IMAGE-BASED BAYBAYIN TO TAGALOG WORD TRANSLATOR USING DEEP LEARNING ALGORITHM

A Research / Capstone Project
Presented to the Faculty of the
College of Computer Studies and Information Technology,
Southern Leyte State University

In Partial Fulfillment of the Requirements for the
Degree Bachelor of Science in Information Technology

By
Lourence B. Aure
Jack E. Bihay
Andrey N. Mejares
Marco Eraño P. Pahamotang
Carl John Albert S. Solo

Ms. Jannie Fleur V. Oraño
Adviser

December, 2021

Republic of the Philippines
SOUTHERN LEYTE STATE UNIVERSITY
Sogod, Southern Leyte
Website: www.slsuonline.edu.ph
Email: slsumaincampus@gmail.com, op@slsuonline.edu.ph
Telefax No. (053) 382-3294

College of Computer Studies and Information Technology

APPROVAL SHEET

The Capstone Project Study entitled *Image-Based Baybayin to Tagalog Word Translator using Deep Learning Algorithm* prepared and submitted by Pahamotang, Bihay, Solo, Mejares and Aure has been examined and is recommended for approval and acceptance.

RECOMMENDED:

JANNIE FLEUR V. ORAÑO	**RHODERICK MALANGSA**
Adviser	Research Facilitator

APPROVED by the Committee on Oral Examination with a grade of PASSED on _____

ALEX C. BACALLA
Chairman

GERALDINE MANGMANG	**JAMES BRIAN FLORES**
Member	Member

ACCEPTED and APPROVED in partial fulfillment of the requirements in Bachelor of Science in Information Technology.

ALEX C. BACALLA
Dean, CCSIT

Date: _____

DEDICATION

Above all, this study is dedicated to our Almighty God for giving us guidance, motivations and strengths in doing this whole project.

To the Instructors, who guided and prepared us to reach our dreams in the future.

To our parents and relatives, who gave us all the support and helped us in our struggles and also gave us inspiration in all the good times and bad times.

To the members of this team, who never falter to push through all the challenges that we have encountered in this college life.

To our Capstone Adviser, who gave her time available just to guide us, teach us and give us motivation to continue doing this study.

And lastly to our Alma Mater, Southern Leyte State University that sheltered and molded us to become a person that we are today.

ACKNOWLEDGEMENT

This study will not be made possible without the help and guidance of all the persons behind the proponents back. The proponents wanted to give their sincere gratitude to all of them.

To the very kind and humble Capstone Adviser, Ms. Jannie Fleur V. Oraño who patiently supported, share her ideas and eagerly taught us everything based on her mastery that we can apply in our study.

To the Capstone Instructor for always giving us motivation to do the study and patiently following up for our progress of our study.

And to the friends and classmates who stayed with us day and night easing up our troubles and problems in doing the study.

ABSTRACT

The Baybayin scripts are the Philippines' oldest writing system that gave Filipinos their identity during the pre-colonial period. However, Baybayin is no longer in use having been replaced by a universal writing system. As part of the Filipino cultural preservation, the Philippine Congress passed House Bill 1022, the "National Writing System Act" in 2018 with the goal of reintroducing Baybayin characters as a writing system in the Philippines. Learning and familiarizing Baybayin is a time-consuming procedure and needs expertise in doing so. Thus, this research intends to develop an alternate approach that makes familiarization easier. This project attempts to construct an optical character recognition model that can categorize Baybayin characters using emerging technologies such as deep learning algorithms, machine learning and computer vision. Using a total of 7,360 training samples, the model was able to attain a testing accuracy of 96.87836 percent. This result shows that the model is accurate enough to categorize Baybayin characters. The generated model is embedded in the classification applications and utilized to translate an image of Baybayin word into Tagalog. In addition, this study used OpenCV2 to extract characters from an image using a thresholding algorithm to separate the available features in an image and be able to get the axis of the segmented characters in the image and crop it as a new image to feed into the model. Having such functionality will make it easier for students, professors or any interested individuals to recognize and interpret Baybayin words or characters and will also help them become more comfortable in using such writing system.

Keywords: deep learning, convolutional neural network, ocr, baybayin, segmentation, opencv2

TABLE OF CONTENTS

LISTS OF FIGURES

LISTS OF TABLES

Chapter I
INTRODUCTION

1.1 Project Context

Alibata or the Baybayin is the oldest Philippine script that was created by our Filipino ancestors. This ancient script gives us identity which we can be proud of as Filipinos during the pre-colonial period. We have witnessed or learned the transition, alteration and modification of our scripts and languages ever since we have been colonized by the other countries. While other countries are hailing and using their own languages and writing systems, we Filipinos do not have that kind of freedom instead we are writing in a universal language in which it is prescribe by todays curriculum. Reviving the Baybayin scripts and writings could help our country to revive our cultural heritage again, hailing it as our national writing system will develop our own culture and promote our nationality around the globe.

In 2018, the Philippine Congress has approved the House Bill 1022 which was entitled as the "National Writing System Act", this bill declares the Baybayin script as the Philippine National Writing System [1]. Since then, the Baybayin have become the center of interest for every research and proposal but majority of those researches were focused more on classification and recognition of the Baybayin characters.

Learning the Baybayin written words and sentences is complex, it needs a manual decrypting of every Baybayin characters to unlock its meaning in a plain Tagalog text. Thus, implementing a system that will classify and recognize a single Baybayin script will also be time consuming in decrypting Baybayin words or sentences. To solve the gap of some previous studies we need a more advance system that will solve the time-consuming issues in decrypting Baybayin words.

1.2 Purpose and Description of the Project

Given the difficulty of recreating the Baybayin writing system, this gave us Filipinos both an advantage and a disadvantage. True, this writing style distinguishes our country's uniqueness from that of other nations, but it will also pose a problem, particularly for educators, who will have to undertake extensive training that will take a long time and money to complete. As a result, with

the help of the Baybayin to Tagalog translation system, this project will also help to solve that specific problem. The project's proponents are tasked with gathering available data sets of Baybayin images, which will be used to train the system so that it can correctly categorize a Baybayin character. The system will take a user-supplied image and perform object detection or character segmentation, classifying all of the Baybayin characters that appear in the image. The goal of this project is to provide a translation from Baybayin script to pure Tagalog text. We'll use image processing and machine learning to create an algorithm that will detect all of the Baybayin characters in the provided image, segment them, then classify them separately to produce their full meaning and Tagalog translation.

Objectives of the Project

The proponents of this project aimed to develop a system that is capable of processing a Baybayin characters or text from an image may it be handwritten or digital and translate it into a plain Tagalog text. Specially, this project intends to attain the following purposes:

1. To implement algorithm for extracting Baybayin characters from a digital image.
2. To implement Deep Learning algorithm for training the classification model.
3. To design and develop a user-friendly Graphical User Interface for Image-Based Baybayin to Tagalog Translator, and;
4. To evaluate the accuracy of the system in translating digital Baybayin words or characters to Tagalog.

1.3 Scope and Limitations of the Project

The proponents had to prepare and research the prerequisites needed for this project in order for it to be achieved. The first step is to collect image data sets that are available locally or on the internet, and then to incorporate the Deep Learning Algorithm, which is the Convolutional Neural Network (CNN), into a program so that it can be trained to accurately categorize Baybayin characters. The proponents will create their Baybayin Optical Character Recognition using the algorithm of their choice, rather than using or relying on current Baybayin optical character recognition models. This project will make use of the Python programming language as well as

Keras and Tensorflow tools. The designing and development of the system will be conducted with a time frame of 1st semester of academic year 2021-2022.

The focus of this effort will be on Baybayin to Tagalog words translations. As a result, the system can only recognize characters that belong to Baybayin, and it can only recognize characters that were included in the classification model's training. The system can only translate photos in a horizontal or landscape format that are crisp, with no noises or pixelated details. Furthermore, the system will only allow one image to be processed at a time and will not enable two or more photographs to be processed at the same time.

Chapter II
REVIEW OF RELATED LITERATURE

2.1 Related Literature/ Theoretical Background

With the recent advancement and innovations, machine learning is one of the most powerful technologies in today's world. Every human that uses any technology has benefitted from machine learning. Some of its countless applications can be found in security systems, biometric measurements, software developments, and fraud news detection (Pino, 2021) [2]. One contribution of machine learning that is a continuously developing field is optical character recognition (OCR). OCR is a technology that automatically recognizes characters trough an optical mechanism. It is designed to process and read images that consist entirely text in handwritten or typewritten form (Mithe, 2013) [3].

These Baybayin OCR studies in the literature are based at the character level, indicating its early development (Recio, 2019) [4] employed a three-step detection approach to edges of texts images with Baybayin transcriptions. A CNN model was proposed for Baybayin character recognition with a Visual Geometry Group 16 (VGG16 type network) where they calculated a 98.84% accuracy (Bague, 2020) [5]. These Baybayin OCR studies in the literature are based at the character level, indicating its early development.

Long Short-Term Memory were also proposed for Baybayin character recognition schemes that converts the input to corresponding Latin syllable which gained a 92.9% recognition accuracy (Nogra, 2019) [6].

Studies on Baybayin character recognition have started gaining popularity. The first Baybayin OCR study was done where they have presented a system that reads automatically the Baybayin characters and outputs the equivalent Latin syllables. Their method utilized the freeman chain and coding and line angle categorization for classification, where they have obtained 66.47% and 51.96% recognition rates respectively (Recario, 2011) [7].

2.2 Related Studies

Baybayin Character Recognition Using Convolutional Neural Network, (Nogra, 2020) [8]. The goal of this study is to build a system in which it can classify a hand drawn Baybayin characters. The proponents have used 7000 hand-drawn Baybayin characters for training and found out that the best neural network for this classification will be composed of three convolutional layers with 32 channels, 64, channels and 128 channels respectively using a 3x3 filters. The model yields a 94% accuracy rate using the validation data.

Recognition of Baybayin Handwritten Letters Using VGG16 Deep Convolutional Neural Network Model, (Bague, 2020) [5]. This study proposes a system that can convert 45 handwritten Baybayin characters into their corresponding Tagalog word through convolutional network using Keras. The proponents have used 1500 images from each of the Baybayin character, the pixels have been resized to 50x50 pixels and was utilized for training the system. The overall accuracy for the testing phase is 98.84%.

A Baybayin Word Recognition System, (Pino, 2021) [2] from the University of the Philippines Diliman. Their study relies on a Baybayin character classifier that is generated using Support Vector Machine (SVM). They have proposed an algorithm which can provide a Latin transliteration of a Baybayin word in an image through classifying each character according to its equivalent syllable in a Latin script and finally concatenate each result to form the transliterated word. The system was tested using novel data set and have achieved a competitive 97.9% recognition accuracy.

Having gone through all the mentioned literature and studies, deep learning techniques revealed many successes in previous attempts of their Baybayin scripts OCR studies. These studies inspired the proponents to do better and more innovative system that could overcome their studies with a newly unique model to be developed.

Chapter III
TECHNICAL BACKGROUND

3.1 Technicality of the Project

This study will be implemented as a Computer-Based and Android software as the proponents aimed for this software to be capable of running even without internet connection. This goal will be able to achieved with the help of the model that will be created to train the available datasets so that it can be able to classify Baybayin scripts easily. The proponents will design the software in a PyQt5 to create the computer-based application using Python programming language and Android Studio in Java programming language for the android-based application.

3.2 Details of the Technologies to be Used

In developing the project, the following technology tools will be used;

- Python 3.9
- Tensorflow – a backend framework for running keras.
- Keras – a python library for developing and evaluating deep learning models.
- Numpy – a python mathematical library for computing multi-dimensional arrays.
- Matplotlib – for displaying images and graphs
- os – an inbuilt python package for accessing the file system.
- Jupyter – for editing and running documents that contains code.
- Qt Designer – for designing and building the GUI of a computer-based application
- PyQt5 – Python bindings for the Qt GUI Framework.
- Android Studio – for developing an android version of the project.
- SQLite – for storing tagalog words, this will be used as a database for both android and computer-based application.

Necessary installations of libraries, packages and frameworks will be made in order for this project to be realized and to support the system to be suitable for classification of Baybayin characters for which the trained model will be going to be deployed for actual implementation.

3.3 How the Project Will Work.

The project will be a computer-based and android application in which the user will be ask to provide an image, by providing the image the user can select two options available either uploading an image in a .jpg/.jpeg format or capturing via third party camera applications that has a capability of capturing a clear image of an object that has a Baybayin script written on it. After the image is ready the application will do image processing to provide the Translated Tagalog output.

Before the applications will be made, the proponents will develop their own Baybayin Character Recognition model using the Convolutional Neural Network, the figure below is a Conceptual Framework on how this model will be made along with the process on how the system will work using the developed model.

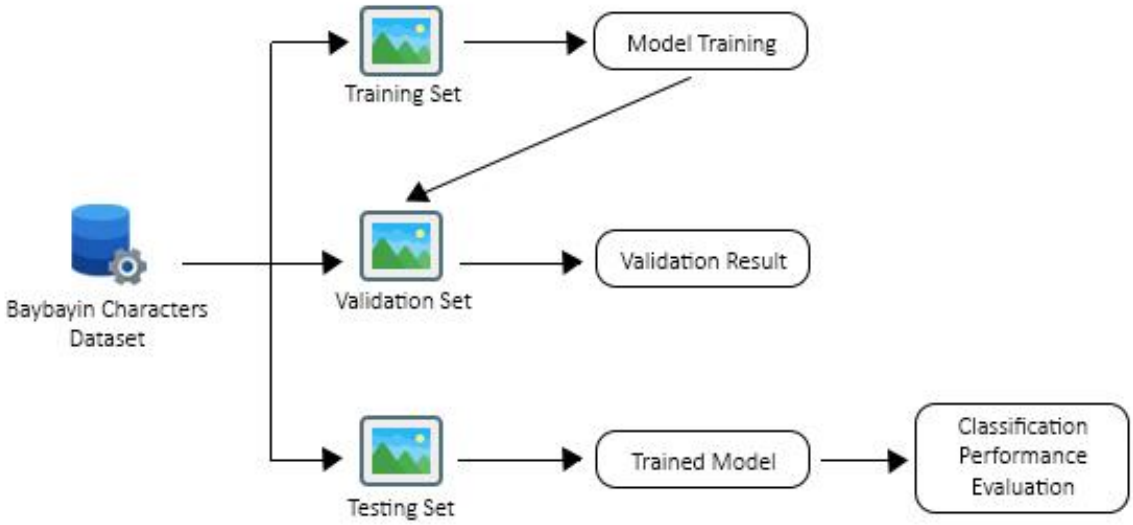


Figure 1. Conceptual Framework of Model Generation and Performance Evaluation

The method of developing the classification model is depicted in Figure 1. The Baybayin datasets will be divided into three sets: training, validation, and testing. Eighty percent of the datasets will be utilized for training, ten percent for validation, and the remaining ten percent for testing. The training and validation sets will be used to train the model, while the testing set will be used to evaluate the model's performance, which will be calculated using all of the matched prediction and mismatch throughout the test.

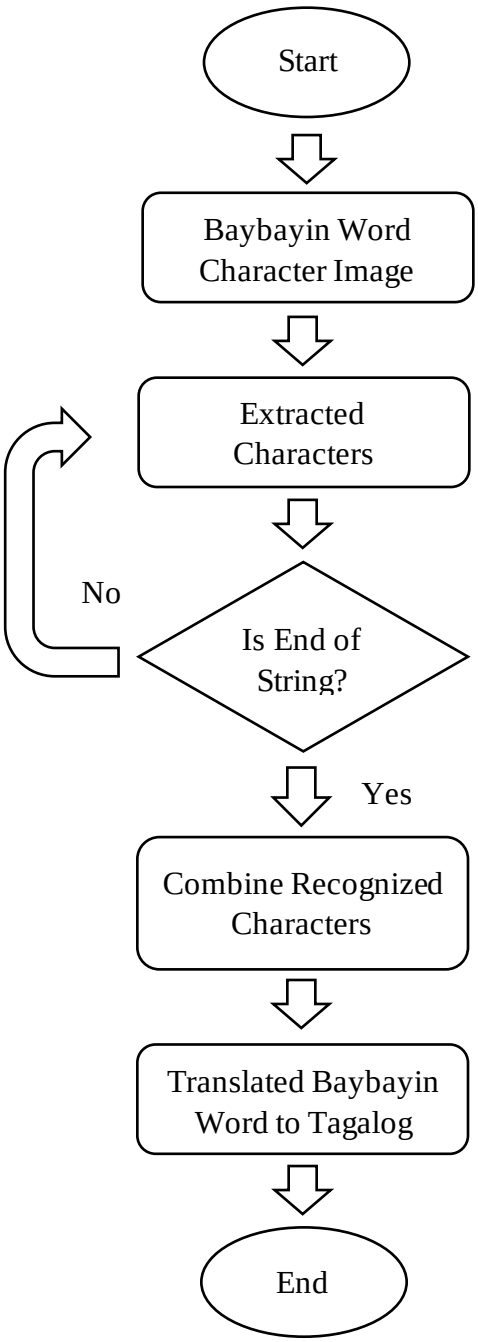


Figure 2. Conceptual Framework of Tagalog Word Translation

Figure 2 depicts the system's classification process for a single Baybayin word. Uploading a Baybayin word picture for character extraction is part of the process. The extracted characters will be fed into the model iteratively until the last character is reached, at which point all of the recognized characters equivalents will be concatenated or combined, and the system will search the database for probable Tagalog equivalents, which will be produced as output of the system.

Chapter IV

METHODOLOGIES

4.1 Requirements Specification

4.1.1 Operational Feasibility

Fishbone Diagram

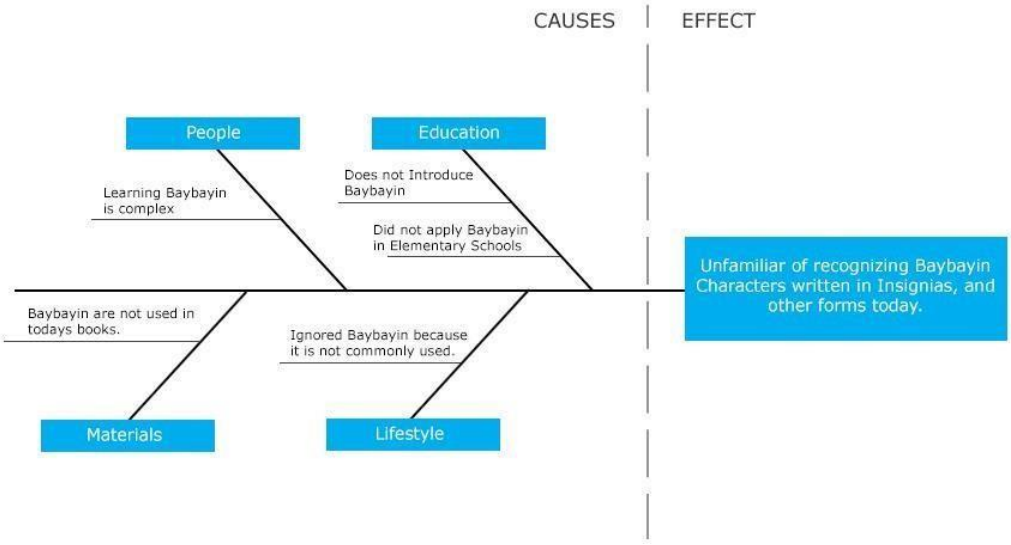


Figure 3. Fishbone Diagram

Figure 3 shows the causes on why we Filipinos are unfamiliar with the Baybayin characters. These following factors affects the performance of people nowadays on recognizing Baybayin characters.

Functional Decomposition Diagram

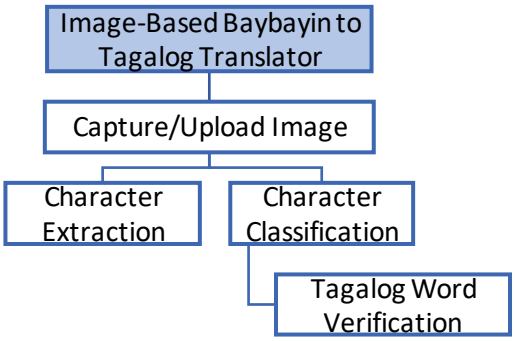


Figure 4. Functional Decomposition Diagram

All of the primary capabilities that will be available in the system are shown in Figure 4. The primary goal of this project is to construct a Baybayin Character Recognition model and an application that can recognize Baybayin words from an image.

Capture/Upload Image

This procedure can be found on the system's main page. This will prompt the user to give an image for the system to process. The user has the option of uploading photographs from the phone's or computer's directories, or using the Android version's camera to take images to use as input. There are several limitations to adding photographs to the system, such as the fact that it should only take Baybayin-related images with good resolution, and that it will not accept inputs that are barely readable or have low resolution.

Character Extraction

This feature will be done with the help of OpenCV2, where the image will be applied an adaptive thresholding technique and will define the region of interest for each character that is visible in the image. The region of interest (ROI) will be the basis of the system to crop certain part of the image and consider it as a new image. These new images will be considered as the extracted characters in the system and will be used for classification.

Character Classification

This feature can be done after the system successfully extracted characters from the input image. This function will individually classify the extracted characters and return a result with the high prediction probability class.

Tagalog Word Verification

This refers to the process where the system will concatenate all of the generated equivalent during the classification. The resulting word will be evaluated to the database to search for existing keywords, if such keyword is found in the database, then the original word saved in the database will be the output of the system.

4.1.2 Technical Feasibility

Compatibility Checking (Hardware/Software)

The computer-based application can be installed in any computer with Microsoft Operating System with a Windows version 7, 8 and 10. The computer-based software can be runnable even without internet connection as long as all of the needed dependencies of the software is also installed in the computer.

The android-based application can be installed in any devices as long as it is android 9 or even higher version of android such as android 10 and 11. The same with the computer-based the android has also the capability to operate offline because all of the needed dependencies are built-in together in the application.

Relevance of Technologies

A computer and an android device are the most needed technology in this project, computer and android is considered as part of the basic necessities of a person hence, deploying this project on such platforms will attract all interested users for this project. Thus, operations and functions of this project will be done in a computer and android devices.

4.1.3 Schedule Feasibility

PROJECT TITLE:	IMAGE-BASED BAYBAYIN TO TAGALOG WORD TRANSLATOR USING DEEP LEARNING ALGORITHM												
Team Bihay													
Pahamotang, Bihay, Solo, Mejares, Aure													
TASK	ASSIGNED TO	PROGRESS	START	END	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
Problem Identification and Requirements Specification	Whole Team	100%	3/15/2021	4/15/2021										
Creating Formal Proposal	Project Manager	100%	4/16/2021	4/30/2021										
Chapter I	Technical Writer	100%	5/3/2021	5/18/2021										
Chapter II	Technical Writer	100%	5/19/2021	6/23/2021										
Chapter III	Technical Writer	100%	6/24/2021	7/10/2021										
System Design	Programmer	100%	7/12/2021	8/23/2021										
System Implementation	Programmer	100%	8/27/2021	11/15/2021										
Testing	Tester	100%	11/16/2021	11/28/2021										
Chapter IV	System Analyst	100%	11/29/2021	12/31/2021										

Figure 5. Gantt Chart

The figure above depicts how this study was carried out, as well as the period and tasks completed by the proponents.

4.1.4 Economic Feasibility

Cost and Benefit Analysis

Expenses	Amount
Internet Expenses	2000.00
Paper and Photocopy Expenses	500.00
Transportation	500.00
Miscellaneous Expenses	450.00
TOTAL	3450.00

Table 1. Cost and Benefit Analysis

Table 1 reflects the lists of the total expenses or costs incurred for the creation of this project. The table reflects that Internet usage has the highest expense due to the proponents' need to use internet to research related articles that can be helpful for this project.

Cost Recovery Scheme

Expenses	Aug	Sept	Oct	Nov	Dec
Internet Expenses	400	400	400	400	400
Paper and Photocopy Expenses	0	0	0	200	300
Transportation	100	100	100	100	100
Miscellaneous Expenses	90	90	90	90	90
TOTAL	590	590	590	790	890

Table 2. Cost Recovery Scheme

Table 2 reflects the division of expenses that is reflected in Table 1 in order to gradually pay the costs incurred upon the creation of this project. The costs of expenses also gradually increase every month this is because the project requires lot of requirements in order for this project to be achieved.

4.1.5 Requirements Modeling

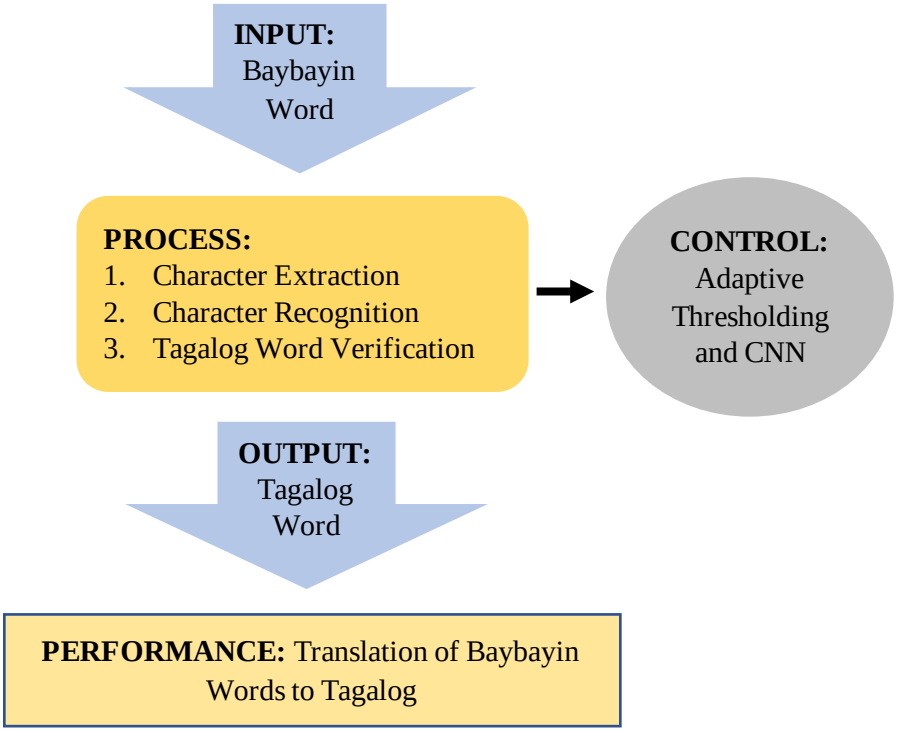


Figure 6. Requirements Modeling

Object Modeling

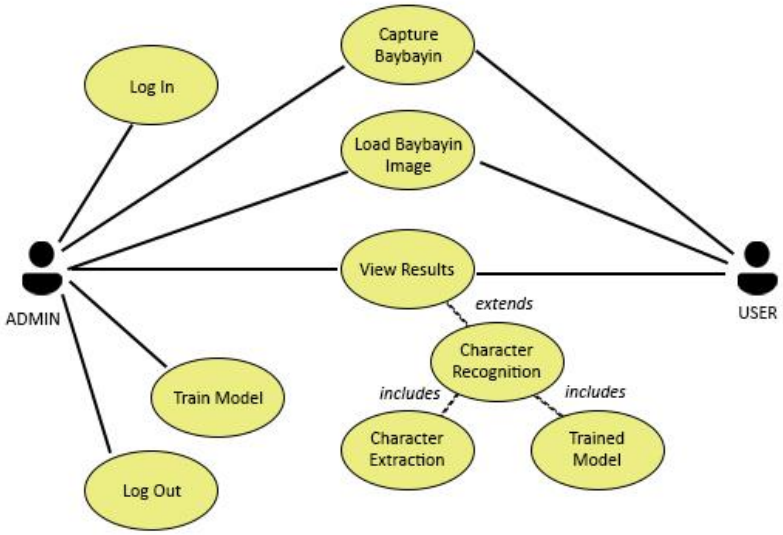


Figure 7. Use Case Diagram

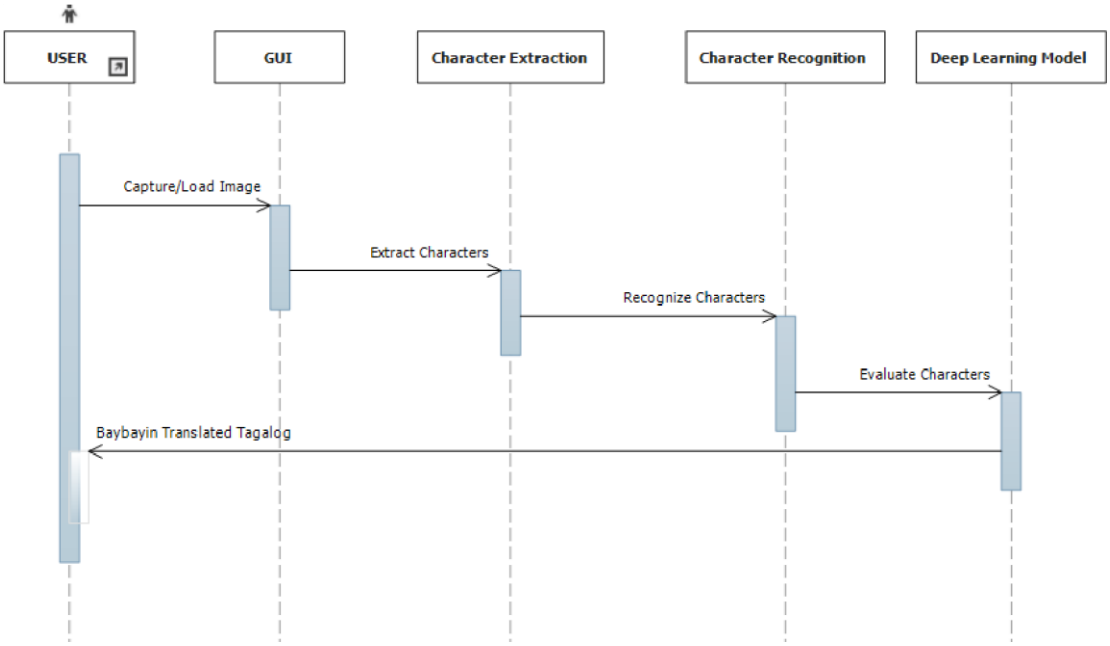


Figure 8. Sequence Diagram

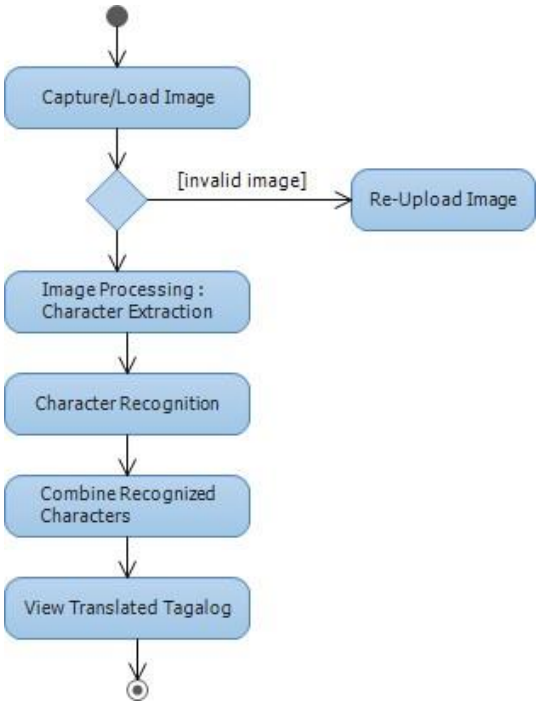


Figure9. Activity Diagram

4.1.6 Risk Assessment Analysis

Risks	Possible Conflict	Mitigation Method
Not Baybayin related image input	There would be a possibility where the system will give inaccurate result to the user.	Applying constraints to the accepted input of the system.
Fancy written Baybayin characters	Fancy written Baybayin characters such as tattoos can also give inaccurate result to the user.	Applying warning message to the user that the result might be inaccurate due to fancy characters.
Blurry or pixelated image input	Blurry images may result to difficulties of retrieving characters, segmentation process may fail due to unwanted details that may possibly still visible during thresholding.	Monitor the numbers of retrieved objects and show some warning if the number of retrieved data from the segmentation process is more than expected.
Tagalog words that are created is not in the database	There might be some instances that other Tagalog words is still not saved in the database and will result for the program to don't display output.	Pull off some warning messages that the generated word is not in the database but may still be correct.
Baybayin characters extracted were not part of the trained model.	There are lot of Baybayin characters version including modern type of characters but those were not included in the training and may result to failure in prediction.	It might not be part of the trained model but it would still return prediction but not accurate hence, the best method we can do is display warning messages that the output may not accurate.

Table 3. Risks Assessment Analysis

4.2 Design

4.2.1 Output and User Interface Design

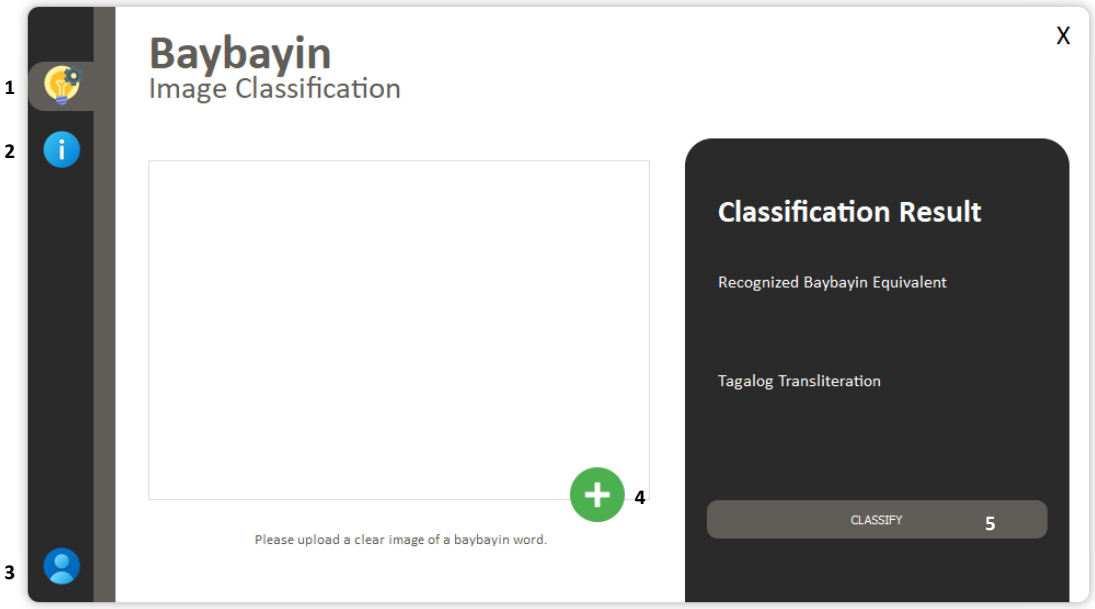


Figure 10. Computer Based Graphical User Interface

The figure above illustrates the interface of the computer-based application. The figure was reflected with digits that are the following;

1. Home Button, this button has the function that will redirect the application to the main activity or the home page which was same as reflected above.
2. User's Guide Button, this button has the function that redirects the application to the Instructions page (figure11) which contains specific instructions in using the application.
3. Sign In Button, this button will redirect to the administrator security feature (figure12). If credentials provided is correct the page will be directed to administrator mode.
4. Add Image Button, this button allows the user to upload Baybayin image in the system, by clicking this button the user will be redirected to the file explorer to select an image.
5. Classify button, this button will process the image to produce the output.

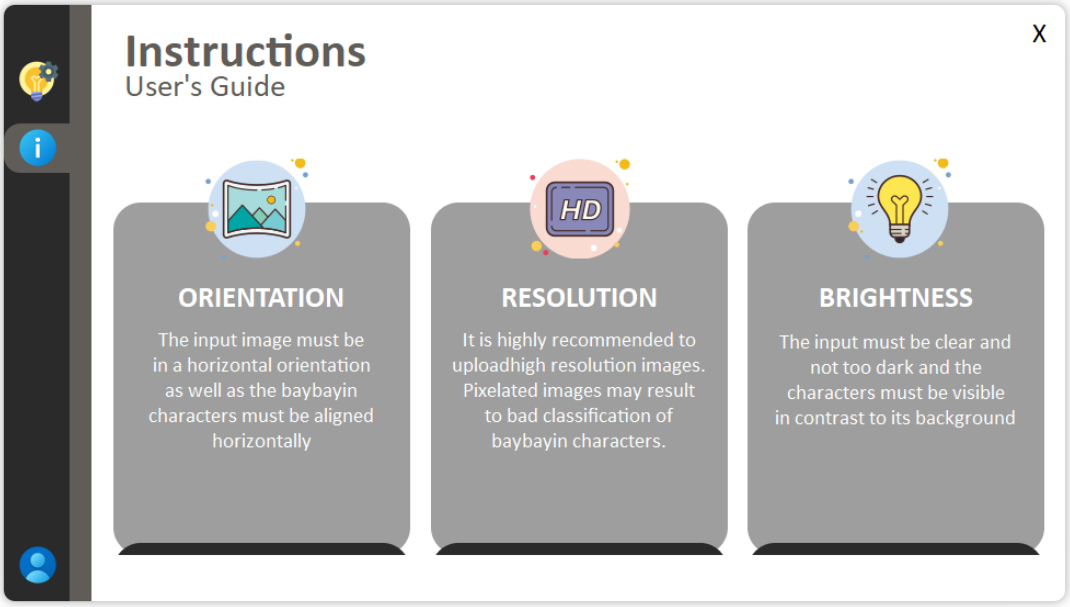


Figure 11. Computer Based Instructions GUI

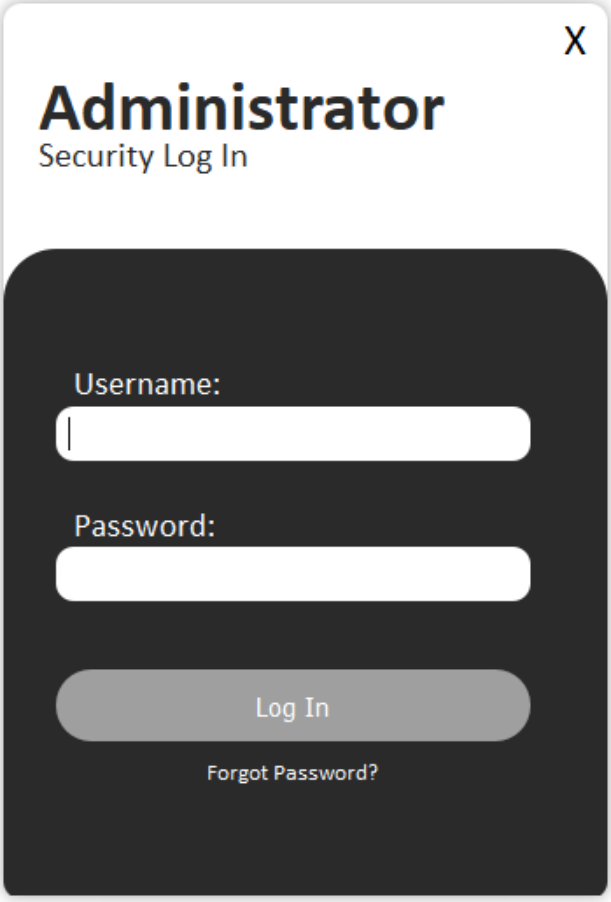


Figure12. Administrator Security GUI

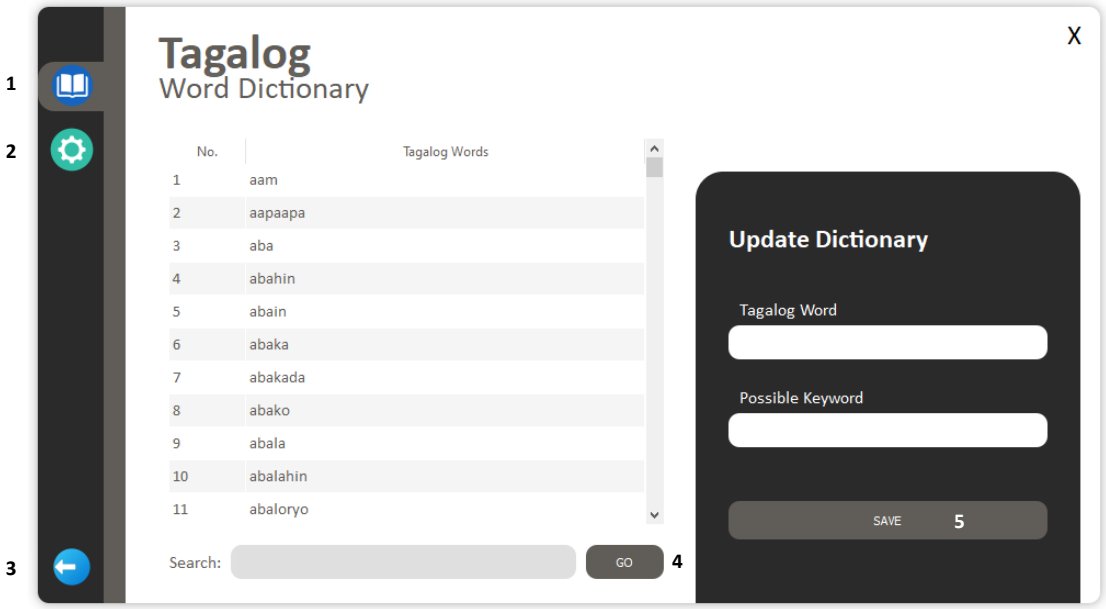


Figure13. Administrator Home Page

Figure13 illustrates the interface for the administrator mode, it has a new set of icons on the side and new buttons which has the following functions;

1. Administrator Home Button, this has the function to switch back to return back to the administrator page.
2. Settings Buttons, this button will redirect to Settings page (figure14) where the administrator can change its username and password in that page. The settings page also contains two buttons that will help the user to confirm its changes to the credentials.
3. Sign Out button, this button can redirect the program to the user-level application where the classification page can be seen (figure10).
4. Search Button, this button can search specific words that is given from the text field and reflect the result to the table that is visible in the page. The table will reflect tagalog words that is only saved in the database otherwise if such a word is not existing in the database, the table will be leave as no rows present.
5. Save Button, this button can save new tagalog words to the database, if there are words that is not existing in the database the administrator can add those words to the system.

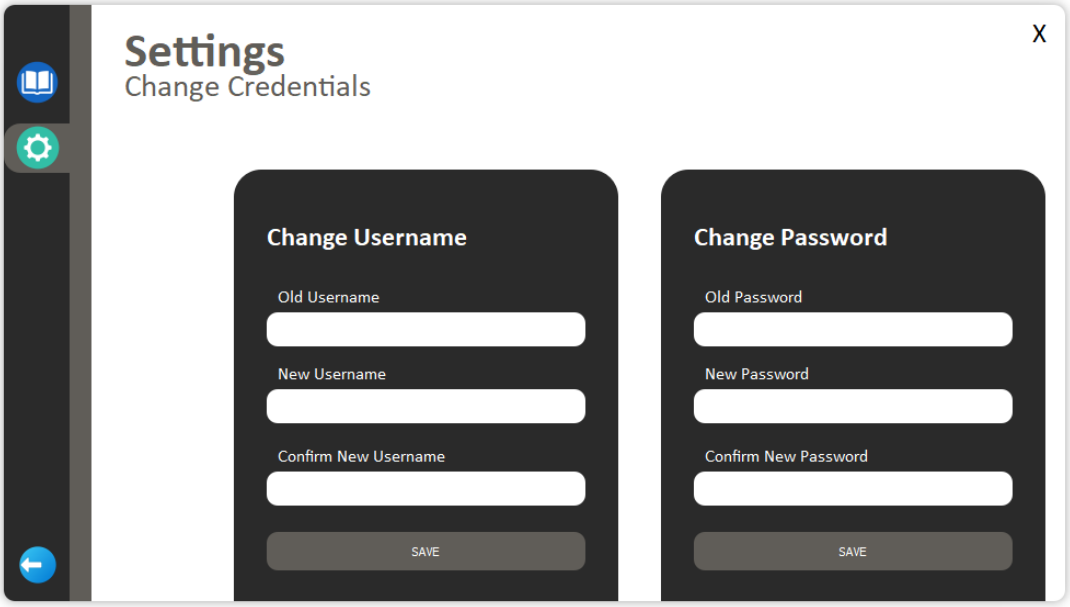


Figure14. Settings Page

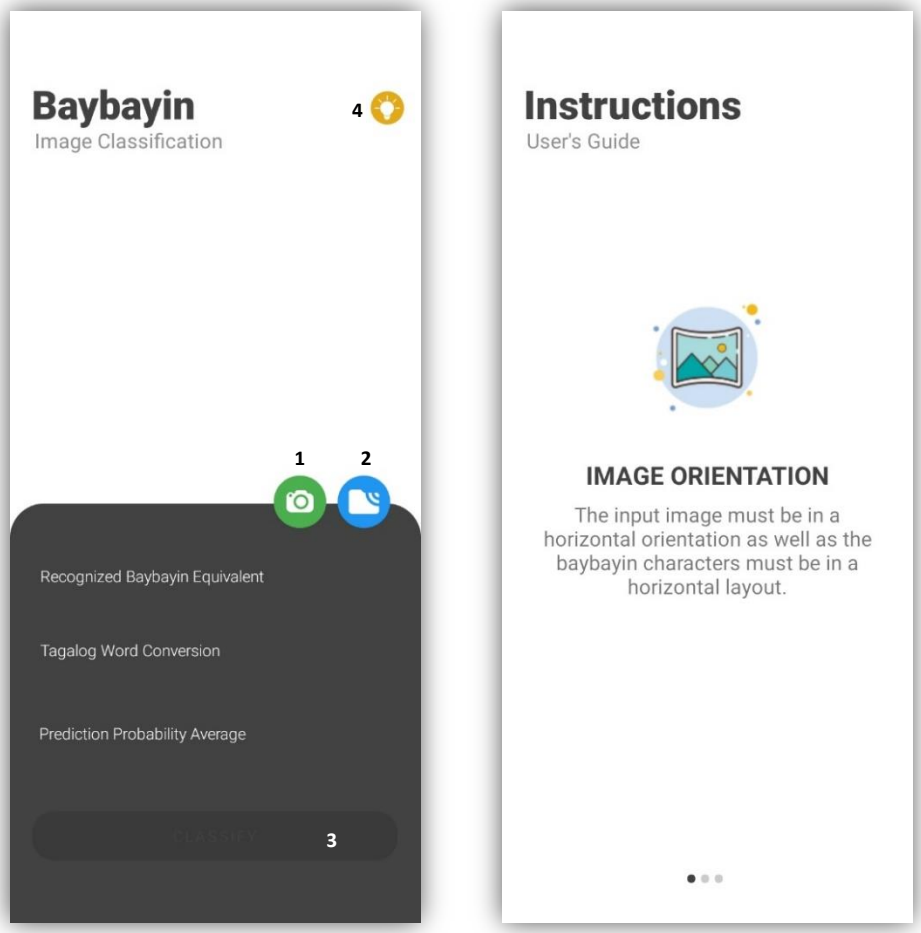


Figure15. Android Graphical User Interface

Figure15 illustrates interfaces of the android version of the software application it has almost the same functionality as the computer-based application but with different icon or button indicator. The controls that are present in the android application are the following;

1. Capture Image button, this button will redirect the application to the phone's default camera that will be used to take a photo for a user input. Any captured images in the camera will be evaluated if it is a Baybayin related to be considered as a valid image input.
2. Upload Image button, the same as the computer-based this button will redirect the application to the phone's file manager to select a specific image. The process will only accept image files and discard any other file format.
3. Classify button, this button can only be enabled if the user already inputted a valid image to be processed and produced an output.
4. User's Guide Button, an android version button that can redirect to the Instructions page where some basic user guides are located.

4.2.2 Data Design

Entity Relationship Diagram

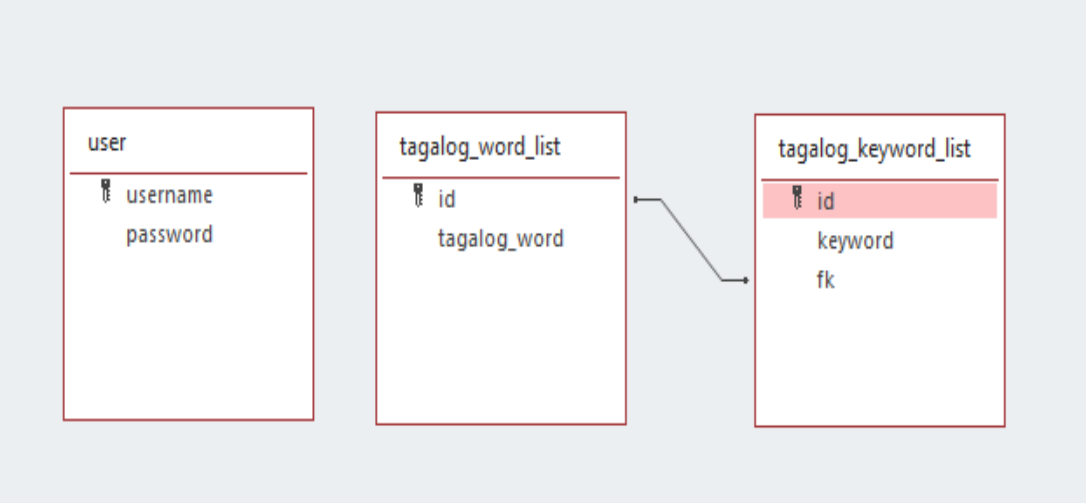


Figure16. Entity Relationship Diagram

Data Dictionary

Keywords	Description
user	This term is the table name for containing the credentials of the administrator.
tagalog_word_list	The term use for the table for name that stores all of the Tagalog words.
tagalog_keyword_list	Table name for the database table that stores all of the created keywords for some Tagalog words.
id	This is the unique identifier of each table.
keyword	This is the field where keywords were stored. Keywords are created for the tagalog words that has a vowel letter /e/ and /u/. Example is akademya, since this word has letter /e/ on it thus it should have a keyword “akadimya”
tagalog_word	This field is where the normal tagalog words will be stored.
fk	This field serves as the foreign key of the table. The data saved in this field must reflect to the unique identifier of the tagalog_word_list table.

Table 4. Data Dictionary

4.2.3 System Architecture

Network Model

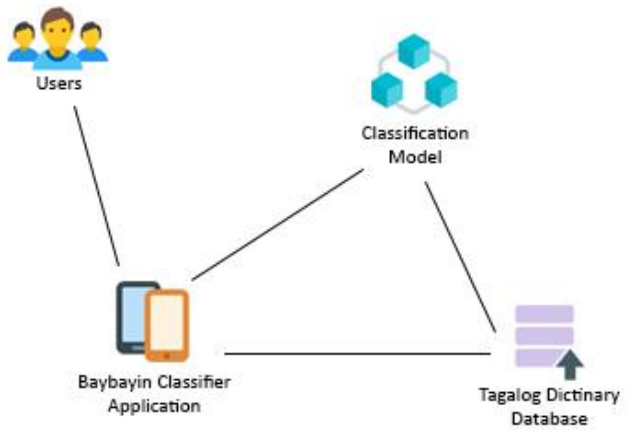


Figure 17. Network Model

Figure 17 represents the relationship of the how does the user can access the application, the figure represents on where do the user interacts with as they use the application.

Security

This project does not require intensive security feature because this was intended to be accessible by all of its users. The functionality of the application of this project is just simply requiring the user an image input and the application will only process the image and generate output. There would be an administrator part of the computer-based project but minimal security is enough for it because the feature that would be available for the administrator is just updates the dictionary of tagalog words that is saved in the database.

4.3 Development

4.3.1 Software Specification

The software that was been used for the development of this project were:

1. Jupyter Notebook, this was used for coding and training the classification model, this was also used during the testing of the model as well as converting the keras model to tensorflow lite for android deployment.

2. VS Code, this software was used for editing the python codes for the application since this software is very convenient to use due to its free extension components that will make coding easier.

3. DB Browser (SQLite), this software was used for creating the database, since we are dealing with computer-based and android projects we used SQLite because this can be accessible by both python and android.

4. Qt Designer, this software was used to design the Graphical User Interface of the computer-based application, this was easy to use because of its drag and drop and customize feature.

5. Android Studio, this software or IDE was been used for creating the android-based application of this project.

4.3.2 Hardware Specification

The hardware components that were used for creating this project were:

1. Laptop
   Intel Core i5-5300U CPU @ 2.30GHz
   4GB DDR3L RAM
   Windows 10 Pro

   This was used for coding and testing the computer-based application

2. Smart Phone
   Qualcomm Snapdragon 665
   Android Version 10
   5.5 Screen Size

   This was used for testing the android-based application of the project.

4.3.3 Program Specification

This project is expected to generate a Tagalog word output from a Baybayin word image that will be uploaded by the user, beforehand this project will do lots of processes to achieve that goal.

This will be able to convert the image input into binary image and apply adaptive thresholding to the image to extract all of the Baybayin characters that is present in the image.

This project will also utilize the trained model from a convolutional neural network that will be used for classification of the Baybayin characters and after the characters has been successfully classified, the system will verify the result of the classification from a tagalog dictionary that is saved in a database.

4.3.4 Programming Environment

Front-End

The front-end of the computer-based application uses Python programming language with PyQt5 and PySide2 for coding this project. PyQt5 enables the codes from Qt Designer to be bind in a python project, PySide2 gives the project to have modern user interface in which users can easily interact.

The android application has its front-end codes that was generated from the Android Studio using Java programming language, in developing the front-end of the android application it also requires some dependencies that are coded in the build gradle of the project.

Back-End

The main back-end programming environment of this project is python with tensorflow and keras for the classification model, the keras model was deployed in the computer-based application and the tensorflow lite model was deployed in android application. To achieve the goal that this project could extract characters, this project also used OpenCV for implementing image processing or thresholding for feature extraction in order for the application to extract all present characters in the input image.

This project also utilized a SQL Lite database using local server to store the most common Tagalog words that will be used to verify the keyword that is constructed during the classification, generally speaking the database will serve as reference for the output generation.

4.3.5 Deployment Diagram

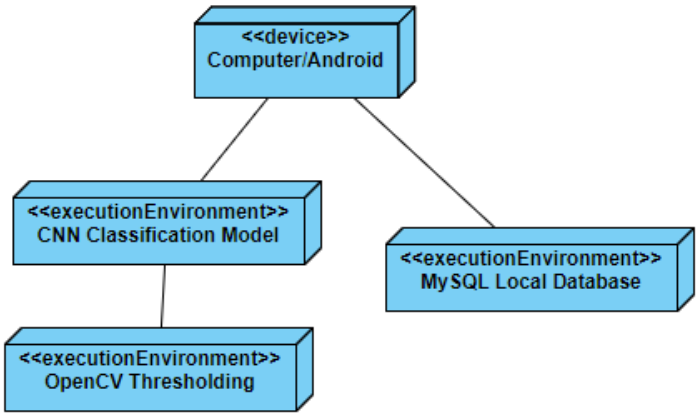


Figure18. Deployment Diagram

4.3.6 Test Plan

TYPE OF TESTING	ACTION PLAN	DURATION
Unit Testing	Test the system's individual units or function such as the model performance and segmentation process	Within October 2021
Integration Testing	Test the integration of the Graphical user Interface and the model.	Within October 2021
Compatibility Testing	Test the application running capability on different platforms.	Within November 2021
Performance Testing	Test the performance of the system in terms of sensitivity, reactivity and stability under a particular workload.	Within November 2021
Stress Testing	Test the robustness of the software beyond the limits of normal operation.	Within December 2021
Load Testing	Test the usage of the software program by multiple users accessing the program concurrently.	Within December 2021
System Testing	Test the complete and fully integrated software product.	Within December 2021

Table5. Test Plan

4.4 Testing

4.4.1 Unit Testing

Evaluating or testing the classification model is a most important test that is needed in this project because everything in this project depends on the quality of the classification model that will be used for classification. The model was created from a total of 7,360 training samples coming from 63 different classes, these classes are based from the Baybayin b18+ra version in which classes are from the native abakada, it was combined together with a 945 validation samples and was trained in a total of 500 epochs. The result of the training gained a total of 0.998 or 99% highest training accuracy. Figure 19 shows a graph of the classification model training plot. It was observed in the graph that the training and validation accuracy plot was maintaining its progress which is close to 1.0.

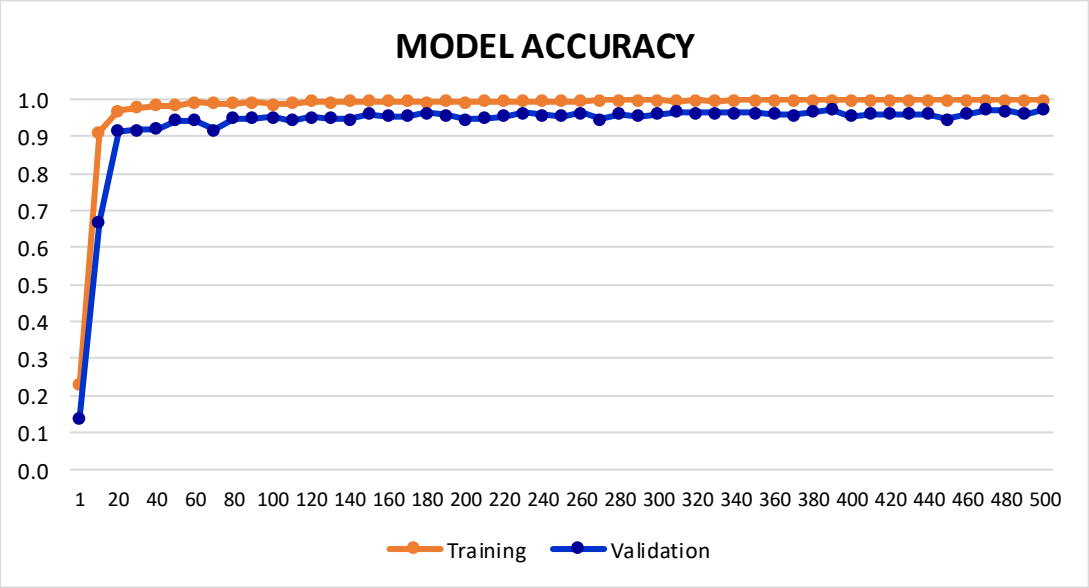


Figure 19. Model Accuracy

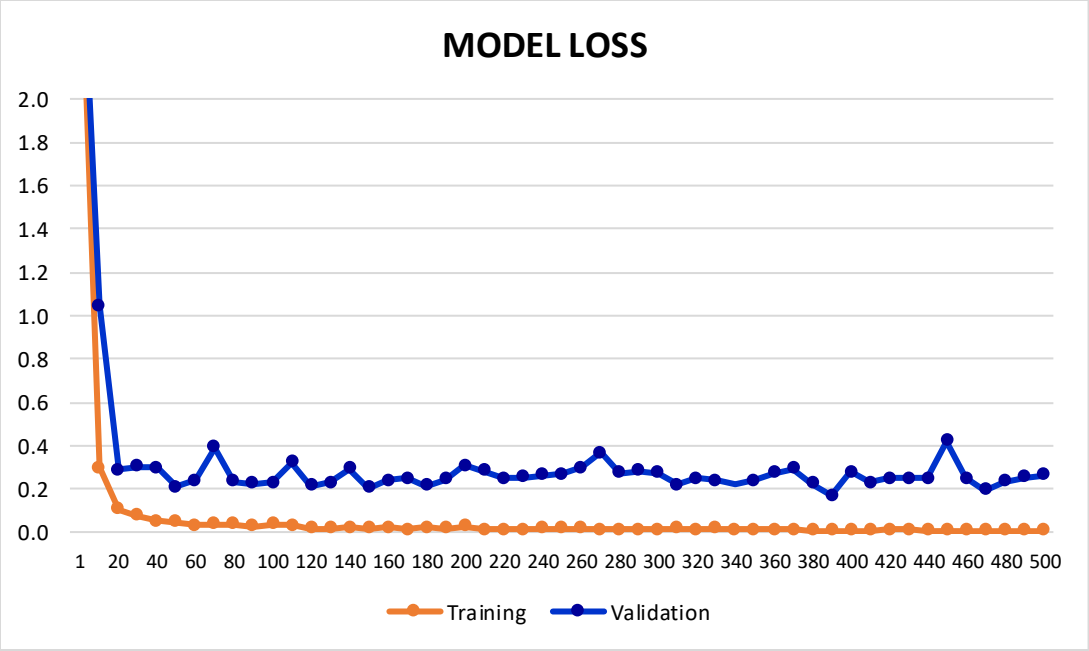


Figure 20. Model Loss

Besides of all this plots it is important to validate assumptions regarding the model's training performance. Hence, the model should be tested by loading the keras model to a testing process (see appendices for codes). A total of another 928 test samples was used for testing the model and it generates a comma delimited file (csv) which contains the lists of predictions of the testing samples, the result (appendix E) was counted in all of its mismatched data samples during the testing and out of 928 test samples there are only 29 samples that have inaccurate prediction result.

$$testing\ accuracy = 100 - (\left(\frac{mismatch}{test\ samples}\right) * 100)$$

To calculate the testing accuracy a formula was provided above, utilizing the formula where mismatched samples is equal to 29 over the total test samples that is 928, the testing accuracy results to 96.87836 percent. Comparing to the previous trials of creating the model this was the highest achieved testing accuracy. Hence, it was concluded that the model was good enough to be integrated to its graphical user interface.

By using the model, the application was able to classify single character Baybayin images, but one of the objectives of this project is to develop an application that is capable of extracting Baybayin characters and be able to translate a Baybayin word. Using OpenCV, the application was able to extract features that is available in the image, its process includes converting the image to grayscale image and apply adaptive threshold that will separate the features and locate the axis of the features extracted. Figure 21 shows the sample result of an image with its extracted features.

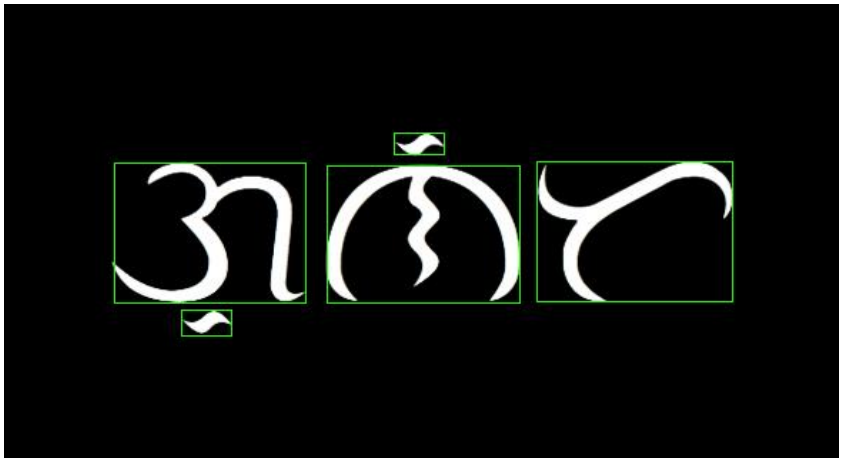


Figure 21. Initial Character Segmentation

The baybayin word in the image consists only of 3 characters which is *gu-ni-ta,* but the segmentation process extracted five features in total because the *kudlits* of some characters are extracted as a different feature as they were separated with few spaces that results to be considered as a different feature. Having 5 features to be feed into the model will result also to 5 characters output thus, it is expected that it will generate a wrong output. The problem has been analyzed by the proponents and came up with an idea that the features extracted will be sorted by its x-axis, where x-axis is the starting point of a feature and added by its width will be the end point of the feature, any feature that will overlap in the said feature will be considered as a *kudlit* of the feature that was overlapped. Figure 22 shows a sample result of feature extraction added by some conditions stated above.

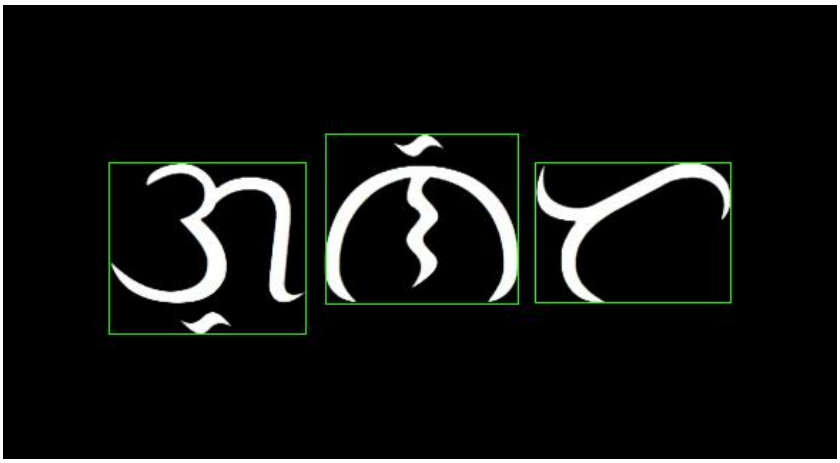


Figure 22. Final Character Segmentation

After adding some conditions to the character segmentation process the result now meets the requirements in which the features in the image were extracted completely, the system can now crop this feature and feed it to the model.

4.4.2 Integration Testing

Integrating the classification model to the Graphical User Interfaces was the next phase of developing this project, it is needed to ensure that integration of the model should be successful. The model was tested as it was integrated with its GUI, a set of testing samples was selected from the 928 test samples during the model performance evaluation. The test samples were uploaded to the GUI manually and it was observed that the results are similar to the performance during the model evaluation hence, it was concluded that deploying the model to a computer-based GUI is successful.

This project includes android version graphical user interface as stated in the previous chapters. Android Studio does not accept keras model (.h5) as its machine learning algorithm but it does support tensorflow lite models (.tflite), to integrate the model to android studio it is needed to convert the keras model to tensorflow lite model to achieve this it just needs to run a set of codes for converting (see appendices for codes). After the model has been converted, the application was tested in its classification ability and the result has slight difference from the original keras model. The model was then tuned to its better prediction ability in android devices.

4.4.3 Compatibility Testing

The application was tested in different computer and android specifications to monitor its capability to operate in different hardware devices. The software application was deployed or installed in the following devices:

Device Type	Model	Specification
Laptop	Dell Latitude E7250	Windows10, IntelCore i5
Laptop	Acer	Windows10, IntelCeleron
Android	Realme 5	Android10
Android	Realme 7	Android11
Android	Samsung A10	Android10
Android	Oppo A74	Android11

Table6. Compatibility Test on Devices

Deploying the software application to the devices stated above was successful and there was no complication in the software and it works in its expected performance. In cases of android devices, the software application will be compatible only in android version 9 up to the latest version this is due to the project minimum SDK was set in android version 9.

4.4.4 Performance Testing

Testing the software application's reactivity, sensitivity and stability was also done in this project. In this test the proponents gathered various Baybayin word images from the internet and created some handwritten Baybayin words to be used as test sample that will be uploaded in the software application to test its overall performance. Testing the performance of the computer-based application and android application in the images that was taken from the internet in which the characters were digitally made was easily recognized and produced accurate output.

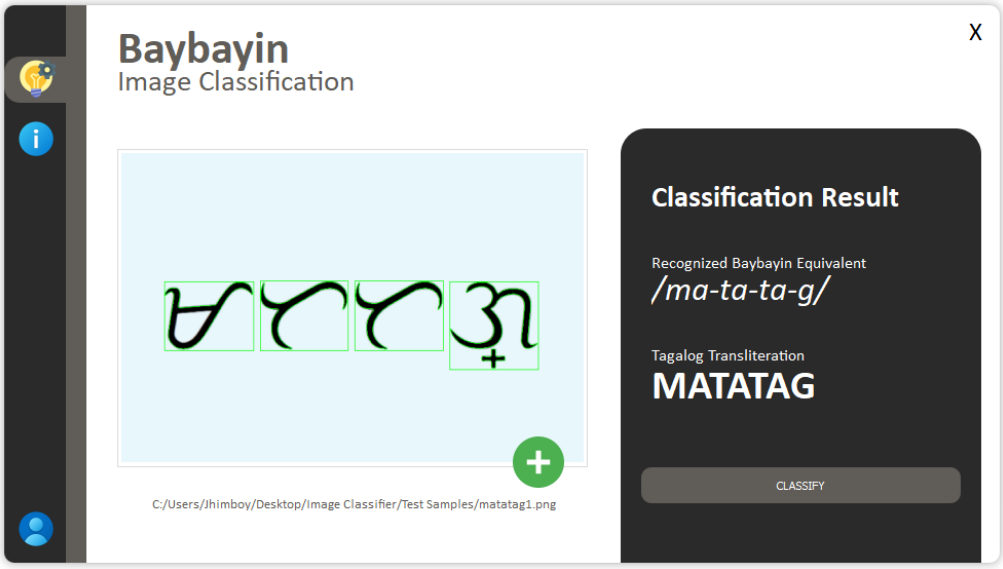


Figure23. Computer-Based Digital Image Sample Output

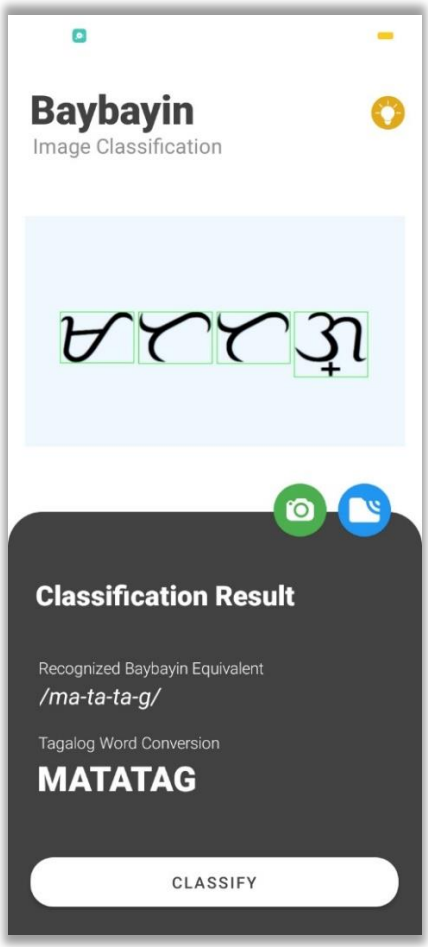


Figure24. Android-Based Digital Image Sample Output

The figure 23 and 24 shows a sample output of character recognition from an image that was taken from the internet, it was able to give the expected output from the image. Digital images were easy to be classified because the characters were well-defined, in this case it is also needed to observe the performance of the software application when dealing with handwritten images because handwritten Baybayin characters may have different stroke to the actual Baybayin characters.

The proponents of this study improvised and write a Baybayin word from a scratch of paper and took a picture of it and was later on uploaded to the software application. Figure 25 shows a sample output of the handwritten Baybayin word that was uploaded into the application, it was able to give a result of a word *kulungan* which was accurate to its Baybayin characters equivalent.

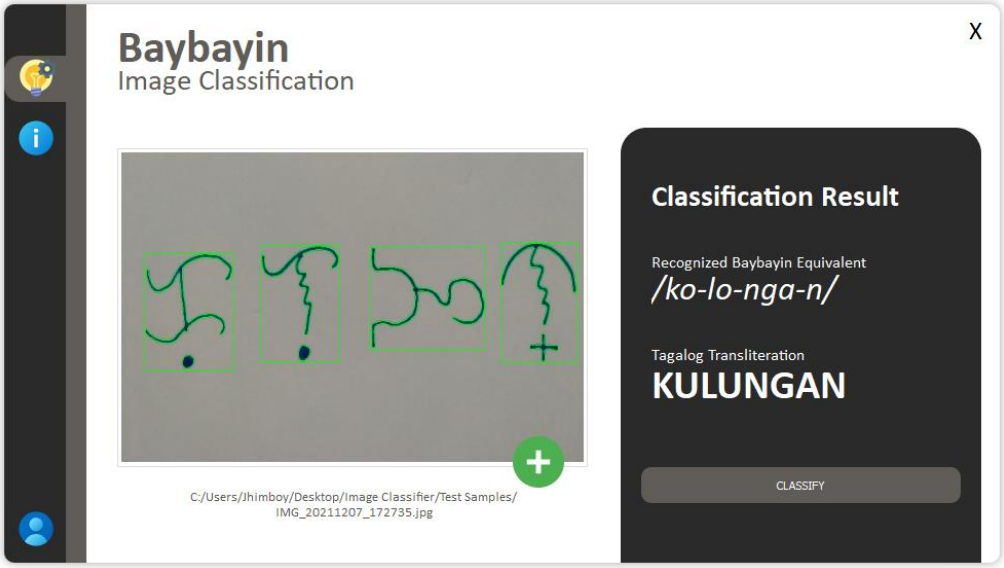


Figure25. Handwritten Sample Output

Doing this kind of test made the proponents convinced that the software application was a huge success. Being able to classify words either from a digital form of Baybayin characters or in handwritten form it was able to give its expected output or result. Even though there were times that the application fails to recognize some characters due to the image quality the test is still in favor for more accurate result than inaccurate result.

4.4.5 Stress Testing

The test for the software application overall performance was a success but it is also important to test the capability of the system when it was used beyond its expected usage. Hence, the software was tested by loading different kinds of images that is not related to the expected input such as image of an architecture, group picture, selfie and many more of its kind, as the image was uploaded to the application there were images that was considered as invalid input, meaning it was not accepted by the application but there were also images that was not related but was accepted with a warning indicator of invalid tagalog word result to emphasized that the result may not be totally accurate.

This method such as not accepting inaccurate input images was achieved with the help of the thresholding of the input image in which the image will not be accepted if it has many different features extracted during the thresholding or segmentation process.

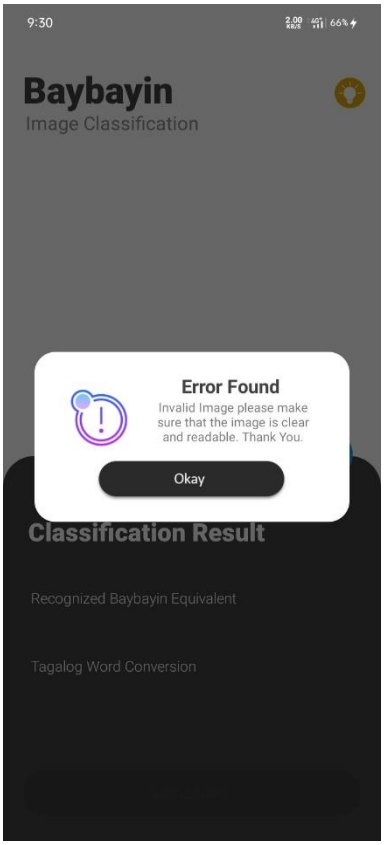


Figure26. Sample Error Message

4.4.6 Load Testing

Testing the usage of the software program by simulating multiple users that will be accessing the program concurrently is part of the testing plan. However, this software is deployed independently in every hardware component such as mobile devices or computers, this does not have any shared client-server model that will affect its performance. Thus, this software will not have any problems in its loading when accessed by multiple users.

4.4.7 System Testing

This project undergoes different testing phase, there was a total of six testing methods that was stated in this study which includes all the necessary feature of the system. With all of the test which was all successful then it can be concluded that the overall system testing was also successful. The system was tested as it was like deployed, it was installed on different computer and android phones and tested its classification ability and the application was able to return its expected result.

CONCLUSION AND RECOMMENDATIONS

This study was able to achieve its main objectives that was to create a classification model that will be able to classify Baybayin characters which was successfully embedded to a computer-based and android-based application, with the help also of the second main objective of this study which was to implement an algorithm that will segment Baybayin characters that were present in an image, the application was able to read and produce a word transliteration from Baybayin writing system to Tagalog writing system. With the model's performance testing which obtained a 96.87836% accuracy this indicates that the application is reliable in predicting Baybayin words equivalent.

However, this study has a lot of rooms for improvement, first is to further improve the performance of the classification model and also consider modern versions of Baybayin characters that are present nowadays. Transliteration of more than one Baybayin word is also a big challenge for improvement in this study, considering a lot of effort to segment many characters such as sentences or phrases to be classified to the model. Also, a new algorithm might be needed for character extraction due to complex structure of segmenting characters for sentences and also to teach the system to read spaces that is present in a sentence.

IMPLEMENTATION PLAN

Project Implementation Checklist

Statement Type	Include	Exclude
Executable	/	
Non-Executable	/	
Declarations	/	
Compiler Directions	/	
Comments	/	
How Produced		
Programmed	/	
Generated with source code generators		/
Converted with automated translators		/
Copied or reused without change		/
Modified	/	
Removed	/	
Origin		
New Work: no prior existence	/	
Prior Work: taken or adapted from		/
A previous version, build or release		/
Commercial off-the-shelf software		/
Government furnished software		/
Another product		/
Other commercial library		/
A reuse library (software designed for reuse)		/
Other software component or library		/
Usage		
In or as part of primary product	/	
External to or in support of primary product		/
Delivery		
Delivered	/	
Delivered as source		/
Delivered in compiled or executable form	/	
Not Delivered		/
Under configuration control		/
Not under configuration control		/
Functionality		
Operative	/	
Inoperative		/
Functional (Intentional dead code, reactivated for special purposes)		/
Nonfunctional (unintentionally present)		/

Replications		
Master source statements (originals)	/	
Physical replications of master statements, stored in master code	/	
Copies inserted, instantiated or expanded when compiling or linking		/
Development Status		
Estimated or planned	/	
Designed	/	
Coded	/	
Model Performance Evaluation Completed	/	
Unit Test Completed	/	
Integrated into components	/	
Test readiness review completed	/	
Software (CSCI computer software configuration item) tests completed	/	
Systems test completed	/	
Evaluation		
User Evaluation Completed	/	

Implementation Contingency

In a project or a software development, there are situations that might affect the implementation process in which these situations should be considered and planned beforehand the implementation period started. Upon analyzing the requirements of this project this was planned to be implemented in the following tech stacks; convolutional neural network (CNN) and OTSU's image thresholding algorithm which will be accompanied with python and java for the software application interface. As part of the project's contingency plan, studying other methods of implementing this project was also considered. Due to the huge uncertainty of the proponents to their ability for the said tech stacks to be used, it was planned to study other algorithm to use that will be helpful to the project, Support Vector Machine (SVM) and OpenCV adaptive thresholding was considered as the alternative solution if convolutional neural network (CNN) and OTSU's algorithm will fail to be implemented.

This study has stated some possible risks that will be encountered in the implemented software application (Table3. Risks Assessment Analysis), these risks have the capability to give some possible conflict in the runtime process of the application which were just aided with some error or warning messages. These were considered by the proponents upon the development of the software application, but its alternative methods are beyond of the scope of this study.

Infrastructure/Deployment

Beforehand the deployment of this project, the proponents will prepare the necessary components that will be needed prior to the project deployment. The project will undergo software packaging or will be converted into executable files and apk file for it to be ready for installation. These files will be deployed into a computer or android devices, the executable file (.exe) will be used for the installation in computer devices and the apk file will be used for installation into android devices. The proponents of this project will select interested users for this project and will deploy the software to the users' preferred device. The users will be guided on how to operate the software and will be given the opportunity to provide their own test samples. An evaluation form ISO 9126 (Appendix B) will also be provided for the user to rate the system's functionality and usability.

Upon the deployment period it will be advised that the user will try out different Baybayin word sample such as Baybayin words from the internet and utilize some handwritten samples also. After several trials that the user will attempt in the software, they will be encouraged to rate the evaluation form honestly and will be asked for some recommendations and concerns regarding the application. The completed ISO 9126 or evaluation forms which were filled up by the selected users will be analyzed by the proponents afterwards the project deployment. These feedbacks will be used by the proponents as their bases on how to further improve the system.

BIBLIOGRAPHY

[1] "House Approves Baybayin as National Writing System", 2018. [Online]. Available: https://www.congress.gov.ph/press/details.php?pressid=10642/.

[2] R. Pino, R. Mendoza and R. Sambayan, "A Baybayin Word Recognition System", 2021

[3] R. Mithe, S. Indalkar and N. Divekar, "Optical Character Recognition", *Proceeding of the International Journal of Recent Technology and Engineering*. 2013

[4] K. Recio and R. Mendoza. "Three-step approach to edge detection of texts", *Philippine Journal of Science*. 2019

[5] L. Bague, RJ Jorda, B. Fortaleza and J. Velasco. "Recognition of Baybayin handwritten letters using VGG16 deep convolutional neural network model". *International Journal of Emerging Trends in Engineering Research.* 2020

[6] JA Nogra, CLS Romana, E. Maravillas. "LSTM neural networks for Baybayin handwriting recognition". *2019 IEEE 4th international conference on computer and communication systems*. 2019

[7] RN Recario, V. Mariano, DA Galvez, CM Lajara. "An automated reader Philippine Baybayin scripting image processing method". *ICCC International Digital Design Invitation Exhibition*. 2011

[8] JA Nogra, CLS Romana, E. Balakrishnan. "Baybayin character recognition using convolutional neural network". *International Journal of Machine Learning and Computing*. 2020

APPENDICES

Appendix A

RELEVANT SOURCE CODE

Model Training Source Code

In [1]:
```
from keras import optimizers
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Activation, Dropout, Flatten, Dense
from keras import backend as K
from keras.callbacks import EarlyStopping, ModelCheckpoint
from keras.layers import BatchNormalization

# image dimension
img_width, img_height = 64, 64

# training and validation set directory
train_data_dir = 'C:/Users/Jhimboy/Desktop/Old Experiment/Training'
validation_data_dir = 'C:/Users/Jhimboy/Desktop/Old Experiment/Validation'

num_train_samples = 7360
num_validation_samples = 945

epochs = 500
batch_size = 32

input_shape = (img_width, img_height, 3)

# this is the augmentation configuration use for training
train_datagen = ImageDataGenerator(rescale=1./255)

# this is the augmentation configuration use for testing:only rescaling
test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    train_data_dir,
    #color_mode = 'grayscale',
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='categorical')

validation_generator = test_datagen.flow_from_directory(
    validation_data_dir,
    #color_mode = 'grayscale',
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='categorical')
```

In [2]:
```
model = Sequential()
model.add(Conv2D(32, (3, 3), input_shape=input_shape))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(128, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())

# Fully connected layer
model.add(Dense(128))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(63))
model.add(Activation('softmax'))

model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

In [3]:
```
#define the checkpoint
filepath="C:/Users/Jhimboy/Desktop/Old Experiment/Trial2/model.h5"
checkpointer = ModelCheckpoint(filepath, monitor='accuracy', verbose=1, save_best_only=True, mode='max')

history = model.fit_generator(
    train_generator,
    steps_per_epoch=num_train_samples // batch_size,
    epochs=epochs,
    validation_data=validation_generator,
    validation_steps=num_validation_samples // batch_size,
    callbacks = [checkpointer]
)

#TRAINING HISTORY VISUALIZATION
import matplotlib.pyplot as plt

# Plot training & validation accuracy values
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()

# Plot training & validation loss values
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()

model.save('C:/Users/Jhimboy/Desktop/Old Experiment/Trial2/model.h5')
```

Model Testing Source Code

In [2]:
```
from keras.preprocessing.image import ImageDataGenerator
from keras.models import load_model
from keras.preprocessing import image
from keras.preprocessing.image import img_to_array, load_img
import numpy as np
import pandas as pd

# dimensions of our images.
num_test_samples = 928
img_width, img_height = 64,64
batch_size = 32

train_data_dir = 'C:/Users/Jhimboy/Desktop/Old Experiment/Training'
test_data_dir = 'C:/Users/Jhimboy/Desktop/Old Experiment/Testing'

test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = test_datagen.flow_from_directory(
    train_data_dir,
    #color_mode = 'grayscale',
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='categorical')

test_generator = test_datagen.flow_from_directory(
    test_data_dir,
    #color_mode = 'grayscale',
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode=None,
    shuffle=False
)


test_generator.reset()

test_model = load_model('C:/Users/Jhimboy/Desktop/Old Experiment/Trial2/keras_model.h5')
```

In [2]:
```
#output in probabilities
pred = test_model.predict_generator(test_generator, verbose=1, steps=num_test_samples/batch_size)

#convert the output into class number
predicted_class_indices=np.argmax(pred,axis=1)

#name of the classes
labels = (train_generator.class_indices)
labels = dict((v,k) for k,v in labels.items())
predictions = [labels[k] for k in predicted_class_indices]

filenames=test_generator.filenames
results=pd.DataFrame({"Filename":filenames,
                      "Predictions":predictions})
export_csv = results.to_csv (r'C:/Users/Jhimboy/Desktop/Old Experiment/Trial2/testing.csv', index = None, header=True)
```

Tensorflow Lite Conversion Source Code

In [1]:
```
import tensorflow as tf
from tensorflow.keras.models import load_model

model = tf.keras.models.load_model('C:/Users/Jhimboy/Desktop/Old Experiment/TFLite/model.h5')
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_quant_model = converter.convert()
open ("model.tflite" , "wb") .write(tflite_quant_model)
```

Segmentation Source Code

```
import cv2
from numpy.core.records import array
from numpy import asarray
from keras.preprocessing import image
# Bubble sort arrays for sorting the characters
def bubble_sort(store):
    print("Sorting the segmented characters.............")
    for i in range(len(store)):
        for j in range(0, len(store) - i - 1):
            if store[j][0] > store[j + 1][0]:
                tempx = store[j][0]
                tempy = store[j][1]
                tempw = store[j][2]
                temph = store[j][3]

                store[j][0] = store[j+1][0]
                store[j][1] = store[j+1][1]
                store[j][2] = store[j+1][2]
                store[j][3] = store[j+1][3]

                store[j+1][0] = tempx
                store[j+1][1] = tempy
                store[j+1][2] = tempw
                store[j+1][3] = temph
```

```
def define_region_of_interest(store, new_img, src_img):
    print("Defining the region of interest.............")
    limit = 0
    image_count = 1
    for x in range(len(store)):
        if store[x][0] > limit:
            region = store[x][0] + store[x][2]
            kudlit = False
            locationY = 0
            locationH = 0
            for xx in range(len(store)):
                if(store[xx][0] > store[x][0] and store[xx][0] < region):
                    kudlit = True
                    locationY = store[xx][1]
                    locationH = store[xx][3]

            X = store[x][0]
            Y = store[x][1]
            W = store[x][2]
            H = store[x][3]


            if kudlit == False:
                roi = new_img[Y-5: Y+H+5, X-5: X+W+5]
                cv2.rectangle(src_img,(X-1,Y-1),(X+W+1,Y+H+1),(0,255,0),1)
                limit = region
                cv2.imwrite("./Segmented Images/"+ str(image_count) + ".jpg", roi)
                print("Saving segmented character..... "+ str(image_count) + ".jpg")
                image_count += 1

            else:
                if locationY > store[x][1]: #nasa ubos
                    roi = new_img[Y-5: locationY+locationH+5, X-5: X+W+5]
                    cv2.rectangle(src_img,(X-1,Y-1),(X+W+1,locationY+locationH+1),(0,255,0),1)
                    limit = region
                    cv2.imwrite("./Segmented Images/"+ str(image_count) + ".jpg", roi)
                    print("Saving segmented character "+ str(image_count) + ".jpg")
                    image_count += 1

                if locationY < store[x][1]: #nasa taas
                    roi = new_img[locationY-5: Y+H+5, X-5: X+W+5]
                    cv2.rectangle(src_img,(X-1,locationY-1),(X+W+1,Y+H+1),(0,255,0),1)
                    limit = region
                cv2.imwrite("./Segmented Images/"+ str(image_count) + ".jpg", roi)
                print("Saving segmented character..... "+ str(image_count) + ".jpg")
                image_count += 1
```

```
    print("Segmentation Process Completed.............")
# Display the image
def path(directory):
    src_img= cv2.imread(directory)
    copy = src_img.copy()
    height = src_img.shape[0]
    width = src_img.shape[1]
    return start_segmentation(src_img, copy, height, width)
def start_segmentation(src_img, copy, height, width):
    # Resizing the image
    src_img = cv2.resize(copy, dsize =(700, int(700*height/width)), interpolation =
cv2.INTER_AREA)
    height = src_img.shape[0]
    width = src_img.shape[1]
    # Applying Threshold
    grey_img = cv2.cvtColor(src_img, cv2.COLOR_BGR2GRAY)
    print("Applying Adaptive Threshold with kernel : 21 X 21")
    bin_img =
cv2.adaptiveThreshold(grey_img,255,cv2.ADAPTIVE_THRESH_MEAN_C,cv2.THRESH_BINARY_INV,21,20)
    new_img =
cv2.adaptiveThreshold(grey_img,255,cv2.ADAPTIVE_THRESH_MEAN_C,cv2.THRESH_BINARY,21,20)
    # Removing small details from the image
    kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(3,3))
    print("Noise Removal From Image.........")
    final_thr = cv2.morphologyEx(bin_img, cv2.MORPH_CLOSE, kernel)
    contr_retrival = final_thr.copy()
    #Detecting Contours at the edge of a character
    contours, _ = cv2.findContours(contr_retrival,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
    cv2.drawContours(src_img, contours, -1, (0,255,0), 1)
    rows = len(contours)
    columns = 4
    store = []
    for r in range(0, rows):
        store.append([0 for c in range(0, columns)])
    count = 0
    for cnt in contours:
        if cv2.contourArea(cnt) > 10:
            x,y,w,h = cv2.boundingRect(cnt)
            store[count][0] = x
            store[count][1] = y
            store[count][2] = w
            store[count][3] = h
            count += 1

    bubble_sort(store)
    define_region_of_interest(store, new_img, src_img)
    return src_img
```

Computer-Based Main Python File Codes

```
import sys
import os
from tkinter import dialog
import numpy as np
import segmentation
import predictions
import fnmatch
import database
from PySide2 import QtCore, QtGui, QtWidgets
from PySide2.QtCore import Qt
from PySide2.QtGui import QColor, QCursor
from PySide2.QtWidgets import *
from keras.models import load_model
from keras.preprocessing import image
from PyQt5.QtWidgets import QMessageBox
## ==> WINDOWS
from ui_Welcome import  Ui_SplashScreen
from ui_MainWindow import Ui_MainWindow
from ui_Instructions import Ui_InstructionWindow
from ui_Admin import Ui_Security
from ui_AdminHome import Ui_AdminHome
from ui_AdminSettings import Ui_AdminSettings

## ==> GLOBALS
counter = 0
class InstructionWindow(QMainWindow):
    def __init__(self):
        QMainWindow.__init__(self)
        self.ui = Ui_InstructionWindow()
        self.ui.setupUi(self)
        self.setWindowFlag(Qt.FramelessWindowHint)
        self.setAttribute(Qt.WA_TranslucentBackground)

        ## DROP SHADOW EFFECT
        self.shadow = QGraphicsDropShadowEffect(self)
        self.shadow.setBlurRadius(20)
        self.shadow.setXOffset(0)
        self.shadow.setYOffset(0)
        self.shadow.setColor(QColor(0,0,0,100))
        self.ui.MainFrame.setGraphicsEffect(self.shadow)

        self.ui.btn_icon1.clicked.connect(self.classifier)
        self.ui.btn_icon4.clicked.connect(self.security)
    def classifier(self):
        self.main = MainWindow()
```

```
        self.main.show()
        self.close()

    def security(self):
        self.main = Security()
        self.main.show()
        self.close()
# YOUR APPLICATION
class MainWindow(QMainWindow):
    def __init__(self):
        QMainWindow.__init__(self)
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)
        self.setWindowFlag(Qt.FramelessWindowHint)
        self.setAttribute(Qt.WA_TranslucentBackground)
        self.ui.btn_classify.isEnabled()

        ## DROP SHADOW EFFECT
        self.shadow = QGraphicsDropShadowEffect(self)
        self.shadow.setBlurRadius(20)
        self.shadow.setXOffset(0)
        self.shadow.setYOffset(0)
        self.shadow.setColor(QColor(0,0,0,100))
        self.ui.MainFrame.setGraphicsEffect(self.shadow)

        self.ui.btn_browse.clicked.connect(self.browse_image)
        self.ui.btn_classify.clicked.connect(self.predict)
        self.ui.btn_close.clicked.connect(self.terminate)
        self.ui.btn_icon2.clicked.connect(self.instructions)
        self.ui.btn_icon4.clicked.connect(self.security)

        # VARIABLES
        self.imageFile = ""
        self.fileNameDirectory = ""
        self.probability = ""
        self.prediction = ""

    def browse_image(self):
        self.ui.lbl_picture.setPixmap(None)
        self.ui.lbl_prediction.setText("")
        self.ui.lbl_prediction_2.setText("")
        self.ui.lbl_file_directory.setText("Preparing image..")
        self.imageFile, _= QtWidgets.QFileDialog.getOpenFileName(self, "Load an image file: ",
"", "Image Files (*.jpg *.jpeg *.png)")

        try:
            if self.imageFile:
                cv2Img = segmentation.path(self.imageFile)
```

```
                cvimage = QtGui.QImage(cv2Img, cv2Img.shape[1], cv2Img.shape[0],
cv2Img.shape[1] * 3,QtGui.QImage.Format_RGB888).rgbSwapped()
                pix = QtGui.QPixmap(cvimage)
                self.ui.lbl_picture.setPixmap(pix)
                self.ui.lbl_picture.setScaledContents(True)
                self.ui.lbl_picture.setAlignment(QtCore.Qt.AlignCenter)
                self.ui.lbl_file_directory.setText(self.imageFile)
            else:
                self.ui.lbl_file_directory.setText("Please upload a clear image of a baybayin
word.")
        except:
            print("Error Occured !")
            self.ui.lbl_file_directory.setText("The image provided is invalid.")
            dialog = QMessageBox()
            dialog.setText("Invalid Image please make sure that the image is clear and
readable. Thank You")
            dialog.setWindowTitle("Error Message")
            dialog.setIcon(QMessageBox.Warning)
            dialog.exec_()

    def predict(self):
        self.ui.btn_classify.setCursor(QCursor(Qt.WaitCursor))
        def classifyImage(image_path):
            self.newImage = image.load_img(image_path, target_size=(64, 64))
            self.newImage = image.img_to_array(self.newImage)
            self.newImage = self.newImage.astype("float32")
            self.newImage /= 255.0
            self.newImage = np.expand_dims(self.newImage, axis=0)

            self.savedModel = load_model("C:/Users/Jhimboy/Desktop/Image Classifier/OCR
Model/model.h5")

            # Classifying the image
            predictx = self.savedModel.predict(self.newImage)
            return predictx

        totalFiles = len(fnmatch.filter(os.listdir('./Segmented Images'), '*.jpg'))
        outputText = ""
        latin = "/"
        print("Reading predictions for segmented characters.....")
        probability = 0
        count = 0
        for i in range(totalFiles):
            path = "./Segmented Images/" + str(i + 1) + ".jpg"
            prediction = classifyImage(path)
            predicted_class = np.argmax(prediction, axis=1)
            probability += prediction
            print("Prediction : ", predicted_class)
```

```
            outputText += predictions.predict(predicted_class)
            latin += predictions.predict(predicted_class)
            if(i != totalFiles-1):
                latin += "-"
            else:
                latin += "/"
            print("Output text : ", outputText)
            count += 1


        self.ui.lbl_prediction.setText(latin.lower())

        print("Removing segmented characters files........")
        for i in range(totalFiles):
            path = "./Segmented Images/" + str(i + 1) + ".jpg"
            os.remove(path)

        #DB part
        db = database.TagalogDB()
        resultWord = db.search(outputText.lower())
        if(resultWord != ""):
            self.ui.lbl_prediction_2.setText(resultWord.upper())
        else:
            print("The word is not in the dictionary")

        self.ui.btn_classify.setCursor(QCursor(Qt.PointingHandCursor))


    def instructions(self):
        self.main = InstructionWindow()
        self.main.show()
        self.close()


    def security(self):
        self.main = Security()
        self.main.show()
        self.close()


    def terminate(self):
        self.close()


class Security(QMainWindow):
    def __init__(self):
        QMainWindow.__init__(self)
        self.ui = Ui_Security()
```

```
        self.ui.setupUi(self)

        ## REMOVE TITLE BAR
        self.setWindowFlag(Qt.FramelessWindowHint)
        self.setAttribute(Qt.WA_TranslucentBackground)

        ## DROP SHADOW EFFECT
        self.shadow = QGraphicsDropShadowEffect(self)
        self.shadow.setBlurRadius(20)
        self.shadow.setXOffset(0)
        self.shadow.setYOffset(0)
        self.shadow.setColor(QColor(0,0,0,100))
        self.ui.dropShadowFrame.setGraphicsEffect(self.shadow)

        self.ui.btn_close.clicked.connect(self.terminate)
        self.ui.btn_login.clicked.connect(self.login)

    def terminate(self):
        self.main = MainWindow()
        self.main.show()
        self.close()

    def login(self):
        db = database.TagalogDB()
        username = db.checkUsername(self.ui.txt_username.text())
        password = db.checkPassword(self.ui.txt_password.text())

        if(username == True and password == True):
            print("Administrator Mode")
            self.main = AdminHome()
            self.main.show()
            self.close()

        if(username == False or password == False):
            print("Invalid Credentials")
            dialog = QMessageBox()
            dialog.setText("The credentials provided is invalid!")
            dialog.setWindowTitle("Error Message")
            dialog.setIcon(QMessageBox.Warning)
            dialog.exec_()

class AdminSettings(QMainWindow):
    def __init__(self):
        QMainWindow.__init__(self)
        self.ui = Ui_AdminSettings()
        self.ui.setupUi(self)
        self.setWindowFlag(Qt.FramelessWindowHint)
        self.setAttribute(Qt.WA_TranslucentBackground)
```

```
        ## DROP SHADOW EFFECT
        self.shadow = QGraphicsDropShadowEffect(self)
        self.shadow.setBlurRadius(20)
        self.shadow.setXOffset(0)
        self.shadow.setYOffset(0)
        self.shadow.setColor(QColor(0,0,0,100))
        self.ui.MainFrame.setGraphicsEffect(self.shadow)

        self.ui.btn_icon4.clicked.connect(self.logout)
        self.ui.btn_close.clicked.connect(self.logout)
        self.ui.btn_icon1.clicked.connect(self.adminHome)
        self.ui.btn_save_username.clicked.connect(self.changeUsername)
        self.ui.btn_save_password.clicked.connect(self.changePassword)

    def changeUsername(self):
        db = database.TagalogDB()
        old_username = db.checkUsername(self.ui.txt_old_username.text())

        if(old_username == True):
            new_username = self.ui.txt_ne_username.text()
            confirm_username = self.ui.txt_confirm_username.text()
            if(new_username == confirm_username):
                db.changeUsername(confirm_username)
                dialog = QMessageBox()
                dialog.setText("Username successfully changed!")
                dialog.setWindowTitle("Success")
                dialog.setIcon(QMessageBox.Information)
                dialog.exec_()
            else:
                dialog = QMessageBox()
                dialog.setText("Please confirm the new username!")
                dialog.setWindowTitle("Warning")
                dialog.setIcon(QMessageBox.Warning)
                dialog.exec_()
        else:
            dialog = QMessageBox()
            dialog.setText("Old username does not match!")
            dialog.setWindowTitle("Error Message")
            dialog.setIcon(QMessageBox.Warning)
            dialog.exec_()

    def changePassword(self):
        pass

    def adminHome(self):
        self.main = AdminHome()
        self.main.show()
```

```
        self.close()

    def logout(self):
        self.main = MainWindow()
        self.main.show()
        self.close()


class AdminHome(QMainWindow):
    def __init__(self):
        QMainWindow.__init__(self)
        self.ui = Ui_AdminHome()
        self.ui.setupUi(self)
        self.setWindowFlag(Qt.FramelessWindowHint)
        self.setAttribute(Qt.WA_TranslucentBackground)

        ## DROP SHADOW EFFECT
        self.shadow = QGraphicsDropShadowEffect(self)
        self.shadow.setBlurRadius(20)
        self.shadow.setXOffset(0)
        self.shadow.setYOffset(0)
        self.shadow.setColor(QColor(0,0,0,100))
        self.ui.MainFrame.setGraphicsEffect(self.shadow)
        self.ui.tableWidget.setColumnWidth(0,70)
        self.ui.tableWidget.setColumnWidth(1,364)
        self.load_data()

        self.ui.btn_search.clicked.connect(self.search)
        self.ui.btn_save.clicked.connect(self.save)
        self.ui.btn_icon4.clicked.connect(self.logout)
        self.ui.btn_close.clicked.connect(self.logout)
        self.ui.btn_icon2.clicked.connect(self.settings)


    def settings(self):
        self.main = AdminSettings()
        self.main.show()
        self.close()

    def logout(self):
        self.main = MainWindow()
        self.main.show()
        self.close()

    def save(self):
        db = database.TagalogDB()
        id = db.getLength() + 1
        word = self.ui.txt_word.text()
```

```
    keyword = self.ui.txt_keyword.text()
    idKeyword = db.getLengthKeyword() + 1

    found = ""
    cur = db.search_word(self.ui.txt_word.text())
    for row in cur:
        found += row[1]

    print(found)
    if(found == ""):
        db.insert(id, word, keyword, idKeyword)
        print("Data is saved in the database")
        dialog = QMessageBox()
        dialog.setText("Data has been saved into the database.")
        dialog.setWindowTitle("Success")
        dialog.setIcon(QMessageBox.Information)
        dialog.exec_()
    else:
        print("Data is already in the dictionary.")
        dialog = QMessageBox()
        dialog.setText("Data is already existing in the database.")
        dialog.setWindowTitle("Warning")
        dialog.setIcon(QMessageBox.Warning)
        dialog.exec_()



def search(self):
    db = database.TagalogDB()
    self.ui.tableWidget.clear()
    self.ui.tableWidget.setHorizontalHeaderLabels(["No.","Tagalog Words"])
    self.ui.tableWidget.setRowCount(1)
    tableRow = 0
    cur = db.search_word(self.ui.txt_search.text())
    for row in cur:
        self.ui.tableWidget.setItem(tableRow, 0, QtWidgets.QTableWidgetItem(str(row[0])))
        self.ui.tableWidget.setItem(tableRow, 1, QtWidgets.QTableWidgetItem(row[1]))

def load_data(self):
    db = database.TagalogDB()
    self.ui.tableWidget.clear()
    self.ui.tableWidget.setHorizontalHeaderLabels(["No.","Tagalog Words"])
    self.ui.tableWidget.setRowCount(3000)
    tableRow = 0
    cur = db.getData()
    for row in cur:
        self.ui.tableWidget.setItem(tableRow, 0, QtWidgets.QTableWidgetItem(str(row[0])))
        self.ui.tableWidget.setItem(tableRow, 1, QtWidgets.QTableWidgetItem(row[1]))
```

```
            print(row)
            tableRow += 1


# SPLASH SCREEN
class SplashScreen(QMainWindow):
    def __init__(self):
        QMainWindow.__init__(self)
        self.ui = Ui_SplashScreen()
        self.ui.setupUi(self)

        ## REMOVE TITLE BAR
        self.setWindowFlag(Qt.FramelessWindowHint)
        self.setAttribute(Qt.WA_TranslucentBackground)

        ## DROP SHADOW EFFECT
        self.shadow = QGraphicsDropShadowEffect(self)
        self.shadow.setBlurRadius(20)
        self.shadow.setXOffset(0)
        self.shadow.setYOffset(0)
        self.shadow.setColor(QColor(0,0,0,100))
        self.ui.dropShadowFrame.setGraphicsEffect(self.shadow)

        ## TIMER
        self.timer = QtCore.QTimer()
        self.timer.timeout.connect(self.progress)
        self.timer.start(30)
        self.show()

    ## ==> APP FUNCTIONS
    ########################################################################
    def progress(self):

        global counter

        if counter > 200:
            self.timer.stop()
            self.main = MainWindow()
            self.main.show()
            self.close()
        counter += 2


if __name__ == "__main__":
    app = QApplication(sys.argv)
    window = SplashScreen()
    sys.exit(app.exec_())
```

Appendix B

EVALUATION TOOL

System Evaluation (ISO 9126)

Instructions: Please evaluate the "Baybayin Image Classifier" using the scale shown below. Check(/) the appropriate score. Thank You.

Marco Eraño Pahamotang	Jannie Fleur V. Oraño
Programmer	Adviser

Qualitative Description per Functionality Indicator

Limits of Scale	Qualitative Description
4.21 – 5.00	Fully Functional
3.21 – 4.20	Mostly Functional
2.61 – 3.20	Functional
1.81 – 2.60	Slightly Functional
1.0 – 1.8	Not Functional

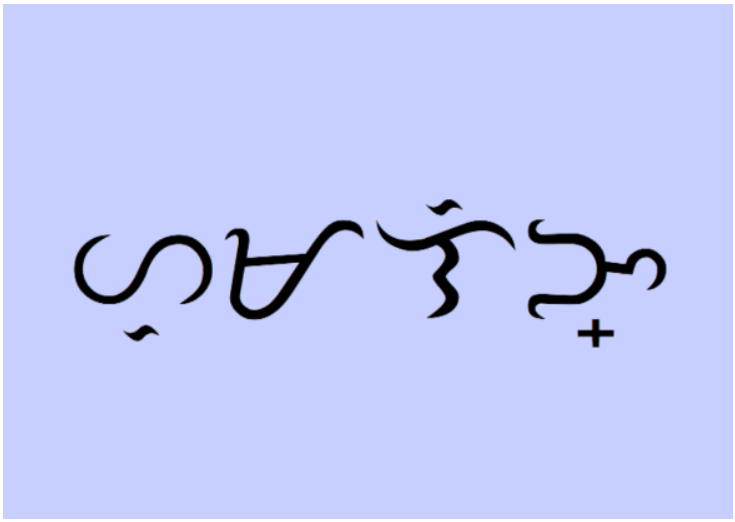
Qualitative Description per Usability Indicator

Limits of Scale	Qualitative Description
4.21 – 5.00	Fully Usable
3.21 – 4.20	Mostly Usable
2.61 – 3.20	Usable
1.81 – 2.60	Slightly Usable
1.0 – 1.8	Not Usable

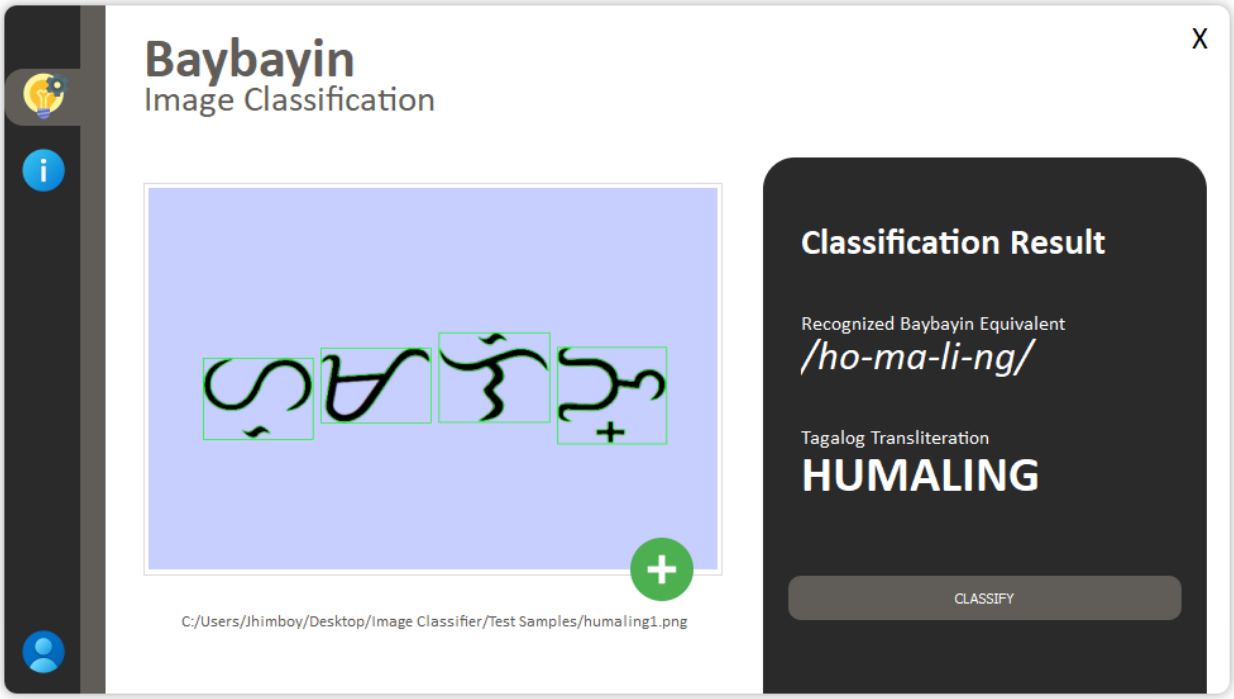
Criteria		**Score**				
Characteristics	*Sub Characteristics*	**1**	**2**	**3**	**4**	**5**
Functionality	The application performs the required functionalities					
	The application provides the expected result					
Usability	The graphical user interface of the application is easy to use or navigate					
	The displayed results of the application are understandable					
	The application is portable					
	Based on the utilized online test dataset, the application can recognize the types of Baybayin characters					
	The application is helpful					

Appendix C

SAMPLE INPUT AND OUTPUT



Sample Input

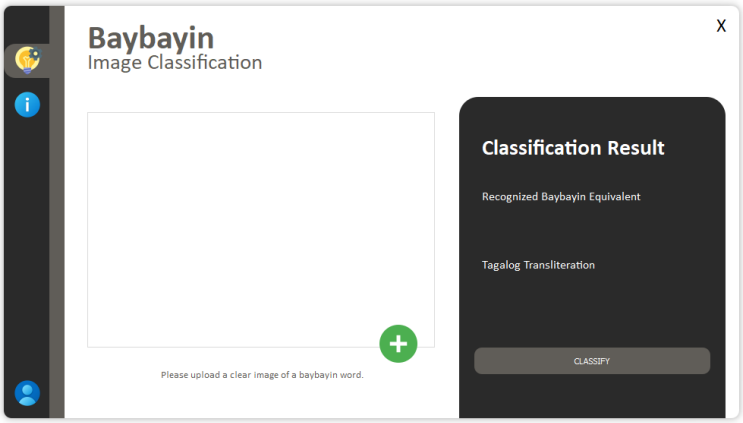


Classification Output

Appendix D

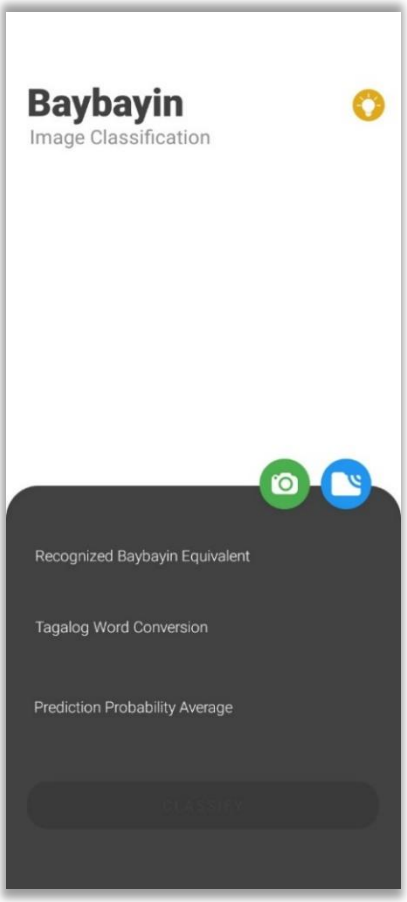
USER'S GUIDE

Computer-Based Manual



1. Click the green plus sign icon to add image to the software.
2. The system will open file explorer dialog, select the image you wanted to classify.
3. The image will be reflected into the system if it is a valid image, otherwise it will display a warning message.
4. Click the classify button to view the classification result.
5. Click again the green plus icon to try another image to classify.

Android-Based Manual



1. Click the file icon to select image from the local file manager of the phone or;
2. Click the camera icon to capture real-time images to display into the system
3. An image will be reflected into the software with an indicated green square that are the features or characters extracted from the image.
4. If a warning image occurred it means that the provided image is not valid, click okay to provide another input image.
5. If an image is already reflected in the system click CLASSIFY button to display the classification result.
6. Click the camera or file icon to try another image.

Appendix E

OTHER RELEVANT DOCUMENTS

Testing Result (testing.csv)

Filename	Prediction	Filename	Prediction	Filename	Prediction
a\a (134).jpg	a	ba\ba (38).jpg	ba	d\d (158).jpg	d
a\a (135).jpg	a	ba\ba (72).jpg	ba	d\d (176).jpg	d
a\a (137).jpg	a	ba\ba (76).jpg	ba	d\d (198).jpg	d
a\a (34).jpg	a	ba\ba (85).jpg	ba	d\d (202).jpg	d
a\a (35).jpg	ya	ba\ba (86).jpg	ba	d\d (213).jpg	d
a\a (40).jpg	a	ba\ba (90).jpg	ba	d\d (91).jpg	d
a\a (41).jpg	a	be\be (100).jpg	be	d\d (92).jpg	d
a\a (42).jpg	a	be\be (101).jpg	be	d\d (93).jpg	d
a\a (43).jpg	a	be\be (103).jpg	be	d\d (95).jpg	d
a\a (44).jpg	a	be\be (104).jpg	be	d\d (96).jpg	d
a\a (47).jpg	a	be\be (105).jpg	be	d\d (97).jpg	d
a\a (48).jpg	a	be\be (106).jpg	be	d\d (98).jpg	d
a\a (50).jpg	a	be\be (25).jpg	be	da\da (126).jpg	da
a\a (69).jpg	a	be\be (26).jpg	be	da\da (134).jpg	da
a\a (70).jpg	a	be\be (27).jpg	be	da\da (145).jpg	da
b\b (140).jpg	b	be\be (40).jpg	be	da\da (146).jpg	da
b\b (141).jpg	b	be\be (47).jpg	be	da\da (159).jpg	da
b\b (142).jpg	b	be\be (53).jpg	be	da\da (169).jpg	da
b\b (143).jpg	b	be\be (97).jpg	be	da\da (42).jpg	da
b\b (159).jpg	b	be\be (98).jpg	be	da\da (43).jpg	da
b\b (160).jpg	b	be\be (99).jpg	be	da\da (45).jpg	da
b\b (172).jpg	b	bo\bo (106).jpg	bo	da\da (46).jpg	da
b\b (173).jpg	b	bo\bo (108).jpg	bo	da\da (48).jpg	da
b\b (18).jpg	b	bo\bo (109).jpg	bo	da\da (49).jpg	da
b\b (19).jpg	b	bo\bo (113).jpg	bo	da\da (50).jpg	da
b\b (20).jpg	bo	bo\bo (114).jpg	bo	da\da (51).jpg	da
b\b (21).jpg	b	bo\bo (115).jpg	bo	da\da (55).jpg	da
b\b (22).jpg	b	bo\bo (116).jpg	bo	de\de (13).jpg	de
b\b (23).jpg	b	bo\bo (117).jpg	b	de\de (22).jpg	de
b\b (24).jpg	b	bo\bo (118).jpg	bo	de\de (28).jpg	de
ba\ba (29).jpg	be	bo\bo (13).jpg	bo	de\de (3).jpg	de
ba\ba (30).jpg	ba	bo\bo (16).jpg	bo	de\de (46).jpg	de
ba\ba (31).jpg	ba	bo\bo (21).jpg	bo	de\de (52).jpg	de
ba\ba (32).jpg	ba	bo\bo (24).jpg	bo	de\de (71).jpg	de
ba\ba (33).jpg	ba	bo\bo (30).jpg	bo	de\de (72).jpg	de
ba\ba (34).jpg	ba	bo\bo (37).jpg	bo	de\de (73).jpg	de
ba\ba (35).jpg	ba	d\d (100).jpg	d	de\de (77).jpg	de
ba\ba (36).jpg	ba	d\d (148).jpg	d	de\de (80).jpg	de
ba\ba (37).jpg	ba	d\d (156).jpg	d	de\de (81).jpg	de

Filename	Prediction	Filename	Prediction	Filename	Prediction
de\de (83).jpg	de	g\g (21).jpg	g	go\go (30).jpg	go
de\de (84).jpg	de	g\g (22).jpg	g	go\go (37).jpg	go
de\de (85).jpg	de	g\g (23).jpg	g	go\go (43).jpg	go
do\do (109).jpg	do	g\g (29).jpg	go	go\go (51).jpg	go
do\do (110).jpg	do	ga\ga (124).jpg	ga	go\go (7).jpg	go
do\do (111).jpg	do	ga\ga (143).jpg	ga	h\h (104).jpg	h
do\do (112).jpg	do	ga\ga (146).jpg	ga	h\h (14).jpg	h
do\do (113).jpg	do	ga\ga (169).jpg	ga	h\h (18).jpg	h
do\do (114).jpg	do	ga\ga (179).jpg	ga	h\h (227).jpg	h
do\do (118).jpg	do	ga\ga (26).jpg	ga	h\h (232).jpg	h
do\do (119).jpg	do	ga\ga (42).jpg	ga	h\h (249).jpg	h
do\do (121).jpg	do	ga\ga (44).jpg	ga	h\h (28).jpg	h
do\do (122).jpg	do	ga\ga (45).jpg	ga	h\h (29).jpg	h
do\do (123).jpg	do	ga\ga (46).jpg	ga	h\h (42).jpg	h
do\do (18).jpg	do	ga\ga (47).jpg	ga	h\h (43).jpg	h
do\do (29).jpg	do	ga\ga (48).jpg	ga	h\h (44).jpg	h
do\do (38).jpg	do	ga\ga (50).jpg	ga	h\h (45).jpg	ha
do\do (8).jpg	do	ga\ga (51).jpg	ga	h\h (46).jpg	h
e\e (136).jpg	e	ga\ga (55).jpg	ga	h\h (5).jpg	h
e\e (153).jpg	e	ge\ge (17).jpg	ge	h\h (51).jpg	h
e\e (158).jpg	e	ge\ge (22).jpg	ge	ha\ha (100).jpg	ha
e\e (183).jpg	e	ge\ge (32).jpg	ge	ha\ha (117).jpg	ha
e\e (21).jpg	e	ge\ge (45).jpg	ge	ha\ha (136).jpg	ha
e\e (22).jpg	e	ge\ge (56).jpg	ge	ha\ha (153).jpg	ha
e\e (23).jpg	e	ge\ge (82).jpg	ge	ha\ha (162).jpg	ha
e\e (25).jpg	e	ge\ge (83).jpg	ge	ha\ha (19).jpg	ha
e\e (26).jpg	e	ge\ge (84).jpg	ge	ha\ha (20).jpg	ha
e\e (27).jpg	e	ge\ge (85).jpg	ge	ha\ha (21).jpg	ha
e\e (28).jpg	e	ge\ge (9).jpg	ge	ha\ha (22).jpg	ha
e\e (29).jpg	e	ge\ge (90).jpg	ge	ha\ha (23).jpg	ha
e\e (30).jpg	e	ge\ge (91).jpg	ge	ha\ha (24).jpg	ha
e\e (31).jpg	e	ge\ge (92).jpg	ge	ha\ha (25).jpg	ha
e\e (32).jpg	e	ge\ge (95).jpg	ge	ha\ha (26).jpg	ha
g\g (107).jpg	g	ge\ge (96).jpg	ge	ha\ha (30).jpg	ha
g\g (116).jpg	g	go\go (120).jpg	go	ha\ha (94).jpg	ha
g\g (124).jpg	g	go\go (121).jpg	go	he\he (176).jpg	he
g\g (129).jpg	g	go\go (125).jpg	go	he\he (184).jpg	he
g\g (141).jpg	g	go\go (126).jpg	go	he\he (193).jpg	he
g\g (152).jpg	g	go\go (127).jpg	go	he\he (199).jpg	he
g\g (16).jpg	g	go\go (129).jpg	go	he\he (215).jpg	he
g\g (17).jpg	g	go\go (132).jpg	go	he\he (37).jpg	he
g\g (175).jpg	g	go\go (133).jpg	go	he\he (39).jpg	he
g\g (19).jpg	g	go\go (134).jpg	go	he\he (40).jpg	he
g\g (20).jpg	g	go\go (14).jpg	go	he\he (41).jpg	he

Filename	Prediction	Filename	Prediction	Filename	Prediction
he\he (43).jpg	he	ka\ka (97).jpg	ka	la\la (16).jpg	la
he\he (44).jpg	he	ke\ke (100).jpg	ke	la\la (17).jpg	la
he\he (46).jpg	he	ke\ke (157).jpg	ke	la\la (18).jpg	la
he\he (48).jpg	he	ke\ke (159).jpg	ke	la\la (22).jpg	la
he\he (49).jpg	he	ke\ke (165).jpg	ke	la\la (23).jpg	la
he\he (51).jpg	he	ke\ke (179).jpg	ke	la\la (24).jpg	la
ho\ho (18).jpg	ho	ke\ke (31).jpg	ke	la\la (25).jpg	la
ho\ho (181).jpg	ho	ke\ke (94).jpg	ke	la\la (26).jpg	la
ho\ho (190).jpg	ho	ke\ke (95).jpg	ke	la\la (27).jpg	la
ho\ho (200).jpg	ho	ke\ke (96).jpg	ke	la\la (28).jpg	la
ho\ho (24).jpg	ho	ke\ke (97).jpg	ke	la\la (29).jpg	la
ho\ho (7).jpg	ho	ke\ke (98).jpg	ke	la\la (30).jpg	la
ho\ho (76).jpg	ho	ke\ke (99).jpg	ke	le\le (169).jpg	le
ho\ho (77).jpg	lo	ko\ko (1).jpg	ko	le\le (181).jpg	le
ho\ho (78).jpg	ho	ko\ko (10).jpg	ko	le\le (192).jpg	le
ho\ho (79).jpg	ho	ko\ko (15).jpg	ko	le\le (29).jpg	le
ho\ho (86).jpg	ho	ko\ko (158).jpg	ko	le\le (6).jpg	le
ho\ho (87).jpg	ho	ko\ko (169).jpg	ko	le\le (60).jpg	le
ho\ho (88).jpg	ho	ko\ko (184).jpg	ko	le\le (64).jpg	le
ho\ho (89).jpg	ho	ko\ko (25).jpg	ko	le\le (65).jpg	le
ho\ho (90).jpg	ho	ko\ko (28).jpg	ko	le\le (66).jpg	le
k\k (17).jpg	k	ko\ko (41).jpg	ko	le\le (68).jpg	le
k\k (28).jpg	k	ko\ko (42).jpg	ko	le\le (69).jpg	le
k\k (31).jpg	k	ko\ko (43).jpg	ko	le\le (70).jpg	le
k\k (41).jpg	k	ko\ko (44).jpg	ko	le\le (71).jpg	le
k\k (50).jpg	k	ko\ko (45).jpg	ko	le\le (87).jpg	le
k\k (56).jpg	k	ko\ko (49).jpg	ko	le\le (88).jpg	le
k\k (60).jpg	k	l\l (192).jpg	l	lo\lo (161).jpg	lo
k\k (73).jpg	k	l\l (201).jpg	la	lo\lo (168).jpg	lo
k\k (84).jpg	k	l\l (210).jpg	l	lo\lo (171).jpg	lo
k\k (88).jpg	k	l\l (232).jpg	l	lo\lo (194).jpg	lo
k\k (89).jpg	k	l\l (235).jpg	l	lo\lo (205).jpg	lo
k\k (90).jpg	k	l\l (6).jpg	l	lo\lo (27).jpg	lo
k\k (92).jpg	k	l\l (60).jpg	l	lo\lo (31).jpg	lo
ka\ka (101).jpg	ka	l\l (61).jpg	l	lo\lo (32).jpg	lo
ka\ka (102).jpg	ka	l\l (62).jpg	l	lo\lo (33).jpg	lo
ka\ka (103).jpg	ka	l\l (63).jpg	lo	lo\lo (38).jpg	lo
ka\ka (104).jpg	ka	l\l (64).jpg	l	lo\lo (39).jpg	lo
ka\ka (105).jpg	ka	l\l (65).jpg	l	lo\lo (40).jpg	lo
ka\ka (61).jpg	ka	l\l (68).jpg	l	lo\lo (41).jpg	lo
ka\ka (68).jpg	ka	la\la (126).jpg	la	m\m (105).jpg	m
ka\ka (76).jpg	ka	la\la (138).jpg	la	m\m (117).jpg	m
ka\ka (83).jpg	ka	la\la (149).jpg	la	m\m (131).jpg	m

Filename	Pred.	Filename	Pred.	Filename	Pred.
m\m (140).jpg	m	mo\mo (119).jpg	mo	ne\ne (2).jpg	ne
m\m (146).jpg	m	mo\mo (120).jpg	m	ne\ne (22).jpg	ne
m\m (159).jpg	m	mo\mo (121).jpg	mo	ne\ne (29).jpg	ne
m\m (45).jpg	m	mo\mo (122).jpg	mo	ne\ne (4).jpg	ne
m\m (46).jpg	m	mo\mo (123).jpg	mo	ne\ne (60).jpg	ne
m\m (52).jpg	m	mo\mo (124).jpg	mo	ne\ne (61).jpg	ne
m\m (56).jpg	m	mo\mo (125).jpg	mo	ne\ne (63).jpg	ne
m\m (57).jpg	m	mo\mo (127).jpg	mo	ne\ne (64).jpg	ne
m\m (58).jpg	m	mo\mo (128).jpg	mo	ne\ne (65).jpg	ne
m\m (60).jpg	m	mo\mo (161).jpg	mo	ne\ne (66).jpg	ne
m\m (94).jpg	m	mo\mo (196).jpg	mo	ne\ne (86).jpg	ne
m\m (98).jpg	m	mo\mo (204).jpg	mo	ne\ne (87).jpg	ne
ma\ma (10).jpg	ma	mo\mo (207).jpg	mo	ne\ne (88).jpg	ne
ma\ma (152).jpg	ma	n\n (135).jpg	n	ne\ne (89).jpg	ne
ma\ma (153).jpg	ma	n\n (147).jpg	n	ng\ng (118).jpg	ng
ma\ma (154).jpg	ma	n\n (154).jpg	n	ng\ng (120).jpg	ng
ma\ma (155).jpg	ma	n\n (164).jpg	n	ng\ng (121).jpg	ng
ma\ma (157).jpg	ma	n\n (168).jpg	n	ng\ng (123).jpg	ng
ma\ma (158).jpg	ma	n\n (177).jpg	n	ng\ng (124).jpg	ng
ma\ma (159).jpg	ma	n\n (183).jpg	n	ng\ng (126).jpg	ng
ma\ma (160).jpg	ma	n\n (64).jpg	n	ng\ng (162).jpg	ng
ma\ma (161).jpg	ma	n\n (69).jpg	n	ng\ng (163).jpg	ng
ma\ma (162).jpg	ma	n\n (70).jpg	n	ng\ng (164).jpg	ng
ma\ma (22).jpg	ma	n\n (71).jpg	n	ng\ng (165).jpg	ng
ma\ma (41).jpg	ma	n\n (72).jpg	n	ng\ng (18).jpg	ng
ma\ma (53).jpg	ma	n\n (74).jpg	n	ng\ng (29).jpg	ng
ma\ma (59).jpg	ma	n\n (75).jpg	n	ng\ng (61).jpg	ng
me\me (112).jpg	me	n\n (77).jpg	n	ng\ng (73).jpg	ng
me\me (18).jpg	me	na\na (114).jpg	na	ng\ng (84).jpg	ng
me\me (24).jpg	me	na\na (129).jpg	na	nga\nga (105).jpg	nga
me\me (29).jpg	nge	na\na (138).jpg	na	nga\nga (107).jpg	nga
me\me (35).jpg	me	na\na (15).jpg	na	nga\nga (108).jpg	nga
me\me (4).jpg	me	na\na (16).jpg	na	nga\nga (110).jpg	nga
me\me (83).jpg	me	na\na (17).jpg	na	nga\nga (111).jpg	nga
me\me (84).jpg	me	na\na (18).jpg	na	nga\nga (112).jpg	nga
me\me (85).jpg	me	na\na (20).jpg	na	nga\nga (113).jpg	nga
me\me (86).jpg	me	na\na (21).jpg	na	nga\nga (117).jpg	nga
me\me (87).jpg	me	na\na (22).jpg	na	nga\nga (118).jpg	nga
me\me (89).jpg	me	na\na (23).jpg	na	nga\nga (18).jpg	nga
me\me (90).jpg	ye	na\na (24).jpg	na	nga\nga (23).jpg	nga
me\me (91).jpg	me	na\na (25).jpg	na	nga\nga (33).jpg	nga
me\me (92).jpg	me	na\na (80).jpg	na	nga\nga (37).jpg	nga
mo\mo (114).jpg	mo	na\na (94).jpg	no	nga\nga (43).jpg	nga
mo\mo (118).jpg	mo	ne\ne (17).jpg	ne	nga\nga (56).jpg	nga

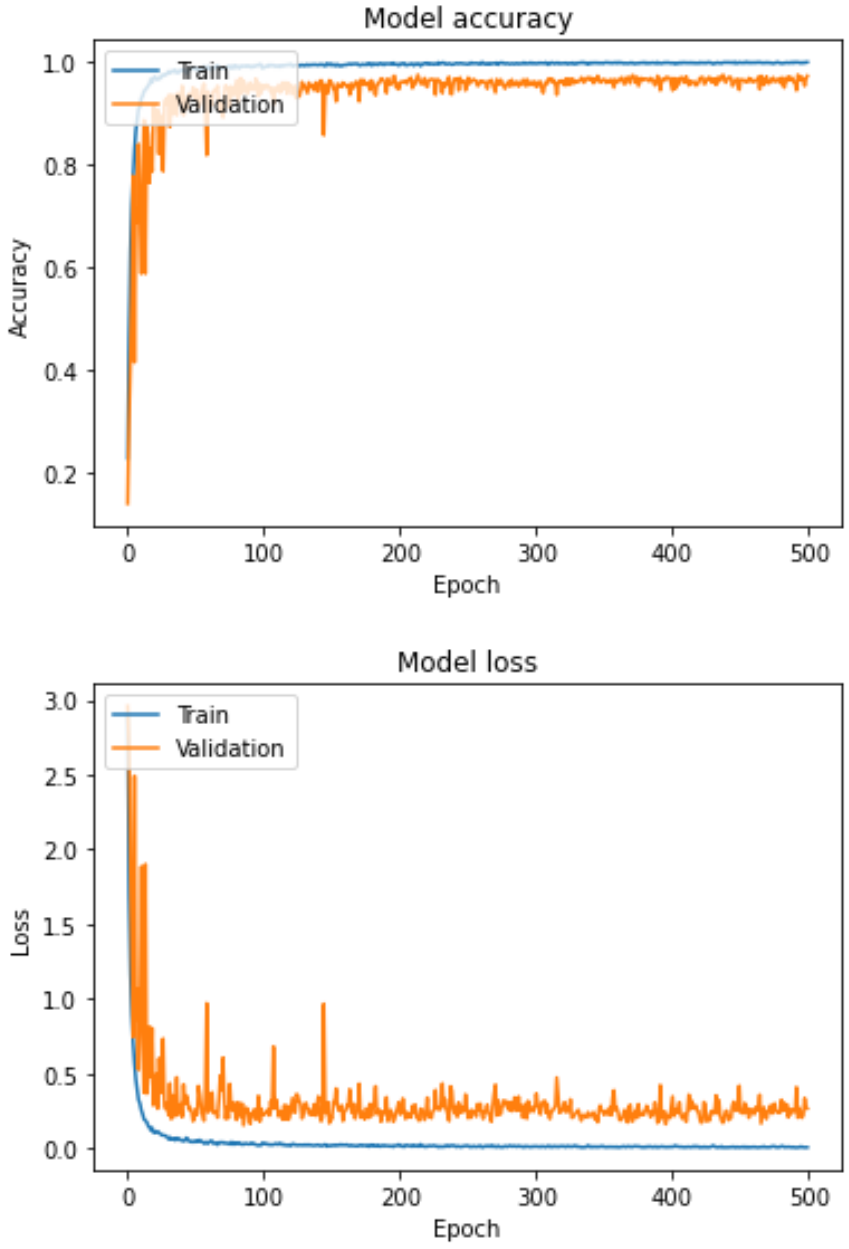
Filename	Pred.	Filename	Pred.	Filename	Pred.
nge\nge (18).jpg	nge	o\o (118).jpg	o	pe\pe (17).jpg	pe
nge\nge (204).jpg	nge	o\o (122).jpg	o	pe\pe (236).jpg	pe
nge\nge (208).jpg	nge	o\o (123).jpg	o	pe\pe (253).jpg	pe
nge\nge (217).jpg	nge	o\o (125).jpg	o	pe\pe (5).jpg	pe
nge\nge (229).jpg	nge	o\o (126).jpg	o	pe\pe (58).jpg	pe
nge\nge (239).jpg	nge	o\o (127).jpg	o	pe\pe (62).jpg	pe
nge\nge (26).jpg	nge	o\o (129).jpg	o	pe\pe (63).jpg	pe
nge\nge (33).jpg	nge	o\o (130).jpg	o	pe\pe (67).jpg	pe
nge\nge (52).jpg	nge	o\o (22).jpg	o	pe\pe (68).jpg	pe
nge\nge (56).jpg	nge	o\o (47).jpg	o	pe\pe (69).jpg	he
nge\nge (57).jpg	nge	o\o (61).jpg	o	pe\pe (71).jpg	pe
nge\nge (59).jpg	nge	o\o (63).jpg	o	pe\pe (72).jpg	pe
nge\nge (60).jpg	nge	o\o (7).jpg	o	pe\pe (73).jpg	pe
nge\nge (63).jpg	nge	p\p (13).jpg	p	pe\pe (75).jpg	pe
nge\nge (64).jpg	nge	p\p (139).jpg	p	po\po (16).jpg	po
ngo\ngo (100).jpg	ngo	p\p (140).jpg	p	po\po (227).jpg	po
ngo\ngo (101).jpg	ngo	p\p (142).jpg	p	po\po (235).jpg	po
ngo\ngo (102).jpg	ngo	p\p (147).jpg	y	po\po (242).jpg	po
ngo\ngo (103).jpg	ngo	p\p (155).jpg	p	po\po (256).jpg	po
ngo\ngo (104).jpg	ngo	p\p (18).jpg	p	po\po (57).jpg	po
ngo\ngo (105).jpg	ngo	p\p (26).jpg	p	po\po (61).jpg	po
ngo\ngo (107).jpg	ngo	p\p (28).jpg	p	po\po (63).jpg	po
ngo\ngo (108).jpg	ngo	p\p (37).jpg	p	po\po (64).jpg	po
ngo\ngo (109).jpg	ngo	p\p (38).jpg	p	po\po (69).jpg	po
ngo\ngo (111).jpg	ngo	p\p (42).jpg	p	po\po (70).jpg	po
ngo\ngo (17).jpg	ngo	p\p (62).jpg	p	po\po (71).jpg	po
ngo\ngo (24).jpg	ngo	p\p (7).jpg	p	po\po (74).jpg	po
ngo\ngo (29).jpg	ngo	p\p (73).jpg	p	po\po (75).jpg	po
ngo\ngo (4).jpg	ngo	pa\pa (100).jpg	pa	po\po (76).jpg	po
ngo\ngo (40).jpg	ngo	pa\pa (102).jpg	pa	r\r (252).jpg	r
no\no (14).jpg	no	pa\pa (32).jpg	pa	r\r (262).jpg	r
no\no (26).jpg	no	pa\pa (33).jpg	pa	r\r (274).jpg	r
no\no (29).jpg	o	pa\pa (83).jpg	pa	r\r (297).jpg	r
no\no (5).jpg	no	pa\pa (84).jpg	pa	r\r (77).jpg	r
no\no (50).jpg	no	pa\pa (85).jpg	pa	r\r (86).jpg	r
no\no (54).jpg	no	pa\pa (86).jpg	pa	r\r (87).jpg	r
no\no (55).jpg	no	pa\pa (87).jpg	pa	r\r (88).jpg	r
no\no (57).jpg	no	pa\pa (88).jpg	pa	r\r (89).jpg	r
no\no (58).jpg	no	pa\pa (89).jpg	pa	r\r (90).jpg	r
no\no (59).jpg	no	pa\pa (90).jpg	pa	r\r (91).jpg	r
no\no (63).jpg	no	pa\pa (97).jpg	pa	r\r (92).jpg	r
no\no (64).jpg	no	pa\pa (98).jpg	pa	r\r (93).jpg	r
o\o (116).jpg	o	pa\pa (99).jpg	pa	r\r (94).jpg	r
o\o (117).jpg	o	pe\pe (11).jpg	pe	r\r (95).jpg	r

Filename	Pred.	Filename	Pred.	Filename	Pred.
ra\ra (133).jpg	ra	ro\ro (91).jpg	ro	se\se (68).jpg	se
ra\ra (134).jpg	ra	s\s (100).jpg	s	se\se (69).jpg	se
ra\ra (135).jpg	ra	s\s (103).jpg	s	so\so (150).jpg	so
ra\ra (136).jpg	ra	s\s (113).jpg	s	so\so (159).jpg	so
ra\ra (137).jpg	ra	s\s (124).jpg	s	so\so (173).jpg	so
ra\ra (139).jpg	ra	s\s (134).jpg	s	so\so (184).jpg	so
ra\ra (141).jpg	ra	s\s (145).jpg	s	so\so (195).jpg	so
ra\ra (142).jpg	ra	s\s (165).jpg	s	so\so (30).jpg	so
ra\ra (143).jpg	ra	s\s (54).jpg	s	so\so (31).jpg	so
ra\ra (144).jpg	ra	s\s (57).jpg	s	so\so (33).jpg	so
ra\ra (246).jpg	ra	s\s (58).jpg	s	so\so (35).jpg	so
ra\ra (261).jpg	ra	s\s (59).jpg	s	so\so (36).jpg	so
ra\ra (272).jpg	ra	s\s (60).jpg	s	so\so (37).jpg	so
ra\ra (73).jpg	ra	s\s (62).jpg	s	so\so (38).jpg	so
ra\ra (85).jpg	ra	s\s (63).jpg	s	so\so (39).jpg	so
re\re (113).jpg	re	s\s (67).jpg	sa	so\so (40).jpg	so
re\re (117).jpg	re	sa\sa (19).jpg	sa	so\so (41).jpg	so
re\re (118).jpg	re	sa\sa (193).jpg	sa	t\t (213).jpg	t
re\re (119).jpg	re	sa\sa (199).jpg	sa	t\t (222).jpg	t
re\re (120).jpg	re	sa\sa (205).jpg	sa	t\t (229).jpg	t
re\re (121).jpg	re	sa\sa (29).jpg	sa	t\t (232).jpg	t
re\re (122).jpg	re	sa\sa (30).jpg	sa	t\t (242).jpg	t
re\re (123).jpg	re	sa\sa (32).jpg	sa	t\t (258).jpg	t
re\re (124).jpg	re	sa\sa (33).jpg	sa	t\t (269).jpg	t
re\re (125).jpg	re	sa\sa (34).jpg	sa	t\t (47).jpg	t
re\re (126).jpg	re	sa\sa (35).jpg	sa	t\t (48).jpg	t
re\re (127).jpg	re	sa\sa (36).jpg	sa	t\t (50).jpg	t
re\re (227).jpg	re	sa\sa (37).jpg	sa	t\t (52).jpg	t
re\re (238).jpg	re	sa\sa (38).jpg	sa	t\t (54).jpg	t
re\re (272).jpg	re	sa\sa (41).jpg	sa	t\t (58).jpg	t
ro\ro (118).jpg	ro	sa\sa (5).jpg	sa	t\t (59).jpg	t
ro\ro (119).jpg	ro	se\se (17).jpg	se	t\t (64).jpg	t
ro\ro (121).jpg	ro	se\se (181).jpg	se	ta\ta (15).jpg	ta
ro\ro (122).jpg	ro	se\se (192).jpg	se	ta\ta (207).jpg	ta
ro\ro (124).jpg	ro	se\se (200).jpg	se	ta\ta (219).jpg	ta
ro\ro (125).jpg	ro	se\se (5).jpg	se	ta\ta (27).jpg	ta
ro\ro (126).jpg	ro	se\se (55).jpg	se	ta\ta (4).jpg	ta
ro\ro (128).jpg	ro	se\se (57).jpg	se	ta\ta (43).jpg	ta
ro\ro (130).jpg	ro	se\se (59).jpg	se	ta\ta (44).jpg	ta
ro\ro (265).jpg	ro	se\se (61).jpg	se	ta\ta (74).jpg	ta
ro\ro (278).jpg	ro	se\se (62).jpg	se	ta\ta (75).jpg	ta
ro\ro (290).jpg	ro	se\se (63).jpg	se	ta\ta (76).jpg	ta
ro\ro (296).jpg	ro	se\se (65).jpg	se	ta\ta (77).jpg	ta
ro\ro (302).jpg	do	se\se (67).jpg	se	ta\ta (78).jpg	ta

Filename	Pred.	Filename	Pred.	Filename	Pred.
ta\ta (83).jpg	ta	w\w (59).jpg	w	wo\wo (30).jpg	ye
ta\ta (84).jpg	ta	w\w (60).jpg	w	wo\wo (32).jpg	wa
ta\ta (85).jpg	ta	w\w (63).jpg	w	wo\wo (45).jpg	wo
te\te (109).jpg	te	w\w (65).jpg	w	wo\wo (55).jpg	wo
te\te (110).jpg	te	wa\wa (168).jpg	wa	wo\wo (8).jpg	wo
te\te (111).jpg	te	wa\wa (170).jpg	wa	y\y (252).jpg	y
te\te (113).jpg	te	wa\wa (177).jpg	wa	y\y (254).jpg	y
te\te (115).jpg	te	wa\wa (189).jpg	wa	y\y (260).jpg	p
te\te (116).jpg	te	wa\wa (29).jpg	wa	y\y (266).jpg	y
te\te (120).jpg	te	wa\wa (30).jpg	wa	y\y (279).jpg	y
te\te (121).jpg	te	wa\wa (31).jpg	wa	y\y (285).jpg	p
te\te (123).jpg	te	wa\wa (32).jpg	wa	y\y (290).jpg	y
te\te (17).jpg	te	wa\wa (34).jpg	wa	y\y (62).jpg	y
te\te (38).jpg	te	wa\wa (35).jpg	wa	y\y (63).jpg	ya
te\te (48).jpg	te	wa\wa (36).jpg	wa	y\y (65).jpg	y
te\te (53).jpg	te	wa\wa (38).jpg	wa	y\y (67).jpg	y
te\te (62).jpg	te	wa\wa (39).jpg	wa	y\y (68).jpg	y
te\te (7).jpg	te	wa\wa (40).jpg	wa	y\y (69).jpg	y
to\to (12).jpg	to	wa\wa (41).jpg	wa	y\y (70).jpg	p
to\to (15).jpg	to	we\we (16).jpg	we	y\y (71).jpg	y
to\to (19).jpg	to	we\we (166).jpg	we	ya\ya (10).jpg	ya
to\to (28).jpg	to	we\we (167).jpg	we	ya\ya (13).jpg	ya
to\to (35).jpg	to	we\we (168).jpg	we	ya\ya (14).jpg	ya
to\to (44).jpg	to	we\we (171).jpg	we	ya\ya (15).jpg	ya
to\to (5).jpg	to	we\we (173).jpg	we	ya\ya (210).jpg	ya
to\to (61).jpg	to	we\we (175).jpg	we	ya\ya (214).jpg	ya
to\to (63).jpg	to	we\we (177).jpg	we	ya\ya (22).jpg	ya
to\to (64).jpg	to	we\we (178).jpg	we	ya\ya (221).jpg	ya
to\to (66).jpg	to	we\we (19).jpg	wa	ya\ya (224).jpg	ya
to\to (67).jpg	to	we\we (26).jpg	wa	ya\ya (230).jpg	ya
to\to (68).jpg	to	we\we (37).jpg	we	ya\ya (24).jpg	ya
to\to (72).jpg	to	we\we (47).jpg	o	ya\ya (250).jpg	ya
to\to (74).jpg	to	we\we (7).jpg	we	ya\ya (262).jpg	ya
w\w (191).jpg	w	we\we (8).jpg	we	ya\ya (6).jpg	ya
w\w (202).jpg	w	wo\wo (133).jpg	wo	ya\ya (7).jpg	ya
w\w (232).jpg	w	wo\wo (134).jpg	wo	ye\ye (144).jpg	ye
w\w (234).jpg	w	wo\wo (136).jpg	wo	ye\ye (148).jpg	ye
w\w (241).jpg	w	wo\wo (137).jpg	wo	ye\ye (15).jpg	ye
w\w (245).jpg	w	wo\wo (138).jpg	wo	ye\ye (150).jpg	ye
w\w (46).jpg	w	wo\wo (140).jpg	wo	ye\ye (151).jpg	ye
w\w (47).jpg	w	wo\wo (142).jpg	wo	ye\ye (152).jpg	ye
w\w (49).jpg	w	wo\wo (16).jpg	wo	ye\ye (153).jpg	ye
w\w (52).jpg	w	wo\wo (21).jpg	la	ye\ye (154).jpg	ye
w\w (53).jpg	w	wo\wo (28).jpg	wa	ye\ye (156).jpg	ye

Filename	Pred.		Filename	Pred.		Filename	Pred.
ye\ye (18).jpg	ye		yo\yo (33).jpg	yo		yo\yo (83).jpg	yo
ye\ye (32).jpg	ye		yo\yo (45).jpg	yo		yo\yo (85).jpg	yo
ye\ye (37).jpg	ye		yo\yo (57).jpg	yo		yo\yo (86).jpg	yo
ye\ye (46).jpg	ye		yo\yo (76).jpg	yo		yo\yo (89).jpg	yo
ye\ye (5).jpg	ye		yo\yo (79).jpg	yo		yo\yo (9).jpg	yo
ye\ye (6).jpg	ye		yo\yo (80).jpg	yo		yo\yo (90).jpg	yo
yo\yo (21).jpg	yo		yo\yo (81).jpg	yo		yo\yo (91).jpg	yo

Training Plot in Jupyter Notebook



Appendix F

WORKING TITLE FORM

Republic of the Philippines
SOUTHERN LEYTE STATE UNVERSITY
Sogod, Southern Leyte
Website: www.slsuonline.edu.ph
Email: slsumaincampus@gmail.com
op@slsuonline.edu.ph
Telefax No. (053) 382-3294

College of Computer Studies and Information Technology

Proponents/Researchers:

1) Jack E. Bihay
2) Marco Erano P. Pahamotang
3) Carl John Albert S. Solo
4) Andrey N. Mejares
5) Lourence B. Aure

Proposed Project Title:

Image-Based Baybayin to Tagalog Word Translator using Deep Learning Algorithm

Submitted by: **JACK E. BIHAY** (Signature of Project Manager over printed name) Date: July 1, 2021	**Noted by:** **JANNIE FLEUR V. ORAÑO, MCS** (Signature of Adviser over printed name) Date:
Recommending Approval: **RHODERICK MALANGSA, DIT** (Signature of Patent Searcher over printed name) Date:	**Approved:** **ALEX C. BACALLA, DIT** (Signature of the Dean over printed name) Date:

Appendix G

GRAMMARIAN'S CERTIFICATION

Republic of the Philippines
SOUTHERN LEYTE STATE UNIVERSITY
Sogod, Southern Leyte
Website: www.slsuonline.edu.ph
Email: slsumaincampus@gmail.com, op@slsuonline.edu.ph
Telefax No. (053) 382-3294

College of Computer Studies and Information Technology

Date: _____

GRAMMARIAN'S CERTIFICATE

This is to certify that the undersigned has reviewed and went through all the pages of the proposal project study / research entitled "Image-Based Baybayin to Tagalog Translator using Deep Learning Algorithm" as against the set of structural rules that governs the composition of sentences, phrases and words in the English language.

Signed:

_____

Grammarian

Conforme:

Jack E. Bihay
Project Manager

Appendix H

CURRICULUM VITAE

JACK E. BIHAY

Zone V, Sogod Southern Leyte
+63936 – 379 – 5975
jackbihay14@gmail.com

OBJECTIVES

To become a part of a team that would enhance my knowledge and develop my skills. To find a place where I can showcase my strengths and gain courage to face my weaknesses.

PERSONAL INFORMATION

Date of Birth	: April 14, 1999
Place of Birth	: Zone V Sogod Southern Leyte
Civil Status	: Single
Citizenship	: Filipino
Height	: 5'7"
Weight	: 49kg
Age	: 22 years old
Blood Type	: O
Gender	: Male
Religion	: Roman Catholic

EDUCATIONAL ATTAINMENT

Level	School Attended	Year Graduated
Primary	Sogod Central Elementary School Zone 1 Sogod Southern Leyte	2011-2012
Secondary	Sogod National High School San Roque Sogod Southern Leyte	2015-2016
Senior High	Southern Leyte State University – Main Sogod Southern Leyte	2017-2018
Tertiary	Southern Leyte State University – Main Sogod Southern Leyte	2018 - present

SKILLS

- Microsoft Office Literate
- Photo Editing and Graphic Designing (Adobe Photoshop, Adobe Illustrator, Adobe XD)
- Video Editing (Sony Vegas, Adobe Premiere)
- Programming Skills (C#, Vb.Net, Java, PhP, Python)

SEMINARS, TRAININGS AND WORKSHOP

Microsoft Office Specialist
December 2019
SLSU Main Campus

CHARACTER REFERENCES

Dr. Alex C. Bacalla, DIT
Dean, College of Computer Studies in Information Technology
SLSU, Sogod Southern Leyte

Dr. James Brian Flores, Ph.D
Department Head, Bachelor of Science in Information Technology
SLSU, Sogod Southern Leyte

I HEREBY CERTIFY that the above information is true and correct to the best of my knowledge and belief.

MARCO ERAÑO P. PAHAMOTANG

T. Dagohoy Street Zone I, Sogod, Southern Leyte
+63975 – 639 – 5386
pahamotangme13@gmail.com

OBJECTIVES

To become a part of a team that would enhance my knowledge and develop my skills. To find a place where I can showcase my strengths and gain courage to face my weaknesses.

PERSONAL INFORMATION

Date of Birth	: January 13, 2000
Place of Birth	: Zone 1 Sogod Southern Leyte
Civil Status	: Single
Citizenship	: Filipino
Height	: 5'7"
Weight	: 42kg
Age	: 22
Blood Type	: O
Gender	: Male
Religion	: Iglesia ni Cristo

EDUCATIONAL ATTAINMENT

Level	School Attended	Year Graduated
Primary	Sogod Central Elementary School Zone 1 Sogod Southern Leyte	2011-2012
Secondary	Sogod National High School San Roque Sogod Southern Leyte	2015-2016
Senior High	Southern Leyte State University – Main Sogod Southern Leyte	2017-2018
Tertiary	Southern Leyte State University – Main Sogod Southern Leyte	2018 - present

SKILLS

- Microsoft Office Literate
- Photo Editing and Graphic Designing (Adobe Photoshop, Adobe Illustrator, Adobe XD)
- Video Editing (Sony Vegas, Adobe Premiere)
- Programming Skills (C#, Vb.Net, Java, PhP, Python)

SEMINARS, TRAININGS AND WORKSHOP

MAPA:Bulig Guiding the Youth to Community Mapping
15th of November 2021
Webinar Hosted by UP Tacloban and SLSU GIS Tech Center

CHARACTER REFERENCES

Dr. Alex C. Bacalla, DIT
Dean, College of Computer Studies in Information Technology
SLSU, Sogod Southern Leyte

Dr. James Brian Flores, Ph.D
Department Head, Bachelor of Science in Information Technology
SLSU, Sogod Southern Leyte

I HEREBY CERTIFY that the above information is true and correct to the best of my knowledge and belief.

CARL JOHN ALBERT S. SOLO

Dolho, Bato Leyte
+63951 – 799 – 6843
solocarl27@gmail.com

OBJECTIVES

To become a part of a team that would enhance my knowledge and develop my skills. To find a place where I can showcase my strengths and gain courage to face my weaknesses.

PERSONAL INFORMATION

Date of Birth	: March 27, 2000
Place of Birth	: Dolho, Bato Leyte
Civil Status	: Single
Citizenship	: Filipino
Height	: 5'7"
Weight	: 60kg
Age	: 21
Blood Type	: O
Gender	: Male
Religion	: Roman Catholic

EDUCATIONAL ATTAINMENT

Level	School Attended	Year Graduated
Primary	Bato Central Elementary School Tinago, Bato Leyte	2011-2012
Secondary	Bato School of Fisheries Tinago, Bato Leyte	2015-2016
Senior High	Bato School of Fisheries Tinago, Bato Leyte	2017-2018
Tertiary	Southern Leyte State University – Main Sogod Southern Leyte	2018 - present

SKILLS

- Microsoft Office Literate
- Photo Editing and Graphic Designing (Adobe Photoshop, Adobe Illustrator, Adobe XD)
- Video Editing (Sony Vegas, Adobe Premiere)
- Programming Skills (C#, Vb.Net, Java, PhP, Python)

SEMINARS, TRAININGS AND WORKSHOP

Microsoft Office Specialist
December 2019
SLSU Main Campus

CHARACTER REFERENCES

Dr. Alex C. Bacalla, DIT
Dean, College of Computer Studies in Information Technology
SLSU, Sogod Southern Leyte

Dr. James Brian Flores, Ph.D
Department Head, Bachelor of Science in Information Technology
SLSU, Sogod Southern Leyte

I HEREBY CERTIFY that the above information is true and correct to the best of my knowledge and belief.

ANDREY N. MEJARES

Ubos, San Francisco Southern Leyte
+63965 – 924 – 3337
ndrmejares@gmail.com

OBJECTIVES

To become a part of a team that would enhance my knowledge and develop my skills. To find a place where I can showcase my strengths and gain courage to face my weaknesses.

PERSONAL INFORMATION

Date of Birth	: July 11, 1999
Place of Birth	: Bagbaguin, Valenzuela City
Civil Status	: Single
Citizenship	: Filipino
Height	: 5'4"
Weight	: 42kg
Age	: 22
Blood Type	: A
Gender	: Male
Religion	: Roman Catholic

EDUCATIONAL ATTAINMENT

Level	School Attended	Year Graduated
Primary	Central Elementary School Central, San Francisco, Southern Leyte	2011-2012
Secondary	CAASAFI Central San Francisco, Southern Leyte	2015-2016
Senior High	Sta. Paz National High School Pasanon, San Francisco, Southern Leyte	2017-2018
Tertiary	Southern Leyte State University – Main Sogod Southern Leyte	2018 - present

SKILLS

- Microsoft Office Literate
- Photo Editing and Graphic Designing (Adobe Photoshop, Adobe Illustrator, Adobe XD)
- Programming Skills (C#, Java)

SEMINARS, TRAININGS AND WORKSHOP

Microsoft Office Specialist
December 2019
SLSU Main Campus

CHARACTER REFERENCES

Dr. Alex C. Bacalla, DIT
Dean, College of Computer Studies in Information Technology
SLSU, Sogod Southern Leyte

Dr. James Brian Flores, Ph.D
Department Head, Bachelor of Science in Information Technology
SLSU, Sogod Southern Leyte

I HEREBY CERTIFY that the above information is true and correct to the best of my knowledge and belief.

LOURENCE B. AURE

Pinamudlan, San Francisco Southern Leyte
+63975 – 7715 – 765
aurelourence22@gmail.com

OBJECTIVES

To become a part of a team that would enhance my knowledge and develop my skills. To find a place where I can showcase my strengths and gain courage to face my weaknesses.

PERSONAL INFORMATION

Date of Birth	: September 25, 1999
Place of Birth	: Pinamudlan, San Francisco, Southern Leyte
Civil Status	: Single
Citizenship	: Filipino
Height	: 5'5"
Weight	: 56kg
Age	: 22
Blood Type	: B+
Gender	: Male
Religion	: Roman Catholic

EDUCATIONAL ATTAINMENT

Level	School Attended	Year Graduated
Primary	Pinamudlan Elementary School Pinamudlan, San Francisco, Southern Leyte	2011-2012
Secondary	Marayag National High School Marayag San Francisco, Southern Leyte	2015-2016
Senior High	Sta. Paz National High School Pasanon, San Francisco, Southern Leyte	2017-2018
Tertiary	Southern Leyte State University – Main Sogod Southern Leyte	2018 - present

SKILLS

- Microsoft Office Literate
- Photo Editing and Graphic Designing (Adobe Photoshop, Adobe Illustrator, Adobe XD)
- Video Editing (Sony Vegas, Adobe Premiere)
- Programming Skills (C#, Vb.Net, Java, PhP, Python)

SEMINARS, TRAININGS AND WORKSHOP

Microsoft Office Specialist
December 2019
SLSU Main Campus

CHARACTER REFERENCES

Dr. Alex C. Bacalla, DIT
Dean, College of Computer Studies in Information Technology
SLSU, Sogod Southern Leyte

Dr. James Brian Flores, Ph.D
Department Head, Bachelor of Science in Information Technology
SLSU, Sogod Southern Leyte

I HEREBY CERTIFY that the above information is true and correct to the best of my knowledge and belief.

GLOSSARY

Adaptive Thresholding – is the method where the threshold value is calculated for smaller regions and therefore, there will be different threshold values for different regions.

Baybayin – is a Philippine script. The script is an alphasyllabary belonging to the family of the Brahmic scripts. It was widely used in Luzon and other parts of the Philippines prior to and during the 16th and 17th centuries before being replaced by the Latin alphabet during the period of Spanish colonization.

Classification – is a task that requires the use of machine learning algorithms that learn how to assign a class label to examples from the problem domain. An easy-to-understand example is classifying emails as "spa*m*" or "not spam."

Convolutional Neural Network – is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data.

Deep Learning – is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks.

Extractions – is a part of the dimensionality reduction process, in which, an initial set of the raw data is divided and reduced to more manageable groups.

Kudlits – is a mark that is used in Baybayin characters that will remove the /a/ sound and put in top to make it sound /e/ or /i/ and put it below to sound as /o/ or /u/

Optical Character Recognition – is the use of technology to distinguish printed or handwritten text characters inside digital images or physical documents, such as scanned paper document. This process involves examining the text and translating the characters into code that can be used for data processing.

Region of Interest – is a portion of an image that you want to filter or operate on in some way. You can represent an ROI as a binary mask image. In the mask image, pixels that belong to the ROI are set to 1 and pixels outside the ROI are set to 0.

Segmentation – is a method in which a digital image is broken down into various subgroups called Image segments which helps in reducing the complexity of the image to make further processing or analysis of the image simpler.

Transliteration - is the process of transferring a word from the alphabet of one language to another