concentrates on developing tools and technologies for Mindanao languages [4]. A work titled 'SultiWag' have collected more than three thousand (3,000) words from the combined Manobo, Kagan, and Davaoeño-Cebuano languages. Additionally, seventy-five (75) raw recordings were produced in three (3) languages during recording sessions. The data went through four (4) phases in pre-processing: Cutting, Extraction, Conversion, and Trimming, then files were saved in mp3 format. The Sultiwag's researchers converted the audio into spectrograms and exported them as JPEG files. The process resulted to 70% language classification accuracy. Primarily, the objective of this study is to explore an alternative method of data pre-processing and data feeding using spectrogram data directly, instead of saving it as JPEG. This approach is chosen to mitigate potential data loss during image export and to enhance the accuracy of audio classification, particularly for indigenous languages.

Indigenous languages serve a deeper purpose beyond communication—they act as a bridge to a community's heritage, nurturing a profound sense of belonging. These languages also act as vessels, carrying the ethical principles passed down through generations, shaping the values of those who speak them. Indigenous languages are disappearing worldwide, but there are numerous efforts and notable achievements in safeguarding these Indigenous languages and culture [12].

Language has the potential to thrive and endure if there is collective effort. With adequate support and resources at various levels – transnational, national, local, community, and individual, these dying languages can be revitalized or preserved [7]. Another study highlighted the ongoing efforts to preserve the diverse Indigenous languages of the Philippines. The Department of Education initiated a program called the Mother Tongue-based Multilingual Education program, aiming to revive and preserve Mother Tongue languages [8].

While Artificial Intelligence offers exciting possibilities for language revitalization, as seen in projects of [9], the key lies in collaboration. Indigenous communities must be active partners in developing AI tools for their languages. This ensures the technology respects the cultural context and avoids past exploitation by large corporations. Furthermore, such collaboration empowers communities to preserve their languages and traditions in the digital age. Imagine an AR experience where children learn Kwak'wala while navigating a virtual potlatch ceremony. AI, used thoughtfully and with respect, can bridge the digital divide and ensure these cultural treasures are not lost.

Another pivotal technology for language preservation and classification is audio data processing. It aids in pattern recognition by leveraging the distinctive acoustic attributes of various languages, thereby enhancing the efficacy of classification algorithms [3]. Moreover, audio data processing finds application in the creation of speech transcription systems for indigenous languages [13]. Techniques such as spectrograms and signal processing have proven instrumental in extracting crucial features from audio signals [2, 10].

Spectograms are graphs of audio signals that present carrier frequency and intensity change over time [5]. These kinds of spectrogram representations help researchers analyze and compare language-specific characteristics, i.e., phonetics and prosody features [10]. It is a 3D representation where time is on the X-axis, frequency on the Y-axis, and frequency amplitude on the Z-axis. This visualization helps identify the significant features and common patterns within language audio signals which helps to have strong models for classification [10]. Working with audio data may be challenging, as the voice quality, background noise, and speaker variations significantly impact the results. Moreover, having no access to datasets that are collected in different languages and different accents could make this task more complicated. To overcome this, data augmentation techniques and robust feature extraction methods can be employed. One of such techniques includes a number of data augmentation methods such as time masking, pitch shifting and noise injection which used for generation of a larger scale of the data from which the models for classification of spoken languages should be learned [6].

2 METHODOLOGY

This study aimed to improve the efficiency and accuracy of language classification for indigenous languages, particularly the Manobo and Kagan datasets. The revisiting researchers developed audio dataset pre-processing to enhance the quality of the recordings, which involved removing background noise and ensuring the clarity and accuracy of spoken words.

2.1 Data Collection

The researchers used two particular subsets, 314 Manobo words, and 405 Kagan words, and collected a total of 2959 audio datasets. The use of such groups was undertaken since it made it possible to investigate the specific problems and understand better the problems to solve with language classification algorithms.

2.2 Data Analysis

A close check on the researcher's data preprocessing methods was done to guarantee the accuracy and reliability of the subsequent analyses. The researchers exhaustively identified numerous speakers, ranging from different accents and cohorts. They processed words spoken by speakers in the audio files through a free and open-source digital audio editor and recording application software to filter out background noise from the specific audio files. Subsequently, they marked and labeled the segmented audio using a consistent format:

> **(English Word)_(Indigenous Translation)(First character of language)**
>
> **For example: abandon_paguyow(M)**

This approach enabled the researchers to organize the data effectively, streamlining further analysis. To determine which features must be extracted and pre-processed before training the model, further examination of the data was conducted. Upon examination, the researchers discovered that certain attributes, such as sample rate and duration, required careful pre-processing and extraction from the audio files. Sample rate refers to the number of audio samples captured per second, determining the frequencies represented in digital audio. Meanwhile, duration is employed to standardize audio clips to a consistent length. As various sounds have distinct sample rates, re-sampling them to a common rate can aid in audio classification.

2.3 Data Pre-processing

The researchers opted to analyze a focused subset of the dataset, specifically Manobo (314 words) and Kagan (405 words), to enable a more in-depth analysis of these languages for classification. This focus, however, resulted in an imbalanced dataset. To address this, class weights were assigned, prioritizing the underrepresented language during model training.

In the preprocessing stage, the audio data underwent standardization to a uniform sample rate of 16,000 Hz. Additionally, researchers investigated the optimal fixed duration for the audio files by analyzing the mean and standard deviation of their lengths. Due to the limited number of audio samples, data augmentation techniques like time shift, time mask, and frequency mask were implemented to enrich the dataset's diversity. Finally, the preprocessed audio data was transformed into spectrograms, a visual representation capturing sound wave information across frequency, time, and intensity. Spectrograms, well-suited for training Convolutional Neural Networks (CNNs) in audio classification tasks, were then converted into a format compatible with the CNN model for further analysis.

2.4 Train Model

In the development of the language classification models, the researchers employed a specific process to train the models:

- **Loss Function:** The Sparse Categorical Crossentropy loss was utilized, suitable for multi-class classification tasks where the labels are integers.
- **Optimizer and Scheduler:** An Adam optimizer was applied with an exponential decay learning rate scheduler. The initial learning rate was set at 0.001, and it reduces by a factor of 0.9 every 1000 steps.
- **Model Checkpointing:** The model's state achieving the highest validation accuracy during the training was preserved. This allows for the use and further fine-tuning of the best-performing model snapshot.
- **Early Stopping:** An early-stopping mechanism was implemented to curtail overfitting. If the validation loss failed to improve for 10 consecutive epochs, the training process was halted, and the weights from the epoch with the lowest validation loss were reinstated.
- **Training:** The models were trained for several epochs, with class weights applied to the loss function to manage class imbalance. The models' progress was monitored using the accuracy metric.

To prevent overfitting, the researchers employed a technique called 5-fold cross-validation. This method splits the data into 5 parts, trains the model on 4 parts, and tests it on the remaining one. This process is repeated 5 times, ensuring the model's generalizability to unseen data.

2.5 Development

Part of the output of this research is the development of a desktop application called 'Minna Transcriber.' This tool was instrumental in facilitating the analysis of the existing dataset in a more efficient manner. It simplifies the file encoding process and reduces the likelihood of errors during file renaming. Moreover, the application was designed to export datasets in convenient formats such as CSV, JSON, and Excel.

During experiments, Google Colab was used to experiment with methods to enhance the accuracy of the machince learning model. Other development tools that were used include Flutter, for the creation of a high-fidelity prototype for the project, and Appwrite, for the back-end infrastructure.

3 PRELIMINARY RESULTS

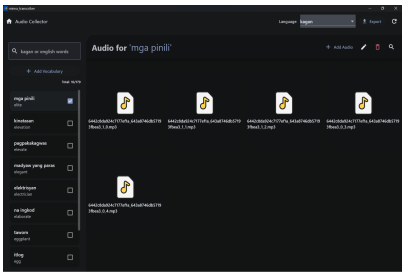


Figure 1: Minna Transcriber Tool

The development of "Minna Transcriber" resulted to streamlining the process of encoding and formatting audio files. This tool effectively prevents issues related to renaming file names while preserving features and target labels, resulting in a more efficient and user-friendly experience. Furthermore, this tool significantly contributed to our data analysis task before pre-processing the dataset.

3.1 Train Model

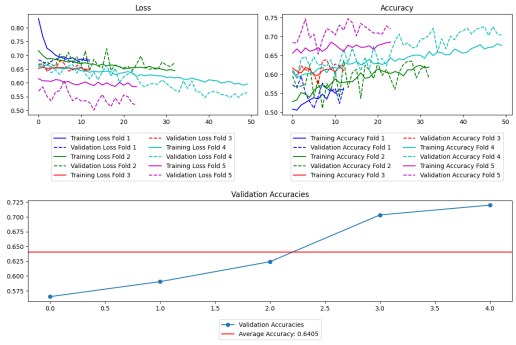


Figure 2: UBLCM001A Results

Looking at Figure 2, the researchers observed a pattern where the training and validation loss decrease while the training and validation accuracy increase as the number of epochs increases. The model was trained for a maximum of 100 epochs but didn't always reach that limit. The researchers implemented an early stop technique, meaning the training stops if the accuracy decreases continuously for five consecutive epochs. When this happens, the model is considered to have achieved good performance, and its weights and validation results are saved.

The model stops training for this specific k-fold training process when the accuracy declines for five consecutive epochs. Then, these saved weights and validation results are used for the next k-fold iteration. This process continues until all five folds are completed.

Figure 2 shows that the model typically reaches convergence around 50 epochs. The average validation accuracy at this point is 64%. If the researchers apply this trained model to new, unseen data (test dataset), it achieves an accuracy of 71%.

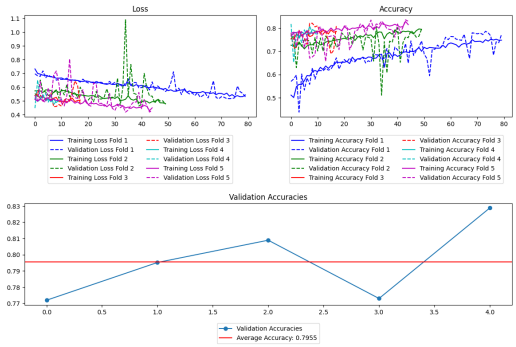


Figure 3: UBLCM002A Results

The results for the UBLCM002A model, as depicted in Figure 3, exhibit a similar trend to the UBLCM001A model. In all five k-folds, the model reached convergence after 80 epochs. On average, the validation accuracy at this point was 80%. When this trained model was evaluated on new, unseen data or test datasets, it achieved an accuracy of 82%.

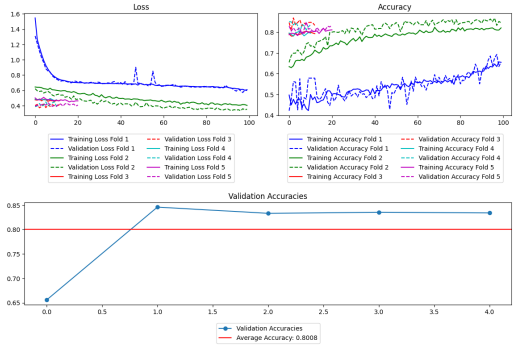


Figure 4: UBLCM005A Results

Referring to Figure 4, the trend observed in the model's performance is consistent with the previous models. However, a key distinction is that for nearly all five k-folds, this model could complete the entire 100 epochs. This indicates the potential for further

tuning by possibly increasing the number of epochs. The model achieved an average validation accuracy of 80% at this stage. More importantly, when tested on new, unseen data, the model's performance was commendable, achieving an accuracy of 83%.

3.2 Unseen Data Testing

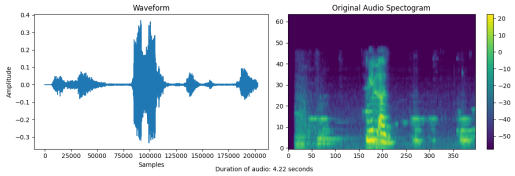


Figure 5: Unseen data: Original Audio Wave and spectrogram

Figure 5 displays the audio wave and spectrogram for the Manobo word "Ngilad". The audio clip is 4.22 seconds long, with the spoken word "Ngilad" appearing between 1.40 and 2 seconds. This data is new and wasn't used when training the model. During preprocessing, the researchers standardized the audio duration to 1.37 seconds. However, important information is missed since "Ngilad" starts later, at 1.40 seconds.

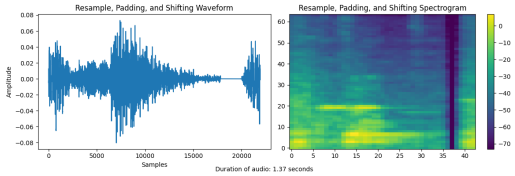


Figure 6: Unseen data: Resampling, Padding/Truncate, and Time Shifting

Figure 6 demonstrates the re-sampling, padding/truncating, and time-shifting process. The audio has been adjusted to a fixed length of 1.37 seconds. However, the audio now mainly consists of noise and silence, and the essential data for prediction has been lost.

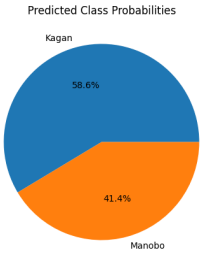


Figure 7: Unseen data: Probability of Prediction

Since setting the fixed duration to 1.37 seconds could result in data loss, it was anticipated that this new, unseen data might result in a poor prediction, as demonstrated in Figure 7.

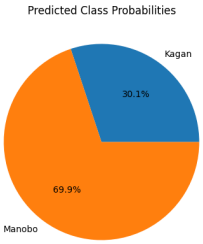


Figure 8: Unseen data: Probability of Prediction (Audio Trimmed)

The researchers noticed a shortfall in the audio pre-processing, specifically that the audio trimming led to the loss of crucial data. To remedy this, the researchers manually trimmed the audio. The prediction improved significantly after this adjustment, as shown in Figure 8. The model correctly classified the word "Ngilad" as belonging to the Manobo language.

REFERENCES

[1] 2011. UNESCO Project: Atlas of the World's Languages in Danger. Programme and meeting document. Retrieved from UNESCO website (Catalog Number: 0000192416).

[2] Francesc Alías, Joan Claudi Socoró, and Xavier Sevillano. 2016. A Review of Physical and Perceptual Feature Extraction Techniques for Speech, Music and Environmental Sounds. *Applied Sciences* 6, 5 (2016). https://doi.org/10.3390/app6050143

[3] Aankit Das, Samarpan Guha, Pawan Kumar Singh, Ali Ahmadian, Norazak Senu, and Ram Sarkar. 2020. A Hybrid Meta-Heuristic Feature Selection Method for Identification of Indian Spoken Languages From Audio Signals. *IEEE Access* 8 (2020), 181432–181449. https://doi.org/10.1109/ACCESS.2020.3028241

[4] DOST. 2022. DOST, UIC layong pangalagaan ang mga lokal na lengwahe ng Mindanao sa tulong ng Language Processing Lab. *Philippine Council for Industry, Energy, and Emerging Technology Research and Development* (September 02 2022).

[5] Zane Durante et al. 2021. Speech Representations and Phoneme Classification for Preserving the Endangered Language of Ladin. *ArXiv* (2021). https://doi.org/10.48550/arxiv.2108.12531 Accessed 20 Apr. 2024.

[6] Tom Ko et al. 2015. Audio Augmentation for Speech Recognition. In *Interspeech 2015*. https://doi.org/10.21437/interspeech.2015-711

[7] Onowa McIvor and Adar Anisman. 2018. Keeping Our Languages Alive: Strategies for Indigenous Language Revitalization and Maintenance. *Handbook of Cultural Security* (May 2018), 90–109. https://doi.org/10.4337/9781786437747.00011

[8] Romylyn A. Metila and et al. 2016. The Challenge of Implementing Mother Tongue Education in Linguistically Diverse Contexts: The Case of the Philippines. *The Asia-Pacific Education Researcher* 25, 5-6 (Sept. 2016), 781–789. https://doi.org/10.1007/s40299-016-0310-5

[9] Annalee Newitz. 2023. How Artificial Intelligence Is Helping Keep Indigenous Languages Alive. *New Scientist* (Sept. 2023). https://www.newscientist.com/article/0-how-artificial-intelligence-is-helping-keep-indigenous-languages-alive/

[10] Ahmed Ouhnini et al. 2023. Towards an Automatic Speech-To-Text Transcription System: Amazigh Language. *International Journal of Advanced Computer Science and Applications* 14, 2 (2023). https://doi.org/10.14569/ijacsa.2023.0140250 Accessed 26 Mar. 2024.

[11] Statista. 2020. *IT Industry: Barriers to Technology Adoption 2020.* www.statista.com/statistics/1239000/it-industry-barriers-to-technology-adoption/

[12] Lindsay J. Whaley. 2011. Some Ways to Endanger an Endangered Language Project. *Language and Education* 25, 4 (July 2011), 339–348. https://doi.org/10.1080/09500782.2011.577221

[13] Guillaume Wisniewski et al. 2020. Phonemic Transcription of Low-Resource Languages: To What Extent Can Preprocessing Be Automated? *shs.hal.science* (2020). https://shs.hal.science/hal-02513914 Accessed 20 Apr. 2024.

[14] Josephine Wolff. 2021. How Is Technology Changing the World, and How Should the World Change Technology? *Global Perspectives* 2, 1 (Aug. 2021), 27353. https://doi.org/10.1525/gp.2021.27353

FrameRL: DNA-Protein Sequence Alignment using Deep Reinforcement Learning

Wai Kei Li
College of Computer Studies
De La Salle University
Manila, Philippines
wai_kei_li@dlsu.edu.ph

Justin Ayuyao
College of Computer Studies
De La Salle University
Manila, Philippines
justin_ayuyao@dlsu.edu.ph

John Carlo Joyo
College of Computer Studies
De La Salle University
Manila, Philippines
john_carlo_joyo@dlsu.edu.ph

Renz Ezekiel Cruz
College of Computer Studies
De La Salle University
Manila, Philippines
renz_cruz@dlsu.edu.ph

Roger Luis Uy
College of Computer Studies
De La Salle University
Manila, Philippines
roger.uy@dlsu.edu.ph

ABSTRACT

Aligning DNA and protein sequences in three reading frames, referred to as Three-Frame Alignment, is a dynamic programming (DP) approach for the alignment between a reference protein sequence and an input DNA sequence. Despite finding optimal alignments with the usage of three matrices, the memory usage of Three-Frame Alignment scales with the lengths of the sequences, making longer reads costly. This paper proposes FrameRL, a deep reinforcement learning approach with an environment based on Zhang's Three-Frame alignment algorithm and agent that can perform DNA-Protein sequence alignment. The resulting approach showed better scalability in memory usage for longer reads, resulting in an overall space complexity of O(1) compared with O(MN) of dynamic programming approaches, with M and N as the lengths of the DNA and protein sequences, respectively.

KEYWORDS

Three-frame alignment, global alignment, deep reinforcement learning, deep Q-learning, agent, environment, dueling double deep Q-networks

1 INTRODUCTION

Sequence alignment is the process of aligning nucleotide or amino acid sequences to identify common regions in order to find commonalities in ancestry, structure, function, and others [1]. Directly translating the DNA and finding the optimal alignments among all other possible amino acid counterparts is computationally heavy; therefore, dynamic programming approaches like the Needleman-Wunsch [5] and Smith-Waterman [8] algorithms were often utilized. Despite finding an optimal alignment, dynamic programming approaches are limited by their space complexity, which is directly proportional to the product of the lengths of each of the sequences. Therefore, these approaches are preferred for shorter sequence reads, as handling and recording large matrices is computationally costly.

Works like the DQNalign [9] and EdgeAlign [3] have already implemented reinforcement learning into sequence alignment, specifically, DNA-to-DNA. DQNalign was the first among the two and proved the viability of using deep reinforcement learning for DNA-to-DNA sequence alignment. It involves converting the sequence alignment task into a sliding windows environment with one window per sequence. Each window has a size of N representing a subsequence of size N. Their agent will then perform alignment on these windows and move them until each of them reaches the end of their respective sequences.

Zhang's Three-Frame algorithm uses three reading frames and three matrices (I, D, and C), effectively detecting and adjusting for frameshifts. The dynamic programming approach utilizes several recurrence relations to find the best score possible from the three matrices. The I matrix looks into the max score given an insertion. The C matrix takes charge of looking for the score of the best scores overall, and the D matrix is for deletion.

Like DQNalign, this paper proposed a deep reinforcement learning approach to DNA and protein sequence alignment to solve the space complexity issues associated with Zhang's three-frame alignment. The proposed method leverages the benefits of reinforcement learning while still attempting to recreate the approach used in Zhang's three-frame alignment. The proposed method was able to distinguish perfect matches and several frameshift errors but still experiences instability for mismatches or substitutions. It was also found to have better memory scalability when more extensive sequences are involved.

Section 2 of this paper discusses the training procedure, the agent and environment, the network architecture, and some experimentations performed. Space complexity, query tests, and agent training progression are discussed in Section 3. Lastly, conclusions and future work are presented in Section 4.

2 DNA TO PROTEIN ALIGNMENT WITH REINFORCEMENT LEARNING

Zhang's Three-Frame approach [12] is a global alignment dynamic programming solution providing the optimal alignment between the DNA and protein sequences. However, the issue with the approach stems from the part where it handles three matrices and its constant traversal for the simple act of attempting to gather the score. The space complexity of such an ordeal is also in O(MN) space, with M being the length of the DNA and N being the length

of the protein. This makes it difficult to scale when the inputs get into millions in length.

The traversal of the three matrices creates a potential pitfall regarding time and space complexity in long reads. The usage of the Deep Reinforcement Learning method, or DRL, aims to solve the problem of scalability due to the neural networks having a time complexity of O(N), where N is the length of the query sequence. The speedup will not be immediately realized due to the huge overhead that the neural network demands. However, it performs better for the space complexity aspect as the total space complexity of the neural network would be O(1) [3]. The overall space complexity would still be O(M+N) due to loading in the reference nucleotide and input proteome, with M as the DNA length and N as the protein sequence length. Still, it is not comparable to Zhang's Three-Frame, wherein the space complexity is still O(MN) multiplied by three due to the I, C, and D matrices.

2.1 Training Procedure

To successfully train an unbiased RL agent, diverse input data sets are necessary to avoid biases associated with limited or overly specific datasets.

The process was initiated by generating a DNA sequence of length 1000. This training length was chosen to expose the agent to as many nucleotides and codons as possible without hindering training time. The first sequence is generated by randomly selecting nucleotides (A, G, C, or T). Then, the mutations were introduced using the JC69 model [2] to the first sequence to create the second sequence. On top of that, insertions and deletions are introduced in both sequences, so the RL agent is exposed to a wide variety of sequences through the Zipfian distribution-based indel length model [7]. Finally, the second DNA sequence is translated into a protein sequence. This is repeated until five sets of DNA and protein pairs are generated.

In addition, random protein sequences were generated for each DNA sequence so that the agent is trained in scenarios wherein the sequences are highly mismatched and where insertions, deletions, and frameshifts occur more frequently. Five training sets were generated to train the agent, each containing one DNA sequence, one directly matched protein sequence, and one randomly generated protein sequence. For convenience and easier tallying and tracking of incorrect actions, the agent's reward system punishes the agent's reward score by -2 when it makes mistakes and rewards it with 0 for every correct action. This reward system has made the agent prioritize looking for and frameshift matches. However, it becomes unstable regarding in/del actions and mismatches. These instabilities also shift the reading frames, which primarily affects the accuracy of the alignment and influences succeeding actions.

2.2 Agent and Environment

The environment mainly consists of the reference proteome of size M and the target DNA sequence of size N, wherein the agent traverses both using respective pointers. The environment is considered a Sequential Partially Observable Markov's Decision Process [10], where it only allows the agent to see a part of the environment because the agent's action will affect the current reading frames. The environment can also be considered deterministic despite having multiple correct actions, and this is due to those actions being sub-optimal.

The main objective of the environment is to recreate the process of Three-Frame Alignment. The environment is deterministic, with many states due to the twenty-one proteins in the codon table and eight of them in a subset comprising the past and current three reading frames alongside their respective proteome. Effectively, 54 billion states (including the stop codon/protein) are possible. Thus, a brute-force approach for both methods is impractical and unfeasible.

Conversely, the agent is a separate entity that interacts with the environment using distinct actions: MATCH (M), FRAMESHIFT_1 (F1), FRAMESHIFT_3 (F3), INDEL, and MISMATCH. The actions F1, M, and F3 pertain to aligning the current protein with the current reading frames 1, 2, and 3, respectively. INDELs, on the other hand, are alignments between the previous protein and any of the current frames or current protein with any previous frames. Lastly, MISMATCH happens when all other actions are not possible; thus, only the substitution of frames is possible. The precedence of these actions is based on the understanding from the simplification of the recurrence relation of Zhang's Three-Frame alignment.

M actions are always preferred, followed by F1 and F3 actions, and then by INDELs before resorting to MISMATCH. This precedence came from calculating the scores in the recurrence relation wherein Matches are favored over Frameshift matches, which are then favored over Indel matches, and which are also favored over Mismatches or Substitution. Depending on the result, these actions move the DNA and protein pointers rightward. The agent is punished if the action it chooses is incorrect, but there are no rewards if the agent is correct.

For the learning stage, the agent employs epsilon-greedy exploration [10], wherein it does random actions when there are higher epsilon values to experience as many states, actions, and rewards as possible and possibly determine and find patterns. These actions eventually update the agent's network, and the epsilon value is decayed over time as the agent's network is further trained.

2.3 Network Architecture

The agent in this paper adopts a Dueling Double Deep Q-Network (DDDQN) architecture similar to the approach used in DQNalign [9] for determining its actions. By leveraging the DDDQN architecture, the agent can expedite its decision-making process and create better learning outcomes. This architecture is particularly beneficial in distinguishing states that have more significance in learning.

The choice of utilizing a convolutional neural network (CNN) in the agent's network architecture is influenced by previous researchers who employed it in their deep reinforcement learning tasks, specifically in DQNalign and EdgeAlign [3]. Notably, the inception of CNNs in DRL can be traced back to Google DeepMind's research [4], which used the Atari games as their environment.

In the subsequent layers of the network, the Dueling architecture was incorporated to effectively segregate the State-action and Action-advantage values, thereby facilitating a more accurate estimation of future Q values by the network.

2.4 Experimentation

2.4.1 Query Experiments. The capacity of the agent to perform alignment was tested on select proteomes from the organism Drosophila melanogaster [11] (Fruit Fly). For the experiment, ten proteomes were selected from the list of Fruit Fly proteomes with a given length of 333. After that, the selected proteins are directly translated into their respective DNA sequences. These DNA sequences will be aligned with several reference proteomes from the previously selected proteomes. A random protein is first selected to simulate a query, and a target substring of length M is also shattered from that protein. After that, the random protein's ID and the translated DNA are recorded. The protein ID will serve as a reference in identifying the parent or origin of the target substring. Afterward, the order of the ten random proteins is shuffled, and the translated DNA is aligned with each protein. By assumption, the protein with the highest alignment score should contain the target substring. This random selection of protein and target substring is tested for each of the ten selected proteins.

2.4.2 Memory Usage Experiments. A synthetic benchmark is performed by generating a DNA sequence of a desired length to serve as the reference. To generate the query, the reference is shattered with mutations incorporated into the sequence based on the JC69 model [2] before being converted to a protein sequence. The RL Agent and a sequential implementation in Python of Zhang's Three-Frame algorithm are then given these sequences to simulate alignment with varying read lengths. The benchmark is performed by aligning on base pair lengths of 10, 30, 60, 100, 300, 500, 800, 1000, 1500, 3000, 4500, 6000, 7500, 9000, 13500, and 15000. For each given base pair length, memory is sampled for every alignment step, which is then averaged to give a comparative memory usage given a base pair length. These results were then compiled, and a graph was generated to visualize the space complexity of the aligners.

3 RESULTS AND DISCUSSION

3.1 Agent Training

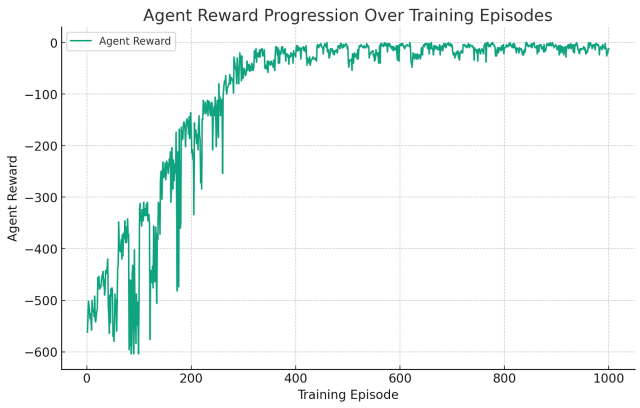


Figure 1: Agent Reward Progression

The training results in Figure 1 show that within the allotted 1000 episodes, the agent could learn the pattern of detecting perfect matches, frameshift_1 matches, and frameshift_3 matches. The first few hundred training episodes include substantial inaccuracy caused by the epsilon-greedy exploration training approach that makes random actions. Eventually, this epsilon value fully decays at episodes 400 to 500. Still, it retains a 1% chance of making a random action to facilitate the exploration even after epsilon has been fully decayed. The random actions prevent our agent from remaining in a local minima while training.

However, in relation to accuracy, even minor random errors in the agent's predictions will cause the reading frames to be shifted. So, in the long term, minor errors like these may cause a massive difference in the overall score but will not affect the result if the given sequences perfectly match.

```
Prev. Frames: _, _, _ => PP: *
Curr Frames: *, M, C => CP: M
Action: 0 (MATCH)
Reward of Action: 0

Prev. Frames: *, M, C => PP: M
Curr Frames: V, L, Y => CP: L
Action: 0 (MATCH)
Reward of Action: 0

Prev. Frames: V, L, Y => PP: L
Curr Frames: T, R, V => CP: R
Action: 0 (MATCH)
Reward of Action: 0
```

Figure 2: An example of the Alignment History

Since the primary approach attempts to mimic the original Three-Frame alignment and account for insertions and deletions, the agent's environment will be given six reading frames: three past frames and three present frames, as well as two proteins: one past protein (PP), and one current protein (CP) as displayed in Figure 2 and with the given information, the agent will predict the correct action to be done.

3.2 Memory Usage and Query Tests

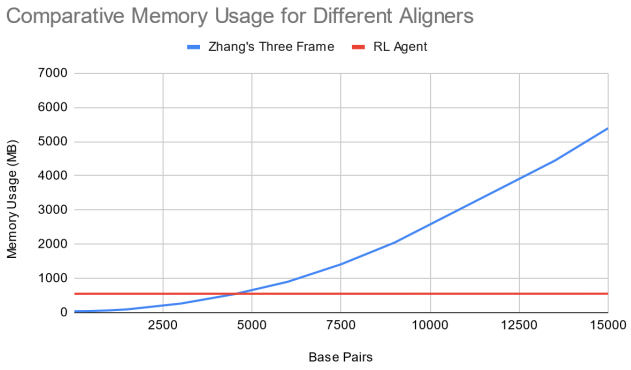


Figure 3: Memory Utilization of the Aligners

3.2.1 Memory Usage Comparison. Figure 3 presents a comparative analysis of the memory usage for the two different approaches in aligning: dynamic programming and reinforcement learning. The comparison was performed across various base pair lengths, emphasizing how each approach scales in terms of memory requirements.

As shown in Figure 3 and Table S2, the agent has a space complexity of O(1) since it only has to load the weights of the Q-network. Memory usage remained constant since the agent was able to take advantage of Python's buffered reader [6]. Zhang's three-frame aligner also takes advantage of the buffered reader, but despite that, the dynamic programming approach has to maintain three matrices, namely the I, D, and C matrix. This results in a space complexity of O(MN) where M is the length of the protein sequence and N is the length of the DNA sequence. For lengths beyond 150 base pairs, typical of long reads, the memory utilization of the dynamic programming approach quickly grows, making it unsuitable for alignment in long reads. Therefore, the agent outperformed the dynamic programming approach regarding memory usage since its space complexity remains constant for long reads. This makes the approach suitable for alignment tasks in devices with a constraint on memory footprint since the agent can perform alignment with a predictable memory utilization pattern.

3.2.2 Agent Query Tests. Table S1 displays the alignment scores for all query tests performed. The target protein represents the shattered sequence or substring of a source protein that the query will try to align with other proteins. The source proteins are also identified via their respective protein IDs. The respective columns represent alignment scores between the translated DNA sequence of the source protein and other proteins in the list. For example, in Row 1, the target protein is FVRIKQSLKP, originating from the protein Q7K1S. When aligning the translated DNA of Q7K1S with another protein whose ID is Q9V3Y7, the resulting alignment score is 79. From the table, it can be seen that the highest alignment scores are from the same protein IDs for each target protein. This further supports the assumption that the protein with the highest alignment score should contain the target substring. This indicates that the agent could fully identify matches between DNA and Protein sequences.

4 FUTURE WORKS

For future works, it is important to note that there are still minor instabilities when the agent encounters situations involving insertion, deletion, and substitution/mismatch. At the same time, further investigation and optimization are required for the incorrect truncation and lack of padding for the last reading frames in the environment. This is because the agent stops performing the alignment when no more current three reading frames are in the DNA sequence. Furthermore, the network architecture and layers could be further tested and optimized, for example, by changing the total number of layers, reconfiguring and replacing convolutional layers, etc.

The reinforcement learning model introduced in this paper is in its initial implementation. As such, it requires further refinement before it could be compared with alignment tools such as BLAST or algorithms like seed-chain-extend. Further refinements would include the optimization of the policy gradients and increasing the model's window size which will allow more reading frames to be evaluated during alignment.

In exploring potential RL models suitable for the deterministic environment of the paper, other models that offer performance enhancements can be considered. One promising model is the Deep Deterministic Gradient Policy (DDGP), which warrants further investigation due to its functionality in environments with continuous action spaces. Given the limitations of Q-learning in utilizing policy gradients effectively, alternatives that optimize through policy gradients could potentially yield a better agent. Additionally, the concept of imitation learning, where another model mimics and potentially outperforms the baseline model by learning from its actions in specific states, presents an interesting avenue for further exploration.

REFERENCES

[1] Stephen F Altschul and Mihai Pop. 2017. Sequence alignment. (2017).
[2] Thomas H Jukes, Charles R Cantor, et al. 1969. Evolution of protein molecules. *Mammalian protein metabolism* 3, 24 (1969), 21–132.
[3] Aryan Lall and Siddharth Tallur. 2023. Deep reinforcement learning-based pairwise DNA sequence alignment method compatible with embedded edge devices. *Scientific Reports* 13, 1 (2023), 2773.
[4] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).
[5] Saul B Needleman and Christian D Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology* 48, 3 (1970), 443–453.
[6] Python. 2024. Built in Functions. https://docs.python.org/3/library/functions.html
[7] Bin Qian and Richard A Goldstein. 2001. Distribution of indel lengths. *Proteins: Structure, Function, and Bioinformatics* 45, 1 (2001), 102–104.
[8] Temple F Smith, Michael S Waterman, et al. 1981. Identification of common molecular subsequences. *Journal of molecular biology* 147, 1 (1981), 195–197.
[9] Yong-Joon Song, Dong Jin Ji, Hyein Seo, Gyu Bum Han, and Dong-Ho Cho. 2021. Pairwise heuristic sequence alignment algorithm based on deep reinforcement learning. *IEEE open journal of engineering in medicine and biology* 2 (2021), 36–43.
[10] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction.* MIT press.
[11] UniProt. [n. d.]. Drosophila melanogaster (Fruit fly). https://www.uniprot.org/proteomes/UP000000803
[12] Zheng Zhang, William R Pearson, and Webb Miller. 1997. Aligning a DNA sequence with a protein sequence. In *Proceedings of the first annual international conference on Computational molecular biology.* 337–343.

5 SUPPLEMENTARY MATERIALS

Target Protein (Shattered)	Source Protein ID	ALIGNMENT SCORE (per protein ID)									
		Q7K1S1	Q9V3Y7	Q9VUJ0	Q9VMX3	Q9W0N1	Q9VXT2	Q9VN80	Q7JVH0	Q8IPF7	Q7KVY7
FVRIKQSLKP	Q7K1S1	1689	79	109	165	122	169	133	133	113	133
LPARLEDNSG	Q9V3Y7	115	1712	140	161	104	129	159	129	127	140
EQRRQRDAVG	Q9VUJ0	119	151	1724	152	104	135	117	125	127	143
LFAYRTEEST	Q9VMX3	170	99	107	1713	103	148	131	110	112	108
MHRLIFEVQQ	Q9W0N1	139	130	109	103	1712	118	152	135	113	147
RLLNLRRHQP	Q9VXT2	88	77	116	146	146	1761	122	141	87	128
NCRQLESLLL	Q9VN80	80	96	92	113	126	149	1725	97	110	99
EDEDFIKSVN	Q7JVH0	100	140	100	156	101	169	140	1681	161	173
ESDQQENSPD	Q8IPF7	105	91	138	125	114	136	121	121	1717	197
KGADELDQAE	Q7KVY7	123	144	102	169	118	137	141	167	143	1672

Supplementary Table 1: Alignment Score per Query Test

Target Protein (Shattered)	Source Protein ID	REWARD SCORE (per protein ID)									
		Q7K1S1	Q9V3Y7	Q9VUJ0	Q9VMX3	Q9W0N1	Q9VXT2	Q9VN80	Q7JVH0	Q8IPF7	Q7KVY7
FVRIKQSLKP	Q7K1S1	0	-26	-24	-34	-28	-22	-36	-18	-30	-32
LPARLEDNSG	Q9V3Y7	-18	0	-40	-22	-18	-28	-16	-14	-28	-24
EQRRQRDAVG	Q9VUJ0	-38	-28	0	-34	-32	-32	-34	-40	-22	-26
LFAYRTEEST	Q9VMX3	-40	-22	-34	0	-20	-18	-34	-32	-28	-36
MHRLIFEVQQ	Q9W0N1	-34	-30	-34	-36	0	-26	-32	-42	-26	-32
RLLNLRRHQP	Q9VXT2	-30	-20	-28	-26	-26	0	-26	-28	-30	-24
NCRQLESLLL	Q9VN80	-20	-26	-36	-20	-32	-28	0	-22	-28	-34
EDEDFIKSVN	Q7JVH0	-28	-26	-26	-24	-34	-24	-24	0	-32	-22
ESDQQENSPD	Q8IPF7	-38	-22	-22	-32	-20	-28	-32	-18	0	-32
KGADELDQAE	Q7KVY7	-30	-28	-18	-20	-28	-28	-16	-30	-30	0

Supplementary Table 2: Agent Reward Score per Query Test

Number of Base Pairs	Memory Usage (Bytes)		Memory Usage (Megabytes)	
	Zhang's Three Frame	RL Agent	Zhang's Three Frame	RL Agent
10	40169472	550031360	40.1695	550.03136
30	40169472	550031360	40.1695	550.03136
60	40427520	550526976	40.4275	550.526976
100	40545657.84	551022592	40.5457	551.022592
300	42661205.33	551025294.5	42.6612	551.0252945
500	46732925.57	551284736	46.7329	551.284736
800	57420523.65	551284736	57.4205	551.284736
1000	65449894.31	551284736	65.4499	551.284736
1500	95004232.9	551284736	95.0042	551.284736
3000	264655709.7	551412417.4	264.6557	551.4124174
4500	538028175.7	551555072	538.0282	551.555072
6000	900794987.2	551586211.6	900.7950	551.5862116
7500	1412761902	551825408	1412.7619	551.825408
9000	2046236231	551840228.1	2046.2362	551.8402281
13500	4441659236	552251106.6	4441.6592	552.2511066
15000	5391433074	552589310.8	5391.4331	552.5893108

Supplementary Table 3: Memory Usage Per Base Pair Counts

Author Index

Institution Index