MALAYSIAN SIGN LANGUAGE & ALPHABETS RECOGNITION USING DEEP LEARNING ALGORITHM AND IMAGE PROCESSING TECHNIQUE

MUHAMMAD FAUZAN BIN ABDUL HAKIM

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

MALAYSIAN SIGN LANGUAGE & ALPHABETS RECOGNITION USING DEEP LEARNING ALGORITHM AND IMAGE PROCESSING TECHNIQUE

MUHAMMAD FAUZAN BIN ABDUL HAKIM

This report is submitted in partial fulfillment of the requirements for the degree of Bachelor of Electronic Engineering with Honours

Faculty of Electronic and Computer Engineering
Universiti Teknikal Malaysia Melaka

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2023

DECLARATION

I declare that this report entitled “Malaysian Sign Language & Alphabets Recognition using Deep Learning Algorithm and Image Processing Technique” is the result of my own work except for quotes as cited in the references.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Signature : ……

Author : Muhammad Fauzan Bin Abdul Hakim

Date : 12/01/2023

APPROVAL

I hereby declare that I have read this thesis and in my opinion, this thesis is sufficient in terms of scope and quality for the award of Bachelor of Electronic Engineering with Honours.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Signature : .....

Supervisor Name : Assoc. Prof. Dr. Masrullizam Bin Mat Ibrahim

Date : 12/01/2023

DEDICATION

To my beloved family.

UTeM
اونيۏرسيتي تيکنيکل مليسيا ملاک
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

ABSTRACT

People who suffer from hearing difficulties use sign language as a way of communication and sign language translators quickly become inadequate to serve the entire deaf community, especially in Malaysia. To address the problem, this project aims to develop a Malaysian Sign Language recognition algorithm and translate it into text form. To achieve the objective, a dataset of Malaysian Sign Language and alphabets are constructed. Furthermore, image processing techniques to extract specific landmarks were used. The developed algorithm is trained using CNN architecture and PyCharm software is used to perform real-time gesture translation into text form. The algorithm shows a promising result with an accuracy of 96.16%. In addition, the result of precision, recall, and F1-Score for every predicted class is as high as 100%.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

ABSTRAK

Orang yang mengalami masalah pendengaran menggunakan bahasa isyarat sebagai cara komunikasi dan penterjemah bahasa isyarat dengan cepat menjadi tidak mencukupi untuk berkhidmat kepada seluruh masyarakat pekak, terutamanya di Malaysia. Untuk menangani masalah tersebut, projek ini bertujuan untuk membangunkan algoritma pengecaman Bahasa Isyarat Malaysia dan menterjemahkannya ke dalam bentuk teks. Untuk mencapai objektif tersebut, set data Bahasa Isyarat Malaysia dan abjad dibina. Tambahan pula, teknik pemprosesan imej untuk mengekstrak tanda tempat tertentu telah digunakan. Algoritma yang dibangunkan dilatih menggunakan seni bina CNN dan perisian PyCharm digunakan untuk melakukan terjemahan gerak isyarat masa nyata ke dalam bentuk teks. Algoritma menunjukkan hasil yang menjanjikan dengan ketepatan 96.16%. Di samping itu, hasil ketepatan, ingat semula dan F1-Score untuk setiap kelas yang diramalkan adalah setinggi 100%.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

ACKNOWLEDGEMENTS

Alhamdulillah, all praise be to Allah S.W.T. the Provider of Guidance, for lending me the strength and wisdom to complete this thesis.

I express my deepest appreciation to my supervisor, Associate Professor Dr. Masrullizam Bin Mat Ibrahim for his countless efforts and patience in helping me to complete this project. I am very grateful to my family, especially my parents, Abdul Hakim Bin Abdul Hamid and Marziana Es, Binti Abdul Manaf for their continuous support and encouragement throughout my studies. My special thanks go to the lecturers, friends, and all those involved in my project directly or indirectly. Last but not least, great appreciation must go to the Universiti Teknikal Malaysia Melaka (UTeM) for providing great facilities, amazing educators, and meaningful memories for me throughout my studies.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

TABLE OF CONTENTS

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

UTeM
اونيۏرسيتي تيکنيکل مليسيا ملاک
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF FIGURES

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

UTeM
اونيۏرسيتي تيکنيکل مليسيا ملاک
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF TABLES

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF SYMBOLS AND ABBREVIATIONS

ANN	:	Artificial Neural Network
CNN	:	Convolutional Neural Network
CSV	:	Comma-Seperated Values
GUI	:	Graphical User Interface
IDE	:	Integrated Development Environment
LSTM	:	Long Short-Term Memory
MB	:	MegaByte
MSL	:	Malaysian Sign Language
OS	:	Operating System
PC	:	Personal Computer
ReLu	:	Rectified Linear Unit
RGB	:	Red Green Blue
RNN	:	Recurrent Neural Network
RoI	:	Region of Interest
SDG	:	Sustainable Development Goal
SLR	:	Sign Language Recognition

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF APPENDICES

UTeM
اونيۏرسيتي تيکنيکل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CHAPTER 1

INTRODUCTION

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

1.1 Overview

The deaf community cannot communicate with others through verbal communication. Since they do not possess the ability to hear, they lose the sense to control their voice. Hence, they have been using sign language to convey their intention, statement, feeling, and others. In Malaysia, the deaf community uses Malaysian Sign Language as the official sign language. However, sign language itself is typically learned by the deaf community only. The lack of awareness and effort from other communities to learn sign language causes the number of individuals who are not deaf that understand sign language still low. Communications between the deaf and others are inevitable since the deaf community still needs to attend court, counsel, or handle official matters. Hence, this complication raises the demand for sign language translators.

The sign language translator in Malaysia is trained to translate the Malaysian Sign Language into Bahasa Melayu. They provide the service of translation when the deaf community needed them. However, in Malaysia, the number of sign language translators is too low compared to the number of deaf people. Hence the process of acquiring a sign language translator becomes difficult. The service is typically needed to be booked and sometimes consumes a lot of time. Therefore the efficiency of the service is low since it cannot fulfill the service on demand. As a result, researchers have comes up with a better solution to automate the translation of sign language using artificial intelligence (A.I.) to reduce the need for a human sign language translator.

1.2 Problem Statement

The deaf community cannot communicate with people who do not understand Malaysian Sign Language. Typically, the Malaysian deaf community will learn Malaysian Sign Language from school. It enables them to communicate with each other and anyone who understands the language. However, the number of non-deaf people who understand this language is really low. This creates a communication gap between the deaf community and the non-deaf community. Since there are a lot of people who are reluctant to learn Malaysian Sign Language, there is a need to develop a device that can quickly translate Malaysian Sign Language without learning it.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Sign Language Recognition (SLR) system using a sensor-based system provide inconvenience restraint to the user and is expensive. There have been numerous efforts to develop a device that can translate sign language. Among them is the use of a sensor-based system. These sensors are attached to the signer to extract information during the signing process. However, the study shows that the implementation of the system makes the signer feel inconvenienced and restrains their movement. This

discourages the deaf community from fully committing to using the system. In addition, the number of sensors used can increase significantly when a higher accuracy of information from the signers is needed. Hence, it will increase the cost of the system. Therefore, an alternative to the sensor-based system, such as a vision-based system can be a solution to the problem.

There is a lack of research and system development focused on Malaysian Sign Language recognition. As for 2021, the number of research focused on Malaysian Sign Language Recognition made up less than 1% of the accumulated research for sign language recognition across the world [1]. The highest number of research is held by the American Sign Language Recognition system which is 32% as in Figure 1.1. This indirectly shows that the number of collected data to be used in Malaysian Sign Language Recognition is significantly lower compared to the rest of the world. Therefore, there is a need to support the lack of advancement in the area of the Malaysian Sign Language Recognition System. One of the solutions is to construct the Malaysian Sign Language dataset.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

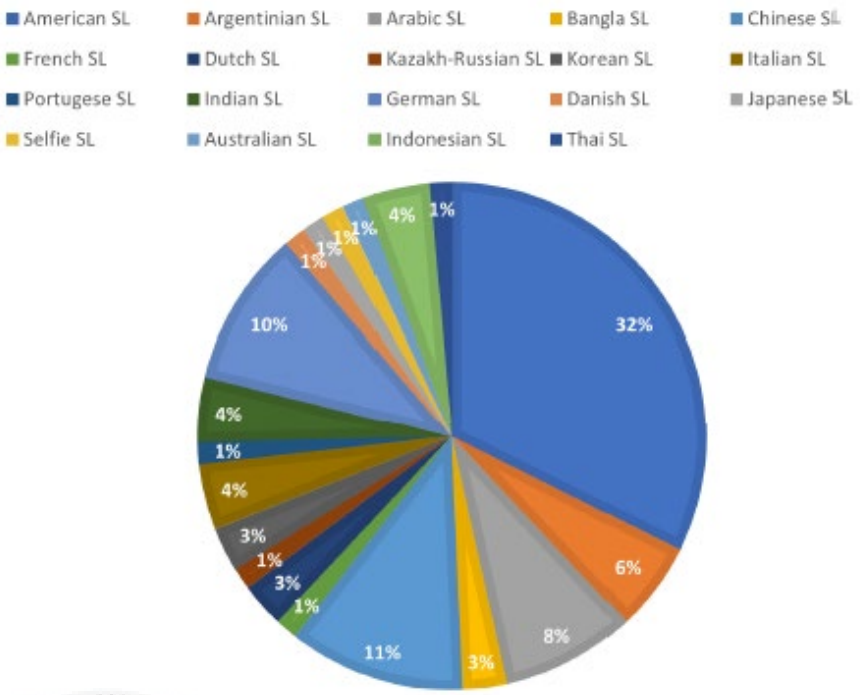


Figure 1.1: Number of publications based on sign language recognition by language [1].

1.3 Objectives and Scopes of Project

1.3.1 Project Objectives

The objectives of the project are as follows:

i. To construct Malaysian Sign Language datasets with the label.

ii. To develop a Malaysian Sign Language Recognition and alphabets algorithm using deep learning models and image processing techniques.

iii. To convert the Malaysian Sign Language and alphabets into text form.

iv. To analyze the performance of the Malaysian Sign Language Recognition algorithm.

1.3.2 Scopes of Project

To develop an artificial intelligence that can recognize Malaysian Sign Language and the alphabet, a set of data is collected. The data is a static image of people that

perform a specific Malaysian Sign Language and alphabet gesture. Hence, the device to capture the image is determined. To improve the effectiveness of the algorithm to be applied on a personal computer (PC), the camera of the PC itself is used to capture all of the images. Although the built-in PC camera has a lower quality compared to other types of cameras, this approach is used to ensure that the neural network is trained for the worst-case scenario.

Even though the initial data is accumulated in image format, the image then is processed to extract the hands and face landmarks. These landmarks then are used as the input data for the neural network. Instead of training the neural network based on the original image, the neural network is trained based on specific finger joints and facial features that are traced on an empty canvas. This approach is taken because the focus of this project is to classify a gesture according to Malaysian Sign Language and alphabet, hence the finger position and the facial features extraction are cleared before feeding to the neural network.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

The neural network was trained based on a deep learning model. The deep learning model eliminates some of the pre-processing problems. Furthermore, the deep learning model can recognize features from its dataset without human interference. This project has used the Convolutional Neural Network (CNN) architecture in the neural network. This is because CNN architecture is excellent at finding strong features in image recognition.

Another scope of the project is that the algorithm can recognize 24 alphabets and 5 words in Malaysian Sign Language. Malaysian Sign Language can be divided into two categories which are the static sign and the dynamic sign. The static sign does not involve a moving gesture while the dynamic sign does. The static sign can be captured

as an image while the dynamic sign needs to be recorded as a video. To limit the project expectation, this project focus only on the static gesture. Hence, the number of gestures is determined based on 5 common words as in Figure 1.3, and the alphabet for Bahasa Melayu as in Figure 1.2 except for the letter 'J' and 'Z' which are dynamic signs.

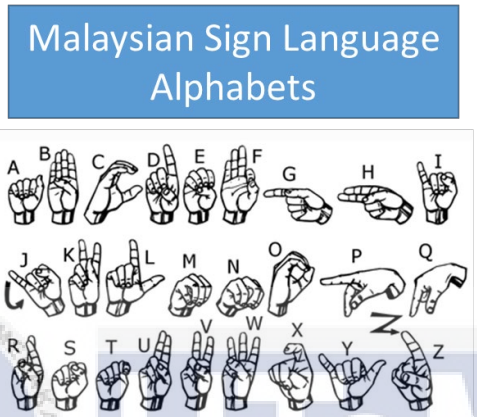


Figure 1.2: The scope of this project includes the Malaysian Sign Language alphabet except for the letter 'J' and 'Z'.

Figure 1.3: The five gestures that are also included in the project scope.

The project has been developed using PyCharm IDE with Python programming language. Although many IDE and programming languages can be used to develop the project such as MatLab, this mentioned approach is selected due to several reasons. Python language is the most popular programming language as of 2022. For this reason, the Python language is selected in hope that it can be easily incorporated into

existing technology. Besides, since Python programming has a large developer community, this project has a higher chance of receiving continuous support and improvement in the future. To demonstrate the prediction of the neural network, a PC application has been developed. The application has been developed using the PyCharm IDE. The application functions to display the live feed of the signer. In real-time, the gesture of the signer is translated into text form, specifically in Bahasa Melayu.

In addition, the algorithm developed has undergone a certain performance analysis. The target of this analysis was to determine the strength and weaknesses of the model. However, the analysis has not been trying to determine the most optimized model for this application but rather to identify points that can be improved when designing this application while using a CNN model.

1.4 Organization of Thesis

There are five (5) chapters organized in this thesis. Each chapter deals with different aspects of the project. Each chapter starts with a short introduction, the content of the chapter, and finally end with its appropriate conclusion. The following is a short overview of each of the chapters.

The first chapter of this thesis explains the background of the project focusing on the need for the Malaysian Sign Language Recognition System. It also describes the scope, objectives, and problem statements of this research. Next, the second chapter expounds on the nature of Malaysian Sign Language. The history, characteristics, and important features that can be extracted from the signer are described. Later, it explains the current trend in Sign Language Recognition Systems (SLR). It also relates the details of the SLR architecture which consists of current technologies and

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

algorithms used. Afterward, the third chapter explains the details of techniques to achieve the project objectives. The process starts with collecting images for the dataset. Later, the neural network based on CNN architecture is developed. Following that, a GUI is developed to allow the user to test the developed model in real-time. Finally, the model is analyzed to determine its efficiency. Then, the fourth chapter provides the result that has been gathered throughout the development of this project. A thorough analysis is done of the developed model and some issues are discussed. Lastly, chapter five covers the overall summary of this project's achievement. Besides, it also discussed the project's impact and potential commercialization. Finally, some future improvements are suggested.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CHAPTER 2

BACKGROUND STUDY

UTeM
اونيۏرسيتي تيکنيکل مليسيا ملاک
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2.1 Introduction

The number of hearing-impaired people in Malaysia has grown to more than 42,000 as of 2022 [2]. Hence, accommodating the needs of these people has become more crucial than ever. The problem of communication typically occurs when deaf people need to communicate in court, official matters, education, and counseling. Hence, many efforts to develop the best Sign Language Recognition (SLR) system can be seen throughout the years. The SLR system hopefully can increase the engagement or interaction between the deaf community with others. To develop the SLR, the Malaysian Sign Language characteristics needed to be identified first. These characteristics can then be utilized with the use of modern technology such as cameras,

accelerometers, and others. The critical part of this system is to develop the appropriate algorithm to decode the sign language.

2.2 Malaysian Sign Language (MSL)

2.2.1 Overview

The philosophy of Total Communication from the USA is brought to Malaysia in 1976 by Frances Parsons [3]. The philosophy encourages children with hearing loss to embrace all ways of communication including formal signs, natural gestures, lipreading, and more. Parsons gave a 6-week course about the philosophy to the Federation School for the Deaf in Penang. In 1978, the philosophy is implemented and a book with approximately 3500 sign words are assigned to Malay base words for educational reasons. Among them, there were 500 American Sign Language sign words adopted to Malaysian Sign Language. Finally, the Malaysia Federation of Deaf was established in 1998 to compile used sign words from the deaf community of Malaysia. Moreover, the compiled sign words are published as a Bahasa Isyarat Malaysia book in 2000 and later republished in 2003 and 2016.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2.2.2 Body Parts Involved in Sign Language

Before one can develop the best SLR system, the nature of general sign language itself needed to be understood first. The major body part involves in conveying sign language consist of facial expression, body posture, and hand gestures [4]. The facial expression is associated with the non-manual sign. This includes the expression of feelings from the signer. Body posture is a major component of sign language since the movement of hand gestures will affect the position of body posture. Figure 2.1 shows one of the methods to categorize sign language, which can be too complicated to comprehend.

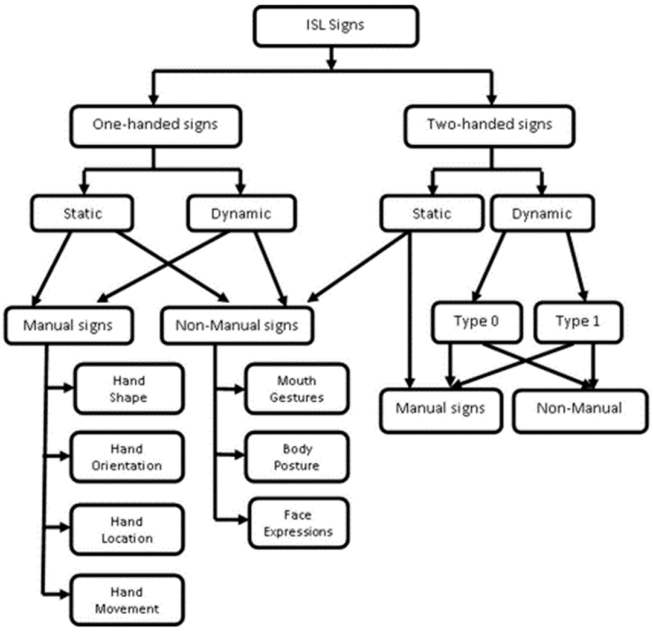


Figure 2.1: Hierarchy of typical sign language [4].

2.2.3 Static/Dynamic Motions in Sign Language

However, to allow the Sign Language Recognition (SLR) system to be developed with less complicated influence by the typical sign language hierarchy, the typical sign language can be categorized that focusing on the hand gesture first. The hand gesture is first classified based on whether there is a movement of the hand or not. In short, the gesture is classified as static and dynamic motion. Then it can be further classified based on the number of hands used in expressing sign language. Therefore, as in Figure 2.2, sign language can also be listed as a static-one-hand gesture, static-two-hand gesture, dynamic-one-hand gesture, and dynamic-two-hand gesture [5]. Nevertheless, a static gesture can easily be identified based on a single frame (image), while a dynamic gesture needed multiple frames (video) to be identified in image processing techniques.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

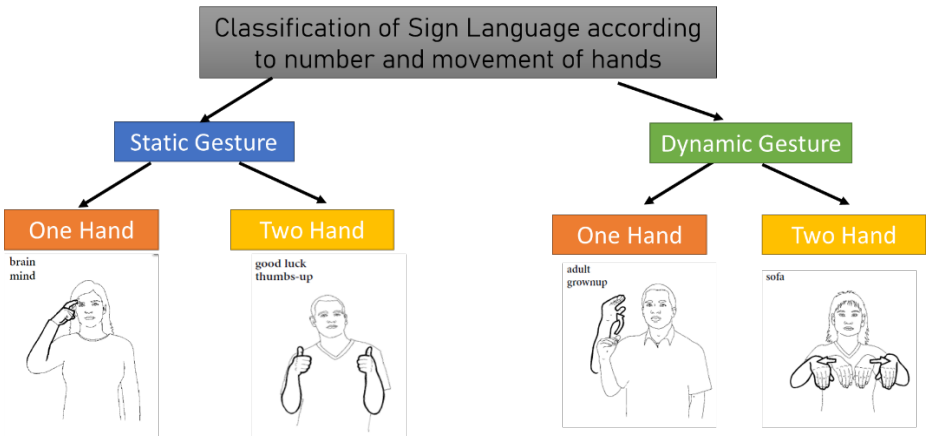


Figure 2.2: Classification of typical sign language according to the number and movement of hands.

2.2.4 Signing Space

To further determine the parameter of the SLR system, it needs to be identified the area of signing space. A typical sign language gesture will utilize the space of the head, neck, body, arm, hand, and some neutral space [6]. It means that the signing space can be virtually visualized as an approximate rectangle. The rectangle starts from the waist height to slightly above one's head. While the width of the rectangle should be less than 30cm to the left or right of one's torso. Based on this information, it can be determined that if one wishes to watch a sign language being performed, the interested body part of the signer is between a few tens of cm from left to right and from the waist to the upper head of the signer as in Figure 2.3.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

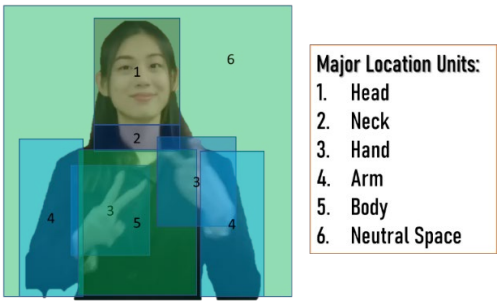


Figure 2.3: Typical signing space used by the signer.

2.2.5 Summary of the Characteristics of Malaysian Sign Language

Malaysian Sign Language is brought and adopted from America in 1976. The sign language is compiled and published with approximately 3500 sign words. A signer will use multiple body parts such as facial expression, body posture, and hand gestures. It creates a complicated hierarchy for a computer to decode. Hence, an approach to only focus on hand gestures is acknowledged. To be able to capture the appropriate data from the signer, a signing space is determined. It includes the region of the head until the waist.

2.3 Dataset for Sign Language Recognition algorithms

2.3.1 Related Works and Dataset on Sensor-Based SLR system

Based on the pieces of information regarding the needs and challenges of the signer, an SLR system can be developed. The SLR architecture can be categorized into two main classifications based on its input: sensor-based and vision-based. One research used smart gloves to acquire measurements such as the positions of hands, joints orientation, and velocity using microcontrollers and specific sensors such as accelerometers, flex sensors, and others [7]. There are other approaches to capturing signs by using motion sensors, such as electromyography (EMG) sensors, Kinect sensors, leap motion controllers, or their combinations [8] [9] [10]. Based on Figure 2.4, it can be seen that most approaches to increase the algorithm accuracy may increase the number of sensors used and the cost of the system.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

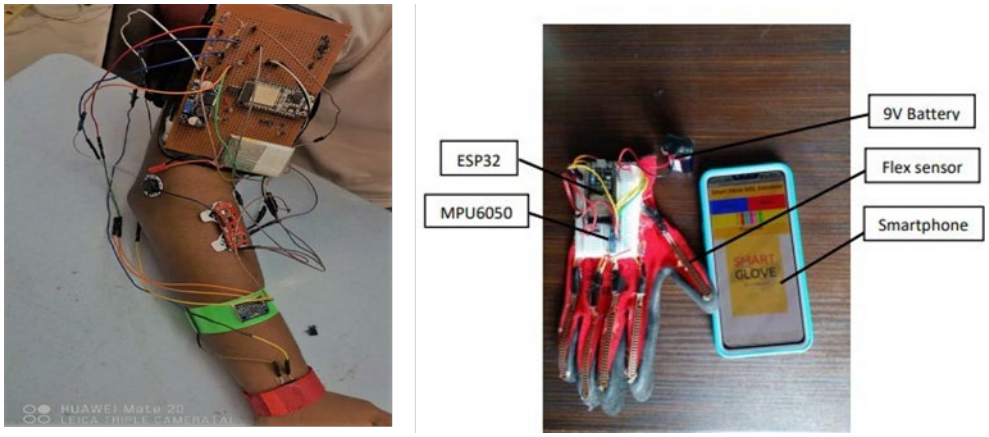


Figure 2.4: Examples of using a sensor-based SLR system.

Nevertheless, when using a sensor-based SLR system, the dataset that needs to be collected can be based on the value of acceleration, angular velocity, degree of rotation, and others. For example, a study collected the value of resistance from 5 flex sensors that are attached to a glove [11]. The study carried out by the researcher shows that the dataset contains 5000 samples representing 10 types of gestures. The gestures are the numerical gesture representing the numbers 0 to 9. Using the dataset, the developed algorithm can only reach 86-87% accuracy.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2.3.2 Related Works and Dataset on Vision-Based SLR

In recent years, research on SLR systems has focused more on vision-based methods because they provide little to no restraints on users, unlike sensor-based approaches. It overcomes the burden of wearing any kind of gloves providing more flexibility to the user than sensor-based systems [1]. The vision-based system utilizes the use of a camera such as a mobile phone camera, webcam, and others. This is one of the advantages of using a vision-based SLR system, as one study manage to construct the sign language dataset using only iPhone 6 as in Figure 2.5 [12]. The idea of image-based systems is to use image-processing techniques and A.I. to recognize

signs. However, the accuracy of the trained model can drop due to the skin tone, type of cloth, and the inclusion of the face.

Figure 2.5: Example use of vision-based input that was recorded using the iPhone 6 [12].

But recent advancement in A.I. allows hands, face, and pose detection to be detected easily through various landmarks [13]. This increases the interest among the researcher to include tracking the hand along with other parts of the body to differentiate Sign Language gestures. In essence, using body features improved the recognition of sign language by 2.27% [1]. These observations can be attributed to the fact that body joints are more dependable and robust than hand joints. Besides, for Malaysian Sign Language, certain signs can only be recognized if the hand position on the face and the body pose is included in the prediction as in Figure 2.6.

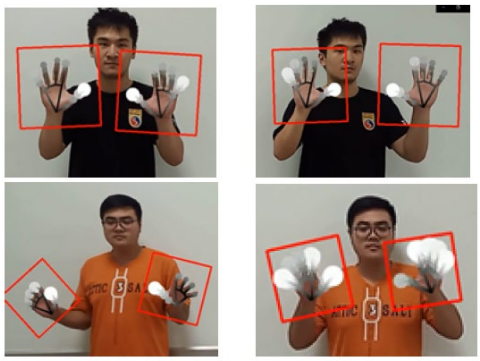


Figure 2.6: Example use of vision-based input that was recorded using the video camera with an accuracy of around 85 percent with 50 epochs [13].

However, when using a vision-based SLR system, the dataset usually only consists of images or videos. For example, a study that only uses a camera to record its dataset has collected 35,000 images [14]. The images represent 100 different gestures comprised of gestures for alphabets, numbers, and commonly used American Sign Language words. There are pre-processing techniques that have been done on the collected images. The technique involves, rescaling the image and normalization of its pixel values.

Another study uses a 0.9MP HP Wide Vision HD laptop webcam to record videos for collecting its dataset [15]. The webcam recorded various videos that consist of RGB frames and their depth images. The reason the dataset is collected in such a manner is due to its pre-processing technique. The images undergo skin segmentation and image background reduction. However, the dataset manages to collect 120,000 images representing the alphabet of Malaysian Sign Language except for the letter 'J' and 'Z'.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2.4 Sign Language Recognition algorithm

2.4.1 SLR algorithm based on Machine Learning

In order to predict Sign Language, a specific algorithm need to be developed. The prominent approach to be used is based on Machine Learning. For Supervised Machine Learning, the features of the images need to be extracted and processed before the system can be trained. Research shows that applying the right processing technique such as the Histogram of Oriented Gradient can increase the accuracy of prediction as in Figure 2.7 [16]. Nevertheless, with larger and more diverse datasets, simple machine learning methods tend to underperform, which is why many of the more sophisticated models are based on the deep learning model [1].

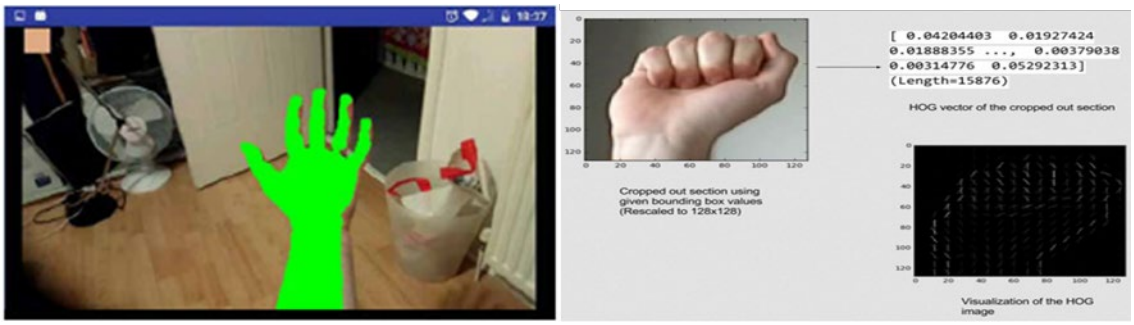


Figure 2.7: Machine Learning approach with Histogram of Oriented Gradient preprocessing applied [16].

2.4.2 SLR algorithm based on Deep Learning

An artificial neural network is typically used to predict sign language in the deep learning approach. One of the artificial neural networks is the Recurrent Neural Network (RNN). The neural network advantage is the feedback loop in the hidden layers acting as a memory for the next input step. Although Recurrent Neural Network is more known to predict temporal sequence data such as Neuro-Linguistic Programming (NLP), text, voice, video, or time-series data, it is also used in SLR systems. Research that uses different types of RNN such as Long-Short Term Memory (LSTM), and Bi-LSTM shows that it can achieve more than 97% accuracy. The research focus on using RNN over CNN since the extracted hand key point's position by using the MediaPipe framework is saved in a text file format (CSV) corresponding to the reference position [17].

A convolutional neural network (CNN) is one of the most popular deep learning algorithms, comprising several convolutional layers and then followed by one or more activation layers before fully connected. The fundamental reason to use a CNN architecture is due to its capability to extract features from data with convolution structures [18]. Using CNN gives multiple advantages. It is effective in reducing parameters and speeding up network convergence. Besides, a group of connections in

a neural network can share the same weights, which reduces parameters further. Finally, one of the techniques in CNN uses pooling which down-sample an image, which reduces the amount of data while retaining useful information. Research shows that using CNN allows the prediction to achieve more than 99% accuracy [19]. The CNN used consists of six convolution layers, three pooling layers, and a fully connected layer besides the input-output layers. It is well known that the CNN model typically excels at image recognition.

2.5 Real-time Sign Language Conversion into Text Form

After the appropriate algorithm for Sign Language Recognition is established, the algorithm should be tested on a real-time system. This allows developers to study whether the algorithm can be implemented on any device. Some studies may implement the algorithm as a mobile application. Other studies may use and implement it as a PC application. However, the effectiveness of the application is influenced by the delay time between the gesture being performed and the predicted gesture being displayed. For example, an algorithm has been developed and implemented as a PC application to perform a real-time gesture prediction [20]. The project uses PyCharm software and the program was written in Python language. The result of the interface is as in Figure 2.8. The project also uses the TensorFlow library for its backend GUI. The system captures 30 frames per second but only displays the predicted gesture every 3 seconds. The model uses 1 frame for every second and allows the algorithm to be fed with at least 3 frames before it displays the predicted gesture.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

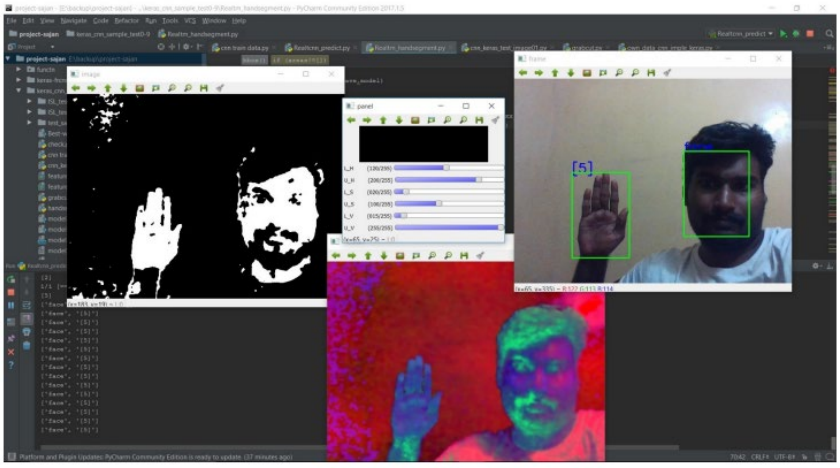


Figure 2.8: Real-time sign language conversion application for Indian Sign Language [20].

2.6 Performance Analysis Method for Artificial Intelligence in SLR

Researchers have used various methods to be able to determine whether an SLR algorithm is effective. The primary concern is centered on the model's capability to distinguish between one gesture to another. Consequently, typical studies use the percentage-based accuracy method. This allows them to quickly make a general assumption about the model's performance based on the training and testing alone. Some researchers tend to analyze further by evaluating the precision, recall, and the F1-Score. However, most studies show that the initial method to evaluate their model is to construct the Confusion Matrix [1]. From there, the above-mentioned detailed analysis can be performed.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2.7 Summary

The Sign Language Recognition System is celebrated among researchers around the world to help the deaf community. The way to collect datasets for the SLR system is depending on the type of the system itself. For a sensor-based SLR system, the expected dataset consists of numerical data. While the vision-based SLR system uses images or videos as its dataset. The development of SLR based on machine learning

or deep learning approaches can achieve an accuracy of up to 99%. Both methods provide different advantages and disadvantages. Among the popular algorithm, used are CNN, LSTM, and others. The developed algorithm can also be used to create an application to make a real-time prediction. Finally, the performance analysis for an SLR system includes the finding of precision, recall, and F1-Score.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CHAPTER 3

METHODOLOGY

3.1 Research Methodology

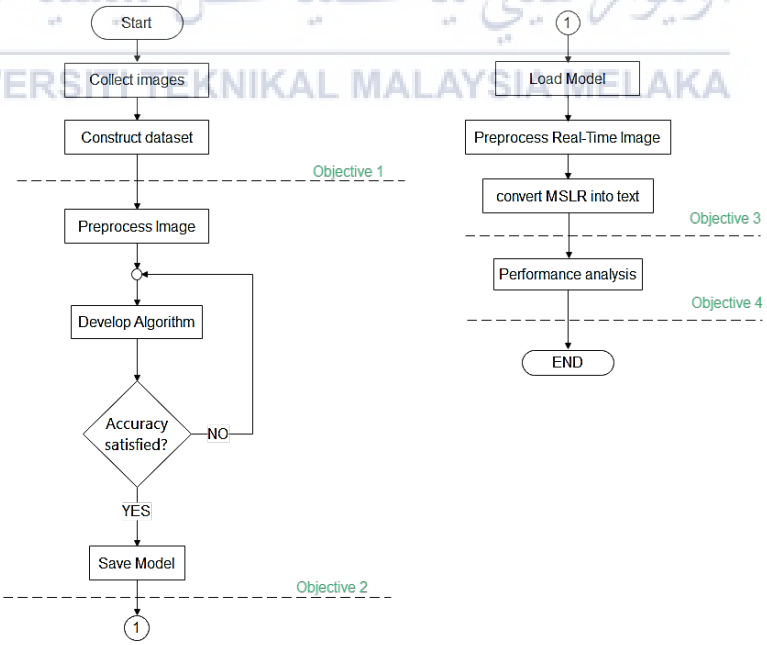


Figure 3.1: Research Methodology of the project.

The research methodology of the project is illustrated in Figure 3.1. The development of this project started by performing research in the field of sign language itself. The general challenges and needs in understanding typical sign language are analyzed. This gave a better understanding of the type of gesture that involves the signing space and others. Next, multiple SLR systems including an SLR system based on Malaysian Sign Language is analyzed. The techniques used in the research are compared among them. Besides, the prediction accuracy and the number of samples collected are considered. Based on the data, the technique used in this project is determined. The deep learning approach based on the CNN technique is applied along with some image processing methods.

After the research on the topic of SLR is done, the process continues with collecting the images that are being used in this project. The collected dataset undergoes certain image processing techniques. After that, the image is labeled and recorded according to their classes. Hence, the first objective of this project to collect and construct the Malaysian Sign Language dataset is achieved.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Next, the constructed dataset is used in the development of the SLR algorithm. The dataset is split into training, and testing datasets. The hyperparameter of the CNN model is determined and the training is done for certain epochs. After the model is trained and evaluated, the prediction value is considered. When the prediction value is satisfied, the algorithm is saved for future use. The accomplishment of the development of the SLR algorithm signified the success of the second objective of this project.

After the neural network model is developed and saved, the GUI for performing live testing is developed. The GUI functions to receive a live video feed of a signer

and convert the gesture of the signer to text form. Specifically, the text displayed is based on alphabets and Malaysian Sign Language words. After the completion of the GUI, and the live test, the third objective of this project was completed.

The final process in this project is to determine the performance of the developed algorithm. The saved model is tested with a dataset and a Confusion Matrix is computed. Based on the Confusion Matrix, other important features of the model can be determined such as the prediction accuracy, recall, precision, and F1-score. The completion of the performance analysis marks the completion of the fourth objective and the project itself.

3.2 Dataset Construction

To achieve the first objective of this project, the dataset for Malaysian Sign Language has been developed. The challenge in the deep learning technique is to acquire high numbers of the dataset. This is because it is typical for the deep learning model to excel with high numbers of datasets compared to machine learning. However, due to the limitation of time for this project to be developed, a non-conventional way to capture the image of the signer has been used. Hence, the process of capturing the image and labeling it is done automatically and the workload is distributed among volunteers. This had reduce the time taken to acquire a high number of datasets.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

3.2.1 App to capture and label sign language

An application is developed to automate the process of capturing the image of the signer and labeling it accordingly. The challenge of automating this process is to control the environment during these pictures are being taken. Thus, the application is designed to allow the volunteer to choose which type of sign to be recorded. Next, the

example of how the sign is performed is shown using an existing image. After the image is shown, the webcam of their laptop will be used to display the live video feed of themselves. Based on the video feed, the MediaPipe Holistic framework is applied to determine whether the volunteer's upper body and head are visible to the camera. If they are visible, then the hand landmark provided by the MediaPipe framework will be used to determine the Region of Interest (RoI). After the RoI is determined, the desired image and label are done automatically. The simplified process of acquiring images and labeling them is illustrated in Figure 3.2

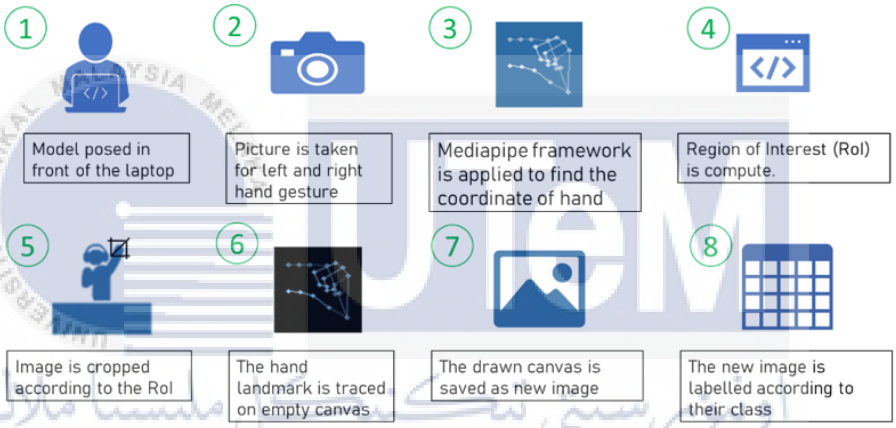


Figure 3.2: The process of acquiring images and labeling them.

On the developed GUI, a window will show a model demonstrating how to pose for the specific gesture. A timer of 5 seconds is given to allow the user to comprehend the dedicated pose. Then, the window will show the live feed of the camera and start to record the frames when it detects the human pose. The speed of image recording is set to 1 frame per second. While doing so, the user is instructed to adjust their pose at a random angle while maintaining the gesture shape. This allows for variation of position, scale, and angle of the user to be recorded. The variation of the images recorded is important to reduce the potential of overfitting the model during the training session. The process of recording the image started with the right hand of the

user and was later, repeated with the left hand. The flowchart in Figure 3.3 illustrates the process in a nutshell.

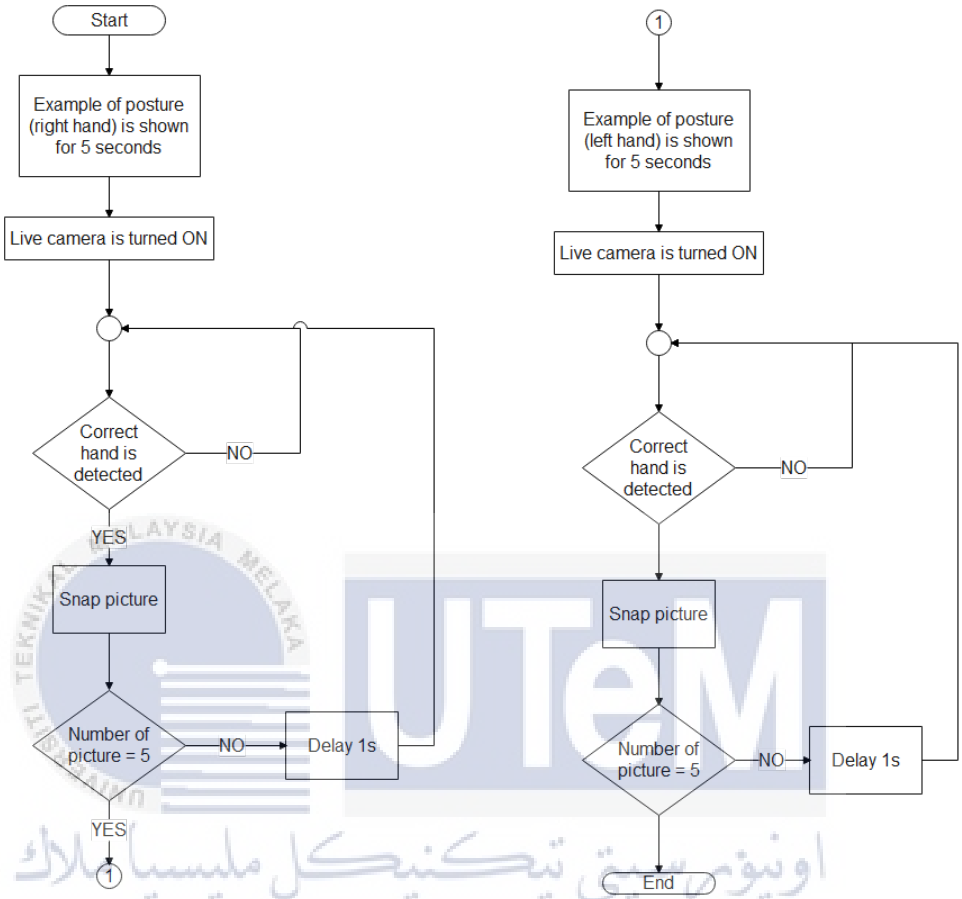


Figure 3.3: The flowchart of how images are recorded.

This basic application is important in increasing the efficiency of collecting the dataset. The application is converted into a .exe file that can be installed in any Windows OS. For this reason, volunteers can personally handle the dataset-acquiring process by installing it on their respective personal computers. The acquired dataset then is collected to be one complete dataset. The overall process of collecting the dataset is illustrated in Figure 3.4.

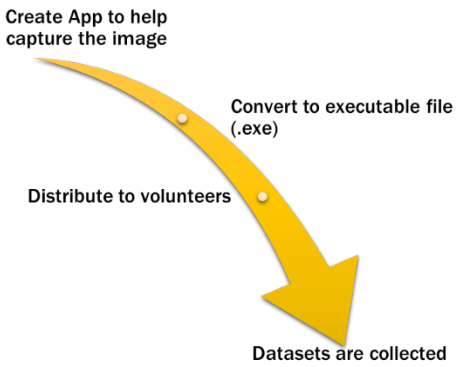


Figure 3.4: The main technique of acquiring a large number of datasets.

3.2.2 Technique to determine the RoI

The MediaPipe framework is a pre-trained machine learning that detects certain features of the human body. For example, the MediaPipe Holistic allows the landmarks of hands, body posture, and face to be identified. In this project, the MediaPipe Holistic is used in two different ways. The body posture landmarks are used to determine whether the volunteer is standing at an appropriate position and distance. By detecting the upper body and head of the volunteer, the environment in which the image is captured is controlled. Since the signing space of a signer is between the upper waist height and approximately above the head, the application can determine whether the volunteer position according to the video feed is appropriate.

The other way that the MediaPipe Holistic framework is used, is to determine the RoI of an image. The hand landmark provided by the framework consists of 21 different normalized coordinates. The details of the hand landmarks based on MediaPipe Holistic are illustrated in Figure 3.5. The coordinates from the hand wrist to the pinky fingertip are used to determine the size and position of the RoI of an image.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

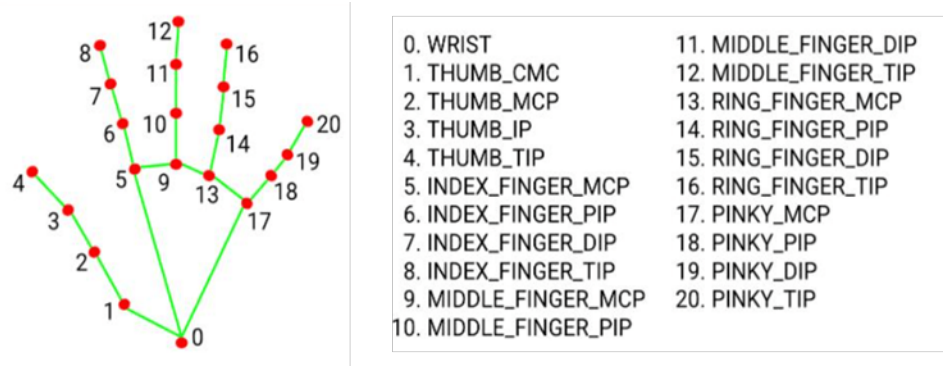


Figure 3.5: The details of hand landmarks based on the MediaPipe Holistic framework.

The process of determining the RoI starts with assigning the width and height of an image as the minimum x-coordinate and y-coordinate respectively. Besides, the maximum x-coordinate and y-coordinate are set to 0. Then the first normalized x and y coordinate of the hand landmark is multiplied by the image width and height to gain the actual hand landmark coordinate. Then the values are compared with existing minimum and maximum x and y coordinates. If the hand landmark coordinate is higher than the existing maximum coordinate, the new maximum coordinate is set to the value of the hand landmark coordinate. Similarly, if the hand landmark coordinate is lower than the existing minimum coordinate, the new minimum coordinate is set to the value of the hand landmark coordinate. The process is repeated for all 21 hand landmarks. Based on that the minimum and maximum of the RoI can be determined according to the acquired minimum and maximum of the x and y coordinate plus a certain offset. The overall process of determining the RoI of an image is illustrated in Figure 3.6. The flowchart is a simplified explanation without considering the value of the offset. The example of RoI drawn on an image is shown in Figure 3.7. The implementation of computing the RoI on an image using Python coding is shown in Appendix A.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

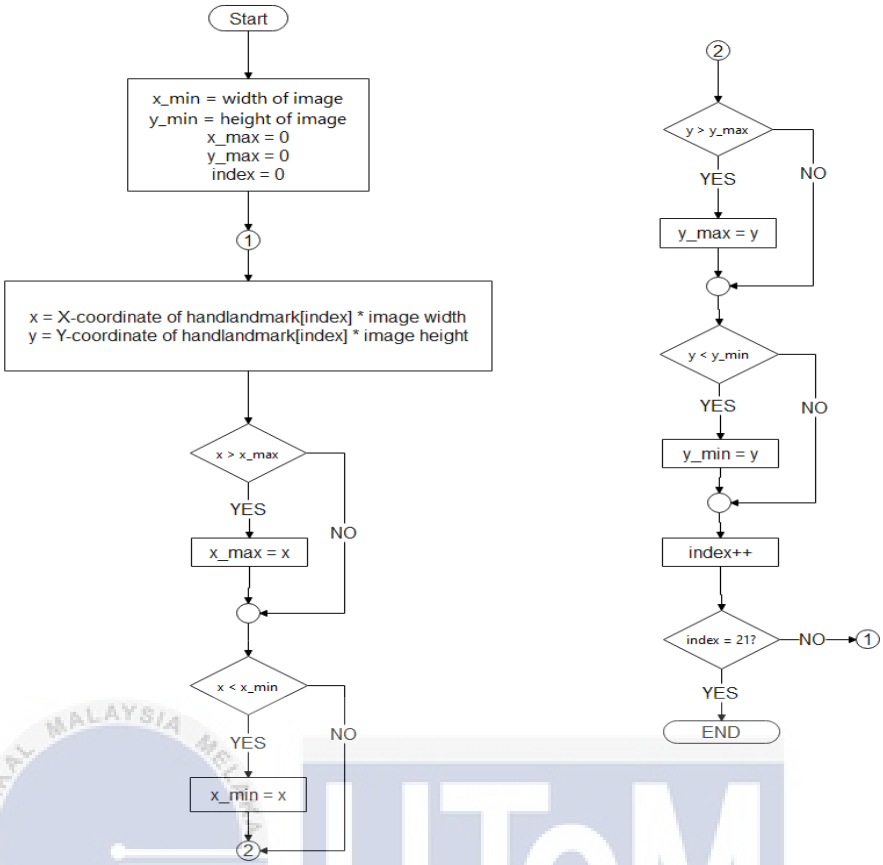


Figure 3.6: The flowchart of determining the RoI without any offset.

Figure 3.7: Example of RoI drawn on an image based on the hand landmark.

3.2.3 Using the RoI and MediaPipe Holistic to determine the dataset image.

After the RoI of an image is acquired, a new image needs to be developed based on the information in the RoI. Based on the research done, the challenge in using a vision-based sign language recognition system is the nature of RoI. For example, the inconsistency of image brightness can affect the effectiveness of the algorithm. Moreover, the skin color of the signer can also influence the result of the prediction.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Hence, an approach is determined, which is to eliminate both possibilities in this project. The acquired hand and face landmarks itself is used as the dataset for this project instead of a cropped image. The face landmark is only included when the RoI overlaps a face. Otherwise, only the hand landmarks are included in the new image. The details of the face landmarks are shown in Figure 3.8.

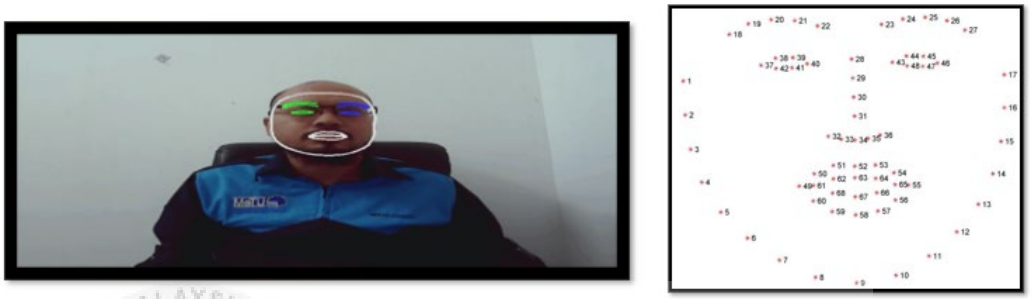


Figure 3.8: The details of face landmarks based on the MediaPipe Holistic framework.

The hand and face landmark is drawn back onto a black canvas of 100x100 pixels. Therefore certain processes needed to be done before the canvas can be saved as a new image. Since this project utilizes the CNN architecture which excels when the input of the model is in the image format, the hand and face landmarks joints and connections are colored with a distinct color. The distinct colors are beneficial in feature extraction since each type of finger can easily be identified compared to a single color (skin color) finger. In addition, the right and left eyes are also colored differently.

Since the hand and face landmarks coordinates are in normalized form, the coordinates of the landmarks are multiplied by the width or height of the image depending on whether it is an x or y coordinate. Then, the minimum and maximum coordinates of the RoI are obtained. Based on these data, equations 3.1, and 3.2 can be used to determine the new coordinate of the landmarks on a 100x100 black canvas.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

$$x_{new} = \frac{(x \times 640) - Rx_{min}}{Rx_{max} - Rx_{min}} \times 100\ pixels \quad (3.1)$$

$$y_{new} = \frac{(y \times 480) - Ry_{min}}{Ry_{max} - Ry_{min}} \times 100\ pixels \quad (3.2)$$

The value for every element in equations 3.1, and 3.2 are extracted as illustrated in Figure 3.9. 6 elements can be derived from the acquired RoI which are the RoI's minimum x coordinate, minimum y coordinate, maximum x coordinate, maximum y coordinate, new x coordinate, and new y coordinate. The equations multiply the x coordinate by 640 and the y coordinate by 480 because the captured video is set to the format of 640x480 pixels. An example of a converted image is shown in Figure 3.11. While the example of rescale image is shown in Figure 3.10.

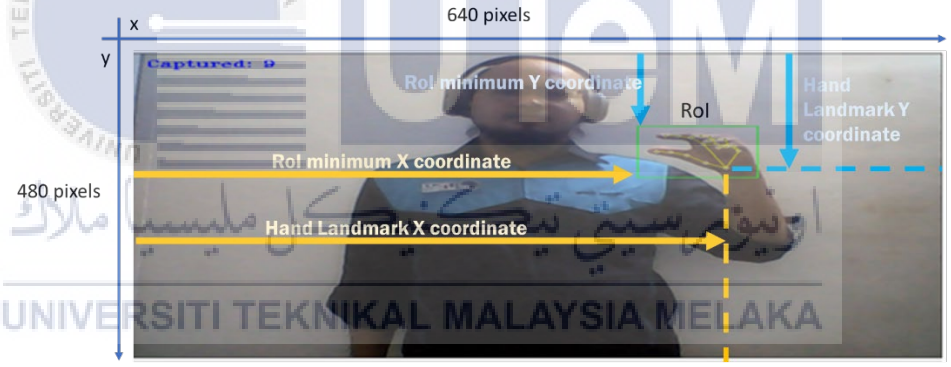


Figure 3.9: The data of RoI coordinates are extracted first before producing a new image can be done.

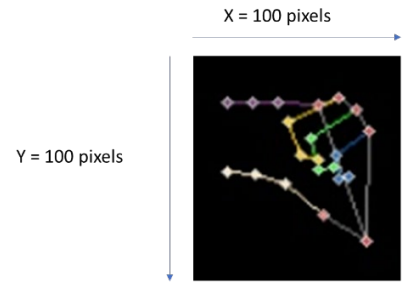


Figure 3.10: Example of new image saved from a 100x100 pixels canvas. It shows the hand landmark for the gesture 'G'.

Figure 3.11: Example of new image saved for the gesture word 'India' which includes hand and face landmarks.

3.2.4 Construction of the dataset.

After the new image is produced, the image is saved into a specific folder. During the process, the image class (according to the type of gesture) and the image file name are saved into a CSV file. This file stores all the file names and the classes of the image accordingly. The file will then use in the development of the neural network. The example of the collected dataset and its labels is shown in Figure 3.12

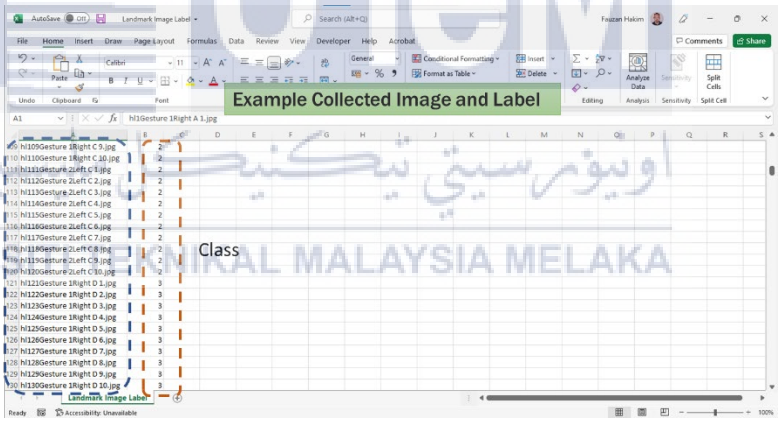


Figure 3.12: Example of the CSV file used to store the information of the dataset.

3.3 Modeling of the Neural Network

Once the development of the dataset is completed, the modeling of the artificial neural network started. The dataset is split into a 70% training set and a 30% test set. The training dataset is used as an input for the CNN model used in this project. After the training is done, the trained model is evaluated using the 30% test set. Once, the test is completed, the prediction result can be obtained. If the prediction result is not

satisfied, the training process will be repeated with different parameters. The general process of developing the neural network is illustrated in Figure 3.13. In addition, the implementation of Python programming for splitting the dataset is shown in Appendix B.

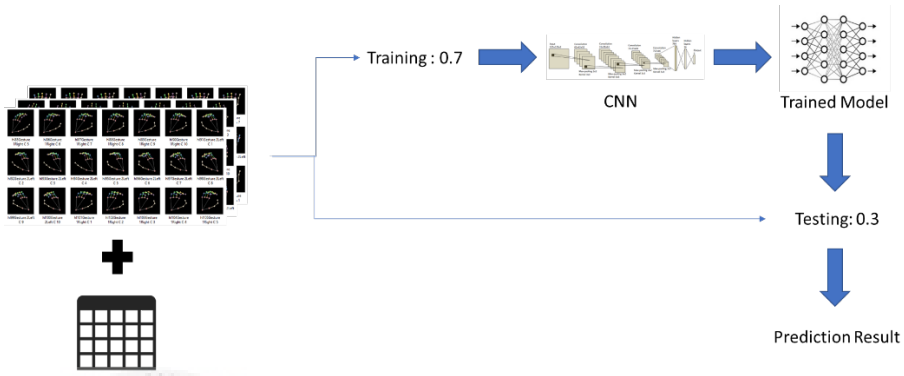


Figure 3.13: Overall process of modeling the Malaysian Sign Language Recognition algorithm.

The input channel of this artificial neural network is set to 3 since the image is in the RGB format. Besides, the number of classes is easily determined by the number of types of gestures to be trained. In this case, the number of classes is 29. Other hyperparameters such as the value of the learning rate, batch size, and the maximum epoch is as in Table 3.1.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Table 3.1: The hyperparameter used in the training.

Hyperparameters	
Input Channel	3
Number of Classes	29
Learning Rate	0.001
Batch Size	128
Maximum Epochs	15

3.3.1 Convolutional Neural Network

The architecture of the CNN starts with a convolution layer. A proper kernel size, padding, and stride are applied. The activation function used in this layer is the ReLu function. Then, the convolution with the activation function is repeated before

a Max Pooling is applied to find the greatest feature. These sets of layers are repeated 3 times. After that, the outputs are flattened and the first fully connected layer is applied. At this point, the ReLu is also used, then the layer is repeated 3 times before the output remains 29 to suit the number of classes.

For every 4th of epoch time, the model is saved. This process is called checkpoint save. This allows the model to be trained and pause when needed. After the completion of the training, the model is tested back with the training dataset and the test dataset. The result of the prediction determines whether the model is saved or not. The Python program for developing the architecture of the CNN is shown in Figure 3.14. Furthermore, the Python coding to train the neural network is shown in Appendix C.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

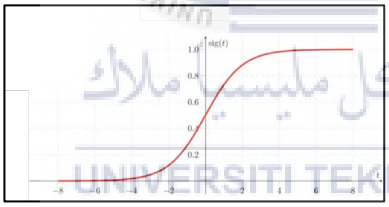
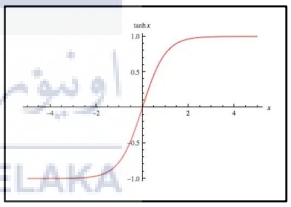
```
model = nn.Sequential(

    nn.Conv2d(3, 32, kernel_size=3, padding=1),
    nn.ReLU(),
    nn.Conv2d(32, 64, kernel_size=3, stride=1, padding=1),
    nn.ReLU(),
    nn.MaxPool2d(2, 2),

    nn.Conv2d(64, 128, kernel_size=3, stride=1, padding=1),
    nn.ReLU(),
    nn.Conv2d(128, 128, kernel_size=3, stride=1, padding=1),
    nn.ReLU(),
    nn.MaxPool2d(2, 2),

    nn.Conv2d(128, 256, kernel_size=3, stride=1, padding=1),
    nn.ReLU(),
    nn.Conv2d(256, 256, kernel_size=3, stride=1, padding=1),
    nn.ReLU(),
    nn.MaxPool2d(2, 2),

    nn.Flatten(),
    nn.Linear(36864, 1024),
    nn.ReLU(),
    nn.Linear(1024, 512),
    nn.ReLU(),
    nn.Linear(512, 29)
)
```

Figure 3.14: The CNN architecture using PyTorch.

3.3.2 Activation Function

The activation function is used in the Artificial Neural Network to help limit the result of each neuron. The estimated output of a single neuron can increase or decrease exponentially as the network becomes more complicated. Hence, the activation function can control or restrict the output of each neuron within the class limits. Besides, the implementation of the activation function helps add non-linearity characteristics to the network. Since the data used in this project is complicated, it is wiser to conclude that the classification pattern should not be linear. Among the popular activation functions is the Sigmoid function, Rectified Linear Unit (ReLu) function, Tanh function, and others.

Table 3.2: Type of Activation Functions and its general formula [16]

Type of Activation Functions		
Sigmoid Function	ReLu Function	Tanh Function
		
$sig(t) = \frac{1}{1+e^{-t}}$	$ReLu(x) = \begin{matrix} 0\ for\ x < 0 \\ x\ for\ x \geq 0 \end{matrix}$	$\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$

Based on Table 3.2, the Sigmoid activation function is not zero-centered, which will cause a Vanishing Gradient problem. The Vanishing Gradient problem is caused when a large input change causes a small output change. This is due to the Sigmoid activation function output being only between 0 and 1. It causes some nodes to completely die and not learn anything. Hence, this method is more appropriate to use in a binary classification system. Since the number of classes in this project is 29, the activation function is not suited. Meanwhile, the Tanh activation function solves

the problem of zero-centered. It also has a smooth gradient converging function. However, it still does not solve the Vanishing Gradient problem. Finally, the ReLu activation function describes an output from 0 to equal its input. It allows the output of the activation function to not saturate and solve the Vanishing Gradient problem. It is also commonly used in the Convolutional Neural Network. Hence, this project used the ReLu activation function.

3.3.3 Pooling and Downsampling

To summarize the various inputs from the previous layer in Artificial Neural Network (ANN), pooling is needed. Although pooling can be seen as performing a convolution layer, the obvious distinction is that the pooling operation is fixed and not learned. It also helps the network to do downsampling. The Max-pooling is used in the ANN to compute the significant maximum values based on its previous inputs. For example, out of n numbers of inputs, the highest value is chosen. This acts as the kernel can identify the distinct features out of a vast number of inputs.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

3.4 Simulation and Real-Time Testing

After the algorithm is developed and saved, a GUI is developed to test the algorithm in real time. The GUI is used to receive and display the video feed of the signer. During that process, the MediaPipe Holistic framework is applied to the video in background processing. Each frame is extracted, and the RoI of the image is determined using the framework. At the same time, the saved model is loaded back. Next, the process of obtaining the hand and face landmarks and retracing them onto the new 100x100 pixels canvas is done. The new image is then fed to the input of the saved Malaysian Sign Language Recognition algorithm. Afterward, a prediction is made by the algorithm based on the Malaysian Sign Language gesture. The predicted gesture is

then displayed on the GUI. The process is repeated as long as the camera is turned on. The overall process of the real-time test of the algorithm is illustrated in Figure 3.15. In addition, the implementation of Python programming for real-time simulation is shown in Appendix D.

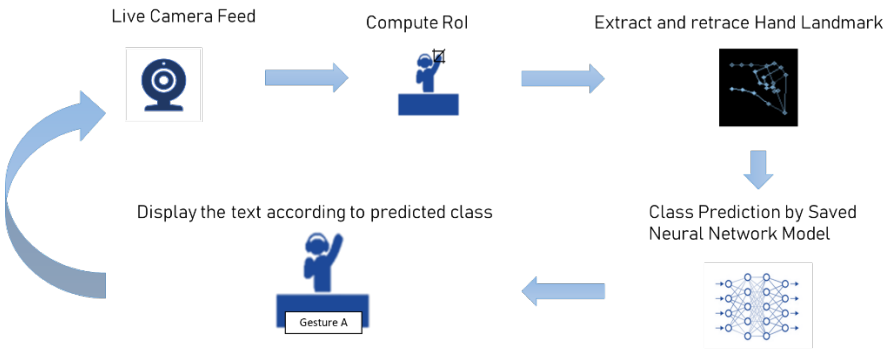


Figure 3.15: Overall process of Real-Time Test using the saved Malaysian Sign Language Recognition algorithm.

3.5 Performance Evaluation

The final stage of this project was to evaluate the performance of the developed algorithm. The technique to evaluate the performance is by using the Confusion Matrix. The Confusion Matrix allows the value of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) to be determined. There is a difference between constructing the Confusion Matrix for Binary-Class ANN and the Multi-Classes ANN. Since this project predicts 29 classes, the Multi-Classes Confusion Matrix is developed. Then, to perform analysis on each class, the Binary-Class Confusion Matrix is derived from it. Figure 3.16 shows the difference between the Multi-Class Confusion Matrix and the Binary-Class Confusion Matrix.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 3.16: Comparison between Binary-Class Confusion Matrix (left) and Multi-Class Confusion Matrix (right).

From the acquired TP, TN, FP, and FN, a further evaluation can be done. The evaluation of accuracy, precision, recall, and F1-Score can be done using collected TP, TN, FP, and FN. The accuracy is the probability that the model prediction is correct. While the precision tells us how much we can trust the model when it predicts an individual as Positive. In addition, recall measures the ability of the model to find all the Positive units in the dataset. Finally, the F1-Score helps to measure unbalanced Recall and Precision at the same time. All the equations used to determine the accuracy, precision, recall, and F1-Score are as follows:

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.3)$$

$$Precision = \frac{TP}{TP + FP} \quad (3.4)$$

$$Recall = \frac{TP}{TP + FN} \quad (3.5)$$

$$F1 - Score = 2 \cdot \left(\frac{Precision \cdot Recall}{Precision + Recall}\right) \quad (3.6)$$

3.6 Summary

In summary, this chapter explained the details method of developing the project. The process of developing this project can be categorized into four parts. The process starts by collecting images to be used as the project dataset. The method to collect the images is by building an application and distributing it to the volunteers so that they can record the images by themselves. This method is done to reduce the workload and the time taken to develop the dataset.

Next, the neural network for this project is developed. The input of this neural network is an image sized $100x100$ pixels. The rescale images are the result of preprocessing technique that involves retracing the skeletal joints of the hands and face landmarks onto an empty canvas. The architecture of the neural network is developed following the architecture of a standard Convolutional Neural Network architecture. The model is trained for several epochs until the desired accuracy is achieved. Then, the model is saved

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Subsequently, the algorithm is intended to be implemented on a live test. So, a GUI is developed to allow the user to test the algorithm in real-time. The computer's camera records the user's pose and automatically determines the RoI. Then, the images are processed to get the hand and face landmarks and feed them to the input of the neural network. Lastly, the model gives its predictions.

Finally, the method to analyze the performance of the model is explained. The analysis is intended to measure its effectiveness and not to find the most optimized model. Hence, a Confusion Matrix is developed. From the Confusion Matrix, various characteristics of the model can be identified such as Accuracy, Precision, Recall, and F1-Score.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Malaysia Sign Language Dataset Construction

To collect the dataset to be used in this project, a GUI is developed. The GUI allows users to record individual images according to different classes (24 alphabets and 5 words). To start recording, first, ones need to select a camera channel. The camera channel can be selected due to certain computers having multiple cameras or external cameras. Then, the camera needs to be turned On. By selecting the appropriate gesture, for example, Gesture A, the process can be started by clicking the snap picture. The layout of the developed GUI is shown in Figure 4.1 and the details of every element in the GUI are explained in Table 4.1. An example of the process to record the image is shown in Figure 4.2.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

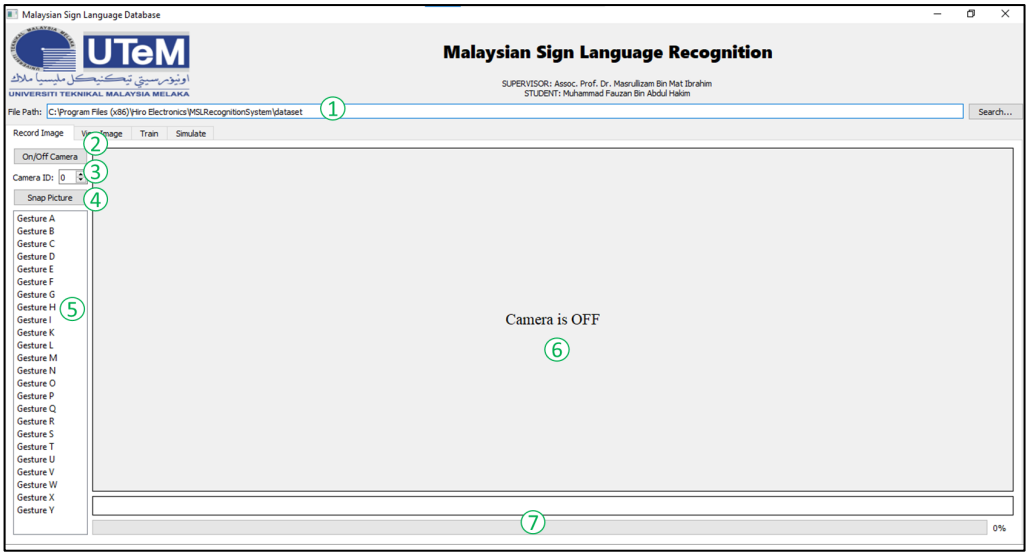


Figure 4.1: The layout of the GUI for recording images.

Table 4.1: Description of different functions in the GUI (Record Image Tab) for recording images.

No.	Label Name	Description
1	File Path	The file path of the application in the computer.
2	On/Off Camera Button	The button allows the computer to turn On or Off the camera.
3	Camera ID	It determines which camera is used when having multiple cameras attached to the computer.
4	Snap Picture Button	The button to start the recording process.
5	Gesture Class	Users need to select which gesture to record.
6	Camera Window	GUI window that allows for live camera feed and model to demonstrate specific gestures.
7	Progress Bar	It keeps tracking the percentage of the recorded image until completion.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

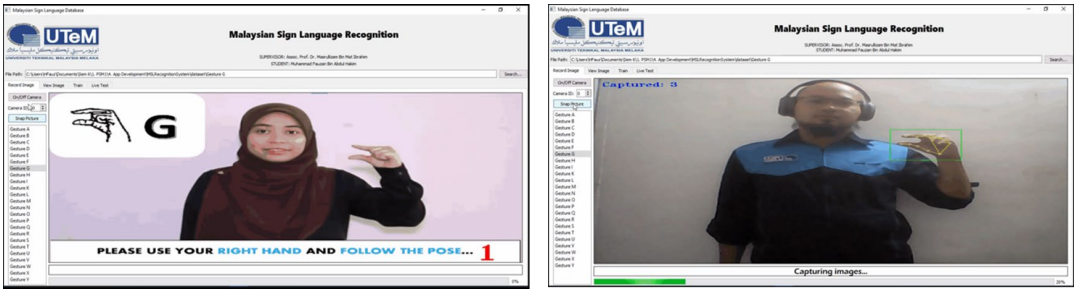


Figure 4.2: The left image shows a model demonstrating how to pose for a specific gesture. The right image shows the live feed when the image is recorded.

4.1.1 RoI Image Recorded and Post Processed

During the recording of the images, various background processes happen. First, the images are organized and labeled according to their gesture classification. Then, the computer automatically determines its Region of Interest (RoI). The determination of the RoI is mentioned in Chapter 3.2.2. The image is cropped according to the RoI dimension. Then, the image is retraced back onto an empty canvas as explained in Chapter 3.2.3. The empty canvas only has the drawing of the skeleton of the hand or face landmark if it is in the RoI. This is called the hand landmark view. The result of the collected landmark images is shown in Figure 4.3.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

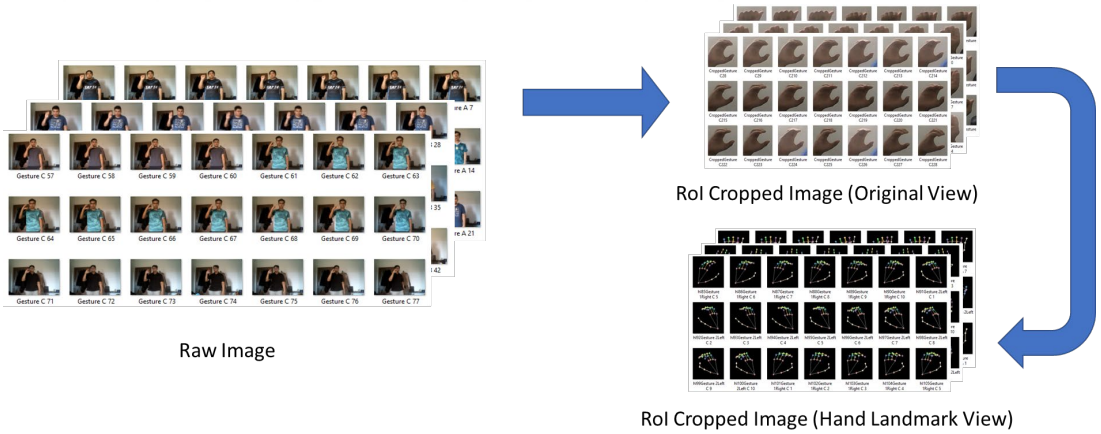


Figure 4.3: The process of getting the landmark view from the raw image.

4.1.2 Dataset Organization

The dataset is organized such that every picture is labeled with the hand side, the type of gesture, and its number. Then, it is grouped into folders named according to their gesture type. These images can be viewed on the GUI by clicking on the “View Image” tab. On that interface, the user can manually select any image that has been recorded and view its image. Besides, the user can also view the image's respective hand or face landmarks to determine whether the image is appropriate. Some images may become corrupt due to the Mediapipe framework problem. In addition, the user can analyze the image details such as the file name, the image’s width and height, or the coordinate of the image in pixels. The layout of the GUI is shown in Figure 4.4 and the details of each element of the GUI are explained in Table 4.2. In addition, the organizational structure of the dataset in the PC is shown in Appendix F.

Figure 4.4: The GUI for viewing recorded images.

Table 4.2: Description of different functions in the GUI (View Image Tab) for recording images.

No.	Label Name	Description
1	File Path	The file name.
2	Original Image	The original cropped image.
3	Post-processed Image	The image after processed.
4	Summary	The details of the image.
5	Previous/Next Button	A quick way to view the previous/next image.

4.1.3 Malaysia Sign Language Dataset Summary

In summary, the total number of gestures type recorded for this project is 29. This is aligned with the number of the alphabet except for the letter 'J' and 'Z'. In addition, the gesture 'Bapa', 'Emak', 'Melayu; 'Cina', and 'India' are collected. 10 persons have volunteered to be recorded for this project. Since the number of images recorded per person is 10 images (5 images for each side of the hands), the total images taken are 2900 images. For each image, the image name and its label (gesture type) are recorded in a .csv file. The summary of the dataset is shown in Table 4.3. Hence, the first objective of this project is achieved.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Table 4.3: The summary of the Malaysian Sign Language Dataset.

Malaysian Sign Language Dataset Summary	
Number of Gestures Type	29
Number of Volunteers	10
Image number for each side of hands per gesture	50
Total number of images for each gesture	100
Total number of images	2900

4.2 Development of Malaysia Sign Language Recognition Algorithm

The input to the neural network model is a cropped and rescale image of $100x100$ pixels. Originally, the RoI dimensions are not fixed due to the position of the user from the camera. However, before feeding the post-processed image into the network,

the image is rescaled as mentioned before so that every image has a consistent dimension. All images have 3 channels that come from the RGB-valued matrices. Moreover, the batch size for the input is 128 which represents the number of samples processed before the model is updated. In short, the input layer's shape for the neural network is $100x100x3$. Then, the network passed through a combination of convolutional layers, activation functions, and pooling. The combination is repeated 3 times. The output of the convolution layer started with 32 channels, and continues to be 64, 128, and finally 256 channels. The activation function used for each combination is the ReLu function. At the same time, pooling using the MaxPool helps to reduce the parameter and computations of the model. The first fully connected layer has an output size of $256x12x12$. The original image of $100x100$ pixels is reduced with each pooling layer to $12x12$ channels (the pooling uses a window stride of 2 and padding size of 2). Finally, the output layer produces 29 outputs aligned with the number of classes that need to be predicted. In summary, the total number of trainable parameters is 39,412,952 parameters. Besides, the total memory size of the neural network model is 4,224.40MB. The detailed shapes of the neural network are listed in Table 4.4. Besides, the block diagram of the developed neural network is shown in Figure 4.5.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Table 4.4: The output shape and the number of parameters of each layer in the trained model.

Layer (Type)	Output Shape	No. of Trainable Parameters
Conv2d-1	128, 32, 100, 100	896
ReLU-2	128, 32, 100, 100	0
Conv2d-3	128, 64, 100, 100	18,496
ReLU-4	128, 64, 100, 100	0
MaxPool2d-5	128, 64, 50, 50	0
Conv2d-6	128, 128, 50, 50	73,856
ReLU-7	128, 128, 50, 50	0
Conv2d-8	128, 128, 50, 50	147,584

ReLU-9	128, 128, 50, 50	0
MaxPool2d-10	128, 128, 25, 25	0
Conv2d-11	128, 256, 25, 25	295,168
ReLU-12	128, 256, 25, 25	0
Conv2d-13	128, 256, 25, 25	590,080
ReLU-14	128, 256, 25, 25	0
MaxPool2d-15	128, 256, 12, 12	0
Flatten-16	128, 36864	0
Linear-17	128, 1024	37,749,760
ReLU-18	128, 1024	0
Linear-19	128, 512	524,800
ReLU-20	128, 512	0
Linear-21	128, 29	12,312

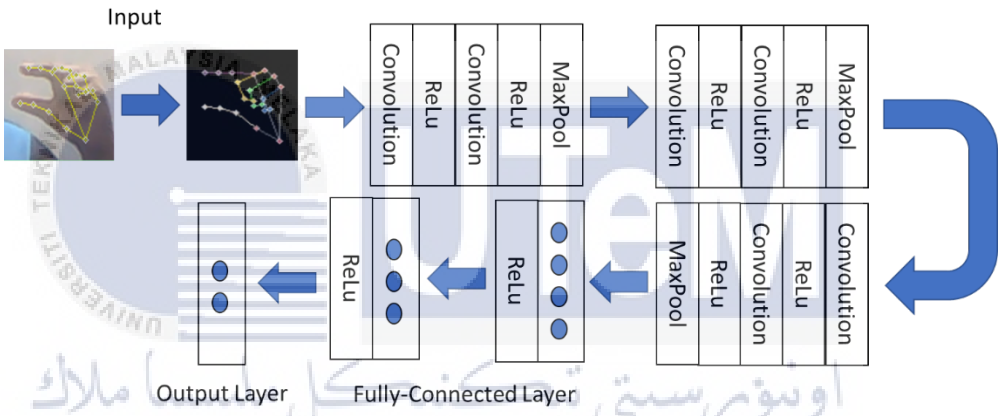


Figure 4.5: Block Diagram of the developed CNN architecture.

4.2.1 Prediction Accuracy

The model is trained with 70% of the total number of images recorded. At the same time, the model is tested with the remaining 30% of the recorded image. The result of the training and testing is shown in Table 4.5. The result shows that the model took 12 epochs before it reach its saturated accuracy result. Further training of the neural networks shows that the accuracy result of the testing set remains the same. However, the model shows a rapid learning curve with the accuracy of the test set increasing rapidly from 14% to 80% in less than 4 epochs as in Figure 4.6.

Table 4.5: The prediction accuracy for the Training set and the test set for every epoch.

Prediction Accuracy Vs No. of Epochs			
Epoch numbers	Cost/Loss	On Training Set (%)	On Testing set (%)
1	3.1871	4.9107	2.0906
2	3.1795	13.2440	6.9686
3	3.0909	27.2321	14.9826
4	2.0298	50.1488	42.5087
5	1.1564	78.4226	65.5052
6	0.5085	86.7560	80.8362
7	0.3889	87.9464	85.3659
8	0.2280	94.9405	89.8955
9	0.1516	96.2798	93.7282
10	0.1063	95.3869	93.7282
11	0.0780	97.3214	95.8188
12	0.0697	98.3631	96.1672
13	0.0429	98.6607	96.1672
14	0.0307	98.8095	96.1672
15	0.0357	98.9583	96.1672

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

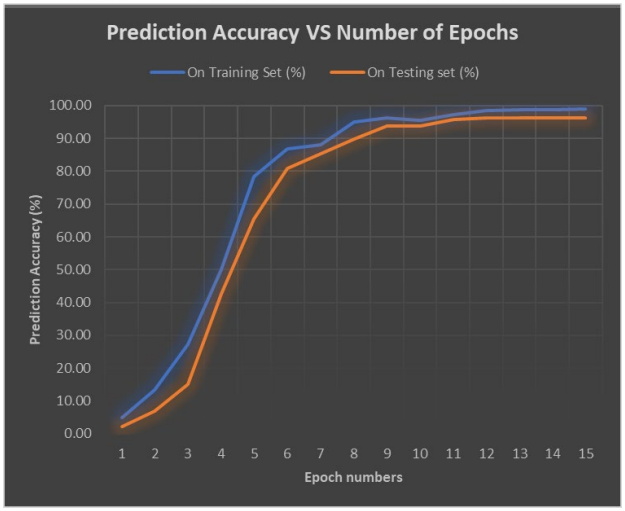


Figure 4.6: The graph of Prediction accuracy vs Number of epochs shows the rapid increase in accuracy from the 3rd epoch to the 6th epoch.

Based on the trained model, a confusion matrix is produced. The confusion matrix is the result of the prediction by the model for the test set. The test set consists of 870 images which is equal to 30 images per class. However, the Binary-Class Confusion Matrix is derived from the table and shown in Appendix E. Based on the result of the Confusion Matrix in Table 4.6, it is discovered that the model has difficulties differentiating between gesture R with U and U with V. The model has predicted the gesture U 3 times when the expected gesture should be R or V. Furthermore, the highest confusion happens between the gesture M with N and S. The model predicted wrongly 8 times for gesture M when the expected gesture should be N. Nevertheless, upon completion of the development of the neural network algorithm, it indicates the completion of the second objective for this project.

Table 4.6: The Multi-Class Confusion Matrix using the test dataset.

Predicted \ Expected	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Bapa	Emak	Melayu	Cina	India
A	30	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
B	0	29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
C	0	0	30	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
D	0	0	0	30	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
E	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	2	27	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	29	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	22	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	1	29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	27	0	2	0	0	0	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	28	0	0	0	0	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	29	0	0	0	0	0	0	0	0	0	0
U	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	26	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	3	26	0	0	0	0	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	30	0	0	0	0	0	0	0
X	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	27	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0
Bapa	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0
Emak	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	30	0	0	0
Melayu	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	30	0	0
Cina	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	30	0
India	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	30

With the help of the Confusion Matrix, the challenging gestures can be identified. Based on Figure 4.7, it can be seen that the clear distinction between gestures U and R is the distance between the index finger and the middle finger. However, the gesture R supposedly makes the index finger crosses the middle finger. But, if the image is captured at a different angle, gesture U is seen as the index finger and the middle finger is crossed. The confusion between gestures U or R with V is also caused by the same

issue. This shows that the representation of the gesture without image depth can cause conflict in determining the gesture's class.

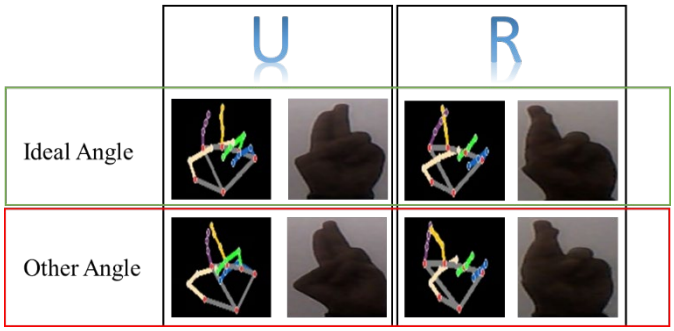


Figure 4.7: The figure shows that there is an angle that makes gesture U look similar to gesture R.

Furthermore, from the Confusion Matrix, the gestures M, N, and S have problems being differentiated. Based on Figure 4.8, the images show that the gestures M and N position its thumb under other fingers. The apparent difference is the position the thumb is placed above the little (pinky) finger for gesture M and the thumb is placed above the ring finger for gesture N. Hence, the position of the tip of the thumb plays a role to determine whether it is a gesture M or N. However, the gesture S positioned its thumb above all other fingers. Nevertheless, since the hand landmark image does not give information about the depth of each finger, it can be seen as all of the three gestures difference is where the tip of the thumb is positioned. This is not only can confuse due to the small differences in the position of the thumb between each class, but it also becomes worst when the angle of images captured is varied.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

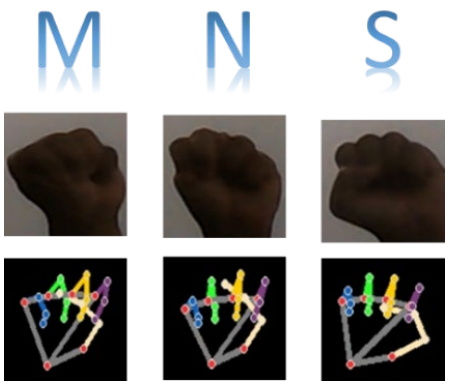


Figure 4.8: The figure shows that without the finger depth information, the only difference between the three classes is the position of the tip of the thumb.

4.3 Conversion of Malaysia Sign Language to Text

After the development of the neural network model, a GUI is developed to allow for a live test with the model. Figure 4.9 shows the developed GUI and Table 4.7 explain the details of each element in the GUI. To start the live test, the user must select the model name to be tested. Then, the user can simply click the "On/Off Camera" to allow a live recording to start. At the "Live Feed Window", the user can pose for any gesture and wait for the prediction. The result of the prediction is displayed on a white rectangle overlayed on the live image. At the right bottom of the GUI, the user can see the landmark view of the RoI in real-time. Furthermore, the model's predicted gesture is listed as well. This GUI makes translating Malaysian Sign Language alphabets (except for J and Z) and five trained words in real-time possible. Thus, the third objective of this project is achieved.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 4.9: The GUI for testing the trained model in real-time.

Table 4.7: Description of different functions in the GUI (Live Test Tab).

No.	Label Name	Description
1	On/Off Camera Button	To turn on or off the camera.
2	Model Name	To select the appropriately trained model.
3	Post-processed RoI Image	This window shows the RoI image in the hand landmark view. The view is in real-time.
4	Model's Prediction	The predicted gesture and the confidence level from the model.
5	Live Feed Window	The window shows the live feed from the camera.
6	Region of Interest	The green bounding box is where the image is cropped.
7	Translated Gesture	This white box shows the translated gesture and allows the user to construct a word using a combination of alphabet gestures.

A test is done to determine the effectiveness of the real-time Malaysian Sign Language conversion using the application developed. All types of trained Malaysian Sign Language gestures which are the gesture alphabets (except for the letters 'J' and 'Z') and the gesture 'Bapa', 'Emak', 'Melayu', 'Cina', and 'India' are shown on the camera. The algorithm then predicts the gesture and displays it on the window. The video records at 30 frames per second and each frame is given to the Malaysian Sign Language Recognition algorithm to make a prediction. However, during that time, if

the algorithm makes the same prediction up to 5 times consecutively, then the predicted gesture is displayed. Otherwise, if the algorithm predicts different gestures for every frame, the predicted class is not displayed. Therefore, some gestures are predicted slower than others. However, the average prediction of the algorithm is recorded at 1.73 seconds. In this real-time test, the same gesture is shown repeatedly 10 times and the correct prediction is recorded. The result of the prediction accuracy for a real-time test is shown in Figure 4.10.

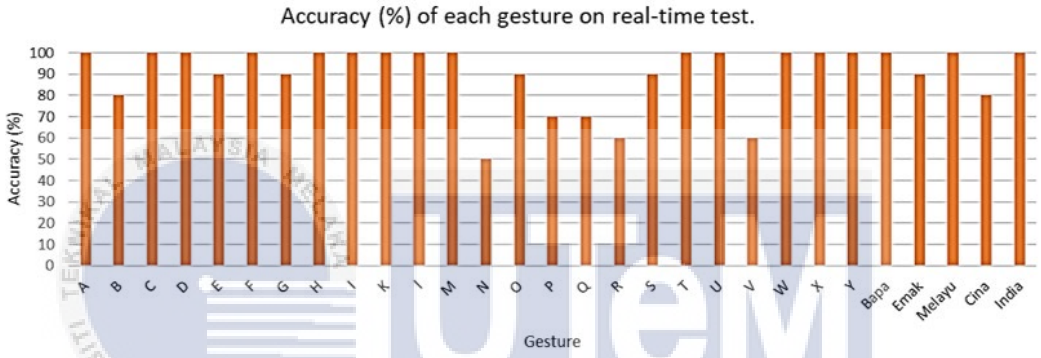


Figure 4.10: The accuracy (%) of the developed model when predicting each gesture in a real-time test.

Based on the result of the real-time test as shown in Figure 4.10, it can be seen that some of the gestures are predicted poorly. For example, the gesture 'B', 'N', 'P', 'Q', 'R', and 'V' scored less than 80%. Although the result of the real-time test is expected based on the result of the Confusion Matrix, there are other gestures prediction that perform poorly that are not detected from the observation on the Confusion Matrix alone. For example, gestures 'E', 'O', 'Emak', and 'Cina' had perfect scores during the test on the testing dataset. However, a similar result cannot be achieved during the real-time test. This shows the importance of implementing the algorithm on a real-time application before a generalization on the performance of an algorithm is made.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

4.4 Algorithm Performance Analysis

The performance of the developed neural network model can be evaluated based on its Confusion Matrix as in Table 4.6. However, a better way to evaluate it is by converting it into a Binary-Class Confusion Matrix. This is because individual classes' performance can be evaluated accurately. Hence, the Binary-Class Confusion Matrix is derived. The technique to derive the matrix for each class is by determining their respective TP, FN, FP, and TN. The TP value of any class is the value of the predicted class cell and its expected class cell is crossed. To obtain its FP value, all elements that are vertical with the TP cell are added. Meanwhile, the FN value is the result of the summing of all the elements that are horizontal with the TP cell. Finally, the TN value is the summation of all other elements in the matrix. Figure 4.11 shows an example of determining the value of TP, FP, FN, and TN for a Multi-Class Confusion Matrix. From the matrix, the TP, FN, FP, and TN for each class are listed in Table 4.8.

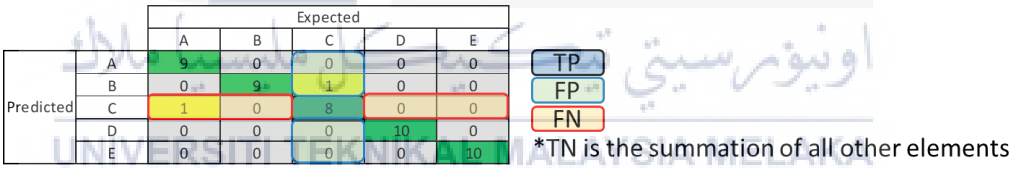


Figure 4.11: The figure shows an example of determining the value of TP, FP, FN, and TN from a Multi-Class Confusion Matrix.

Table 4.8: Result of TP, FN, FP, and TN for every class that is derived from the Binary-Class Confusion Matrix based on Appendix E.

Class	TP	FP	FN	TN
A	30	0	0	840
B	29	1	0	840
C	30	0	0	840
D	30	0	1	839
E	30	0	0	840
F	30	0	0	840
G	30	0	2	838
H	27	3	0	840
I	29	1	1	839

K	29	1	0	840
L	30	0	0	840
M	22	8	1	839
N	29	1	10	830
O	30	0	0	840
P	30	0	1	839
Q	27	3	0	840
R	30	0	5	835
S	28	2	5	835
T	29	1	0	840
U	26	4	3	837
V	26	4	0	840
W	30	2	2	836
X	27	1	0	842
Y	30	0	1	839
Bapa	30	0	0	840
Emak	30	0	0	840
Melayu	30	0	0	840
Cina	30	0	0	840
India	30	0	0	840

Using those values, further analysis can be done. The value of Accuracy, Precision, Recall, and F1-Score is calculated as explained in Equation (3.5.1) until Equation (3.5.4). The result of the analysis is listed in Table 4.9. Based on the result, it can be seen that all classes achieve at least 98% accuracy. However, the value of accuracy does not give a comprehensive understanding of the performance of the model for each prediction by class. By evaluating the precision and recall value for each class, it can be seen that the model struggles with the gestures M, N, R, S, U, and V as mentioned previously. However, prediction on M has the lowest precision with a value of around 73%. This shows that the model has predicted the gesture M incorrectly for around a quarter of the test sample. Meanwhile, the recall value for the gesture M is high which is 95.65%. Furthermore, the result of precision and recall for gesture N is the inverse of the result of gesture M. The model precision and recall value for class N is 96.67% and 74.36% respectively. It can be concluded that when

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

the model is conflicting to decide between the gesture M or N, it has a higher tendency to choose gesture M. Similarly, based on the value of precision and recall of gestures R, U, and V, the model is likely to predict it as gesture U or V.

Since the value of accuracy does not reflect the total performance of the model, the F1-Score value can be used. The F1-Score is the harmonic mean of the precision and recall. Hence, the overall performance of the model by each class can be evaluated with a single value. Table 4.9 shows that there are a total of eleven (11) gestures that score 100%. In the meantime, fourteen (14) gestures score more than 90%. Finally, the classes M, N, S, and U score less than 90%.

Table 4.9: The results of Accuracy, Precision, Recall, and F1-Score for each class.

Class	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
A	100.00	100.00	100.00	100.00
B	99.89	96.67	100.00	98.31
C	100.00	100.00	100.00	100.00
D	99.89	100.00	96.77	98.36
E	100.00	100.00	100.00	100.00
F	100.00	100.00	100.00	100.00
G	99.77	100.00	93.75	96.77
H	99.66	90.00	100.00	94.74
I	99.77	96.67	96.67	96.67
K	99.89	96.67	100.00	98.31
L	100.00	100.00	100.00	100.00
M	98.97	73.33	95.65	83.02
N	98.74	96.67	74.36	84.06
O	100.00	100.00	100.00	100.00
P	99.89	100.00	96.77	98.36
Q	99.66	90.00	100.00	94.74
R	99.43	100.00	85.71	92.31
S	99.20	93.33	84.85	88.89
T	99.89	96.67	100.00	98.31
U	99.20	86.67	89.66	88.14
V	99.54	86.67	100.00	92.86
W	99.54	93.75	93.75	93.75
X	99.89	96.43	100.00	98.18

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Y	99.89	100.00	96.77	98.36
Bapa	100.00	100.00	100.00	100.00
Emak	100.00	100.00	100.00	100.00
Melayu	100.00	100.00	100.00	100.00
Cina	100.00	100.00	100.00	100.00
India	100.00	100.00	100.00	100.00

Nevertheless, this model has achieved a remarkable performance based on the accuracy of the test dataset alone. A comparison has been done with five other models as in Table 4.10. All of the models compared are developed based on CNN architecture for fairness purposes. Although this model does not score the highest accuracy compared to the others, this model exhibits some important advantages. The other 5 models are trained using advanced hardware such as a modern GPU. Besides, the size of the dataset collected by four other studies is more than this project has collected. The fact that this model can be trained with a far less dataset size and achieve an almost similar result with other models shows that the dataset development and the image processing technique for this project are better. The accuracy of the model is majorly influenced by the quality of the dataset collected and the features extracted during the pre-processing phase. In addition, the number of training epochs can show how good an architecture of the algorithm is created. This project manage to achieve 96% accuracy with less than 15 epochs compared to other models that need to be trained for at least 50 epochs before achieving the same result. It shows the developed architecture of this model is better than others.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Table 4.10: Accuracy comparison with other studies that used CNN architecture.

Ref	Sign Language	Class Size	Dataset Size	Architecture	Epoch	Accuracy, (%)
[19]	America Sign Language	24	65,000	CNN	50	99.99
[22]	Arabic Sign Language	32	50,000	CNN	100	97.6
This Project	Malaysian Sign Language	29	2,900	CNN	15	96.16
[23]	Japanese Sign Language	41	10,999	CNN	100	94
[24]	Bengali Sign Language	37	1,147	CNN	50	84.68
[15]	Malaysian Sign Language	24	53,298	CNN	100	79.54

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

4.5 Discussion

Based on the results acquired and the analysis that has been performed in this project, several issues can be highlighted. Firstly, this project has developed a GUI to support various objectives of this project. The first objective which is to construct the dataset has been made easy by the development of an application that can be distributed among the volunteers. In less than 7 weeks, 2900 images have been collected. The images also have been labeled, processed, and sorted automatically. Moreover, a GUI is also constructed that allows users to quickly review recorded

images and its landmark so they can be evaluated efficiently. Finally, a GUI is also constructed to allow the user to test the algorithm in real-time. Although the main objective of this project is to develop an algorithm that can translate Malaysian Sign Language gestures into text form, the construction of those GUI helps increase the efficiency of the project development.

Secondly, based on the result of the model's accuracy, it is shown that the model can achieve an accuracy of more than 95%. The scope of this project is to develop the algorithm based on the CNN architecture. Hence, no other model is developed to compare its performance in recognizing Malaysian Sign Language. Nevertheless, the CNN architecture shows a promising result when the model can be trained with few images (100 images per class) and achieve more than 95% accuracy. In addition, the amount of training the model required is very low which is less than 15 epochs. It shows that the CNN architecture is suitable to be used in developing a sign language recognition system when the input for the model is in image format.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Next, based on the analysis of the Confusion Matrix, it is shown that the neural network model is imperfect. The major factor behind the poor result of the F1-Score for classes M, N, S, U, V, and R is the lack of image depth. The angle of the image recorded can influence the result of the processed image and can cause certain gestures to appear the same as others. Therefore, to improve the result, the image processed should be more detailed than simply extracting the skeletal structure of the hand. However, the use of various colors to represent different joints and links of the hand structure should be maintained.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

5.1 Conclusion

The development of this project has been centered on providing a platform for people to communicate with the deaf community without having to learn Malaysian Sign Language. The main objective of this project is to develop an algorithm that can predict Malaysian Sign Language gestures and translate them into a text form. Hence, the process of this project started with constructing the dataset for Malaysian Sign Language gestures. The dataset of this project consists of static gestures which are the 24 alphabets and 5 common words only. The process of recording images for the dataset is performed in less than 2 months. The total number of images successfully recorded is 2900 images which are equal to 100 images per class. The method of image processing, labeling, and sorting is designed to be automatically done by the computer. Therefore, the first objective of this project is achieved.

Then, the algorithm of this project is developed. CNN has been decided as the architecture of the neural network. The decision is done by performing various research and comparing its advantages and disadvantages. The input to this neural network is a processed image. The skeletal structure of the hands and face landmark recorded in the dataset is retraced into a new canvas. The skeletal structure is also given a distinct color for different joints and links. This was aimed to increase the prediction performance since the variation of skin color, background images, and camera quality can be eliminated. Later, the CNN architecture is designed with 21 layers and 39,412,952 trainable parameters. The model is trained for 15 epochs and the accuracy of the model is 96.1672% on the test set. With the development of the algorithm and satisfying accuracy, the second objective of this project is achieved.

Next, a GUI is developed to test the algorithm in real-time. The GUI is a simple design that allows the user to load the saved model and enable it to make a real-time prediction. The GUI also includes the image of the hand and face landmarks in real time so that the user can verify the processed image. Besides, the predicted gesture is also displayed. The GUI was also designed to let the user construct words using a combination of alphabet gestures. A test is done to see the effectiveness of the algorithm when performing real-time prediction. The result shows that the algorithm prediction accuracy ranges from 50% to 100 % depending on gesture type. The prediction accuracy of the real-time test is approximately similar to the prediction accuracy of the test dataset. Hence, the third objective of this project is achieved.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Finally, the developed model is analyzed. The Multi-Class Confusion Matrix is acquired from the model. Later, the Binary-Class Confusion Matrix is derived from it. From the result of the confusion matrix, it can be seen that the model is not performing

well when predicting the gestures M, N, S, T, U, and V. Moreover, after performing the calculation to determine the precision, recall, and F1-Score, it can be concluded that the model is likely to predict M when gesture M, N, or S is shown. In addition, the model is likely to predict V or U when gesture V, U, or R is shown. This shows, that among the 29 gestures trained, only 23 gestures can be predicted with minimum to no error. Nonetheless, the completion of the analysis marks the completion of the final objective and the completion of the whole project.

5.2 Project Impacts and Commercialization

This project is determined to gives a better impact on the community and environment. This project is also conducted in a manner that is aligned with some of the Sustainable Development Goals (SDG), created by the United Nations General Assembly. Figure 5.1 shows some of the SDGs that are related to this project. Firstly, since this project has been developed without the need for any extra electronic hardware except for the existing PC/Laptop, the commercialization of this kind of project will not increase the amount of e-waste. SLR that uses sensor-based will requires hardware such as sensors and processors to operate. In addition, an advanced sensor-based SLR typically have a higher number of sensors and processor. This product will eventually become e-waste. Hence, the implementation of this project based on existing computer vision can help reduce e-waste which is aligned with SDG 12.5 (By 2030, substantially reduce waste generation through prevention, reduction, recycling, and reuse). The incineration of e-wastes can emit toxic fumes and gases, thereby polluting the surrounding air. Improperly monitored landfills can cause environmental hazards. Thus, this project is aligned with SDG 3.9 (substantial reduction of health impacts from hazardous substances).

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

The importance of having better access to communication between deaf people and others should not be undermined. A lot of education organizations could not include the deaf community in their organization due to the lack of sign language translators. To celebrate SDG 4.a (Build and upgrade education facilities that are child, disability, and gender sensitive and provide safe, non-violent, inclusive, and effective learning environments for all), this project aims to reduce the obstacle that the deaf community has to access a better education system. Not all universities in the world can support the inclusion of deaf people in their institutions. This project can be used by the deaf community when attending educational programs without the need for a sign-language translator.

In addition, the deaf community has to communicate with normal people when dealing with the court, official matters, and others. Government and non-profit societies have tried to fulfill the need for translators but it has never been completely successful. This has created a social gap between the deaf community, and others. The social gap then becomes discrimination towards the deaf community in terms of economic and status opportunities. This project is proven to try to achieve SDG 10.2 (By 2030, empower and promote the social, economic, and political inclusion of all, irrespective of age, sex, disability, race, ethnicity, origin, religion, or economic or another status). The government or other sectors such as banks can set up a PC to run this program to communicate with deaf people at their counter. Besides, when deaf people need to communicate with persons such as lawyers or counselors, they can eliminate the need for a translator. This conversation sometimes is considered private, thus, this project can help protect their secrecy.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

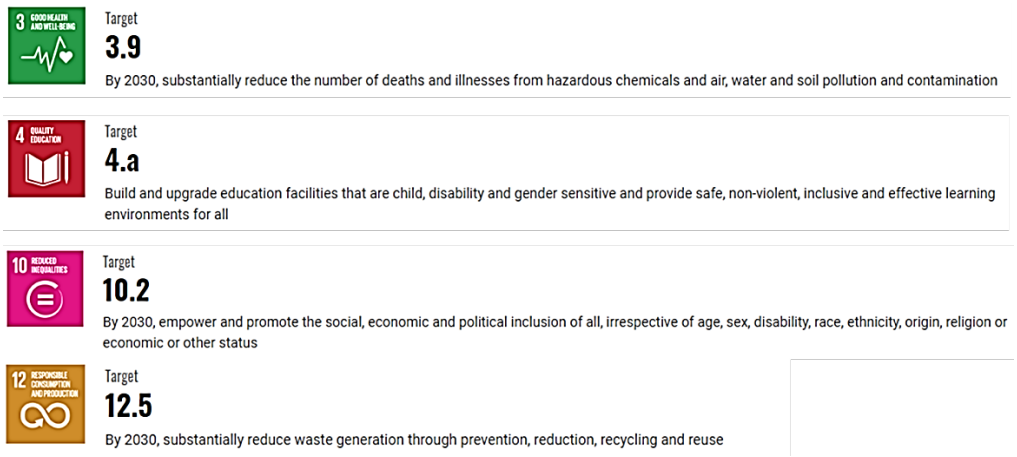


Figure 5.1: Some of the Sustainable Development Goals and their explanation.

This project has a wide potential for commercialization. The project has proven that a working real-time Malaysian Sign Language translation can be done by a computer. The need for Malaysian Sign Language translators can be reduced if the project is improved in certain aspects. It can be estimated that the role of translating Malaysian Sign Language can be shifted to the A.I. alone in a near future. The cost of implementing a vision-based SLR system is significantly low compared to the sensor-based SLR system. The only expected cost of the system is the cost of having a computer, but only if the user does not poses it. This allows the developed system to be distributed quickly and conveniently.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

The system can also be commercialized by implementing the algorithm in a microcontroller. The microcontroller allows the system to have better mobility and become a specific application embedded-based device. The device can be developed to have a specific function which is to convert Malaysian Sign Language only. Since this project has constructed and reserved the Malaysian Sign Language dataset, the algorithm can be trained to function on any device besides the computer.

5.3 Improvement and Suggestion

Although the project achieves all its objectives, the system can still be improved. The performance and robustness of this system can be improved in various ways such as:

i. The typical challenge to increase the performance of a deep learning model has always been due to the dataset size. Since deep learning trains without human interference, the model requires a huge number of datasets. Hence, it is suggested to collaborate with the deaf community or deaf society so that the dataset can have a huge number of samples and can be quickly constructed.

ii. Images have been used as input for the model for this project. However, Malaysian Sign Language cannot be fully translated if only images are used. To increase the robustness of this system, the input for this model has to be changed to video. This allows a dynamic gesture to be trained as well.

iii. The performance of the developed model for certain classes can be further improved. For example, the developed model has problems when the angle of the recorded images is varied. To solve this issue, the image should be processed into a 3D mesh of a hand. This gives richer information for the neural network to train instead of a 2D hand image.

iv. The method of this project has been to develop a model solely on a CNN architecture. To acquire the best prediction model, one should train multiple models and find the best architecture from them.

v. Finally, this system should be deployed as a web-based application. The process of collecting datasets becomes unrestricted and anyone with internet access can participate as a volunteer. In addition, the deaf community will have

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

unrestricted access to the system and they can quickly access the system when needed.

UTeM
اونيۏرسيتي تيکنيکل مليسيا ملاک
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

REFERENCES

[1] M. K. T. &. S. R. Al-Qurishi, "Deep Learning for Sign Language Recognition: Current Techniques, Benchmarks, and Open Issues.," IEEE Access, 2021.

[2] Jabatan Kebajikan Masyarakat, "Portal Rasmi Jabatan Kebajikan Masyarakat," Jabatan Kebajikan Masyarakat, 30 11 2022. [Online]. Available: https://www.jkm.gov.my/jkm/index.php?r=portal/full&id=ZUFHVTB1NnJWM0EreGtwNC9Vb1hvdz09. [Accessed 1 1 2023].

[3] C. V. Yee, "Development of Malaysian Sign Language in Malaysia," *Journal of Special Needs Education,* vol. 8, pp. 15-24, 2018.

[4] A. a. P. K. Wadhawan, "Sign language recognition systems: A decade systematic literature review.," *Archives of Computational Methods in Engineering,* vol. 28, no. 3, pp. 785-813, 2021.

[5] R. R. D. L. P. a. V. C. Hills, The Gallaudet dictionary of American sign language., Gallaudet University Press, 2021.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

[6] J. R. P. a. A. H. e. Quer, "1.2.3 Location," in *The Routledge Handbook of Theoretical and Experimental Sign Language Research*, Routledge, 2021.

[7] J. L. N. S. Mohammad, "Smart Glove Malaysian Sign Language Translator.," *Evolution in Electrical and Electronic Engineering,* vol. 2, no. 2, pp. 57-64, 2021.

[8] K. S. G. &. A. R. Van Murugiah, "Wearable IOT based Malaysian sign language recognition and text translation system.," *Journal of Applied Technology and Innovation,* vol. 5, no. 4, p. 51, 2021.

[9] K. D. a. P. D. D. Konstantinidis, "Sign language recognition based on hand and body skeletal data.," in *2018-3DTV-Conference: The True Vision-Capture, Transmission, and Display of 3D Video (3DTV-CON)*, 2018.

[10] P. A. D. G. D. P. L. a. R. C. R. J. A. Deja, "MyoSL: A Framework for measuring usability of two-arm gestural electromyography for sign language.," in *Proc. International Conference on Universal Access in Human-Computer Interaction*, 2018.

[11] P. D. e. a. Rosero-Montalvo, "Sign language recognition based on intelligent glove using machine learning techniques.," in *2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM)*, 2018.

[12] &. Y. X. K. Bantupalli, "American Sign Language Recognition using Deep Learning and Computer Vision.," in *IEEE International Conference on Big Data*, 2018.

[13] B. T. P. D. T. T. P. H. N. B. A. &. T. N. S. Duy Khuat, "Vietnamese sign language detection using Mediapipe.," in *10th International Conference on Software and Computer Applications*, 2021.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

[14] A. a. P. K. Wadhawan, "Deep learning-based sign language recognition system for static signs," *Neural computing and applications .,* vol. 32, no. 12, pp. 7957-7968, 2020.

[15] A. S. L. a. L. K. Y. Liew, "Gesture Recognition-Malaysian Sign Language Recognition using Convolutional Neural Network.," in *International Conference on Digital Transformation and Applications*, 2020.

[16] &. O. J. I. Quinn M., "British sign language recognition in the wild based on multi-class SVM.," in *Federated Conference on Computer Science and Information Systems*, 2019.

[17] A. S. K. &. Y. T. Chaikaew, "Thai sign language recognition: an application of deep neural network.," in *Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering*, 2021.

[18] Z. L. F. Y. W. P. S. &. Z. J. Li, "A survey of convolutional neural networks: analysis, applications, and prospects.," *IEEE Transactions on Neural Networks and Learning Systems,* 2021.

[19] M. M. I. M. S. R. M. H. S. R. R. M. W. &. A. M. Rahman, "A new benchmark on American Sign Language recognition using convolutional neural network.," in *International Conference on Sustainable Technologies for Industry 4.*, 2019.

[20] T. D. a. M. V. B. Sajanraj, "Indian sign language numeral recognition using region of interest convolutional neural network," in *Second*

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018.

[21] P. N. Stuart Russell, "Deep Learning," in *Artificial intelligence a modern approach.*, Pearson Education, Inc., 2022, pp. 801-839.

[22] G. e. a. Latif, "An automatic Arabic sign language recognition system based on deep CNN: an assistive system for the deaf and hard of hearing.," *International Journal of Computing and Digital Systems ,* vol. 9, p. 4, 2020.

[23] N. T. S. S. a. B. K. Nguen, "Deep CNN-based recognition of JSL finger spelling.," in *International Conference on Hybrid Artificial Intelligence Systems*, 2019.

[24] M. A. e. a. Hossen, "Bengali sign language recognition using deep convolutional neural network," in *2018 joint 7th international conference on informatics, electronics & vision (iciev) and 2018 2nd international conference on imaging, vision & pattern recognition (icIVPR)*, 2018.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

APPENDICES

APPENDIX A: Code for RoI Detection and Drawing Algorithm

```
        #if hand is successfully processed
        if handResult:
            #draw handlandmark on video
            x_min, x_max, y_min, y_max = self.drawBox(h, w, handResult)
            cropped_image = frame[y_min:y_max, x_min:x_max]
            cropped_image = cv2.flip(cropped_image, 1)

            cv2.rectangle(Image, (x_min, y_min), (x_max, y_max), (0, 255, 0),
        1)
            mpDraws.draw_landmarks(Image, handResult,
        mpHolistic.HAND_CONNECTIONS,

        mpDrawingStyles.get_default_hand_landmarks_style(),

        mpDrawingStyles.get_default_hand_connections_style())

def drawBox(self, h, w, hand_landmarks):
    x_max = 0
    y_max = 0
    x_min = w
    y_min = h
    for c in range(21):
        x = int(hand_landmarks.landmark[c].x * w)
        y = int(hand_landmarks.landmark[c].y * h)
        if x > x_max:
            x_max = x
        if x < x_min:
            x_min = x
        if y > y_max:
            y_max = y
        if y < y_min:
            y_min = y
    return x_min-15, x_max+15, y_min-15, y_max+15
```

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

APPENDIX B: Code for Dataset Preparation

```
class PrepareImageDataset:
    def SplitDataset(self, imgPerGesture, train):
        test = 1-train

        rootPath = os.path.dirname(os.path.abspath(__file__))
        originDatasetPath = rootPath + "\landmarkDataset"
        targetDatasetPath = rootPath + "\\torchDataset"
        trainDatasetPath = targetDatasetPath+"\\training"
        trainLabelPath = trainDatasetPath + "\\train.csv"
        testDatasetPath = targetDatasetPath + "\\test"
        testLabelPath = testDatasetPath + "\\test.csv"

        if not os.path.exists(trainDatasetPath):
            os.makedirs(trainDatasetPath)
        if not os.path.exists(testDatasetPath):
            os.makedirs(testDatasetPath)
        if not os.path.exists(trainLabelPath):
            f = open(trainLabelPath, 'w')
            f.close()
        if not os.path.exists(testLabelPath):
            f = open(testLabelPath, 'w')
            f.close()

        trainRows = []
        testRows = []
        counter = 0
        for file in os.listdir(originDatasetPath):
            if file.endswith(".jpg"):

                counter = counter + 1
                filename = file
                labelIndex = 0
                numberIndex = 0
                for x in range(3):
                    filename = filename[1:]
                    imgLabelIndex = filename.find(" ")
                    filename = filename[imgLabelIndex:]
                    numberIndex = numberIndex+imgLabelIndex+1
                    if x == 1:
                        labelIndex = numberIndex

                imgLabelIndex2 = file.find(".jpg")
                imageLabel = file[labelIndex:numberIndex]
                imageClass = 0

                for x in range(len(classes)):
                    if classes[x] is imageLabel[1:]:
                        imageClass = x
                imgLabelCounter = file[numberIndex:imgLabelIndex2]

                if int(imgLabelCounter) <= train*imgPerGesture/2:
                    print("copy:" + file[labelIndex:] )
                    trainRows.append([file, imageClass])
                    shutil.copy(originDatasetPath + "\\" + file,
trainDatasetPath + "\\" + file)
                elif int(imgLabelCounter) >
imgPerGesture/2+test*imgPerGesture/2:
                    print("copy:" + file[labelIndex:] )
                    trainRows.append([file, imageClass])
                    shutil.copy(originDatasetPath + "\\" + file,
```

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

APPENDIX C: Code for Training Neural Network

```
model.to(device)

criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=learning_rate)

if load_Model:
    TrainingNN.load_checkpoint(torch.load(rootPath + "\\" + model_Name),
model, optimizer)

# Train Network
for epoch in range(num_epochs):
    losses = []

    if epoch % 4 == 0 and epoch > 1:
        checkpoint = {'state_dict': model.state_dict(), 'optimizer':
optimizer.state_dict()}
        TrainingNN.save_checkpoint(checkpoint, filename="\\" + model_Name)
        TrainingNN.create_confusionMatrix(test_loader, model)

    for batch_idx, (data, targets) in enumerate(train_loader):
        # Get data to cuda if possible
        data = data.to(device=device)
        targets = targets.to(device=device)

        # forward
        scores = model(data)
        loss = criterion(scores, targets)

        losses.append(loss.item())

        # backward
        optimizer.zero_grad()
        loss.backward()

        # gradient descent or adam step
        optimizer.step()

    print(f"Cost at epoch {epoch} is {sum(losses) / len(losses)}")
    print("Checking accuracy on Training Set")
    TrainingNN.check_accuracy(train_loader, model)

    print("Checking accuracy on Test Set")
    TrainingNN.check_accuracy(test_loader, model)

checkpoint = {'state_dict': model.state_dict(), 'optimizer':
optimizer.state_dict()}
TrainingNN.save_checkpoint(checkpoint, filename="\\" + model_Name)
TrainingNN.create_confusionMatrix(test_loader, model)
```

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

APPENDIX D: Code for Real-Time Test

```
class Worker3(QThread):
    global cameraID, cameraButtonState, mpHolistic, mpDraws,
mpDrawingStyles, holistic, modelName, tempPath

    ImageUpdate = pyqtSignal(numpy.ndarray, str, str, str, int)
    ImageRoI = pyqtSignal(QImage)

    def run(self):
        self.loadModel()
        self.Capture = cv2.VideoCapture(cameraID)
        self.ThreadActive = True

        while self.ThreadActive:
            ret, frame = self.Capture.read()
            takenImage = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

            handResult = 0
            h, w, c = takenImage.shape
            results = holistic.process(takenImage)

            if results.right_hand_landmarks:
                handResult = results.right_hand_landmarks
            if results.left_hand_landmarks:
                handResult = results.left_hand_landmarks
            if handResult:
                x_min, x_max, y_min, y_max = self.drawBox(h, w, handResult)
                cv2.rectangle(takenImage, (x_min, y_min), (x_max, y_max),
(0, 255, 0), 1)

                if x_min>0 and x_max<w and y_min>0 and y_max<h:
                    # create empty canvas and retrace image
                    canvas = np.zeros((x_max - x_min, y_max - y_min, 3),
np.uint8)
                    landmarkIndex = 0
                    for landmarkIndex in range(21):
                        handResult.landmark[landmarkIndex].x =
(((handResult.landmark[

landmarkIndex].x) / 1 * 640) - x_min) / (

x_max - x_min)
                        handResult.landmark[landmarkIndex].y =
(((handResult.landmark[

landmarkIndex].y) / 1 * 480) - y_min) / (

y_max - y_min)
                    mpDraws.draw_landmarks(canvas, handResult,
mpHolistic.HAND_CONNECTIONS,

mpDrawingStyles.get_default_hand_landmarks_style(),

mpDrawingStyles.get_default_hand_connections_style())
                    canvas = cv2.resize(canvas, (100, 100))
                    canvas = cv2.flip(canvas, 1)

                    cv2.imwrite(tempPath+"\\temp.png", canvas)
                    im_pil = io.imread(tempPath+"\\temp.png")
                    im_pil = Image.fromarray(im_pil)
                    imageClass, conf = self.predictImageClass(im_pil)
```

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

APPENDIX E: Binary-Class Confusion Matrix Result

Class A		Expected	
		Positive	Negative
Predicted	Positive	30	0
	Negative	0	840.00

Class B		Expected	
		Positive	Negative
Predicted	Positive	29	1
	Negative	0	840

Class N		Expected	
		Positive	Negative
Predicted	Positive	29	1
	Negative	10	830

Class C		Expected	
		Positive	Negative
Predicted	Positive	30	0
	Negative	0	840

Class D		Expected	
		Positive	Negative
Predicted	Positive	30	0
	Negative	1	839

Class P		Expected	
		Positive	Negative
Predicted	Positive	30	0
	Negative	1	839

Class E		Expected	
		Positive	Negative
Predicted	Positive	30	0
	Negative	0	840

Class F		Expected	
		Positive	Negative
Predicted	Positive	30	0
	Negative	0	840

Class R		Expected	
		Positive	Negative
Predicted	Positive	30	0
	Negative	5	835

Class G		Expected	
		Positive	Negative
Predicted	Positive	30	0
	Negative	2	838

Class H		Expected	
		Positive	Negative
Predicted	Positive	27	3
	Negative	0	840

Class T		Expected	
		Positive	Negative
Predicted	Positive	29	1
	Negative	0	840

Class I		Expected	
		Positive	Negative
Predicted	Positive	29	1
	Negative	1	839

Class K		Expected	
		Positive	Negative
Predicted	Positive	29	1
	Negative	0	840

Class V		Expected	
		Positive	Negative
Predicted	Positive	26	4
	Negative	0	840

Class L		Expected	
		Positive	Negative
Predicted	Positive	30	0
	Negative	0	840

Class M		Expected	
		Positive	Negative
Predicted	Positive	22	8
	Negative	1	839

Class X		Expected	
		Positive	Negative
Predicted	Positive	27	1
	Negative	0	842

Class O		Expected	
		Positive	Negative
Predicted	Positive	30	0
	Negative	0	840

Class Bapa		Expected	
		Positive	Negative
Predicted	Positive	30	0
	Negative	0	840

Class Q		Expected	
		Positive	Negative
Predicted	Positive	27	3
	Negative	0	840

Class Emak		Expected	
		Positive	Negative
Predicted	Positive	30	0
	Negative	0	840

Class S		Expected	
		Positive	Negative
Predicted	Positive	28	2
	Negative	5	835

Class Melayu		Expected	
		Positive	Negative
Predicted	Positive	30	0
	Negative	0	840

Class U		Expected	
		Positive	Negative
Predicted	Positive	26	4
	Negative	3	837

Class Cina		Expected	
		Positive	Negative
Predicted	Positive	30	0
	Negative	0	840

Class W		Expected	
		Positive	Negative
Predicted	Positive	30	2
	Negative	2	836

Class India		Expected	
		Positive	Negative
Predicted	Positive	30	0
	Negative	0	840

Class Y		Expected	
		Positive	Negative
Predicted	Positive	30	0
	Negative	1	839

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

APPENDIX F: Sample Images Recorded in Dataset

UNIVERSITI TEKNIKAL MALAYSIA MELAKA