Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processing and Deep Learning

by

Cruz, Julian Jesus P.
Gonzales, Joshua Peterson T.
Pichay, Kayley Mae T.
Suamina, Hyan Jan Y.

Submitted in Partial Fulfillment of the Requirements for the Degree of

Bachelor of Science in Electronics Engineering

at

FEU Institute of Technology

June 2024

Pocholo James M. Loresco, Ph. D., PECE
Thesis Adviser

©2024 Cruz, Gonzales, Pichay, Suamina.
All Rights Reserved

The authors grant FEU Institute of Technology permission to reproduce and distribute the contents of this document in whole or in part.

APPROVAL AND ACCEPTANCE SHEET

The thesis entitled **"Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processing and Deep Learning"** prepared and submittedby:

Cruz, Julian Jesus P.
Gonzales, Joshua Peterson T.
Pichay, Kayley Mae T.
Suamina, Hyan Jan Y.

In partial fulfillment of the course requirement for the Degree of Bachelor of Science in Electronics Engineering has been examined and is hereby recommended for approval.

Engr. Danilyn Joy O. Aquino
Panelist 1

Engr. Juan Miguel H. Villarroel
Panelist 2

Engr. Rex Paolo C. Gamara
Head Panelist

Accepted as partial fulfillment of the requirements for the **Degree of Bachelor of Science in Electronics Engineering.**

Pocholo James M. Loresco, Ph. D., PECE
Thesis Adviser

Engr. Antipas T. Teologo Jr.
Course Adviser

Engr. Antipas T. Teologo Jr.
Department Head

Date

ACKNOWLEDGEMENT

The school year started with the students being tasked to make a research proposal from start to finish. The students persevered and fought through every challenge to reach the end goal. This time the students sacrificed to make the prototype and gave lots of knowledge and ideas for future projects.

The researchers would like to thank God for everything He did, guiding them throughout this journey and giving them the strength to accomplish this research. The blessing He bestowed upon the researchers helped them through their academic lives.

The researchers are very thankful to their thesis adviser, Dr. Pocholo James M. Loresco, for the guidance and advice that he gave that made what the study is today. His knowledge and ideas inspired the researchers to continue pursuing their study.

To the head of the Electronics Engineering department, as well as the course adviser for this subject, Engr. Antipas T. Teologo, together with their inspiring panelists, Engr. Rex Paolo C. Gamara, Engr. Danilyn O. Aquino, and Engr. Juan Miguel H. Villarroel. The researchers are thankful for taking time out of their way, to share their knowledge and to give comments on how to further improve the research study.

The researchers are also thankful to other faculty members of the Electronics Engineering Department, together with their friends and families for giving them the energy they need to complete their research.

Lastly, this research is the outcome of their sacrifices, passion, perseverance, and devotion to complete this task. They owe it all to their beloved parents, who never stopped giving them their love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS

3D - 3 Dimensional

ADDIE - Analysis, Design, Development, Implementation, and Evaluation

AI - Artificial Intelligence

AK - Alphabet Knowledge

ALD - Assistive Listening Device

API - Application Programming Interface

ASL - American Sign Language

ASLR - American Sign Language Recognition

AT - Assistive Technologies

ArSL - Arabic Sign Language

BRISK -Binary Robust Invariant Scalable Key

CAI - Computer-Assisted Instruction

CNN - Convolutional Neural Network

DC - Degree of Certainty

DL - Deep Learning

DNN - Deep Neural Network

DOF - Degrees of Freedom

DOST - Department of Science and Technology

FC - Fully Connected

FN - False Negative

FP - False Positive

FSL - Filipino Sign Language

GPU - Graphics Processing Unit

GTTS - Google Text-to-Speech

GUI - Graphical User Interface

HMM - Hidden Markov Model

HSV - Hue Saturation Value Color Model

ICT - Information and Communication Technologies

IDEA - Individuals with Disabilities Education Act

ISL - Indian Sign Language

KNN - K-Nearest Neighbors Algorithm

LSTM - Long Short-Term Memory

MATLAB - Matrix Laboratory

MCC - Matthew's Correlation Coefficient

ML -Machine Learning

MSL - Mexican Sign Language

MSL-VRT - MSL-Virtual Robotic Trainer

N - Negative

NSO - National Statistical Office

P - Positive

PC - Personal Computer

PCB - Printed Circuit Board

PID - Process Identifier

RA - Republic Act

RNN - Recurrent Neural Network

SIFT - Scale Invariant Feature Transform

SLR - Sign Language Recognition

SPED/ SpEd - Special Education

SQL - Structured Query Language

SURF - Speed Up Robust Features

TN - True Negative

TP - True Positive

V - Vocabulary Development

WHO - World Health Organization

Chapter 1

INTRODUCTION

In this chapter, a comprehensive perspective of the study is presented, addressing the identified issue and its engineering resolution. The study's background, its significance, objectives, scope, and limitations, along with the conceptual framework are addressed. Furthermore, this chapter provides definitions for the terms related to the study.

1.1. Background of the Study

Communication between people is important to be able to form social relationships with each other. It also plays a significant role in a person's everyday life. People who communicate with one another have a clear understanding of what the other person is trying to point out. However, there are groups of people like the deaf, individuals who suffer hearing loss, and the mute community who have other ways to communicate with other people. An example of this is sign language, which is a non-verbal communication system that consists of hand and body movements to represent words [1] . In certain circumstances, deaf people are also considered mute, which causes them to face social isolation, miscommunications, and workplace discrimination [2].

In the Philippines, a study that was conducted by the National Statistical Office (NSO) in 2000 shows that the number of Filipinos that have disabilities is 942,098, where 352,398 of those people have a disability in both hearing and speaking, and the number of deaf people in the total population is 76,731 [3]. Recently, this number has increased with up to 1.23% of the total population considered to be deaf, mute, or hearing impaired [4].

The World Health Organization also expects an estimate of more than 700 million people to suffer with hearing disabilities by 2050 in which they will need some form of hearing rehabilitation [5]. These rehabilitation methods will include the use of sign language as a form of communication.

To bridge the gap that exists in communication between the deaf and mute communities and other individuals, sign language is often utilized to convey messages between parties. The official sign language of the Philippines' deaf minority is called the Filipino Sign Language (FSL) and it is also the official sign language of the government in transactions that involve people who have a deaf disability, which is under the Republic Act No. 11106 [6]. The use of sign language is an essential tool for the deaf and hearing-impaired community of not only the Philippines but the whole world.

Even with the use of sign language for communication, challenges still exist among the disabled community with the most considered as the lack of education in proper sign language application. Learning sign language is often seen as a difficult task for hearing people leading to a lack of knowledge in its usage when communicating with deaf and mute individuals [7]. Furthermore, out of the 72 million deaf people in the world, only 4.3 million are capable of communicating through sign language which leaves the remaining 92% with limited or even no access to information sources [7]. Additionally, 80% of the total deaf community is uneducated with no proper learning from schools due to their disability as mentioned by the World Federation of the Deaf [8]. Due to this reason, technological advancements, specifically Assistive Technologies, including machine translation and sign language recognition have been developed to further lessen the challenges faced by the deaf community.

Through the years of technological enhancements, Assistive Technologies (AT)

have been useful to overcome limitations and improve the functions of an individual. These technologies are developed to help disabled individuals in their daily lives, reduce discrimination, and improve their quality of life [9]. There are two main applications where AT is most commonly utilized: the first is the medical application that tries to reduce physical or cognitive impairments, and the second is the social application that operates in the surrounding environment focusing on social barriers and discrimination [9]. For the deaf and mute communities, assistive technologies have also been developed to cater to their communication needs. These can be categorized into two, namely, wearables and vision-based. Wearable AT includes smart gloves which have been invented in the past years as a form of assistive device that translates sign language into speech or text. Flex sensors are used to indicate the bend of the fingers and the movement of the hands [10]. However, the challenging issue with these technologies is that it is invasive to the ones who will use the smart gloves and will limit and restrict the movements of the user. The use of gloves is often interfaced with microcontrollers, Printed Circuit Boards (PCB), wires, and other components which could also factor the size and weight of the device, and can be undesirable for the user. Furthermore, past studies have also faced connectivity issues between the gloves and the main processing computer limiting the AT [11].

Vision-based AT are those sign language recognition systems that were also made with the use of a color-based hand tracking system [12], or through the detection of hand gestures with special types of cameras such as 3D cameras, Kinect Cameras, or Stereo Cameras [13]. However, an approach with this sign language recognition architecture is still inadequate since it will require a higher computational power and is less accurate than other methods. A study conducted in 2020 [14] employed the use of a Microsoft Kinect as the camera for the Indian Sign Language recognition system and was only able to gather

an accuracy of 71.85% in detecting sign language with a slow response time. Moreover, the use of varying types of special cameras can also be expensive, which can affect accessibility.

Other methods for sign language recognition include machine learning [31], deep learning [12], and image processing [16]. These studies paved the way for AT to be non-invasive in recognizing gestures of sign language. These are more common in recent studies due to their better accuracy and response time features. However, most research studies interfaced with machine learning techniques are restricted to static signs which are gestures that do not require any movement. Other studies with image processing techniques also require a highly functional GPU which is quite expensive for most users [11].

Assistive technologies are also important in the education sector such that they are capable of addressing issues in communications, reading and writing, mobility, physical education, and mathematics [17]. Instructional technologies, a specific form of assistive technology, are often implemented inside classrooms to enhance the design and delivery of instruction of lessons through innovative technology which can impact positive learning. Assistive technologies are also used to enhance the instruction of communication in schools including sign languages. Most often, learning of sign languages occurs when individuals with deafness or hearing disabilities begin school and start socializing with other deaf children [18]. However, their education should be catered to them with specific approaches on how they learn various topics in school. Dr. Rosalie Ricasa mentions how the traditional learning of sign languages with “mouthing” of words is not enough as it only gets 30% of the exact message across to the intended recipient [18]. She further stated that teachers should find the right strategies that involves inclusion, integration, and right opportunities for learning with the deaf students.

Sign language tutoring applications are considered one of the assistive technologies developed to teach deaf and non-deaf students the use of proper sign language. Tutoring systems have been made with a bioinspired virtual controlled robotic arm to instruct students with static signs from the Mexican Sign Language [19]. Other similar studies have also used multimedia-based instruction approaches such as images and videos to display how sign languages such as Arabic Sign Language and Filipino Sign Language are done [20][21]. However, these technologies are either limited or invasive for the users. Some would require the use of specific gloves, cameras, and even restricted to static sign languages. These do not have the capability to detect and recognize the sign language input of the users as a form of assessment and provide feedback.

In this study, the proponents will utilize the recognition of sign language in developing a system for sign language tutoring. A computer application dedicated to tutoring learners with Filipino Sign Language acts in accordance with UNESCO's goal of "Leveraging ICT to achieve Education 2030" [22]. It is a project aimed at utilizing the potential of ICT in improving and developing innovative approaches in providing quality education and lifelong opportunities by 2030. Additionally, the International Convention on the Rights of People with Disabilities of the United Nations was made to promote research and innovations on technological advancements for elevating the quality of life of individuals suffering from disabilities [23]. The United Nations' Sustainable Development Goals also states the following objectives for Quality Education, Reduced Inequalities, Industry, Innovation, and Infrastructure which the proposed study will comply with [24].

In the context of the Philippines, a law was recently passed last 2018 officially recognizing FSL as the sign language of deaf Filipinos. Furthermore, it was also mandated to be used in transactions involving the deaf including broadcast media, and workplaces

and should even be taught in schools for SPED. Law Act No. 10533, also known as the Basic Education Act of 2013, also requires Kindergarten until third grade of elementary education to be taught in students' native language, including FSL as a SPED policy.

This study plans to develop an application that will help abled and disabled individuals to learn static, and dynamic Filipino sign language without the use of invasive AT. A system will be trained with FSL signs and gestures which will be processed into a personal computer (PC) to take a live video stream of the user's hand motion and signals. Through the program, the live video of Filipino Sign Language is processed with the Deep Learning system for a predictive translation of the gestures. Afterwards, the system will recognize and show whether the gestured sign language is correct or incorrect and highlights the specific parts that are deemed wrong. This will be made into an application wherein individuals will be tutored in static, and dynamic of Filipino sign language by the application.

This paper adheres to the artificial intelligence of the Harmonized National Research and Development Agenda for 2022-2028 of the Department of Science and Technology where "Image and Video Recognition" is one of the priorities for 2023 in the Artificial Intelligence section [25].

1.2. Significance of the Study

This study aims to develop a tutoring application for FSL utilizing deep learning and image processing. It will benefit the following:

1.2.1. Special Education Teachers

The main benefactors of this research are the Special Education Teachers

that has Filipino Sign Language in their curriculum. The proposed system will be used as a tutoring tool to help instruct their students effectively learn FSL.

1.2.2. Deaf Filipino Students

This study will benefit deaf students such that it allows them to effectively learn Filipino Sign Language with automated feedback based on their input. The project's prototype will significantly aid students in assessing their gestures whether they are doing it right or wrong.

1.2.3. Department of Science and Technology (DOST)

The concept and applications presented in the study will benefit the Department of Science and Technology (DOST). The study is in line with the Harmonized National Research and Development Agenda for 2022-2028 of the Department of Science and Technology whose major focus is on "Image and Video Recognition" which is one of the priorities of the Artificial Intelligence section.

1.2.4. Future Researchers

This study will serve as foundational research for further developments and innovations in educating deaf Filipinos and improve upon the limitations of the study. The theories, ideas, and processes done in the current study shall be used as a reference for future works.

1.3. Objectives

1.3.1. General Objective

The study's general objective is to develop an automatic Filipino Sign Language Tutor with Static, and Dynamic FSL Recognition using Image Processing and Deep Learning.

1.3.2. Specific Objectives

The study specifically aims to:

1. To develop a dataset or corpus of FSL gestures considering static and dynamic gestures obtained through data acquisition of videos from a camera;
2. To develop a Python-based program interfaced with a deep learning model for the FSL recognition, translation of FSL to text, and feedback evaluation based on the user's input;
3. To develop a GUI of the Python-based tutoring program incorporated with the loading of FSL lessons, FSL recognition, feedback evaluation and correction, and an audio interface;
4. To evaluate the performance of the tutoring system using various metrics including precision, recall, specificity, F1 score, Matthew's correlation coefficient, reliability formulas and confusion matrix based on its recognition of Filipino sign language with an obtained accuracy of at least 93.8%.
5. To evaluate and compare the effectiveness of Developing an Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processing and Deep Learning to the traditional methods of Filipino Sign Language tutoring using Likert Scale survey questionnaire.

1.4. Scope and Delimitations

The goal of this research is to develop a PC-based tutoring application that teaches Filipino Sign Language (FSL) as a computer-aided teaching tool for Special Education

(SPED) teachers. Republic Acts No. 10410 and 10533 recognize Filipino Sign Language as a medium of instruction and curriculum for the education of early childhood to the secondary level [26][27]. The study also aims to incorporate a machine learning system to input information on Filipino Sign Language hand gestures and the corresponding meaning of the hand signs. It also highlights an image processing system that detects the hand gesture that wants to input new data to the tutoring system. The proponents would like to emphasize that the validation of the tutoring application shall be limited to experts in FSL and special education teachers who teach FSL and not to students due to ethical considerations. For confirmation, there is a test that determines if the sign language performed is correct or incorrect, which will be displayed on the PC screen. This assessment will provide feedback that highlights the specific input of the user considered incorrect. An indicator of number 1 is inputted to the algorithm of the tutoring application if it is correct, and a number 0 if it is incorrect.

The system which is a machine learning and image processing system features an application that is capable of being a tutoring tool that assists special education teachers in teaching Filipino Sign Language. The proposed system also eliminates complicated set-up as it is simple to install that works on any PC device. Unlike the studies [28][29][30], the system proposed by the researchers will be non-invasive as it is only a PC based system. Considering the application with image processing and machine learning, the proposed accuracy of the tutoring system would be at least 93.8%.

The proposed study is intended for the use of Filipino Sign Language experts and special education teachers that have sign language in their curriculum. As a reference, studies like [15][29] only target the recognition of the FSL gesture and do not have any tutoring application or system. The tutoring system only focuses on teaching Filipino Sign

Language as it is the official sign language of the Philippines which is stated in RA 11106 [6]. Other sign languages from different countries are out of the scope of the study. The proposed application will test and train different gestures of the Filipino Sign Language that is inputted to the machine learning system. This is also limited to the camera's vision, meaning that when a sign language gesture is added to the application, the hands of the user must be clear and visible to the camera. Furthermore, the specialized education teacher will only use the application as a tutoring tool to help students advance their skills in Filipino Sign Language. Due to these reasons, the learning curve of the students will not be monitored and compared with traditional methods of teaching FSL.

The proposed prototype will only cover 50 signs based on the Department of Education's Kindergarten Basic Education Curriculum. Specifically, "Section G. Language, Literacy, and Communication" which covers Oral Language. To be exact, the curriculum's competencies in "Alphabet Knowledge (AK)", "Vocabulary Development (V)", "Use the proper expression in introducing oneself" and "Use polite greetings and courteous expressions in appropriate situations". The codes of the included core competencies are: "LLKVPD-Ia-13", "LLKOL-Ia-1", "LLKV-00-1", and "LLKV-00-2".

In conclusion, the developed system's design includes the following: Presentation (FSL Lessons), Assessment (FSL recognition of the user's input), and Feedback and Correction (Highlights the incorrect gesture).

1.5. Conceptual Framework

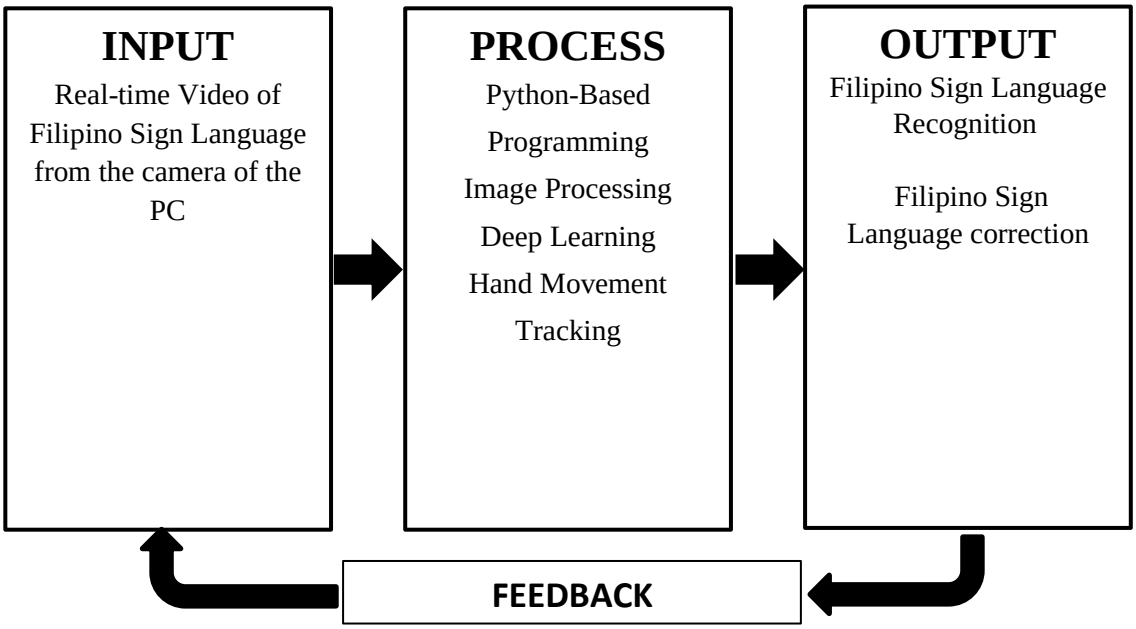


Figure 1. Conceptual Framework

The proposed study aims to develop real-time Filipino sign language (FSL) detection with the use of image processing and deep learning for FSL tutoring. The system begins by gathering real-time video footage of the FSL movements using the computer camera. The video will be the main input to be processed. The image processing is then implemented using Python-based programming to separate and evaluate hand gesture features such as shape, position, and movement. Deep learning is also trained on a variety of FSL gesture datasets, FSL learning materials, and FSL curriculum for recognizing these features. Once the FSL gestures are accurately recognized, the algorithm will analyze if the FSL detected is correct or not, only on the basis of 1 and 0. If the algorithm detects a wrong gesture, it will notify the user what part of the gesture is wrong and will project the correct way of doing it. After the projection of the correct FSL gesture, the system will have feedback, returning to gathering real-time video footage.

1.6. Definition of Terms

Artificial Intelligence (AI) – a specific field in computer science that incorporates machine automation and human intelligence encompassing Deep Learning and Machine Learning [31]

Assistive Technologies – this refers to any equipment or system that are utilized by individuals with disabilities to improve and increase their functional capabilities [32]

Deep Learning (DL) - refers to a subset of ML that trains computers on processing of data which imitates the architecture of a human brain's biological neural networks [33]

Filipino Sign Language (FSL) – inspired by the American Sign Language which caters to Filipinos and Filipino words and phrases [34]

Image Processing – a computational process that involves the use of mathematical algorithms for the analysis, manipulation, and processing of digital images to extract data [35]

Machine Learning (ML) - refers to a subset of AI that allows machines to make decisions based on the data it is exposed to [31]

Python - a high-level computer programming language used for website development, data analytics, and automation [36]

Chapter 2

REVIEW OF RELATED LITERATURE

This chapter of the study provides a thorough review of the literature on the research topic. It comprehensively examines previous studies and literature that served as the foundation for the concepts and ideas underpinning this thesis study.

2.1. Related Studies

In this section of the related literature review, it shows a thorough and critical examination of the work of foreign and local scholars on the topic in line with the thesis study. This review of relevant foreign and local related studies aims to contribute to and expand the existing knowledge based on the study's topic.

2.1.1. Foreign Related Studies

2.1.1.1. Assisted Sign Language Tutoring System Using a Bioinspired Virtual Controlled Robotic Arm

To effectively teach Mexican Sign Language to deaf and non-deaf people, the study developed a virtual robotic trainer platform that was able to mimic a total of 21 one-handed static signs of the language. The study utilized SolidWorks with a numerical evaluation of the control actions done within Simulink of Matlab. The main objective of this study was to develop an interactive platform called the MSL-Virtual Robotic Trainer (MSL-VRT) that would teach both deaf and non-deaf people Mexican Sign Language (MSL). The MSL-VRT has 25 degrees of freedom (DOF) and is modeled after the human upper limb, including the hand, forearm, and wrist. This allows it

to show finger-spelling and static, one-handed signs effectively, covering a total of 21 signs from the MSL alphabet. Furthermore, a proportional-integral-derivative controller was also created for the automation of the virtual robotic arm. The study showed a 90% accuracy in terms of the performance of the static signs, proving that the technology was a viable option in teaching Mexican Sign Language. This indicates that the MSL-Virtual Robotic Trainer (MSL-VRT) efficiently transmits dactylology and executes static-hand gestures in Mexican Sign Language (MSL). However, it must also be noted that it is still very limited such that it can only replicate static signs where they had to exclude some alphabets due to complex movements it cannot mimic [19].

2.1.1.2. An Intelligent System to Help Deaf Students Learn Arabic Sign Language

Figure 2. System Translation of ArSL into Speech and Text [20]

This study proposes a novel way to translate Arabic Sign Language based on a speech recognition engine and GTTS library. The process was divided into two sub-systems, namely, the Speech to ArSL and ArSL to Speech which were used on both disabled and normal individuals who are in need or interested in learning the Arabic sign language. The system was also interfaced with

tokenization to further break up words, phrases, or symbols into elements recognizable by the system. Another advantage of the system is its scalability and expandability due to its capability in adding more signs to the database accessible online. With a 99% recognition rate, the proposed system shows that it was effective in teaching the Arab deaf students sign language which improved significantly [20].

2.1.1.3. An Intelligent Computer-Based System for Sign Language Tutoring

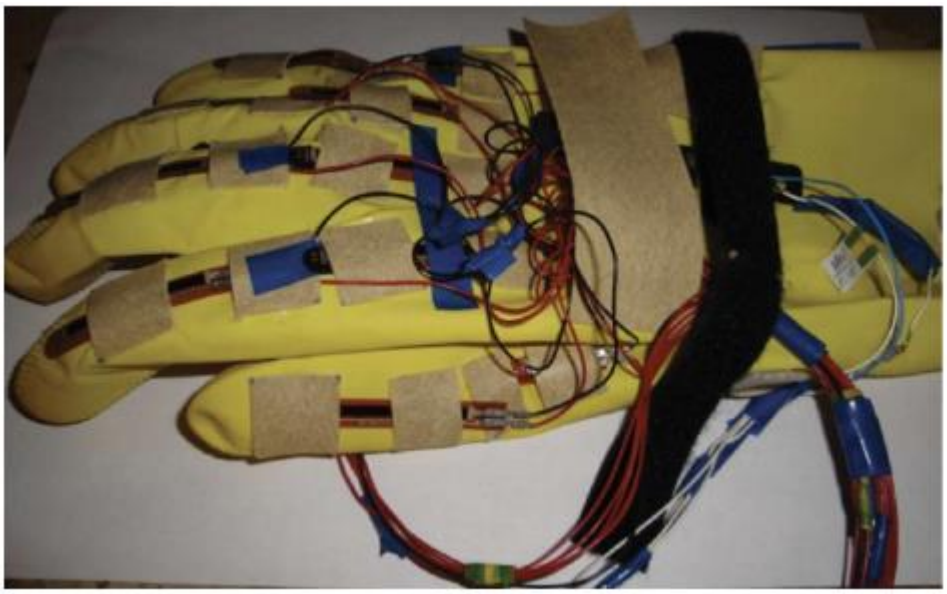


Figure 3. Data-Gloves of the Computer-Based SL Tutor [28]

The study proposes a computer-based system for tutoring sign language developed with a low-cost data glove. The study also uses a software application for the processing of movement signals for signs in real-time which uses a Pattern Machine technique for the decision if the user was able to closely replicate the teacher's sign movements. The data gloved used in this study provides 17 movement signals from the Bend sensors and push down switches which is initially used by the teacher for recording the selected signs. A total of four Arabic Sign

Language (ArSL) teachers have provided 65 common posture and gesture signs in which 10 users that were gathered in the local community were asked to test the system. The overall average accuracy in replicating the signs which was achieved during the testing is 93.8%, despite having the gestures harder to perform than the postures. Approximately 18% of the signs were difficult to replicate due to the specific thumb or finger, and wrist bending angles. After a usability questionnaire was conducted to the users and teachers, it revealed that they preferred the approach of the study to sign-language tutoring than the traditional human-based method that they have already experienced [28].

2.1.1.4. D-Talk: Sign Language Recognition System for People with Disability Using Machine Learning and Image Processing

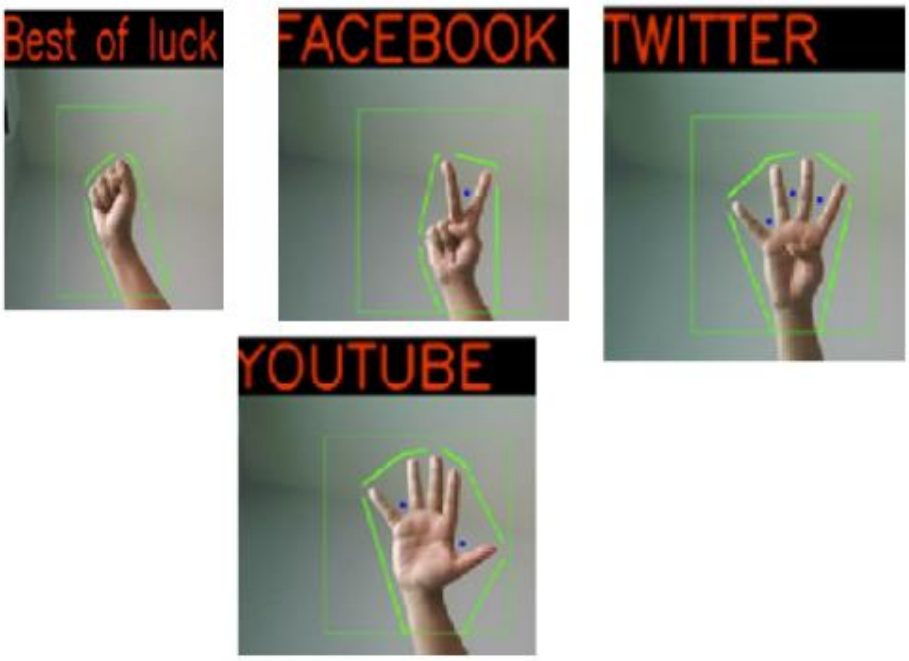


Figure 4. Signs Detected by the System [15]

With Artificial Intelligence, D-Talk was developed to provide detailed hand

gestures as interpretation for deaf and dumb users to effectively understand speech in communication. For this study, the Scale Invariant Feature Transform (SIFT) algorithm was used to extract information vectors from the image. Results showed that gesture interpretation was achieved, which was helpful for deaf individuals. However, lighting conditions severely affect the recognition of the sign language which leads to inaccurate hand segmentation resulting in inaccurate gesture prediction [15].

2.1.1.5. A Smart Glove with Integrated Triboelectric Nanogenerator for Self-powered Gesture Recognition and Language Experience

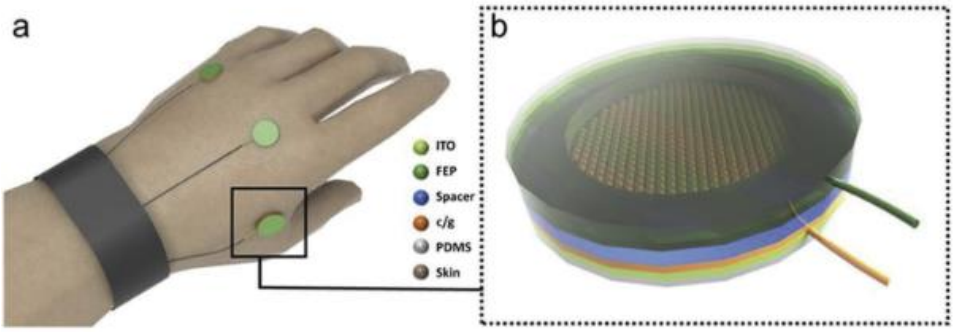


Figure 5. Illustration of Self-Powered Gesture Glove with TENG [29]

The study talks about the functional characteristics of flexible electronics and how it has proved to be a stepping stone in wearable devices. The study proposes a self-powered gesture-sensing system attached at the back of the hands, which can then distinguish hand gestures by measuring the output signal of the triboelectric nanogenerator. Since the sensor is attached to the back of the hand, the displacement of the tendons can be sensed which will detect the gestures. The sensor that is proposed in the study can be used as an electronic sign language translator by the process of converting the gestures into text [29].

2.1.1.6. Deep Learning-Based Sign Language Recognition System for Static Signs

The study proposes a convolutional neural network (CNN) deep learning-based system for Indian Sign Language. 35,000 sign images of 100 static signs were collected from different users for the study. The system's efficiency was evaluated on about 50 CNN models. The basis of different optimizers was also evaluated to get the results, and it was observed that the approach proposed achieved a training accuracy of 99.72% and 99.90% on both colored and grayscale images. The precision, recall, and F-score evaluation are also used to test its performance. However, the study achieved a high accuracy rating due to recognition of static signs only [12].

2.1.1.7. Providing Feedback in Ukrainian Sign Language Tutoring Software

Figure 6. Main Window of the Ukrainian SL Tutoring Application [37]

The proposed study focuses on the methods for video recognition that is implemented in the Ukrainian Sign Language Tutoring Software. The software verifies the understanding of users to signs and sentences. Body posture recognition methods are used to allow interaction with the users in the process of learning signs and their verification process. Feedback is provided by the software to the user

through capturing of a person's gesture using a web camera to improve the training success rate. The use of background modelling, image recognition, and machine learning methods are needed in the process of human posture reconstruction in a web camera in real-time. The proposed study was able to obtain a success rate of 91.7% with a test set of 85 signs [37].

2.1.1.8. Natural User Interface Based American Sign Language Tutoring Program

The proposed study designed a system that is an online system with interaction with people for them to learn American Sign Language (ASL) using hand-tracking peripherals, like Ultraleap's Leap Motion controller. A course structure was also designed in this study to teach the ASL most used words using time-spaced practices [38].

2.1.1.9. SignSpeaker: A Real-time, High-Precision SmartWatch-based Sign Language Translator

In this study, the first smartwatch-based end-to-end sentence-level ASLR system was proposed. This was implemented on a pair of commercial-off-the-shelf smartwatches and smartphones. The study proposed a more comfortable, portable, user-friendly ASLR system that offers accessibility anytime, anywhere. The detection ratio and reliability averages 99.2% and 99.5%, respectively. The proposed system currently can recognize 26 alphabet signs and 103 common; nonetheless, it is still far from being used in everyday situations [30].

2.1.1.10. Interpretation and Translation of American Sign Language for Hearing Impaired Individuals Using Image Processing

This study mainly focuses on the development of software that can convert

American sign language to communicative English language and vice-versa. This is accomplished with the use of image-processing. In this study, image-processing is done by using MATLAB, software by MathWorks [16].

2.1.1.11. i-Sign: Sign Language Learning Application Via Gamification

Figure 7. Quiz Section of the Application [39]

The study focuses on eradicating the difficulties faced by students with hearing disabilities through I-Sign, a learning application that features a simple gamification of learning materials that assists students to better understand and learn the alphabet and basic words of sign language. The model used for the instructional design methodology for the design was the ADDIE, or Analysis, Design, Development, Implementation, and Evaluation. The application was incorporated with two modes, learning and quiz mode where students are able to test what they have learned. Results have shown that in terms of effectiveness, and efficiency, the application was able to meet the expectations of 91.89% of the respondents who suffer with hearing loss [39].

2.1.1.12. Sign language detection using convolutional neural network for teaching and learning application

The objective of the study is to develop a method for detecting sign language which will be used for teaching and learning activities. A convolutional neural network was utilized to determine sign language with hand gestures. However, the proposed method still has limitations which include inaccurate recognition with gestures with similar shapes. Additionally, the system heavily depends on computational power indicating the need for high-end hardware for better optimization and performance [40].

2.1.1.13. ML Based Sign Language Recognition System

The study proposes a system of an automated Sign Language Recognition (SLR) system that utilizes a convex hull for feature extraction and KNN for classification. The process consists of pre-processing, contour-based segmentation, feature extraction, and classification [41].

2.1.1.14. SignQuiz: A Quiz Based Tool for Learning Fingerspelled Signs in Indian Sign Language Using ASLR

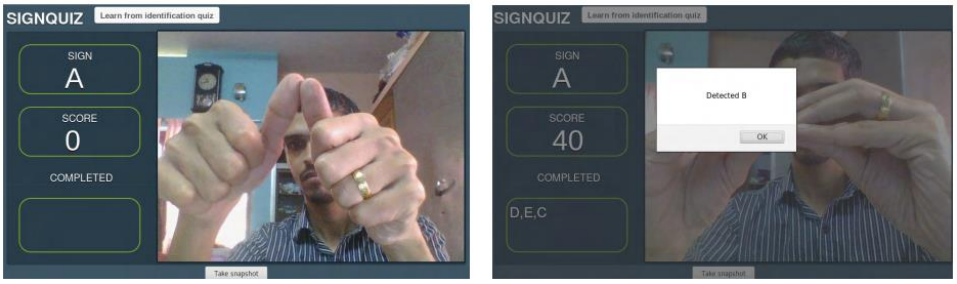


Figure 8. SignQuiz Application [42]

The study proposed is named SignQuiz, which is said to be a cost-effective web-based fingerspelled sign learning application for the Indian Sign Language (ISL) using a technique for an automatic sign language recognition. This teaches people signs without the use of external help. The study uses a Deep Neural Network (DNN) in the web-based application to learn sign language. The

application can be applied to both the deaf and non-deaf community which they can easily use [42].

2.1.2. Local Related Studies

2.1.2.1. Jetson Nano-Based Two-Way Communication System with Filipino Sign Language Recognition Using LSTM Deep Learning Model for Able and Deaf-Mute Persons

In this research, a two-way communication device was developed utilizing Jetson Nano. It ensures accuracy of 93.33% translation of Filipino Sign Language (FSL) into both text and speech, as well as the conversion of spoken input into text. The study focused on ten dynamic FSL gestures. The device leveraged an LSTM Deep Learning Model for recognizing FSL gestures, subsequently transforming them into speech using the Google Text-to-Speech (gTTS) API [43].

2.1.2.2. Senyas: A 3D Animated Filipino Sign Language Interpreter Using Speech Recognition

Figure 9. Implementation of LSTM in Senyas [21]

The purpose of the study is to aid the communication gap between people

who can hear and people who are deaf, a web application is created which interprets spoken and written inputs to Filipino Sign Language (FSL) that is executed by the 3D avatar with the use of speech recognition with sentiment analysis and a Human Pose Detection and Tracking which is used by a software. An LSTM model is used to create the proposed system and achieved an accuracy of 88.86% which is said to have a successful result in the classification of the translated sign language's sentiment which is either positive or negative. The LSTM model and the tracking software made gave the application permission to display the virtual interpreter that has a corresponding mood while signing, this gives the users both an interactive and convenient platform to build an inclusive conversation [21].

2.1.2.3. Sign to Speech Convolutional Neural Network-Based Filipino Sign Language Hand Gesture Recognition System

The proposed study developed a real-time Filipino Sign Language hand gesture recognition system that is based on a Convolutional Neural Network model. The researchers manually gathered the datasets for the system consisting of 237 video clips that have 20 different gestures. The dataset provided went through data cleaning and augmentation with the use of image pre-processing techniques. For the training of the Filipino Sign Language model, an inflated 3D convolutional neural network was used. Retraining the pre-trained model that has top layers, and all layers is considered for the experiments. The FSL recognition application is developed and assessed the usability by the target market with the use of Rapid Application Development Model [44].

2.1.2.4. TEXT2FSL: A Filipino Sign Language Phrase Translator Tool for Deaf and Mute

Figure 10. Video Screenshot Used in the Translator Tool [45]

The study aims to develop a Computer-Assisted Instruction (CAI), specifically, an e-Filipino Sign Language (FSL), and determine how it is capable of aiding deaf and mute students through its utilization in daily learning materials and assessments. The study also focuses on the creation of E-learning materials in mobile applications through videos and pictures as a method of learning FSL. Additionally, the application features a CAI program with multiple learning approaches for the students to try which includes games, quizzes, flashcards, and even tutorials. To develop the mobile application, Android Studio and Bootstrap were used together with Xamarin software on the application architecture, while the database was made with SQLite and Microsoft SQL server. The results showed that through evaluation and feedback from the students and FSL experts, the mobile application and its features helped the students remember sign language. However, they have also emphasized the difficulty in understanding the instruction module since these were only made of texts and pictures and had to rely on the arrows for

the hand movements [45].

2.1.2.5. Sign-Language Recognition Through Gesture & Movement Analysis (SIGMA)

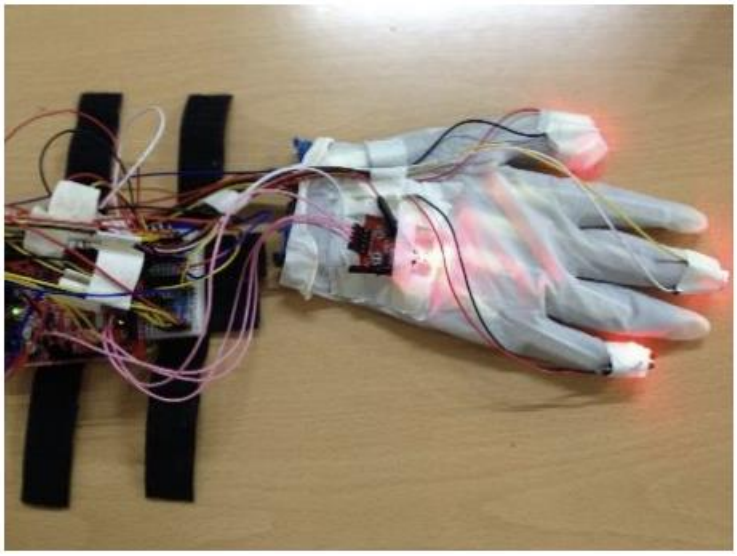


Figure 11. Data Gloves of SIGMA [46]

This study is a system that has a combination of a data glove and computer vision for the translation of FSL. In order to find the best gesture, fit from the Hidden Markov Model (HMM) a Viterbi algorithm is used [46]. For the validation of the accuracy of the recognition of the system, the system uses three training sets. A recognition accuracy of 71.8% was achieved in the first set which focuses on alphabets and numbers. For the second set, a recognition accuracy of 80.6% was achieved that consists of health-care-related words. Finally, a recognition accuracy of 80.55% was achieved that consists of the numbers zero to 9, the full vocabulary of 26 letters in the alphabet, and 30 common words that are used in health-care-related work in the third set [46].

2.2. Related Literatures

In this section of the related literature review, it provides a comprehensive and critical analysis of studies on the important subtopics covered by the thesis study. The review of these related literatures aims to deepen the understanding on these subtopics that will be covered in the study and to support the thesis study.

2.2.1. Deaf community in the Philippines

Based on The Philippine Deaf Resource Center there is an estimate of more than 1 million deaf people in the Philippines, constituting roughly 2% of the population. Since the early 1900s, deaf Filipinos have been actively building their community with a unique language known as Filipino Sign Language, rooted in American Sign Language. In recent times, they have been advocating for their rights, aiming to improve access to education, employment, and various essential services. However, significant barriers remain due to insufficient accessibility and a general lack of awareness about their special needs. Unfortunately, only a small percentage of deaf children receive adequate educational help, and many deaf individuals struggle to find employment. There is also a scarcity of sign language interpreters and other critical support services. As a result, deaf people in the Philippines typically experience experiences of social isolation and rejection [47].

2.2.2. Non-verbal Communication in Deaf Communities

Deaf people usually count on sign language, since it is a non-verbal language, for them to connect in social environments. This is based on the visual signs through the hands, face, eyes, mouth, and body. Gestures or symbols in sign languages are arranged in a linguistic way. Different forms of sign languages exist with over 300 varying types used by over 70 million deaf individuals in the world. Each sign language is distinct from each other and contains their own linguistic

rules. These sign languages are a central part of a deaf community's culture which makes up most of their communication medium [48].

2.2.3. Communication in Learning

Verbal and non-verbal communication play an integral role in teaching and learning activities between teachers and students. These two forms of communication are interconnected and essential in order to have effective learning. Examples of verbal communication are talking, discussing, joking, and other forms of spoken language. On the other hand, nonverbal communication includes eye contact, body language, touch, hearing, and other forms of nonverbal expression. Both verbal and nonverbal communication skills must continually be honed to create a high-quality learning environment. Communication in education is a unique form of communication that is designed to help teachers understand their students and vice-versa [49].

In the deaf community, there are various ways to receive information for the students who are deaf or hard of hearing. These ways are through an interpreter, through speechreading, through an assistive listening device (ALD), real-time captioning, or system of speech to print transcription. There are times that these students would use a combination of methods. An example of this would be using ALD and a system of speech to print transcription [50].

2.2.4. Integration of Deaf Individuals

In the past, individuals who require special assistance were frequently separated from the traditional school system. A shift toward integration happened in the past 30 years. Deaf individuals were one of the first groups to be integrated, as their normal appearance and compliance made them ideal candidates for this

approach. Concerning the integration of deaf individuals, most research focuses on systematic changes rather than individual progress [51].

Deaf individuals face many challenges when integrated in a coeducational setting with the hearing community. One of the biggest challenges they face is not their hearing impairment itself, but rather the lack of understanding and support from the hearing academic community [52]. Inclusive learning environments are to be done more for all students, including the deaf students.

2.2.5. Assistive Technologies

Assistive technology plays a vital role in promoting inclusivity and active participation, particularly for people with disabilities, the elderly, and those suffering from non-communicable diseases. The fundamental objective of assistive tools is to maintain or improve an individual's capabilities and self-sufficiency, therefore enhancing their overall well-being. These tools enable people to live healthy, fruitful, autonomous, and dignified lives by allowing them to pursue educational, professional, and civic goals. According to the World Health Organization (WHO), 2.5 billion people today require at least one assistive product, such as wheelchairs, hearing aids, or communication and cognition software. This amount is expected to approach 3.5 billion by 2050 [53].

Assistive technology is also defined under the Individuals with Disabilities Education Act (IDEA) of 2004 as any commercially available or modified item, equipment, or product system used to enhance, preserve, or advance the functional abilities of children with disabilities. When students with disabilities are provided with the appropriate assistive technology tool, they can better access curriculum, learning, and opportunities for success both inside and outside of the classroom

[54].

2.2.6. Computer-Assisted Learning

Innovations have rapidly affected educational systems with the implementation of technological advancements in learning materials and approaches to teaching students. Due to this reason, the development of ICT and its influence on education is considered a significant factor in affecting how students learn at schools. Furthermore, it has been shown that the application of ICT in learning can improve the quality of education, increase learning efficiency, and allow for more active learning for the students [55]. Blended learning models with the use of technological supplements such as computer applications, mobile-based learning materials, and more were seen to effectively improve teaching effects [56]. Teachers also believe that innovations in teaching materials with technology can be used as learning innovations.

2.2.7. Filipino Sign Language Laws in the Philippines

Republic Act No. 11106 is a law passed in 2018 which officially recognizes Filipino Sign Language as the national sign language of the deaf community which shall be used in all transactions involving the deaf. This law also recognizes its mandatory use in schools, broadcast media, and workplaces. Its implementation shall also adhere to the previous laws enacted which are Early Years Act (Republic Act No. 10410) and the Enhanced Basic Education Act (Republic Act No. 10533) that recognizes the Filipino Sign Language as a medium of instruction and curriculum for education from early childhood up to the secondary level. Furthermore, this includes the addition of training and evaluation programs by local governments to improve training of the educators and effectively evaluate their

methods in teaching [57].

2.2.8. Computer-Assisted Instruction in SPED schools in the Philippines

Computer-Assisted Instruction (CAI) were used among elementary students in the Philippines wherein its performance and effectivity in teaching was tested. The use of Computer-Assisted Instruction helps SPED students in acquiring knowledge such that its interactive and intuitive features influence the grasp of the students on the learning materials. Efficient CAI approaches should have easy-to-use functions with fast response times to minimize impatience and boredom when used by the users for studying. Its benefits to the SPED students include increase of motivation and retention, decrease in negative behaviors, and better effectiveness in learning [58].

2.3. Synthesis

Filipino sign language is now officially recognized as the national sign language of the deaf community in the Philippines. Due to the law requiring its use for situations that involve the deaf, it is necessary for effective communication to occur among disabled and non-disabled parties. For this to occur, education on proper usage of FSL must be conducted as early as the childhood phase of the disabled individuals. The law has also mandated teaching of FSL on Kindergarten and secondary level students with materials and medium of instruction on the sign language. To effectively teach FSL to younger students, the researchers of this study have found the idea of utilizing a Computer-Assisted Learning method implemented as a supplementary tool in teaching Filipino Sign Language.

Most studies conducted within the Philippines focused on the recognition and translation of FSL into either text, or speech. Additionally, learning applications in the

context of FSL are also limited in terms of functionality. Supplementary multimedia mediums such as images and videos were added to these applications to aid learning FSL. However, these do not actively recognize the efforts of the user whether they are correctly replicating the sign language shown in the learning materials. Due to this limitation, this research will focus on implementing a recognition model into teaching students FSL with feedback based on the input of the users.

The existing foreign studies found have also shown features that may hinder the full potential of learning sign language. Some were invasive such that they needed the use of gloves for recognition of sign language to occur. The use of invasive features may hinder movement or can be considered as an obstruction when used limiting what the user can do during operation. Other methods of sign language recognition are also costly with requirements of special types of cameras or a computer GPU for high computational power which restricts accessibility for the users especially SPED teachers and educators.

Another limitation shown on related studies comes on the recognition of sign language and the scope of what it can detect. Most studies are restricted to static signs which puts a cap on what students can learn, or the system can translate. This study aims to solve this limitation by developing a system capable of recognizing the user's dynamic sign language.

In conclusion, the proposed study will develop a computer-assisted tutoring application for Filipino Sign Language which addresses the past limitations exhibited by the related studies presented previously. These include a non-invasive approach and the capability of recognizing both static, and dynamic sign language for teaching deaf students with feedback to successfully achieve desired results.

2.4. Research Gap

This particular section of the chapter shows how the related studies are similar or different from the proposed system. Each study is briefly explained with its significant details and findings explained with it. This is presented in a tabular form as shown below:

Table 1. Research Gap

STUDY	REMARKS	GAP
FOREIGN STUDIES		
Assisted sign language tutoring system using a bioinspired virtual controlled robotic arm [19]	• A virtual robotic arm was developed to mimic the Mexican Sign Language • 21 one-handed static gestures from the Mexican Sign Language can be replicated by the robotic arm • Solidworks, Simulink, and PID were used to create the automation of the robotic arm	• Only Mexican Sign Language can be taught • Limited to static gestures which limits what it can teach • No recognition of sign language from the user
An intelligent system to help deaf students learn Arabic	• The use of GTTS library and speech recognition was	• Limited to Arabic Sign Language • Cannot recognize

Sign Language [20]	effective in translating Arabic Sign Language • Adding of sign language gestures to the database are not difficult	sign language performed by the user • Limited to images and videos of the sign language • Does not provide feedback
An Intelligent Computer-Based System for Sign Language Tutoring [28]	• A data glove was used to track the hand landmarks of the user • The computer application processes the movement signals in real-time through a Pattern Machine algorithm • The accuracy in recognizing the gestures from the gloves is 93.8% with 18% of the signs considered as difficult to replicate due to specific thumb or finger, and wrist bending angles	• Limited to recognizing Arabic Sign Language • Need the use of gloves • Can not recognize most of the provided gestures by the teachers • Complex sign language gestures may not be recognized immediately due to the hand's positioning • Does not provide feedback
D-Talk: Sign	• Machine learning	• Lighting conditions

Language Recognition System for People with Disability Using Machine Learning and Image Processing [15]	and image processing were utilized to interpret sign language for deaf and dumb individuals • Scale Invariant Feature Transform algorithm was utilized in extracting information during image processing	affect the results; cannot effectively recognize sign language with poor lighting • Not a tutoring application
A Smart Glove with Integrated Triboelectric Nanogenerator for Self-powered Gesture Recognition and Language Experience [29]	• Through Triboelectric Nanogenerator integrated into smart gloves, sign language can be recognized and translated into text for communication	• Needs the use of smart gloves; invasive for users • Limited to static sign language gestures • Not a tutoring application
Deep Learning-Based Sign Language Recognition System for Static Signs [12]	• A Convolutional Neural Network (CNN) deep learning approach was interfaced for the recognition of Indian Sign Language. • 35,000 sign images	• Not a tutoring application; limited to only recognizing sign language • Limited to static sign language gestures

	of 100 static signs were collected from different users for the study	
Providing Feedback in Ukrainian Sign Language Tutoring Software [37]	• This approach allows for user participation during sign learning and verification by utilizing body posture recognition algorithms. The software gives the user feedback by capturing their gestures using a web camera, increasing the effectiveness of the training process. • This approach makes use of a single web camera rather than depth sensors. Background modeling, image segmentation, and machine learning approaches are used	• Limited to Ukrainian Sign Language only • Feedback is limited to right and wrong; does not highlight which part is wrong

	in the real-time reconstruction of human posture from a web camera. • A 91.7% recognition success rate was attained for sign recognition utilizing a test set of 85 signs.	
Natural User Interface Based American Sign Language Tutoring Program [38]	• The research designed a system that has an interactive online system for ASL learning using hand-tracking peripherals.	• Sign languages are only available for American Sign Language
SignSpeaker: A Real-time, High-Precision SmartWatch-based Sign Language Translator [30]	• The first smartwatch-based end-to-end sentence-level ASLR system was proposed in this work. • Deployed on a smartwatch along with a smartphone; the smartwatch	• Limited to American Sign Language Recognition • The system is based on a wearable device with inbuilt motion sensors • Invasive

	collects the sign signals and the smartphone outputs translation through an inbuilt loudspeaker.	
Interpretation and Translation of American Sign Language for Hearing Impaired Individuals Using Image Processing [16]	• The study mainly focuses on the development of software that can convert American Sign Language to communicative english language and vice versa. • This is accomplished with the use of image-processing. In this study, image-processing is done by using MATLAB, software by MathWorks.	• Limited to static sign language • Limited to American Sign Language • Translates sign language to speech and vice-versa
i-Sign: Sign Language Learning Application Via Gamification [39]	• Conducts an interview and survey with the teachers and students of SK	• Tested children for gathering the effectiveness of the application • Is limited to

	Pendidikan Khas Melaka • From the interview and survey, it identified the problems in the learning spectrum of the children that have hearing disabilities • i-Sign was developed as a learning application which incorporates simple gamification • A usability test was conducted to 8 children that have hearing loss and found that 91.89% of the respondent found it meets the expectation of the application	alphabets and only basic words
Sign language detection using convolutional neural network for teaching and learning application [40]	• The purpose of the research is to design a sign language detection scheme for teaching and activity learning	• Does not indicate what sign language was used • Sometimes have a hard time in identifying hand gestures with

	• Images from the hand gestures of the teacher or presenter is taken using a web camera for the anticipation and displaying the image's name of the system • The scheme detects hand movements that can convert it to meaningful information	identical shape
ML Based Sign Language Recognition System [41]	• The study proposes a system of an automated Sign Language Recognition (SLR) system that utilizes a convex hull for feature extraction and KNN for classification. • The process consists of pre-processing, contour-based segmentation, feature extraction, and classification.	• Accuracy of sign language detection was only 65% • Detection of gestures were less when hands were moved at a fast pace • Only static signs

SignQuiz: A Quiz Based Tool for Learning Fingerspelled Signs in Indian Sign Language Using ASLR [42]	• The research developed an application named SignQuiz that is a cost-effective web-based fingerspelled learning application.	• Applies only for Indian Sign Language
	LOCAL STUDIES	
Jetson Nano-Based Two-Way Communication System with Filipino Sign Language Recognition Using LSTM Deep Learning Model for Able and Deaf-Mute Persons [43]	• Developed a two-way communication device using Jetson Nano. • Ensures an accuracy of 93.33% in the translation of Filipino Sign Language (FSL) to text and speech • The study focuses on ten dynamic FSL gestures	• Limited to basic sign language gestures • Faces inconsistencies in dynamic gestures • Requires additional components
Senyas: A 3D Animated Filipino Sign Language Interpreter Using Speech Recognition [21]	• Aims to create a web application which interprets both spoken and written inputs into FSL	• Limited to only speech recognition and sentiment analysis • Only test the accuracy of sign

	• A 3D avatar is used to execute the speech recognition • Uses a LSTM model in the creation of the system • Achieved an accuracy of 86.86% using the proposed model • LSTM model and tracking software allows the application to display the virtual interpreter	language translation
Sign to Speech Convolutional Neural Network-Based Filipino Sign Language Hand Gesture Recognition System [44]	• The research developed a Convolutional Neural Network based real-time Filipino Sign Language hand gesture recognition system • Gathered a dataset consisting of 237 video clips that have 20 different gestures	• Limited to only sign language recognition • Not a tutoring application

TEXT2FSL: A Filipino Sign Language Phrase Translator Tool for Deaf and Mute [45]	• A Computer-Assisted Instruction (CAI) that provides text and questions, or even problems to learners. • Focuses on how e-Filipino Sign Language aids deaf and mute students using learning materials and assessments. • A positive result was derived which helped and aid students or users and was easily familiarized and remembered the sign language	• Uses a mobile application to teach students • Uses only videos of sign language translation • Does not use deep learning models and image processing
Sign-Language Recognition Through Gesture & Movement Analysis (SIGMA) [46]	• The research uses a data glove prototype with computer vision for FSL translation for medical purposes. • A Viterbi algorithm is used in the system for finding the best gesture fit	• Is invasive • Limited for medical purposes only • Does not integrate a tutoring system

	from the HMM modeling.	

Table 1 provides the gaps of the foreign and local studies as compared to the proposed design. The table explicitly shows each feature presented for each study and what it lacks from the other studies. The specific gaps are divided into eight groups which are the static and dynamic words of the SL recognition system, the non-invasiveness of the system, the use of an application, its computer-assisted learning feature, whether it uses FSL or not, and whether there is feedback provided in the system. Table 2 as seen below describes the complete checklist of the research gaps on what the studies have. It also shows that the proposed study will contain all the stated categories in the checklist. Furthermore, Table 2 further proves the need for the proposed study since it accomplishes all the features presented from the related studies.

Table 2. Research Gap Checklist

Related Studies	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	Proposed Study
Static	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓
Dynamic								✓						✓	✓		✓		✓	✓
Non-Invasive	✓	✓					✓	✓		✓	✓	✓		✓		✓	✓	✓		✓
Application							✓	✓	✓		✓	✓		✓		✓		✓		✓
Computer Assisted Learning	✓	✓	✓				✓	✓			✓	✓								✓
FSL															✓	✓	✓	✓	✓	✓
SL Recognition			✓	✓	✓	✓		✓	✓	✓		✓	✓	✓	✓				✓	✓
Feedback																				✓

Related Studies

A. Assisted sign language tutoring system using a bioinspired virtual controlled

robotic arm

B. An intelligent system to help deaf students learn Arabic Sign Language

C. An Intelligent Computer-Based System for Sign Language Tutoring

D. D-Talk: Sign Language Recognition System for People with Disability Using Machine Learning and Image Processing

E. A Smart Glove with Integrated Triboelectric Nanogenerator for Self-powered Gesture Recognition and Language Experience

F. Deep Learning-Based Sign Language Recognition System for Static Signs

G. Providing Feedback in Ukrainian Sign Language Tutoring Software

H. Natural User Interface Based American Sign Language Tutoring Program

I. SignSpeaker: A Real-time, High-Precision SmartWatch-based Sign Language Translator

J. Interpretation and Translation of American Sign Language for Hearing Impaired Individuals Using Image Processing

K. i-Sign: Sign Language Learning Application Via Gamification

L. Sign language detection using convolutional neural network for teaching and learning application

M. ML Based Sign Language Recognition System

N. SignQuiz: A Quiz Based Tool for Learning Fingerspelled Signs in Indian Sign Language Using ASLR

O. Jetson Nano-Based Two-Way Communication System with Filipino Sign Language Recognition Using LSTM Deep Learning Model for Able and Deaf-Mute Persons

P. Senyas: A 3D Animated Filipino Sign Language Interpreter Using Speech

Recognition

Q. Sign to Speech Convolutional Neural Network-Based Filipino Sign Language Hand Gesture Recognition System

R. TEXT2FSL: A Filipino Sign Language Phrase Translator Tool for Deaf and Mute

S. Sign-Language Recognition Through Gesture & Movement Analysis (SIGMA)

Chapter 3

METHODOLOGY

The primary objective of this chapter focuses on the procedures applied to conduct the necessary process in data collection and its approach. Moreover, this includes a discussion regarding the location of the test, sampling technique, and the hardware and software of the research instruments used. Lastly, this chapter will also describe the statistical treatment of the collected data.

3.1. Theoretical Framework

Artificial Intelligence (AI), Machine Learning, and Deep Learning are connected with each other in learning algorithms for a much easier life for people. These may sometimes be interchangeable when experts use them. For further simplification, deep learning is a part of machine learning. Additionally, machine learning is a part of artificial intelligence. For proper simulation of the complicated behavior of humans, an artificial intelligence algorithm is used to give the computer permission to simulate it. Labeled data sets are a requirement of machine learning to be properly trained. Meanwhile, due to the spanning networks that can identify patterns, trained data sets are not required for deep learning. For the proper understanding of the machine to the features of the object, users set labeled data set parameters [59].

Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), and Generative Adversarial Network (GAN) are the

most popular algorithms used in neural networks that are under deep learning. The use of LSTM is suggested in the proponents of the proposed research since image processing and object detection is mostly used in this network [60].

The component of the image's matrix is utilized by the Long Short-Term Memory for data extraction. The LSTM deep learning model is an improved version of the RNN since it has a single hidden state that passes as time progresses, which makes the long-term dependencies hard to learn for the network. The LSTM model can address the problem that RNN, which introduces a memory cell in the network, this is a container that is able to hold information for an extended period. This model has the upper hand when it comes to prediction tasks and capturing long-term dependencies. Since LSTM is capable of learning long-term dependencies in the sequential data, which makes LSTM well suited for language translation, speech recognition, and time series forecasting [61].

3.2. Block Diagram

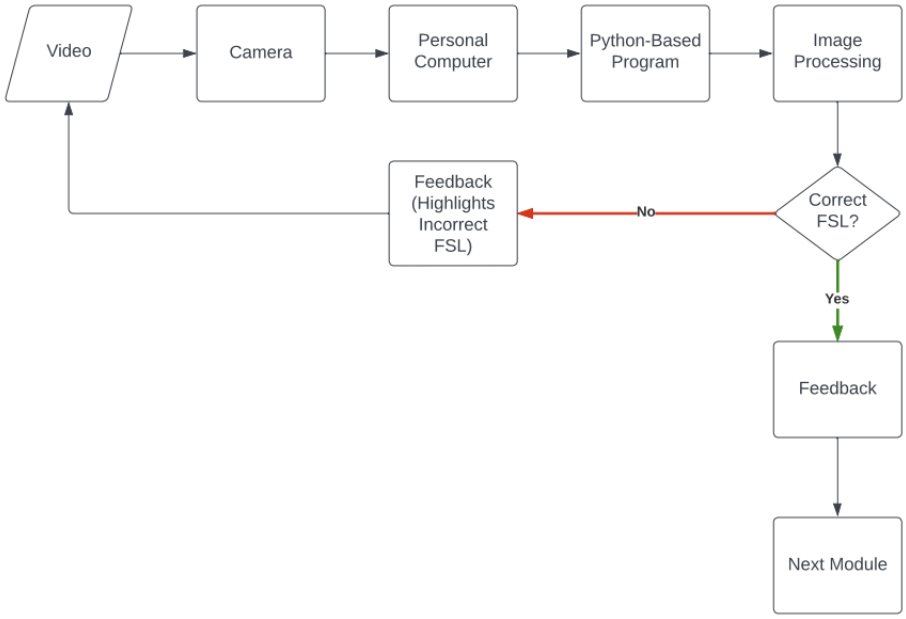


Figure 12. Block Diagram of the Prototype

The figure above shows the proposed set-up of the prototype and the modules that will be employed during training and testing. The process begins as the system is fed with data through a real-time video stream of the user's sign language captured by the camera of the prototype. The proposed study does not have any required specific camera for this program to function. Afterwards, the video will be collected by the user's personal computer in which the Python-based program developed by the proponents will be interfaced with image processing and deep learning for recognition of the user's sign language. Once recognized, the system can determine if the sign language done by the user is similar to the ones currently presented in the lesson from the program. If correct, the user may proceed to the next module. If it does not meet the requirements of the sign language, feedback will be shown on the screen on which particular part the user has made a mistake and shall repeat the sign language.

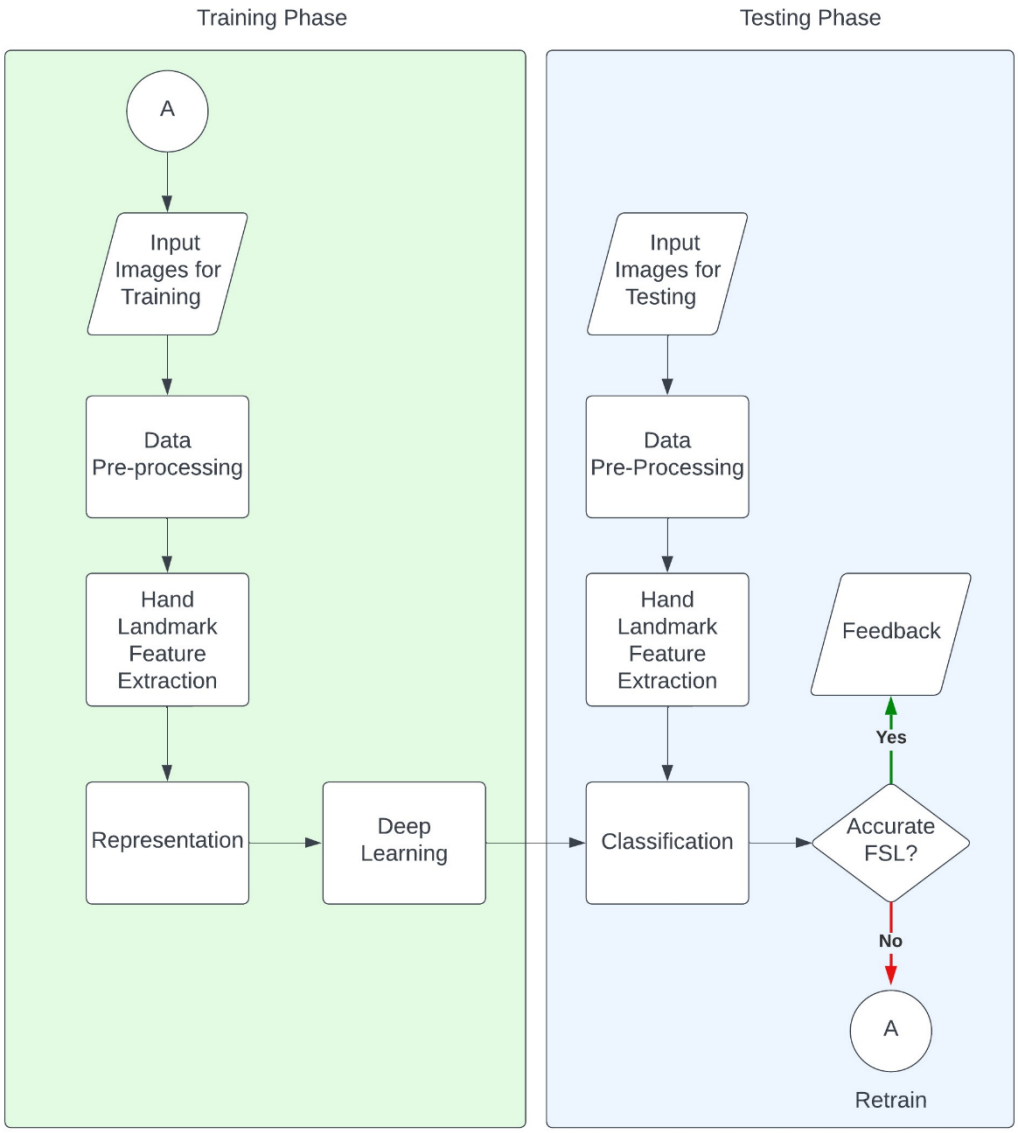


Figure 13. Block Diagram of the Deep Learning Algorithm

The block diagram shown exhibits the entire process of the deep learning algorithm interfaced into the system and how the data was trained for the prototype. As seen from the figure, the process is divided into two phases, namely, the training and testing phase. In order to train the model, the training phase must first occur where images shall serve as an input for training the model. These images will go through the algorithm and processed for its features to be extracted, specifically, the feature landmarks. Once done, the network

learns to classify or predict the target variable based on the patterns and relationships in the features. Once the system has been trained, the model should undergo a testing phase to ensure that the input of data accurately represents what is needed by the system. After evaluation, if the system determines that the trained model accurately depicts the FSL input of the user, it shall proceed in providing feedback. If the trained model does not achieve the required results, it must undergo the training phase once again.

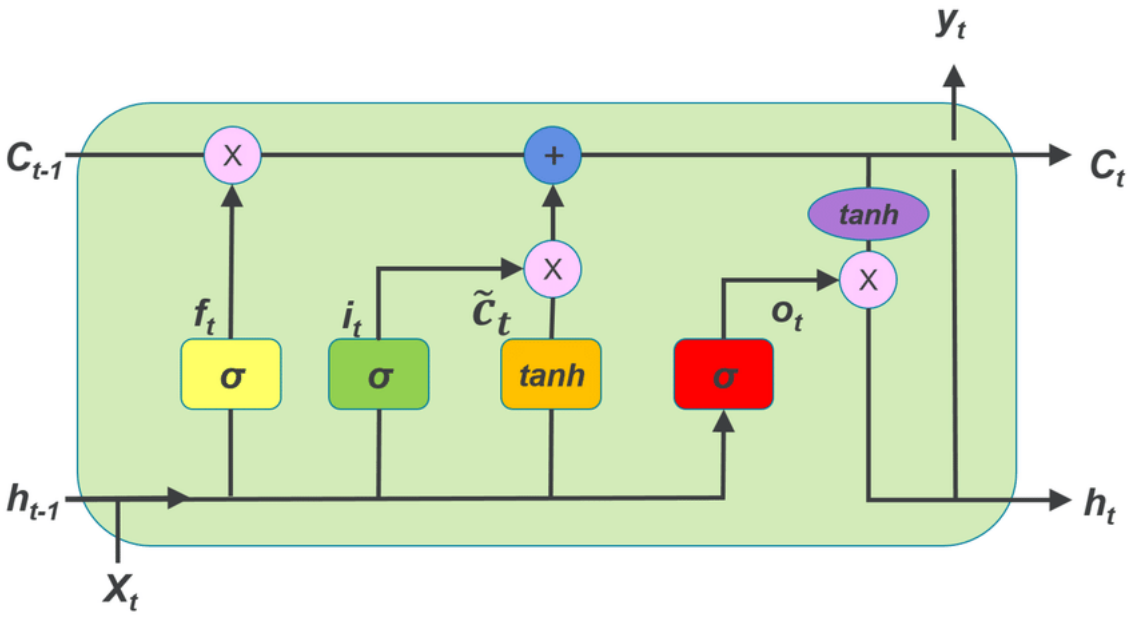


Figure 14. Block Diagram of the Long Short-Term Memory Neural Network [62]

The study will be interfaced with the Long Short-Term Memory which will be responsible for understanding the sequences of gestures, which are temporal in nature, that have been extracted into data as landmarks. Since dynamic sign languages are sequential, the its order and context are crucial in terms of utilizing long-term dependencies. For the input, a live video stream will be captured by the system's camera where it will be processed and extracted as keypoints. From each frame, these keypoints are then processed as coordinated of landmarks of hands, pose, and face. These are arranged into sequences representing each sign and are fed into the LSTM model classifying the sequences into

different sign language gestures. Figure 14 shows a Long Short-Term Memory Neural Network cell. These memory cells have the ability to retain information over long sequences, each containing an input gate, forget gate, and an output gate which regulates the flow of information in and out of the memory cell.

3.3. Deep Learning Model

There are multiple deep learning techniques considered for the proposed sign language recognition application. Various studies and works of literature related to sign language recognition were examined to determine the most capable and effective deep learning model compatible with the system. Most studies employed either Convolutional Neural Network, Long- Short-Term Memory, or K-Nearest Neural Network in their sign language recognition system. In order to decide which deep learning model should be used, a comparison table was made to identify each of the advantages and disadvantages each model offers. The table is shown below:

Table 3. Comparison of Deep Learning Models

	Advantages	**Disadvantages**
Convolutional Neural Network (CNN) [63]	Excellent for image-based tasks Hierarchical feature learning Translation-invariance Automatically learns hierarchical features	May require large amounts of labeled data Training can be computationally expensive May not perform well with limited data
Long Short-Term	Good at capturing temporal	May struggle with long-

Memory (LSTM) [64]	dependencies Suitable for sequential data Handles variable-length sequences well	range dependencies Training can be computationally expensive May not capture fine-grained spatial features
K-Nearest Neural Network [65]	Simple and easy to implement No training phase, instance-based learning Works well with small datasets No assumptions about data distribution	Computationally expensive during inference Sensitive to irrelevant or redundant features Not suitable for high-dimensional data Prediction time grows with the size of the dataset

Most deep learning computations used for analyzing sequential data utilize Long Short-Term Memory due to its ability to retain information with its memory cells [66]. LSTM models are also capable of capturing temporal dependencies, allowing to predict events or outcomes based on historical data [67]. Since sign language involved a sequence of gestures extracted as keypoints, capturing temporal context is essential for accurate interpretation. LSTM can also learn patterns across time steps, making it ideal for sign language recognition and interpreting complex signs [68]. Lastly, other deep learning models such as CNN and KNN have their own merits in certain contexts, but they were not chosen since they are not as optimal for dynamic sign language recognition compared to

LSTM.

3.4. Deep Learning Training Model

Sign language would be recognized through image processing with a deep learning algorithm to detect the corresponding meaning of each gesture. Specifically, Long Short-Term Memory was utilized since it is one of the most popular and effective methods in vision-based algorithms and object detection [69]. Additionally, it offers a competitive advantage against other models due to its capability to automatically detect significant temporal dependencies on sequential data.

The dataset that will be used for the training of the model through deep learning will be acquired through the system's camera. There will be a total of 50 signs that will be included in the system's database for FSL recognition. A recommended of 33 or more data will be used for training each sign until the desired recognition is achieved by the model [70]. For this reason, there will be 1650 or more total data sets that will be used for training and testing. Specifically, 70% will be utilized for training while the remaining 30% is for testing the trained model.

The proposed study will only focus 50 essential sign language gestures which includes 28 letters from the Filipino Alphabet. These were considered based on the Department of Education's Kindergarten Basic Education Curriculum [71]. To be exact, this will only cover the curriculum's competencies for "Use the proper expression in introducing oneself" and "Use polite greetings and courteous expressions in appropriate situations" with codes "LLKVPD-Ia-13" and "LLKOL-Ia-1", respectively. These, along with Alphabet Knowledge (AK), would fall under "Oral Language" under "Section G. Language, Literacy, and Communication". Furthermore, it will also cover "Vocabulary Development (V)" with codes "LLKV-00-1", "LLKV-00-2". The study is only limited to

50 signs that will be chosen based on the amount of its usage under the aforementioned core competencies of the curriculum.

Table 4. List of FSL Words and Phrases to be Used

Use the proper expression in introducing oneself	Use polite greetings and courteous expressions in appropriate situations	Alphabet Knowledge		Vocabulary Development (V)
Ako si/ Ang pangalan ko ay	Magandang Umaga	A	Ñ	Bahay
		B	NG	
Ano ang pangalan mo?	Magandang Hapon	C	O	Pinto
Ilang taon ka na?	Magandang Gabi	D	P	Silid
		E	Q	
Sino	Kumusta ka	F	R	Sala
	Magandang araw	G	S	Kusina
	Paalam	H	T	Guro
	Ingat ka	I	U	Kailan
	Maraming Salamat	J	V	Ube
	Pasensya na	K	W	Dilaw
		L	X	
		M	Y	
		N	Z	

3.5. Image Processing Techniques

The image processing approach for the image recognition in the input is a necessary process that enables the system to extract the necessary features from the live video stream. The proposed prototype will be interfaced with a Scale Invariant Feature Transform (SIFT) image processing technique since it offers numerous advantages. This approach allows the system to extract features vectors which define the local patches of the image input. SIFT is considered to be more accurate than other image processing techniques and can describe

the key features of an object within the image better than other approaches [15].

The advantage and disadvantages of the image processing techniques considered for this study is indicated in the following table:

Table 5. Comparison of Image Processing Techniques

Image Processing Technique	Advantages	Disadvantages
Scale Invariant Feature Transform (SIFT) [15]	• Applies a key point descriptor • Distinctive keypoints • Good performance with varying lighting	• Computationally intensive • May not work well with non-textured images
Speed Up Robust Features (SURF) [71]	• Similar to SIFT, based on the determinant of the Hessian Matrix • Faster than SIFT • Works well with textured images	• Less distinctive than SIFT • Can be sensitive to noise
Binary Robust Invariant Scalable Key (BRISK) [71]	• Very fast feature extraction • Good for real-time applications	• Less distinctive than SIFT and SURF • May not work well with smooth images
Hue Saturation Value Color Model (HSV) [71]	• Simple and efficient for color-based feature extraction • Useful for skin tone detection	• Limited in capturing texture information • Not suitable for non-color images • Limited in handling complex lighting

3.6. System Flowchart

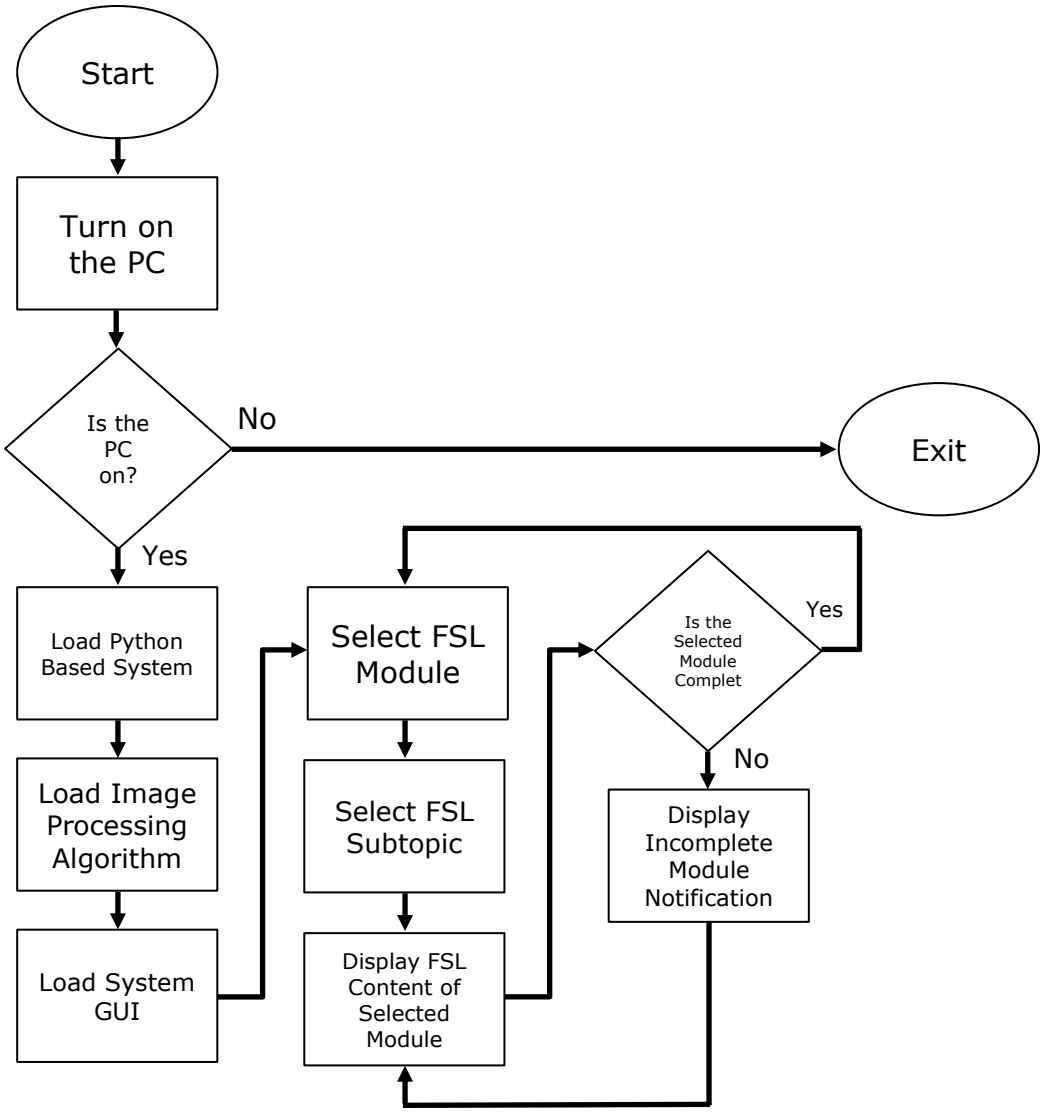


Figure 15. Flow Chart of the System

The flow chart is shown in Figure 5 which explains the flow of the system, and how the decision-making of the program functions. The system begins by turning on the PC that hosts the FSL tutoring application, and the initialization of the camera for data collection. Once the hardware system has been completed setting up, the Python-based system is then loaded up including the algorithms, GUI, and programs necessary in FSL recognition and tutoring. When the application has been initialized and opened, the users will be given the

option to choose the module and subtopic that they desire to learn. Afterwards, the GUI will show the lesson and the necessary content for the user to study. An assessment will be given after to ensure mastery of the topic, and if completed, will only be given the option to proceed to the succeeding modules.

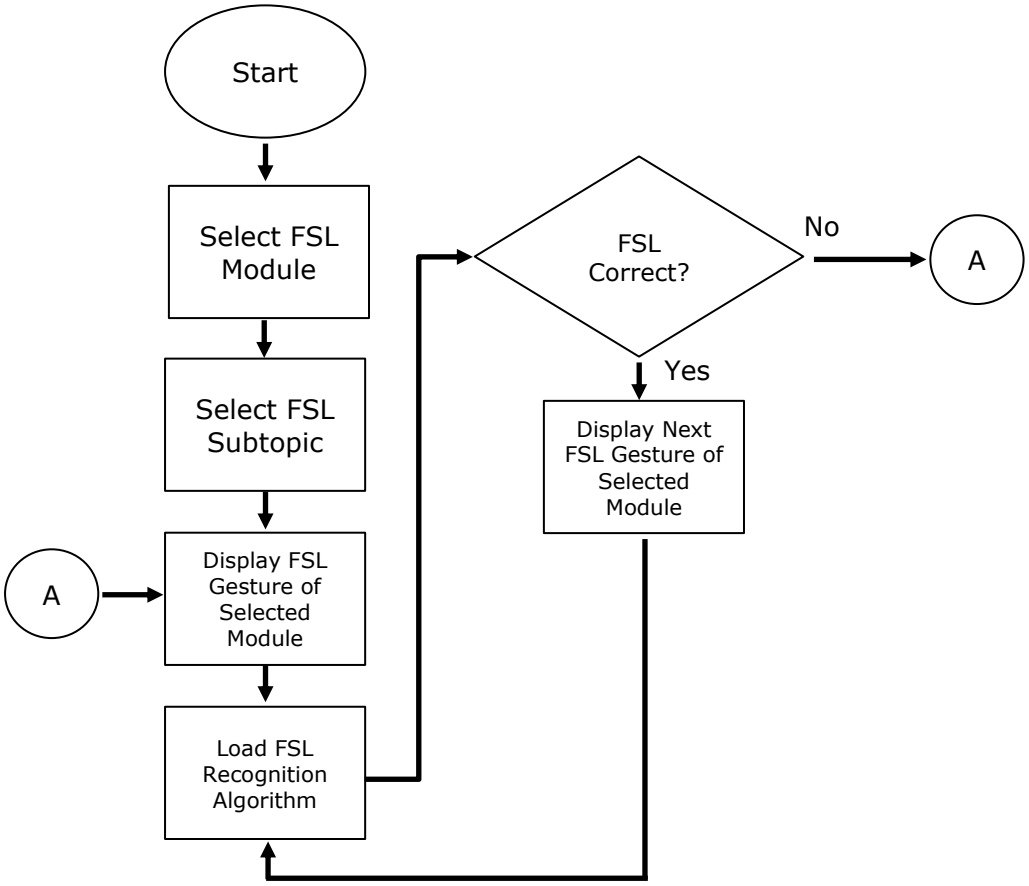


Figure 16. Flow Chart of the Tutor Module System

The flowchart shown in Figure 6 shows the flow of how the tutor system will work with the user's actions. The user will start by selecting a module and then, the subtopic under the module. The FSL recognition algorithm will be loaded after this, and the user will now be able to do the FSL gesture which will be captured by the camera. If the FSL gesture is correct, the user can now proceed to the next FSL gesture of the selected module to learn. If the FSL gesture is incorrect, the system will notify the user that the gesture

recognized was incorrect. The system will display once again the video of the correct way to gesture the FSL.

3.7. System Algorithm

The pseudo codes of the algorithm for the system to function include:

3.7.1. Algorithm for Sign Language Detection during Lessons

1. Load the trained model during the training phase for feature landmark detection.
2. Detect the feature landmarks from the video data.
3. If a gesture corresponds to the system's database, correct feedback is shown.
4. If not, incorrect feedback is shown with the mistakes highlighted.

3.7.2. Algorithm for Feedback Evaluation and Correction

1. Load the lesson chosen by the user and show the supplementary materials.
2. Load the FSL assessment phase to determine the correctness of the user's FSL.
3. Check the corresponding value of each gesture and check if it is similar to the ones presented in the lesson.
4. If not similar, highlight the different ones which will be classified as mistakes.

3.7.3. Algorithm for Application

1. Load the application.
2. Present the home screen which includes the last lessons of the user.
3. Display all modules at the lessons tab if chosen by the user.

4. Begin the lesson chosen by the user and start the sign language recognition.
5. Display the profile of the user if profile tab is chosen.

3.8. Test Population

A study about a tutoring system for Filipino Sign Language used a dataset of 50 total signs that are deemed essential for communication for deaf individuals [72]. However, for this particular study, the signs that will be included in the 50 sign language gestures will be based on the DepEd Kindergarten Basic Education Curriculum [71]. Specifically, it will prioritize “Section G. Language, Literacy, and Communication” which covers “Oral Language” with codes “LLKVPD-Ia-13" and “LLKOL-Ia-1". These course codes would cover the competencies for “Use the proper expression in introducing oneself” and “Use polite greetings and courteous expressions in appropriate situations”. 28 of the 50 signs in the database will cover the alphabet since there are twenty-eight letters in the Filipino alphabet which falls under the curriculum’s “Alphabet Knowledge”. Lastly, it will also “Vocabulary Development (V)” with codes “LLKV-00-1", “LLKV-00-2". These signs will be validated by three educators chosen through purposive sampling who teach Filipino Sign Language to students as a profession.

3.9. Hypothesis

Null Hypothesis (H_0): There is no significant difference in the effectiveness of learning Filipino Sign Language between the automatic FSL tutor with static and dynamic FSL recognition using Image Processing and Deep Learning and traditional FSL tutoring methods.

Alternative Hypothesis (H_1): The development of an automatic FSL tutor with static and dynamic FSL recognition using Image processing and Deep Learning is more effective

than traditional FSL tutoring methods in teaching Filipino Sign Language.

3.10. Treatment of Data and Statistical Analysis

For Specific Objective 4:

To effectively analyze the data within the matrix, the proponents will utilize various metrics including precision, recall, specificity, F1 score, Matthew's correlation coefficient, and reliability formulas. These metrics were used as it was applied in a study that also utilized a dataset for both testing and training a detection system based on a deep learning model. For the assessment of the system's performance, a confusion matrix will be employed. To determine the required number of trials for testing, the proponents will employ the Fundamental Formula of Gambling. These metrics will be used as it were also used on a similar study [73].

Definition:

- Positive (P) - FSL gesture is correct.
- Negative (N) - FSL gesture is incorrect.
- True Positive (TP) - FSL gesture is correct, and the system recognizes it as correct.
- True Negative (TN) - FSL gesture is incorrect, and the system recognizes it as incorrect.
- False Positive (FP) - FSL gesture is incorrect, and the system recognizes it as correct.
- False Negative (FN) - FSL gesture is correct, and the system recognizes it as incorrect.

3.10.1. Number of Trials

To compute for the number of trials needed for testing, the proponents will utilize the Fundamental Formula for Gambling which is given as follows:

$$\boldsymbol{N = \frac{\log(1 - DC)}{\log(1 - p)}}$$

Equation 1. Computation for No. of Trials

where,

N = Number of trials

DC = Degree of Certainty

p = event of probability

To determine the required number of trials for a 95% degree of certainty (DC) with an event probability (p) of 0.5, we use this value of p since there are only two potential outcomes: the correctness or incorrectness of the Filipino Sign Language gesture.

$$N = \frac{\log(1 - 0.95)}{\log(1 - 0.5)} = 4.3219809489 \cong 5$$

3.10.2. Accuracy

Accuracy measures the correctly predicted outcome of the application. It shows how often the sign language recognition shows a correct output [75]. Accuracy is given by the equation:

$$\boldsymbol{Accuracy = \left(\frac{TP + TN}{TP + FN + TN + FP}\right) \times 100\%}$$

Equation 2. Accuracy Formula

3.10.3. Precision

Precision measures if the application correctly predicts the positive class of

the table. The application shows how often the positive predictions are correct [75]. Precision percentage can be calculated by taking the total number of true positive values, dividing it by the sum of true positive values and false positive values, and expressing the result as a percentage.

$$\mathbf{Pr}\,\boldsymbol{e\,cision} = \left(\frac{\boldsymbol{TP}}{\boldsymbol{TP + FP}}\right) \times \mathbf{100}\%$$

Equation 3. Precision Formula

3.10.4. Recall

Recall measures how the application can correctly identify the true positives from the actual positive samples in the dataset [75]. Recall can be calculated by taking the proportion of true positive values divided by the sum of true positive values and false negative values, expressed as a percentage.

$$\boldsymbol{Recall} = \left(\frac{\boldsymbol{TP}}{\boldsymbol{TP + FN}}\right) \times \mathbf{100}\%$$

Equation 4. Recall Formula

3.10.5. Specificity

Specificity is the ratio of true negative values to the sum of true negative values and false positive values, expressed as a percentage. This shows how well the application will identify true negatives when used [76].

$$\boldsymbol{Specificity} = \left(\frac{\boldsymbol{TN}}{\boldsymbol{TN + FP}}\right) \times \mathbf{100}\%$$

Equation 5. Specificity Formula

3.10.6. F1 Score

F1 score evaluates the applications performance [77]. F1 score can be defined as twice the ratio of precision value multiplied by the recall value and the sum of precision value and the recall value.

$$F1 = 2 \times \left(\frac{precision \times recall}{precision + recall}\right)$$

Equation 6. F1 Score Formula

3.10.7. Matthew's Correlation Coefficient

An additional statistical metric is employed to provide a comprehensive assessment of the system's performance. Matthew's correlation coefficient (MCC) is considered a more reliable measure than F1 score for evaluating a machine learning system because it takes into account all four values of the confusion matrix (true positives, true negatives, false positives, and false negatives) and provides a balanced measure that is especially useful when there is a significant imbalance between the classes. The results can range from -1 to 1, with a high value near 1 indicating effective predictions for both classes. The classes used in this study are categorized as the "correct" and "incorrect" FSL gestures [23][24]. MCC can be calculated using the equation:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Equation 7. Matthew's Correlation Coefficient Formula

3.10.8. Reliability

The reliability formula calculates the operational stability of a system by dividing the number of successful outcomes by the total number of trials, with the result expressed as a percentage.

$$Reliability = \left(\frac{Number\ of\ Success}{Number\ of\ Trials}\right) \times 100\%$$

Equation 8. Reliability Formula

3.10.8.1. Reliability of Individual Components

The individual reliability of components for pretesting the monitoring device, sound alarm, and android alarm/notification is determined using the following formulas:

Definitions:

- NS = Number of Success
- NT = Number of Trials
- R_C = Reliability of camera
- R_{CP} = Reliability of computer
- R_A = Reliability of Application

3.10.8.1.1. Camera Reliability

The reliability of the camera can be determined by dividing the number of successful operations when used by the total number of trials, and then expressing this as a percentage.

$$\%\boldsymbol{R_c} = \left(\frac{\boldsymbol{NS_{camera}}}{\boldsymbol{NT_{camera}}}\right) \times \mathbf{100}$$

Equation 9. Camera Reliability formula

3.10.8.1.2. Computer Reliability

The computer's reliability can be calculated by taking the ratio of successful operations to the total number of trials performed on the computer, and then multiplying this by 100.

$$\%\boldsymbol{R_{CP}} = \left(\frac{\boldsymbol{NS_{computer}}}{\boldsymbol{NT_{computer}}}\right) \times \mathbf{100}$$

Equation 10. Computer Reliability formula

3.10.8.1.3. Application Reliability

The application's reliability can be determined by calculating the ratio of successful outcomes to the total number of trials, and then expressing this as a percentage.

$$\%R_A = \left(\frac{NS_{application}}{NT_{application}}\right) \times 100$$

Equation 11. Application Reliability formula

3.11. Proposed System

This section of the study discusses the ideal setup of the proposed project. It is divided into two parts, specifically, Hardware and Software System, which will discuss the components used in each part of the project.

Figure 17. Ideal setup of the user and the proposed prototype

3.11.1. Hardware System

The proposed setup of the hardware system is shown in figure 17. It consists of a personal computer, and a webcam which will be used by the educator as a learning tool in aiding the students in learning FSL.

3.11.1.1. Personal Computer

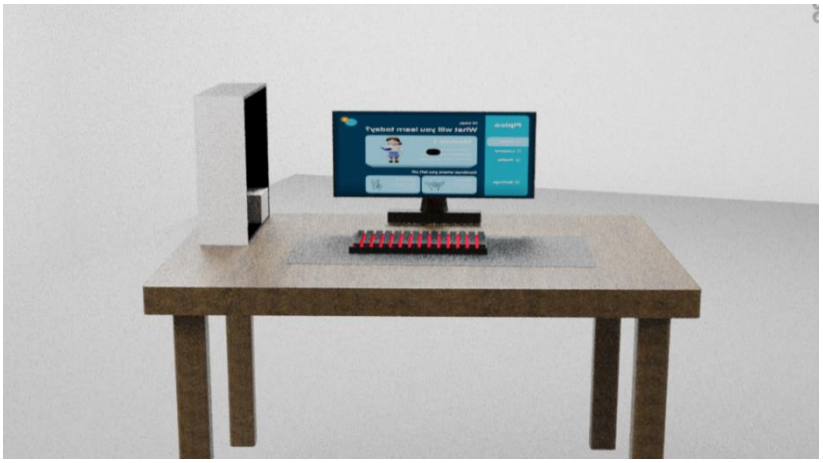


Figure 18. Personal Computer

The personal computer will be the ones necessary in handling, and processing of the data. It will also host the software, or program, of the proposed study where the users may access the module and take assessments for their FSL. There is no required specification for the personal computer, or laptop in the proposed study. The program will be able to run on most laptops which are accessible by most educators. No other equipment is needed aside from the camera.

3.11.1.2. Camera

Figure 19. Sample Camera

The camera needed in the system does not have any required specifications. Most laptops come with webcams that are already accessible for the user to utilize.

The camera will serve as the input of the system since it will capture the video in real-time of the user's sign language. This data will be transferred to the personal computer to be processed.

3.11.2. Software System

The following describes the software that will be used during the operation of the proposed system. This includes the Python-based program developed by the proponents and the features it includes.

3.11.2.1. Proposed Home Screen GUI of the FSL Tutoring System

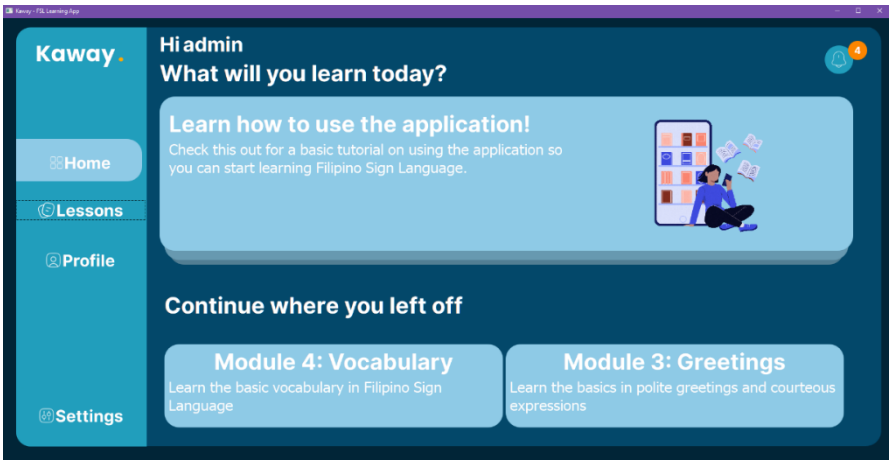


Figure 20. Home Screen GUI

Figure 20 illustrates the home screen of the application where it shows the modules that the user may learn during that session. On the left side, users may switch tabs to choose lessons or their profile page.

3.11.2.2. Proposed Module Screen GUI of the FSL Tutoring System

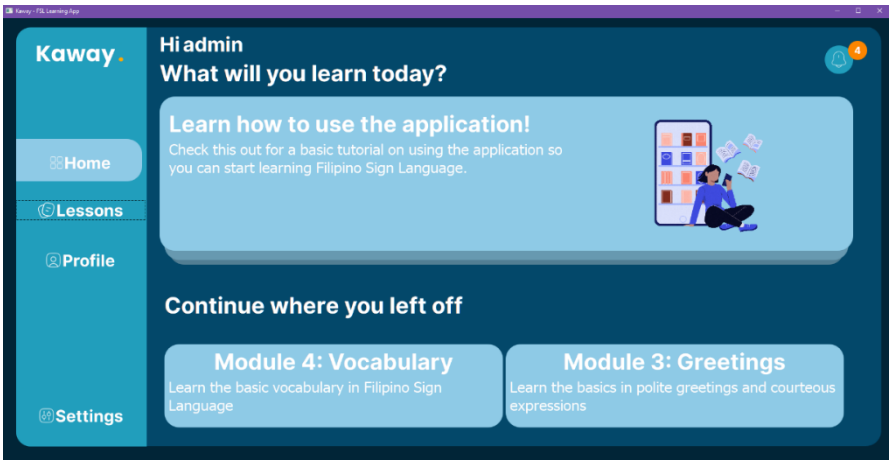


Figure 21. Modules Screen GUI

The complete modules of the system will be shown in the Lessons tab of the application as seen on Figure 21.

3.11.2.3. Proposed FSL Assessment GUI of the FSL Tutoring System

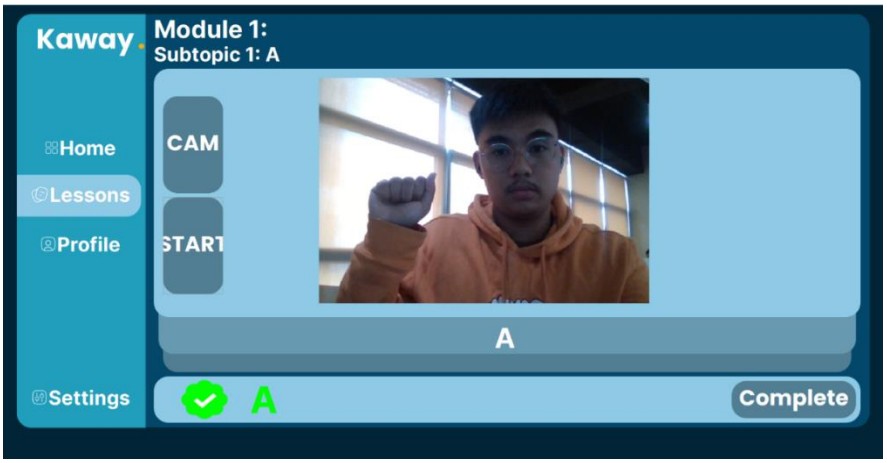


Figure 22. FSL Assessment Screen GUI

Once the user has been shown the proper use of the FSL gesture, they will be assessed as they try to mimic the sign language. If incorrect, the part they have done wrong will be highlighted red. If correct, they may proceed to the next part of the lesson.

3.11.2.4. Proposed Profile Screen GUI of the FSL Tutoring System

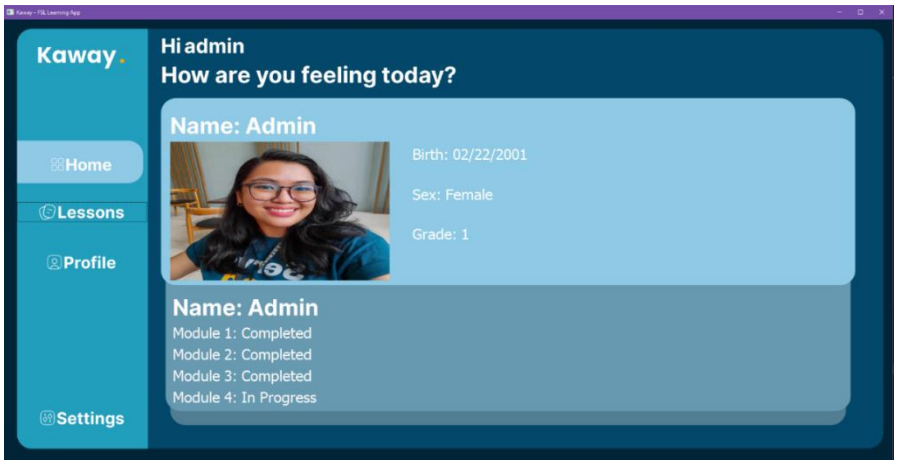


Figure 23. FSL Assessment Screen GUI

The user may be able to enter their corresponding data on their own device to track the lessons they have completed and are in progress.

3.12. Testing Procedures

The proponents of the study will perform a series of experiments for the evaluation of the accuracy of the proposed method along with its key function as a PC based tutoring system. Before the testing of the proposed system the researchers will give an informed consent to the person who will be testing the system. The testing will take place in a classroom setting for teachers to validate the tutoring system.

3.12.1. Informed Consent Form

Consent Form

Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processing and Deep Learning

I _____ (participant name), agree to participate in the research project titled Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processing and Deep Learning, conducted by Julian Cruz, Joshua Gonzales, Kayley Pichay, and Hyan Suamina who has (have) discussed the research project with me.

I have received, read and kept a copy of the information letter/plain language statement. I have had the opportunity to ask questions about this research and I have received satisfactory answers. I understand the general purposes, risks and methods of this research.

I consent to participate in the research project and the following has been explained to me:

- the research may not be of direct benefit to me.
- my participation is completely voluntary.
- my right to withdraw from the study at any time without any implications to me.
- the steps that have been taken to minimise any possible risks.
- what I am expected and required to do.
- whom I should contact for any complaints with the research or the conduct of the research.
- I am able to request a copy of the research findings and reports.
- security and confidentiality of my personal information.

In addition, I consent to:

- audio-visual recording of any part of or all research activities (if applicable)
- publication of results from this study on the condition that my identify will not be revealed.

Name: _____ (please print)

Signature: _____

Date: _____

Figure 24. Informed Consent Form

Figure 14 shows the informed consent form that the researchers will use for the teachers that will test the proposed system.

3.12.2. Testing of Tutoring System Algorithm

a) Place the PC or Laptop on a desk where the person can be seen in the range of the camera.

b) Turn on device.

c) Place the teacher in the range of the camera to enact Filipino Sign Languages to the system

d) Record the results on each sign language whether the gesture should fall as

TP, TN, FP, FN.

e) Compute the accuracy, precision, specificity, reliability, F1 score, and MCC score.

f) If the accuracy of the model of the tutoring system is below the target accuracy, then retraining and testing with the use of different parameters should be done until the wanted accuracy is shown.

3.12.3. Testing the Desktop and Web Camera

a) Check the status of the Desktop and Web camera if it is fully functional

b) Record together with the testing of sign language recognition. Check if the camera is able to pick up the hand gestures of the sign language user.

3.12.4. Testing the Tutoring System

a) Check the GUI of the system if it is properly displaying the data

b) Check and record if the system is able to properly identify if the gesture is right or wrong.

3.13. Proposed Table for Test and Results

For Specific Objective 4:

Table 6. Truth Table for True Positive, True Negative, False Positive, False Negative

Ground Truth	Test Results (Prediction)	Verdict
0	0	True Negative
0	1	False Positive
1	0	False Negative
1	1	True Positive

Table 7. Ground Truth and Testing Table for FSL Recognition

	Trial # 1			
Sign Language	**Ground Truth**	**Test Result (Prediction)**	**Verdict**	**No. Of Trials**
A				
B				

C				
D				
...				
Y				
Z				
Magandang Umaga				
Magandang Hapon				
Magandang Gabi				
Kumusta ka				
Magandang araw				
Kailan				
Sino				
Ako si/ Ang pangalan ko ay				
Ano ang pangalan mo?				
Paalam				
Ingat ka				
Maraming Salamat				
Pasensya na				
Bahay				
Pinto				
Silid				
Sala				
Kusina				
Guro				
Ilang taon ka na?				
Ube				
Dilaw				

Table 8. Total TP, TN, FP and FN of Filipino Sign Language

Sign Language	No. of TP	No. of TN	No. of FP	No. of FN
A				
B				
C				
D				

...				
Y				
Z				
Magandang Umaga				
Magandang Hapon				
Magandang Gabi				
Kumusta ka				
Magandang araw				
Kailan				
Sino				
Ako si/ Ang pangalan ko ay				
Ano ang pangalan mo?				
Paalam				
Ingat ka				
Maraming Salamat				
Pasensya na				
Bahay				
Pinto				
Silid				
Sala				
Kusina				
Guro				
Ilang taon ka na?				
Ube				
Dilaw				
TOTAL				

Table 9. Confusion Matrix

	A	**B**	**C**	**D**	**...**	**X**	**Y**	**Z**	**...**
A									
B									
C									
D									
...									
X									

Y									
Z									
...									

	Correct-Output	**Incorrect-Output**	
Correct FSL	TP=	FN=	Total Correct FSL=
Incorrect FSL	FP=	TN=	Total Incorrect FSL=
	Total Correct-Output=	Total-Incorrect Output=	

Testing Functionality

Table 10. Table for testing the Camera

Trial No.	**Result (1- Correct; 0- Incorrect)**
1	
2	
3	
4	
5	

Table 11. Table for testing the laptop

Trial No.	**Result (1- Correct; 0- Incorrect)**
1	
2	
3	
4	
5	

Table 12. Table for testing the Application

Trial No.	**Result (1- Correct; 0- Incorrect)**
1	
2	
3	
4	
5	

For Specific Objective 5:

Testing Effectiveness as a Tutoring Application

To effectively achieve the study's primary objective, the system must be able to proficiently serve its purpose as a tutoring application and deploy the lessons and

assessments efficiently to the user. In order to test its effectiveness in tutoring, the test population will each be given a Likert-Scale survey to evaluate the various components of the application. A similar approach was done in a related study, specifically, "i-Sign: Sign Language Learning Application Via Gamification" in which users were made to answer questionnaires with 3-point scale evaluation attributes testing for its efficiency, effectiveness and satisfaction [39]. These three attributes correspond to the most frequent usability attributes for testing mobile applications as mentioned in similar research evaluating software usability [74]. However, this study will also feature the less frequent attributes as they are deemed to be necessary in the assessment of the system in tutoring. Specifically, these are learnability, memorability, and cognitive load. The table below highlights the following criteria used for the survey and its specific questions. A 5-point evaluation attribute was also implemented from Strongly Disagree (1), Disagree (2), Neither Agree or Disagree (3), Agree (4), and Strongly Agree (5).

Table 13. Table for Likert-Scale Survey – FSL Tutoring Application

	1	**2**	**3**	**4**	**5**
Effectiveness					
The FSL tutoring application effectively helps reinforce and apply FSL concepts in practical scenarios.					
The user was effectively taught the basics of Filipino Sign Language by the tutoring application.					
The user feels confident in their ability to replicate FSL after the use of the application.					
The FSL tutoring application accurately assessed the user's sign language gestures.					
The real-time feedback provided during the assessment was helpful for correcting my sign language gestures.					
Efficiency					
The FSL tutoring application allows for efficient and timely delivery of FSL learning materials and resources.					
The FSL tutoring application is streamlined in providing a structured and organized learning environment.					
The FSL tutoring application efficiently integrates assessment and					

feedback mechanisms for user progress.					
The time spent using the application was well worth the learning outcomes.					
The FSL tutoring application efficiently guided the user through the lessons on Filipino Sign Language.					
Learnability					
The user easily grasps and understands the FSL concepts through the FSL tutoring application.					
The FSL tutoring application is intuitive and clear with the learning pathways and structures.					
The lessons were presented in a way that was easy to understand and follow.					
The lessons provided sufficient guidance for learning new gestures.					
The user feels confident in their ability to follow along with the lessons provided by the FSL tutoring application.					
Memorability					
The user significantly demonstrates improved learning outcomes using the FSL tutoring application.					
The user notably shows progress and achievement through the FSL tutoring application.					
The user consistently shows learning progress through the assessments provided by the FSL tutoring application.					
The user can easily recall and replicate the FSL gestures taught by the application.					
The assessment feedback aided in reinforcing the memorization of gestures.					
Cognitive Load					
The application presented information in a way that was not overwhelming.					
The user did not feel mentally exhausted after using the application for learning.					
The application effectively managed the amount of information presented at one time.					
The assessment process did not create an undue mental burden.					
The user feels the cognitive demands of using the application is reasonable.					
Satisfaction					
The user is satisfied with the system and functionalities of the FSL tutoring application for teaching purposes.					
The user is likely to recommend the FSL tutoring application to other FSL educators.					
The user is satisfied with the overall learning experience provided by the application.					
The user feels confident in their ability to communicate using FSL after using the application.					

The application has increased the user's interest in learning the Filipino Sign Language.					

Table 14. Table for Likert-Scale Survey – Traditional FSL Tutoring

	1	2	3	4	5
Effectiveness					
The traditional FSL tutoring method effectively helps reinforce and apply FSL concepts in practical scenarios in a classroom.					
The instructor can effectively teach the basics of Filipino Sign Language without any hindrances in a traditional classroom.					
The students feel confident in their ability to replicate FSL after traditional tutoring sessions.					
The instructor can accurately assess sign language gestures of all students in a traditional classroom.					
The feedback provided by the instructor during tutoring sessions was helpful for correcting sign language gestures of all students in a traditional classroom.					
Efficiency					
The traditional FSL tutoring method allows for efficient and timely delivery of FSL learning materials and resources.					
The traditional FSL tutoring method is streamlined in providing a structured and organized learning environment.					
The traditional tutoring method efficiently integrates assessment and feedback mechanisms for user progress.					
The time spent in traditional tutoring sessions was well worth the learning outcomes.					
The traditional tutoring method can efficiently guide all students through the lessons on Filipino Sign Language.					
Learnability					
The user can easily grasp and understand the FSL concepts through the traditional FSL tutoring.					
The traditional tutoring method is intuitive and clear with the learning pathways and structures.					
The lessons were presented in a way that was easy to understand and follow during traditional FSL tutoring.					
The lessons provided sufficient guidance to all students for learning new gestures.					
The students feel confident in their ability to follow along with the lessons provided by the traditional FSL tutoring method.					
Memorability					
The students can significantly demonstrate improved learning outcomes using the traditional FSL tutoring method.					
The students can notably show progress and achievement through the traditional learning every session.					

The student can consistently show learning progress through the assessments provided by the traditional FSL tutoring method.					
The student can easily recall and replicate the FSL gestures taught during traditional sessions.					
The feedback from the instructor aided in reinforcing the memorization of gestures done by all the students in a classroom.					
Cognitive Load					
The instructor can present information in a way that was not overwhelming to all students in a classroom.					
The students do not feel mentally exhausted after traditional tutoring sessions.					
The instructor effectively managed the amount of information presented at one time.					
The assessment process did not create an undue mental burden to the students.					
The students feel that the cognitive demands during tutoring sessions were reasonable.					
Satisfaction					
The students seem satisfied with the traditional FSL tutoring method and its functionalities for teaching purposes.					
I am likely to recommend traditional FSL tutoring to others interested in learning FSL than with an FSL tutoring application.					
The students seem satisfied with the overall learning experience provided by the traditional FSL tutoring.					
The students feel confident in their ability to communicate using FSL after using the application.					
Traditional FSL tutoring has increased the students' interest in learning Filipino Sign Language.					

3.14. Design Constraints

As with every study, the system design for this study contains limitations. When developing applications, the researchers should be aware of these flaws. To avoid any errors in the system design, it is important to identify the design constraints before beginning the design. Certain aspects such as economic, health and safety, environment, manufacturability, time, and sustainability will be considered in this study.

A. Economic

This part talks about the cost and supply of materials which will be used by

the system. These materials are namely the hardware: Personal Computer/Laptop and Web Camera. These materials are known to be relatively cheap compared to related studies explored within the study since the set up will only need a web camera and a PC. Additionally, the web camera installed in the user's personal laptop can also be used in the set up. This study will mainly focus on the programming and coding of the system.

B. Health and Safety

The health and safety constraints consider the identification of hazards and potential risks. Since the proponents of this research tries to make a system that will be non-invasive to the users and will only need a PC and a webcam or a personal laptop, it can be ergonomically correct to say that it will be not much of a threat since it is only a system. This will have no physical contact with users since this is a non-invasive system and will only require the users to interact with the system through SL gestures. The only possible hazard that can be considered is when used on the students and the amount of time they look at the screen which may affect their vision.

C. Environment

The environmental constraints focus on the harm that the system can incur in the environment. Since the proposed research mainly focuses on the design of the tutoring system, there will be no direct harm to the environment. The only environmental effect it might incur would be during the manufacturing stage of the electronic equipment used in the proposed design. The system also does not need a large amount of electricity since the tutoring system will be used in a PC or laptop where the webcam will be connected to. Lastly, the equipment used in the study

can also be reused for future purposes.

D. Manufacturability

The manufacturability constraint will not be a problem for this research since it won't use many hardware components. The only hardware components used are the PC or personal laptop and the web camera. The system will mostly be done in software components for the programming, coding, and design.

E. Time

Time constraint can be considered in this research since the proponents are the Filipino sign language teachers who will validate the proposed system and the students who are enrolled on a tri-semester mode. The activity that will take the most time is the training of the deep learning model, coding of the system, and system testing.

F. Ethical

The study will have to examine the ethical considerations that come with the procedures of the proposed testing. The prototype was made with young students in mind since they will benefit the most with the study. However, due to ethical reasons, they will not be the ones used for testing the prototype. Approval of ethical permissions will take quite some time and will not fit within the timeline expected by the proponents. Experts who have taught FSL were chosen instead to validate the effectiveness and reliability of the proposed FSL tutoring system.

G. Sustainability

The system that the researchers designed can last a long time and if there are any new Filipino Sign Language gestures can be updated and trained in the algorithm of the system. Additionally, any new lessons that are added by the school

or teacher can be added to the application. The system can be used in any PC or laptop that the teacher has, and the web cam can be easily found on the internet.

Chapter 4

RESULTS AND DISCUSSIONS

This chapter presents the results of the study according to the stated objectives. Each component was discussed thoroughly and to meet the target accuracy, a series of samples were trained and tested. The interpretations were based on tables with sample computations.

4.1. Prototype

4.1.1. Final Design

Figure 25. Actual Prototype

The hardware prototype consists of a Personal Laptop or Computer, and a Camera. The application contains the system and GUI for the lessons and recognition of the FSL. It is important to place the laptop or computer together with the camera in a well-lighted area, and the user's upper body must be clearly seen in

the camera interface of the application. This will allow the system prototype to recognize and detect the FSL properly.

4.1.2. Parts of the Prototype

4.1.2.1. Hardware Setup of the Prototype

Figure 26. Setup of the Personal Laptop

Figure 26 shows the Personal Laptop setup; a computer may also be used if there is a connected camera. The laptop, with a working camera, is placed in front of the user, the subtopic of the module will be shown on the screen with a video tutorial of the FSL for the user to learn first. After learning how to sign the given FSL, the user can try it for themselves using the camera recognition of the system.

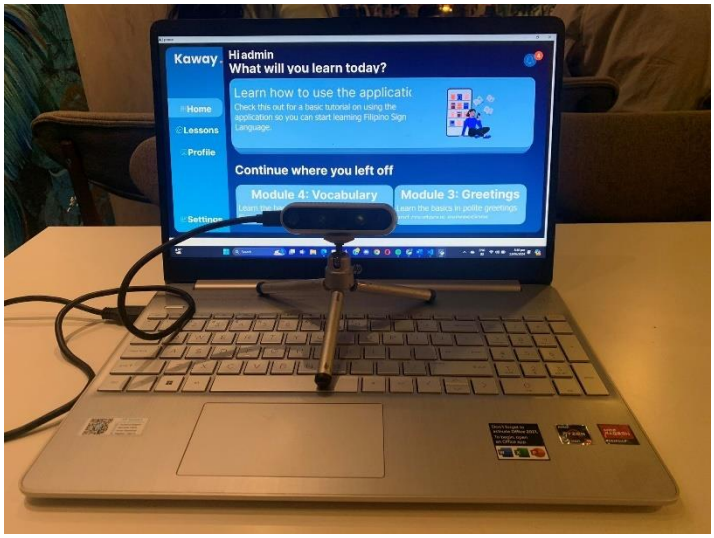


Figure 27. Setup of Camera

Figure 27 shows the ideal setup for the camera if the laptop has no built-in camera. The camera used must have a clear view of the user for a more accurate recognition of the FSL.

4.1.3. Software Development

4.1.3.1. FSL Recognition

For the application's detection algorithm, the proponents mainly relied on Python, specifically, Python 3.8.7, and its primary dependencies such as OpenCV, NumPy, scikit-learn, and Keras. The specific version of Python was also chosen to allow compatibility with all required dependencies for the FSL recognition and the application's GUI. These dependencies and libraries were necessary for FSL detection and were installed on the Personal Computer (PC) where the system will be deployed. Lastly, the application's detection algorithm was developed with an Integrated Development Environment (IDE), specifically, Visual Studio Code.

Table 15. Table of System's Detection's Dependencies

Dependency	Version	Description

Python	3.8.7	For the system, Python was chosen as the programming language due to its convenience with readily available libraries for machine learning and object detection.
OpenCV	4.9.0	OpenCV was utilized as a software library for machine vision for the application. This library allowed the system to integrate with the camera for video and image implementation.
NumPy	1.24.3	The system uses multidimensional arrays to store data gathered in real time from the feature landmarks. For this case, NumPy was utilized for efficient handling of data, and array manipulation.
scikit-learn	1.3.2	Scikit-learn was utilized as the system's machine-learning library integrated with NumPy to manipulate the data manipulation during the training and testing phase.
Keras	2.13.1	Keras was integrated into the system for neural networks API for the training of the model with LSTM, and its configuration of the parameters for deep learning.

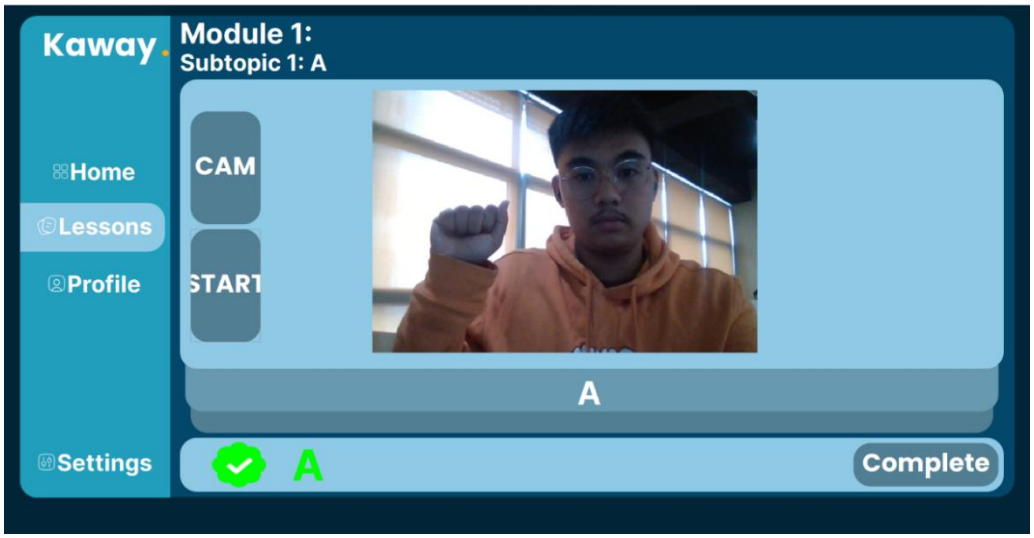


Figure 28. FSL Detection of the Application

Figure 28 shows the application's detection, recognition, and feedback evaluation with the user's FSL gestures. During the detection phase, the system's algorithm extracts the feature landmarks for each frame gathered by the application. For Module 1 and the alphabet, the algorithm considers 15 frames before the system concludes with a result. For Modules 2 to 4, dynamic signs are longer, thus, 40

frames were collected for a result to be shown. A frame has 1662 values, which consider face, pose, and right and left-hand landmarks, with sizes of 468x3, 33x4, 21x3, and 21x3, respectively.

	A	B	C	D	E	F
1	pose_x	pose_y	pose_z	pose_visibility	pose_x	pose_y
2	0.506271362304688	0.375679969787598	-1.28452622890472	0.999747931957245	0.540310382843018	0.290903449058533

	O	P	Q	R	S	T
1	pose_z	pose_visibility	pose_x	pose_y	pose_z	pose_visibility
2	-1.20040965080261	0.999396562576294	0.476144969463348	0.302424788475037	-1.18562233448029	0.999533116817474

	EM	EN	EO	EP	EQ	ER
1	face_y	face_z	face_x	face_y	face_z	face_x
2	0.298609018325806	-0.0294812880456448	0.522286057472229	0.324865460395813	-0.0500769540667534	0.522192597389221

	FC	FD	FE	FF	FG	FH
1	face_z	face_x	face_y	face_z	face_x	face_y
2	0.00217413040809333	0.522225975990295	0.220047876238823	0.0044926842674613	0.521860301494598	0.15738433599472

	BGD	BGE	BGF	BGG	BGH	BGI	BGJ	BGK	BGL	BGM	BGN	BGO	BGP	BGQ	BGR	BGS	BGT	BGU	BGV	BGW
1	lh_y	lh_z	lh_x	lh_y	lh_z	lh_x	lh_y	lh_z	lh_x	lh_y	lh_z	lh_x	lh_y	lh_z	lh_x	lh_y	lh_z	lh_x	lh_y	lh_z
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	BIU	BIV	BIW	BIX	BIY	BIZ
1	rh_y	rh_z	rh_x	rh_y	rh_z	rh_x
2	0.493714869022369	-0.00709237810224295	0.447323441505432	0.435970634222031	-0.00567082641646266	0.464314579963684

	BKC	BKD	BKE	BKF	BKG	BKH
1	rh_z	rh_x	rh_y	rh_z	rh_x	rh_y
2	-0.0348071828484535	0.560912370681763	0.460950940847397	-0.035884153097868	0.546769618988037	0.423764824867249

Figure 29. Sample Landmark Data

The images above show the sample data structured into a multi-dimensional array with NumPy. For each frame, the extracted data are loaded into a pre-trained LSTM model, which outputs the probabilities for each action (sign). The action with the highest probability is considered the recognized sign.

4.1.3.2. Desktop Application

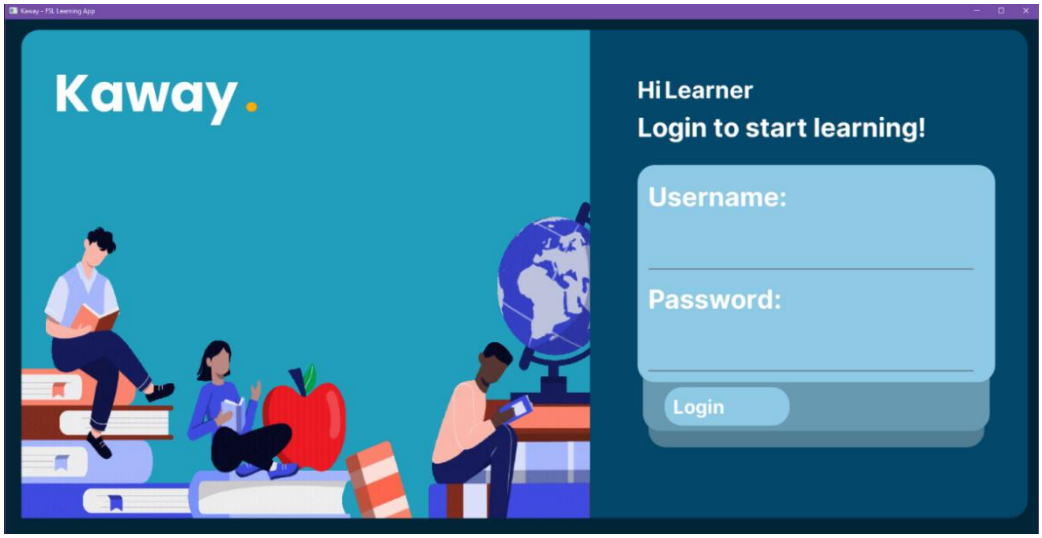


Figure 30. Application Login Screen

Figure 30 shows the login screen of the application once opened. The application features animated screens with an audio interface such as a background music and audio for feedback evaluation. Its GUI was made with PyQt, a cross-platform GUI toolkit that features a convenient process in developing pages for the application.

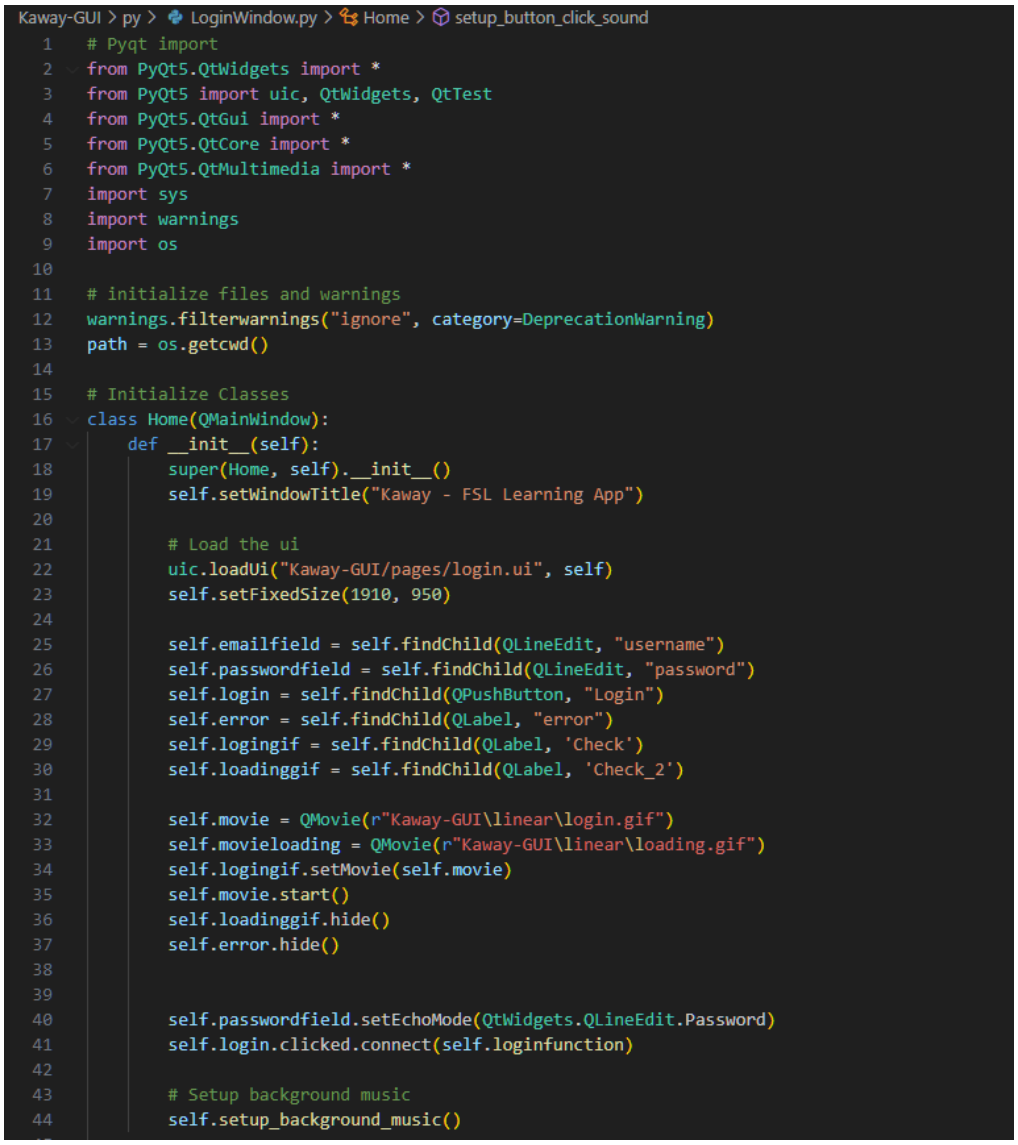


Figure 31. Sample Code for Kaway Application

Figure 31 shows the sample code for the Kaway application built with the programming language Python and its libraries, which provides tools for creating a GUI, handling multimedia functionalities, and sign language detection. Its code starts with the initialization of the necessary modules, and sets up the GUI, video tutorials, and assessment.

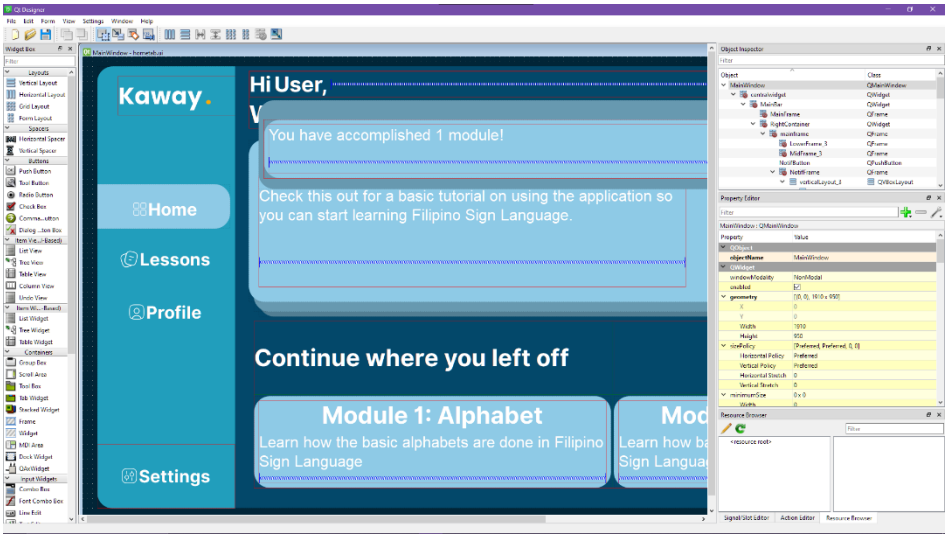
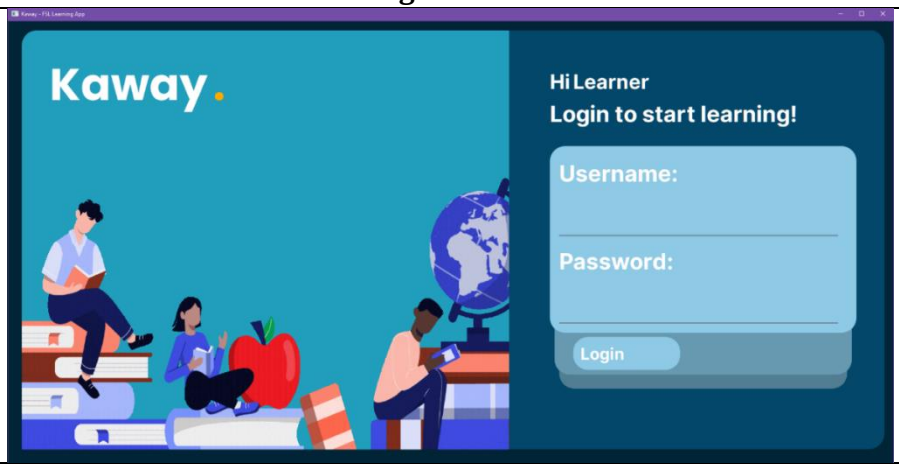


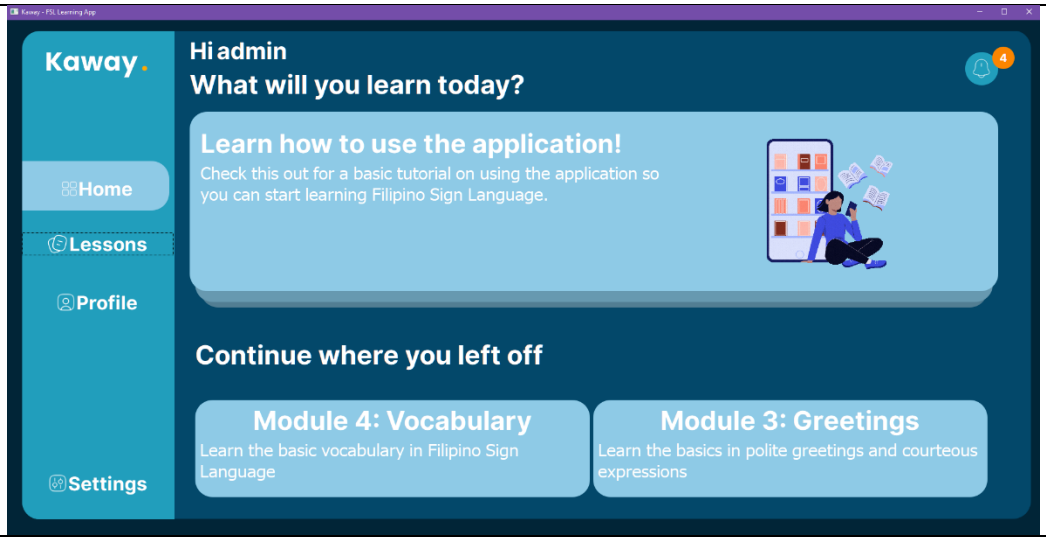
Figure 32. PyQt Designer

For each page of the GUI, PyQt Designer was utilized in creating the .ui files necessary to be loaded whenever a new screen is presented. In total, the application utilized 12 .ui files for all the pages. The login screen loads up as a QMainWindow, while the rest of the pages were made as QWidgets for compatibility of switching screens through QStackedWidgets. Additionally, the application features QPushButton, QLabel, QFrame, QScrollBar, and QVideoWindow to complete each page.

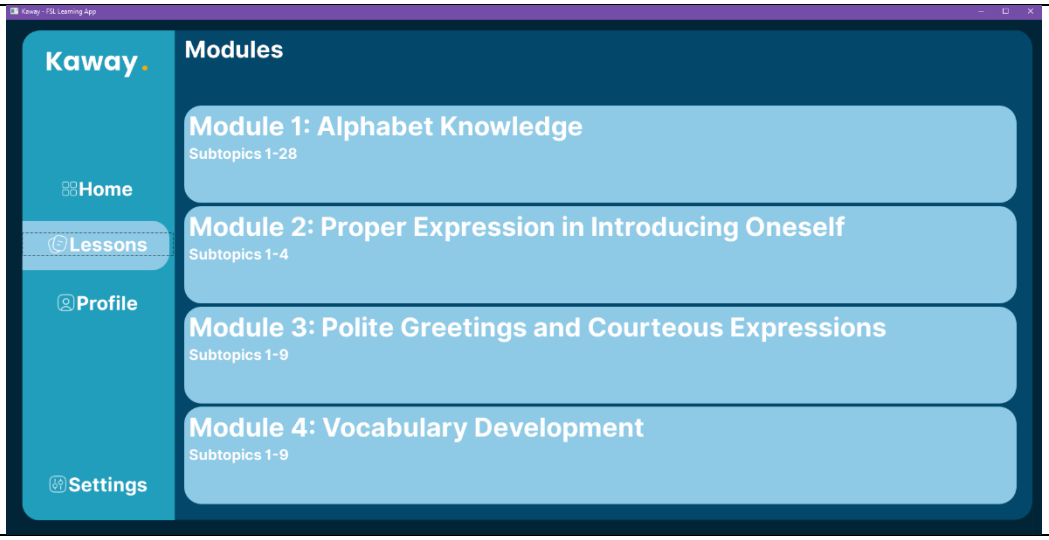
Table 16. List of Application Pages

Login Screen


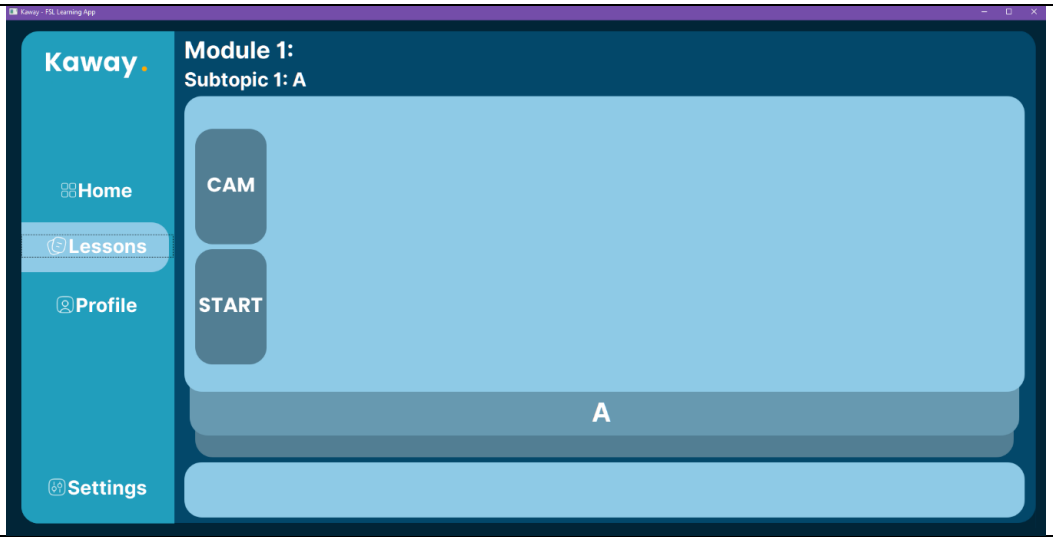
Home Screen



Lesson Screen



Assessment Screen



4.1.3.3. Machine Learning Model Creation

4.1.3.3.1. Data Collection and Classification

As discussed in the previous chapter, the machine learning model for detecting FSL would need to be trained with at least 33 samples for each gesture. In order to acquire the dataset, the proponents have developed a process in Python utilizing OpenCV to record a total of 40 frames thirty-three times for each dynamic FSL gesture. On the other hand, for static letters, each gesture had a range of around 100-400 frames collected to reinforce the model further.

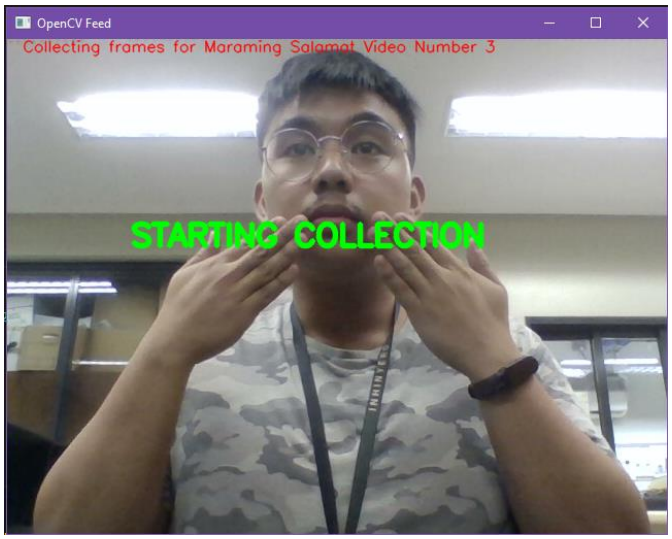


Figure 33. Data Collection

The figure above shows the process in collecting the data that will be used for the creation of the system's model. 70% of the total dataset will be used for training the model and the remaining 30% will be used for testing the trained model. During the process of training, the shuffle parameter was set to 'True' so that the data would be randomly shuffled before splitting, which helps prevent any ordering bias in the dataset.

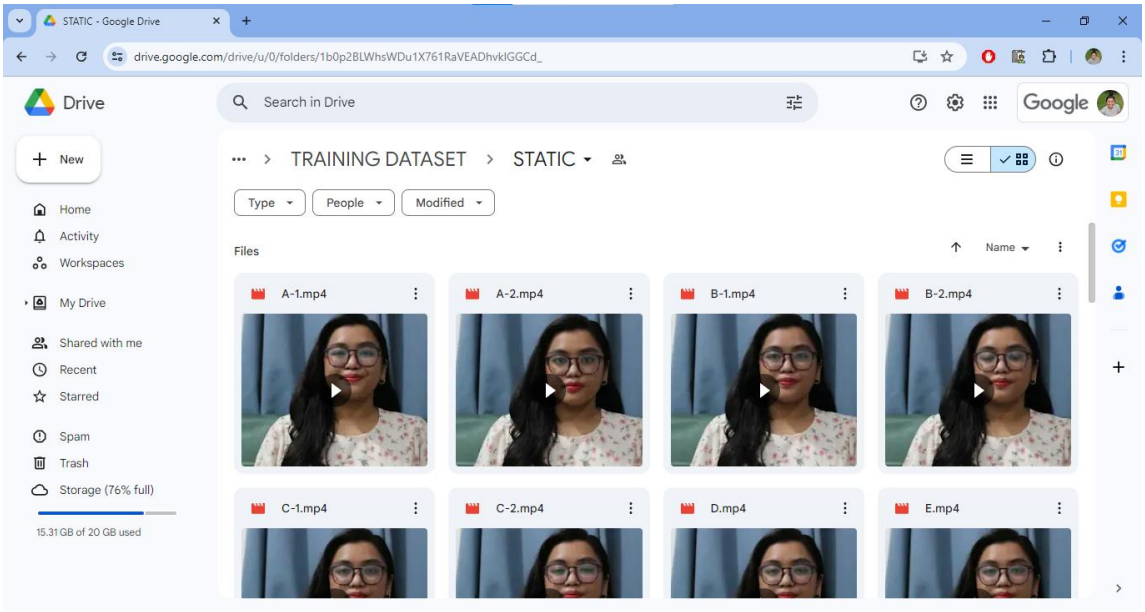


Figure 34. Static Training Dataset Add description and hyperlink

The proponents have a Google Drive folder which contains the recorded training dataset and tutorial videos. The figure illustrates the Google Drive folder where the proponents keep the collected datasets used to train the KAWAY program. This folder contains over 33 data samples for each FSL Static sign, enabling the program to learn effectively. The proponents also have a similar folder for the FSL Dynamic signs. Additionally, the proponents have a folder which includes recorded tutorial videos that users can view, learn from, and follow, providing comprehensive resources for mastering KAWAY. The hyperlink for the Google Drive folder is the following:

https://drive.google.com/drive/folders/1M4c9ujLgEGlSHxXON2GPeu9rFoyRqb6B?usp=sharing

4.1.3.3.2. Model Training

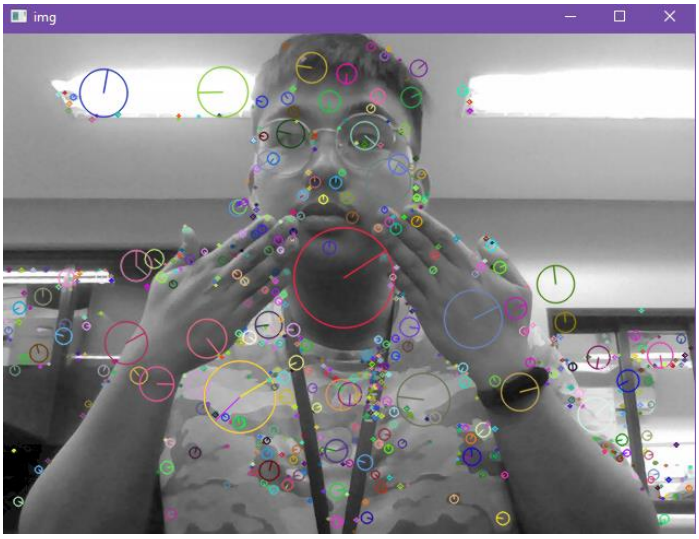


Figure 35. Scale Invariant Feature Transform

After collection of the data, the training system will then loop through the images and read its grayscale version, highlighting its features with an image processing technique called Scale Invariant Feature Transform. The system will then process the image with keypoints and extract face, pose, and hand landmarks. These landmarks are normalized and added to the list with the corresponding labels.

Table 17. Summary of Hyperparameters

Hyperparameter	Value
Model Architecture	
LSTM Layers	1
LSTM Units	64
Activation Function	Relu
Dense Layers	2
Dense Layer (1)	16
Dense Units (Output)	5
Activation (Output)	Softmax
Training Configuration	
Optimizer	Adam
Loss Function	Categorical Crossentropy
Accuracy Metric	Categorical Accuracy
Epochs	2000
Minimum Delta	0.0001
Patience	20

Verbose	1

The FSL recognition model utilized Long Short-Term Memory neural networks for sequence classification consisting of Model Training and Evaluation. The system creates a dictionary for the classification of the data. The values of extracted keypoints are in NumPy files for efficient handling in Python. Afterwards, a Sequential model from Keras was initialized which is a linear stack of layers. Additionally, it adds an LSTM layer with an initial 64 units, a 'relu' activation function, and input shape that matches the sequences gathered during the data collection. Two dense layers were also added, with the second as the output later with a number of units equal to the number of gestures trained and uses the 'softmax' activation function to output probabilities. Lastly, the LSTM model used the Adam optimizer with the 'categorical_accuracy' to track the model accuracy. This LSTM model would then be used to train the FSL recognition model for up to 2000 epochs but will stop if the categorical_accuracy does not improve significantly after a certain number of epochs.

```
Epoch 52/2000
3/3 [==============================] - 0s 54ms/step - loss: 0.1725 - categorical_accuracy: 0.9348
Epoch 53/2000
3/3 [==============================] - 0s 55ms/step - loss: 0.2963 - categorical_accuracy: 0.8913
Epoch 54/2000
3/3 [==============================] - 0s 52ms/step - loss: 0.3338 - categorical_accuracy: 0.8804
Epoch 55/2000
3/3 [==============================] - 0s 52ms/step - loss: 0.1812 - categorical_accuracy: 0.9457
Epoch 56/2000
3/3 [==============================] - 0s 52ms/step - loss: 0.1063 - categorical_accuracy: 0.9783
Epoch 57/2000
3/3 [==============================] - 0s 56ms/step - loss: 0.1448 - categorical_accuracy: 0.9565
Epoch 58/2000
3/3 [==============================] - 0s 54ms/step - loss: 0.1187 - categorical_accuracy: 0.9783
Epoch 59/2000
3/3 [==============================] - 0s 53ms/step - loss: 0.1324 - categorical_accuracy: 0.9457
Epoch 60/2000
3/3 [==============================] - 0s 53ms/step - loss: 0.1070 - categorical_accuracy: 0.9783
Epoch 61/2000
3/3 [==============================] - 0s 55ms/step - loss: 0.0670 - categorical_accuracy: 0.9783
Epoch 62/2000
3/3 [==============================] - 0s 55ms/step - loss: 0.0528 - categorical_accuracy: 0.9891
Epoch 63/2000
3/3 [==============================] - 0s 54ms/step - loss: 0.0411 - categorical_accuracy: 0.9891
Epoch 64/2000
3/3 [==============================] - 0s 51ms/step - loss: 0.0366 - categorical_accuracy: 0.9891
Epoch 65/2000
3/3 [==============================] - 0s 50ms/step - loss: 0.0326 - categorical_accuracy: 0.9891
Epoch 66/2000
3/3 [==============================] - 0s 55ms/step - loss: 0.0313 - categorical_accuracy: 0.9891
Epoch 67/2000
3/3 [==============================] - 0s 51ms/step - loss: 0.0339 - categorical_accuracy: 0.9891
Epoch 68/2000
3/3 [==============================] - 0s 53ms/step - loss: 0.0277 - categorical_accuracy: 0.9891
Epoch 69/2000
3/3 [==============================] - 0s 53ms/step - loss: 0.0243 - categorical_accuracy: 0.9891
Epoch 69: early stopping
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 64)                442112

 dense (Dense)               (None, 16)                1040

 dense_1 (Dense)             (None, 4)                 68

=================================================================
Total params: 443220 (1.69 MB)
Trainable params: 443220 (1.69 MB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 36. Model Training

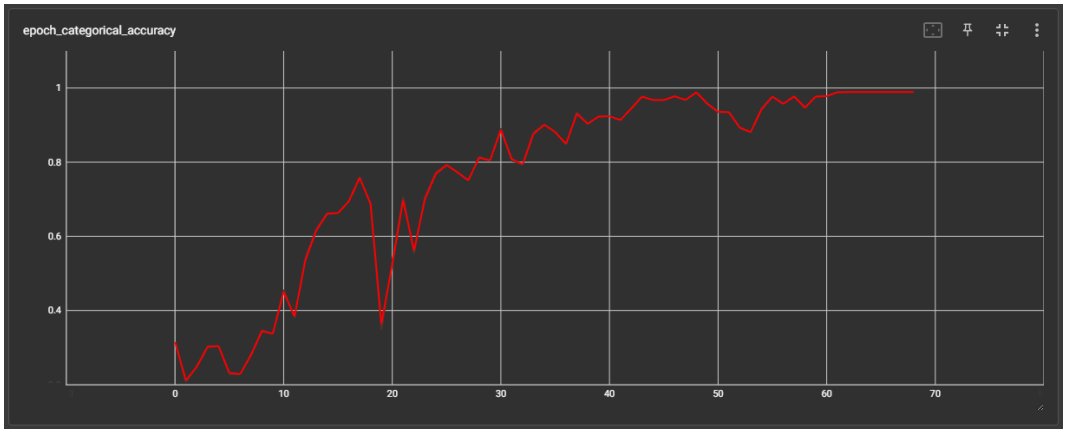


Figure 37. Categorical Accuracy

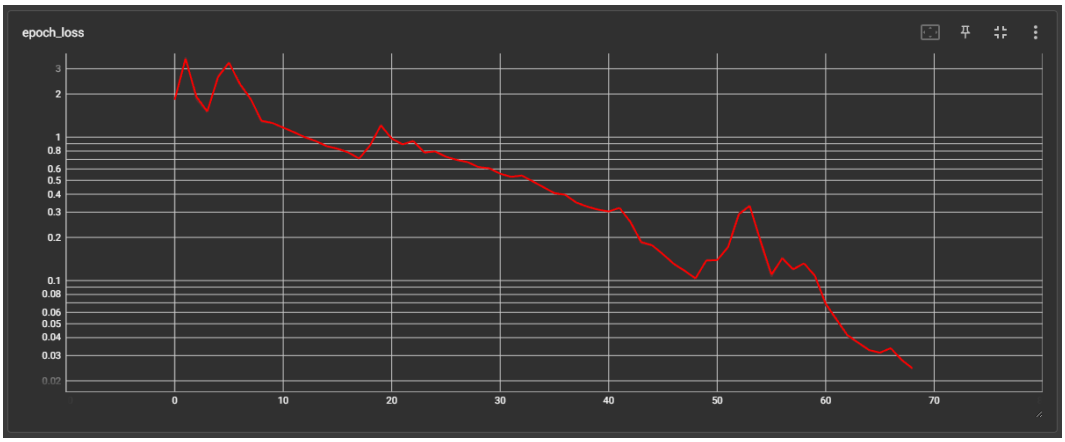


Figure 38. Model Training Loss

Figure 36 shows the process of training the model used in the recognition of FSL gestures. The model's accuracy based on its own metrics as tested by the 30% of the data is 98.91% and was stopped at that point when it has not reached further improvements. As seen on Figure 37, the model reached 0.9891 and began to stagnate. There is also an inversely proportional relationship to the system's accuracy and its loss. It is necessary to consider the loss during the training process to prevent the system from being overtrained and shows how well the model's predictions match the actual labels. A lower loss indicates that the model's predicted probability distribution is closer to the actual distribution.

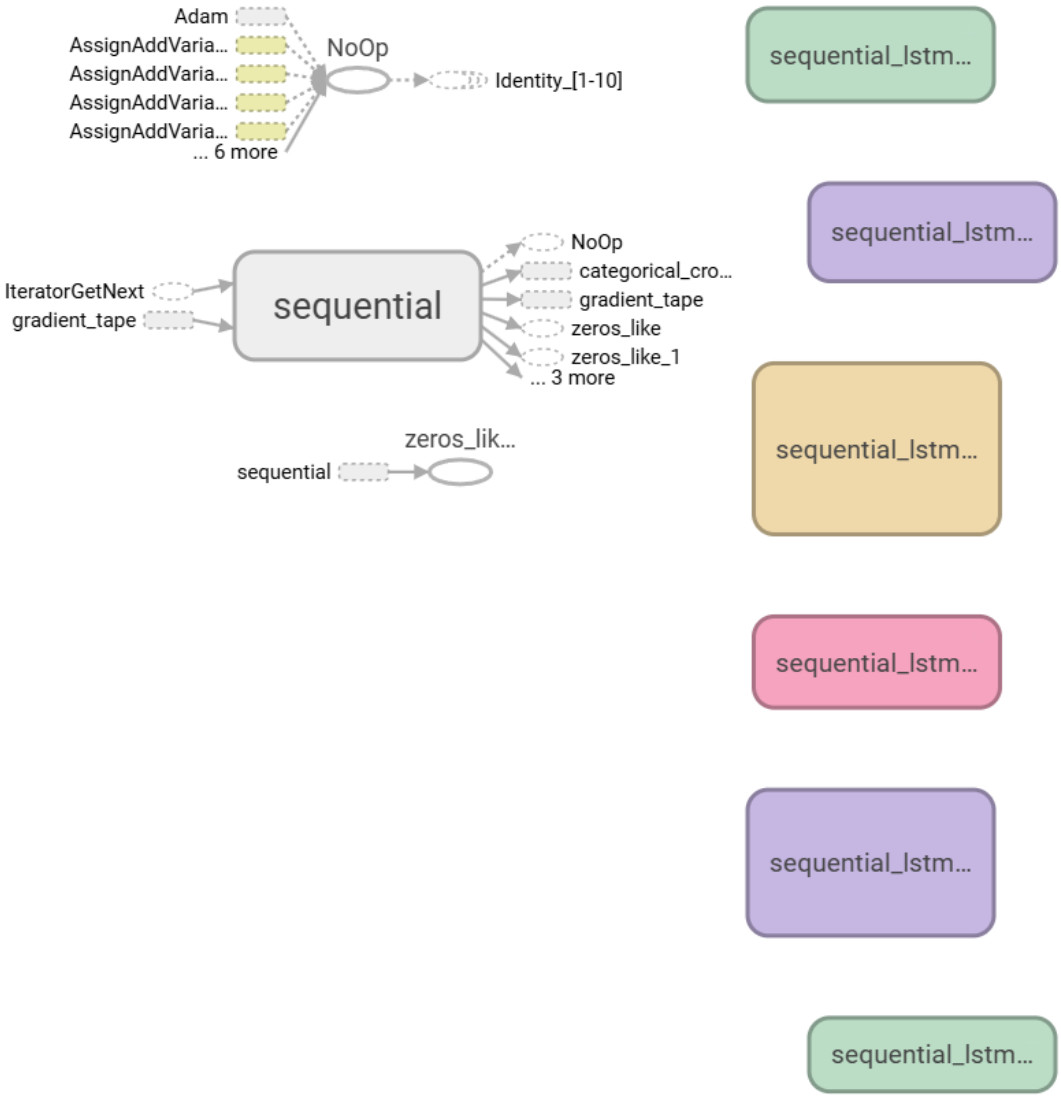


Figure 39. Model Computational Graph – Sequential LSTM Layers

The graph shown on Figure 39 represents the computational graph of the model during the training, showing the flow of data and operations, including the loss function and the optimizers. The graph includes the various layers of the Sequential model utilized such as the LSTM and Dense layers. The Adam optimizer was used in updating the model weights and is connected to the gradient computation. The Gradient Tape block is used to record the operations for

automatic differentiation and captures the gradients of the loss with respect to the model's weights.

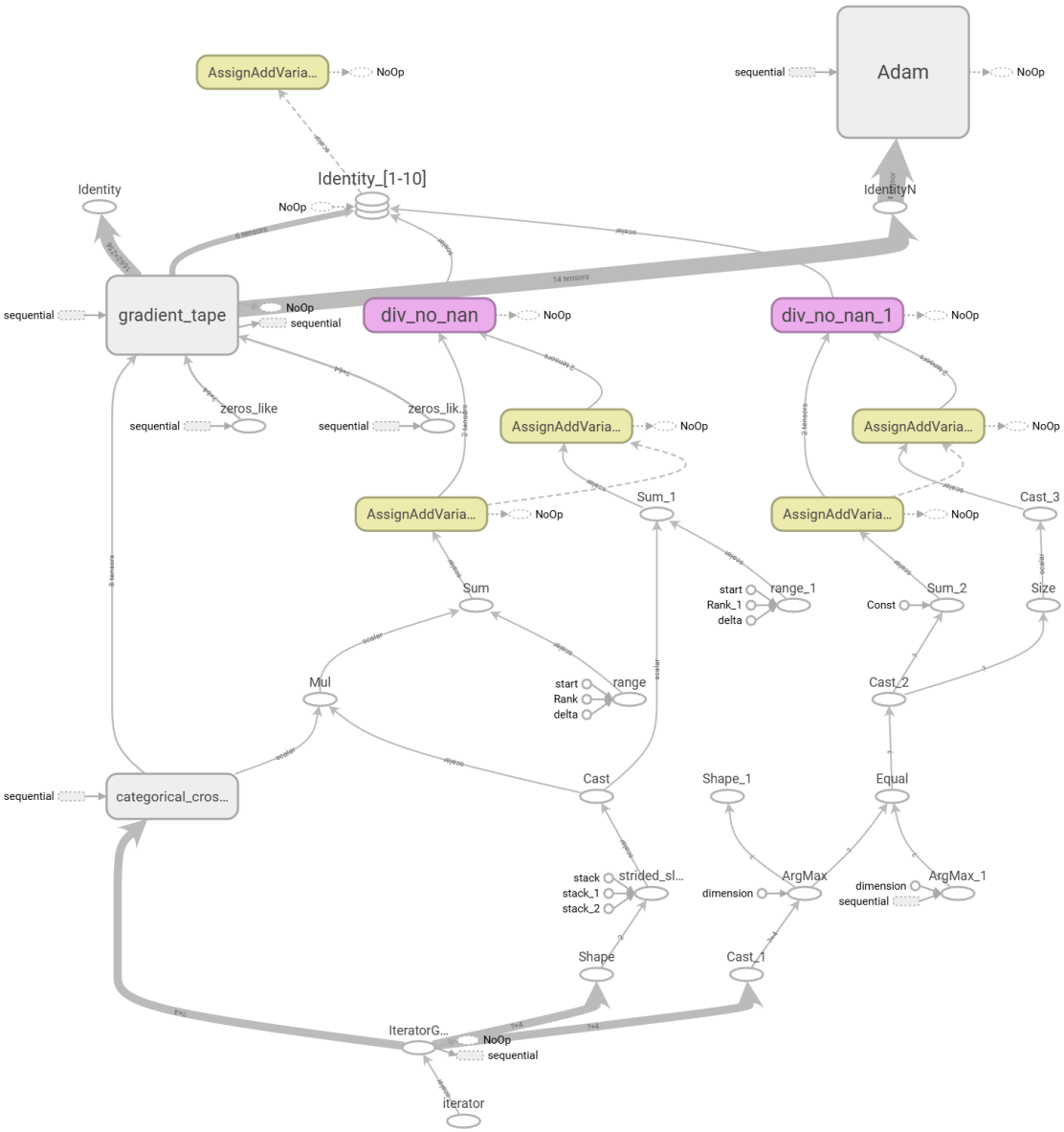


Figure 40. Model Computational Graph – Update of Model Weights

4.2. Tables and Discussion of Results

4.2.1. Model Accuracy

Table 18. Ground Truth and Testing Table for FSL Recognition of Expert no. 1

	Trial # 1				Trial # 2		
Sign Language	**Ground Truth**	**Test Result (Prediction)**	**Verdict**	**Sign Language**	**Ground Truth**	**Test Result (Prediction)**	**Verdict**
A	1	1	TP	**A**	1	1	TP
B	1	1	TP	**B**	1	1	TP
C	1	1	TP	**C**	1	1	TP
D	1	1	TP	**D**	1	1	TP
E	0	0	TN	**E**	1	1	TP
F	1	1	TP	**F**	1	1	TP
G	1	1	TP	**G**	0	0	TN
H	1	1	TP	**H**	1	1	TP
I	1	1	TP	**I**	1	1	TP
J	0	0	TN	**J**	1	1	TP
K	1	1	TP	**K**	1	1	TP
L	1	1	TP	**L**	1	1	TP
M	1	1	TP	**M**	1	1	TP
N	1	1	TP	**N**	1	1	TP
Ñ	1	1	TP	**Ñ**	1	1	TP
NG	1	1	TP	**NG**	1	1	TP
O	1	1	TP	**O**	1	1	TP
P	0	0	TN	**P**	1	1	TP
Q	1	1	TP	**Q**	1	1	TP
R	1	1	TP	**R**	1	1	TP
S	1	1	TP	**S**	1	1	TP
T	0	0	TN	**T**	1	1	TP
U	1	1	TP	**U**	1	1	TP
V	1	1	TP	**V**	1	1	TP
W	1	1	TP	**W**	1	1	TP
X	0	0	TN	**X**	1	1	TP
Y	1	1	TP	**Y**	1	1	TP
Z	1	1	TP	**Z**	1	1	TP
Ako si	1	1	TP	**Ako si**	1	1	TP

Ano ang pangalan mo?	1	1	TP	**Ano ang pangalan mo?**	1	1	TP
Ilang taon ka na?	1	1	TP	**Ilang taon ka na?**	1	1	TP
Sino	1	1	TP	**Sino**	1	1	TP
Magandang Umaga	1	1	TP	**Magandang Umaga**	1	1	TP
Magandang Hapon	1	1	TP	**Magandang Hapon**	1	1	TP
Magandang Gabi	1	1	TP	**Magandang Gabi**	1	1	TP
Magandang Araw	1	1	TP	**Magandang Araw**	1	1	TP
Kumusta ka	1	1	TP	**Kumusta ka**	1	1	TP
Paalam	1	1	TP	**Paalam**	1	1	TP
Ingat ka	1	1	TP	**Ingat ka**	1	1	TP
Maraming Salamat	1	1	TP	**Maraming Salamat**	1	1	TP
Pasensya na	1	1	TP	**Pasensya na**	1	1	TP
Bahay	1	1	TP	**Bahay**	1	1	TP
Pinto	1	1	TP	**Pinto**	1	1	TP
Silid	1	1	TP	**Silid**	1	1	TP
Sala	1	1	TP	**Sala**	1	1	TP
Kailan	1	1	TP	**Kailan**	1	1	TP
Guro	1	1	TP	**Guro**	1	1	TP
Kusina	1	1	TP	**Kusina**	1	1	TP
Dilaw	1	1	TP	**Dilaw**	1	1	TP
Ube	1	1	TP	**Ube**	1	1	TP

	Trial # 3				**Trial # 4**		
Sign Language	**Ground Truth**	**Test Result (Prediction)**	**Verdict**	**Sign Language**	**Ground Truth**	**Test Result (Prediction)**	**Verdict**
A	1	0	FN	**A**	1	1	TP
B	1	1	TP	**B**	1	1	TP
C	1	1	TP	**C**	1	1	TP

D	1	1	TP	**D**	1	1	TP
E	1	1	TP	**E**	1	1	TP
F	1	1	TP	**F**	1	0	FN
G	1	1	TP	**G**	1	1	TP
H	1	1	TP	**H**	1	1	TP
I	1	1	TP	**I**	1	1	TP
J	1	1	TP	**J**	1	1	TP
K	1	1	TP	**K**	1	1	TP
L	1	1	TP	**L**	1	1	TP
M	1	1	TP	**M**	1	1	TP
N	1	1	TP	**N**	1	1	TP
Ñ	1	1	TP	**Ñ**	1	1	TP
NG	1	1	TP	**NG**	1	1	TP
O	1	1	TP	**O**	1	1	TP
P	1	1	TP	**P**	1	1	TP
Q	1	1	TP	**Q**	1	1	TP
R	1	1	TP	**R**	1	1	TP
S	1	1	TP	**S**	1	1	TP
T	1	0	FN	**T**	1	1	TP
U	1	0	FN	**U**	1	1	TP
V	1	1	TP	**V**	1	1	TP
W	1	1	TP	**W**	1	1	TP
X	1	1	TP	**X**	1	1	TP
Y	1	1	TP	**Y**	1	1	TP
Z	1	1	TP	**Z**	1	1	TP
Ako si	1	1	TP	**Ako si**	1	1	TP
Ano ang pangalan mo?	1	1	TP	**Ano ang pangalan mo?**	1	1	TP
Ilang taon ka na?	1	1	TP	**Ilang taon ka na?**	1	1	TP
Sino	1	1	TP	**Sino**	1	1	TP
Magandang Umaga	1	1	TP	**Magandang Umaga**	1	1	TP
Magandang	1	1	TP	**Magandang**	1	1	TP

Hapon				**Hapon**			
Magandang Gabi	1	1	TP	**Magandang Gabi**	1	1	TP
Magandang Araw	1	1	TP	**Magandang Araw**	1	1	TP
Kumusta ka	1	0	FN	**Kumusta ka**	1	1	TP
Paalam	1	1	TP	**Paalam**	1	1	TP
Ingat ka	1	1	TP	**Ingat ka**	1	1	TP
Maraming Salamat	1	1	TP	**Maraming Salamat**	1	1	TP
Pasensya na	1	1	TP	**Pasensya na**	1	1	TP
Bahay	1	1	TP	**Bahay**	1	1	TP
Pinto	1	1	TP	**Pinto**	1	1	TP
Silid	1	1	TP	**Silid**	1	0	FN
Sala	1	1	TP	**Sala**	1	1	TP
Kailan	1	1	TP	**Kailan**	1	1	TP
Guro	1	1	TP	**Guro**	1	1	TP
Kusina	1	1	TP	**Kusina**	1	1	TP
Dilaw	1	1	TP	**Dilaw**	1	1	TP
Ube	1	1	TP	**Ube**	1	1	TP

	Trial # 5		
Sign Language	**Ground Truth**	**Test Result (Prediction)**	**Verdict**
A	1	1	TP
B	1	1	TP
C	1	1	TP
D	1	1	TP
E	1	1	TP
F	1	1	TP
G	1	1	TP
H	1	1	TP
I	1	1	TP
J	1	1	TP
K	1	1	TP

L	1	1	TP
M	1	0	FN
N	1	1	TP
Ñ	1	1	TP
NG	1	1	TP
O	1	1	TP
P	1	1	TP
Q	1	1	TP
R	1	1	TP
S	1	1	TP
T	1	1	TP
U	1	1	TP
V	1	1	TP
W	1	1	TP
X	1	1	TP
Y	1	1	TP
Z	1	1	TP
Ako si	1	1	TP
Ano ang pangalan mo?	1	1	TP
Ilang taon ka na?	1	1	TP
Sino	1	0	FN
Magandang Umaga	1	1	TP
Magandang Hapon	1	1	TP
Magandang Gabi	1	1	TP
Magandang Araw	1	1	TP
Kumusta ka	1	1	TP
Paalam	1	0	FN
Ingat ka	1	1	TP
Maraming Salamat	1	1	TP
Pasensya na	1	1	TP
Bahay	1	1	TP
Pinto	1	1	TP
Silid	1	1	TP

Sala	1	1	TP
Kailan	1	1	TP
Guro	1	1	TP
Kusina	1	1	TP
Dilaw	1	1	TP
Ube	1	1	TP

Table 19. Total TP, TN, FP and FN of Filipino Sign Language for Expert no. 1

Sign Language	No. of TP	No. of TN	No. of FP	No. of FN
A	4	0	0	1
B	5	0	0	0
C	5	0	0	0
D	5	0	0	0
E	4	1	0	0
F	4	0	0	1
G	4	1	0	0
H	5	0	0	0
I	5	0	0	0
J	4	1	0	0
K	5	0	0	0
L	5	0	0	0
M	4	0	0	1
N	5	0	0	0
Ñ	5	0	0	0
NG	5	0	0	0
O	5	0	0	0
P	4	1	0	0
Q	5	0	0	0
R	5	0	0	0
S	5	0	0	0
T	3	1	0	1
U	4	0	0	1

V	5	0	0	0
W	5	0	0	0
X	4	1	0	0
Y	5	0	0	0
Z	5	0	0	0
Ako si	5	0	0	0
Ano ang pangalan mo?	5	0	0	0
Ilang taon ka na?	5	0	0	0
Sino	4	0	0	1
Magandang Umaga	5	0	0	0
Magandang Hapon	5	0	0	0
Magandang Gabi	5	0	0	0
Magandang Araw	5	0	0	0
Kumusta ka	4	0	0	1
Paalam	4	0	0	1
Ingat ka	5	0	0	0
Maraming Salamat	5	0	0	0
Pasensya na	5	0	0	0
Bahay	5	0	0	0
Pinto	5	0	0	0
Silid	4	0	0	1
Sala	5	0	0	0
Kailan	5	0	0	0
Guro	5	0	0	0
Kusina	5	0	0	0
Dilaw	5	0	0	0
Ube	5	0	0	0
TOTAL	**235**	**6**	**0**	**9**

Table 20. Ground Truth and Testing Table for FSL Recognition of Expert no. 2

	Trial # 1				Trial # 2		
Sign Language	**Ground Truth**	**Test Result (Prediction)**	**Verdict**	**Sign Language**	**Ground Truth**	**Test Result (Prediction)**	**Verdict**
A	1	1	TP	**A**	1	1	TP
B	1	1	TP	**B**	1	1	TP
C	1	1	TP	**C**	1	1	TP
D	1	1	TP	**D**	1	1	TP
E	1	1	TP	**E**	1	1	TP
F	1	0	FN	**F**	1	1	TP
G	1	1	TP	**G**	1	1	TP
H	1	1	TP	**H**	1	1	TP
I	1	1	TP	**I**	1	1	TP
J	0	0	TN	**J**	0	0	TN
K	1	1	TP	**K**	1	1	TP
L	1	1	TP	**L**	1	1	TP
M	1	1	TP	**M**	1	1	TP
N	1	1	TP	**N**	1	1	TP
Ñ	1	1	TP	**Ñ**	1	0	FN
NG	1	0	FN	**NG**	1	1	TP
O	1	1	TP	**O**	1	1	TP
P	1	1	TP	**P**	1	1	TP
Q	1	1	TP	**Q**	1	1	TP
R	1	1	TP	**R**	1	1	TP
S	1	1	TP	**S**	1	0	FN
T	1	1	TP	**T**	1	0	FN
U	1	1	TP	**U**	1	1	TP
V	1	1	TP	**V**	1	1	TP
W	1	1	TP	**W**	1	1	TP
X	1	1	TP	**X**	1	1	TP
Y	1	1	TP	**Y**	1	1	TP
Z	1	1	TP	**Z**	1	1	TP
Ako si	1	1	TP	**Ako si**	1	1	TP

Ano ang pangalan mo?	1	1	TP	**Ano ang pangalan mo?**	1	1	TP
Ilang taon ka na?	0	0	TN	**Ilang taon ka na?**	1	1	TP
Sino	1	0	FN	**Sino**	1	1	TP
Magandang Umaga	1	1	TP	**Magandang Umaga**	1	1	TP
Magandang Hapon	1	0	FN	**Magandang Hapon**	1	1	TP
Magandang Gabi	1	1	TP	**Magandang Gabi**	1	1	TP
Magandang Araw	1	1	TP	**Magandang Araw**	1	1	TP
Kumusta ka	1	1	TP	**Kumusta ka**	1	1	TP
Paalam	1	1	TP	**Paalam**	1	1	TP
Ingat ka	1	1	TP	**Ingat ka**	1	1	TP
Maraming Salamat	1	1	TP	**Maraming Salamat**	1	1	TP
Pasensya na	1	1	TP	**Pasensya na**	1	1	TP
Bahay	1	1	TP	**Bahay**	1	1	TP
Pinto	1	1	TP	**Pinto**	1	1	TP
Silid	1	1	TP	**Silid**	1	1	TP
Sala	1	1	TP	**Sala**	1	1	TP
Kailan	1	0	FN	**Kailan**	1	1	TP
Guro	1	1	TP	**Guro**	1	1	TP
Kusina	1	1	TP	**Kusina**	1	0	FN
Dilaw	1	1	TP	**Dilaw**	1	1	TP
Ube	1	1	TP	**Ube**	1	1	TP

	Trial # 3				**Trial # 4**		
Sign Language	**Ground Truth**	**Test Result (Prediction)**	**Verdict**	**Sign Language**	**Ground Truth**	**Test Result (Prediction)**	**Verdict**
A	0	0	TN	**A**	1	1	TP
B	1	1	TP	**B**	1	1	TP

C	1	1	TP	**C**	1	1	TP
D	1	1	TP	**D**	1	1	TP
E	1	1	TP	**E**	1	1	TP
F	1	1	TP	**F**	1	1	TP
G	1	1	TP	**G**	1	1	TP
H	1	1	TP	**H**	1	1	TP
I	1	1	TP	**I**	1	1	TP
J	1	1	TP	**J**	1	1	TP
K	1	1	TP	**K**	1	1	TP
L	1	1	TP	**L**	1	1	TP
M	1	1	TP	**M**	1	1	TP
N	1	1	TP	**N**	1	1	TP
Ñ	1	1	TP	**Ñ**	1	1	TP
NG	1	1	TP	**NG**	0	0	TN
O	1	1	TP	**O**	1	1	TP
P	1	1	TP	**P**	1	1	TP
Q	1	1	TP	**Q**	1	1	TP
R	1	1	TP	**R**	1	1	TP
S	1	1	TP	**S**	1	1	TP
T	1	1	TP	**T**	1	1	TP
U	1	0	FN	**U**	1	1	TP
V	1	1	TP	**V**	1	1	TP
W	1	1	TP	**W**	1	1	TP
X	1	1	TP	**X**	1	1	TP
Y	1	1	TP	**Y**	1	1	TP
Z	1	1	TP	**Z**	1	0	FN
Ako si	1	1	TP	**Ako si**	1	1	TP
Ano ang pangalan mo?	1	1	TP	**Ano ang pangalan mo?**	1	1	TP
Ilang taon ka na?	1	1	TP	**Ilang taon ka na?**	1	1	TP
Sino	1	1	TP	**Sino**	1	1	TP
Magandang Umaga	1	1	TP	**Magandang Umaga**	1	1	TP

Magandang Hapon	0	0	TN	**Magandang Hapon**	1	1	TP
Magandang Gabi	1	1	TP	**Magandang Gabi**	1	1	TP
Magandang Araw	1	1	TP	**Magandang Araw**	1	1	TP
Kumusta ka	1	1	TP	**Kumusta ka**	1	1	TP
Paalam	1	1	TP	**Paalam**	1	1	TP
Ingat ka	1	1	TP	**Ingat ka**	1	1	TP
Maraming Salamat	1	1	TP	**Maraming Salamat**	1	1	TP
Pasensya na	1	1	TP	**Pasensya na**	1	1	TP
Bahay	1	1	TP	**Bahay**	1	1	TP
Pinto	1	1	TP	**Pinto**	1	1	TP
Silid	1	0	FN	**Silid**	1	1	TP
Sala	1	1	TP	**Sala**	1	1	TP
Kailan	1	1	TP	**Kailan**	1	1	TP
Guro	1	1	TP	**Guro**	1	1	TP
Kusina	1	1	TP	**Kusina**	1	1	TP
Dilaw	1	1	TP	**Dilaw**	1	1	TP
Ube	1	1	TP	**Ube**	1	1	TP

	Trial # 5		
Sign Language	**Ground Truth**	**Test Result (Prediction)**	**Verdict**
A	1	1	TP
B	1	1	TP
C	1	1	TP
D	1	1	TP
E	1	1	TP
F	1	1	TP
G	1	1	TP
H	1	1	TP

I	1	1	TP
J	1	1	TP
K	1	1	TP
L	1	1	TP
M	1	1	TP
N	1	1	TP
Ñ	1	1	TP
NG	1	1	TP
O	1	1	TP
P	1	1	TP
Q	1	1	TP
R	1	0	FN
S	1	1	TP
T	1	1	TP
U	1	1	TP
V	1	1	TP
W	1	1	TP
X	1	1	TP
Y	1	1	TP
Z	1	1	TP
Ako si	1	1	TP
Ano ang pangalan mo?	1	1	TP
Ilang taon ka na?	1	1	TP
Sino	1	1	TP
Magandang Umaga	1	1	TP
Magandang Hapon	1	1	TP
Magandang Gabi	1	1	TP
Magandang Araw	1	1	TP
Kumusta ka	1	1	TP
Paalam	0	0	TN
Ingat ka	1	1	TP
Maraming Salamat	1	1	TP
Pasensya na	1	1	TP

Bahay	1	1	TP
Pinto	1	1	TP
Silid	1	1	TP
Sala	1	1	TP
Kailan	1	1	TP
Guro	1	1	TP
Kusina	1	1	TP
Dilaw	1	1	TP
Ube	1	1	TP

Table 21. Total TP, TN, FP and FN of Filipino Sign Language for Expert no. 2

Sign Language	No. of TP	No. of TN	No. of FP	No. of FN
A	4	1	0	0
B	5	0	0	0
C	5	0	0	0
D	5	0	0	0
E	5	0	0	0
F	4	0	0	1
G	5	0	0	0
H	5	0	0	0
I	5	0	0	0
J	3	2	0	0
K	5	0	0	0
L	5	0	0	0
M	5	0	0	0
N	5	0	0	0
Ñ	4	0	0	1
NG	3	1	0	1
O	5	0	0	0
P	5	0	0	0
Q	5	0	0	0
R	4	0	0	1

S	4	0	0	1
T	4	0	0	1
U	4	0	0	1
V	5	0	0	0
W	5	0	0	0
X	5	0	0	0
Y	5	0	0	0
Z	4	0	0	1
Ako si	5	0	0	0
Ano ang pangalan mo?	5	0	0	0
Ilang taon ka na?	4	1	0	0
Sino	4	0	0	1
Magandang Umaga	5	0	0	0
Magandang Hapon	3	1	0	1
Magandang Gabi	5	0	0	0
Magandang Araw	5	0	0	0
Kumusta ka	5	0	0	0
Paalam	4	1	0	0
Ingat ka	5	0	0	0
Maraming Salamat	5	0	0	0
Pasensya na	5	0	0	0
Bahay	5	0	0	0
Pinto	5	0	0	0
Silid	4	0	0	1
Sala	5	0	0	0
Kailan	4	0	0	1
Guro	5	0	0	0
Kusina	4	0	0	1
Dilaw	5	0	0	0
Ube	5	0	0	0

TOTAL	230	7	0	13

Table 22. Ground Truth and Testing Table for FSL Recognition of Expert no. 3

	Trial # 1				**Trial # 2**		
Sign Language	**Ground Truth**	**Test Result (Prediction)**	**Verdict**	**Sign Language**	**Ground Truth**	**Test Result (Prediction)**	**Verdict**
A	1	1	TP	**A**	1	1	TP
B	1	1	TP	**B**	1	1	TP
C	1	1	TP	**C**	1	1	TP
D	1	1	TP	**D**	1	1	TP
E	1	1	TP	**E**	1	1	TP
F	1	1	TP	**F**	1	1	TP
G	1	1	TP	**G**	1	1	TP
H	1	1	TP	**H**	1	1	TP
I	1	1	TP	**I**	1	1	TP
J	0	0	TN	**J**	1	1	TP
K	1	1	TP	**K**	1	1	TP
L	1	1	TP	**L**	1	1	TP
M	1	1	TP	**M**	1	1	TP
N	1	1	TP	**N**	1	1	TP
Ñ	1	0	FN	**Ñ**	1	0	FN
NG	1	0	FN	**NG**	1	1	TP
O	1	1	TP	**O**	1	1	TP
P	1	1	TP	**P**	1	1	TP
Q	0	0	TN	**Q**	1	1	TP
R	1	1	TP	**R**	1	1	TP
S	1	0	FN	**S**	1	1	TP
T	1	0	FN	**T**	1	1	TP
U	1	1	TP	**U**	1	1	TP
V	1	1	TP	**V**	1	1	TP
W	1	1	TP	**W**	0	0	TN
X	1	1	TP	**X**	1	1	TP
Y	1	1	TP	**Y**	1	1	TP
Z	1	1	TP	**Z**	1	1	TP

Ako si	1	1	TP	**Ako si**	1	1	TP
Ano ang pangalan mo?	1	0	FN	**Ano ang pangalan mo?**	1	1	TP
Ilang taon ka na?	1	1	TP	**Ilang taon ka na?**	1	1	TP
Sino	1	1	TP	**Sino**	1	1	TP
Magandang Umaga	1	1	TP	**Magandang Umaga**	1	1	TP
Magandang Hapon	1	1	TP	**Magandang Hapon**	1	0	FN
Magandang Gabi	0	0	TN	**Magandang Gabi**	1	1	TP
Magandang Araw	1	1	TP	**Magandang Araw**	1	0	FN
Kumusta ka	1	1	TP	**Kumusta ka**	1	1	TP
Paalam	1	1	TP	**Paalam**	1	1	TP
Ingat ka	1	1	TP	**Ingat ka**	1	1	TP
Maraming Salamat	1	1	TP	**Maraming Salamat**	1	1	TP
Pasensya na	1	1	TP	**Pasensya na**	1	1	TP
Bahay	1	1	TP	**Bahay**	1	1	TP
Pinto	1	1	TP	**Pinto**	1	1	TP
Silid	1	1	TP	**Silid**	1	1	TP
Sala	1	1	TP	**Sala**	1	1	TP
Kailan	1	1	TP	**Kailan**	1	1	TP
Guro	1	1	TP	**Guro**	1	1	TP
Kusina	1	1	TP	**Kusina**	1	1	TP
Dilaw	1	1	TP	**Dilaw**	1	1	TP
Ube	1	1	TP	**Ube**	1	1	TP

	Trial # 3				**Trial # 4**		
Sign Language	**Ground Truth**	**Test Result (Prediction)**	**Verdict**	**Sign Language**	**Ground Truth**	**Test Result (Prediction)**	**Verdict**
A	1	1	TP	**A**	1	1	TP

B	1	1	TP	**B**	1	1	TP
C	1	1	TP	**C**	1	1	TP
D	1	1	TP	**D**	0	0	TN
E	1	1	TP	**E**	1	1	TP
F	1	1	TP	**F**	1	0	FN
G	1	1	TP	**G**	1	1	TP
H	1	1	TP	**H**	1	1	TP
I	1	1	TP	**I**	1	1	TP
J	1	1	TP	**J**	1	1	TP
K	1	1	TP	**K**	1	1	TP
L	1	1	TP	**L**	1	1	TP
M	1	1	TP	**M**	1	1	TP
N	1	1	TP	**N**	1	1	TP
Ñ	1	1	TP	**Ñ**	1	1	TP
NG	1	1	TP	**NG**	1	1	TP
O	0	0	TN	**O**	1	1	TP
P	1	1	TP	**P**	1	1	TP
Q	1	1	TP	**Q**	1	1	TP
R	1	1	TP	**R**	1	1	TP
S	1	1	TP	**S**	1	1	TP
T	1	1	TP	**T**	1	1	TP
U	1	1	TP	**U**	1	0	FN
V	1	1	TP	**V**	1	1	TP
W	1	1	TP	**W**	1	1	TP
X	1	1	TP	**X**	1	1	TP
Y	1	1	TP	**Y**	1	1	TP
Z	1	0	FN	**Z**	1	1	TP
Ako si	1	1	TP	**Ako si**	1	1	TP
Ano ang pangalan mo?	1	1	TP	**Ano ang pangalan mo?**	1	1	TP
Ilang taon ka na?	1	1	TP	**Ilang taon ka na?**	1	1	TP
Sino	1	0	FN	**Sino**	1	1	TP

Magandang Umaga	1	1	TP	**Magandang Umaga**	1	1	TP
Magandang Hapon	1	1	TP	**Magandang Hapon**	1	1	TP
Magandang Gabi	1	1	TP	**Magandang Gabi**	1	1	TP
Magandang Araw	1	1	TP	**Magandang Araw**	1	1	TP
Kumusta ka	1	1	TP	**Kumusta ka**	1	1	TP
Paalam	1	1	TP	**Paalam**	1	1	TP
Ingat ka	1	1	TP	**Ingat ka**	1	1	TP
Maraming Salamat	1	1	TP	**Maraming Salamat**	0	0	TN
Pasensya na	1	1	TP	**Pasensya na**	1	1	TP
Bahay	1	1	TP	**Bahay**	1	1	TP
Pinto	1	1	TP	**Pinto**	1	1	TP
Silid	1	1	TP	**Silid**	1	1	TP
Sala	1	1	TP	**Sala**	1	1	TP
Kailan	1	1	TP	**Kailan**	1	0	FN
Guro	1	1	TP	**Guro**	1	1	TP
Kusina	1	1	TP	**Kusina**	1	1	TP
Dilaw	1	1	TP	**Dilaw**	1	1	TP
Ube	1	1	TP	**Ube**	1	1	TP

	Trial # 5		
Sign Language	**Ground Truth**	**Test Result (Prediction)**	**Verdict**
A	1	1	TP
B	1	1	TP
C	1	1	TP
D	1	1	TP
E	1	1	TP
F	1	1	TP
G	1	1	TP

H	1	1	TP
I	1	1	TP
J	1	1	TP
K	1	1	TP
L	1	1	TP
M	1	1	TP
N	1	1	TP
Ñ	1	1	TP
NG	1	1	TP
O	1	1	TP
P	1	1	TP
Q	1	1	TP
R	1	1	TP
S	1	1	TP
T	1	1	TP
U	1	1	TP
V	1	1	TP
W	1	1	TP
X	1	1	TP
Y	1	1	TP
Z	1	1	TP
Ako si	1	1	TP
Ano ang pangalan mo?	0	0	TN
Ilang taon ka na?	1	1	TP
Sino	1	1	TP
Magandang Umaga	1	1	TP
Magandang Hapon	1	0	FN
Magandang Gabi	1	1	TP
Magandang Araw	1	1	TP
Kumusta ka	1	1	TP
Paalam	1	1	TP
Ingat ka	1	1	TP
Maraming Salamat	1	1	TP

Pasensya na	1	1	TP
Bahay	1	1	TP
Pinto	1	1	TP
Silid	1	1	TP
Sala	1	1	TP
Kailan	1	1	TP
Guro	1	1	TP
Kusina	1	1	TP
Dilaw	1	1	TP
Ube	1	1	TP

Table 23. Total TP, TN, FP and FN of Filipino Sign Language for Expert no. 3

	No. of TP	**No. of TN**	**No. of FP**	**No. of FN**
A	5	0	0	0
B	5	0	0	0
C	5	0	0	0
D	4	1	0	0
E	5	0	0	0
F	4	0	0	1
G	5	0	0	0
H	5	0	0	0
I	5	0	0	0
J	4	1	0	0
K	5	0	0	0
L	5	0	0	0
M	5	0	0	0
N	5	0	0	0
Ñ	3	0	0	2
NG	4	0	0	1
O	4	1	0	0
P	5	0	0	0
Q	4	1	0	0

R	5	0	0	0
S	4	0	0	1
T	4	0	0	1
U	4	0	0	1
V	5	0	0	0
W	4	1	0	0
X	5	0	0	0
Y	5	0	0	0
Z	4	0	0	1
Ako si	5	0	0	0
Ano ang pangalan mo?	3	1	0	1
Ilang taon ka na?	5	0	0	0
Sino	4	0	0	1
Magandang Umaga	5	0	0	0
Magandang Hapon	3	0	0	2
Magandang Gabi	4	1	0	0
Magandang Araw	4	0	0	1
Kumusta ka	5	0	0	0
Paalam	5	0	0	0
Ingat ka	5	0	0	0
Maraming Salamat	4	1	0	0
Pasensya na	5	0	0	0
Bahay	5	0	0	0
Pinto	5	0	0	0
Silid	5	0	0	0
Sala	5	0	0	0
Kailan	4	0	0	1
Guro	5	0	0	0
Kusina	5	0	0	0
Dilaw	5	0	0	0

Ube	5	0	0	0
TOTAL	228	8	0	14

Table 24. Overall Total TP, TN, FP and FN of Static Filipino Sign Language

	No. of TP	**No. of TN**	**No. of FP**	**No. of FN**
Expert 1	129	6	0	5
Expert 2	128	4	0	8
Expert 3	127	5	0	8
TOTAL	**384**	**15**	**0**	**21**

Table 25. Overall Total TP, TN, FP and FN of Dynamic Filipino Sign Language

	No. of TP	**No. of TN**	**No. of FP**	**No. of FN**
Expert 1	106	0	0	4
Expert 2	102	3	0	5
Expert 3	101	3	0	6
TOTAL	**309**	**6**	**0**	**15**

Tables 18 to 25 illustrate the results of evaluating the accuracy of the FSL recognition, tested by three (3) PRC certified teachers with years of experience in teaching FSL for Special Education (SPED) classes. Each FSL expert conducted 5 trials for each sign language. The rows list the sign languages tested, while the columns show the ground truth, prediction and the verdict (TP, TN, FP, FN). The results shown in each expert's trial indicate that the recognition of the FSL application performs well, with 693 True Positive (TP) and 21 True Negative (TN) results out of 750 total verdicts, indicating a high rate of correct identification of the FSL. However, with 36 out of 750 total verdicts being False Negative, this indicates that there were FSLs unidentified incorrectly that needs improvement, including letters F, Ñ, NG, T, and U, and phrases "Ilang taon ka na?", "Magandang Hapon", and "Pinto",

Table 26. Confusion Matrix of Static FSL

Sign Language	A	B	C	D	E	F	G	H	I	J	K	L	M	N
A	TP=13	0	0	0	0	0	0	0	0	0	0	0	0	TN=1
B	0	TP=15	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	TP=15	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	TP=14	0	TN=1	0	0	0	0	0	0	0	0
E	0	0	TN=1	0	TP=14	0	0	0	0	0	0	0	0	0
F	0	FN=3	0	0	0	TP=12		0	0	0	0	0	0	0
G	0	0	0	0	0	0	TP=14	TN=1	0	0	0	0	0	0
H	0	0	0	0	0	0	0	TP=15	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	TP=15	0	0	0	0	0
J	TN=1	0	0	0	0	0	0	0	TN=2	TP=11	0	0	0	TN=1
K	0	0	0	0	0	0	0	0	0	0	TP=15	0	0	0
L	0	0	0	0	0	0	0	0	0	0	0	TP=15	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	TP=14	FN=1
N	0	0	0	0	0	0	0	0	0	0	0	0	0	TP=15
ENYE	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NG	0	0	0	0	0	0	TN=1	0	0	0	0	0	0	FN=2
O	0	0	0	TN=1	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	TN=1	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	TN=1	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S	FN=2	0	0	0	0	0	0	0	0	0	0	0	0	0
T	TN=1	0	0	0	0	0	0	0	0	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0
X	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENYE	NG	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	0	0	0	0	0	FN=1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
TP=12	0	0	0	0	0	0	0	0	0	0	0	0	FN=3
0	TP=12	0	0	0	0	0	0	0	0	0	0	0	0
0	0	TP=14	0	0	0	0	0	0	0	0	0	0	0
0	0	0	TP=14	0	0	0	0	0	0	0	0	0	0
0	0	0	0	TP=14	0	0	0	0	0	0	0	0	0
0	0	0	0	0	TP=14	0	0	FN=1	0	0	0	0	0
0	0	0	0	0	0	TP=13	0	0	0	0	0	0	0
0	0	0	0	0	0	0	TP=11	0	0	0	FN=3	0	0
0	0	0	0	0	0	0	0	TP=12	FN=3	0	0	0	0
0	0	0	0	0	0	0	0	0	TP=15	0	0	0	0
0	0	0	0	0	0	0	0	0	TN=1	TP=14	0	0	0
0	0	0	0	0	0	0	TN=1	0	0	0	TP=14	0	0
0	0	0	0	0	0	0	0	0	0	0	0	TP=15	0
FN=2	0	0	0	0	0	0	0	0	0	0	0	0	TP=13

Table 27. Confusion Matrix of Dynamic FSL

Sign Language	Ako si	Ano ang pangalan mo?	Ilang taon ka na?	Sino	Magandang Umaga	Magandang Hapon	Magandang Gabi	Magandang Araw	Kumusta ka	Paalam	Ingat ka	Maraming Salamat	Pasensya na	Bahay
Ako si	TP=15	0	0	0	0	0	0	0	0	0	0	0	0	0
Ano ang pangalan mo?	TN=1	TP=13	TN=1	0	0	0	0	0	0	0	0	0	0	0
Ilang taon ka na?	0	TN=1	TP=14	0	0	0	0	0	0	0	0	0	0	0
Sino	0	0	FN=3	TP=12	0	0	0	0	0	0	0	0	0	0
Magandang Umaga	0	0	0	0	TP=15	0	0	0	0	0	0	0	0	0
Magandang Hapon	0	0	0	0	FN=3	TP=11	0	TN=1	0	0	0	0	0	0
Magandang Gabi	0	0	0	0	0	0	TP=14	TN=1	0	0	0	0	0	0
Magandang Araw	0	0	0	0	0	0	FN=1	TP=14	0	0	0	0	0	0
Kumusta ka	0	0	0	0	0	0	0	0	TP=14	0	FN=1	0	0	0
Paalam	0	0	0	0	0	0	0	0	0	TP=13	FN=1	TN=1	0	0
Ingat ka	0	0	0	0	0	0	0	0	0	0	TP=15	0	0	0
Maraming Salamat	0	0	0	0	0	0	0	0	TN=1	0	0	TP=14	0	0
Pasensya na	0	0	0	0	0	0	0	0	0	0	0	0	TP=15	0
Bahay	0	0	0	0	0	0	0	0	0	0	0	0	0	TP=15
Pinto	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Silid	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sala	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Kailan	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Guro	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Kusina	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Dilaw	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ube	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Pinto	Silid	Sala	Kailan	Guro	Kusina	Dilaw	Ube
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
TP=15	0	0	0	0	0	0	0
FN=2	TP=13	0	0	0	0	0	0
0	0	TP=15	0	0	0	0	0
0	0	FN=2	TP=13	0	0	0	0
0	0	0	0	TP=15	0	0	0
0	FN=1	0	0	0	TP=14	0	0
0	0	0	0	0	0	TP=15	0
0	0	0	0	0	0	0	TP=15

Table 28. Total Confusion Matrix of Static and Dynamic FSL

Correct FSL	**TP= 693**	**FN=36**	**Total Correct FSL= 729**
Incorrect FSL	**FP=0**	**TN=21**	**Total Incorrect FSL= 21**
	Total Correct Output= 693	**Total Incorrect Output= 57**	

The confusion matrix shown on tables 26 and 27 evaluates the performance of the application's static and dynamic FSL recognition. The rows represent the ground truth, while the columns represent the predicted sign language. The result

of the matrix indicates that the FSL recognition of the application has a high-performance rate, showing high True Positive (TP) count for most of the sign languages, which indicates high recognition rate. This can be observed through sign languages like ‘A’ and ‘B’ that have 13 and 15 True Positive (TP) counts respectively, which indicates that these signs have high recognition rate. However, certain sign languages like ‘F’ and ‘Ñ’ that both have 12 True Positive (TP) counts but also 3 False Negative (FN) counts, indicates that the recognition of these sign languages sometimes fails and has room for improvement.

4.2.2. Detection Model Evaluation Metrics

4.2.2.1. Computation of Detection Model Metrics

Accuracy

$$\boldsymbol{Accuracy} = \left(\frac{\boldsymbol{TP + TN}}{\boldsymbol{TP + FN + TN + FP}}\right) \times \mathbf{100\%}$$

$$\boldsymbol{Accuracy\ (Static)} = \left(\frac{\mathbf{384 + 15}}{\mathbf{384 + 21 + 15 + 0}}\right) \times \mathbf{100\%} = \mathbf{95\%}$$

$$\boldsymbol{Accuracy\ (Dynamic)} = \left(\frac{\mathbf{309 + 6}}{\mathbf{309 + 15 + 6 + 0}}\right) \times \mathbf{100\%} = \mathbf{95.45\%}$$

$$\boldsymbol{Accuracy(Total)} = \left(\frac{\mathbf{0.95 + 0.9545}}{\mathbf{2}}\right) \times \mathbf{100\%}$$

$$\boldsymbol{Accuracy(Total)} = \mathbf{95.22\%}$$

After testing the system, the researchers were able to achieve a recognition accuracy of 95.22%. Recognition accuracy of 95% for static and 95.45% for dynamic, with a total recognition accuracy of 95.22% 4. Accuracy represents the proportion of correctly classified instances, and in this case, shows a 95.22% effectiveness in making correct predictions. The accuracy calculated shows that the application was able to obtain a higher recognition percentage than proposed in

specific objective 4.

Precision

$$\mathbf{Pr}\,\boldsymbol{e\,cision} = \left(\frac{\boldsymbol{TP}}{\boldsymbol{TP + FP}}\right) \times \mathbf{100\%}$$

$$\mathbf{Pr}\,\boldsymbol{e\,cision}\,(\boldsymbol{Static}) = \left(\frac{\mathbf{384}}{\mathbf{384 + 0}}\right) \times \mathbf{100\%} = \mathbf{100\%}$$

$$\mathbf{Pr}\,\boldsymbol{e\,cision}\,(\boldsymbol{Dynamic}) = \left(\frac{\mathbf{309}}{\mathbf{309 + 0}}\right) \times \mathbf{100\%} = \mathbf{100\%}$$

$$\mathbf{Pr}\,\boldsymbol{e\,cision}\,(\boldsymbol{Total}) = \left(\frac{\mathbf{1 + 1}}{\mathbf{2}}\right) \times \mathbf{100\%}$$

$$\mathbf{Pr}\,\boldsymbol{e\,cision}\,(\boldsymbol{Total}) = \mathbf{100\%}$$

After testing the system, the researchers were able to achieve a precision rate of 100% for both static and dynamic FSL, with a total recognition precision of 100%. This rate measures the accuracy of the positive predictions made by the model highlighting that the application made 100% correct positive predictions. A high precision score means that the application has no false positive prediction. The researchers can confidently say that the sign language that was signed in the application are all correct. Meaning that the higher the precision is the better [74].

Recall

$$\boldsymbol{Recall} = \left(\frac{\boldsymbol{TP}}{\boldsymbol{TP + FN}}\right) \times \mathbf{100\%}$$

$$\boldsymbol{Recall}\,(\boldsymbol{Static}) = \left(\frac{\mathbf{384}}{\mathbf{384 + 21}}\right) \times \mathbf{100\%} = \mathbf{94.81\%}$$

$$\boldsymbol{Recall}\,(\boldsymbol{Dynamic}) = \left(\frac{\mathbf{309}}{\mathbf{309\ +\ 15}}\right) \times \mathbf{100\%} = \mathbf{95.37\%}$$

$$\boldsymbol{Recall}\,(\boldsymbol{Total}) = \left(\frac{\mathbf{0.9481 + 0.9537}}{\mathbf{2}}\right) \times \mathbf{100\%}$$

$$\boldsymbol{Recall}\,(\boldsymbol{Total}) = \mathbf{95.06\%}$$

After testing the system, the researchers were able to achieve a recall rate of 94.81% for static FSL and 95.37% for dynamic FSL, which attains a total recognition recall rate of 95.06%. This metric measures the proportion of the actual positive instances that are correctly identified by the model. In contrast to precision, recall focuses on the model's ability to capture all positive instances, while precision focuses on positive predictions. Having a high recall value means that there are only a few false negatives that can be found in the application, it filters out all the positive outcome which the study aims to do [75].

Specificity

$$\boldsymbol{Specificity} = \left(\frac{\boldsymbol{TN}}{\boldsymbol{TN + FP}}\right) \times \mathbf{100\%}$$

$$\boldsymbol{Specificity}\ (\boldsymbol{Static}) = \left(\frac{\mathbf{15}}{\mathbf{15 + 0}}\right) \times \mathbf{100\%} = \mathbf{100\%}$$

$$\boldsymbol{Specificity}\ (\boldsymbol{Dynamic}) = \left(\frac{\mathbf{6}}{\mathbf{6 + 0}}\right) \times \mathbf{100\%} = \mathbf{100\%}$$

$$\boldsymbol{Specificity}\ (\boldsymbol{Total}) = \left(\frac{\mathbf{1 + 1}}{\mathbf{2}}\right) \times \mathbf{100\%}$$

$$\boldsymbol{Specificity}\ (\boldsymbol{Total}) = \mathbf{100\%}$$

After testing the system, the researchers were able to achieve a specificity of 100% for both static and dynamic FSL, with a total recognition specificity of 100%. This shows the measure a classification model's ability to correctly identify negative instances. Having a high rate of specificity means that even when signing a different FSL letter or phrase in a different module the application is still able to detect the letter or phrase signed. A high specificity rate means that the application is effective in recognizing the FSL [76].

F1 Score

$$F1 = 2 \times \left(\frac{precision \times recall}{precision + recall}\right)$$

$$F1 = 2 \times \left(\frac{1 \times 0.9509}{1 + 0.9509}\right)$$

$$F1 = 0.974$$

After testing the system, the researchers were able to achieve an F1 Score of 0.974. This value shows the harmonic mean of precision and recall. The high score possibly suggests that the model is accurate in positive predictions and positive instances. To know if F1 score can be considered good it should have a value of 0.7 or higher [77]. Since the F1 score of the application is 0.974 it means that it can be considered as a good and functioning application.

Matthew's Correlation Coefficient

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

$$MCC = \frac{693 \times 21 - 0 \times 36}{\sqrt{(693 + 0)(693 + 36)(21 + 0)(21 + 36)}}$$

$$MCC = 0.592$$

After testing the system, the researchers were able to achieve a Matthew's Correlation Coefficient of 0.592. Since the value of the MCC is a high value near 1 it indicates that it has effective predictions for both correct and incorrect. An MCC score of 0 means that the system is just performing by random chance, while a score of 0.3 is moderately accurate, and a score of 0.5 or above is considered to have a strong performance [78]. Since the application was able to attain an MCC score of 0.592 it can be said that it has a strong performance in recognizing FSL.

4.3. Tables and Discussion of System Reliability

4.3.1. Testing Functionality

Table 29. Table for testing the Camera

Trial No.	Result (1- Correct; 0- Incorrect)
1	1
2	1
3	1
4	1
5	1

$$\%\boldsymbol{R_c} = \left(\frac{\boldsymbol{NS_{camera}}}{\boldsymbol{NT_{camera}}}\right) \times \mathbf{100}$$

$$\%\boldsymbol{R_c} = \left(\frac{\mathbf{5}}{\mathbf{5}}\right) \times \mathbf{100}$$

$$\%\boldsymbol{R_c} = \mathbf{100}\%$$

Table 30. Table for testing the laptop

Trial No.	Result (1- Correct; 0- Incorrect)
1	1
2	1
3	1
4	1
5	1

$$\%\boldsymbol{R_{CP}} = \left(\frac{\boldsymbol{NS_{computer}}}{\boldsymbol{NT_{computer}}}\right) \times \mathbf{100}$$

$$\%\boldsymbol{R_{CP}} = \left(\frac{\mathbf{5}}{\mathbf{5}}\right) \times \mathbf{100}$$

$$\%\boldsymbol{R_{CP}} = \mathbf{100}\%$$

Table 31. Table for testing the Application

Trial No.	Result (1- Correct; 0- Incorrect)
1	1
2	1
3	1

4	1
5	1

$$\%R_A = \left(\frac{NS_{application}}{NT_{application}}\right) \times 100$$

$$\%R_A = \left(\frac{5}{5}\right) \times 100$$

$$\%R_A = 100\%$$

Tables 28, 29, and 30 shows the reliability of the system, it shows that after testing within 5 trials if the camera, laptop, and application are fully functional, the obtained reliability are all 100%, meaning that the whole system is working.

4.4. Discussion of Comparison of Tutoring Methods

4.4.1. Likert Scale

4.4.1.1. Likert Scale Survey on Kaway

Figure 41. Likert-Scale Survey Results on KAWAY Effectiveness

The effectiveness of the KAWAY application is shown on Figure 41 and results showed an overall 80% strong agreement on its effectiveness. All experts unanimously agreed that KAWAY effectively reinforces FSL concepts in practical

scenarios. For teaching basics, 67% strongly agreed and 33% agreed, indicating a need for slight improvements. Confidence in replicating FSL was universally strong, with 100% strongly agreeing. The accuracy of gesture assessment and the helpfulness of real-time feedback received mixed strong agreement (33%) and agreement (67%), suggesting areas for enhancement. The application shows strong potential in its effectiveness, with expert feedback guiding future improvements.

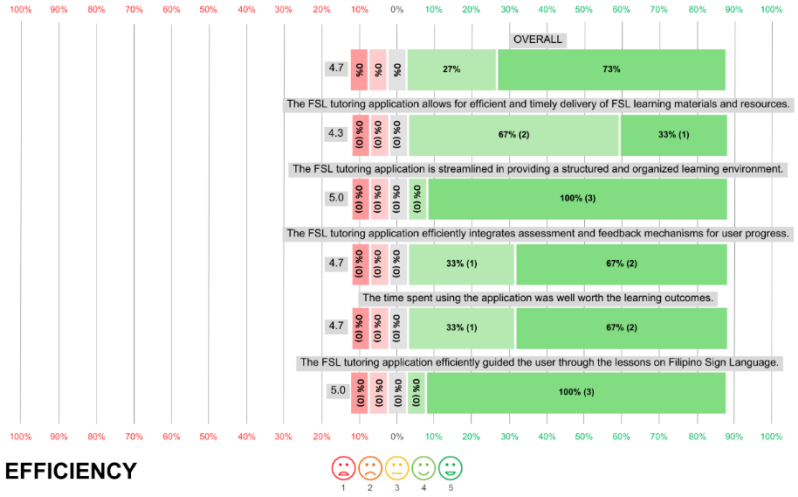


Figure 42. Likert-Scale Survey Results on KAWAY Efficiency

Figure 42 shows the efficiency of the KAWAY FSL tutoring application that was evaluated and yielded a 73% strongly agree overall rating. The application was highly praised for providing a structured and organized learning environment, with 100% of experts strongly agreeing. In delivering FSL materials timely and efficiently, 67% of experts agreed, and 33% strongly agreed, indicating a positive but improvable aspect. The integration of assessment and feedback mechanisms received 67% strong agreement and 33% agreement, while the time spent on the app being worthwhile garnered the same distribution. Moreover, 100% of experts strongly agreed that the application efficiently guided users through FSL lessons.

The application is deemed highly efficient, especially in structuring lessons and guiding users, with minor areas for further enhancement.

Figure 43. Likert-Scale Survey Results on KAWAY Learnability

Figure 43 shows the learnability of the application that was evaluated through a Likert scale survey, achieving a 93% strongly agree overall rating. The experts unanimously agreed (100%) that users easily grasp and understand FSL concepts through the application. Additionally, the clarity and intuitiveness of the learning pathways and structures were highly rated, with 67% strongly agreeing and 33% agreeing. The presentation of lessons was also praised, with 100% of experts strongly agreeing that the lessons were easy to understand and follow, providing sufficient guidance for learning new gestures. Furthermore, all experts strongly agreed that users feel confident in their ability to follow the lessons provided by the application. The application is considered highly learnable, effectively aiding users in understanding and practicing FSL with clear, intuitive, and well-structured lessons.

Figure 44. Likert-Scale Survey Results on KAWAY Memorability

In Figure 44, the memorability of the application was evaluated and its results achieved an overall 80% strongly agree rating. Experts unanimously agreed (100%) that users demonstrate significantly improved learning outcomes with the application. Additionally, 67% of the experts strongly agreed, and 33% agreed that users show notable progress and achievement through the application. The consistency of learning progress was similarly rated, with 67% strongly agreeing and 33% agreeing. Experts also felt that users can easily recall and replicate the FSL gestures taught, with 67% strongly agreeing and 33% agreeing. Moreover, the assessment feedback was highlighted as particularly effective, with 100% of experts strongly agreeing that it aids in reinforcing the memorization of gestures. The application is highly effective in enhancing memorability, helping users to consistently improve, recall, and replicate FSL gestures.

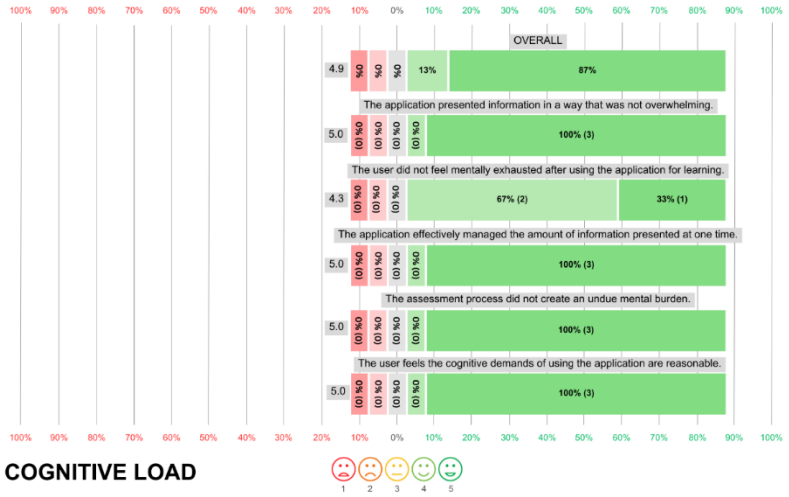


Figure 45. Likert-Scale Survey Results on KAWAY Cognitive Load

The cognitive load of the application was evaluated and achieved an overall 87% strongly agree rating as seen in Figure 45. Experts unanimously agreed (100%) that the application presented information in a way that was not overwhelming, effectively managing the amount of information presented at one time. Additionally, 67% of experts agreed, and 33% strongly agreed that users did not feel mentally exhausted after using the application for learning. All experts (100%) also strongly agreed that the assessment process did not create an undue mental burden and that the cognitive demands of using the application are reasonable. The application is highly effective in managing cognitive load, presenting information clearly and efficiently without overwhelming users or causing mental exhaustion.

Figure 46. Likert-Scale Survey Results on KAWAY Satisfaction

The satisfaction with the application was evaluated and achieved an overall 93% strongly agree rating. Experts unanimously agreed (100%) that users are highly satisfied with the system and functionalities of the application for teaching purposes. Additionally, 67% of experts strongly agreed, and 33% agreed that users are likely to recommend the application to other FSL educators. The overall learning experience provided by the application was highly rated, with 100% of experts strongly agreeing on user satisfaction. Experts also unanimously agreed that users feel confident in their ability to communicate using FSL after using the application and that the application has increased users' interest in learning Filipino Sign Language. The application is highly satisfying to users, effectively supporting teaching, boosting confidence in communication, and sparking greater interest in FSL.

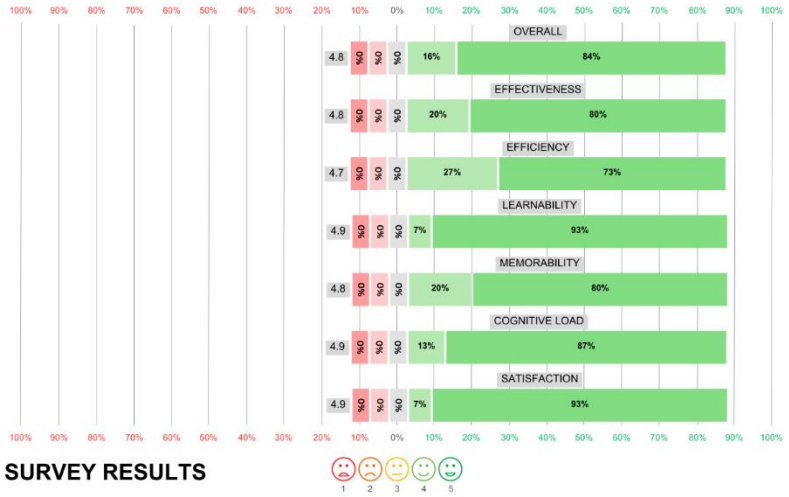


Figure 47. Likert-Scale Survey Results on KAWAY

The Likert scale survey conducted on the KAWAY thesis application revealed an overall satisfaction rating of 84% strongly agree. Across various dimensions, the application received commendable scores. It was deemed 80% effective in achieving educational goals, demonstrating its ability to effectively teach FSL concepts. In terms of efficiency, it garnered a 73% strongly agree, indicating it efficiently delivers learning materials, with potential for further streamlining. Notably, the application excelled in learnability with a high score of 93% strongly agree, reflecting its user-friendly interface and effective teaching methods. Memorability also received 80% strongly agree, suggesting users found it effective in aiding recall of FSL gestures. Managing cognitive load was another strength, rated at 87% strongly agree, indicating users did not feel mentally burdened during use. Moreover, satisfaction was high at 93% strongly agree, underscoring users' overall positive experience and increased interest in learning FSL. While the application showcases strengths in learnability, satisfaction, and cognitive load management, enhancing efficiency and further improving

memorability could bolster its comprehensive effectiveness in teaching Filipino Sign Language.

4.4.1.2. Likert Scale on Traditional Method

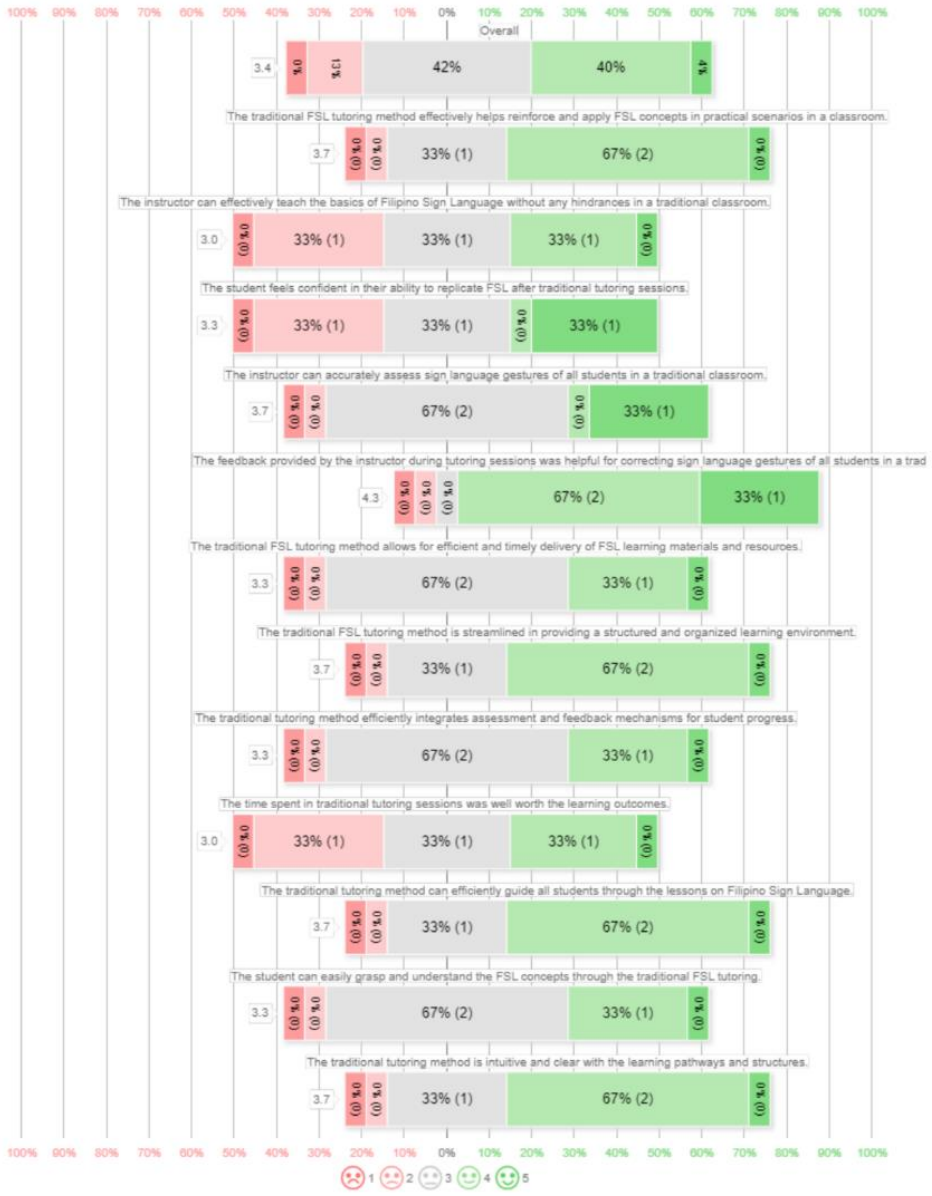


Figure 48. First Part of the Likert Scale Results on Traditional Tutoring

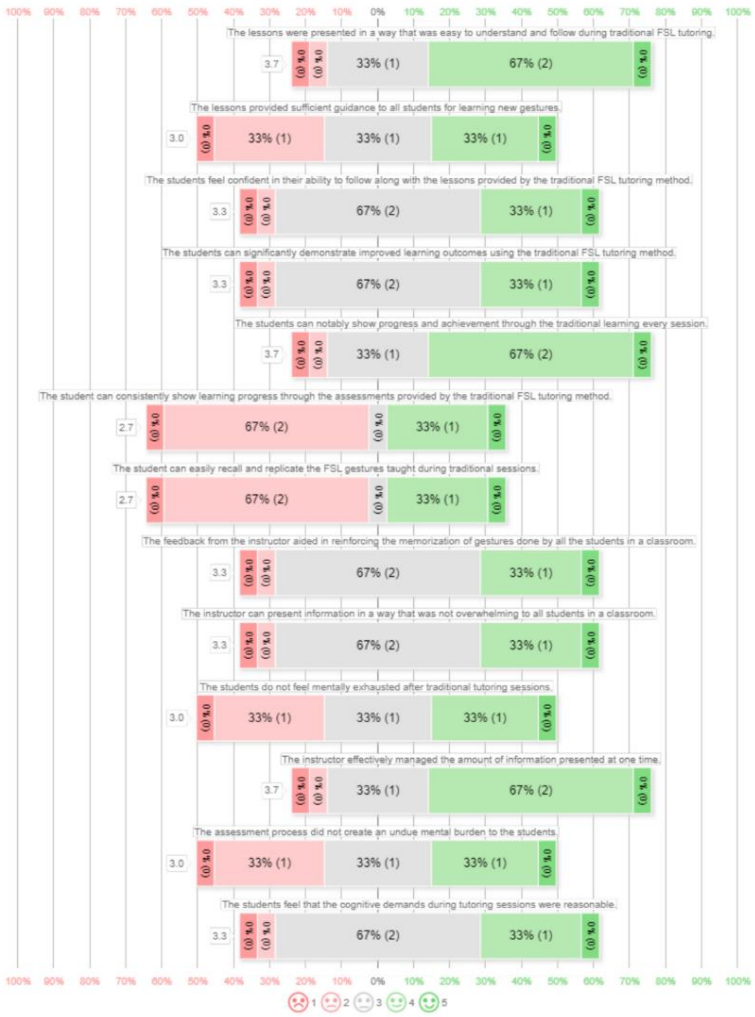


Figure 49. Second Part of the Likert Scale Results on Traditional Tutoring

Figure 50. Third Part of the Likert Scale Results on Traditional Tutoring

As seen in the Figure 48 to Figure 50, the Likert scale survey on the traditional method of teaching FSL, evaluated by three FSL professional experts, reveals a mixed reception. Overall, responses indicate a varied perspective among experts regarding the effectiveness and various aspects of traditional FSL tutoring.

The survey's results show that the traditional FSL tutoring method garnered 0% strongly disagree, 13% disagree, 42% neither agree nor disagree, 40% agree, and 4% strongly agree. This indicates a divergence in opinions, with a significant portion neither strongly agreeing nor disagreeing across different evaluation criteria.

In terms of effectiveness, opinions are split. While 67% of experts agree that the method helps reinforce FSL concepts practically in the classroom, only 33% believe it effectively teaches the basics without hindrances. Similarly, confidence in replicating FSL after sessions and accurate assessment of gestures showed a mixed response, with 33% strongly agreeing in each case and the remainder neither agreeing nor disagreeing.

Regarding the learning environment, 67% of experts agree that the traditional method provides a structured approach, while integration of assessment and feedback mechanisms saw a similar response. However, opinions on learning outcomes, progress, and efficient guidance through lessons were notably positive, with 67% agreeing in each case.

Perceptions on cognitive load management and satisfaction with learning experiences showed neutrality, with 67% neither agreeing nor disagreeing. However, a significant portion (67%) disagreed with the notion that traditional tutoring efficiently guides learning progress through assessments and aids in

gesture memorization.

Overall, while traditional FSL tutoring received mixed feedback, particularly in areas of assessment efficacy and cognitive demands, there is recognition of its strengths in structuring learning environments and fostering student progress and confidence in communication. In contrast, the high satisfaction and effectiveness ratings of the KAWAY thesis application suggest a more advantageous approach to teaching FSL compared to traditional methods.

4.4.2. Hypothesis Testing

Null Hypothesis (H_0): There is no significant difference in the effectiveness of learning Filipino Sign Language between the automatic FSL tutor with static and dynamic FSL recognition using Image Processing and Deep Learning and traditional FSL tutoring methods.

Alternative Hypothesis (H_1): The development of an automatic FSL tutor with static and dynamic FSL recognition using Image processing and Deep Learning is more effective than traditional FSL tutoring methods in teaching Filipino Sign Language.

To determine the significant difference between the two samples collected from the Likert Scale evaluating the Traditional FSL tutoring and the FSL Application, the proponents have tested and compared both values to either reject H_0 or accept H_1. The following are the parameters utilized to test the samples:

1. **H_0**: $p_1 = p_2$
2. **H_1**: $p_1 < p_2$
3. **α:** 0.05

Where p_1 refers to the Traditional FSL Tutoring and p_2 refers to the FSL Application Tutoring; The samples will also be considered as a special experimental condition called paired observations since both observations in a pair have something in common [79]. Through these observations, the data will be tested as Paired Samples t-Test to compare the means of two measurements.

Table 32. Table for Descriptive Statistics

Traditional		***Application***	
Mean	3.355555556	**Mean**	4.844444444
Standard Error	0.08105285819	**Standard Error**	0.03841787164
Standard Deviation	0.7689349282	**Standard Deviation**	0.3644639318
Sample Variance	0.5912609238	**Sample Variance**	0.1328339576
Sum	302	**Sum**	436
Count	90	**Count**	90
Confidence Level (95%)	0.1610502993	**Confidence Level (95%)**	0.076335491

Table 31 provides the descriptive statistics for the two methods for FSL Tutoring, namely, Traditional and with the use of the Kaway Application. The mean score for the Kaway Application FSL Tutoring method is significantly higher than that of the Traditional method suggesting that participants rated the use of the application more favorably on average compared to the Traditional method. The Sample Variance of the Traditional method is also higher which reinforces that answers were more divided and mixed throughout the Likert Scale Survey. This suggests that the participants may have mixed opinions regarding the Traditional

Method unlike the use of the application with more favorable results in its use as a form of tutoring FSL. Since the Standard Deviation and Standard Error for the Application method is lower, this also indicates that the scores are more consistent and that the participants' experiences with its use were more uniformly positive. Overall, the results show an evident perception that the utilization of Kaway Application as method in FSL tutoring is seen more positively than that Traditional method.

Table 33. Table for t-Test: Paired Two Sample for Means

	Traditional	***Application***
Mean	3.355555556	4.844444444
Variance	0.5912609238	0.1328339576
Observations	90	90
t Stat		-15.65102645
P(T<=t) one-tail		0.00000000000000000000000001363
t Critical one-tail		1.662155266

1. **Calculating for the sample mean**

a. $\bar{x} = \left(\frac{x1 + x2 + x3 + \cdots + xn}{n}\right)$

Traditional = 3.355555556

Application = 4.844444444

2. **Calculating for the sample mean difference**

a. $\bar{x}\text{diff} = \bar{x}1 - \bar{x}2$

x_{diff} **= -1.488888889**

3. Calculating for the sample standard deviation

a. $s = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$

***s* = 0.9024865083**

4. Calculating the standard error of the mean difference

a. $SE(d) = \frac{s_d}{\sqrt{n}}$

SE(d) = 0.09513043079

5. Calculating the test statistic

a. $T = \frac{\bar{d}}{SE(d)}$

***T* = -15.65102645**

6. Probability of test statistic

a. Acquiring the P-Value from a t-table or an online statistical software, the proponents have arrived with the value of P(T<=t) = 0.00000000000000000000000001363.

Looking at the values of the mean, FSL Application Tutoring (p_1) equals to 4.8444 while the Traditional Tutoring (p_2) scores a value of 3.3555. The difference between these values suggests that the FSL experts rated the FSL Application Tutoring more favorably than average. The P-value acquired is P(T<=t) = 0.00000000000000000000000001363 which is significantly less than the significance level of 0.05, with a P-value this small strongly indicates that the observed difference in means is statistically significant, leading us to reject the null hypothesis (H_0). This suggests that the FSL experts, the validators, rated the FSL Application significantly higher than the Traditional FSL tutoring method.

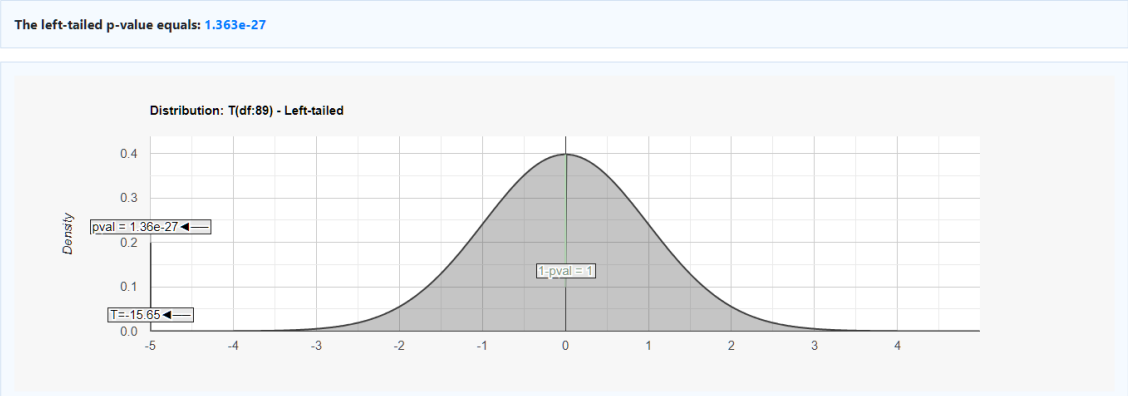


Figure 51. P-Value from Online Statistical Software

Chapter 5

CONCLUSION

With the continuous development of technologies in advancing the fields of education, there must also be an equally important emphasis placed upon improving Special Education in the country as well. These forms of developments allow more underprivileged individuals in learning the skills they need to live comfortably. Based on the objectives provided in the study, the proponents have successfully achieved the goals of the research in developing an automatic Filipino Sign Language Tutoring Application that could effectively aid users in learning sign language with feedback evaluation. This kind of application demonstrates the significant potential of integrating advanced technologies such as deep learning into the field of special education providing enhanced learning tools for students with hearing impairments.

The proponents have developed successfully a dataset of FSL gestures implemented into a deep learning model for gesture recognition. Additionally, dataset for the tutorial videos used within the application's modules were also recorded and utilized before the assessment of the sign language.

The researchers have successfully developed a program using Python that implements the LSTM deep learning model for the recognition of FSL gestures. FSL gesture translation to text was also achieved using the same deep learning model and Python program which also implements a feedback evaluation on whether the signed FSL gesture is correct or incorrect.

The application also features a visually attracting and responsive GUI, and audio interface that helps in user retentiveness. Additionally, the application can load the FSL lessons which is divided into modules. The recognition of the FSL gestured can be recognized by the application as seen in the tables in Chapter 4, together with a feedback evaluation of the signed language.

The application has achieved a statistical improvement from the proposed formulas having an accuracy of 95.2%, a precision of 100%, a recall of 95.06%, a specificity of 100%, an F1 score of 0.974. and an MCC rating of 0.592. The researchers provided tables of the values of TP, TN, FP, and FN for the recognition of FSL gestures along with a confusion matrix based on its recognition.

As compared with the traditional method of tutoring, a Likert Scale survey given out to FSL experts have highlighted the effectiveness of the application in delivering quality education and engagement.

Overall, testing computations and tables can be seen on Chapter 4 proving that the application has a system reliability of 100% with a 95.2% detection accuracy rate and an effective Graphical User Interface.

Chapter 6

RECOMMENDATIONS

The proponents have achieved a tutoring application for the Filipino Sign Language, allowing effective learning of the language based on the results gathered from Chapter 4. To the future researchers willing to further extend the scope of this study, the proponents suggest improving the scalability and deployment of the application. Firstly, optimization and compatibility for most operating systems would allow more students and educators in learning the language. Another option is to deploy Kaway as a web application to remove installations and hardware requirements for each user.

It was also observed that there were tendencies that the detection model confused certain phrases and words. Numbers and other sign languages were not included in the system since the proponents followed the approved curriculum of the experts who tested the system. In line with this, the researchers may also add to the curriculum more gestures to have more lessons in the systems. The researchers can also incorporate a better way to distinguish two sign languages and find a way to utilize a better detection model that is able to recognize certain hand movements.

For the future researchers that want to continue working on this study, instead of just using the LSTM deep learning model to recognize and detect sign language, a CNN model can be added to the LSTM model for a more accurate and efficient recognition of sign language in the system.

Overall, the researchers hope that the study can provide other researchers or readers to have a brief understanding on using LSTM deep learning model for the detection and recognition of the sign language. In addition, the researchers hope that the study can help future research of the field prove that the developed algorithms can be used to recognize sign language with multiple hand movements and that the prototype will be used by the schools that teach Filipino Sign Language in the future.

BIBLIOGRAPHY

[1] J. Rivera and C. Ong, "Facial Expressions Recognition in Filipino Sign Language: Classification using 3D Animation Units," PCSC, 2018.

[2] H. Mangampo, M. Lou, V. Presto, and T. Ramos Herradura, "Senyas: A 3D Animated Filipino Sign Language Interpreter Using Speech Recognition," 2022.

[3] "2000 CENSUS OF POPULATION AND HOUSING NATIONAL STATISTICS OFFICE." Available: https://rsso08.psa.gov.ph/sites/default/files/2000%20Census%20of%20Population%20Rpt%20No.%202_Samar.pdf

[4] A. Wadhawan and P. Kumar, "Deep learning-based sign language recognition system for static signs," *Neural Computing and Applications*, Jan. 2020, doi: https://doi.org/10.1007/s00521-019-04691-y.

[5] "Deafness and hearing loss," *www.who.int*. https://www.who.int/zh/news-room/fact-sheets/detail/deafness-and-hearing-loss.

[6] "Republic Act No. 11106 | GOVPH," *Official Gazette of the Republic of the Philippines*, Oct. 30, 2018. https://www.officialgazette.gov.ph/2018/10/30/republic-act-no-11106/ (accessed Oct. 30, 2023).

[7] Rakesh Kumar Attar, V. Goyal, and L. Goyal, "Open issues and challenges in automatic sign language translation systems," *Nucleation and Atmospheric Aerosols*, Jan. 2022, doi: https://doi.org/10.1063/5.0100903.

[8] J. Henner, R. Novogrodsky, C. Caldwell-Harris, and R. Hoffmeister, "The Development of American Sign Language–Based Analogical Reasoning in Signing Deaf Children," *Journal of Speech, Language, and Hearing Research*, vol. 62, no. 1, pp. 93–105, Jan. 2019, doi: https://doi.org/10.1044/2018_jslhr-l-18-0062.

[9] M. Leo, G. Medioni, M. Trivedi, T. Kanade, and G. M. Farinella, "Computer vision for assistive technologies," *Computer Vision and Image Understanding*, vol. 154, pp. 1–15, Jan. 2017, doi: https://doi.org/10.1016/j.cviu.2016.09.001.

[10] A. J. Abougarair, "Smart glove for sign language translation," *International Robotics & Automation Journal*, vol. 8, no. 3, pp. 109–117, Dec. 2022, doi: https://doi.org/10.15406/iratj.2022.08.00253.

[11] N. B. Ibrahim, H. H. Zayed, and M. M. Selim, "Advances, Challenges and Opportunities in Continuous Sign Language Recognition," *Journal of Engineering and Applied Sciences*, vol. 15, no. 5, pp. 1205–1227, Dec. 2019, doi: https://doi.org/10.36478/jeasci.2020.1205.1227.

[12] A. Wadhawan and P. Kumar, “Deep learning-based sign language recognition system for static signs,” *Neural Computing and Applications*, Jan. 2020, doi: https://doi.org/10.1007/s00521-019-04691-y.

[13] M. Oszust and J. Krupski, “Isolated Sign Language Recognition with Depth Cameras,” *Procedia Computer Science*, vol. 192, pp. 2085–2094, 2021, doi: https://doi.org/10.1016/j.procs.2021.08.216.

[14] T. Raghuveera, R. Deepthi, R. Mangalashri, and R. Akshaya, “A depth-based Indian Sign Language recognition using Microsoft Kinect,” *Sādhanā*, vol. 45, no. 1, Jan. 2020, doi: https://doi.org/10.1007/s12046-019-1250-6.

[15] B. M. Saleh, “D-Talk: Sign Language Recognition System for People with Disability using Machine Learning and Image Processing,” *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 4, pp. 4374–4382, Aug. 2020, doi: https://doi.org/10.30534/ijatcse/2020/29942020.

[16] S. Rajan, R. Nagarajan, A. K. Sahoo, and M. G. Sethupati, “Interpretation and Translation of American Sign Language for Hearing Impaired Individuals using Image Processing,” *International Journal of Recent Technology and Engineering*, vol. 8, no. 4, pp. 415–420, Nov. 2019, doi: https://doi.org/10.35940/ijrte.d6966.118419.

[17] C. Atanga, B. A. Jones, L. E. Krueger, and S. Lu, “Teachers of Students With Learning Disabilities: Assistive Technology Knowledge, Perceptions, Interests, and Barriers,” *Journal of Special Education Technology*, vol. 35, no. 4, p. 016264341986485, Jul. 2019, doi: https://doi.org/10.1177/0162643419864858.

[18] L. Notarte-Balanquit, “Insights from the first filipino sign language (FSL) summit (february 20-21) & the prospects for filipino sign linguistics,” Feb. 2021.

[19] V. Lara-Ortiz, R. Fuentes-Aguilar, and J. I. Chairez, “Assisted Sign Language Tutoring System Using a Bioinspired Virtual Controlled Robotic Arm,” *SSRN Electronic Journal*, 2023, doi: https://doi.org/10.2139/ssrn.4334505.

[20] D. M. Elbourhamy and H. M. Mohammdi, “An intelligent system to help deaf students learn Arabic Sign Language,” *Interactive Learning Environments*, pp. 1–16, Apr. 2021, doi: https://doi.org/10.1080/10494820.2021.1920431.

[21] H. Mangampo, M. Lou, V. Presto, and T. Ramos Herradura, “Senyas: A 3D Animated Filipino Sign Language Interpreter Using Speech Recognition,” 2022.

[22] “Leveraging ICT to achieve Education 2030: UNESCO-Wei Dong Group Funds-in-Trust Project,” *Unesco.org*, 2023. https://unesdoc.unesco.org/ark:/48223/pf0000265598 (accessed Oct. 29, 2023).

[23] United Nations, “Convention on the Rights of Persons with Disabilities (CRPD) | Division for Inclusive Social Development (DISD),” *social.desa.un.org*, 2006. https://social.desa.un.org/issues/disability/crpd/convention-on-the-rights-of-persons-with-disabilities-crpd

[24] United Nations, “The 17 Sustainable Development Goals,” *United Nations*. https://sdgs.un.org/goals

[25] Department of Science and Technologies, “Updated Harmonized National Research and Development Agenda HNRDA-AANR 2022-2028,” *Updated Harmonized National Research and Development Agenda HNRDA-AANR 2022-2028*, Oct. 13, 2021. https://www.pcaarrd.dost.gov.ph/index.php/hnrda-aanr-2022-2028/file (accessed Oct. 29, 2023).

[26] “Republic Act No. 10410 | GOVPH,” *Official Gazette of the Republic of the Philippines*, Mar. 26, 2013. https://www.officialgazette.gov.ph/2013/03/26/republic-act-no-10410/

[27] Official Gazette, “Republic Act No. 10533 | GOVPH,” *Official Gazette of the Republic of the Philippines*, May 15, 2013. https://www.officialgazette.gov.ph/2013/05/15/republic-act-no-10533/ (accessed Oct. 30, 2023).

[28] T. Ritchings, A. Khadragi, and M. Saeb, “An Intelligent Computer-Based System for Sign Language Tutoring,” *Assistive Technology*, vol. 24, no. 4, pp. 299–308, Dec. 2012, doi: https://doi.org/10.1080/10400435.2012.680662.

[29] C.-M. Chiu, S.-W. Chen, Y.-P. Pao, M.-Z. Huang, S.-W. Chan, and Z.-H. Lin, “A smart glove with integrated triboelectric nanogenerator for self-powered gesture recognition and language expression,” *Science and Technology of Advanced Materials*, vol. 20, no. 1, pp. 964–971, Sep. 2019, doi: https://doi.org/10.1080/14686996.2019.1665458.

[30] J. Hou *et al.*, “SignSpeaker,” *The 25th Annual International Conference on Mobile Computing and Networking*, Aug. 2019, doi: https://doi.org/10.1145/3300061.3300117.

[31] D. Jakhar and I. Kaur, “Artificial intelligence, machine learning and deep learning: definitions and differences,” *Clinical and Experimental Dermatology*, vol. 45, no. 1, Jun. 2019, doi: https://doi.org/10.1111/ced.14029.

[32] J. M. Fernández-Batanero, M. Montenegro-Rueda, J. Fernández-Cerero, and I. García-Martínez, “Assistive technology for the inclusion of students with disabilities: a systematic review,” *Educational technology research and development*, vol. 70, no. 5, Jun. 2022, doi: https://doi.org/10.1007/s11423-022-

10127-7.

[33] M. Shamsuzzaman, “Explainable and Interpretable Deep Learning Models,” *Global Journal of Engineering Sciences*, vol. 5, no. 5, Jun. 2020, doi: https://doi.org/10.33552/gjes.2020.05.000621.

[34] N. A. A. Empe, R. C. L. Echon, H. D. A. Vega, P. L. C. Paterno, M. N. Jamis, and E. R. Yabut, “SimboWika: A Mobile and Web Application to Learn Filipino Sign Language for Deaf Students in Elementary Schools,” *2020 IEEE 8th R10 Humanitarian Technology Conference (R10-HTC)*, Dec. 2020, doi: https://doi.org/10.1109/r10-htc49770.2020.9357056.

[35] V. Monga, Y. Li, and Y. C. Eldar, “Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing,” *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, Mar. 2021, doi: https://doi.org/10.1109/msp.2020.3016905.

[36] S. Yuill and H. Halpin, “Python,” 2006.

[37] M. Davydov, Iouri Nikolski, Volodymyr Pasichnyk, Oles Hodych, and Yuriy Shcherbyna, “Providing Feedback in Ukrainian Sign Language Tutoring Software,” *Springer eBooks*, pp. 241–261, Jan. 2013, doi: https://doi.org/10.1007/978-3-642-30341-8_13.

[38] B. Allen, “Natural User Interface Based American Sign Language Tutoring Program,” 2021. Accessed: Oct. 29, 2023. [Online]. Available: https://ideaexchange.uakron.edu/cgi/viewcontent.cgi?article=2682&context=honors_research_projects

[39] F. I. Seman, N. F. Mhd Shariff, and N. I. S. Nasaruddin, “i-Sign: Sign Language Learning Application Via Gamification,” *Asian Journal of University Education*, vol. 15, no. 3, p. 187, Dec. 2019, doi: https://doi.org/10.24191/ajue.v15i3.7569.

[40] W. Mohd, Nur’Ain Najiha Zakaria, Mohd Murtadha Mohamad, W. Yassin, Ahmad, and Ngo Hea Choon, “Sign language detection using convolutional neural network for teaching and learning application,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 28, no. 1, pp. 358–358, Oct. 2022, doi: https://doi.org/10.11591/ijeecs.v28.i1.pp358-364.

[41] K. Amrutha and P. Prabu, “ML Based Sign Language Recognition System,” *IEEE Xplore*, Feb. 01, 2021. https://ieeexplore.ieee.org/document/9399594

[42] J. Joy, K. Balakrishnan, and M. Sreeraj, “SignQuiz: A Quiz Based Tool for Learning Fingerspelled Signs in Indian Sign Language Using ASLR,” *IEEE Access*, vol. 7, pp. 28363–28371, 2019, doi: https://doi.org/10.1109/access.2019.2901863.

[43] R. K. B. Cabigting, C. J. U. Grantoza, L. D. Valiente, and E. D. Dimaunahan, "Jetson Nano-Based Two-Way Communication System with Filipino Sign Language Recognition Using LSTM Deep Learning Model for Able and Deaf-Mute Persons," *ieeexplore.ieee.org*, Apr. 10, 2023. https://ieeexplore.ieee.org/document/10092971 (accessed Oct. 21, 2023).

[44] M. B. D. Jarabese, C. S. Marzan, J. Q. Boado, R. R. M. F. Lopez, L. G. B. Ofiana, and K. J. P. Pilarca, "Sign to Speech Convolutional Neural Network-Based Filipino Sign Language Hand Gesture Recognition System," *IEEE Xplore*, Nov. 01, 2021. https://ieeexplore.ieee.org/abstract/document/9644284 (accessed May 16, 2022).

[45] M. Co, J. K. Balan, J. D. Dolendo, J. C. De Goma, and M. J. C. Samonte, "TEXT2FSL: A Filipino Sign Language Phrase Translator Tool for Deaf and Mute," *Proceedings of the 2020 The 6th International Conference on Frontiers of Educational Technologies*, Jun. 2020, doi: https://doi.org/10.1145/3404709.3404741.

[46] C. Ong, I. Lim, J. Lu, C. Ng, and T. Ong, "Sign-Language Recognition Through Gesture & Movement Analysis (SIGMA)," *Mechatronics and Machine Vision in Practice 3*, pp. 235–245, 2018, doi: https://doi.org/10.1007/978-3-319-76947-9_17.

[47] S. J, "2% Of The Philippines Population Is Deaf – But They Face Many Challenges – ICPHS," *International Congress of Phonetic Sciences ICPhS 2019*, Nov. 05, 2022. https://www.icphs2019.org/2-of-the-philippines-population-is-deaf-but-they-face-many-challenges (accessed Oct. 30, 2023).

[48] ConnectHear, "Everything you need to know about sign language," *ConnectHear*, Aug. 11, 2020. https://www.connecthear.org/post/everything-you-need-to-know-about-sign-language#:~:text=Sign%20language%20is%20a%20non (accessed Oct. 30, 2023).

[49] A. Wahyuni, "The Power of Verbal and Nonverbal Communication in Learning," *www.atlantis-press.com*, 2018. https://www.atlantis-press.com/proceedings/icigr-17/25890842

[50] S. Downs and C. Owen, "Make a Difference TIPS FOR TEACHING STUDENTS WHO ARE DEAF OR HARD OF HEARING -HANDBOOK," 2000. Accessed: Oct. 30, 2023. [Online]. Available: https://www.umaryland.edu/media/umb/oaa/campus-life/disability-services-/documents/Tips-for-Teaching-Students-Who-Are-Deaf-or-Hard-of-Hearing.pdf

[51] J. G. Kyle, "Integration of deaf children," *European Journal of Special Needs Education*, vol. 8, no. 3, pp. 201–220, doi: https://doi.org/10.1080/0885625930080303.

[52] K. Alasim, “Participation and Interaction of Deaf and Hard-of-Hearing Students in Inclusion Classroom,” *INTERNATIONAL JOURNAL OF SPECIAL EDUCATION*, vol. 33, no. 2, 2018, Available: https://files.eric.ed.gov/fulltext/EJ1185582.pdf

[53] “Assistive technology,” *www.who.int*. https://www.who.int/health-topics/assistive-technology#tab=tab_1 (accessed Oct. 30, 2023).

[54] J. Sullivan, “Rethinking Assistive Technology,” *Edutopia*, Feb. 21, 2019. https://www.edutopia.org/article/rethinking-assistive-technology (accessed Oct. 30, 2023).

[55] Mochamad Kamil Budiarto, Triana Rejekiningsih, and Sudiyanto, “Implementation of Computer-assisted Learning in High School: Teachers and Students‟ Perspective,” *International journal of education and management engineering*, vol. 11, no. 4, pp. 26–34, Aug. 2021, doi: https://doi.org/10.5815/ijeme.2021.04.03.

[56] X. Bi and X. Shi, “On the Effects of Computer-assisted Teaching on Learning Results Based on Blended Learning Method,” *International Journal of Emerging Technologies in Learning (iJET)*, vol. 14, no. 01, p. 58, Jan. 2019, doi: https://doi.org/10.3991/ijet.v14i01.9458.

[57] “An Act Declaring The Filipino Sign Language As The National Sign Language Of The Filipino Deaf And The Official Sign Language Of Government In All Transactions Involving The Deaf, And Mandating Its Use In Schools, Broadcast Media, And Workplaces – RA 11106 | National Council on Disability Affairs.” https://ncda.gov.ph/disability-laws/republic-acts/ra-11106/

[58] J. Maturan, “The Usability and Portability of Computer-Aided Instruction (CAI) in teaching elementary students with learning disability in SPED Schools of Quezon City,” *www.academia.edu*, Accessed: Oct. 20, 2023. [Online]. Available: https://www.academia.edu/12560321/The_Usability_and_Portability_of_Computer_Aided_Instruction_CAI_in_teaching_elementary_students_with_learning_disability_in_SPED_Schools_of_Quezon_City

[59] IBM Data and AI Team, “AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What’s the difference?,” *IBM Blog*, Jul. 06, 2023. https://www.ibm.com/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks/

[60] A. Biswal, “Top 10 Deep Learning Algorithms You Should Know in (2023),” *Simplilearn.com*, Aug. 29, 2023. https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-algorithm

[61] A. Chugh, “Deep Learning | Introduction to Long Short Term Memory,” GeeksforGeeks, Jan. 16, 2019. https://www.geeksforgeeks.org/deep-learning-

introduction-to-long-short-term-memory/ A. Ismail, T. Wood, and H. Bravo, 'Improving Long-Horizon Forecasts with Expectation-Biased LSTM Networks', 04 2018.

[62] Moez Krichen, "Convolutional Neural Networks: A Survey," Computers, vol. 12, no. 8, pp. 151–151, Jul. 2023, doi: https://doi.org/10.3390/computers12080151.

[63] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," Artificial Intelligence Review, vol. 53, no. 8, May 2020, doi: https://doi.org/10.1007/s10462-020-09838-1.

[64] K. Taunk, S. De, S. Verma, and A. Swetapadma, "A Brief Review of Nearest Neighbor Algorithm for Learning and Classification," IEEE Xplore, May 01, 2019. https://ieeexplore.ieee.org/document/9065747

[65] D. Kothadiya et al., "Deepsign: Sign language detection and recognition using Deep Learning," Electronics, vol. 11, no. 11, p. 1780, Jun. 2022. doi:10.3390/electronics11111780

[66] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," Artificial Intelligence Review, vol. 53, no. 8, pp. 5929–5955, May 2020. doi:10.1007/s10462-020-09838-1

[67] "Long Short-Term Memory (LSTM) based deep learning models for predicting univariate time series data," International Journal of Machine Learning, vol. 14, no. 1, 2024. doi:10.18178/ijml.2024.14.1.1154

[68] D. Kumari and R. S. Anand, "Isolated video-based sign language recognition using a hybrid CNN-LSTM framework based on attention mechanism," Electronics, vol. 13, no. 7, p. 1229, Mar. 2024. doi:10.3390/electronics13071229

[69] S. Sharma and K. Kumar, "ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks," *Multimedia Tools and Applications*, vol. 80, no. 17, pp. 26319–26331, May 2021, doi: https://doi.org/10.1007/s11042-021-10768-5.

[70] "Republic of the Philippines Department of Education Standards and Competencies for Five-Year-Old Filipino Children K to 12 BASIC EDUCATION CURRICULUM K to 12 Kindergarten Curriculum Guide," 2016. Available: https://www.deped.gov.ph/wp-content/uploads/2019/01/Kinder-CG_0.pdf

[71] M. G. Garcia, C. Ian San Luis, and M. J. C. Samonte, "E-tutor for Filipino Sign Language," *IEEE Xplore*, Aug. 01, 2016. https://ieeexplore.ieee.org/document/7581584

[72] L. R. Chua, M. Ignacio, P. J. Intoy, and R. Lasaleta, "REAL-TIME FILIPINO

SIGN LANGUAGE TRANSLATOR FOR VIDEO CALLING PLATFORM USING VIDEO PROCESSING," Jul. 2021.

[73] P. Weichbroth, "Usability of Mobile Applications: A Systematic Literature Study," IEEE Access, vol. 8, pp. 55563–55577, 2020, doi: https://doi.org/10.1109/access.2020.2981892.

[74] [1]"Accuracy vs. precision vs. recall in machine learning: what's the difference?," www.evidentlyai.com. https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall#what-is-accuracy

[75] T. F. Monaghan et al., "Foundational Statistical Principles in Medical Research: Sensitivity, Specificity, Positive Predictive Value, and Negative Predictive Value," Medicina, vol. 57, no. 5, p. 503, May 2021, doi: https://doi.org/10.3390/medicina57050503.

[76] "F1 Score in Machine Learning," GeeksforGeeks, Dec. 27, 2023. https://www.geeksforgeeks.org/f1-score-in-machine-learning/

[77] I. Logunova, "F1 Score in Machine Learning," Serokell Software Development Company, Jul. 11, 2023. https://serokell.io/blog/a-guide-to-f1-score

[78] "What is Matthews Correlation Coefficient (MCC)," www.activeloop.ai. https://www.activeloop.ai/resources/glossary/matthews-correlation-coefficient-mcc/#:~:text=Generally%2C%20an%20MCC%20score%20closer (accessed Jun. 26, 2024).

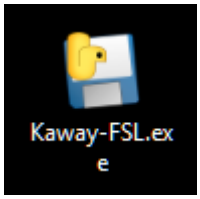
[79] E. R. Walpole, Probability & Statistics for engineers & scientists. 2006. [Online]. Available: https://ndl.ethernet.edu.et/bitstream/123456789/33763/1/Ronald%20E.%20Walpole_2012.pdf

APPENDICES

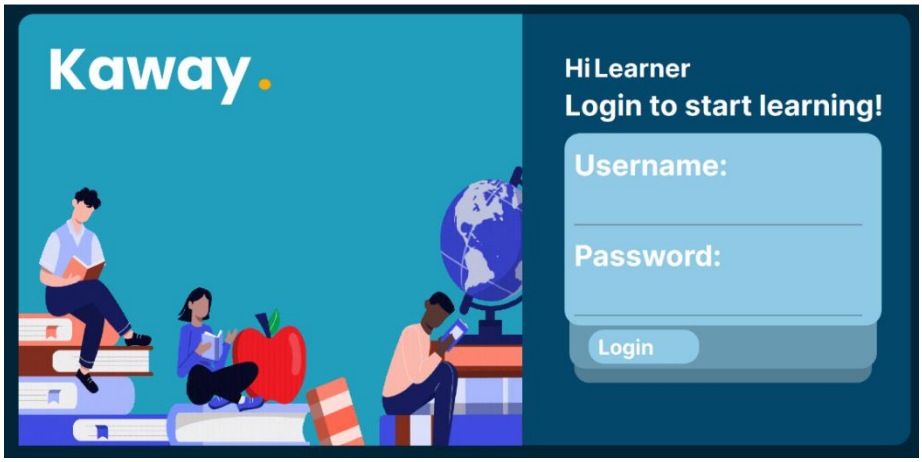
APPENDIX A: USER MANUAL

I. Starting up the Application

1. Open the “Kaway-FSL App”.

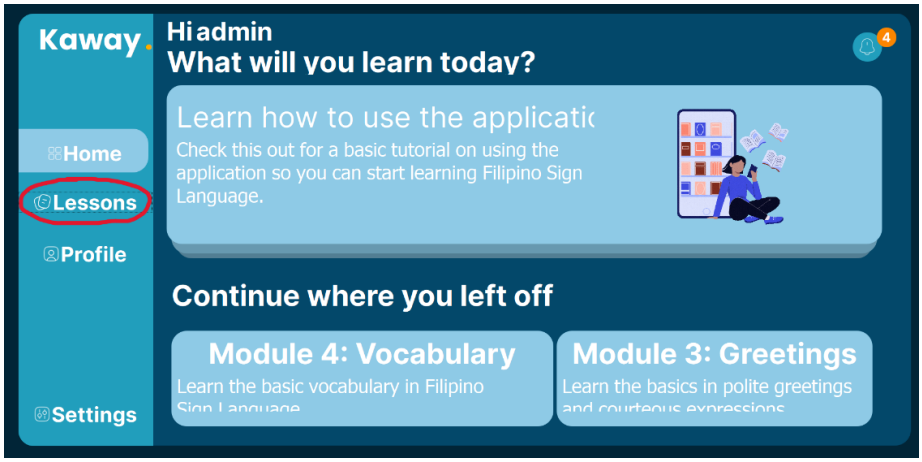


2. Log-in to your account by entering Username and Password.

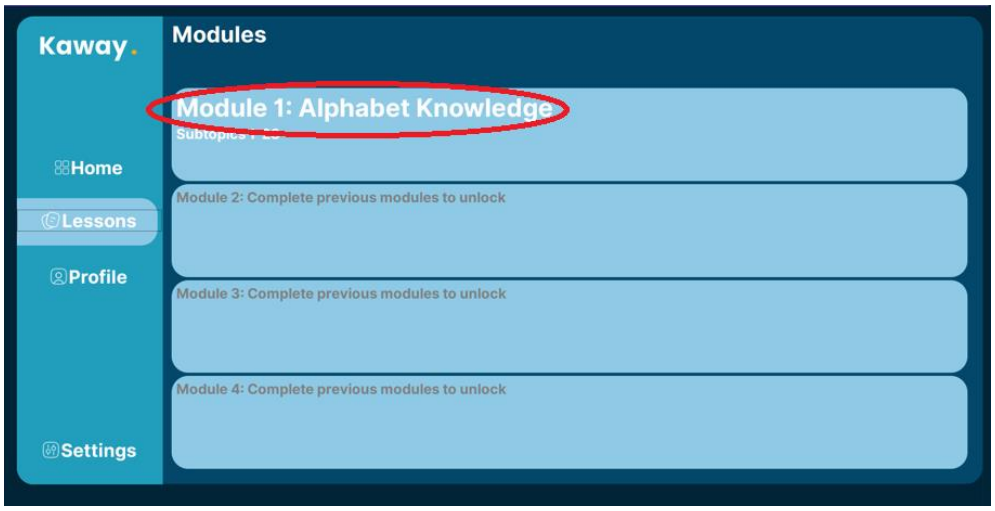


II. Navigating through the Modules

1. Click on the “Lessons” tab.

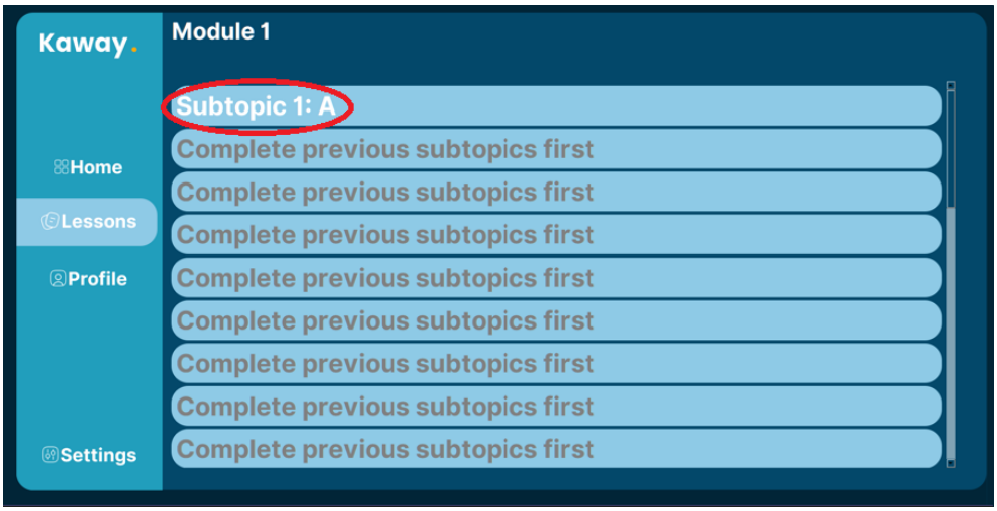


2. Click on the “Module” tab to open the module.



Note: You can only proceed to the next module after completing the first module.

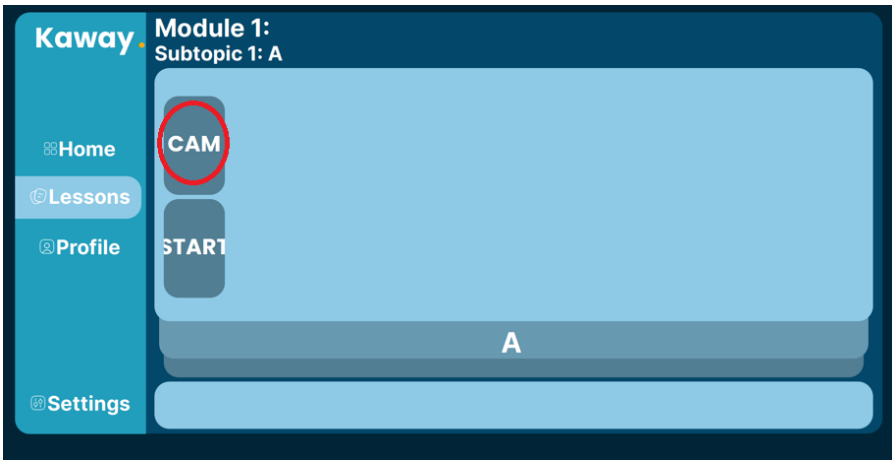
3. In the Module 1 tab, click on the “Subtopic” tab to open the subtopic.



Note: You can only proceed to the next subtopic after completing the first subtopic

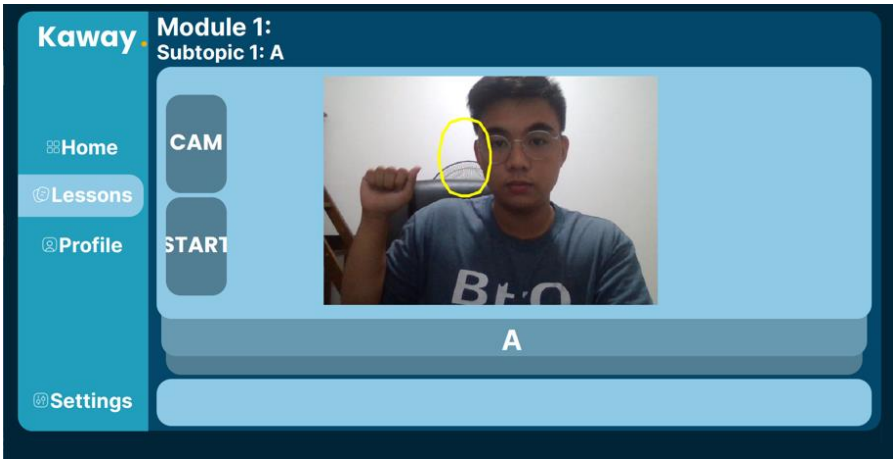
4. Watch the Video tutorial of the FSL, then click “Practice now” to start the test.

5. Click on the “Cam” button to open the camera.



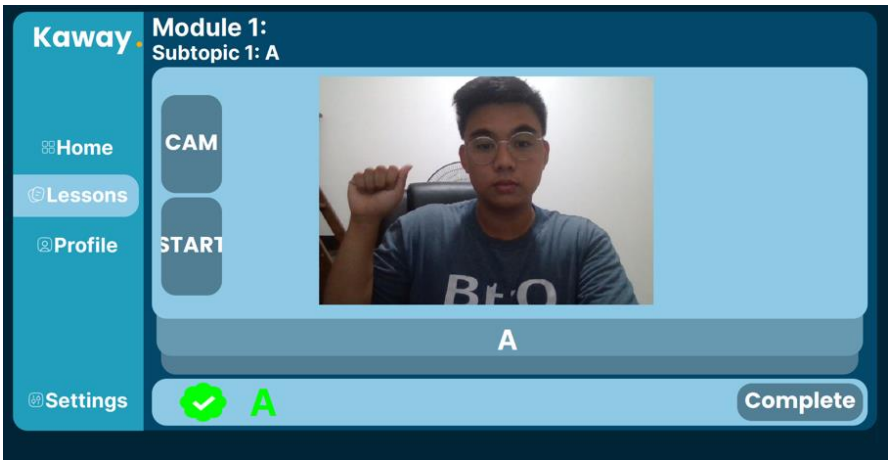
6. Click “Start” to begin the FSL test.

7. Begin doing the sign language after the countdown from 3 to 0.

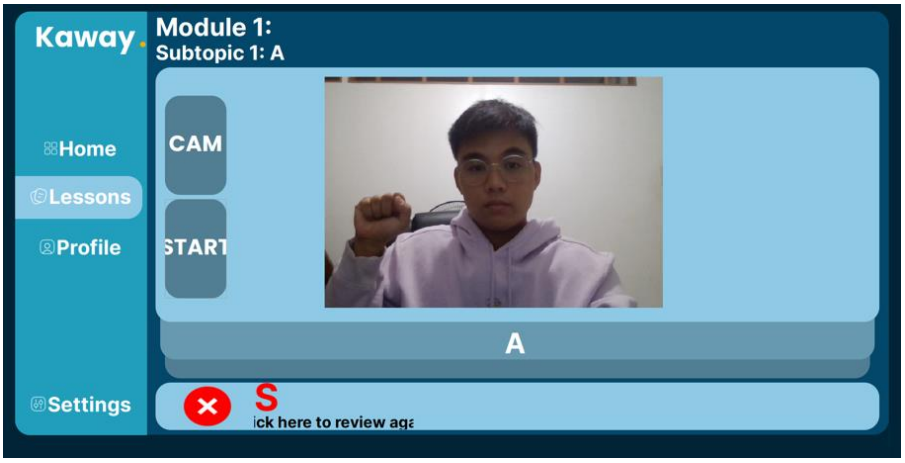


8. After the recognition:

    a. If the sign language is correct, click the “Complete” button to proceed to the next subtopic.

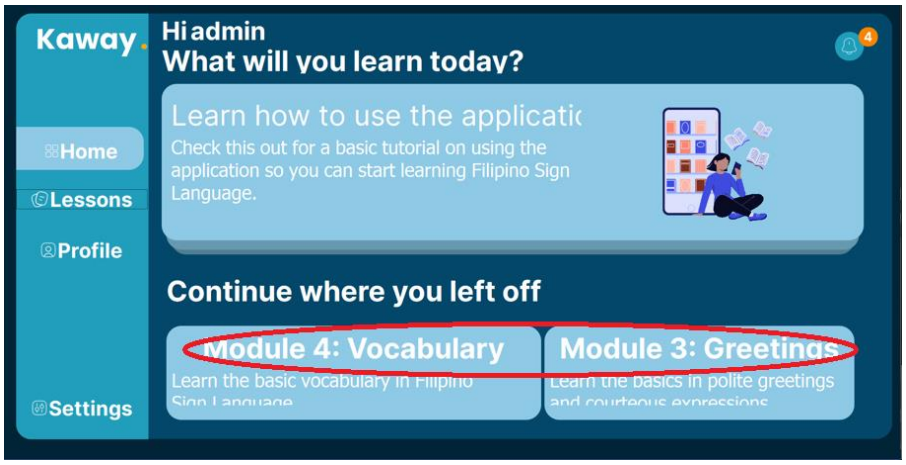


b. If the sign language is incorrect, click start again and repeat the sign language until it is correct.



Note: Refer to the instructions in section 3. for the correct camera postion

9. Repeat steps until you have completed all modules.
10. You can click on the “Where you left off” tab in the Home tab to continue through the modules.



III. **Optimal Camera Positioning**

To ensure the best recognition of Filipino Sign Language, follow the guidelines for camera positioning and setup

1. **Position Yourself in the Center.**

   Make sure that you are positioned in the center of the camera frame. This will allow the system to accurately capture hand movements for the best recognition result.

2. **Show Half of Your Body:**

   Adjust the camera so that it can capture from your head to lower chest.

3. **Ensure Good Lighting:**

   Use adequate lighting to avoid shadows and ensure that your face and hands are well-lit. Position the light in front of you rather than behind to prevent backlighting issues.

4. **Sample Camera Position**

APPENDIX B: SOFTWARE CODE

Python Code for GUI Application

LoginWindow.py

```
# Pyqt import
from PyQt5.QtWidgets import *
from PyQt5 import uic, QtWidgets, QtTest
from PyQt5.QtGui import *
from PyQt5.QtCore import *
from PyQt5.QtMultimedia import *
import sys
import warnings
import os

# initialize files and warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
path = os.getcwd()

# Initialize Classes
class Home(QMainWindow):
    def __init__(self):
        super(Home, self).__init__()
        self.setWindowTitle("Kaway - FSL Learning App")

        # Load the ui
        uic.loadUi("Kaway-GUI/pages/login.ui", self)
        self.setFixedSize(1910, 950)

        self.emailfield = self.findChild(QLineEdit, "username")
        self.passwordfield = self.findChild(QLineEdit, "password")
        self.login = self.findChild(QPushButton, "Login")
```

```
        self.error = self.findChild(QLabel, "error")
        self.logingif = self.findChild(QLabel, 'Check')
        self.loadinggif = self.findChild(QLabel, 'Check_2')

        self.movie = QMovie(r"Kaway-GUI\linear\login.gif")
        self.movieloading = QMovie(r"Kaway-GUI\linear\loading.gif")
        self.logingif.setMovie(self.movie)
        self.movie.start()
        self.loadinggif.hide()
        self.error.hide()


        self.passwordfield.setEchoMode(QtWidgets.QLineEdit.Password)
        self.login.clicked.connect(self.loginfunction)

        # Setup background music
        self.setup_background_music()

    def setup_button_click_sound(self):
        # Create a QSoundEffect object
        self.buttonClickSound = QSoundEffect()

        # Set the sound file
        self.buttonClickSound.setSource(QUrl.fromLocalFile("Kaway-
GUI/audio/button.mp3"))

        # Set volume (0.0 to 1.0)
        self.buttonClickSound.setVolume(1)

        # Connect button click to play sound
        self.login.clicked.connect(self.play_button_click_sound)

    def play_button_click_sound(self):
        self.buttonClickSound.play()

    def setup_background_music(self):
        # Create a QMediaPlayer object
        self.mediaPlayer = QMediaPlayer()

        # Create a QMediaPlaylist object
        self.playlist = QMediaPlaylist()

        # Add a music file to the playlist
        self.playlist.addMedia(QMediaContent(QUrl.fromLocalFile("Kaway-
GUI/audio/music.mp3")))
```

```
        # Set the playlist to loop
        self.playlist.setPlaybackMode(QMediaPlaylist.Loop)

        # Set the playlist to the media player
        self.mediaPlayer.setPlaylist(self.playlist)

        # Set volume (0 to 100)
        self.mediaPlayer.setVolume(100)

        # Play the music
        self.mediaPlayer.play()


    def loginfunction(self):
        user = self.emailfield.text()
        password = self.passwordfield.text()

        if len(user)==0 or len(password)==0:
            print("Please input all fields.")
            self.error.show()
            self.error.setText("Please input all fields.")

        else:
            if user == 'admin' and password == 'admin':
                self.error.setText("Successfully logged in.")
                self.loadinggif.show()
                self.loadinggif.setMovie(self.movieloading)
                self.movieloading.start()
                QtTest.QTest.qWait(3000)


                from home import Home

                print("Button clicked!")
                home = Home(self.widget)
                self.widget.addWidget(home)
                self.widget.setCurrentWidget(home)
            else:
                self.error.show()
                self.error.setText("Invalid username or password")



# initialize the app
app = QApplication(sys.argv)
MainWindowApp = Home()
```

```
MainWindowApp.widget = QStackedWidget()
MainWindowApp.widget.setWindowTitle("Kaway - FSL Learning App")
MainWindowApp.widget.addWidget(MainWindowApp)
MainWindowApp.widget.show()
sys.exit(app.exec_())
```

Home.py

```
# Pyqt import
from PyQt5.QtWidgets import *
from PyQt5 import uic
from PyQt5.QtGui import *
from PyQt5.QtCore import *
import warnings
import os

from db import database

# initialize files and warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
path = os.getcwd()

# Initialize Classes
class Home(QMainWindow):
    def __init__(self, stacked_widget):
        super(Home, self).__init__()
        self.stacked_widget = stacked_widget

        # Load the ui
        uic.loadUi("Kaway-GUI/pages/hometab.ui", self)
        self.setFixedSize(1910, 950)

        # define buttons
        self.lessontabButton = self.findChild(QPushButton, "Lessons")
        self.tutorialButton = self.findChild(QPushButton, "Tutorial")
        self.userLabel = self.findChild(QLabel, "User")
        self.homegif = self.findChild(QLabel, 'GIF')
        self.movie = QMovie(r"Kaway-GUI\linear\home.gif")
        self.homegif.setMovie(self.movie)
        self.movie.start()


        self.FrameTwo = self.findChild(QFrame, 'UpperFrame_5')
        self.ContinueOneButton = self.findChild(QPushButton, "ContinueOne")
        self.ContinueTwoButton = self.findChild(QPushButton, "ContinueTwo")
        self.ContinueOneLabel = self.findChild(QLabel, "SubContinueOne")
        self.ContinueTwoLabel = self.findChild(QLabel, "SubContinueTwo")
```

```
        self.loadModules()

        self.Notif = self.findChild(QFrame, "NotifFrame")
        self.NotifButton = self.findChild(QPushButton, "NotifButton")
        self.NotifNum = self.findChild(QLabel, "NotifNum")
        self.NotifText = self.findChild(QLabel, "NotifText")

        self.Notif.hide()
        database.setValue()
        self.NotifButton.clicked.connect(self.gotoNotif)
        self.updateNotif()



        # Define what buttons do
        self.lessontabButton.clicked.connect(self.gotoLessons)
        self.tutorialButton.clicked.connect(self.gotoTutorial)
        self.ContinueOneButton.clicked.connect(self.gotoContinueOne)
        self.ContinueTwoButton.clicked.connect(self.gotoContinueTwo)

        # Set username to page
        username = database.loadUser()
        self.userLabel.setText(username)

        self.profile = self.findChild(QPushButton, "Profile")
        self.profile.clicked.connect(self.gotoProfile)
        self.settings = self.findChild(QPushButton, "Settings")
        self.settings.clicked.connect(self.gotoSettings)

    def gotoSettings(self):
        from settings import Settings
        print("Button clicked!")
        settings = Settings(self.stacked_widget)
        self.stacked_widget.addWidget(settings)
        self.stacked_widget.setCurrentWidget(settings)

    def gotoProfile(self):
        from profile import Profile
        print("Button clicked!")
        profile = Profile(self.stacked_widget)
        self.stacked_widget.addWidget(profile)
        self.stacked_widget.setCurrentWidget(profile)

    def gotoLessons(self):
        #import functions
        from lessonstab import Lessons
```

```
        print("Button clicked!")
        lessons = Lessons(self.stacked_widget)
        self.stacked_widget.addWidget(lessons)
        self.stacked_widget.setCurrentWidget(lessons)

    def gotoNotif(self):
        notif = database.getCam('notif', 1)
        trigger = database.getCam('trigger', 1)

        if trigger == 'True':
            if notif == 1:
                self.NotifText.setText('You have accomplished 1 module!')
            elif notif == 2:
                self.NotifText.setText('You have accomplished 2 modules!')
            elif notif == 3:
                self.NotifText.setText('You have accomplished 3 modules!')
            elif notif == 4:
                self.NotifText.setText('You have accomplished 4 modules!')
            self.Notif.show()

        else:
            self.Notif.hide()
        database.setValue()

    def updateNotif(self):
        notif = database.getLatestLesson()
        if notif < 29:
            self.NotifNum.setText('1')
            database.updateNotif(1)
        elif notif < 33 and notif > 28:
            self.NotifNum.setText('2')
            database.updateNotif(2)
        elif notif < 42 and notif > 32:
            self.NotifNum.setText('3')
            database.updateNotif(3)
        elif notif > 42:
            self.NotifNum.setText('4')
            database.updateNotif(4)



    def gotoTutorial(self):

        from tutorial import Modules
        modules = Modules(self.stacked_widget)
```

```
        self.stacked_widget.addWidget(modules)
        self.stacked_widget.setCurrentWidget(modules)

    def loadModules(self):
        lessons = database.getLatestLesson()
        if lessons < 29:
            self.FrameTwo.hide()
        elif lessons > 28 and lessons < 33:
            self.FrameTwo.show()
            self.ContinueOneButton.setText('Module 2: Introduction')
            self.ContinueOneLabel.setText('Learn the basics in proper expressions
in introducing oneself')
            self.ContinueTwoButton.setText('Module 1: Alphabets')
            self.ContinueTwoLabel.setText('Learn how the basic alphabets are done
in Filipino Sign Language')

        elif lessons > 32 and lessons < 42:
            self.FrameTwo.show()
            self.ContinueOneButton.setText('Module 3: Greetings')
            self.ContinueOneLabel.setText('Learn the basics in polite greetings
and courteous expressions')
            self.ContinueTwoButton.setText('Module 2: Introduction')
            self.ContinueTwoLabel.setText('Learn the basics in proper expressions
in introducing oneself')

        elif lessons > 41:
            self.FrameTwo.show()
            self.ContinueOneButton.setText('Module 4: Vocabulary')
            self.ContinueOneLabel.setText('Learn the basic vocabulary in Filipino
Sign Language')
            self.ContinueTwoButton.setText('Module 3: Greetings')
            self.ContinueTwoLabel.setText('Learn the basics in polite greetings
and courteous expressions')

    def gotoContinueOne(self):
        lessons = database.getLatestLesson()
        if lessons < 29:
            from lessonsAlphabet import LessonsAlphabet
            lessonsalphabet = LessonsAlphabet(self.stacked_widget)
            self.stacked_widget.addWidget(lessonsalphabet)
            self.stacked_widget.setCurrentWidget(lessonsalphabet)
        elif lessons > 28 and lessons < 33:
            from introduction import Introduction
            introduction = Introduction(self.stacked_widget)
            self.stacked_widget.addWidget(introduction)
            self.stacked_widget.setCurrentWidget(introduction)
```

```
        elif lessons > 32 and lessons < 42:
            from greetings import Greetings
            greetings = Greetings(self.stacked_widget)
            self.stacked_widget.addWidget(greetings)
            self.stacked_widget.setCurrentWidget(greetings)
        elif lessons > 41:
            from vocab import Vocab
            vocab = Vocab(self.stacked_widget)
            self.stacked_widget.addWidget(vocab)
            self.stacked_widget.setCurrentWidget(vocab)

    def gotoContinueTwo(self):
        lessons = database.getLatestLesson()
        if lessons > 28 and lessons < 33:
            from lessonsAlphabet import LessonsAlphabet
            lessonsalphabet = LessonsAlphabet(self.stacked_widget)
            self.stacked_widget.addWidget(lessonsalphabet)
            self.stacked_widget.setCurrentWidget(lessonsalphabet)
        elif lessons > 32 and lessons < 42:
            from introduction import Introduction
            introduction = Introduction(self.stacked_widget)
            self.stacked_widget.addWidget(introduction)
            self.stacked_widget.setCurrentWidget(introduction)
        elif lessons > 41:
            from greetings import Greetings
            greetings = Greetings(self.stacked_widget)
            self.stacked_widget.addWidget(greetings)
            self.stacked_widget.setCurrentWidget(greetings)
```

Lessonstab.py

```
# Pyqt import
from PyQt5.QtWidgets import *
from PyQt5 import uic
from PyQt5.QtGui import *
from PyQt5.QtCore import *
import warnings
import os

#import functions
from lessonsAlphabet import LessonsAlphabet
from db import database


# initialize files and warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
path = os.getcwd()
```

```
class Lessons(QWidget):
    def __init__(self, stacked_widget):
        super(Lessons, self).__init__()
        self.stacked_widget = stacked_widget

        # Load the ui
        uic.loadUi("Kaway-GUI/pages/lessonstab.ui", self)
        self.setFixedSize(1910, 950)

        # define buttons
        self.lessonAlphabetButton = self.findChild(QPushButton, "ModuleOne")
        self.homeButton = self.findChild(QPushButton, "Home")
        self.moduleOne = self.findChild(QPushButton, 'ModuleOne')
        self.moduleTwo = self.findChild(QPushButton, 'ModuleTwo')
        self.moduleThree = self.findChild(QPushButton, 'ModuleThree')
        self.moduleFour = self.findChild(QPushButton, 'ModuleFour')
        self.subOne = self.findChild(QLabel, 'Subtopic1')
        self.subTwo = self.findChild(QLabel, 'Subtopic2')
        self.subThree = self.findChild(QLabel, 'Subtopic3')
        self.subFour = self.findChild(QLabel, 'Subtopic4')

        # Define what buttons do
        self.lessonAlphabetButton.clicked.connect(self.gotoLessonsAlphabet)
        self.moduleTwo.clicked.connect(self.gotoIntroduction)
        self.moduleThree.clicked.connect(self.gotoGreetings)
        self.moduleFour.clicked.connect(self.gotoVocab)
        self.homeButton.clicked.connect(self.gotoHome)


        self.hideModule(database.getLatestLesson())


    # define side tab buttons
    def gotoLessonsAlphabet(self):
        lessonsalphabet = LessonsAlphabet(self.stacked_widget)
        self.stacked_widget.addWidget(lessonsalphabet)
        self.stacked_widget.setCurrentWidget(lessonsalphabet)

    def gotoIntroduction(self):
        from introduction import Introduction
        introduction = Introduction(self.stacked_widget)
        self.stacked_widget.addWidget(introduction)
        self.stacked_widget.setCurrentWidget(introduction)

    def gotoGreetings(self):
```

```
        from greetings import Greetings
        greetings = Greetings(self.stacked_widget)
        self.stacked_widget.addWidget(greetings)
        self.stacked_widget.setCurrentWidget(greetings)

    def gotoVocab(self):
        from vocab import Vocab
        vocab = Vocab(self.stacked_widget)
        self.stacked_widget.addWidget(vocab)
        self.stacked_widget.setCurrentWidget(vocab)

    def gotoHome(self):
        from home import Home
        home = Home(self.stacked_widget)
        self.stacked_widget.addWidget(home)
        self.stacked_widget.setCurrentWidget(home)

    def gotoLessons(self):
        from lessonstab import Lessons
        lessons = Lessons(self.stacked_widget)
        self.stacked_widget.addWidget(lessons)
        self.stacked_widget.setCurrentWidget(lessons)

    def hideModule(self, lesson):
        if lesson < 29:
            self.moduleTwo.hide()
            self.moduleThree.hide()
            self.moduleFour.hide()

            self.subTwo.setText("Module 2: Complete previous modules to unlock")
            self.subTwo.setStyleSheet('color: rgb(128, 128, 128)')
            self.subThree.setText("Module 3: Complete previous modules to unlock")
            self.subThree.setStyleSheet('color: rgb(128, 128, 128)')
            self.subFour.setText("Module 4: Complete previous modules to unlock")
            self.subFour.setStyleSheet('color: rgb(128, 128, 128)')

        elif lesson > 28 and lesson < 33:
            self.moduleThree.hide()
            self.moduleFour.hide()

            self.subThree.setText("Module 3: Complete previous modules to unlock")
            self.subThree.setStyleSheet('color: rgb(128, 128, 128)')
            self.subFour.setText("Module 4: Complete previous modules to unlock")
            self.subFour.setStyleSheet('color: rgb(128, 128, 128)')

        elif lesson > 32 and lesson < 42:
```

```
            self.moduleFour.hide()

            self.subFour.setText("Module 4: Complete previous modules to unlock")
            self.subFour.setStyleSheet('color: rgb(128, 128, 128)')

        elif lesson > 41:
            return
```

lessonsAlphabet.py

```
# Pyqt import
from PyQt5.QtWidgets import *
from PyQt5 import uic
from PyQt5.QtGui import *
from PyQt5.QtCore import *
from functools import partial

from db import database



class LessonsAlphabet(QWidget):
    def __init__(self, stacked_widget):
        super(LessonsAlphabet, self).__init__()
        self.stacked_widget = stacked_widget

        # Load the ui
        uic.loadUi("Kaway-GUI/pages/lessons_alphabet_tab.ui", self)

        lessonName = ''

        # define a-z buttons
        self.aButton = self.findChild(QPushButton, "A")
        self.bButton = self.findChild(QPushButton, "B")
        self.cButton = self.findChild(QPushButton, "C")
        self.dButton = self.findChild(QPushButton, "D")
        self.eButton = self.findChild(QPushButton, "E")
        self.fButton = self.findChild(QPushButton, "F")
        self.gButton = self.findChild(QPushButton, "G")
        self.hButton = self.findChild(QPushButton, "H")
        self.iButton = self.findChild(QPushButton, "I")
        self.jButton = self.findChild(QPushButton, "J")
```

```
        self.kButton = self.findChild(QPushButton, "K")
        self.lButton = self.findChild(QPushButton, "L")
        self.mButton = self.findChild(QPushButton, "M")
        self.nButton = self.findChild(QPushButton, "N")
        self.enyeButton = self.findChild(QPushButton, "ENYE")
        self.ngButton = self.findChild(QPushButton, "NG")
        self.oButton = self.findChild(QPushButton, "O")
        self.pButton = self.findChild(QPushButton, "P")
        self.qButton = self.findChild(QPushButton, "Q")
        self.rButton = self.findChild(QPushButton, "R")
        self.sButton = self.findChild(QPushButton, "S")
        self.tButton = self.findChild(QPushButton, "T")
        self.uButton = self.findChild(QPushButton, "U")
        self.vButton = self.findChild(QPushButton, "V")
        self.wButton = self.findChild(QPushButton, "W")
        self.xButton = self.findChild(QPushButton, "X")
        self.yButton = self.findChild(QPushButton, "Y")
        self.zButton = self.findChild(QPushButton, "Z")

        # Define side buttons
        self.homeButton = self.findChild(QPushButton, "Home")
        self.homeButton.clicked.connect(self.gotoHome)
        self.lessontabButton = self.findChild(QPushButton, "Lessons")
        self.lessontabButton.clicked.connect(self.gotoLessons)

        # Define what buttons do
        self.aButton.clicked.connect(partial(self.gotoAssessment, 'A'))
        self.bButton.clicked.connect(partial(self.gotoAssessment, 'B'))
        self.cButton.clicked.connect(partial(self.gotoAssessment, 'C'))
        self.dButton.clicked.connect(partial(self.gotoAssessment, 'D'))
        self.eButton.clicked.connect(partial(self.gotoAssessment, 'E'))
        self.fButton.clicked.connect(partial(self.gotoAssessment, 'F'))
        self.gButton.clicked.connect(partial(self.gotoAssessment, 'G'))
        self.hButton.clicked.connect(partial(self.gotoAssessment, 'H'))
        self.iButton.clicked.connect(partial(self.gotoAssessment, 'I'))
        self.jButton.clicked.connect(partial(self.gotoAssessment, 'J'))
        self.kButton.clicked.connect(partial(self.gotoAssessment, 'K'))
        self.lButton.clicked.connect(partial(self.gotoAssessment, 'L'))
        self.mButton.clicked.connect(partial(self.gotoAssessment, 'M'))
        self.nButton.clicked.connect(partial(self.gotoAssessment, 'N'))
        self.enyeButton.clicked.connect(partial(self.gotoAssessment, 'Ñ'))
        self.ngButton.clicked.connect(partial(self.gotoAssessment, 'NG'))
        self.oButton.clicked.connect(partial(self.gotoAssessment, 'O'))
        self.pButton.clicked.connect(partial(self.gotoAssessment, 'P'))
        self.qButton.clicked.connect(partial(self.gotoAssessment, 'Q'))
        self.rButton.clicked.connect(partial(self.gotoAssessment, 'R'))
```

```
        self.sButton.clicked.connect(partial(self.gotoAssessment, 'S'))
        self.tButton.clicked.connect(partial(self.gotoAssessment, 'T'))
        self.uButton.clicked.connect(partial(self.gotoAssessment, 'U'))
        self.vButton.clicked.connect(partial(self.gotoAssessment, 'V'))
        self.wButton.clicked.connect(partial(self.gotoAssessment, 'W'))
        self.xButton.clicked.connect(partial(self.gotoAssessment, 'X'))
        self.yButton.clicked.connect(partial(self.gotoAssessment, 'Y'))
        self.zButton.clicked.connect(partial(self.gotoAssessment, 'Z'))

        self.hideSubtopic(database.getLatestLesson())

    def gotoHome(self):
        from home import Home
        print("Button clicked!")
        home = Home(self.stacked_widget)
        self.stacked_widget.addWidget(home)
        self.stacked_widget.setCurrentWidget(home)

    def gotoLessons(self):
        #import functions
        from lessonstab import Lessons

        print("Button clicked!")
        lessons = Lessons(self.stacked_widget)
        self.stacked_widget.addWidget(lessons)
        self.stacked_widget.setCurrentWidget(lessons)

    def gotoAssessment(self, value):
        from modules import Modules
        LessonsAlphabet.lessonName = value
        print(LessonsAlphabet.lessonName)
        database.putChosenLesson(value)

        modules = Modules(self.stacked_widget)
        self.stacked_widget.addWidget(modules)
        self.stacked_widget.setCurrentWidget(modules)

    def hideSubtopic(self, latest):
        buttons = [self.aButton, self.bButton, self.cButton, self.dButton,
self.eButton, self.fButton, self.gButton, self.hButton,
                   self.iButton, self.jButton, self.kButton, self.lButton,
self.mButton, self.nButton, self.enyeButton, self.ngButton,
                   self.oButton, self.pButton, self.qButton, self.rButton,
self.sButton, self.tButton, self.uButton, self.vButton,
                   self.wButton, self.xButton, self.yButton, self.zButton]
```

```
        for index, button in enumerate(buttons):
            if index < latest:
                button.setEnabled(True)
            else:
                button.setEnabled(False)
                button.setText("Complete previous subtopics first")
                button.setStyleSheet('color: rgb(128, 128, 128)')
```

Assessments_static.py

```
# Pyqt import
from PyQt5.QtWidgets import *
from PyQt5 import uic
from PyQt5.QtGui import *
from PyQt5.QtCore import *
import sys
import warnings
import os
import cv2
from functools import partial

#Detection req import
import cv2
import numpy as np
import os
from matplotlib import pyplot as plt
import time
import datetime
import mediapipe as mp
from keras.models import Sequential
from keras.layers import LSTM, Dense
from keras.callbacks import TensorBoard
from sklearn.model_selection import train_test_split
from keras.utils import to_categorical
from scipy import stats
import pickle

from db import database

#initialize mediapipe
model_dict = pickle.load(open('./data_letters/model.p', 'rb'))
model = model_dict['model']
startDetection = 0
threadCamera = False

mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils
```

```
mp_drawing_styles = mp.solutions.drawing_styles

hands = mp_hands.Hands(static_image_mode=True, min_detection_confidence=0.3)

labels_dict = {0: 'A', 1: 'B', 2: 'C', 3: 'D', 4: 'E', 5: 'F', 6: 'G', 7: 'H', 8:
'I', 9: 'J', 10: 'K', 11: 'L', 12: 'M', 13: 'N', 14: 'Ñ', 15: 'O', 16: 'P', 17:
'Q', 18: 'R', 19: 'S', 20: 'T', 21: 'U', 22: 'V', 23: 'W', 24: 'X', 25: 'Y', 26:
'Z'}

class UI(QMainWindow):
    def __init__(self, stacked_widget):
        super(UI, self).__init__()
        self.stacked_widget = stacked_widget
        # Load the ui
        uic.loadUi("Kaway-GUI\pages\Assessments.ui", self)
        # Define the widgets here
        self.cameraButton = self.findChild(QPushButton, "StartCamera")
        self.cameraFrame = self.findChild(QLabel, "CameraFrame")
        self.detectionButton = self.findChild(QPushButton, "StartDetection")
        self.nextModuleButton = self.findChild(QPushButton, 'NextModule')
        self.answerLogo = self.findChild(QLabel, 'Check')
        self.reviewButton = self.findChild(QPushButton, 'ReviewButton')
        self.cameraFrame = self.findChild(QLabel, "CameraFrame")
        self.Error = self.findChild(QFrame, "Error")
        self.ErrorText = self.findChild(QLabel, "ErrorText")

        # Define what widgets do
        self.cameraButton.clicked.connect(self.startCameraGUI)
        self.detectionButton.clicked.connect(self.startTimer)
        self.answerLogo.hide()
        self.nextModuleButton.hide()
        self.reviewButton.hide()
        self.nextModuleButton.clicked.connect(self.gotoLessonsAlphabet)
        self.Error.hide()

        # Instance variable for capturing camera frames
        self.cap = None
        self.Detection = Detection()
        self.Detection.CameraFrame.connect(self.UpdateFrame)
        self.Detection.error.connect(self.CameraError)

        self.Detection.start()

        # change page details
        lesson = self.getLesson()
        self.rightAnswer = self.findChild(QLabel, 'UserAnswer')
```

```
        self.rightAnswer.hide()
        self.Detection.LabelTextChanged.connect(self.updateLabelText)
        self.Detection.CheckAnswer.connect(self.checkAnswer)
        self.Detection.loading.connect(self.loading)
        self.reviewButton.clicked.connect(self.goReview)


        # Define labels
        self.moduleLabel = self.findChild(QLabel, "Module")
        self.subtopicLabel = self.findChild(QLabel, "subtopic")
        self.answerText = self.findChild(QLabel, 'AnswerText')


        # Rename labels
        self.subtopicLabel.setText(database.getValue('module',
database.findRowIDValue('right_answer', lesson)))
        self.answerText.setText(lesson)

        # Define side buttons
        self.homeButton = self.findChild(QPushButton, "Home")
        self.homeButton.clicked.connect(self.gotoHome)
        self.lessontabButton = self.findChild(QPushButton, "Lessons")
        self.lessontabButton.clicked.connect(self.gotoLessons)


    def goReview(self):
        from modules import Modules

        self.Detection.stopCamera()
        modules = Modules(self.stacked_widget)
        self.stacked_widget.addWidget(modules)
        self.stacked_widget.setCurrentWidget(modules)

    def loading(self, bool):
        # Loading the GIF
        self.movie = QMovie(r"Kaway-GUI\linear\loading.gif")
        self.answerLogo.setMovie(self.movie)
        self.movie.start()
        self.answerLogo.show()
        self.rightAnswer.show()
        self.reviewButton.hide()
        self.rightAnswer.setStyleSheet('color: rgb(0, 255, 0)')
        self.rightAnswer.setText("Detecting")
        self.answerLogo.setMovie(self.movie)
        self.movie.start()
```

```
    def updateLabelText(self, text):
        self.rightAnswer.setText(text)

    def CameraError(self):
        self.answerLogo.hide()
        self.rightAnswer.hide()
        self.ErrorText.setText('Use only one hand')
        self.Error.show()

    def checkAnswer(self, bool):
        if not bool:
            self.rightAnswer.setStyleSheet('color: rgb(255, 0, 0)')
            self.answerLogo.setPixmap(QPixmap.fromImage(QImage("Kaway-
GUI\linear\cross.png")))
            self.rightAnswer.show()
            self.answerLogo.show()
            self.reviewButton.show()
        else:
            self.rightAnswer.setStyleSheet('color: rgb(0, 255, 0)')
            self.answerLogo.setPixmap(QPixmap.fromImage(QImage("Kaway-
GUI\linear\check.png")))
            self.nextModuleButton.show()
            self.rightAnswer.show()
            self.answerLogo.show()
            self.reviewButton.hide()

    def startCameraGUI(self):
        self.Error.hide()
        self.Detection.startCamera()

    def UpdateFrame(self, img):
        self.cameraFrame.setPixmap(QPixmap.fromImage(img))

    def startTimer(self):
        if not self.Detection.cap or not self.Detection.cap.isOpened():
            self.Error.show()
            self.ErrorText.setText("Error: Open Camera First")
            return
        else:
            self.Detection.startTimer()

    def getLesson(self):
        from lessonsAlphabet import LessonsAlphabet
        lesson = LessonsAlphabet.lessonName
        return lesson
```

```
    def gotoHome(self):
        from home import Home
        print("Button clicked!")
        self.Detection.stopCamera()
        home = Home(self.stacked_widget)
        self.stacked_widget.addWidget(home)
        self.stacked_widget.setCurrentWidget(home)

    def gotoLessons(self):
        #import functions
        from lessonstab import Lessons


        self.Detection.stopCamera()
        lessons = Lessons(self.stacked_widget)
        self.stacked_widget.addWidget(lessons)
        self.stacked_widget.setCurrentWidget(lessons)

    def gotoLessonsAlphabet(self):
        if database.findRowIDValue('right_answer', database.getChosenLesson()) ==
database.getLatestLesson():
            database.updateLatest()

        self.Detection.stopCamera()
        from lessonsAlphabet import LessonsAlphabet
        lessonsalphabet = LessonsAlphabet(self.stacked_widget)
        self.stacked_widget.addWidget(lessonsalphabet)
        self.stacked_widget.setCurrentWidget(lessonsalphabet)




class Detection(QThread):
    # Initialize Class UI
    CameraFrame = pyqtSignal(QImage)
    LabelTextChanged = pyqtSignal(str)
    CheckAnswer = pyqtSignal(bool)
    loading = pyqtSignal(bool)
    error = pyqtSignal(bool)
    global threadCamera
    threadCamera = False

    def __init__(self):
        super().__init__()
        self.cap = None
```

```
    def startCamera(self):
        self.cap = cv2.VideoCapture(database.getCam('camera', 1))  # Open the
camera (0 for integrated camera, check device list to confirm)
        if not self.cap.isOpened():
            print("Error: Couldn't open camera.")
            return

        # Set camera resolution (adjust as needed)
        # self.cap.set(cv2.CAP_PROP_FRAME_WIDTH, 1280)
        # self.cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 720)

        # Set up timer to read frames and update GUI
        self.timer = QTimer(self)
        self.timer.timeout.connect(self.run)
        global threadCamera
        threadCamera = True
        self.timer.start(1000 // 20)  # Read frames every 33 ms (30 fps)

    def stopCamera(self):
        if self.cap and self.cap.isOpened():
            self.cap.release()
        self.timer.stop()

    def startTimer(self):
        TIMER = int(3)
        prev = time.time()
        while TIMER >= 0:
            ret, image = self.cap.read()
            # Display countdown on each frame
            # specify the font and draw the
            # countdown using puttext
            font = cv2.FONT_HERSHEY_SIMPLEX
            cv2.putText(image, str(TIMER),
                        (200, 250), font,
                        7, (0, 255, 255),
                        4, cv2.LINE_AA)
            if ret:
                # Resize frame
                # image = cv2.resize(image, (960, 540))  # Adjust the dimensions
as needed
                # Convert frame to RGB format
                rgbImage = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

                # Convert RGB image to QImage
                h, w, ch = rgbImage.shape
                bytesPerLine = ch * w
```

```
                qImg = QImage(rgbImage.data, w, h, bytesPerLine,
QImage.Format_RGB888)
                # Convert QImage to QPixmap to display in QLabel
                pixmap = qImg.scaled(960, 540, aspectRatioMode=Qt.KeepAspectRatio)
                self.CameraFrame.emit(pixmap)
            cv2.waitKey(125)

            # current time
            cur = time.time()

            # Update and keep track of Countdown
            # if time elapsed is one second
            # then decrease the counter
            if cur-prev >= 1:
                prev = cur
                TIMER = TIMER-1
                global startDetection
                startDetection = 1

        self.loading.emit(True)



    def getLesson(self):
        from lessonsAlphabet import LessonsAlphabet
        lesson = LessonsAlphabet.lessonName
        return lesson

    def run(self):
        answer = []
        answer_character = []

        try:
            if threadCamera == True:
                while self.cap and self.cap.isOpened():
                    # Read feed
                    ret, frame = self.cap.read()

                    # Show to screen and wait for key to be pressed
                    if ret:
                        # Resize frame
                        # image = cv2.resize(image, (960, 540))  # Adjust the
dimensions as needed
                        # Convert frame to RGB format
                        rgbImage = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
```

```
                        # Convert RGB image to QImage
                        h, w, ch = rgbImage.shape
                        bytesPerLine = ch * w
                        qImg = QImage(rgbImage.data, w, h, bytesPerLine,
QImage.Format_RGB888)
                        # Convert QImage to QPixmap to display in QLabel
                        pixmap = qImg.scaled(960, 540,
aspectRatioMode=Qt.KeepAspectRatio)
                        self.CameraFrame.emit(pixmap)
                    k = cv2.waitKey(125)

                    data_aux = []
                    x_ = []
                    y_ = []
                    count = 0

                    global startDetection
                    if startDetection == 1:
                        ret, frame = self.cap.read()
                        H, W, _ = frame.shape

                        frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

                        results = hands.process(frame_rgb)
                        if results.multi_hand_landmarks:
                            # for hand_landmarks in results.multi_hand_landmarks:
                            #     mp_drawing.draw_landmarks(
                            #         frame,  # image to draw
                            #         hand_landmarks,  # model output
                            #         mp_hands.HAND_CONNECTIONS,  # hand
connections
                            #         mp_drawing_styles.get_default_hand_landmarks
_style(),
                            #         mp_drawing_styles.get_default_hand_connectio
ns_style())

                            for hand_landmarks in results.multi_hand_landmarks:
                                for i in range(len(hand_landmarks.landmark)):
                                    x = hand_landmarks.landmark[i].x
                                    y = hand_landmarks.landmark[i].y

                                    x_.append(x)
                                    y_.append(y)

                                for i in range(len(hand_landmarks.landmark)):
                                    x = hand_landmarks.landmark[i].x
```

```
                                    y = hand_landmarks.landmark[i].y
                                    data_aux.append(x - min(x_))
                                    data_aux.append(y - min(y_))

                                x1 = int(min(x_) * W) - 10
                                y1 = int(min(y_) * H) - 10

                                x2 = int(max(x_) * W) - 10
                                y2 = int(max(y_) * H) - 10

                                prediction = model.predict([np.asarray(data_aux)])

                                predicted_character = labels_dict[int(prediction[0])]

                                # cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 0),
4)
                                # cv2.putText(frame, predicted_character, (x1, y1 -
10), cv2.FONT_HERSHEY_SIMPLEX, 1.3, (0, 0, 0), 3,
                                #             cv2.LINE_AA)

                                answer.append(prediction)
                                answer_character.append(predicted_character)
                                count+=1
                                print(answer[-1])

                                if len(answer) == 15:

                                    if answer[3] == answer[9]:
                                        print(answer_character[9])
                                        # Emit the character string
                                        self.LabelTextChanged.emit(answer_character[9]
)


                                        if self.getLesson() == answer_character[9]:
                                            self.CheckAnswer.emit(True)
                                        else:
                                            self.LabelTextChanged.emit(answer_characte
r[9])

                                            self.CheckAnswer.emit(False)
                                    elif answer[3] == '13' and answer[9] == '6':
                                        print('NG')
                                        # Emit the 'NG' string
                                        self.LabelTextChanged.emit('NG')

                                        if self.getLesson() == 'NG':
                                            self.CheckAnswer.emit(True)
```

```
                                    else:
                                        self.CheckAnswer.emit(False)

                                elif answer[12] == '14':
                                    print('Ñ')
                                    self.LabelTextChanged.emit('Ñ')

                                    if self.getLesson() == 'Ñ':
                                        self.CheckAnswer.emit(True)
                                    else:
                                        self.CheckAnswer.emit(False)

                                elif answer[12] == '9':
                                    print('Ñ')
                                    self.LabelTextChanged.emit('J')

                                    if self.getLesson() == 'J':
                                        self.CheckAnswer.emit(True)
                                    else:
                                        self.CheckAnswer.emit(False)

                                elif answer[12] == '26':
                                    print('Ñ')
                                    self.LabelTextChanged.emit('Z')

                                    if self.getLesson() == 'Z':
                                        self.CheckAnswer.emit(True)
                                    else:
                                        self.CheckAnswer.emit(False)
                                else:
                                    self.LabelTextChanged.emit('Can not detect')
                                    self.CheckAnswer.emit(False)



                            if len(answer) == 15:
                                print("get out")
                                startDetection = 0
                                answer = []
                                answer_character = []


                            if ret:
                                # Resize frame
                                # image = cv2.resize(image, (960, 540))  # Adjust
the dimensions as needed
```

```
                                # Convert frame to RGB format
                                rgbImage = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)


                                # Convert RGB image to QImage
                                h, w, ch = rgbImage.shape
                                bytesPerLine = ch * w
                                qImg = QImage(rgbImage.data, w, h, bytesPerLine,
QImage.Format_RGB888)
                                # Convert QImage to QPixmap to display in QLabel
                                pixmap = qImg.scaled(960, 540,
aspectRatioMode=Qt.KeepAspectRatio)
                                self.CameraFrame.emit(pixmap)
                            cv2.waitKey(1)

                # Release the video capture and destroy OpenCV windows
                if self.cap and self.cap.isOpened():
                    self.cap.release()

        except ValueError:
            print('two hands')
            self.stopCamera()
            self.error.emit(True)
```

Assessments.py

```
# Pyqt import
from PyQt5.QtWidgets import *
from PyQt5 import uic
from PyQt5.QtGui import *
from PyQt5.QtCore import *
import sys
import warnings
import os
import cv2
from functools import partial

#Detection req import
import cv2
import numpy as np
import os
from matplotlib import pyplot as plt
import time
import datetime
import mediapipe as mp
from keras.models import Sequential
from keras.layers import LSTM, Dense
```

```
from keras.callbacks import TensorBoard
from sklearn.model_selection import train_test_split
from keras.utils import to_categorical
from scipy import stats
from db import database
import csv

#initialize mediapipe
mp_holistic = mp.solutions.holistic # Holistic model
mp_drawing = mp.solutions.drawing_utils # Drawing utilities

class UI(QMainWindow):
    def __init__(self, stacked_widget):
        super(UI, self).__init__()
        self.stacked_widget = stacked_widget

        # Load the ui
        uic.loadUi("Kaway-GUI\pages\Assessments.ui", self)

        # Define the widgets here
        self.cameraButton = self.findChild(QPushButton, "StartCamera")
        self.cameraFrame = self.findChild(QLabel, "CameraFrame")
        self.detectionButton = self.findChild(QPushButton, "StartDetection")
        self.nextModuleButton = self.findChild(QPushButton, 'NextModule')
        self.answerLogo = self.findChild(QLabel, 'Check')
        self.reviewButton = self.findChild(QPushButton, 'ReviewButton')
        self.cameraFrame = self.findChild(QLabel, "CameraFrame")
        self.Error = self.findChild(QFrame, "Error")
        self.ErrorText = self.findChild(QLabel, "ErrorText")

        # Define what widgets do
        self.cameraButton.clicked.connect(self.startCameraGUI)
        self.detectionButton.clicked.connect(self.startTimer)
        self.answerLogo.hide()
        self.nextModuleButton.hide()
        self.reviewButton.hide()
        self.nextModuleButton.clicked.connect(self.gotoSubtopics)
        self.Error.hide()

        # Initialize Detection
        self.initDetection()

        # Instance variable for capturing camera frames
        self.cap = None
        self.Detection = Detection()
        self.Detection.CameraFrame.connect(self.UpdateFrame)
```

```
        self.Detection.start()

        # change page details
        lesson = database.getChosenLesson()
        self.rightAnswer = self.findChild(QLabel, 'UserAnswer')
        self.rightAnswer.hide()
        self.Detection.LabelTextChanged.connect(self.updateLabelText)
        self.Detection.CheckAnswer.connect(self.checkAnswer)
        self.Detection.loading.connect(self.loading)
        self.reviewButton.clicked.connect(self.goReview)

        # Define labels
        self.moduleLabel = self.findChild(QLabel, "Module")
        self.subtopicLabel = self.findChild(QLabel, "subtopic")
        self.answerText = self.findChild(QLabel, 'AnswerText')

        # Rename labels
        self.subtopicLabel.setText(database.getValue('module',
database.findRowIDValue('right_answer', lesson)))
        self.answerText.setText(lesson)
        self.changeModule()


        # Define side buttons
        self.homeButton = self.findChild(QPushButton, "Home")
        self.homeButton.clicked.connect(self.gotoHome)
        self.lessontabButton = self.findChild(QPushButton, "Lessons")
        self.lessontabButton.clicked.connect(self.gotoLessons)

        #check module
        moduleCheck = False

    def initDetection(self):
        self.Detection = Detection()
        self.Detection.CameraFrame.connect(self.UpdateFrame)
        self.Detection.LabelTextChanged.connect(self.updateLabelText)
        self.Detection.CheckAnswer.connect(self.checkAnswer)
        self.Detection.loading.connect(self.loading)
        self.Detection.start()

    def closeEvent(self, event):
        self.Detection.stopCamera()
        self.Detection.quit()
        self.Detection.wait()
        event.accept()
```

```
    def goReview(self):
        from modules_two import Modules

        self.Detection.stopCamera()
        modules = Modules(self.stacked_widget)
        self.stacked_widget.addWidget(modules)
        self.stacked_widget.setCurrentWidget(modules)

    def loading(self, bool):
        # Loading the GIF
        self.movie = QMovie(r"Kaway-GUI\linear\loading.gif")
        self.answerLogo.setMovie(self.movie)
        self.movie.start()
        self.answerLogo.show()
        self.rightAnswer.show()
        self.reviewButton.hide()
        self.rightAnswer.setStyleSheet('color: rgb(0, 255, 0)')
        self.rightAnswer.setText("Detecting")



    def updateLabelText(self, text):
        self.rightAnswer.setText(text)

    def checkAnswer(self, bool):
        if bool == False:
            self.rightAnswer.setStyleSheet('color: rgb(255, 0, 0)')
            self.answerLogo.setPixmap(QPixmap.fromImage(QImage("Kaway-
GUI\linear\cross.png")))
            self.rightAnswer.show()
            self.answerLogo.show()
            self.reviewButton.show()
        else:
            self.rightAnswer.setStyleSheet('color: rgb(0, 255, 0)')
            self.answerLogo.setPixmap(QPixmap.fromImage(QImage("Kaway-
GUI\linear\check.png")))
            self.nextModuleButton.show()
            self.rightAnswer.show()
            self.answerLogo.show()
            self.reviewButton.hide()

    def startCameraGUI(self):
        self.Error.hide()
        self.Detection.startCamera()
```

```
    def UpdateFrame(self, img):
        self.cameraFrame.setPixmap(QPixmap.fromImage(img))

    def startTimer(self):
        if not self.Detection.cap or not self.Detection.cap.isOpened():
            self.Error.show()
            self.ErrorText.setText("Error: Open Camera First")
            return
        else:
            self.Detection.startTimer()

    def gotoHome(self):
        from home import Home
        print("Button clicked!")
        self.Detection.stopCamera()
        home = Home(self.stacked_widget)
        self.stacked_widget.addWidget(home)
        self.stacked_widget.setCurrentWidget(home)

    def gotoLessons(self):
        #import functions
        from lessonstab import Lessons
        self.Detection.stopCamera()
        print("Button clicked!")
        lessons = Lessons(self.stacked_widget)
        self.stacked_widget.addWidget(lessons)
        self.stacked_widget.setCurrentWidget(lessons)

    def changeModule(self):
        if database.findRowIDValue('right_answer', database.getChosenLesson()) <
33:
            self.moduleLabel.setText("Module 2")
        elif database.findRowIDValue('right_answer', database.getChosenLesson()) <
42 and database.findRowIDValue('right_answer', database.getChosenLesson()) > 33:
            self.moduleLabel.setText("Module 3")
        elif database.findRowIDValue('right_answer', database.getChosenLesson()) >
41:
            self.moduleLabel.setText("Module 4")

    def gotoSubtopics(self):
        if database.findRowIDValue('right_answer', database.getChosenLesson()) ==
database.getLatestLesson():
            database.updateLatest()
```

```
        if database.findRowIDValue('right_answer', database.getChosenLesson()) <
33:
            self.Detection.stopCamera()
            from introduction import Introduction
            introduction = Introduction(self.stacked_widget)
            self.stacked_widget.addWidget(introduction)
            self.stacked_widget.setCurrentWidget(introduction)
        elif database.findRowIDValue('right_answer', database.getChosenLesson()) <
42 and database.findRowIDValue('right_answer', database.getChosenLesson()) > 32:
            self.Detection.stopCamera()
            from greetings import Greetings
            greetings = Greetings(self.stacked_widget)
            self.stacked_widget.addWidget(greetings)
            self.stacked_widget.setCurrentWidget(greetings)
        elif database.findRowIDValue('right_answer', database.getChosenLesson()) >
41:
            self.Detection.stopCamera()
            from vocab import Vocab
            vocab = Vocab(self.stacked_widget)
            self.stacked_widget.addWidget(vocab)
            self.stacked_widget.setCurrentWidget(vocab)




class Detection(QThread):
    #Initialize Class UI
    CameraFrame = pyqtSignal(QImage)
    LabelTextChanged = pyqtSignal(str)
    CheckAnswer = pyqtSignal(bool)
    loading = pyqtSignal(bool)
    global threadCamera
    threadCamera = False

    def save_keypoints_to_csv(self, keypoints, filename="keypoints.csv"):
        # Save the keypoints to a CSV file
        with open(filename, mode='w', newline='') as file:
            writer = csv.writer(file)
            # Write the header
            header = ['pose_x', 'pose_y', 'pose_z', 'pose_visibility'] * 33 + \
                     ['face_x', 'face_y', 'face_z'] * 468 + \
                     ['lh_x', 'lh_y', 'lh_z'] * 21 + \
                     ['rh_x', 'rh_y', 'rh_z'] * 21
            writer.writerow(header)
            # Write the keypoints
            writer.writerow(keypoints)
```

```
    def __init__(self):
        super(Detection, self).__init__()
        self.cap = None  # Initialize cap to None
        self.running = False  # Flag to indicate if the thread should be running
        self.actions = self.defineWords()
        self.model = self.initializeModel()


    def startCamera(self):
        self.cap = cv2.VideoCapture(database.getCam('camera', 1))  # Open the
camera
        if not self.cap.isOpened():
            print("Error: Couldn't open camera.")
            return

        self.running = True  # Set the running flag to True

        # Set up timer to read frames and update GUI
        self.timer = QTimer(self)
        self.timer.timeout.connect(self.run)
        global threadCamera
        threadCamera = True
        self.timer.start(1000 // 20)  # Read frames every 33 ms (30 fps)

    def stopCamera(self):
        self.running = False  # Set the running flag to False
        if self.cap and self.cap.isOpened():
            self.cap.release()
            self.timer.stop()
        if hasattr(self, 'timer') and self.timer.isActive():
            self.timer.stop()
        self.wait()  # Wait for the thread to finish



    def startTimer(self):
        TIMER = int(3)
        prev = time.time()
        while TIMER >= 0:
            ret, image = self.cap.read()
            # Display countdown on each frame
            # specify the font and draw the
            # countdown using puttext
            font = cv2.FONT_HERSHEY_SIMPLEX
            cv2.putText(image, str(TIMER),
```

```
                        (200, 250), font,
                        7, (0, 255, 255),
                        4, cv2.LINE_AA)
            if ret:
                # Resize frame
                # image = cv2.resize(image, (960, 540))  # Adjust the dimensions
as needed
                # Convert frame to RGB format
                rgbImage = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)


                # Convert RGB image to QImage
                h, w, ch = rgbImage.shape
                bytesPerLine = ch * w
                qImg = QImage(rgbImage.data, w, h, bytesPerLine,
QImage.Format_RGB888)
                # Convert QImage to QPixmap to display in QLabel
                pixmap = qImg.scaled(960, 540, aspectRatioMode=Qt.KeepAspectRatio)
                self.CameraFrame.emit(pixmap)


            cv2.waitKey(125)

            # current time
            cur = time.time()

            # Update and keep track of Countdown
            # if time elapsed is one second
            # then decrease the counter
            if cur-prev >= 1:
                prev = cur
                TIMER = TIMER-1
                global startDetection
                startDetection = 1

        self.loading.emit(True)



    def mediapipe_detection(self, image, model):
        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB) # COLOR CONVERSION BGR 2
RGB
        image.flags.writeable = False                  # Image is no longer
writeable
        results = model.process(image)                 # Make prediction
        image.flags.writeable = True                   # Image is now writeable
```

```
        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR) # COLOR COVERSION RGB 2 BGR
        return image, results

    def draw_landmarks(self, image, results):
        mp_drawing.draw_landmarks(image, results.face_landmarks,
mp_holistic.FACEMESH_TESSELATION) # Draw face connections
        mp_drawing.draw_landmarks(image, results.pose_landmarks,
mp_holistic.POSE_CONNECTIONS) # Draw pose connections
        mp_drawing.draw_landmarks(image, results.left_hand_landmarks,
mp_holistic.HAND_CONNECTIONS) # Draw left hand connections
        mp_drawing.draw_landmarks(image, results.right_hand_landmarks,
mp_holistic.HAND_CONNECTIONS) # Draw right hand connections

    def draw_styled_landmarks(self, image, results):
        # Draw face connections
        mp_drawing.draw_landmarks(image, results.face_landmarks,
mp_holistic.FACEMESH_TESSELATION,
                                  mp_drawing.DrawingSpec(color=(80,110,10),
thickness=1, circle_radius=1),
                                  mp_drawing.DrawingSpec(color=(80,256,121),
thickness=1, circle_radius=1)
                                  )
        # Draw pose connections
        mp_drawing.draw_landmarks(image, results.pose_landmarks,
mp_holistic.POSE_CONNECTIONS,
                                  mp_drawing.DrawingSpec(color=(80,22,10),
thickness=2, circle_radius=4),
                                  mp_drawing.DrawingSpec(color=(80,44,121),
thickness=2, circle_radius=2)
                                  )
        # Draw left hand connections
        mp_drawing.draw_landmarks(image, results.left_hand_landmarks,
mp_holistic.HAND_CONNECTIONS,
                                  mp_drawing.DrawingSpec(color=(121,22,76),
thickness=2, circle_radius=4),
                                  mp_drawing.DrawingSpec(color=(121,44,250),
thickness=2, circle_radius=2)
                                  )
        # Draw right hand connections
        mp_drawing.draw_landmarks(image, results.right_hand_landmarks,
mp_holistic.HAND_CONNECTIONS,
                                  mp_drawing.DrawingSpec(color=(245,117,66),
thickness=2, circle_radius=4),
                                  mp_drawing.DrawingSpec(color=(245,66,230),
thickness=2, circle_radius=2)
                                  )
```

```
    def defineWords(self):
        vocabA = [42, 44, 45]
        vocabB = [48, 49, 50]
        vocabC = [43, 46, 47]
        lesson = database.findRowIDValue('right_answer',
database.getChosenLesson())
        if lesson < 33:
            actions = np.array(['Ako si', 'Ano ang pangalan mo', 'Ilang taon ka
na', 'Sino'])
        elif lesson < 42 and lesson > 32:
            actions = np.array(['Ingat ka', 'Kumusta ka', 'Magandang Araw',
'Magandang Gabi', 'Magandang Hapon', 'Magandang Umaga', 'Maraming Salamat',
'Paalam', 'Pasensya na'])
        elif lesson in vocabA:
            actions = np.array(['Bahay', 'Sala', 'Silid'])
        elif lesson in vocabB:
            actions = np.array(['Dilaw', 'Kusina', 'Ube'])
        elif lesson in vocabC:
            actions = np.array(['Guro', 'Kailan', 'Pinto'])

        return actions

    def definemodel(self):
        vocabA = [42, 44, 45]
        vocabB = [48, 49, 50]
        vocabC = [43, 46, 47]
        lesson = database.findRowIDValue('right_answer',
database.getChosenLesson())
        if lesson < 33:
            model = 'Kaway-GUI/model/introduction.h5'
            return model
        elif lesson < 42 and lesson > 32:
            model = 'Kaway-GUI/model/greetings.h5'
            return model
        elif lesson in vocabA:
            model = 'Kaway-GUI/model/vocabA.h5'
            return model
        elif lesson in vocabB:
            model = 'Kaway-GUI/model/vocabB.h5'
            return model
        elif lesson in vocabC:
            model = 'Kaway-GUI/model/vocabC.h5'
            return model
```

```
    def initializeModel(self):
        model = Sequential()
        model.add(LSTM(64, return_sequences=False, activation='relu',
input_shape=(40, 1662)))
        model.add(Dense(16, activation='relu'))
        model.add(Dense(self.actions.shape[0], activation='softmax'))
        model.load_weights(self.definemodel())
        return model

    colors = [(245,117,16), (117,245,16), (16,117,245)]
    def prob_viz(self, res, actions, input_frame, colors):
        output_frame = input_frame.copy()
        return output_frame

    def extract_keypoints(self, results):
        pose = np.array([[res.x, res.y, res.z, res.visibility] for res in
results.pose_landmarks.landmark]).flatten() if results.pose_landmarks else
np.zeros(33*4)
        face = np.array([[res.x, res.y, res.z] for res in
results.face_landmarks.landmark]).flatten() if results.face_landmarks else
np.zeros(468*3)
        lh = np.array([[res.x, res.y, res.z] for res in
results.left_hand_landmarks.landmark]).flatten() if results.left_hand_landmarks
else np.zeros(21*3)
        rh = np.array([[res.x, res.y, res.z] for res in
results.right_hand_landmarks.landmark]).flatten() if results.right_hand_landmarks
else np.zeros(21*3)
        return np.concatenate([pose, face, lh, rh])

    def run(self):
        # 1. New detection variables
        sequence = []
        sentence = []
        predictions = []
        threshold = 0.5
        global startDetection
        startDetection = 0
        global threadCamera

        if self.cap is None:
            return  # Exit if the camera was not successfully initialized

        if threadCamera == True:
```

```
            with mp_holistic.Holistic(min_detection_confidence=0.5,
min_tracking_confidence=0.5) as holistic:
                while self.cap.isOpened() and self.running:

                    TIMER = int(3)
                    # Read feed
                    ret, frame = self.cap.read()
                    image, results = self.mediapipe_detection(frame, holistic)
                    # self.draw_styled_landmarks(image, results)


                    # Show to screen and wait for key to be pressed
                    if ret:
                        # Resize frame
                        # image = cv2.resize(image, (960, 540))  # Adjust the
dimensions as needed
                        # Convert frame to RGB format
                        rgbImage = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)


                        # Convert RGB image to QImage
                        h, w, ch = rgbImage.shape
                        bytesPerLine = ch * w
                        qImg = QImage(rgbImage.data, w, h, bytesPerLine,
QImage.Format_RGB888)
                        # Convert QImage to QPixmap to display in QLabel
                        pixmap = qImg.scaled(960, 540,
aspectRatioMode=Qt.KeepAspectRatio)
                        self.CameraFrame.emit(pixmap)
                    k = cv2.waitKey(125)



                    while (startDetection == 1):
                        # Make detections
                        ret, frame = self.cap.read()
                        image, results = self.mediapipe_detection(frame, holistic)
                        # self.draw_styled_landmarks(image, results)

                        if ret:
                            # Resize frame
                            # image = cv2.resize(image, (960, 540))  # Adjust the
dimensions as needed
```

```
                            # Convert frame to RGB format
                            rgbImage = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)


                            # Convert RGB image to QImage
                            h, w, ch = rgbImage.shape
                            bytesPerLine = ch * w
                            qImg = QImage(rgbImage.data, w, h, bytesPerLine,
QImage.Format_RGB888)
                            # Convert QImage to QPixmap to display in QLabel
                            pixmap = qImg.scaled(960, 540,
aspectRatioMode=Qt.KeepAspectRatio)
                            self.CameraFrame.emit(pixmap)
                        k = cv2.waitKey(1)

                        # 2. Prediction logic
                        keypoints = self.extract_keypoints(results)
                        self.save_keypoints_to_csv(keypoints,
"detection_keypoints.csv")

                        sequence.append(keypoints)
                        sequence = sequence[-40:]


                        if len(sequence) == 40:

                            res = self.model.predict(np.expand_dims(sequence,
axis=0))[0]
                            print(self.actions[np.argmax(res)])

                            if self.actions[np.argmax(res)] ==
database.getChosenLesson():
                                print('correct')
                                self.LabelTextChanged.emit(self.actions[np.argmax(
res)])
                                self.CheckAnswer.emit(True)
                            else:
                                self.LabelTextChanged.emit(self.actions[np.argmax(
res)])
                                self.CheckAnswer.emit(False)



                            predictions.append(np.argmax(res))
                            startDetection = 0
                            sequence = []
```

```
                        #3. Viz logic
                            if np.unique(predictions[-10:])[0]==np.argmax(res):
                                if res[np.argmax(res)] > threshold:

                                    if len(sentence) > 0:
                                        if self.actions[np.argmax(res)] !=
sentence[-1]:
                                            sentence.append(self.actions[np.argmax
(res)])
                                    else:
                                        sentence.append(self.actions[np.argmax(res
)])

                                if len(sentence) > 5:
                                    sentence = sentence[-5:]


                                # Viz probabilities
                                image = self.prob_viz(res, self.actions, image,
self.colors)

                        # ret, image = self.cap.read()  # Read frame from camera
            # Release the video capture and destroy OpenCV windows
            if self.cap and self.cap.isOpened():
                self.cap.release()
```

Greetings.py

```
# Pyqt import
from PyQt5.QtWidgets import *
from PyQt5 import uic
from PyQt5.QtGui import *
from PyQt5.QtCore import *
from functools import partial


from db import database
from modules_two import Modules

class Greetings(QWidget):
    def __init__(self, stacked_widget):
        super(Greetings, self).__init__()
        self.stacked_widget = stacked_widget
```

```
        # Load the ui
        uic.loadUi("Kaway-GUI/pages/lessons_greetings_tab.ui", self)

        # define a-z buttons
        self.umagaButton = self.findChild(QPushButton, "UMAGA")
        self.haponButton = self.findChild(QPushButton, "HAPON")
        self.gabiButton = self.findChild(QPushButton, "GABI")
        self.arawButton = self.findChild(QPushButton, "ARAW")
        self.kumustaButton = self.findChild(QPushButton, "KUMUSTA")
        self.paalamButton = self.findChild(QPushButton, "PAALAM")
        self.ingatButton = self.findChild(QPushButton, "INGAT")
        self.salamatButton = self.findChild(QPushButton, "SALAMAT")
        self.pasensyaButton = self.findChild(QPushButton, "PASENSYA")

        # Define side buttons
        self.homeButton = self.findChild(QPushButton, "Home")
        self.homeButton.clicked.connect(self.gotoHome)
        self.lessontabButton = self.findChild(QPushButton, "Lessons")
        self.lessontabButton.clicked.connect(self.gotoLessons)

        # Define what buttons do
        self.umagaButton.clicked.connect(partial(self.gotoAssessment, 'Magandang
Umaga'))
        self.haponButton.clicked.connect(partial(self.gotoAssessment, 'Magandang
Hapon'))
        self.gabiButton.clicked.connect(partial(self.gotoAssessment, 'Magandang
Gabi'))
        self.arawButton.clicked.connect(partial(self.gotoAssessment, 'Magandang
Araw'))
        self.kumustaButton.clicked.connect(partial(self.gotoAssessment, 'Kumusta
ka'))
        self.paalamButton.clicked.connect(partial(self.gotoAssessment, 'Paalam'))
        self.ingatButton.clicked.connect(partial(self.gotoAssessment, 'Ingat ka'))
        self.salamatButton.clicked.connect(partial(self.gotoAssessment, 'Maraming
Salamat'))
        self.pasensyaButton.clicked.connect(partial(self.gotoAssessment, 'Pasensya
na'))

        self.hideSubtopic(database.getLatestLesson())

    def gotoHome(self):
        from home import Home
        print("Button clicked!")
        home = Home(self.stacked_widget)
        self.stacked_widget.addWidget(home)
        self.stacked_widget.setCurrentWidget(home)
```

```
    def gotoLessons(self):
        #import functions
        from lessonstab import Lessons

        print("Button clicked!")
        lessons = Lessons(self.stacked_widget)
        self.stacked_widget.addWidget(lessons)
        self.stacked_widget.setCurrentWidget(lessons)

    def gotoAssessment(self, value):
        print(value)
        database.putChosenLesson(value)
        print(database.getChosenLesson)

        modules = Modules(self.stacked_widget)
        self.stacked_widget.addWidget(modules)
        self.stacked_widget.setCurrentWidget(modules)

    def hideSubtopic(self, latest):
        buttons = [self.umagaButton, self.haponButton, self.gabiButton,
self.arawButton, self.kumustaButton, self.paalamButton, self.ingatButton,
self.salamatButton, self.pasensyaButton]

        for index, button in enumerate(buttons):
            comp = index + 32
            if comp < latest:
                button.setEnabled(True)
            else:
                button.setEnabled(False)
                button.setText("Complete previous subtopics first")
                button.setStyleSheet('color: rgb(128, 128, 128)')
```

Introduction.py

```
# Pyqt import
from PyQt5.QtWidgets import *
from PyQt5 import uic
from PyQt5.QtGui import *
from PyQt5.QtCore import *
from functools import partial


from db import database
from modules_two import Modules
```

```
class Introduction(QWidget):
    def __init__(self, stacked_widget):
        super(Introduction, self).__init__()
        self.stacked_widget = stacked_widget

        # Load the ui
        uic.loadUi("Kaway-GUI/pages/lessons_introduce_tab.ui", self)

        lessonName = ''

        # define a-z buttons
        self.akosiButton = self.findChild(QPushButton, "AKOSI")
        self.pangalanButton = self.findChild(QPushButton, "PANGALAN")
        self.sinoButton = self.findChild(QPushButton, "SINO")
        self.taonButton = self.findChild(QPushButton, "TAON")

        # Define side buttons
        self.homeButton = self.findChild(QPushButton, "Home")
        self.homeButton.clicked.connect(self.gotoHome)
        self.lessontabButton = self.findChild(QPushButton, "Lessons")
        self.lessontabButton.clicked.connect(self.gotoLessons)

        # Define what buttons do
        self.akosiButton.clicked.connect(partial(self.gotoAssessment, 'Ako si'))
        self.pangalanButton.clicked.connect(partial(self.gotoAssessment, 'Ano ang
pangalan mo'))
        self.sinoButton.clicked.connect(partial(self.gotoAssessment, 'Sino'))
        self.taonButton.clicked.connect(partial(self.gotoAssessment, 'Ilang taon
ka na'))

        self.hideSubtopic(database.getLatestLesson())

    def gotoHome(self):
        from home import Home
        print("Button clicked!")
        home = Home(self.stacked_widget)
        self.stacked_widget.addWidget(home)
        self.stacked_widget.setCurrentWidget(home)

    def gotoLessons(self):
        #import functions
        from lessonstab import Lessons
```

```
        print("Button clicked!")
        lessons = Lessons(self.stacked_widget)
        self.stacked_widget.addWidget(lessons)
        self.stacked_widget.setCurrentWidget(lessons)

    def gotoAssessment(self, value):
        print(value)
        database.putChosenLesson(value)
        print(database.getChosenLesson)

        modules = Modules(self.stacked_widget)
        self.stacked_widget.addWidget(modules)
        self.stacked_widget.setCurrentWidget(modules)

    def hideSubtopic(self, latest):
        buttons = [self.akosiButton, self.pangalanButton, self.sinoButton,
self.taonButton]

        for index, button in enumerate(buttons):
            comp = index + 28
            if comp < latest:
                button.setEnabled(True)
            else:
                button.setEnabled(False)
                button.setText("Complete previous subtopics first")
                button.setStyleSheet('color: rgb(128, 128, 128)')
```

Vocab.py

```
# Pyqt import
from PyQt5.QtWidgets import *
from PyQt5 import uic
from PyQt5.QtGui import *
from PyQt5.QtCore import *
from functools import partial


from db import database
from modules_two import Modules

class Vocab(QWidget):
    def __init__(self, stacked_widget):
        super(Vocab, self).__init__()
        self.stacked_widget = stacked_widget

        # Load the ui
        uic.loadUi("Kaway-GUI/pages/lessons_vocab_tab.ui", self)
```

```
        # define a-z buttons
        self.bahayButton = self.findChild(QPushButton, "BAHAY")
        self.pintoButton = self.findChild(QPushButton, "PINTO")
        self.silidButton = self.findChild(QPushButton, "SILID")
        self.salaButton = self.findChild(QPushButton, "SALA")
        self.kusinaButton = self.findChild(QPushButton, "KUSINA")
        self.guroButton = self.findChild(QPushButton, "GURO")
        self.kailanButton = self.findChild(QPushButton, "KAILAN")
        self.dilawButton = self.findChild(QPushButton, "DILAW")
        self.ubeButton = self.findChild(QPushButton, "UBE")

        # Define side buttons
        self.homeButton = self.findChild(QPushButton, "Home")
        self.homeButton.clicked.connect(self.gotoHome)
        self.lessontabButton = self.findChild(QPushButton, "Lessons")
        self.lessontabButton.clicked.connect(self.gotoLessons)

        # Define what buttons do
        self.bahayButton.clicked.connect(partial(self.gotoAssessment, 'Bahay'))
        self.pintoButton.clicked.connect(partial(self.gotoAssessment, 'Pinto'))
        self.silidButton.clicked.connect(partial(self.gotoAssessment, 'Silid'))
        self.salaButton.clicked.connect(partial(self.gotoAssessment, 'Sala'))
        self.kusinaButton.clicked.connect(partial(self.gotoAssessment, 'Kusina'))
        self.guroButton.clicked.connect(partial(self.gotoAssessment, 'Guro'))
        self.kailanButton.clicked.connect(partial(self.gotoAssessment, 'Kailan'))
        self.dilawButton.clicked.connect(partial(self.gotoAssessment, 'Dilaw'))
        self.ubeButton.clicked.connect(partial(self.gotoAssessment, 'Ube'))

        self.hideSubtopic(database.getLatestLesson())

    def gotoHome(self):
        from home import Home
        print("Button clicked!")
        home = Home(self.stacked_widget)
        self.stacked_widget.addWidget(home)
        self.stacked_widget.setCurrentWidget(home)

    def gotoLessons(self):
        #import functions
        from lessonstab import Lessons

        print("Button clicked!")
        lessons = Lessons(self.stacked_widget)
        self.stacked_widget.addWidget(lessons)
        self.stacked_widget.setCurrentWidget(lessons)
```

```
    def gotoAssessment(self, value):
        print(value)
        database.putChosenLesson(value)
        print(database.getChosenLesson)

        modules = Modules(self.stacked_widget)
        self.stacked_widget.addWidget(modules)
        self.stacked_widget.setCurrentWidget(modules)

    def hideSubtopic(self, latest):
        buttons = [self.bahayButton, self.pintoButton, self.silidButton,
self.salaButton, self.kailanButton, self.guroButton, self.kusinaButton,
self.dilawButton, self.ubeButton]

        for index, button in enumerate(buttons):
            comp = index + 41
            if comp < latest:
                button.setEnabled(True)
            else:
                button.setEnabled(False)
                button.setText("Complete previous subtopics first")
                button.setStyleSheet('color: rgb(128, 128, 128)')
```

Modules.py

```
# Pyqt import
from PyQt5.QtWidgets import *
from PyQt5 import uic
from PyQt5.QtGui import *
from PyQt5.QtCore import *
import sys
import warnings
import os
from PyQt5.QtMultimedia import QMediaContent, QMediaPlayer
from PyQt5.QtMultimediaWidgets import QVideoWidget


#import functions
from db import database


# initialize files and warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
path = os.getcwd()

class Modules(QWidget):
```

```
    def __init__(self, stacked_widget):
        super(Modules, self).__init__()
        self.stacked_widget = stacked_widget

        # Load the ui
        uic.loadUi("Kaway-GUI/pages/modules.ui", self)
        self.setFixedSize(1910, 950)

        # initialize mediaplayer
        self.mediaPlayer = QMediaPlayer(self)
        self.videoWidget = self.findChild(QVideoWidget, "Player")
        self.playButton = self.findChild(QPushButton, "Play")
        self.pauseButton = self.findChild(QPushButton, "Pause")
        self.positionSlider = self.findChild(QSlider, "Slider")

        # Load video file
        if database.getChosenLesson() == 'Ñ':
            video_path = f"Kaway-GUI/videos/enye.mp4"
        else:
            video_path = f"Kaway-GUI/videos/{database.getChosenLesson()}.mp4"
        self.loadVideo(video_path)

        # Define labels
        self.moduleLabel = self.findChild(QLabel, "Module")
        self.subtopicLabel = self.findChild(QLabel, "subtopic")
        self.answerText = self.findChild(QLabel, 'AnswerText')
        self.practiceButton = self.findChild(QLabel, 'PracticeNow')
        lesson = database.getChosenLesson()

        # Rename labels
        self.subtopicLabel.setText(database.getValue('module',
database.findRowIDValue('right_answer', lesson)))
        self.answerText.setText(lesson)


        # define buttons
        self.practiceButton = self.findChild(QPushButton, "PracticeNow")
        self.homeButton = self.findChild(QPushButton, "Home")
        self.lessonsButton = self.findChild(QPushButton, "Lessons")


        # Define what buttons do
        # self.practiceButton.clicked.connect(self.gotoAssessment)
        self.homeButton.clicked.connect(self.gotoHome)
        self.lessonsButton.clicked.connect(self.gotoLessons)
        self.playButton.clicked.connect(self.play)
```

```
        self.pauseButton.clicked.connect(self.pause)
        self.positionSlider.sliderMoved.connect(self.setPosition)
        self.practiceButton.clicked.connect(self.practiceNow)

        self.mediaPlayer.setVideoOutput(self.videoWidget)
        self.mediaPlayer.positionChanged.connect(self.positionChanged)
        self.mediaPlayer.durationChanged.connect(self.durationChanged)
        self.mediaPlayer.mediaStatusChanged.connect(self.handleMediaStatusChanged)


    def handleMediaStatusChanged(self, status):
        if status == QMediaPlayer.EndOfMedia:
            self.mediaPlayer.setPosition(0)
            self.mediaPlayer.play()


    def practiceNow(self):
        from assessments_static import UI

        practice = UI(self.stacked_widget)
        self.stacked_widget.addWidget(practice)
        self.stacked_widget.setCurrentWidget(practice)
        self.mediaPlayer.stop()

    # def getLesson(self):
    #     from lessonsAlphabet import LessonsAlphabet
    #     lesson = LessonsAlphabet.lessonName
    #     return lesson

    # define side tab buttons
    def gotoLessons(self):
        #import functions
        from lessonstab import Lessons

        print("Button clicked!")
        lessons = Lessons(self.stacked_widget)
        self.stacked_widget.addWidget(lessons)
        self.stacked_widget.setCurrentWidget(lessons)

    def gotoHome(self):
        from home import Home
        print("Button clicked!")
        home = Home(self.stacked_widget)
        self.stacked_widget.addWidget(home)
        self.stacked_widget.setCurrentWidget(home)
```

```
    # def gotoAssessment(self, value):
    #     LessonsAlphabet.lessonName = value
    #     print(LessonsAlphabet.lessonName)

    #     assessment = UI()
    #     self.stacked_widget.addWidget(assessment)
    #     self.stacked_widget.setCurrentWidget(assessment)

    def play(self):
        if self.mediaPlayer.state() == QMediaPlayer.PausedState:
            self.mediaPlayer.play()

    def pause(self):
        if self.mediaPlayer.state() == QMediaPlayer.PlayingState:
            self.mediaPlayer.pause()

    def positionChanged(self, position):
        self.positionSlider.setValue(position)

    def durationChanged(self, duration):
        self.positionSlider.setRange(0, duration)

    def setPosition(self, position):
        self.mediaPlayer.setPosition(position)

    def loadVideo(self, video_path):
        media = QMediaContent(QUrl.fromLocalFile(video_path))
        self.mediaPlayer.setMedia(media)
        self.mediaPlayer.play()
```

Modules_two.py

```
# Pyqt import
from PyQt5.QtWidgets import *
from PyQt5 import uic
from PyQt5.QtGui import *
from PyQt5.QtCore import *
import sys
import warnings
import os
from PyQt5.QtMultimedia import QMediaContent, QMediaPlayer
from PyQt5.QtMultimediaWidgets import QVideoWidget


#import functions
from db import database
```

```
# initialize files and warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
path = os.getcwd()

class Modules(QWidget):
    def __init__(self, stacked_widget):
        super(Modules, self).__init__()
        self.stacked_widget = stacked_widget

        # Load the ui
        uic.loadUi("Kaway-GUI/pages/modules.ui", self)
        self.setFixedSize(1910, 950)

        # initialize mediaplayer
        self.mediaPlayer = QMediaPlayer(self)
        self.videoWidget = self.findChild(QVideoWidget, "Player")
        self.playButton = self.findChild(QPushButton, "Play")
        self.pauseButton = self.findChild(QPushButton, "Pause")
        self.positionSlider = self.findChild(QSlider, "Slider")

        # Load video file
        video_path = f"Kaway-GUI/videos/{database.getChosenLesson()}.mp4"
        self.loadVideo(video_path)

        # Define labels
        self.moduleLabel = self.findChild(QLabel, "Module")
        self.subtopicLabel = self.findChild(QLabel, "subtopic")
        self.answerText = self.findChild(QLabel, 'AnswerText')
        self.practiceButton = self.findChild(QLabel, 'PracticeNow')
        lesson = database.getChosenLesson()

        # Rename labels
        self.subtopicLabel.setText(database.getValue('module',
database.findRowIDValue('right_answer', lesson)))
        self.answerText.setText(lesson)
        self.changeModule()


        # define buttons
        self.practiceButton = self.findChild(QPushButton, "PracticeNow")
        self.homeButton = self.findChild(QPushButton, "Home")
        self.lessonsButton = self.findChild(QPushButton, "Lessons")
```

```
        # Define what buttons do
        # self.practiceButton.clicked.connect(self.gotoAssessment)
        self.homeButton.clicked.connect(self.gotoHome)
        self.lessonsButton.clicked.connect(self.gotoLessons)
        self.playButton.clicked.connect(self.play)
        self.pauseButton.clicked.connect(self.pause)
        self.positionSlider.sliderMoved.connect(self.setPosition)
        self.practiceButton.clicked.connect(self.practiceNow)

        self.mediaPlayer.setVideoOutput(self.videoWidget)
        self.mediaPlayer.positionChanged.connect(self.positionChanged)
        self.mediaPlayer.durationChanged.connect(self.durationChanged)
        self.mediaPlayer.mediaStatusChanged.connect(self.handleMediaStatusChanged)

    def handleMediaStatusChanged(self, status):
        if status == QMediaPlayer.EndOfMedia:
            self.mediaPlayer.setPosition(0)
            self.mediaPlayer.play()


    def practiceNow(self):
        from assessments import UI

        practice = UI(self.stacked_widget)
        self.stacked_widget.addWidget(practice)
        self.stacked_widget.setCurrentWidget(practice)
        self.mediaPlayer.stop()

    # def getLesson(self):
    #     from lessonsAlphabet import LessonsAlphabet
    #     lesson = LessonsAlphabet.lessonName
    #     return lesson

    # define side tab buttons
    def gotoLessons(self):
        #import functions
        from lessonstab import Lessons
        self.mediaPlayer.stop()
        print("Button clicked!")
        lessons = Lessons(self.stacked_widget)
        self.stacked_widget.addWidget(lessons)
        self.stacked_widget.setCurrentWidget(lessons)

    def gotoHome(self):
        from home import Home
        self.mediaPlayer.stop()
```

```
        print("Button clicked!")
        home = Home(self.stacked_widget)
        self.stacked_widget.addWidget(home)
        self.stacked_widget.setCurrentWidget(home)

    # def gotoAssessment(self, value):
    #     LessonsAlphabet.lessonName = value
    #     print(LessonsAlphabet.lessonName)

    #     assessment = UI()
    #     self.stacked_widget.addWidget(assessment)
    #     self.stacked_widget.setCurrentWidget(assessment)

    def play(self):
        if self.mediaPlayer.state() == QMediaPlayer.PausedState:
            self.mediaPlayer.play()

    def pause(self):
        if self.mediaPlayer.state() == QMediaPlayer.PlayingState:
            self.mediaPlayer.pause()

    def positionChanged(self, position):
        self.positionSlider.setValue(position)

    def durationChanged(self, duration):
        self.positionSlider.setRange(0, duration)

    def setPosition(self, position):
        self.mediaPlayer.setPosition(position)

    def loadVideo(self, video_path):
        media = QMediaContent(QUrl.fromLocalFile(video_path))
        self.mediaPlayer.setMedia(media)
        self.mediaPlayer.play()

    def changeModule(self):
        lesson = database.findRowIDValue('right_answer',
database.getChosenLesson())
        if lesson < 33:
            self.moduleLabel.setText("Module 2")
        elif lesson < 42 and lesson > 33:
            self.moduleLabel.setText("Module 3")
        elif lesson > 41:
            self.moduleLabel.setText("Module 4")
```

Profile.py

```
# Pyqt import
from PyQt5.QtWidgets import *
from PyQt5 import uic
from PyQt5.QtGui import *
from PyQt5.QtCore import *
import warnings
import os

from db import database

# initialize files and warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
path = os.getcwd()

# Initialize Classes
class Profile(QMainWindow):
    def __init__(self, stacked_widget):
        super(Profile, self).__init__()
        self.stacked_widget = stacked_widget

        # Load the ui
        uic.loadUi("Kaway-GUI/pages/profile.ui", self)
        self.setFixedSize(1910, 950)

        # define buttons
        self.lessontabButton = self.findChild(QPushButton, "Lessons")
        self.homeButton = self.findChild(QPushButton, "Home")
        self.tutorialButton = self.findChild(QPushButton, "Tutorial")
        self.userLabel = self.findChild(QLabel, "User")
        self.profile = self.findChild(QLabel, 'GIF')
        self.profile.setPixmap(QPixmap.fromImage(QImage(r"Kaway-
GUI\linear\user.jpg")))




        # Define what buttons do
        self.lessontabButton.clicked.connect(self.gotoLessons)
        self.homeButton.clicked.connect(self.gotoHome)


        # Set username to page
        username = database.loadUser()
        self.userLabel.setText(username)
```

```
    def gotoLessons(self):
        #import functions
        from lessonstab import Lessons

        print("Button clicked!")
        lessons = Lessons(self.stacked_widget)
        self.stacked_widget.addWidget(lessons)
        self.stacked_widget.setCurrentWidget(lessons)

    def gotoHome(self):
        from home import Home
        home = Home(self.stacked_widget)
        self.stacked_widget.addWidget(home)
        self.stacked_widget.setCurrentWidget(home)
```

Settings.py

```
# Pyqt import
from PyQt5.QtWidgets import *
from PyQt5 import uic
from PyQt5.QtGui import *
from PyQt5.QtCore import *
import warnings
import os

from db import database

# initialize files and warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
path = os.getcwd()

# Initialize Classes
class Settings(QMainWindow):
    def __init__(self, stacked_widget):
        super(Settings, self).__init__()
        self.stacked_widget = stacked_widget

        # Load the ui
        uic.loadUi("Kaway-GUI/pages/settings.ui", self)
        self.setFixedSize(1910, 950)

        # define buttons
        self.lessontabButton = self.findChild(QPushButton, "Lessons")
        self.homeButton = self.findChild(QPushButton, "Home")
        self.tutorialButton = self.findChild(QPushButton, "Tutorial")
        self.userLabel = self.findChild(QLabel, "User")
```

```
        self.camera = self.findChild(QLineEdit, "camera")
        self.save = self.findChild(QPushButton, "save")

        # Define what buttons do
        self.lessontabButton.clicked.connect(self.gotoLessons)
        self.homeButton.clicked.connect(self.gotoHome)

        # Set username to page
        username = database.loadUser()
        self.userLabel.setText(username)

        self.save.clicked.connect(self.saveSettings)

        self.profile = self.findChild(QPushButton, "Profile")
        self.profile.clicked.connect(self.gotoProfile)

    def saveSettings(self):
        camSettings = self.camera.text()

        if camSettings == '0':
            database.setCam(0)

        elif camSettings == '1':
            database.setCam(1)

        else:
            print("Please input all fields.")

    def gotoLessons(self):
        #import functions
        from lessonstab import Lessons

        print("Button clicked!")
        lessons = Lessons(self.stacked_widget)
        self.stacked_widget.addWidget(lessons)
        self.stacked_widget.setCurrentWidget(lessons)

    def gotoHome(self):
        from home import Home
        home = Home(self.stacked_widget)
        self.stacked_widget.addWidget(home)
        self.stacked_widget.setCurrentWidget(home)

    def gotoProfile(self):
        from profile import Profile
        print("Button clicked!")
```

```
        profile = Profile(self.stacked_widget)
        self.stacked_widget.addWidget(profile)
        self.stacked_widget.setCurrentWidget(profile)
```

Tutorial.py

```
# Pyqt import
from PyQt5.QtWidgets import *
from PyQt5 import uic
from PyQt5.QtGui import *
from PyQt5.QtCore import *
import sys
import warnings
import os
from PyQt5.QtMultimedia import QMediaContent, QMediaPlayer
from PyQt5.QtMultimediaWidgets import QVideoWidget



# initialize files and warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
path = os.getcwd()

class Modules(QWidget):
    def __init__(self, stacked_widget):
        super(Modules, self).__init__()
        self.stacked_widget = stacked_widget

        # Load the ui
        uic.loadUi("Kaway-GUI/pages/tutorial.ui", self)
        self.setFixedSize(1910, 950)

        # initialize mediaplayer
        self.mediaPlayer = QMediaPlayer(self)
        self.videoWidget = self.findChild(QVideoWidget, "Player")
        self.playButton = self.findChild(QPushButton, "Play")
        self.pauseButton = self.findChild(QPushButton, "Pause")
        self.positionSlider = self.findChild(QSlider, "Slider")

        # Load video file
        video_path = f"Kaway-GUI/videos/test.mp4"
        self.loadVideo(video_path)




        self.homeButton = self.findChild(QPushButton, "Home")
```

```
        self.lessonsButton = self.findChild(QPushButton, "Lessons")

        # Define what buttons do
        # self.practiceButton.clicked.connect(self.gotoAssessment)
        self.homeButton.clicked.connect(self.gotoHome)
        self.lessonsButton.clicked.connect(self.gotoLessons)
        self.playButton.clicked.connect(self.play)
        self.pauseButton.clicked.connect(self.pause)
        self.positionSlider.sliderMoved.connect(self.setPosition)


        self.mediaPlayer.setVideoOutput(self.videoWidget)
        self.mediaPlayer.positionChanged.connect(self.positionChanged)
        self.mediaPlayer.durationChanged.connect(self.durationChanged)




    def gotoLessons(self):
        #import functions
        from lessonstab import Lessons

        self.mediaPlayer.stop()
        print("Button clicked!")
        lessons = Lessons(self.stacked_widget)
        self.stacked_widget.addWidget(lessons)
        self.stacked_widget.setCurrentWidget(lessons)


    def gotoHome(self):
        from home import Home
        print("Button clicked!")
        self.mediaPlayer.stop()
        home = Home(self.stacked_widget)
        self.stacked_widget.addWidget(home)
        self.stacked_widget.setCurrentWidget(home)

    def play(self):
        if self.mediaPlayer.state() == QMediaPlayer.PausedState:
            self.mediaPlayer.play()

    def pause(self):
        if self.mediaPlayer.state() == QMediaPlayer.PlayingState:
            self.mediaPlayer.pause()

    def positionChanged(self, position):
```

```
        self.positionSlider.setValue(position)

    def durationChanged(self, duration):
        self.positionSlider.setRange(0, duration)

    def setPosition(self, position):
        self.mediaPlayer.setPosition(position)

    def loadVideo(self, video_path):
        media = QMediaContent(QUrl.fromLocalFile(video_path))
        self.mediaPlayer.setMedia(media)
        self.mediaPlayer.play()
```

Python Code for Detection Model Collection and Training

Test_collect.py

```
import cv2
import numpy as np
import os
from matplotlib import pyplot as plt
import time
import mediapipe as mp


mp_holistic = mp.solutions.holistic # Holistic model
mp_drawing = mp.solutions.drawing_utils # Drawing utilities

def mediapipe_detection(image, model):
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB) # COLOR CONVERSION BGR 2 RGB
    image.flags.writeable = False                  # Image is no longer writeable
    results = model.process(image)                 # Make prediction
    image.flags.writeable = True                   # Image is now writeable
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR) # COLOR COVERSION RGB 2 BGR
    return image, results
def draw_landmarks(image, results):
    mp_drawing.draw_landmarks(image, results.face_landmarks,
mp_holistic.FACEMESH_TESSELATION) # Draw face connections
    mp_drawing.draw_landmarks(image, results.pose_landmarks,
mp_holistic.POSE_CONNECTIONS) # Draw pose connections
    mp_drawing.draw_landmarks(image, results.left_hand_landmarks,
mp_holistic.HAND_CONNECTIONS) # Draw left hand connections
    mp_drawing.draw_landmarks(image, results.right_hand_landmarks,
mp_holistic.HAND_CONNECTIONS) # Draw right hand connections

def draw_styled_landmarks(image, results):
    # Draw face connections
```

```
    mp_drawing.draw_landmarks(image, results.face_landmarks,
mp_holistic.FACEMESH_TESSELATION,
                              mp_drawing.DrawingSpec(color=(80,110,10),
thickness=1, circle_radius=1),
                              mp_drawing.DrawingSpec(color=(80,256,121),
thickness=1, circle_radius=1)
                              )
    # Draw pose connections
    mp_drawing.draw_landmarks(image, results.pose_landmarks,
mp_holistic.POSE_CONNECTIONS,
                              mp_drawing.DrawingSpec(color=(80,22,10), thickness=2,
circle_radius=4),
                              mp_drawing.DrawingSpec(color=(80,44,121),
thickness=2, circle_radius=2)
                              )
    # Draw left hand connections
    mp_drawing.draw_landmarks(image, results.left_hand_landmarks,
mp_holistic.HAND_CONNECTIONS,
                              mp_drawing.DrawingSpec(color=(121,22,76),
thickness=2, circle_radius=4),
                              mp_drawing.DrawingSpec(color=(121,44,250),
thickness=2, circle_radius=2)
                              )
    # Draw right hand connections
    mp_drawing.draw_landmarks(image, results.right_hand_landmarks,
mp_holistic.HAND_CONNECTIONS,
                              mp_drawing.DrawingSpec(color=(245,117,66),
thickness=2, circle_radius=4),
                              mp_drawing.DrawingSpec(color=(245,66,230),
thickness=2, circle_radius=2)
                              )

def extract_keypoints(results):
    pose = np.array([[res.x, res.y, res.z, res.visibility] for res in
results.pose_landmarks.landmark]).flatten() if results.pose_landmarks else
np.zeros(33*4)
    face = np.array([[res.x, res.y, res.z] for res in
results.face_landmarks.landmark]).flatten() if results.face_landmarks else
np.zeros(468*3)
    lh = np.array([[res.x, res.y, res.z] for res in
results.left_hand_landmarks.landmark]).flatten() if results.left_hand_landmarks
else np.zeros(21*3)
    rh = np.array([[res.x, res.y, res.z] for res in
results.right_hand_landmarks.landmark]).flatten() if results.right_hand_landmarks
else np.zeros(21*3)
    return np.concatenate([pose, face, lh, rh])
```

```
# Path for exported data, numpy arrays
path = os.getcwd()
DATA_PATH = os.path.join(path,'MP_Hyan')
print(DATA_PATH)
# Actions that we try to detect
actions = np.array(['Sino_'])

# Thirty videos worth of data
no_sequences = 33

# Videos are going to be 30 frames in length
sequence_length = 40

for action in actions:
    for sequence in range(no_sequences):
        try:
            os.makedirs(os.path.join(DATA_PATH, action, str(sequence)))
        except:
            pass

cap = cv2.VideoCapture(0)
if not cap.isOpened():
    print("Error: Could not open video capture.")
else:
    with mp_holistic.Holistic(min_detection_confidence=0.5,
min_tracking_confidence=0.5) as holistic:

        # NEW LOOP
        # Loop through actions
        for action in actions:
            # Loop through sequences aka videos
            for sequence in range(no_sequences):
                # Loop through video length aka sequence length
                for frame_num in range(sequence_length):

                    # Read feed
                    ret, frame = cap.read()

                    # Make detections
                    image, results = mediapipe_detection(frame, holistic)
    #                 print(results)

                    # Draw landmarks
                    draw_styled_landmarks(image, results)
```

```
                    # NEW Apply wait logic
                    if frame_num == 0:
                        cv2.putText(image, 'STARTING COLLECTION', (120,200),
                                    cv2.FONT_HERSHEY_SIMPLEX, 1, (0,255, 0), 4,
cv2.LINE_AA)
                        cv2.putText(image, 'Collecting frames for {} Video Number
{}'.format(action, sequence), (15,12),
                                    cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1,
cv2.LINE_AA)
                        # Show to screen
                        cv2.imshow('OpenCV Feed', image)
                        cv2.waitKey(2000)
                    else:
                        cv2.putText(image, 'Collecting frames for {} Video Number
{}'.format(action, sequence), (15,12),
                                    cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1,
cv2.LINE_AA)
                        # Show to screen
                        cv2.imshow('OpenCV Feed', image)

                    # NEW Export keypoints
                    keypoints = extract_keypoints(results)
                    npy_path = os.path.join(DATA_PATH, action, str(sequence),
str(frame_num))
                    np.save(npy_path, keypoints)

                    # Break gracefully
                    if cv2.waitKey(10) & 0xFF == ord('q'):
                        break

        cap.release()
        cv2.destroyAllWindows()

    cap.release()
    cv2.destroyAllWindows()
```

Test_train.py

```
import cv2
import numpy as np
import os
from matplotlib import pyplot as plt
import time
import mediapipe as mp


mp_holistic = mp.solutions.holistic # Holistic model
```

```
mp_drawing = mp.solutions.drawing_utils # Drawing utilities

def mediapipe_detection(image, model):
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB) # COLOR CONVERSION BGR 2 RGB
    image.flags.writeable = False                  # Image is no longer writeable
    results = model.process(image)                 # Make prediction
    image.flags.writeable = True                   # Image is now writeable
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR) # COLOR COVERSION RGB 2 BGR
    return image, results

def draw_landmarks(image, results):
    mp_drawing.draw_landmarks(image, results.face_landmarks,
mp_holistic.FACEMESH_TESSELATION) # Draw face connections
    mp_drawing.draw_landmarks(image, results.pose_landmarks,
mp_holistic.POSE_CONNECTIONS) # Draw pose connections
    mp_drawing.draw_landmarks(image, results.left_hand_landmarks,
mp_holistic.HAND_CONNECTIONS) # Draw left hand connections
    mp_drawing.draw_landmarks(image, results.right_hand_landmarks,
mp_holistic.HAND_CONNECTIONS) # Draw right hand connections

def draw_styled_landmarks(image, results):
    # Draw face connections
    mp_drawing.draw_landmarks(image, results.face_landmarks,
mp_holistic.FACEMESH_TESSELATION,
                              mp_drawing.DrawingSpec(color=(80,110,10),
thickness=1, circle_radius=1),
                              mp_drawing.DrawingSpec(color=(80,256,121),
thickness=1, circle_radius=1)
                              )
    # Draw pose connections
    mp_drawing.draw_landmarks(image, results.pose_landmarks,
mp_holistic.POSE_CONNECTIONS,
                              mp_drawing.DrawingSpec(color=(80,22,10), thickness=2,
circle_radius=4),
                              mp_drawing.DrawingSpec(color=(80,44,121),
thickness=2, circle_radius=2)
                              )
    # Draw left hand connections
    mp_drawing.draw_landmarks(image, results.left_hand_landmarks,
mp_holistic.HAND_CONNECTIONS,
                              mp_drawing.DrawingSpec(color=(121,22,76),
thickness=2, circle_radius=4),
                              mp_drawing.DrawingSpec(color=(121,44,250),
thickness=2, circle_radius=2)
                              )
    # Draw right hand connections
```

```
    mp_drawing.draw_landmarks(image, results.right_hand_landmarks,
mp_holistic.HAND_CONNECTIONS,
                              mp_drawing.DrawingSpec(color=(245,117,66),
thickness=2, circle_radius=4),
                              mp_drawing.DrawingSpec(color=(245,66,230),
thickness=2, circle_radius=2)
                              )

def extract_keypoints(results):
    pose = np.array([[res.x, res.y, res.z, res.visibility] for res in
results.pose_landmarks.landmark]).flatten() if results.pose_landmarks else
np.zeros(33*4)
    face = np.array([[res.x, res.y, res.z] for res in
results.face_landmarks.landmark]).flatten() if results.face_landmarks else
np.zeros(468*3)
    lh = np.array([[res.x, res.y, res.z] for res in
results.left_hand_landmarks.landmark]).flatten() if results.left_hand_landmarks
else np.zeros(21*3)
    rh = np.array([[res.x, res.y, res.z] for res in
results.right_hand_landmarks.landmark]).flatten() if results.right_hand_landmarks
else np.zeros(21*3)
    return np.concatenate([pose, face, lh, rh])

# Path for exported data, numpy arrays
path = "C:\\Users\\hyanx\\Documents\\Thesis\\"
DATA_PATH = os.path.join(path,'MP_HyanTest')
print(DATA_PATH)
# Actions that we try to detect
actions = np.array(['Ako si', 'Ano pangalan mo', 'Ilang taon ka na', 'Sino'])

# Thirty videos worth of data
no_sequences = 33

# Videos are going to be 30 frames in length
sequence_length = 40

for action in actions:
    for sequence in range(no_sequences):
        try:
            os.makedirs(os.path.join(DATA_PATH, action, str(sequence)))
        except:
            pass


#TRAINING STARTS HERE
#TRAINING STARTS HERE
```

```
#TRAINING STARTS HERE
#TRAINING STARTS HERE
#TRAINING STARTS HERE
#TRAINING STARTS HERE
from sklearn.model_selection import train_test_split
from keras.layers import LSTM, Dense, Dropout
from keras.utils import to_categorical
from keras.optimizers import SGD

label_map = {label:num for num, label in enumerate(actions)}

sequences, labels = [], []
for action in actions:
    for sequence in np.array(os.listdir(os.path.join(DATA_PATH,
action))).astype(int):
        window = []
        for frame_num in range(sequence_length):
            res = np.load(os.path.join(DATA_PATH, action, str(sequence),
"{}.npy".format(frame_num)))
            window.append(res)
        sequences.append(window)
        labels.append(label_map[action])

np.array(sequences).shape
np.array(labels).shape
X = np.array(sequences)
X.shape
y = to_categorical(labels).astype(int)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.05)
y_test.shape

from keras.models import Sequential
from keras.layers import LSTM, Dense
from keras.callbacks import TensorBoard
from keras.optimizers import AdamW
from keras.callbacks import EarlyStopping

log_dir = os.path.join(path, 'Logs')
if not os.path.exists(log_dir):
    os.makedirs(log_dir)
tb_callback = TensorBoard(log_dir=log_dir)
earlystopping_callback =
EarlyStopping(monitor='categorical_accuracy',verbose=1,min_delta=0.0001,patience=2
0,baseline=None)
model = Sequential()
```

```
model.add(LSTM(64, return_sequences=False, activation='relu',
input_shape=(40,1662)))
# model.add(LSTM(128, return_sequences=True, activation='relu'))
# model.add(LSTM(64, return_sequences=False, activation='relu'))
# model.add(Dense(64, activation='relu'))
model.add(Dense(16, activation='relu'))
model.add(Dense(actions.shape[0], activation='softmax'))

#opt = SGD(lr=0.0001)
model.compile(optimizer='Adam', loss='categorical_crossentropy',
metrics=['categorical_accuracy'])
model.fit(X_train, y_train, epochs=2000, callbacks=[tb_callback,
earlystopping_callback])
model.summary()

#Save model
#model.save('C:/Users/hyanx/Documents/Thesis/MP_HyanLetters/letters_2.h5')
#model.save(r'C:\Users\hyanx\Documents\Thesis\MP_Hyan\introduction.h5')
model.save(r'C:\Users\hyanx\Documents\Thesis\Kaway-GUI\model\m2.h5')
```

Test_camerascreen.py

```
import cv2
import numpy as np
import os
from matplotlib import pyplot as plt
import time
import datetime
import mediapipe as mp
from keras.models import Sequential
from keras.layers import LSTM, Dense
from keras.callbacks import TensorBoard
from sklearn.model_selection import train_test_split
from keras.utils import to_categorical
from scipy import stats


mp_holistic = mp.solutions.holistic # Holistic model
mp_drawing = mp.solutions.drawing_utils # Drawing utilities

def mediapipe_detection(image, model):
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB) # COLOR CONVERSION BGR 2 RGB
    image.flags.writeable = False                  # Image is no longer writeable
    results = model.process(image)                 # Make prediction
    image.flags.writeable = True                   # Image is now writeable
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR) # COLOR COVERSION RGB 2 BGR
    return image, results
```

```
def draw_landmarks(image, results):
    mp_drawing.draw_landmarks(image, results.face_landmarks,
mp_holistic.FACEMESH_TESSELATION) # Draw face connections
    mp_drawing.draw_landmarks(image, results.pose_landmarks,
mp_holistic.POSE_CONNECTIONS) # Draw pose connections
    mp_drawing.draw_landmarks(image, results.left_hand_landmarks,
mp_holistic.HAND_CONNECTIONS) # Draw left hand connections
    mp_drawing.draw_landmarks(image, results.right_hand_landmarks,
mp_holistic.HAND_CONNECTIONS) # Draw right hand connections

def draw_styled_landmarks(image, results):
    # Draw face connections
    mp_drawing.draw_landmarks(image, results.face_landmarks,
mp_holistic.FACEMESH_TESSELATION,
                              mp_drawing.DrawingSpec(color=(80,110,10),
thickness=1, circle_radius=1),
                              mp_drawing.DrawingSpec(color=(80,256,121),
thickness=1, circle_radius=1)
                              )
    # Draw pose connections
    mp_drawing.draw_landmarks(image, results.pose_landmarks,
mp_holistic.POSE_CONNECTIONS,
                              mp_drawing.DrawingSpec(color=(80,22,10), thickness=2,
circle_radius=4),
                              mp_drawing.DrawingSpec(color=(80,44,121),
thickness=2, circle_radius=2)
                              )
    # Draw left hand connections
    mp_drawing.draw_landmarks(image, results.left_hand_landmarks,
mp_holistic.HAND_CONNECTIONS,
                              mp_drawing.DrawingSpec(color=(121,22,76),
thickness=2, circle_radius=4),
                              mp_drawing.DrawingSpec(color=(121,44,250),
thickness=2, circle_radius=2)
                              )
    # Draw right hand connections
    mp_drawing.draw_landmarks(image, results.right_hand_landmarks,
mp_holistic.HAND_CONNECTIONS,
                              mp_drawing.DrawingSpec(color=(245,117,66),
thickness=2, circle_radius=4),
                              mp_drawing.DrawingSpec(color=(245,66,230),
thickness=2, circle_radius=2)
                              )

def extract_keypoints(results):
```

```
    pose = np.array([[res.x, res.y, res.z, res.visibility] for res in
results.pose_landmarks.landmark]).flatten() if results.pose_landmarks else
np.zeros(33*4)
    face = np.array([[res.x, res.y, res.z] for res in
results.face_landmarks.landmark]).flatten() if results.face_landmarks else
np.zeros(468*3)
    lh = np.array([[res.x, res.y, res.z] for res in
results.left_hand_landmarks.landmark]).flatten() if results.left_hand_landmarks
else np.zeros(21*3)
    rh = np.array([[res.x, res.y, res.z] for res in
results.right_hand_landmarks.landmark]).flatten() if results.right_hand_landmarks
else np.zeros(21*3)
    return np.concatenate([pose, face, lh, rh])

actions = np.array(['Ako si', 'Ano pangalan mo', 'Ilang taon ka na', 'Sino'])
model = Sequential()
model.add(LSTM(64, return_sequences=False, activation='relu',
input_shape=(40,1662)))
# model.add(LSTM(128, return_sequences=True, activation='relu'))
# model.add(LSTM(64, return_sequences=False, activation='relu'))
# model.add(Dense(64, activation='relu'))
model.add(Dense(16, activation='relu'))
model.add(Dense(actions.shape[0], activation='softmax'))
model.load_weights(r'C:\Users\hyanx\Documents\Thesis\Kaway-GUI\model\m2.h5')

colors = [(245,117,16), (117,245,16), (16,117,245)]
def prob_viz(res, actions, input_frame, colors):
    output_frame = input_frame.copy()
    return output_frame

# 1. New detection variables
sequence = []
sentence = []
predictions = []
threshold = 0.5



cap = cv2.VideoCapture(1)

# Set mediapipe model
with mp_holistic.Holistic(min_detection_confidence=0.5,
min_tracking_confidence=0.5) as holistic:
    while cap.isOpened():
        for total_attempts in range(5):
            TIMER = int(3)
```

```
            # Read feed
            ret, frame = cap.read()
            if not ret:
                print("Error: Failed to capture frame")
                continue

            image, results = mediapipe_detection(frame, holistic)
            draw_styled_landmarks(image, results)


            # Show to screen and wait for key to be pressed
            cv2.imshow('OpenCV Feed', image)
            k = cv2.waitKey(125)

            if k == ord('s'):
                prev = time.time()

                while TIMER >= 0:
                    ret, image = cap.read()
                    # Display countdown on each frame
                    # specify the font and draw the
                    # countdown using puttext
                    font = cv2.FONT_HERSHEY_SIMPLEX
                    cv2.putText(image, str(TIMER),
                                (200, 250), font,
                                7, (0, 255, 255),
                                4, cv2.LINE_AA)
                    cv2.imshow('OpenCV Feed', image)
                    cv2.waitKey(125)

                    # current time
                    cur = time.time()

                    # Update and keep track of Countdown
                    # if time elapsed is one second
                    # then decrease the counter
                    if cur-prev >= 1:
                        prev = cur
                        TIMER = TIMER-1
                        startDetection = 1

                while(startDetection == 1):
                    # Make detections
                    ret, frame = cap.read()
                    image, results = mediapipe_detection(frame, holistic)
                    draw_styled_landmarks(image, results)
```

```
                    # 2. Prediction logic
                    keypoints = extract_keypoints(results)

                    sequence.append(keypoints)
                    sequence = sequence[-40:]


                    if len(sequence) == 40:
                        res = model.predict(np.expand_dims(sequence, axis=0))[0]
                        print(actions[np.argmax(res)])
                        predictions.append(np.argmax(res))
                        startDetection = 0
                        sequence = []


                    #3. Viz logic
                        if np.unique(predictions[-10:])[0]==np.argmax(res):
                            if res[np.argmax(res)] > threshold:

                                if len(sentence) > 0:
                                    if actions[np.argmax(res)] != sentence[-1]:
                                        sentence.append(actions[np.argmax(res)])
                                else:
                                    sentence.append(actions[np.argmax(res)])

                        if len(sentence) > 5:
                            sentence = sentence[-5:]


                        # Viz probabilities
                        image = prob_viz(res, actions, image, colors)

                    cv2.imshow('OpenCV Feed', image)

                    # Break gracefully
                    if cv2.waitKey(10) & 0xFF == ord('q'):
                        cap.release()
                        cv2.destroyAllWindows()
                        break
```

Collect_imgs.py

```
import os

import cv2
```

```
DATA_DIR = './data_letters'
if not os.path.exists(DATA_DIR):
    os.makedirs(DATA_DIR)

number_of_classes = 1
dataset_size = 100

cap = cv2.VideoCapture(0)
for j in range(number_of_classes):
    if not os.path.exists(os.path.join(DATA_DIR, str(j))):
        os.makedirs(os.path.join(DATA_DIR, str(j)))

    print('Collecting data for class {}'.format(j))

    done = False
    while True:
        ret, frame = cap.read()
        cv2.putText(frame, 'Ready? Press "Q" ! :)', (100, 50),
cv2.FONT_HERSHEY_SIMPLEX, 1.3, (0, 255, 0), 3,
                    cv2.LINE_AA)
        cv2.imshow('frame', frame)
        if cv2.waitKey(25) == ord('q'):
            break

    counter = 0
    while counter < dataset_size:
        ret, frame = cap.read()
        cv2.imshow('frame', frame)
        cv2.waitKey(25)
        cv2.imwrite(os.path.join(DATA_DIR, str(j), '{}.jpg'.format(counter)),
frame)

        counter += 1

cap.release()
cv2.destroyAllWindows()
```

Create_datasetSIFT.py

```
import os
import pickle

import mediapipe as mp
import cv2
import matplotlib.pyplot as plt
```

```
mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles

hands = mp_hands.Hands(static_image_mode=True, min_detection_confidence=0.3)

DATA_DIR = './data'

data = []
labels = []
counter = 0  # Counter to track the number of iterations

# Ensure only directories are processed
for dir_ in os.listdir(DATA_DIR):
    dir_path = os.path.join(DATA_DIR, dir_)
    if os.path.isdir(dir_path):
        for img_path in os.listdir(dir_path):
            if counter >= 100:
                break  # Break the loop if the counter reaches 100

            data_aux = []
            x_ = []
            y_ = []

            img_path = os.path.join(dir_path, img_path)
            imgGray = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)

            sift = cv2.SIFT_create()
            keypoints = sift.detect(imgGray, None)
            imgKeypoints = cv2.drawKeypoints(imgGray, keypoints, None,
flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)

            results = hands.process(imgKeypoints)
            if results.multi_hand_landmarks:
                for hand_landmarks in results.multi_hand_landmarks:
                    for i in range(len(hand_landmarks.landmark)):
                        x = hand_landmarks.landmark[i].x
                        y = hand_landmarks.landmark[i].y

                        x_.append(x)
                        y_.append(y)

                    for i in range(len(hand_landmarks.landmark)):
                        x = hand_landmarks.landmark[i].x
                        y = hand_landmarks.landmark[i].y
```

```
                        data_aux.append(x - min(x_))
                        data_aux.append(y - min(y_))

                data.append(data_aux)
                labels.append(dir_)
                counter += 1  # Increment the counter after processing each image

# Save the data
with open(os.path.join(DATA_DIR, 'data.pickle'), 'wb') as f:
    pickle.dump({'data': data, 'labels': labels}, f)
```

Inference_classifer.py

```
import pickle

import cv2
import mediapipe as mp
import numpy as np

model_dict = pickle.load(open('./data/model.p', 'rb'))
model = model_dict['model']

cap = cv2.VideoCapture(0)

mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles

hands = mp_hands.Hands(static_image_mode=True, min_detection_confidence=0.3)

labels_dict = {0: 'G'}
while True:

    data_aux = []
    x_ = []
    y_ = []

    ret, frame = cap.read()

    H, W, _ = frame.shape

    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

    results = hands.process(frame_rgb)
    if results.multi_hand_landmarks:
        for hand_landmarks in results.multi_hand_landmarks:
            mp_drawing.draw_landmarks(
```

```
                frame,  # image to draw
                hand_landmarks,  # model output
                mp_hands.HAND_CONNECTIONS,  # hand connections
                mp_drawing_styles.get_default_hand_landmarks_style(),
                mp_drawing_styles.get_default_hand_connections_style())

        for hand_landmarks in results.multi_hand_landmarks:
            for i in range(len(hand_landmarks.landmark)):
                x = hand_landmarks.landmark[i].x
                y = hand_landmarks.landmark[i].y

                x_.append(x)
                y_.append(y)

            for i in range(len(hand_landmarks.landmark)):
                x = hand_landmarks.landmark[i].x
                y = hand_landmarks.landmark[i].y
                data_aux.append(x - min(x_))
                data_aux.append(y - min(y_))

        x1 = int(min(x_) * W) - 10
        y1 = int(min(y_) * H) - 10

        x2 = int(max(x_) * W) - 10
        y2 = int(max(y_) * H) - 10

        prediction = model.predict([np.asarray(data_aux)])

        predicted_character = labels_dict[int(prediction[0])]

        cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 0), 4)
        cv2.putText(frame, predicted_character, (x1, y1 - 10),
cv2.FONT_HERSHEY_SIMPLEX, 1.3, (0, 0, 0), 3,
                    cv2.LINE_AA)

    cv2.imshow('frame', frame)
    cv2.waitKey(1)


cap.release()
cv2.destroyAllWindows()
```

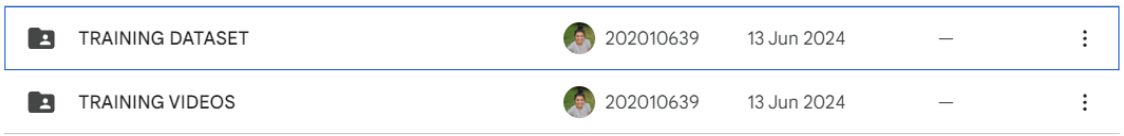
Train_classifier.py

```
import pickle

from sklearn.ensemble import RandomForestClassifier
```

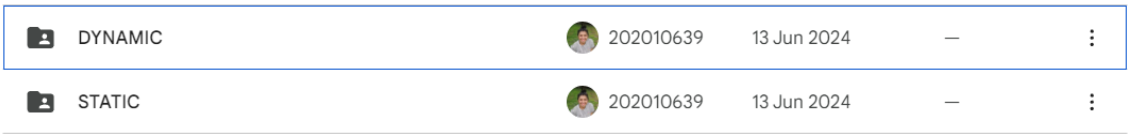
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import numpy as np


data_dict = pickle.load(open('./data/data.pickle', 'rb'))

data = np.asarray(data_dict['data'])
labels = np.asarray(data_dict['labels'])

x_train, x_test, y_train, y_test = train_test_split(data, labels, test_size=0.3,
shuffle=True, stratify=labels)

model = RandomForestClassifier()

model.fit(x_train, y_train)

y_predict = model.predict(x_test)

score = accuracy_score(y_predict, y_test)

print('{}% of samples were classified correctly !'.format(score * 100))

f = open('./data/model.p', 'wb')
pickle.dump({'model': model}, f)
f.close()
```

APPENDIX C: TRAINING, TESTING, AND MODEL DATA

Training and Testing data can be viewed here: https://drive.google.com/drive/folders/1M4c9ujLgEGlSHxXON2GPeu9rFoyRqb6B?usp=sharing



Training Dataset: https://drive.google.com/drive/folders/1P13SRZRJAAaozMjjaBOjVRd-owXGb5wh?usp=drive_link



Training Dataset:

https://drive.google.com/drive/folders/1GqMLPF60z28ItINUzitvcpoeVvs4T0iV?usp=drive_link

The full documentation of the data can be viewed here:

https://drive.google.com/drive/folders/1P13SRZRJAAaozMjjaBOjVRd-owXGb5wh?usp=drive_link

APPENDIX D: ERRATUM

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

Date	May 13, 2024
To	Engr. Rex Paolo Gamara Engr. Danilyn Joy Aquino Engr. Juan Miguel Villaroel
Thru	Engr. Antipas Teologo Jr. **Course Adviser**
Noted by	Dr. Pocholo James Loresco **Thesis Adviser**
From	Julian Jesus Cruz Joshua Peterson Gonzales Kayley Mae Pichay Hyan Jan Suamina **Thesis Proponents**
Subject	**Thesis Erratum**

Good day!

The thesis, titled "Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processing and Deep Learning", is progressing well. However, as the project advances, we have encountered some minor issues regarding the type of machine learning proposed in the system.

We are looking forward to your verdict and approval regarding the changes to be made in our study.

Thank you very much!

SUMMARY

Item No.	ORIGINAL	PROPOSED CHANGES	REASON
1	The study will be utilizing a deep learning model, specifically, a Convolutional Neural Network (CNN) for its sign language classification which will be used to detect the user's gestures.	The proponents decided to use Long Short-Term Memory for the classification of sign language.	After training a CNN model for sign language detection, it has been determined that dynamic signs are better suited with Long Short-Term Memory. LSTM is more compatible with time-progression data similar to the data gathered from the dataset after feature extraction. Since dynamic signs would vary from start to beginning, these changes are better trained and detected by an LSTM model. When CNN has been trained for dynamic signs, it gives out inaccurate answers. Lastly, Python has a dedicated library with LSTM integration into the code which allows for more efficient training.

Specific Objectives

Specific Objectives	Remarks
To develop a dataset or corpus of FSL gestures considering static and dynamic gestures obtained through data acquisition of videos from a camera	No Changes
To develop a Python-based program interfaced with a deep learning model for the FSL recognition, translation of FSL to text, and feedback evaluation based on the user's input	No Changes
To develop a GUI of the Python-based tutoring program incorporated with the loading of FSL lessons, FSL recognition, feedback evaluation and correction, and an audio interface	No Changes
To evaluate the performance of the tutoring system using various metrics including precision, recall, specificity, F1	No Changes

score, Matthew's correlation coefficient, reliability formulas and confusion matrix based on its recognition of Filipino sign language with an obtained accuracy of at least 93.8%	
To evaluate and compare the effectiveness of Developing an Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processing and Deep Learning to the traditional methods of Filipino Sign Language tutoring using Likert Scale survey questionnaire	No Changes

Noted By:

Dr. Pocholo James Loresco

Thesis Adviser

Engr. Antipas Teologo Jr.

Course Adviser

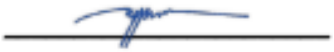
Approved By:

Engr. Danilyn Joy Aquino

Panelist 1

Engr. Juan Miguel Villaroel

Panelist 2



Engr. Rex Paolo Gamara

Head Panelist

APPENDIX E: DOCUMENTS

Consent Form

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

CONSENT FORM

Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processing and Deep Learning

I MARIANNE JOY S. KAHULUGAN (participant name), agree to participate in the research project titled Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processing and Deep Learning, conducted by Julian Cruz, Joshua Gonzales, Kayley Pichay, and Hyan Suamina who has (have) discussed the research project with me.

I have received, read, and kept a copy of the information letter/plain language statement. I have had the opportunity to ask questions about this research and I have received satisfactory answers. I understand the general purposes, risks, and methods of this research.

I consent to participate in the research project and the following has been explained to me:

- The research may not be of direct benefit to me.
- My participation is completely voluntary.
- My right to withdraw from the study at any time without any implications to me.
- The steps that have been taken to minimize any possible risks.
- What I am expected and required to do.
- Whom I should contact for any complaints about the research or the conduct of the research.
- I can request a copy of the research findings and reports.
- Security and confidentiality of my personal information.

In addition, I consent to:

- Audio-visual recording of any part of or all research activities (if applicable)
- Publication of results from this study on the condition that my identity will not be revealed.

Name: MARIANNE JOY S. KAHULUGAN

Signature:

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

CONSENT FORM

Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processing and Deep Learning

I MARIA C. PERDON (participant name), agree to participate in the research project titled Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processing and Deep Learning, conducted by Julian Cruz, Joshua Gonzales, Kayley Pichay, and Hyan Suamina who has (have) discussed the research project with me.

I have received, read, and kept a copy of the information letter/plain language statement. I have had the opportunity to ask questions about this research and I have received satisfactory answers. I understand the general purposes, risks, and methods of this research.

I consent to participate in the research project and the following has been explained to me:

- The research may not be of direct benefit to me.
- My participation is completely voluntary.
- My right to withdraw from the study at any time without any implications to me.
- The steps that have been taken to minimize any possible risks.
- What I am expected and required to do.
- Whom I should contact for any complaints about the research or the conduct of the research.
- I can request a copy of the research findings and reports.
- Security and confidentiality of my personal information.

In addition, I consent to:

- Audio-visual recording of any part of or all research activities (if applicable)
- Publication of results from this study on the condition that my identity will not be revealed.

Name: MARIA MAI G. PERDON

Signature:

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

CONSENT FORM

Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processing and Deep Learning

I Mary Ethelyne M. Marasigan (participant name), agree to participate in the research project titled Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processing and Deep Learning, conducted by Julian Cruz, Joshua Gonzales, Kayley Pichay, and Hyan Suamina who has (have) discussed the research project with me.

I have received, read, and kept a copy of the information letter/plain language statement. I have had the opportunity to ask questions about this research and I have received satisfactory answers. I understand the general purposes, risks, and methods of this research.

I consent to participate in the research project and the following has been explained to me:

- The research may not be of direct benefit to me.
- My participation is completely voluntary.
- My right to withdraw from the study at any time without any implications to me.
- The steps that have been taken to minimize any possible risks.
- What I am expected and required to do.
- Whom I should contact for any complaints about the research or the conduct of the research.
- I can request a copy of the research findings and reports.
- Security and confidentiality of my personal information.

In addition, I consent to:

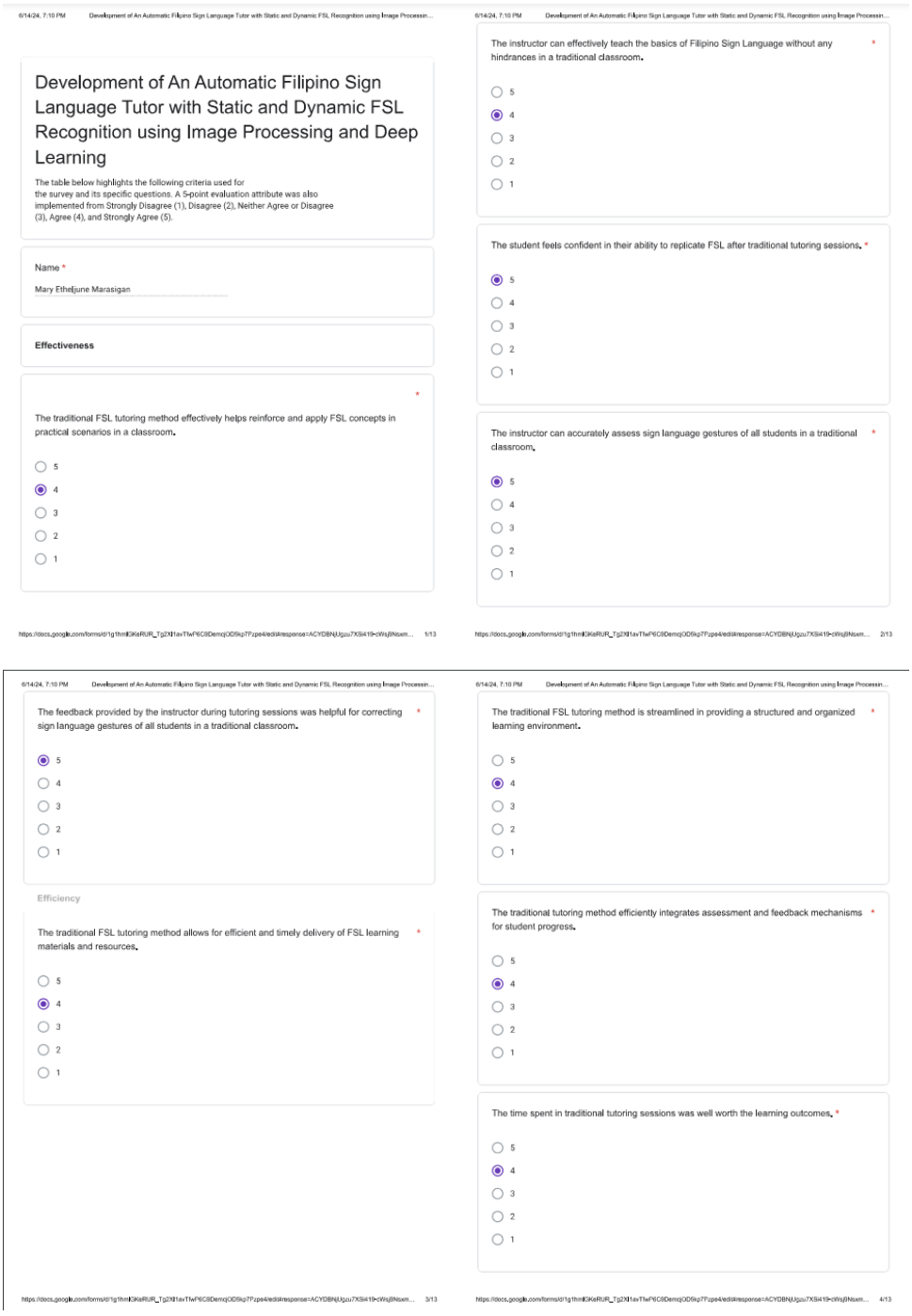
- Audio-visual recording of any part of or all research activities (if applicable)
- Publication of results from this study on the condition that my identity will not be revealed.

Name: Mary Ethelyne M. Marasigan

Signature:

FSL Teachers' PRC License

Likert Scale Survey for Traditional Teaching



6/14/24, 7:10 PM Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processin...

Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processing and Deep Learning

The table below highlights the following criteria used for the survey and its specific questions. A 5-point evaluation attribute was also implemented from Strongly Disagree (1), Disagree (2), Neither Agree or Disagree (3), Agree (4), and Strongly Agree (5).

Name *

Mary Ethel June Marasigan

Effectiveness

The traditional FSL tutoring method effectively helps reinforce and apply FSL concepts in practical scenarios in a classroom. *

- 5
- 4
- 3
- 2
- 1

https://docs.google.com/forms/d/1g1hmI3KeRUR_Tg2Nt1avTfwP6C9DemcjOD5kp7Pzpe4/edit#response=ACYDBNjUgzu7XSi419-cWsjBNoxm... 1/13

6/14/24, 7:10 PM Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processin...

The instructor can effectively teach the basics of Filipino Sign Language without any hindrances in a traditional classroom. *

- 5
- 4
- 3
- 2
- 1

The student feels confident in their ability to replicate FSL after traditional tutoring sessions. *

- 5
- 4
- 3
- 2
- 1

The instructor can accurately assess sign language gestures of all students in a traditional classroom. *

- 5
- 4
- 3
- 2
- 1

https://docs.google.com/forms/d/1g1hmI3KeRUR_Tg2Nt1avTfwP6C9DemcjOD5kp7Pzpe4/edit#response=ACYDBNjUgzu7XSi419-cWsjBNoxm... 2/13

6/14/24, 7:10 PM Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processin...

The feedback provided by the instructor during tutoring sessions was helpful for correcting sign language gestures of all students in a traditional classroom. *

- 5
- 4
- 3
- 2
- 1

Efficiency

The traditional FSL tutoring method allows for efficient and timely delivery of FSL learning materials and resources. *

- 5
- 4
- 3
- 2
- 1

https://docs.google.com/forms/d/1g1hmI3KeRUR_Tg2Nt1avTfwP6C9DemcjOD5kp7Pzpe4/edit#response=ACYDBNjUgzu7XSi419-cWsjBNoxm... 3/13

6/14/24, 7:10 PM Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processin...

The traditional FSL tutoring method is streamlined in providing a structured and organized learning environment. *

- 5
- 4
- 3
- 2
- 1

The traditional tutoring method efficiently integrates assessment and feedback mechanisms for student progress. *

- 5
- 4
- 3
- 2
- 1

The time spent in traditional tutoring sessions was well worth the learning outcomes. *

- 5
- 4
- 3
- 2
- 1

https://docs.google.com/forms/d/1g1hmI3KeRUR_Tg2Nt1avTfwP6C9DemcjOD5kp7Pzpe4/edit#response=ACYDBNjUgzu7XSi419-cWsjBNoxm... 4/13

The traditional tutoring method can efficiently guide all students through the lessons on Filipino Sign Language. *

- 5
- (4)
- 3
- 2
- 1

Learnability

The student can easily grasp and understand the FSL concepts through the traditional FSL tutoring. *

- 5
- (4)
- 3
- 2
- 1

The traditional tutoring method is intuitive and clear with the learning pathways and structures. *

- 5
- (4)
- 3
- 2
- 1

The lessons were presented in a way that was easy to understand and follow during traditional FSL tutoring. *

- 5
- (4)
- 3
- 2
- 1

The lessons provided sufficient guidance to all students for learning new gestures. *

- 5
- (4)
- 3
- 2
- 1

The students feel confident in their ability to follow along with the lessons provided by the traditional FSL tutoring method. *

- 5
- (4)
- 3
- 2
- 1

Memorability

The students can significantly demonstrate improved learning outcomes using the traditional FSL tutoring method. *

- 5
- (4)
- 3
- 2
- 1

The students can notably show progress and achievement through the traditional learning every session. *

- 5
- (4)
- 3
- 2
- 1

The student can consistently show learning progress through the assessments provided by the traditional FSL tutoring method. *

- 5
- (4)
- 3
- 2
- 1

The student can easily recall and replicate the FSL gestures taught during traditional sessions. *

- 5
- (4)
- 3
- 2
- 1

The feedback from the instructor aided in reinforcing the memorization of gestures done by all the students in a classroom. *

- 5
- 4
- 3
- 2
- 1

Cognitive Load

The instructor can present information in a way that was not overwhelming to all students in a classroom. *

- 5
- 4
- 3
- 2
- 1

The students do not feel mentally exhausted after traditional tutoring sessions. *

- 5
- 4
- 3
- 2
- 1

The instructor effectively managed the amount of information presented at one time. *

- 5
- 4
- 3
- 2
- 1

The assessment process did not create an undue mental burden to the students. *

- 5
- 4
- 3
- 2
- 1

The students feel that the cognitive demands during tutoring sessions were reasonable. *

- 5
- 4
- 3
- 2
- 1

Satisfaction

The students seem satisfied with the traditional FSL tutoring method and its functionalities for teaching purposes. *

- 5
- 4
- 3
- 2
- 1

I am likely to recommend traditional FSL tutoring to others interested in learning FSL than with an FSL tutoring application. *

- 5
- 4
- 3
- 2
- 1

The students seem satisfied with the overall learning experience provided by the traditional FSL tutoring. *

- 5
- 4
- 3
- 2
- 1

The students feel confident in their ability to communicate using FSL after using the application. *

- 5
- 4
- 3
- 2
- 1

6/14/24, 7:10 PM Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processin...

Traditional FSL tutoring has increased the students' interest in learning Filipino Sign Language. *

- ○ 5
- ◉ 4
- ○ 3
- ○ 2
- ○ 1

This content is neither created nor endorsed by Google.

Google Forms

https://docs.google.com/forms/d/1g1hmIGKeRUR_Tg2XI1avTfwP6C9DemcjOD5kp7Pzpe4/edit#response=ACYDBNjUgzu7XSi419-cWsj9Nsx... 13/13

6/14/24, 7:16 PM Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processin...

Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processing and Deep Learning

The table below highlights the following criteria used for the survey and its specific questions. A 5-point evaluation attribute was also implemented from Strongly Disagree (1), Disagree (2), Neither Agree or Disagree (3), Agree (4), and Strongly Agree (5).

Name *

Mary Ann G. Perdon

Effectiveness

*

The traditional FSL tutoring method effectively helps reinforce and apply FSL concepts in practical scenarios in a classroom.

- ○ 5
- ○ 4
- ◉ 3
- ○ 2
- ○ 1

https://docs.google.com/forms/d/1g1hmIGKeRUR_Tg2XI1avTfwP6C9DemcjOD5kp7Pzpe4/edit#response=ACYDBNg5HGhVB_6mmYqBLFFW... 1/13

6/14/24, 7:16 PM Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processin...

The instructor can effectively teach the basics of Filipino Sign Language without any hindrances in a traditional classroom. *

- ○ 5
- ○ 4
- ○ 3
- ◉ 2
- ○ 1

The student feels confident in their ability to replicate FSL after traditional tutoring sessions. *

- ○ 5
- ○ 4
- ○ 3
- ◉ 2
- ○ 1

The instructor can accurately assess sign language gestures of all students in a traditional classroom. *

- ○ 5
- ○ 4
- ◉ 3
- ○ 2
- ○ 1

https://docs.google.com/forms/d/1g1hmIGKeRUR_Tg2XI1avTfwP6C9DemcjOD5kp7Pzpe4/edit#response=ACYDBNg5HGhVB_6mmYqBLFFW... 2/13

The feedback provided by the instructor during tutoring sessions was helpful for correcting sign language gestures of all students in a traditional classroom. *

- ○ 5
- ◉ 4
- ○ 3
- ○ 2
- ○ 1

Efficiency

The traditional FSL tutoring method allows for efficient and timely delivery of FSL learning materials and resources. *

- ○ 5
- ○ 4
- ◉ 3
- ○ 2
- ○ 1

The traditional FSL tutoring method is streamlined in providing a structured and organized learning environment. *

- ○ 5
- ◉ 4
- ○ 3
- ○ 2
- ○ 1

The traditional tutoring method efficiently integrates assessment and feedback mechanisms for student progress. *

- ○ 5
- ○ 4
- ◉ 3
- ○ 2
- ○ 1

The time spent in traditional tutoring sessions was well worth the learning outcomes. *

- ○ 5
- ○ 4
- ○ 3
- ◉ 2
- ○ 1

The traditional tutoring method can efficiently guide all students through the lessons on Filipino Sign Language. *

- ○ 5
- ○ 4
- ◉ 3
- ○ 2
- ○ 1

Learnability

The student can easily grasp and understand the FSL concepts through the traditional FSL tutoring. *

- ○ 5
- ○ 4
- ◉ 3
- ○ 2
- ○ 1

The traditional tutoring method is intuitive and clear with the learning pathways and structures. *

- ○ 5
- ○ 4
- ◉ 3
- ○ 2
- ○ 1

The lessons were presented in a way that was easy to understand and follow during traditional FSL tutoring. *

- ○ 5
- ○ 4
- ◉ 3
- ○ 2
- ○ 1

The lessons provided sufficient guidance to all students for learning new gestures. *

- ○ 5
- ○ 4
- ○ 3
- ◉ 2
- ○ 1

The students feel confident in their ability to follow along with the lessons provided by the traditional FSL tutoring method. *

- 5
- 4
- (•) 3
- 2
- 1

Memorability

The students can significantly demonstrate improved learning outcomes using the traditional FSL tutoring method. *

- 5
- 4
- (•) 3
- 2
- 1

The students can notably show progress and achievement through the traditional learning every session. *

- 5
- 4
- (•) 3
- 2
- 1

The student can consistently show learning progress through the assessments provided by the traditional FSL tutoring method. *

- 5
- 4
- 3
- (•) 2
- 1

The student can easily recall and replicate the FSL gestures taught during traditional sessions. *

- 5
- 4
- 3
- (•) 2
- 1

The feedback from the instructor aided in reinforcing the memorization of gestures done by all the students in a classroom. *

- 5
- 4
- (•) 3
- 2
- 1

Cognitive Load

The instructor can present information in a way that was not overwhelming to all students in a classroom. *

- 5
- 4
- (•) 3
- 2
- 1

The students do not feel mentally exhausted after traditional tutoring sessions. *

- 5
- 4
- 3
- (•) 2
- 1

The instructor effectively managed the amount of information presented at one time. *

- 5
- (•) 4
- 3
- 2
- 1

The assessment process did not create an undue mental burden to the students. *

- 5
- 4
- 3
- (•) 2
- 1

The students feel that the cognitive demands during tutoring sessions were reasonable. *

- 5
- 4
- 3
- 2
- 1

Satisfaction

The students seem satisfied with the traditional FSL tutoring method and its functionalities for teaching purposes. *

- 5
- 4
- 3
- 2
- 1

I am likely to recommend traditional FSL tutoring to others interested in learning FSL than with an FSL tutoring application. *

- 5
- 4
- 3
- 2
- 1

The students seem satisfied with the overall learning experience provided by the traditional FSL tutoring. *

- 5
- 4
- 3
- 2
- 1

The students feel confident in their ability to communicate using FSL after using the application. *

- 5
- 4
- 3
- 2
- 1

Traditional FSL tutoring has increased the students' interest in learning Filipino Sign Language. *

- 5
- 4
- 3
- 2
- 1

This content is neither created nor endorsed by Google.

Google Forms

Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processing and Deep Learning

The table below highlights the following criteria used for the survey and its specific questions. A 5-point evaluation attribute was also implemented from Strongly Disagree (1), Disagree (2), Neither Agree or Disagree (3), Agree (4), and Strongly Agree (5).

Name *

Marianne Joy S. Kahulugan

Effectiveness

*

The traditional FSL tutoring method effectively helps reinforce and apply FSL concepts in practical scenarios in a classroom.

- ○ 5
- ◉ 4
- ○ 3
- ○ 2
- ○ 1

The instructor can effectively teach the basics of Filipino Sign Language without any hindrances in a traditional classroom. *

- ○ 5
- ○ 4
- ◉ 3
- ○ 2
- ○ 1

The student feels confident in their ability to replicate FSL after traditional tutoring sessions. *

- ○ 5
- ○ 4
- ◉ 3
- ○ 2
- ○ 1

The instructor can accurately assess sign language gestures of all students in a traditional classroom. *

- ○ 5
- ○ 4
- ◉ 3
- ○ 2
- ○ 1

The feedback provided by the instructor during tutoring sessions was helpful for correcting sign language gestures of all students in a traditional classroom. *

- ○ 5
- ◉ 4
- ○ 3
- ○ 2
- ○ 1

Efficiency

The traditional FSL tutoring method allows for efficient and timely delivery of FSL learning materials and resources. *

- ○ 5
- ○ 4
- ◉ 3
- ○ 2
- ○ 1

The traditional FSL tutoring method is streamlined in providing a structured and organized learning environment. *

- ○ 5
- ○ 4
- ◉ 3
- ○ 2
- ○ 1

The traditional tutoring method efficiently integrates assessment and feedback mechanisms for student progress. *

- ○ 5
- ○ 4
- ◉ 3
- ○ 2
- ○ 1

The time spent in traditional tutoring sessions was well worth the learning outcomes. *

- ○ 5
- ○ 4
- ◉ 3
- ○ 2
- ○ 1

The traditional tutoring method can efficiently guide all students through the lessons on Filipino Sign Language. *

- 5
- (4)
- 3
- 2
- 1

Learnability

The student can easily grasp and understand the FSL concepts through the traditional FSL tutoring. *

- 5
- 4
- (3)
- 2
- 1

The traditional tutoring method is intuitive and clear with the learning pathways and structures. *

- 5
- (4)
- 3
- 2
- 1

The lessons were presented in a way that was easy to understand and follow during traditional FSL tutoring. *

- 5
- (4)
- 3
- 2
- 1

The lessons provided sufficient guidance to all students for learning new gestures. *

- 5
- 4
- (3)
- 2
- 1

The students feel confident in their ability to follow along with the lessons provided by the traditional FSL tutoring method. *

- 5
- 4
- (3)
- 2
- 1

Memorability

The students can significantly demonstrate improved learning outcomes using the traditional FSL tutoring method. *

- 5
- 4
- (3)
- 2
- 1

The students can notably show progress and achievement through the traditional learning every session. *

- 5
- (4)
- 3
- 2
- 1

The student can consistently show learning progress through the assessments provided by the traditional FSL tutoring method. *

- 5
- 4
- 3
- (2)
- 1

The student can easily recall and replicate the FSL gestures taught during traditional sessions. *

- 5
- 4
- 3
- (2)
- 1

The feedback from the instructor aided in reinforcing the memorization of gestures done by all the students in a classroom. *

- 5
- 4
- (•) 3
- 2
- 1

Cognitive Load

The instructor can present information in a way that was not overwhelming to all students in a classroom. *

- 5
- 4
- (•) 3
- 2
- 1

The students do not feel mentally exhausted after traditional tutoring sessions. *

- 5
- 4
- (•) 3
- 2
- 1

The instructor effectively managed the amount of information presented at one time. *

- 5
- 4
- (•) 3
- 2
- 1

The assessment process did not create an undue mental burden to the students. *

- 5
- 4
- (•) 3
- 2
- 1

The students feel that the cognitive demands during tutoring sessions were reasonable. *

- 5
- 4
- (•) 3
- 2
- 1

Satisfaction

The students seem satisfied with the traditional FSL tutoring method and its functionalities for teaching purposes. *

- 5
- 4
- (•) 3
- 2
- 1

I am likely to recommend traditional FSL tutoring to others interested in learning FSL than with an FSL tutoring application. *

- 5
- 4
- 3
- (•) 2
- 1

The students seem satisfied with the overall learning experience provided by the traditional FSL tutoring. *

- 5
- 4
- (•) 3
- 2
- 1

The students feel confident in their ability to communicate using FSL after using the application. *

- 5
- (•) 4
- 3
- 2
- 1

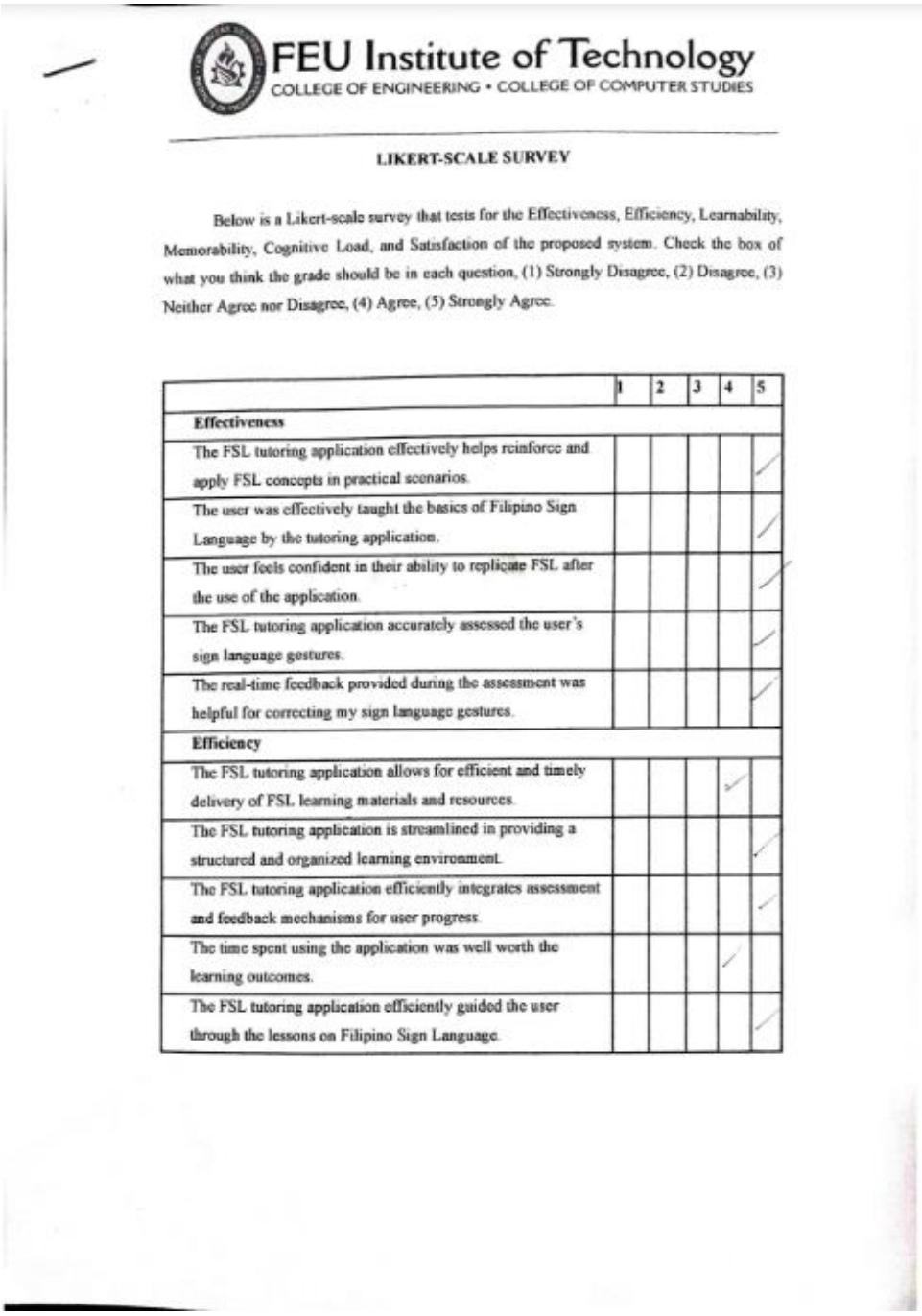
Traditional FSL tutoring has increased the students' interest in learning Filipino Sign Language. *

- ( ) 5
- ( ) 4
- (•) 3
- ( ) 2
- ( ) 1

This content is neither created nor endorsed by Google.

Google Forms

Likert Scale Survey for Tutoring Application



FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

LIKERT-SCALE SURVEY

Below is a Likert-scale survey that tests for the Effectiveness, Efficiency, Learnability, Memorability, Cognitive Load, and Satisfaction of the proposed system. Check the box of what you think the grade should be in each question, (1) Strongly Disagree, (2) Disagree, (3) Neither Agree nor Disagree, (4) Agree, (5) Strongly Agree.

	1	2	3	4	5
Effectiveness					
The FSL tutoring application effectively helps reinforce and apply FSL concepts in practical scenarios.					✓
The user was effectively taught the basics of Filipino Sign Language by the tutoring application.					✓
The user feels confident in their ability to replicate FSL after the use of the application.					✓
The FSL tutoring application accurately assessed the user's sign language gestures.					✓
The real-time feedback provided during the assessment was helpful for correcting my sign language gestures.					✓
Efficiency					
The FSL tutoring application allows for efficient and timely delivery of FSL learning materials and resources.				✓	
The FSL tutoring application is streamlined in providing a structured and organized learning environment.					✓
The FSL tutoring application efficiently integrates assessment and feedback mechanisms for user progress.					✓
The time spent using the application was well worth the learning outcomes.				✓	
The FSL tutoring application efficiently guided the user through the lessons on Filipino Sign Language.					✓

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

Learnability					
The user easily grasps and understands the FSL concepts through the FSL tutoring application.					✓
The FSL tutoring application is intuitive and clear with the learning pathways and structures.					✓
The lessons were presented in a way that was easy to understand and follow.					✓
The lessons provided sufficient guidance for learning new gestures.					✓
The user feels confident in their ability to follow along with the lessons provided by the FSL tutoring application.					✓
Memorability					
The user significantly demonstrates improved learning outcomes using the FSL tutoring application.					✓
The user notably shows progress and achievement through the FSL tutoring application.					✓
The user consistently shows learning progress through the assessments provided by the FSL tutoring application.				✓	
The user can easily recall and replicate the FSL gestures taught by the application.				✓	
The assessment feedback aided in reinforcing the memorization of gestures.					✓
Cognitive Load					
The application presented information in a way that was not overwhelming.					✓
The user did not feel mentally exhausted after using the application for learning.				✓	
The application effectively managed the amount of information presented at one time.					✓
The assessment process did not create an undue mental burden.					✓

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

The user feels the cognitive demands of using the application are reasonable.					
Satisfaction					
The user is satisfied with the systems and functionalities of the FSL tutoring application for teaching purposes.					
The user is likely to recommend the FSL tutoring application to other FSL educators.					
The user is satisfied with the overall learning experience provided by the application.					
The user feels confident in their ability to communicate using FSL after using the application.					
The application has increased the user's interest in learning the Filipino Sign Language.					

Validated by:

Ms. Mary Ann G. Perdon
Legarda Elementary School
FSL Teacher

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

LIKERT-SCALE SURVEY

Below is a Likert-scale survey that tests for the Effectiveness, Efficiency, Learnability, Memorability, Cognitive Load, and Satisfaction of the proposed system. Check the box of what you think the grade should be in each question, (1) Strongly Disagree, (2) Disagree, (3) Neither Agree nor Disagree, (4) Agree, (5) Strongly Agree.

	1	2	3	4	5
Effectiveness					
The FSL tutoring application effectively helps reinforce and apply FSL concepts in practical scenarios.					✓
The user was effectively taught the basics of Filipino Sign Language by the tutoring application.					✓
The user feels confident in their ability to replicate FSL after the use of the application.					✓
The FSL tutoring application accurately assessed the user's sign language gestures.				✓	
The real-time feedback provided during the assessment was helpful for correcting my sign language gestures.					✓
Efficiency					
The FSL tutoring application allows for efficient and timely delivery of FSL learning materials and resources.					✓
The FSL tutoring application is streamlined in providing a structured and organized learning environment.					✓
The FSL tutoring application efficiently integrates assessment and feedback mechanisms for user progress.					✓
The time spent using the application was well worth the learning outcomes.					✓

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

The FSL tutoring application efficiently guided the user through the lessons on Filipino Sign Language.					✓
Learnability					
The user easily grasps and understands the FSL concepts through the FSL tutoring application.					✓
The FSL tutoring application is intuitive and clear with the learning pathways and structures.					✓
The lessons were presented in a way that was easy to understand and follow.					✓
The lessons provided sufficient guidance for learning new gestures.					✓
The user feels confident in their ability to follow along with the lessons provided by the FSL tutoring application.					✓
Memorability					
The user significantly demonstrates improved learning outcomes using the FSL tutoring application.					✓
The user notably shows progress and achievement through the FSL tutoring application.					✓
The user consistently shows learning progress through the assessments provided by the FSL tutoring application.					✓
The user can easily recall and replicate the FSL gestures taught by the application.					✓
The assessment feedback aided in reinforcing the memorization of gestures.					✓
Cognitive Load					
The application presented information in a way that was not overwhelming.					✓
The user did not feel mentally exhausted after using the application for learning.					✓
The application effectively managed the amount of information presented at one time.					✓

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

The assessment process did not create an undue mental burden.					✓
The user feels the cognitive demands of using the application are reasonable.					✓
Satisfaction					
The user is satisfied with the system and functionalities of the FSL tutoring application for teaching purposes.					✓
The user is likely to recommend the FSL tutoring application to other FSL educators.					✓
The user is satisfied with the overall learning experience provided by the application.					✓
The user feels confident in their ability to communicate using FSL after using the application.					✓
The application has increased the user's interest in learning the Filipino Sign Language.					✓

Validated by:

Ms. Mary Ethelyne M. Marasigan
Legarda Elementary School
FSL Teacher

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

LIKERT-SCALE SURVEY

Below is a Likert-scale survey that tests for the Effectiveness, Efficiency, Learnability, Memorability, Cognitive Load, and Satisfaction of the proposed system. Check the box of what you think the grade should be in each question. (1) Strongly Disagree, (2) Disagree, (3) Neither Agree nor Disagree, (4) Agree, (5) Strongly Agree.

	1	2	3	4	5
Effectiveness					
The FSL tutoring application effectively helps reinforce and apply FSL concepts in practical scenarios.					✓
The user was effectively taught the basics of Filipino Sign Language by the tutoring application.				✓	
The user feels confident in their ability to replicate FSL after the use of the application.					✓
The FSL tutoring application accurately assessed the user's sign language gestures.				✓	
The real-time feedback provided during the assessment was helpful for correcting my sign language gestures.					✓
Efficiency					
The FSL tutoring application allows for efficient and timely delivery of FSL learning materials and resources.				✓	
The FSL tutoring application is streamlined in providing a structured and organized learning environment.					✓
The FSL tutoring application efficiently integrates assessment and feedback mechanisms for user progress.				✓	
The time spent using the application was well worth the learning outcomes.					✓
The FSL tutoring application efficiently guided the user through the lessons on Filipino Sign Language.					✓

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

Learnability					
The user easily grasps and understands the FSL concepts through the FSL tutoring application.					✓
The FSL tutoring application is intuitive and clear with the learning pathways and structures.				✓	
The lessons were presented in a way that was easy to understand and follow.					✓
The lessons provided sufficient guidance for learning new gestures.					✓
The user feels confident in their ability to follow along with the lessons provided by the FSL tutoring application.					✓
Memorability					
The user significantly demonstrates improved learning outcomes using the FSL tutoring application.					✓
The user notably shows progress and achievement through the FSL tutoring application.				✓	
The user consistently shows learning progress through the assessments provided by the FSL tutoring application.				✓	
The user can easily recall and replicate the FSL gestures taught by the application.					✓
The assessment feedback aided in reinforcing the memorization of gestures.					✓
Cognitive Load					
The application presented information in a way that was not overwhelming.					✓
The user did not feel mentally exhausted after using the application for learning.				✓	
The application effectively managed the amount of information presented at one time.					✓
The assessment process did not create an undue mental burden.					✓

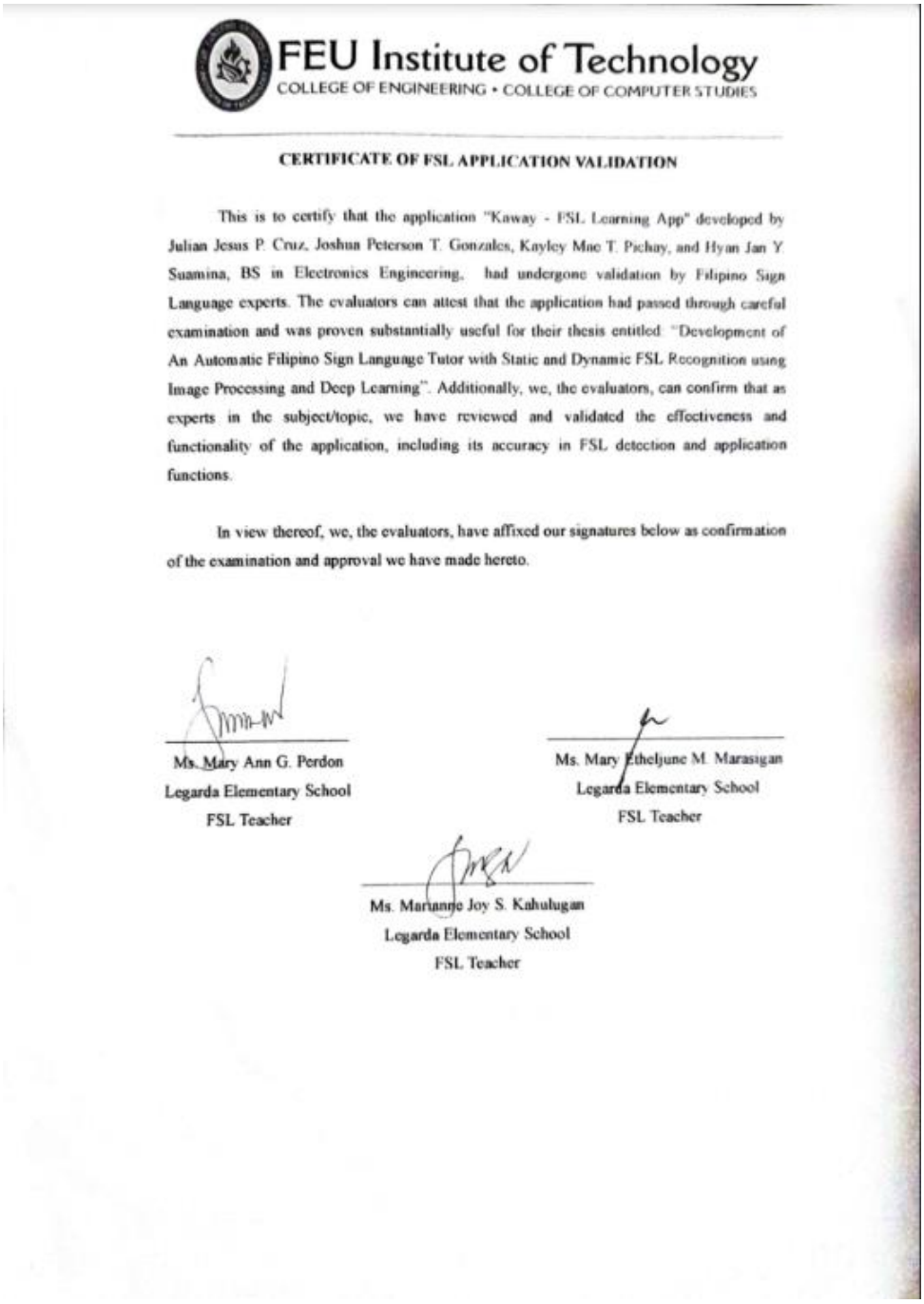
FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

The user feels the cognitive demands of using the application are reasonable.					✓
Satisfaction					
The user is satisfied with the system and functionalities of the FSL tutoring application for teaching purposes.					✓
The user is likely to recommend the FSL tutoring application to other FSL educators.				✓	✓
The user is satisfied with the overall learning experience provided by the application.					✓
The user feels confident in their ability to communicate using FSL after using the application.					✓
The application has increased the user's interest in learning the Filipino Sign Language.					✓

Validated by:

Ms. Marianne Joy S. Kabulagan
Legarda Elementary School
FSL Teacher

Certificate of FSL Application Validation



FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

CERTIFICATE OF FSL APPLICATION VALIDATION

This is to certify that the application "Kaway - FSL Learning App" developed by Julian Jesus P. Cruz, Joshua Peterson T. Gonzales, Kayley Mae T. Pichay, and Hyan Jan Y. Suamina, BS in Electronics Engineering, had undergone validation by Filipino Sign Language experts. The evaluators can attest that the application had passed through careful examination and was proven substantially useful for their thesis entitled: "Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processing and Deep Learning". Additionally, we, the evaluators, can confirm that as experts in the subject/topic, we have reviewed and validated the effectiveness and functionality of the application, including its accuracy in FSL detection and application functions.

In view thereof, we, the evaluators, have affixed our signatures below as confirmation of the examination and approval we have made hereto.

Ms. Mary Ann G. Perdon
Legarda Elementary School
FSL Teacher

Ms. Mary Etheljune M. Marasigan
Legarda Elementary School
FSL Teacher

Ms. Marianne Joy S. Kahulugan
Legarda Elementary School
FSL Teacher

Certification of FSL Curriculum Validation

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

CERTIFICATION OF FSL CURRICULUM VALIDATION

The study entitled "Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processing and Deep Learning", by the researchers Cruz, Julian Jesus P., Gonzales, Joshua Peterson T., Pichay, Kayley Mae T., Suamina, Hyan Jan Y., Bachelor of Science Electronics Engineering of FEU Institute of Technology, had sought out for the validation of the curriculum that will be used in the study.

This certificate is issued in connection to the request of the researchers for the conduct of their study.

Table for FSL Curriculum

Module 1: Alphabet Knowledge	Module 1: Alphabet Knowledge	Module 2: Proper Expression in Introducing Oneself	Module 3: Polite Greetings and Courteous Expressions	Module 4: Vocabulary Development
A	Ñ	Ako si / Ang pangalan ko ay	Magandang Umaga	Bahay
B	NG	Ano ang pangalan mo?	Magandang Hapon	Pinto
C	O	Sino?	Magandang Gabi	Silid
D	P	Ilang taon ka na?	Magandang Araw	Sala
E	Q		Kumusta ka	Kailan
F	R		Paalam	Guro
G	S		Ingat ka	Kusina

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

H	T		Maraming Salamat	Dilaw
I	U		Pasensya na	Ube
J	V			
K	W			
L	X			
M	Y			
N	Z			

Approved by:

Ms. Mary Ann G. Perdon
Legarda Elementary School
FSL Teacher

Ms. Mary Ethelyne M. Marasigan
Legarda Elementary School
FSL Teacher

Ms. Marianne Joy S. Kahulugan
Legarda Elementary School
FSL Teacher

Certificate of FSL Application Recognition Test Validation

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

CERTIFICATE OF FSL APPLICATION RECOGNITION TEST VALIDATION

The study entitled "Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processing and Deep Learning", by the researchers Cruz, Julian Jesus P., Gonzales, Joshua Peterson T., Pichay, Kayley Mae T., Suamina, Hyan Jan Y., Bachelor of Science Electronics Engineering of FEU Institute of Technology, had tested for the validation of the recognition and detection of FSL alphabet and phrases used in the study.

This certificate is issued in connection to the request of the researchers for the conduct of their study.

Module 1: Alphabet Knowledge	RECOGNIZED	NOT RECOGNIZED
A	///	
B	///	
C	///	
D	///	
E	///	
F	///	/
G	///	
H	///	
I	///	
J	///	

Validated by:

Ms. Mary Ann G. Perdon
Legarda Elementary School
FSL Teacher

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

K		
L		
M		
N		
Ñ		
NG		
O		
P		
Q		
R		
S		
T		
U		
V		
W		
X		
Y		
Z		

Validated by:

Ms. Mary Ann G. Perdon
Legarda Elementary School
FSL Teacher

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

Module 2: Proper Expression in Introducing Oneself	RECOGNIZED	NOT RECOGNIZED
Ako si		
Ano ang pangalan mo?		
Sino		
Ilang taon ka na?		

Module 3: Polite Greetings and Courteous Expressions	RECOGNIZED	NOT RECOGNIZED
Magandang Umaga		
Magandang Hapon		
Magandang Gabi		
Magandang Araw		
Kamusta ka		
Paalam		
Ingat ka		
Maraming Salamat		
Pasensya na		

Validated by:

Ms. Mary Ann G. Perdon
Legarda Elementary School
FSL Teacher

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

Module 4: Vocabulary Development	RECOGNIZED	NOT RECOGNIZED
Bahay		
Pinto		
Silid		
Sala		
Kailan		
Guro		
Kusina		
Dilaw		
Ube		

Validated by:

Ms. Mary Ann G. Perdon
Legarda Elementary School
FSL Teacher

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

CERTIFICATE OF FSL APPLICATION RECOGNITION TEST VALIDATION

The study entitled "Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processing and Deep Learning", by the researchers Cruz, Julian Jesus P., Gonzales, Joshua Peterson T., Pichay, Kayley Mae T., Suamina, Hyan Jan Y., Bachelor of Science Electronics Engineering of FEU Institute of Technology, had tested for the validation of the recognition and detection of FSL alphabet and phrases used in the study.

This certificate is issued in connection to the request of the researchers for the conduct of their study.

Module 1: Alphabet Knowledge	RECOGNIZED	NOT RECOGNIZED
A	✓✓✓✓✓	
B	✓✓✓✓✓	
C	✓✓✓✓✓	
D	✓✓✓✓✓	
E	✓✓✓✓✓	
F	✓✓✓✓	
G	✓✓✓✓✓	
H	✓✓✓✓ ✓	
I	✓✓✓ ✓✓	
J	✓✓✓✓✓	

Validated by:

Ms. Marianne Joy S. Kahulugan
Legarda Elementary School
FSL Teacher

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

K	✓✓✓✓✓	
L	✓✓✓✓✓	
M	✓✓✓✓✓	
N	✓✓✓✓✓	
Ñ	✓✓ ✓	
NG	✓✓✓✓	
O	✓✓✓✓✓	
P	✓✓✓✓✓	
Q	✓✓✓✓✓	
R	✓✓✓✓✓	
S	✓✓✓✓	
T	✓✓✓ ✓	
U	✓✓✓✓	
V	✓✓✓✓✓	
W	✓✓✓✓✓	
X	✓✓✓✓✓	
Y	✓✓✓✓✓	
Z	✓✓✓✓	

Validated by:

Ms. Marianne Joy S. Kahulugan
Legarda Elementary School
FSL Teacher

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

Module 2: Proper Expression in Introducing Oneself	RECOGNIZED	NOT RECOGNIZED
Ako si	✓✓✓✓✓	
Ano ang pangalan mo?	✓✓✓✓	
Sino	✓✓✓✓	
Ilang taon ka na?	✓✓✓✓✓	

Module 3: Polite Greetings and Courteous Expressions	RECOGNIZED	NOT RECOGNIZED
Magandang Umaga	✓✓✓✓✓	
Magandang Hapon	✓✓✓	
Magandang Gabi	✓✓✓✓✓	
Magandang Araw	✓✓✓✓	
Kumusta ka	✓✓✓✓✓	
Paalam	✓✓✓✓✓	
Ingat ka	✓✓✓✓✓	
Maraming Salamat	✓✓✓✓	
Pasensya na	✓✓✓✓✓	

Validated by:

Ms. Marianne Joy S. Kahulugan
Legarda Elementary School
FSL Teacher

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

Module 4: Vocabulary Development	RECOGNIZED	NOT RECOGNIZED
Bahay	✓✓✓✓	
Pinto	✓✓✓✓	
Silid	✓✓✓✓	
Sala	✓✓✓✓	
Kailan	✓✓✓✓	
Guro	✓✓✓✓	
Kusina	✓✓✓✓	
Dilaw	✓✓✓✓	
Ube	✓✓✓✓	

Validated by:

Ms. Marianne Joy S. Kahulugan
Legarda Elementary School
FSL Teacher

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

CERTIFICATE OF FSL APPLICATION RECOGNITION TEST VALIDATION

The study entitled "Development of An Automatic Filipino Sign Language Tutor with Static and Dynamic FSL Recognition using Image Processing and Deep Learning", by the researchers Cruz, Julian Jesus P., Gonzales, Joshua Peterson T., Pichay, Kayley Mae T., Suamina, Hyan Jan Y., Bachelor of Science Electronics Engineering of FEU Institute of Technology, had tested for the validation of the recognition and detection of FSL alphabet and phrases used in the study.

This certificate is issued in connection to the request of the researchers for the conduct of their study.

Module 1: Alphabet Knowledge	RECOGNIZED	NOT RECOGNIZED
A	✓ ✓ ✓ ✓	
B	✓ ✓ ✓ ✓ ✓	
C	✓ ✓ ✓ ✓ ✓	
D	✓ ✓ ✓ ✓ ✓	
E	✓ ✓ ✓ ✓ ✓	
F	✓ ✓ ✓ ✓	
G	✓ ✓ ✓ ✓ ✓	
H	✓ ✓ ✓ ✓ ✓	
I	✓ ✓ ✓ ✓ ✓	
J	✓ ✓ ✓ ✓ ✓	

Validated by:

Ms. Mary Ethelyne M. Marasigan
Legarda Elementary School
FSL Teacher

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

K	✓ ✓ ✓ ✓ ✓	
L	✓ ✓ ✓ ✓ ✓	
M	✓ ✓ ✓ ✓ ✓	
N	✓ ✓ ✓ ✓ ✓	
Ñ	✓ ✓ ✓ ✓	
NG	✓ ✓ ✓ ✓	
O	✓ ✓ ✓ ✓ ✓	
P	✓ ✓ ✓ ✓ ✓	
Q	✓ ✓ ✓ ✓ ✓	
R	✓ ✓ ✓ ✓	
S	✓ ✓ ✓ ✓	
T	✓ ✓ ✓ ✓	
U	✓ ✓ ✓ ✓	
V	✓ ✓ ✓ ✓	
W	✓ ✓ ✓ ✓ ✓	
X	✓ ✓ ✓ ✓ ✓	
Y	✓ ✓ ✓ ✓ ✓	
Z	✓ ✓ ✓ ✓	

Validated by:

Ms. Mary Ethelyne M. Marasigan
Legarda Elementary School
FSL Teacher

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

Module 2: Proper Expression in Introducing Oneself	RECOGNIZED	NOT RECOGNIZED
Ako si	✓ ✓ ✓ ✓ ✓	
Ano ang pangalan mo?	✓ ✓ ✓ ✓ ✓	
Sino	✓ ✓ ✓ ✓ ✓	
Ilang taon ka na?	✓ ✓ ✓ ✓ ✓	

Module 3: Polite Greetings and Courteous Expressions	RECOGNIZED	NOT RECOGNIZED
Magandang Umaga	✓ ✓ ✓ ✓ ✓	
Magandang Hapon	✓ ✓ ✓ ✓	
Magandang Gabi	✓ ✓ ✓ ✓ ✓	
Magandang Araw	✓ ✓ ✓ ✓ ✓	
Kumusta ka	✓ ✓ ✓ ✓ ✓	
Paalam	✓ ✓ ✓ ✓ ✓	
Ingat ka	✓ ✓ ✓ ✓ ✓	
Maraming Salamat	✓ ✓ ✓ ✓ ✓	
Pasensya na	✓ ✓ ✓ ✓ ✓	

Validated by:

Ms. Mary Ethelyne M. Marasigan
Legarda Elementary School
FSL Teacher

FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

Module 4: Vocabulary Development	RECOGNIZED	NOT RECOGNIZED
Bahay	✓✓ ✓✓✓	
Pinto	✓✓✓✓✓	
Silid	✓✓✓✓	
Sala	✓✓✓ ✓✓	
Kailan	✓✓✓✓	
Guro	✓✓✓✓✓	
Kusina	✓✓✓✓	
Dilaw	✓✓✓ ✓✓	
Ube	✓✓✓ ✓✓	

Validated by:

Ms. Mary Etheljune M. Marasigan
Legarda Elementary School
FSL Teacher

APPENDIX F: CURRICULUM VITAE OF RESEARCHERS

JULIAN JESUS P. CRUZ

BS Electronics Engineering

Objective

A determined electronics engineering student looking to leverage his knowledge and practical skills in a professional environment. Seeking to gain immersive real-world experience, collaborate with professionals, play a significant role in the advancement of innovative technologies, and further develop his expertise in the engineering field through an internship opportunity.

Accomplished Projects

- *Elevate: Race to Topnotch- ECESS Academic Competition* | **Junior Officer**
- *IECEP-MSC Academic Festival 2023- Interscholastic Quiz Show* | **Assistant Event Head**
- *IECEP-MSC Academic Festival 2023- Math Olympiad* | **General Secretary**
- *Navigating the Future: CNS-ATM Systems Unveiled – FEU-IT Seminars and Colloquium Project* | **Secretary/Host**
- *IECEP-MSC Elite Electronics Showdown* | **General Secretary**
- *IECEP-MSC Datathon* | **General Secretary**

Education

BS Electronics Engineering
FEU Institute of Technology
P. Paredes St., Sampaloc, Manila
Expected Graduation - 2024

High School
Nyongani Inc.
Blk. 12 Osmeña St., Interville III Subdivision,
Brgy. San Agustin, Novaliches, Quezon
City, Quezon City, Philippines
2020

Personal Information

Address	: #30 Magsaysay St., T.S Cruz Subdivision, Novaliches, Quezon City
Mobile	: 09165820197
E-mail	: cruzjulian004@gmail.com
Birthday	: July 4, 2002
Father	: Frederick G. Cruz
Mother	: Mel P. Cruz

Hardware and Software Proficiency

Adobe Photoshop
Microsoft Word
Microsoft Excel
Microsoft PowerPoint
Cisco
Java
AutoCAD
Multisim
Matlab
Arduino IDE
Arduino UNO

Awards

PinaSigla – Champion

Graduated High School with Honors

Certifications

CCNAv7 – Introduction to Networks - CISCO Certificate

ABS-CBN Work Immersion Certificate

Organizations

Member, Recreation and Athletics Club (RAC) | 2020 - 2021

Director for Events, Electronics Engineering Students Society (ECESS) | 2022 - Present

General Secretary, Institute of Electronics Engineering of the Philippines- Manila Student Chapter (IECEP-MSC) | Present

Other Interests

Sketching

Singing and dancing

Photo Editing

Swimming

FO-ACAD-IISC-002/15AUG2019/REV.1

Copyright 2023 © Julian Jesus Cruz

JOSHUA PETERSON T. GONZALES

BS Electronics and Communications Engineering

Objective

I am highly results driven and aggressive with personal goals. I am gearing towards a position which would help increase my skills in electronics, communication, and programming so that I can use it further in the electronics and communication engineering field.

Accomplished Projects

- *IECEP-MSC Academic Festival 2023 – Math Olympiad* | **Board of Director**
- *Navigating the Future: CNS-ATM Systems Unveiled – FEU-IT Seminars and Colloquium Project* | **Head Organizer/Technical Committee**
- *IECEP-MSC Academic Festival 2023- Interscholastic Quiz Show* | **Board of Director**
- *CRC-16* : a prototype of a CRC-16 wherein inputs can be solved | **CRC-16 Developer**
- *Line Follower Car*: a prototype of a line follower car wherein the car will be able to follow a black line | **Line Follower Car Developer**
- *Security Lock System:* a prototype of a security lock system wherein when scanned with a coded RFID card it will open | **Security Lock System Developer**
- *Audio Amplifier:* a prototype of an audio amplifier wherein it can be connected through an auxiliary wire | **Audio Amplifier Developer**
- *Color Sorting Machine:* a prototype of a color sorting machine using Arduino wherein the colors red, blue, yellow, and green can be sorted | **Color Sorting Machine Developer**

Education

BS in Electronics and Communication Engineering
FEU Institute of Technology
P. Paredes St., Sampaloc, Manila
Expected Graduation - 2024

High School
Immaculate Conception Cathedral School
39 Lantana St., Cubao Quezon City
2020

Personal Information

Address	:	30 Unit H Manhattan St., Cubao Quezon City
Mobile	:	09190614193
E-mail	:	gonzales.joshua0108@gmail.com
Birthday	:	January 8,2002
Father	:	John Q. Gonzales
Mother	:	Pamela Jean M. Talavera

Hardware and Software Proficiency

AutoCAD

Microsoft Word

Microsoft Excel

Microsoft PowerPoint

Arduino IDE

CISCO Packet Tracer

Awards

With Honors, Senior Highschool 2020

Certifications

CCNAv7 – Introduction to Networks – CISCO Certificate 2023

REX Bookstore Work Immersion 2020

Organizations

Director of Sports, Electronics Engineering Students' Society | 2023 – Present

Board of Director, Institute of Electronics Engineers of the Philippines – Manila Student Chapter | 2023 – Present

Other Interests

Basketball

Programming

Singing

Anime and Manga

FO-ACAD-HGC-002/15AUG2019/REV 1

Copyright 2022 © Joshua Peterson Talavera Gonzales

KAYLEY MAE T. PICHAY

BS Electronics Engineering

Objective

To obtain an Electronics Engineering internship, utilizing my proficiency in electronics, communications, and programming, in addition to my competence in written and verbal communication, organizational skills, and adaptability, with the aim to demonstrate efficient task completion, effective teamwork, and consistently high-quality customer service to contribute to the attainment of the company's goal

Accomplished Projects

- *NATIVE DEVICE IN SALT AND VINEGAR SOLUTION AS AN ALTERCHARGING A PHONE*: a special science project about an alternative device for charging | **Lead Researcher**
- *COMPARISON BETWEEN THE EFFECTIVENESS OF A HAND-WINDING CHARGER AND A COMMERCIAL CHARGER*: a special science project about the effectiveness of an alternative device for charging| **Lead Researcher**
- *"AY, ENGINEER KA MA'AM?"*: a comparative study on the work experience between male and female in terms of their social income status in the field of electronic engineering | **Lead Researcher**
- *A Comparative Study on Eco-Brick as a Multipurpose Building Block Material*: a research study focusing on an alternative eco-friendly building block material | **Lead Researcher**

Education

BS in Electronics Engineering (BSECE)
FEU Institute of Technology
P. Paredes St., Sampaloc, Manila
Expected Graduation - 2024

Senior High School
Science, Technology, Engineering, and Mathematics Strand
Dominican School Manila
1044 Lacson St, Sampaloc, Manila
2019

Personal Information

Address	:	Blk. 14 Lot 2 Rosas St. Pembo, Taguig City
Mobile	:	09999828076
E-mail	:	kayleymaep23@gmail.com
Birthdate	:	23 February 2002
Father	:	Ricky T. Pichay
Mother	:	Marchinet T. Pichay

Hardware and Software Proficiency

Adobe Express
Microsoft Word
Microsoft Excel
AutoCAD
Cisco Packet Tracer
MATLAB
NetBeans
Canva

Awards

Rank 4, Academic Excellence Award 2019

Mentor's List, Academic Excellence Award 2017

Rank 8, Academic Excellence Award 2013

Certifications

CCNAv7: Introduction to networks

Creotec Philippines Inc.: Work Immersion Completion

Organizations

Secretary, Electronics Engineering Students' Society | 2023 – Present

Finance Committee Board Member, Institute of Electronics Engineers of the Philippines – Manila Student Chapter | 2023 – Present

Other Interests

Singing

Drawing and Painting

Crocheting

Video Gaming

Anime, Manga, and Manhwa

Copyright 2023 © Kayley Mae T. Pichay

Hyan Jan Y. Suamina

BS in Electronics Engineering

Objective

To obtain a position as an intern and gain knowledgeable experience while developing skills necessary in the field of Electronics Engineering which includes improvement in software proficiency and its application in solving complex problems. Additionally, to effectively apply what I've learned into real-world applications and establish a network with professionals with the same goals.

Accomplished Projects

- *Piezoelectric Disc Circuit Driver*: a PCB that takes input from a DAC that amplifies voltage to drive a Piezo Disc | **Upwork Freelancer, Lead Designer**
- *Microchip and Network Systems Research*: research dedicated to pricing the tangible applications of microchips in data centers for tax calculations | **Upwork Freelancer, Lead Researcher**
- *Infinity Mirror Project*: a project plan containing the technical specifications and plans for the methodology in making a complex infinity mirror setup| **Upwork Freelancer, Lead Project Planner**
- *IECEP-MSC Acadfest 2023, Math Olympiad and Interscholastic Quiz Show*: an academic competition among ECE students in Manila| **Organizer, Board of Directors**
- *NASA Space Apps 2023 Managing Fire: Increasing Community-based Fire Management Opportunities, Project FIRESAFE*: a mobile application with image processing for immediate Fire Response and safety management | **Assistant Developer**
- *Navigating the Future: CNS-ATM Systems Unveiled*: a seminar discussing the communication systems used in air traffic | **Co-Organizer, Documentation Head**
- *PagECEpan* : an ECESS Outreach Program for Recto High School students | **Organizer, Creatives Committee**
- *Elevate: Race to Topnotch*: an academic competition among Manila ECE students | **Organizer, Creatives Committee**
- *Forza 2023: ECESS General Assembly*: a general assembly for ECESS | **Organizer, Creatives Committee**

Education

BS in Electronics Engineering

FEU Institute of Technology
P. Paredes St., Sampaloc, Manila
Expected Graduation - 2024

High School

International Philippine School in Jeddah
Gharnatah St., Jeddah, KSA
2020

Personal Information

Address	:	Block 3, Lot 21, 1st Avenue, Larlin Village, Apalit, Pampanga
Mobile	:	09694181458
E-mail	:	hyanjansuamina@gmail.com
Birthday	:	January 4, 2002
Father	:	Hernani A. Suamina
Mother	:	Rina Y. Suamina

Hardware and Software Proficiency

Arduino IDE
AWS
Raspberry Pi
STM32
Proteus
Altium Designer
KiCAD
Multisim
MATLAB
AutoCAD
Cisco Packet Tracer
Python, Java, C
Adobe Premiere Pro
Adobe After Effects
Adobe Photoshop
Adobe Lightroom
Blender

Awards

Participant, NASA Space Apps 2023

Participant, Broadkast ng Talino 2023

Champion, PINASigla 2023

Top Performing Student Scholar 2022

Best Research Presenter 2020

Certifications

CCNAv7: Introduction to Networks

CS50x Introduction to Computer Science

IELTS 8.0 English Proficiency

NASA Space Apps 2023

Organizations

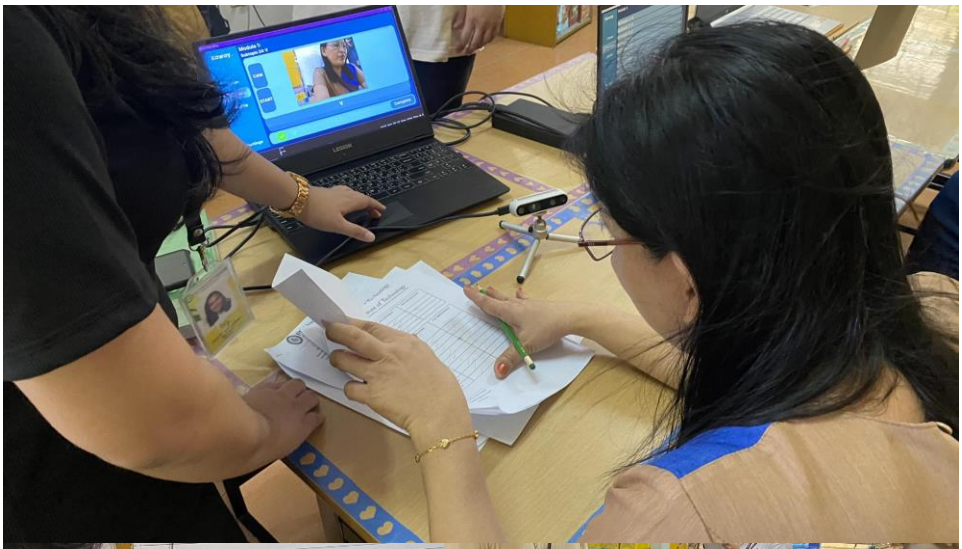
Team Manager, Shell Eco Marathon | 2023 – Present

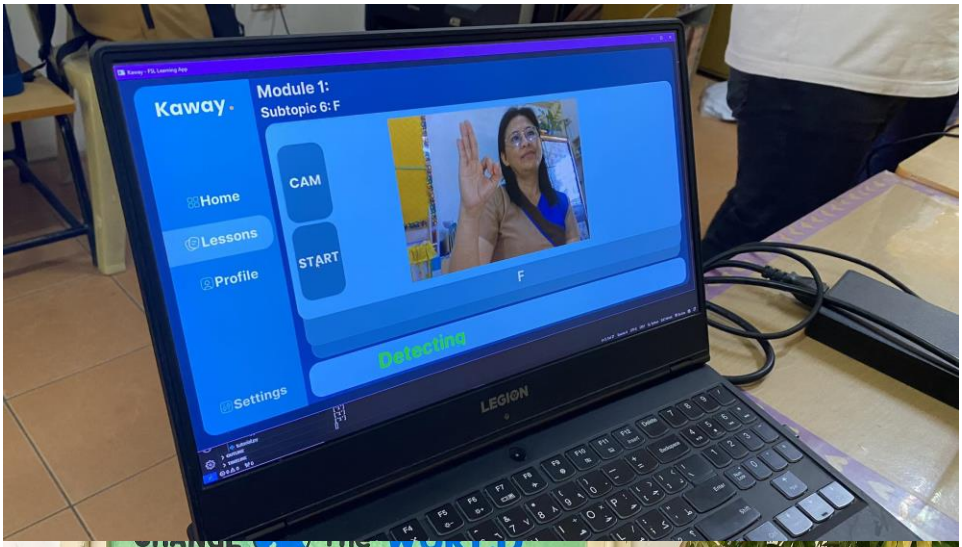
Director for Creatives, Electronics Engineering Students' Society | 2023 – Present

Board of Director, IECEP - MSC| 2023 – Present

Copyright 2023 © Hyan Jan Y. Suamina

APPENDIX J: DOCUMENTATION OF TESTING BY FSL EXPER




Kaway
Module 1:
Subtopic 6: F
Home
Lessons
Profile
Settings
CAM
START
F
Detecting
LEGION


NELSON Mandela
I SpED
TODAY
A READ
TOMORROW