Automated Hand Gesture Recognition for Enhancing Sign Language Communication

By

Lee Teck Junn

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)
Faculty of Information and Communication Technology
(Kampar Campus)

JAN 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Automated Hand Gesture Recognition for Enhancing Sign Language Communication

Academic Session: Jan 2024

I LEE TECK JUNN

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature) (Supervisor's signature)

Address:

52, Taman Pelangi 2,
Lorong Pelangi 2, 32000
Sitiawan, Perak, Malaysia

Ts. Dr. Vikneswary Jayapal
Supervisor's name

Date: 23/4/2024 **Date**: 24/04/24

FACULTY/INSTITUTE* OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

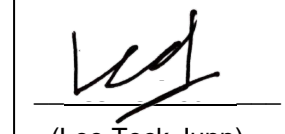
Date: 22/4/2024

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that **Lee Teck Junn** (ID No: **2002030**) has completed this final year project/ dissertation/ thesis* entitled "Automated Hand Gesture Recognition for Enhancing Sign Language Communication" under the supervision of Dr. Vikneswary a/p Jayapal (Supervisor) from the Department of Computer and Communication Technology, Faculty/Institute* of Information and Communication Technology

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,



(Lee Teck Junn)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Automated Hand Gesture Recognition for Enhancing Sign Language Communication**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : 

Name : Lee Teck Junn

Date : 22/4/2024

ACKNOWLEDGEMENTS

I extend my heartfelt gratitude and appreciation to my supervisors, Dr. Vikneswary a/p Jayapal and Dr. Ashvaany a/p Egambaram, for providing me with the invaluable opportunity to participate in an AI project. This experience marks my initial stride towards establishing a career in the field of AI. A million thanks to you.

Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

This paper introduces a novel approach aimed at enhancing communication between individuals who are deaf or hard of hearing and those unfamiliar with sign language. The project addresses this challenge by developing a mobile application that harnesses the power of smartphone cameras, coupled with a deep learning model, to interpret hand gestures and provide real-time contextual information to users. It emphasizes the widespread adoption of smartphones and the practical applicability of mobile applications in real-life scenarios. Furthermore, the paper proposes a new methodology leveraging Google's MediaPipe, which outperforms traditional approaches such as transfer learning with pre-trained object detection models in deep learning model development. Of paramount importance is the seamless integration of the deep learning model with the mobile application, enabling real-time detection and recognition on the mobile application.

TABLE OF CONTENTS

PLAGIARISM CHECK RESULT

Place your Turnitin plagiarism checking result here.

CHECK LISTS

Place the check lists here at end of report. Check items using '√' and sign.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
AR	Augmented Reality
ASL	American Sign Language
SDK	Software Development Kit
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks
LSTM	Long Short-term Memory Networks
GRU	Gated Recurrent Units
SSD	Single Shot MultiBox Detector

CHAPTER 1

Introduction

1.1 Background Information

In recent times, there have been significant advancements in technologies like **artificial intelligence (AI)** and **augmented reality (AR)**. These advancements have led to creative ways of solving problems in the real world. One really important problem is **communication obstacles** that people who are **deaf, i.e., those who are facing hearing difficulty encounters during communication,** especially **with those who unfamiliar with sign language**. Simultaneously, with the rise of smartphones that are everywhere, and these technologies becoming more common, this project wants to tackle this problem. The idea is to use these advancements to create a special kind **of mobile application using AR and AI**. This could potentially change how the deaf and those who are facing hearing difficulty connects with the world around them.

At its core, this project wants to make an AR-based mobile app. It is designed to use a combination of **deep learning and computer vision techniques**, to make computers/ machines intelligent in such a way to interpret and visualize given inputs. In the realm of understanding, deep learning stands as a method in artificial intelligence (AI) that teaches computers to process data in a way that is inspired by the human brain [1]. Deep learning models can recognize complex patterns in pictures, text, sounds, and other data to produce accurate insights and predictions [1]. For visual representations, refer to Figure 1.1.

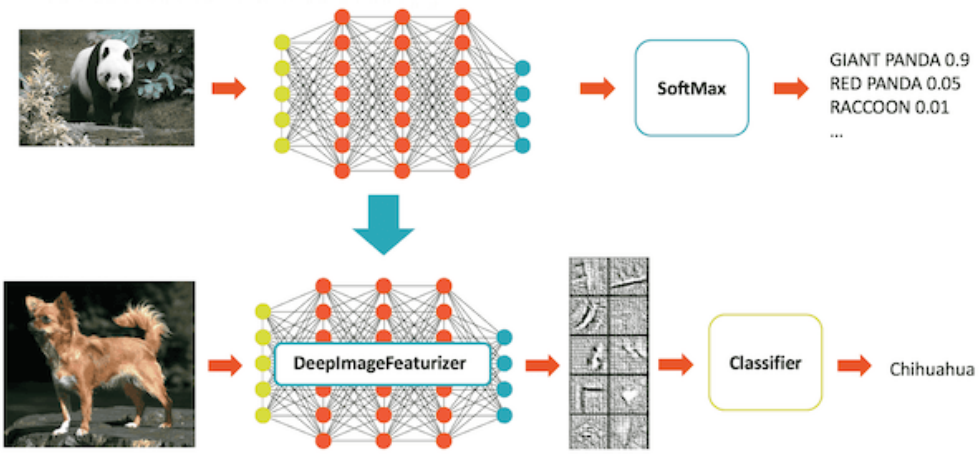


Figure 1.1 Deep learning pipeline. Adapted from [2]

On a different note, computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos, and other visuals [3] as depicted in Figure 1.2. By using these tools, the app aims to visually interpret and comprehend the intricate hand movements that come from sign language. This could be really impactful: an app that can smartly turn these hand movements into understandable communication.

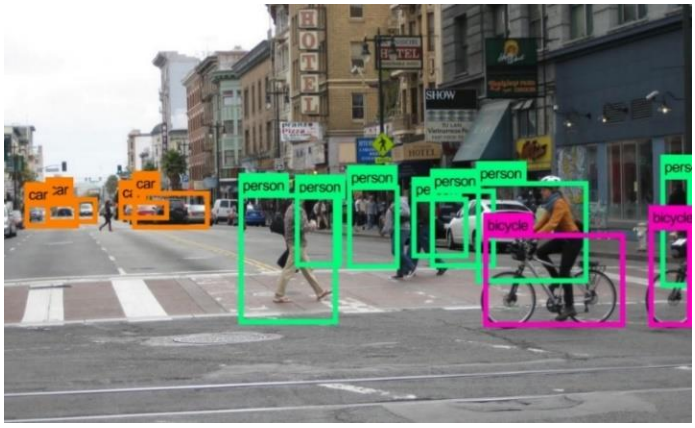


Figure 1.2 Example usage of computer vision. Adapted from [3].

This project brings together the power of AI, AR, deep learning, computer vision, and the universal of smartphones, to solve a long-standing communications challenge. By creating a clear and easy way to understand and use sign language, the project hopes to change how people with hearing difficulties interact with the world. This could help their voices be heard loud and clear, harmonizing with the world around them.

1.2 Problem Statement

The main problem this project aims to address revolves around the difficulties people who are deaf and those who are facing hearing difficulty encounter when communicating. These individuals face significant challenges in effectively communicating, especially when interacting with those who don't understand sign language like **American Sign Language (ASL)**. This issue isn't limited to a certain age group or region, it affects people from diverse backgrounds and cultures, across the globe. According to the World Health Organization, 430 million deafness and hearing

loss are widespread and found across all regions and nations [4]. Currently more than 1.5 billion people (nearly 20% of the global population) live with hearing loss [4].

The importance of addressing this problem arises from its impact on the lives of individuals who are deaf and those who are facing hearing difficulty. Communication is essential for social interaction and participating in daily activities, enabling the exchange of ideas, emotions, and experiences. When people who are deaf and those who are facing hearing difficulty face communication obstacles, it can lead to exclusion from conversation, education, job opportunities, and even basic services. This exclusion not only affects their personal well-being but also weakness the richness of diversity in society and hinders social integration.

Past efforts to address this issue have often been limited in their effectiveness and widespread use. Specialized devices for translating sign language have been bulky, expensive, and required additional hardware. Traditional text-based solutions lack the visual and emotional nuances of sign language, and speech recognition systems aren't suitable for individuals who rely on non-verbal communication. These shortcomings highlight the need for an innovative approach that uses the latest technological advancements, like augmented reality (AR) and artificial intelligence (AI), to create a more seamless and inclusive communication experience.

1.3 Motivation

The motivation for embarking on this project stems from the incredible opportunity to transform communication for individuals who are deaf and those who are facing hearing difficulty. While earlier research endeavours have laid the groundwork for innovation in this area, what makes this project truly exceptional is its novel approach that combines augmented reality (AR) and artificial intelligence (AI) to devise an entirely new solution. This project serves as evidence of the fusion between cutting-edge technologies and real-world challenges, offering the potential to bring about tangible improvements in the lives of many.

What distinguishes this project is its use of AR technology, which has rapidly advanced and become accessible on common smartphones. Unlike previous attempts that relied on specialized and often costly equipment, this project embraces the idea of making technology available to a broader audience. This shift from exclusive hardware

to widely available devices mark a significant leap forward, making the benefits of AR technology more inclusive and impactful than ever before.

In this era of technological advancement, AI-powered systems like ChatGPT have demonstrated the impressive capabilities of AI to comprehend and respond in natural language strongly supports the feasibility of this project, as it emphasizes the real potential of AI-driven solutions.

The combination of AI and AR is no longer a distant goal; it's a hopeful path that can lead to real-world positive changes. It holds the promise of creating an application that is both inventive and practical, aimed at addressing the communication barriers faced by those who are deaf and those who are facing hearing difficulty in a way that's easy for users.

1.4 Project Objectives

To achieve the goals of this project, the following objectives have been outlined:

I. To train a deep learning model that can accurately recognize hand gestures in real-time.
II. To develop an augmented reality (AR) mobile application.
III. To integrate the augmented reality (AR) mobile application with deep learning model.

1.5 Project Scope and Direction

The proposed project, titled "Automated Hand Gesture Recognition for Enhancing Sign Language Communication" aims to deliver a real-time augmented reality (AR) mobile application that utilizes deep learning and computer vision techniques for hand gesture recognition to enhance sign language communication. This application will serve as a solution to the existing problem faced between individuals who do not understand sign language and deaf and hard-of-hearing community in effectively communicating.

The proposed solution also involves creating a user-friendly mobile application that recognizes and interprets hand gestures, translating them into text or audio output to facilitate seamless communication between the deaf and hard-of-hearing community and those unfamiliar with sign language. The application will leverage computer vision techniques and deep learning models to recognize a range of hand gestures used in sign language. This solution involves real-time processing of gestures captured through the device's camera, converting them into text or spoken language, which can then be easily understood by non-sign language users.

The project scope encompasses the following deliverables:

I. Implementation of sequential deep learning models with MediaPipe to accurately recognize and interpret gestures in real-time with computer vision technique.
II. Development of mobile application to translate sign language to meaningful context using Android Studio.
III. Integration of the mobile application with deep learning model through TensorFlow Lite.

1.6 Contributions

This project brings several significant contributions that hold the potential to make a meaningful impact. Its primary aims are to improve the lives of people who are deaf or have difficulty hearing by providing them with a powerful tool to communicate effectively with individuals who don't know sign language. Through accurate hand gesture recognition using AR and AI, this project makes communication more inclusive and accessible, overcoming the barriers that have long hindered their interactions.

Additionally, the project advances technology in both AR and AI fields. It introduces an innovative approach by coming AR and AI, highlighting how these technologies can work together to create new solutions. Success in this project could lead to better AR systems that suit diverse needs and encourage further exploration of AI-driven applications for real-time visual recognition.

Moreover, the project's applications are extensive. While it directly improves sign language communication, the technology developed could also have broader uses.

It might revolutionize education by creating interactive learning experiences, assist in making public services more accessible, and find applications in entertainment and gaming, offering users unique and innovative interfaces.

The project also opens up a new field of study at the intersection of AR, AI and sign language communication enhancement. This interdisciplinary focus encourages researchers and developers to delve into uncharted territories, fostering the growth of a specialized area that has the potential to transform how people communicate and connect.

1.7 Report Organization

The report is structured into distinct chapters, with the **first chapter** serving as the introduction. This section outlines the problem statement and motivation behind the project, delves into the project's scope and objectives, explores its impact, significance, and contribution, and provides relevant background information.

The **second chapter** of the report focuses on the literature review, shedding light on prior research and works related to the problem at hand. It aims to highlight the strengths and limitations of previous studies, emphasizing the gaps that necessitate the introduction of the new proposed method in this project.

In the **third chapter**, the report introduces the proposed method and approach for the project. It elaborates on the methodologies and procedural frameworks, offering a comprehensive insight into the system design proposed for both approaches.

In the **fourth chapter**, the report elucidates the entire process involved in system design, delineating between two methods: a novel approach and a conventional one. It delineates the development of the project, furnishing all requisite details pertinent to the system.

In the **fifth chapter**, the report demonstrates the implementation of the system. It also presents the setup, configuration, and operational aspects of the system, alongside a discussion of implementation challenges and issues.

In the **sixth chapter**, the report conducts a comprehensive evaluation of the system and discusses the entirety of the project. This includes system testing,

performance metrics analysis, and an examination of project challenges and the evaluation of the objectives.

The **final chapter** serves as the conclusion for the project, offering a comprehensive summary of the findings and outcomes. It discusses areas that can be improved based on the project's current status and outlines potential avenues for future work and research initiatives.

CHAPTER 2

Literature Reviews

2.1 Previous Works on Deep Learning

2.1.1 Static and Dynamic Hand-Gesture Recognition for Augmented Reality Applications [5]

This paper explores into a fresh approach regarding systems that can recognize gestures, especially in the context of Augmented Reality (AR). The central focus on this research lies in the development of an advanced automatic gesture recognition system. The principal objective is to seamlessly integrate virtual content into the user's actual surroundings, aiming for a heightened level of realism. This pursuit is driven by the aspiration to establish a fully immersive Augmented Reality (AR) application, intending to replace conventional non-natural interaction methods like mice or keyboards with intuitive human interaction techniques such as speech or gestures.

This system is designed to differentiate between static (gestures that are held in one position) and dynamic (gestures involve movement) gestures. The core of this system is a tracking mechanism that employs infrared technology, instrumented infrared (IR) tracking system (ITS) containing a six-camera array. This involves attaching small infrared markers, referred to as light-weighted IR-tracking targets (IRTTs) in the paper, to the user's thumbs and index fingers, allowing the system to understand where and how the fingers are positioned, as illustrated in Figure 2.1.

Figure 2.1 IRTTs mounted at user's fingers. Adapted from [5]

In the paper, there is a clear distinction is made between two types of gestures: static and dynamic. These two types vary based on the angle formed between the user's fingers. Static gestures maintain a constant finger angle that remains unchanged over time, such as pointing or grasping gestures. On the other hand, dynamic gestures involve altering finger angles as time progresses, like waving or forming letters in the air. It's noteworthy that the hand's position can change for both static and dynamic gestures during their execution.

The outcomes of these identified gestures are subsequently transmitted to interconnected applications for further processing and subsequent actions. As an illustration, the paper presents a demonstration application featuring virtual building blocks seamlessly incorporated into the Augmented Reality (AR) environment, like children's toys as depicted in Figure 2.2. These virtual building blocks can be manipulated using the user's real hand, simulating the act of manipulating an actual kit with a high degree of realism. Within this application context, the study proceeds to evaluate the outcomes of their assessments, including ratings and task execution times. The comparison is drawn between non-natural interaction methods such as mouse/keyboard usage and gesture recognition. The summarized results of these evaluations are presented in Table 1.

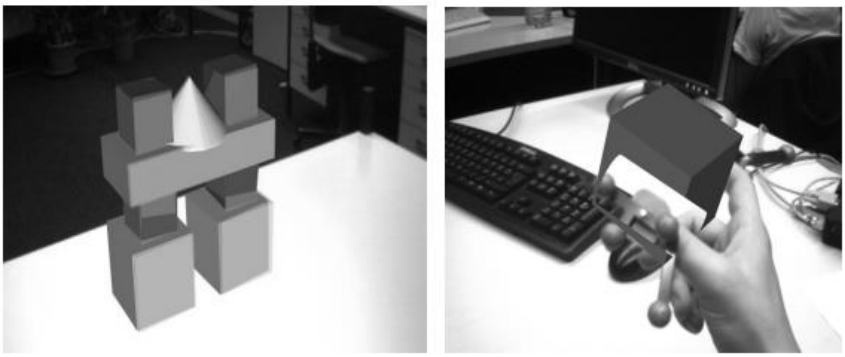


Figure 2.2 Example of demonstration application. Adapted from [5].

Table 1.1 Subject's mean rating and task execution time

	Reality	Mouse/Keyboard	Gesture Recognition
Task Execution time in [s]	9	89	57
Intuitiveness	5	1,8	4
Comfort	5	1,9	1,5

The results demonstrate a noteworthy decrease in the average time needed to complete tasks by about one-third when using proposed gesture recognition system compared to typical mouse/keyboard arrangement. Additionally, the outcomes emphasize the interacting in the real world is not only more intuitive but also more comfortable when it comes to accomplishing the assigned task.

Nevertheless, the paper stipulates that their demonstration application possesses the capability to identify solely two dynamic gestures. These gestures are executed by either the right or left hand, forming the symbols "X" or "O" in the air. Additionally, a significant limitation arises from the reliance on specialized equipment, light-weighted IR tracking targets (IRTTs). These components are not widely prevalent and accessible, which could hinder the broader adoption of the proposed system. Furthermore, the incorporation of such equipment may entail additional costs, potentially posing a financial barrier for users and limiting the system's reach and applicability.

2.1.2 Real Time Hand Gesture Recognition System for Dynamic Applications [6]

The paper introduces a hand gesture recognition system that works in real-time and is designed for dynamic applications. The authors highlight the effectiveness of using hand gestures to control virtual environments, allowing users to interact with objects using natural gestures. The system is built upon computer vision methods that help accurately track and recognize these hand gestures, replacing the need for traditional

tools like keyboards and mice or even additional tools such as physical markers and additional equipment on user's hand.

Figure 2.3 shows the application architecture design for manipulating virtual objects using hand gestures. The way the system is structed involves using a webcam to capture a series of images. These images then go through initial processing steps, including removing the background and identifying the colour of the skin. To keep track of hand movements, the system uses a technique called camshaft. When it comes to recognizing specific gestures, a method like the Haar technique is used. Haar techniques is done by finding out the difference of the average of the pixel values at the darker region and the average of the pixel values at the lighter region [7].

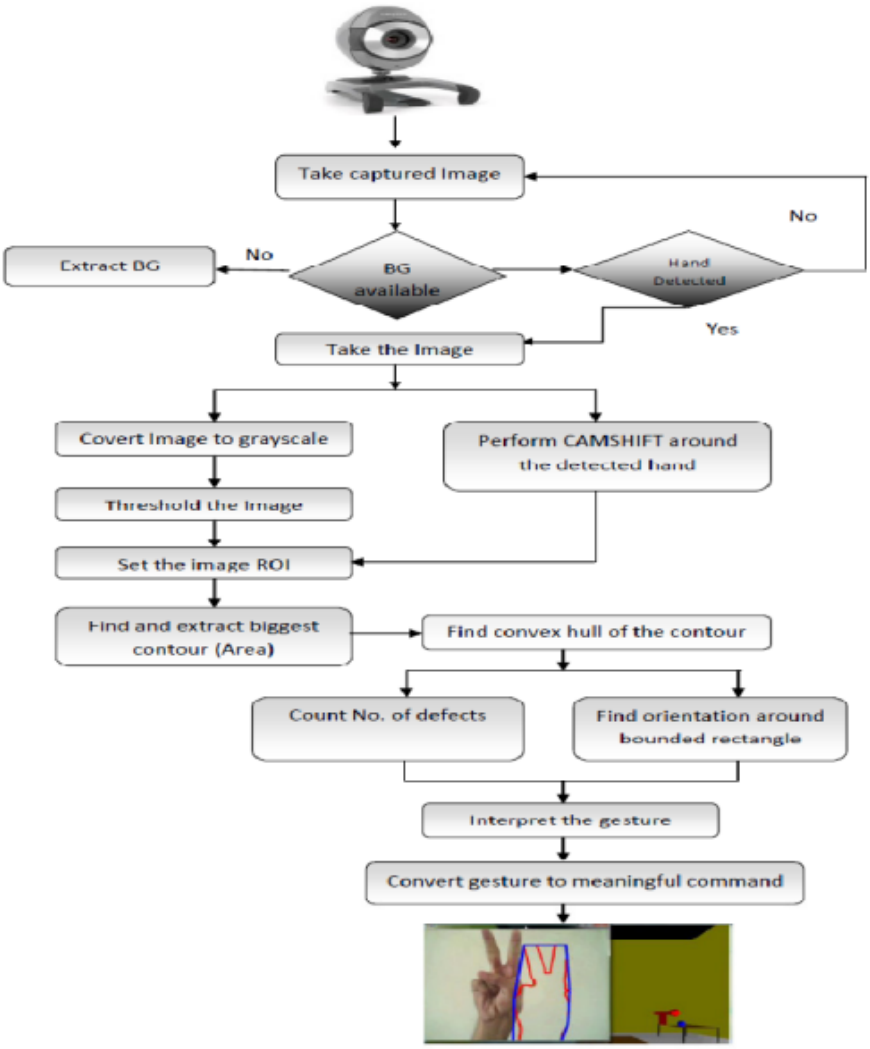


Figure 2.3 Application architecture design. Adapted from [6].

The paper demonstrates the system's efficiency by presenting various gestures that can control virtual objects in real time. The user initiates interaction by placing their hand in front of the webcam, which then detects the hand by creating a bounding rectangle around it. Once the hand is detected, the application tracks various gestures performed by the user's hand and generates contours around it. These gestures include moving left, moving right, moving up, and moving down, each associated with a specific function for manipulating virtual objects, as showed in Figure 2.4.

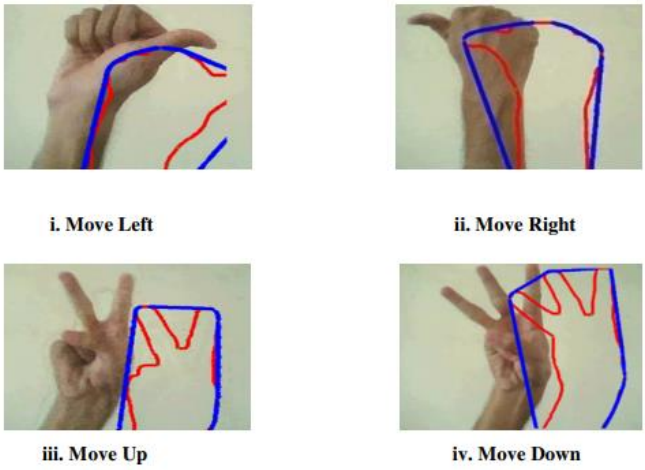


Figure 2.4 Gestures for manipulating virtual objects. Adapted from [6]

The following figures clearly demonstrate the results achieved by using different hand gestures to effectively control objects in virtual environments. These movements correspond to specific commands and successfully lead to the intended actions.

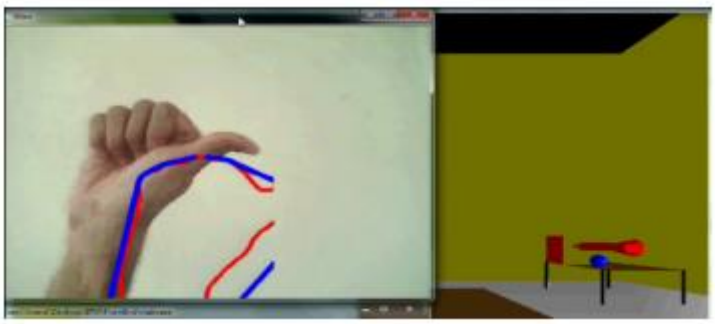


Figure 2.5 Gesture for moving left. Adapted from [6].

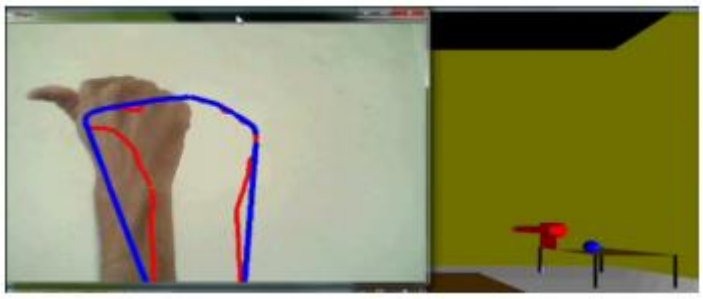


Figure 2.6 Gesture for moving right. Adapted from [6].

Figure 2.7 Gesture for moving up. Adapted from [6]

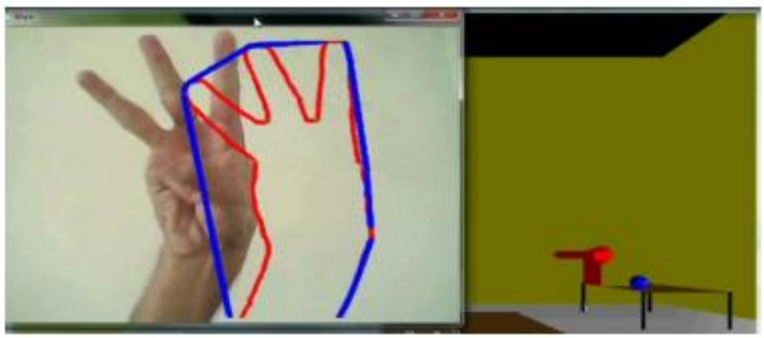


Figure 2.8 Gesture for moving down. Adapted from [6]

A notable weakness of this paper lies in its limited exploration of challenges related to noisy environments. Although the practical experiments display positive results, they only take place in controlled conditions with low levels of noise, specifically objects that resemble human skin colour, and with consistent lighting. This omission means that the paper doesn't thoroughly examine how well the system would work in more complicated and diverse real-life situations. These could involve various

types of noises, changing lighting, and obstructions, all of which could potentially reduce the accuracy and dependability of the proposed gesture recognition system.

2.1.3 SignAR: A Sign Language Translator Application with Augmented Reality using Text and Image Recognition [8]

The paper highlights the significant differences that students with hearing impairments face when it comes to understanding what they read. In response to this issue, a new solution called SignAR has been introduced. SignAR is designed to help children with hearing impairments learn both English and sign language. What's interesting is that this solution can also be used by people who can hear to learn sign language. How SignAR works is quite interesting, it uses technology that combines augmented reality (AR) and smartphones. Basically, SignAR use the camera on smartphone to capture words, and then it shows animations of the corresponding sign language on the screen. Each word is carefully connected to its matching animation, creating a way to learn that interactive and well-coordinated, as illustrated in Figure 2.9.

Figure 2.9 Example of SignAR displaying animations. Adapted from [8].

This is accomplished through the utilization of a software development kit (SDK) such as the Vuforia platform, which operates by monitoring and cataloguing real-world objects and aligning them with their virtual counterparts. The process involves taking a picture of the surroundings, identifying a marker, determine the camera's position and angle, and subsequently overlaying a virtual object onto the

image, which is then showcased on the screen. A flowchart outlining the sequence of steps for this straightforward augmented reality system is illustrated in Figure 2.10.

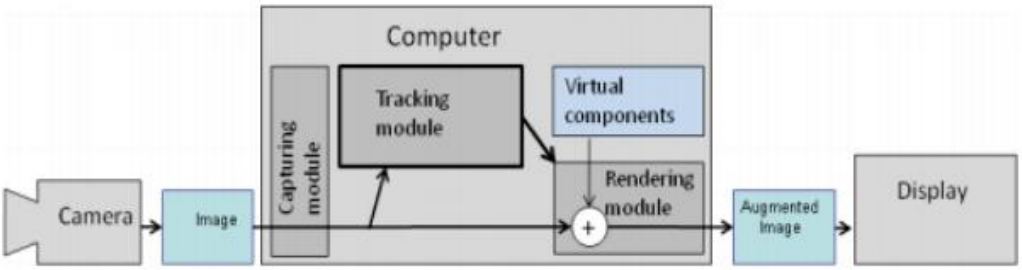


Figure 2.10 Flowchart of simple augmented reality system. Adapted from [8].

In the paper, the authors also explore a newer technique gaining popularity is Natural Feature Tracking, an image-based tracking method that relies on features naturally present in images, like corners, edges, and blobs, without requiring specific markers, as illustrated in Figure 2.11.

Figure 2.11 Example of natural features detected. Adapted from [8].

SignAR acts as a bridge to help hearing-impaired kids learn both English and sign language too. This approach is meant to really decrease the communication difficulties between those who can hear and those who can’t, creating a more inclusive and understanding environment.

The paper has certain limitations that need to be acknowledged. Firstly, its focus lies primarily in translating specific images and words, which restricts it effectiveness in enabling direct communication between individuals who use sign language and those

who do not. Instead, it appears better suited as a tool for learning rather than fully supporting communication. Additionally, there are other ways to learn sign language that might be more convenient and dependable compared to the solution proposed in the paper. As a result, the practical use and adoption of the paper approach could be constrained by the existence of more effective and accessible methods for learning sign language.

2.1.4 American Sign Language Recognition using Deep Learning and Computer Vision [9]

The main goal of this study is to create a computer program that helps people to use sign language communicate with those who don't understand sign language. This program uses deep learning, specifically Convolutional neural Networks (CNNs) and Recurrent Neural Networks (RNNs), to understand and explain American Sign Language (ASL) gestures.

Figure 2.12 showed the high-level system architecture of the program. The process starts by putting videos of ASL gestures into the program. These videos show the moving hands and body positions that make up sign language. The program works in two steps: figuring out the shapes and movements (spatial features) and understanding the order of the gestures (temporal features).

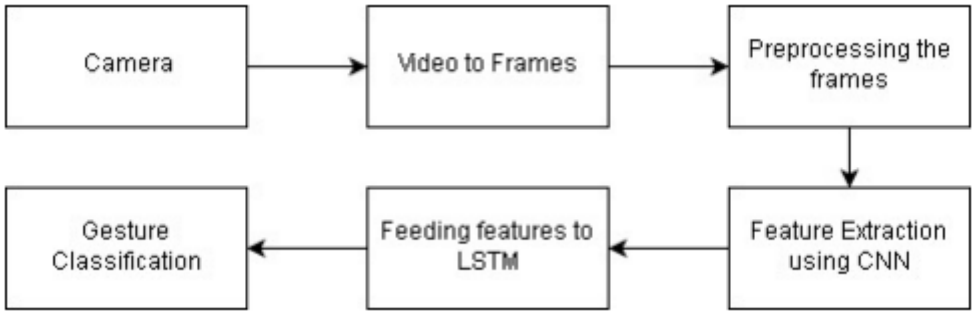


Figure 2.12 High level system architecture. Adapted from [9].

Fiest, it uses a type of CNN called Inception. This CNN is good at looking at images and finding complex patterns. It looks s at each frame of the video to find out

how hands are shaped, how they move, and the positions they take in ASL gestures. This helps the program understand and translate the gestures.

Next, the program focuses on how the gesture happen over time. It uses a special kind of RNN, Long Short-term Memory Networks (LSTM), which is good at handling sequences of data, like frames in a video. This LSTM is trained using the information from the video sequences. This helps the program understand the order in which gestures happen and what they mean.

Finally, by using Inception and LSTM, the program can understand ASL gestures and turn them into text, allowing effective communication between people who using sign language and those who don't.

The accuracy values in the Table 2.2 are based on two different version of the same gesture. The table displays the predictions for each label made by the CNN SoftMax and RNN models.

Table 2.1 Performance with varying sample sizes

# of signs	Accuracy with SoftMax Layer	Accuracy with Pool Layer
10	90%	55%
50	92%	58%
100	93%	58%
150	91%	55%

A drawback of the paper is that it doesn't thoroughly discuss the real-life uses of the developed system. Even though the focus is on the technical details of making a vision-based app for translating sign language, the paper doesn't go into depth about how this system could be applied in practical situations. Due to the limited discussion about various scenarios where the app could be beneficial and the advantages it offers, readers may not completely grasp the significance and practically of the system. The absence of this information leads to uncertainties about how the system could integrated into places such as education, social, or professional settings to enhance communication between sign language users and non-signers.

2.1.5 Interactive Hand Gesture-based Assembly for Augmented Reality Applications [10]

The paper introduces an Augmented Reality (AR) assembly system designed to facilitate the interactive assembly of 3D models of technical systems. The system relies on hand tracking and gesture recognition using the Microsoft Kinect video camera, as illustrated in Figure 2.13. This technology enables users to select, manipulate, and assemble virtual 3D models of mechanical systems.

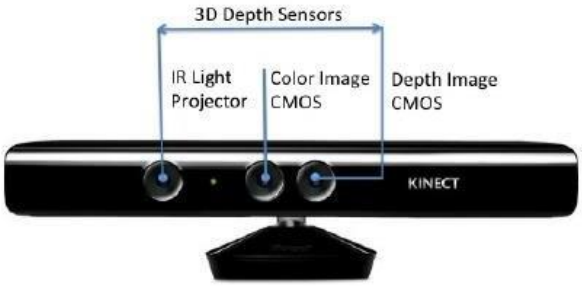


Figure 2.13 Microsoft Kinect camera. Adapted from [11].

The authors emphasize the significance of intuitive interaction techniques in AR applications, especially using hand gestures for natural and instinctive manipulation of virtual objects. They acknowledge the challenges in achieving such interaction, highlighting those previous systems often required physical markers or attachments on user's hands. However, new video cameras like the Microsoft Kinect have enabled free-hand interaction without the need for additional attachments.

The paper outlines the hardware setup of the AR application depicted in Figure 2.14, including a monitor-based AR system and the Kinect video camera. The Kinect camera observes the user's hands and gestures, which are then translated into interactions with virtual objects. Additionally, the authors comprehensively elucidate the interactive selection, manipulation, and assembly techniques within the paper.

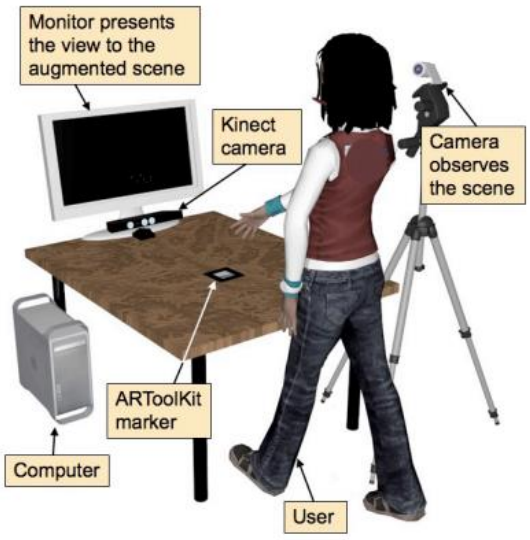


Figure 2.14 Overview of the hardware setup of the AR system. Adapted from [10].

However, the paper does exhibit certain weaknesses and limitations that need to be considered. To start with, the AR application's inability to be easily moved or used in different places is a notable drawback. This makes it less adaptable and useful in practical situations, as it's mostly confined to a specific setting.

Moreover, even though the authors highlight the removal of physical markers or attachments on users' hands as an advantage, the introduction of the Microsoft Kinect camera as an alternative solution raises some concerns. This extra piece of hardware, which allows for interaction using hand movements, but is not universally prevalent and can come with a high cost. This limited availability could make it hard for the system to be more widely accepted and used.

2.2 Limitation of Previous Studies

In summary, the studies we reviewed collectively highlighted several drawbacks that need recognition in the development of successful systems for recognizing gestures and translating sign language. These limitations touch on technological, practical, and accessibility aspects, emphasizing the importance of addressing these issues to create solutions that are more encompassing and reliable.

A common thread throughout these studies in the need to connect technical progress with practical usefulness. While authors such as S. S. Rautaray [6] and K. Bantupalli [9] highlight improvements in gesture recognition technology, they often miss the mark in fully exploring the broader contexts in which these systems would operate. The **lack of thorough discussions about real-world scenarios**, noisy environments, and the diverse needs of users raises doubts about how practical and effective the proposed solutions truly are.

Furthermore, **accessibility** is key concern. For instance, certain studies, like those by S. Reifinger [5] and R. Radkowski [10] that introduce technologies like IRTTs and the Microsoft Kinect camera, present equipment or technologies that might not be widely accessible or affordable, which could limit the extend to which these systems can be adopted by the intended users.

In the field of sign language translation, it’s very important to think about the communication needs of different people. This includes those who are deaf or have trouble hearing, and also those who don’t know sign language. The limitations found in the research on SignAR by N.-I.-N. Soogund [8] show how crucial it is to create solutions that can adapt to different situations, include everyone, and consider cultural differences.

CHAPTER 3

Proposed System Methodology/Approach

In this chapter, we delve into two distinct methodologies proposed for the development of a hand gesture recognition model. Each approach is meticulously explored, employing different techniques and tools to attain the desired outcomes. The methods can be categorized as follows:

(1) Novel Method: Training a sequential deep learning model from scratch utilizing TensorFlow/Keras, augmented by the support of MediaPipe.
(2) Conventional Method: Employing transfer learning by utilizing a pre-trained MobileNet model, supplemented by Single Shot MultiBox Detector (SSD).

3.1 Novel Method: Training Sequential Model assisted with MediaPipe

The **Novel Method** involves training a sequential deep learning model (RNN/LSTM/GRU) using TensorFlow/ Keras, assisted by MediaPipe. The method will commernce by gathering image data, followed by extracting keypoint values that denote hand join positions using MediaPipe. These values will be stored in a numpy array rather than within the image file. Subsequently, a sequential deep learning model will be constructed and trained from the ground up, incorporating sequence layers such as RNN/LSTM/GRU. Ultimately, the model can be tested in real-time scenarios. Figure 3.1 illustrates the general steps of method 1, with further detailed information to be provided in Section 4.1.

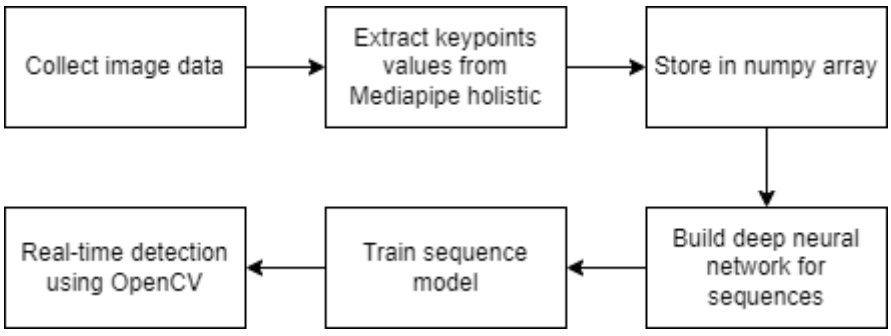


Figure 3.1 General Flow Chart of Method 1

3.2 Conventional Method: Transfer Learning with Pre-trained model

The **Conventional Method** involves leveraging transfer learning by utilizing a pre-trained MobileNet model, supported by object detection algorithms like Single Shot MultiBox Detector (SSD), specifically MobileNet-SSD. Initially, image data will be collected and processed through LabelImg to label the images and generate annotations. These annotated image values will be stored in an XML file for subsequent preprocessing into TF records. The pre-trained MobileNet model is modified to align with the specific requirements of our project, following which the model undergoes training and real-time testing. Figure 3.2 illustrates the general steps of method 2, with further detailed information to be provided in Section 4.2.

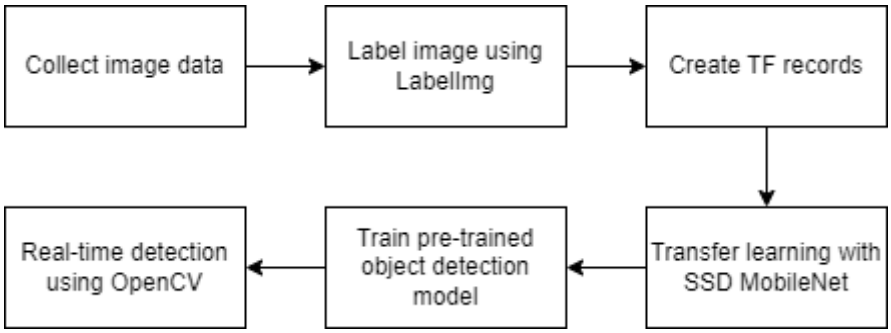


Figure 3.2 General Flow Chart of Method 2

CHAPTER 4

System Design

4.1 Novel Method – Training Sequential Model assisted with MediaPipe

An innovative approach is introduced to construct a hand gesture recognition system, leveraging a sequential deep learning model (RNN/LSTM/GRU) trained with TensorFlow/Keras, assisted by MediaPipe. The incorporation of MediaPipe is pivotal in this methodology, significantly reducing training time with minimal data samples required, and necessitating only a straightforward sequential model structure.

The entire system is divided into various key components, commencing with data collection, followed by preprocessing of data, building, and training the model, conducting evaluations, and ultimately testing in real-time, as illustrated in Figure 3.4.

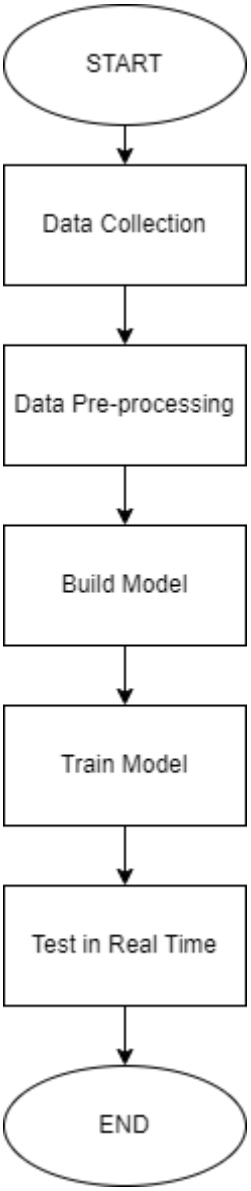


Figure 4.1 High Level System Flow Chart

4.1.1 Data Collection and Pre-processing

For this project, a selection of 10 American Sign Language (ASL) signs has been made, along with a default sign for when the user is not actively engaging in any sign language, referred to as “idle”. The chosen signs include “me”, “hello”, “thanks”, “iloveyou”, “please”, “help”, “what”, “learn”, “sign” and “more”, amounting to a total of 11 distinct classes.

1 Sign Language -> 90 Sequences -> 2700 Frames

Ultimately, each sign language class will consist of 90 datasets, referred to as sequences. In this context, a sequence can equivalently be termed as a video, with each sequence comprising 30 frames. This yields a cumulative total of 990 sequences across the 11 classes, encompassing a grand total of 29,700 frames of data. However, the collected frames are not stored directly in the image file format. Instead, they undergo multiple processing steps to optimize efficiency in storage and utilization, as depicted in Figure 3.5.

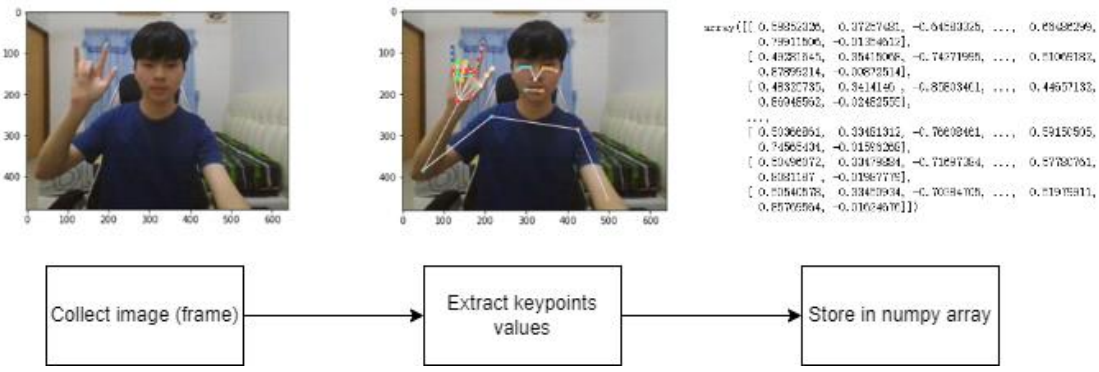
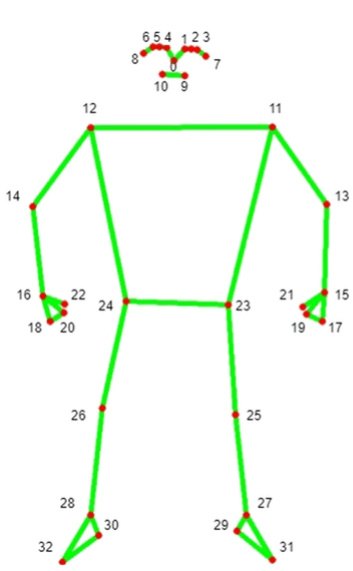


Figure 4.2 Low level data collection diagram.

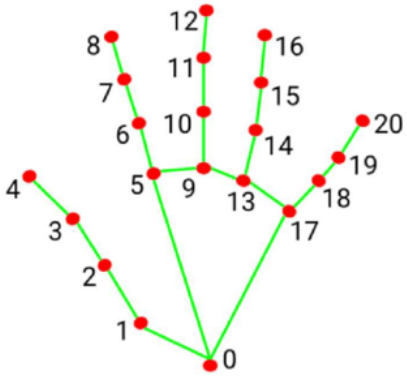
MediaPipe Holistic, developed by Google, seamlessly integrates components for detecting pose, facial, and hand landmarks, creating a comprehensive landmarking system for the entire human body. This holistic model processes image frames in real-time, accurately detecting and predicting landmarks. It outputs a total of 543 landmarks, including 33 pose landmarks, 468 face landmarks, and 21 hand landmarks per hand. To aid visualization, landmarks are depicted based on the obtained results.

Each collected image frame undergoes processing through MediaPipe Holistic to detect body pose and both left and right hands landmarks. The decision is deliberately made to exclude facial landmarks, as they are not crucial for hand gesture recognition and their inclusion, with 468 landmarks, could significantly impact the system's performance. Consequently, 75 landmarks are gathered, comprising 33 pose landmarks as illustrated in Figure 4.6, and 21 hand landmarks per hand as illustrated in Figure 4.7.



0. nose
1. left eye inner
2. left eye
3. left eye outer
4. right eye inner
5. right eye
6. right eye outer
7. left ear
8. right ear
9. mouth left
10. mouth right
11. left shoulder
12. right shoulder
13. left elbow
14. right elbow
15. left wrist
16. right wrist
17. left pinky
18. right pinky
19. left index
20. right index
21. left thumb
22. right thumb
23. left hip
24. right hip
25. left knee
26. right knee
27. left ankle
28. right ankle
29. left heel
30. right heel
31. left foot index
32. right foot index

Figure 4.3 MediaPipe pose landmarks. Adapted from [16].



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Figure 4.4 MediaPipe hand landmarks. Adapted from [16].

After the step, the subsequent action entails extracting the keypoint values from the resulting 75 landmarks. The result comprises a total of 258 keypoint from both pose and hands landmarks. Among these, the 33 pose landmarks, each represented by 4 keypoint values (coordinates: x, y, z, and visibility), contribute to 132 keypoints. Meanwhile, the 21 landmarks for each hand, represented by 3 keypoint values (coordinates: x, y, z), without visibility, contribute to 126 keypoints. This structure is organized as follows: pose (132 keypoints), left hand (126 keypoints), and right hand (126 keypoints). Figure 4.8 showcases the structure of hand landmark output result by MediaPipe Holistic.

```
Landmarks:
  Landmark #0:
    x            : 0.638852
    y            : 0.671197
    z            : -3.41E-7
  Landmark #1:
    x            : 0.634599
    y            : 0.536441
    z            : -0.06984
  ... (21 landmarks for a hand)
```

Figure 4.5 Example of hand landmark output result

1 Sign Language -> 90 Sequences -> 2700 Frames -> 696600 Keypoint Values

In the final step, the extracted keypoints are concatenated into a NumPy array and saved in a designated folder, serving as both training and testing data. A label map is then generated, with each numerical index corresponding to a specific sign language gesture. Ultimately, our dataset takes the shape (90, 30, 258), representing 90 samples, each consisting of 30 frames with 258 keypoints. Additionally, the categorical labels are structured as (90, 11), providing categorical information for the 90 samples.

4.1.2 Build and Train Model

First and foremost, the construction of the sequential deep learning model entails leveraging TensorFlow, specifically Keras, providing convenient access and customization options for the model layers. As illustrated in Figure 3.8, the sequential model will be structured with an input layer, multiple hidden layers, and an output layer, forming the neural network. This architecture design is intended to streamline the information flow within the network: the input layer receives data, the hidden layers extract features, and the output layer produces the final predictions.

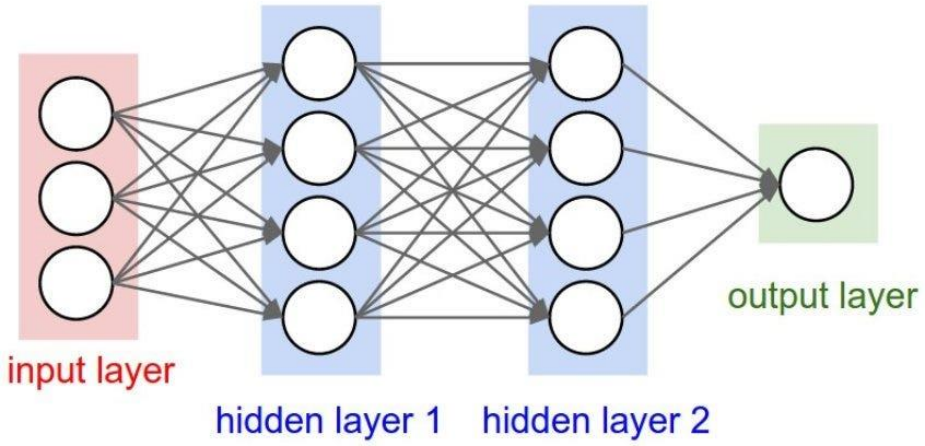


Figure 4.6 Example of neural network. Adapted from [17].

Within the architecture of this model, a total of six layers are thoughtfully integrated. These layers encompass three recurrent layers (RNN/LSTM/GRU), each designed to capture temporal dependencies and patterns within the sequential data. Additionally, three Dense layers, known for their capability to capture complex relationships in the data, are employed for further process and distil the extracted features.

At the heart of this model lies the input layer, characterized by a shape of (30, 258), which indicates the presence of 30 frames, each containing 258 keypoints representing significant aspects of the input data. The recurrent layers, each configured with distinct features sizes of 64, 128, and 64, play a crucial role in capturing the sequential nature of the data and uncovering meaningful patterns over time. Moreover, following each layer is a Rectified Linear Unit (ReLU) activation function, which introduces non-linearity into the model, enabling it to capture complex relationships and patterns present in the data. The last layer serves as the output layer, with the number of units sized to match our dataset's classes. This pivotal layer employs SoftMax activation, specifically tailored for multi-class classification tasks. Its primary function is to transform the model's raw output into probabilities assigned to each class. The comprehensive depiction of the sequential model architecture can be found in Figure 4.10.

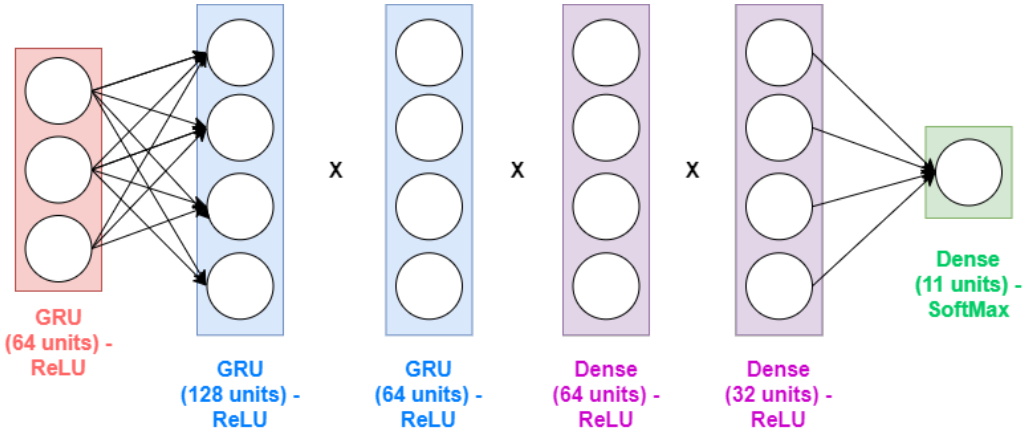


Figure 4.7 Overview of the sequential model architecture.

The Adam optimizer is employed to regulate the update of the model's weights during training, aiming to minimize the specified loss function. Adam stands out as a widely embraced and popular optimization algorithm within the domain of deep learning.

Model: "sequential"

Layer (type)	Output Shape	Param #
gru (GRU)	(None, 30, 64)	62208
gru_1 (GRU)	(None, 30, 128)	74496
gru_2 (GRU)	(None, 64)	37248
dense (Dense)	(None, 64)	4160
dense_1 (Dense)	(None, 32)	2080
dense_2 (Dense)	(None, 11)	363

Total params: 180,555
Trainable params: 180,555
Non-trainable params: 0

Figure 4.8 Summary of the sequential model.

The model is set to undergo training for 15 epochs for all three architectures (RNN/LSTM/GRU). Training and testing results are shown in system evaluation section 6.1.

4.2 Conventional Method – Transfer Learning with Pre-trained model

The conventional approach to developing hand gesture recognition typically involves leveraging transfer learning. This entails utilizing a pre-trained object detection model such as MobileNet, VGG, or ResNet, within a convolutional neural network (CNN) architecture. Complementing this, object detection algorithms like Single Shot MultiBox Detector (SSD) or You Only Look Once (YOLO) are often employed to enhance the accuracy and efficiency of gesture detection. MobileNet-SSD, a notable example, combines MobileNet with the object detection framework SSD. MobileNet functions as the feature extractor, extracting useful features from input images, while the object detection framework is responsible for predicting bounding boxes and class labels based on these features. Further details will be elaborated in the Training Model section 4.2.4.

This method follows a series of steps to achieve real-time sign language recognition. The process begins with the collection and preparation of a substantial number of image labels using Labelimg. Subsequently, a pre-trained object detection model by TensorFlow is retrained using transfer learning, specifically MobileNet-SSD. The final step involves real-time detection using OpenCV.

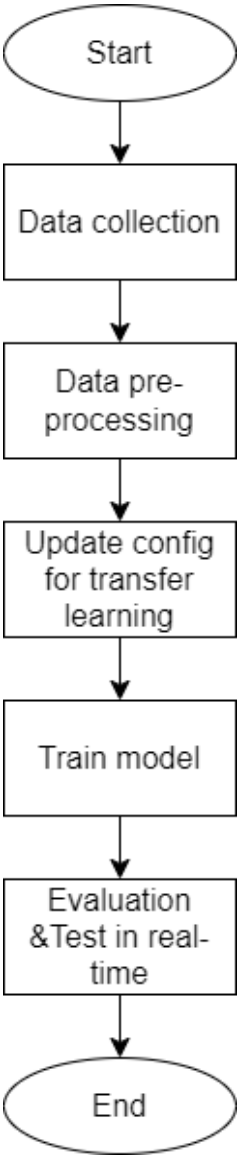


Figure 4.9 High level system flowchart.

4.2.1 Project Setup

Before initialling the process, the system setup is crucial. Firstly, the TensorFlow Object Detection API is installed by cloning the TensorFlow Models repository through Git. Next, the installation and compilation of Protocol Buffers (protobuf) are undertaken. Protobuf is utilized by the TensorFlow Object Detection API to configure model and training parameters. Downloading and compiling protobuf libraries is a prerequisite for using the framework.

4.2.2 Data Collection

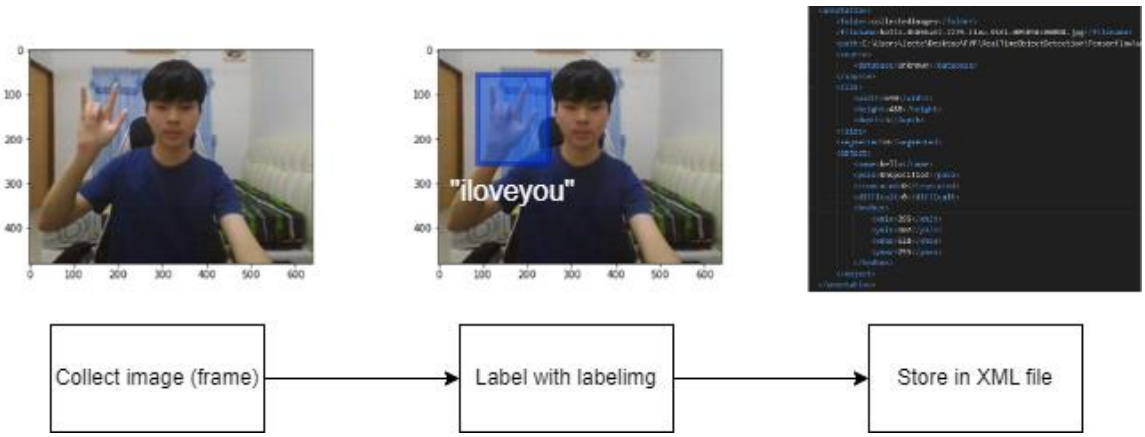


Figure 4.10 High level data collection flow chart.

At the outset, we commenced by collecting images to serve as the dataset for training the model. For the method, 3 sign languages with 90 samples each have been selected. Additionally, 3 American Sign Language (ASL) signs have been chosen, including “hello”, “thanks”, “iloveyou”. Additionally, a default sign “idle” represents when the user is not actively engaging in any sign language. In total, there are 4 classes in this dataset.

LabelImg is used to label the collected images, annotating the gestures within each image with the corresponding sign language action name. LabelImg is a lightweight and user-friendly image annotation tool designed for labelling object bounding boxes in images.

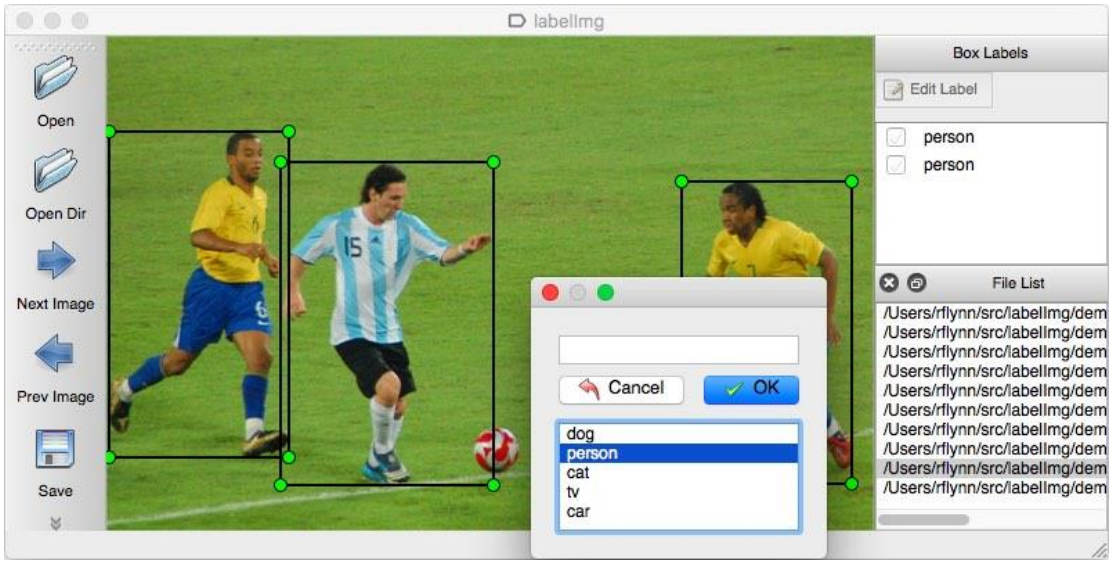
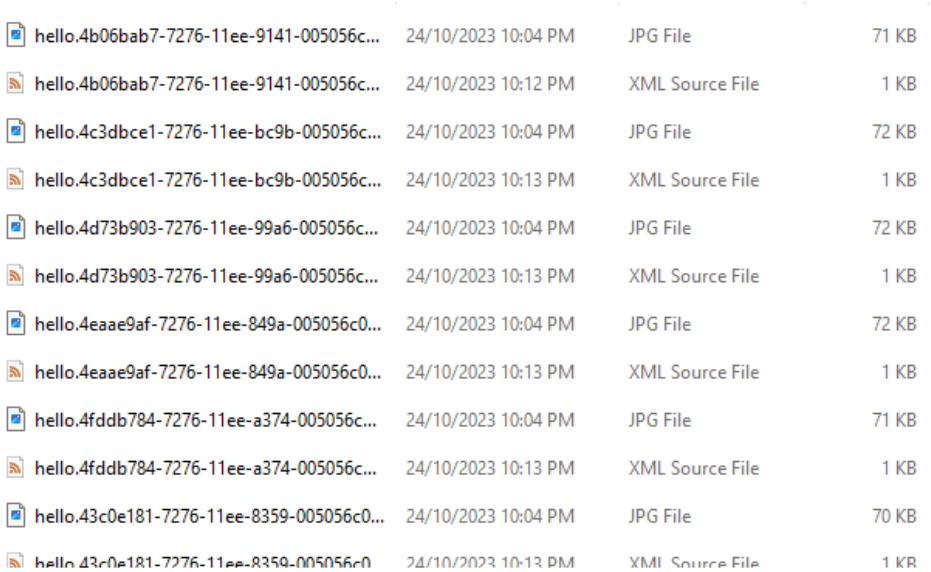


Figure 4.11 Example of LabelImg. Adapted from [18].

Each image is labelled individually, and for each image, a corresponding XML file is generated. This XML file stores the label values needed for training the model, such as the size of the image (width, height, depth), the annotation name, and the position of the bounding box (x and y) inside the image. The stored values will later be used to guide the model on the location of the sign language gestures within the images during the training process, effectively presenting the object for recognition.



hello.4b06bab7-7276-11ee-9141-005056c...	24/10/2023 10:04 PM	JPG File	71 KB
hello.4b06bab7-7276-11ee-9141-005056c...	24/10/2023 10:12 PM	XML Source File	1 KB
hello.4c3dbce1-7276-11ee-bc9b-005056c...	24/10/2023 10:04 PM	JPG File	72 KB
hello.4c3dbce1-7276-11ee-bc9b-005056c...	24/10/2023 10:13 PM	XML Source File	1 KB
hello.4d73b903-7276-11ee-99a6-005056c...	24/10/2023 10:04 PM	JPG File	72 KB
hello.4d73b903-7276-11ee-99a6-005056c...	24/10/2023 10:13 PM	XML Source File	1 KB
hello.4eaae9af-7276-11ee-849a-005056c0...	24/10/2023 10:04 PM	JPG File	72 KB
hello.4eaae9af-7276-11ee-849a-005056c0...	24/10/2023 10:13 PM	XML Source File	1 KB
hello.4fddb784-7276-11ee-a374-005056c...	24/10/2023 10:04 PM	JPG File	71 KB
hello.4fddb784-7276-11ee-a374-005056c...	24/10/2023 10:13 PM	XML Source File	1 KB
hello.43c0e181-7276-11ee-8359-005056c0...	24/10/2023 10:04 PM	JPG File	70 KB
hello.43c0e181-7276-11ee-8359-005056c0	24/10/2023 10:13 PM	XML Source File	1 KB

Figure 4.12 View of XML files generated.

```
<annotation>
    <folder>collectedimages</folder>
    <filename>hello.4b06bab7-7276-11ee-9141-005056c00008.jpg</filename>
    <path>C:\Users\leete\Desktop\FYP\RealTimeObjectDetection\Tensorflow\
    <source>
        <database>Unknown</database>
    </source>
    <size>
        <width>640</width>
        <height>480</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>hello</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>395</xmin>
            <ymin>102</ymin>
            <xmax>528</xmax>
            <ymax>295</ymax>
        </bndbox>
    </object>
</annotation>
```

Figure 4.13 Example of the XML file.

4.2.3 Data Pre-processing

After data collection, the next step involves preprocessing before utilizing it to train our model. This entails creating TFRecords to store both image and annotation information. TFRecords are a binary format ideal for effectively encoding long sequences of TensorFlow data, making them easily loadable within TensorFlow. Following this, the collected dataset is portioned into training and testing sets, with a test size of 0.2, ensuring robust evaluation of model performance.

4.2.4 Modify and Train the pre-trained model

Before initiating the training process, it crucial to customize the existing object detection model MobileNet-SSD to suit our project's specific requirements and dataset. This customization is typically accomplished through the pipeline configuration file, where various settings can be adjusted. The primary consideration includes updating the number of classes to reflect the categories in our dataset which is 11, configuring the batch size to 4 (which determines the amount of data processed within each training epoch), setting up model checkpoints to save the model's progress during training, and ensuring accurate file paths for essential files such as label_map and TFRecord containing our annotated data.

We employ a transfer learning technique by utilizing a pre-trained object detection model called MobileNet-SSD which has been trained on a large dataset for object detection tasks, setting a checkpoint on the model to continue retraining from its previous knowledge and fine-tuning it on a specific dataset related to our desired application. This technique is known as Transfer Learning. Transfer Learning involves the reuse of a pre-trained model on a new problem, where a machine leverages the knowledge gained from previous task to enhance generalization on another. This approach can expedite the learning process, improve accuracy, and potentially require less training data.

```
  sync_replicas: true
  optimizer {
    momentum_optimizer {
      learning_rate {
        cosine_decay_learning_rate {
          learning_rate_base: 0.08
          total_steps: 50000
          warmup_learning_rate: 0.026666
          warmup_steps: 1000
        }
      }
      momentum_optimizer_value: 0.9
    }
    use_moving_average: false
  }
  fine_tune_checkpoint: "Tensorflow/workspace/pre-trained-models/ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8/checkpoint/ckpt-0"
  num_steps: 50000
  startup_delay_steps: 0.0
  replicas_to_aggregate: 8
  max_number_of_boxes: 100
  unpad_groundtruth_tensors: false
  fine_tune_checkpoint_type: "detection"
  fine_tune_checkpoint_version: V2
}
train_input_reader {
  label_map_path: "Tensorflow/workspace/annotations/label_map.pbtxt"
  tf_record_input_reader {
    input_path: "Tensorflow/workspace/annotations/train.record"
  }
}
eval_config {
  metrics_set: "coco_detection_metrics"
  use_moving_averages: false
}
eval_input_reader {
  label_map_path: "Tensorflow/workspace/annotations/label_map.pbtxt"
  shuffle: false
  num_epochs: 1
  tf_record_input_reader {
    input_path: "Tensorflow/workspace/annotations/test.record"
  }
}
```

Figure 4.14 Example of pipeline.config file.

The rationale behind employing the MobileNet-SDD combination lies in the unique strengths of MobileNet and SSD. MobileNet and SSD are two distinct components commonly utilized in computer vision tasks, including object detection and image recognition.

A. MobileNet

MobileNet is a lightweight CNN architecture designed for mobile and embedded vision applications, renowned for its computational efficiency without sacrificing accuracy. It achieves this through depth-wise separable convolutions, making it ideal for resource constrained devices.

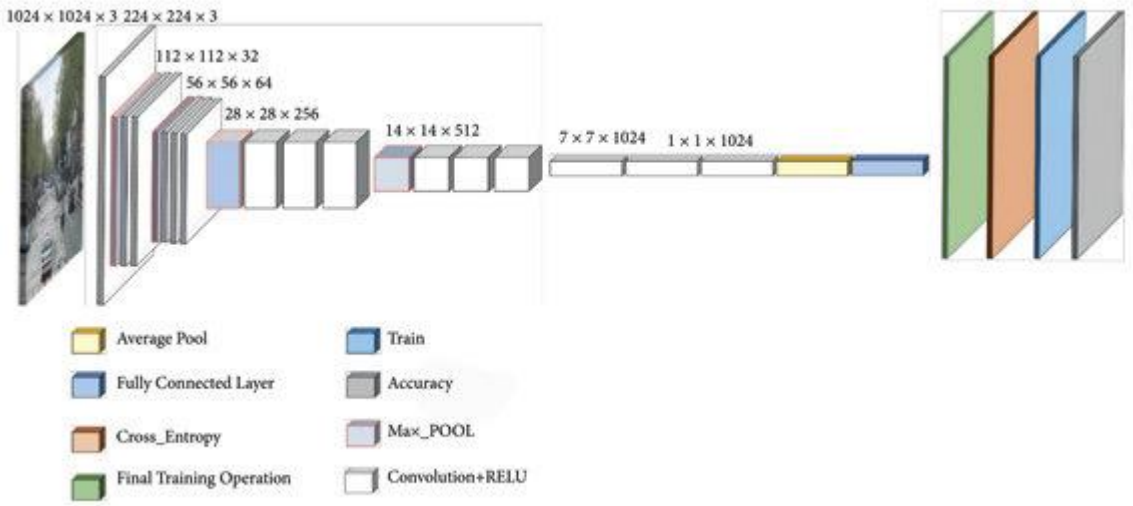


Figure 4.15 MobileNet architecture. Adapted from [19].

B. Single Shot MultiBox Detector (SSD)

On the other hand, SSD is unified object detection algorithm that combines localization and classification tasks within a single neural network. It operates simultaneously across multiple scales in the input image, resulting in faster inference times and improved efficiency.

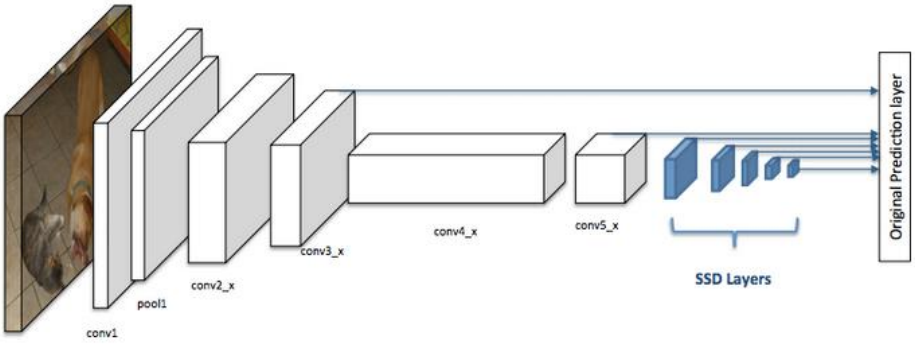


Figure 4.16 Architecture of Single-shot detector (SSD). Adapted from [20].

C. MobileNet-SSD

In the MobileNet-SSD combination, MobileNet serves as the backbone for feature extraction, while SSD handles the object detection tasks. Achieving a good balance between speed and accuracy, making it well suited for a real-time object

detection task on resource-constrained devices like mobile phones or embedded systems.

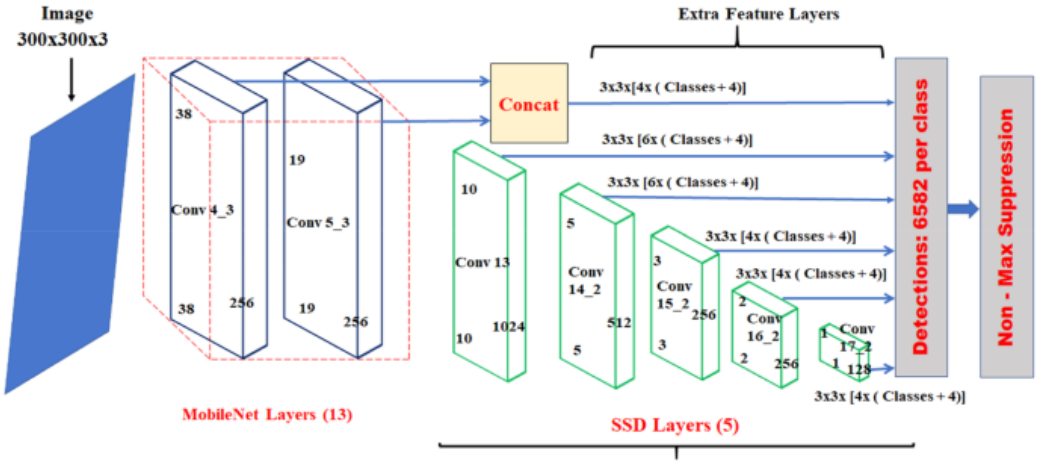


Figure 4.17 Architecture of MobileNet SSD. Adapted from [21].

Now comes the training model phase, we simply execute the train_ssd python script provided by TensorFlow. This script automates the entire training process using our configured pipeline, and a specific number of training steps is set, typically around 10,000.

CHAPTER 5

System Implementation

5.1 Mobile Application Development

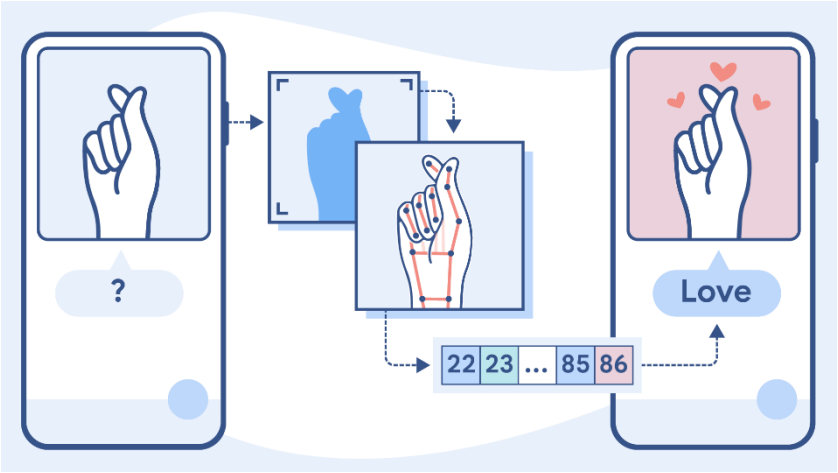


Figure 5.1 Illustration of mobile application idea. Adapted from [22].

We have developed a mobile application for utilizing our trained deep learning model for real-world applications. Android Studio was chosen as the development platform for its robust features and compatibility.

Prior to deploying a deep learning model in a mobile application, it's necessary to convert the trained model into TensorFlow Lite format. TensorFlow Lite is a streamlined variant of TensorFlow, tailored for mobile and embedded devices, offering optimized performance and efficiency.

As illustrated in Figure 5.2, the mobile application is designed to leverage the capabilities of a smartphone camera to capture real-time live video frames. These frames are then processed through pose landmark detection and hand landmark detection for both left and right hands. The keypoint values will be collected from the pose and both hands landmarks. These collected keypoints are then passed into our deep learning model, which has been trained specifically for gesture recognition as detailed in section 4. Finally, the result is returned by our deep learning model to the mobile app, where it is displayed to the user in real-time.

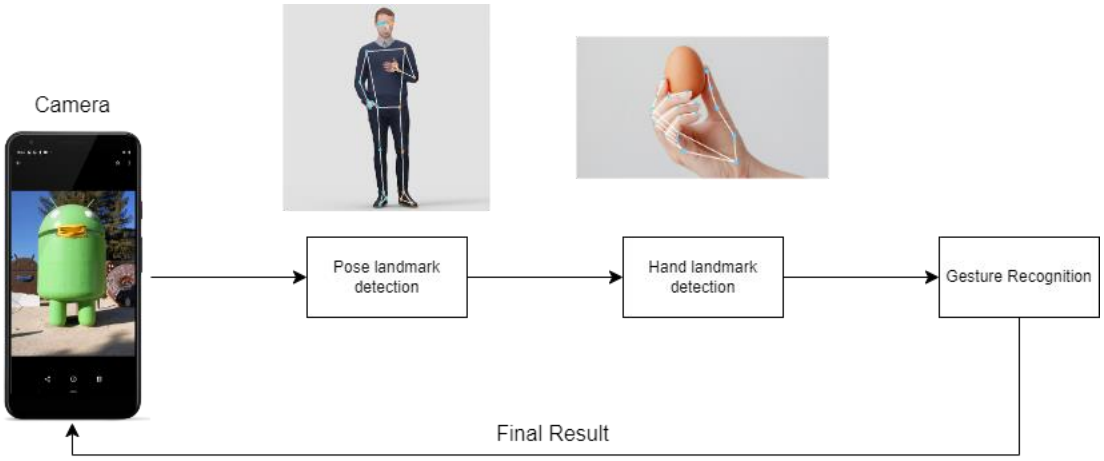


Figure 5.2 Overview of mobile application architecture.

Figure 5.3 Preview of mobile application UI.

Figure 5.3 displays the user interface of the mobile application.

5.2 Implementation Issues and Challenges

In our previous development of the deep learning model, we utilized MediaPipe Holistic for simultaneous detection of pose and hand landmarks. However, as MediaPipe Holistic is yet unavailable on the Android platform, we devised a solution by leveraging separate MediaPipe components for pose landmark and hand landmark detection.

Significant issues exist in the implementation, primarily due to the heavy processing involved in passing through three different detection and recognize models. This can result in noticeable latency, impacting the smooth operation of the mobile application. Additionally, this complexity can lead to incorrect predictions and failure to recognize gestures accurately.

CHAPTER 6

System Evaluation and Discussion

6.1 Model Performance Metrics

6.1.1 Novel Method: Training Sequential model assisted with MediaPipe

The model evaluation entails utilizing identical training and testing datasets with a uniform partitioning 0.2, comprising 11 classes, each containing 90 samples. Performance assessment revolves around key metrics including accuracy, loss, precision, and recall, all averaged across the 11 classes. Consistency is maintained with 15 epochs for each architecture. Optimal values are accentuated in bold.

- Accuracy: Measures how often the model's predictions are correct.
- Loss: Represents the model's error or discrepancy between predicted and actual outputs; lower is better.
- Precision: Reflects the accuracy of positive predictions made by the model
- Recall: Indicates the model's ability to identify all actual positive instances

Table 6.1 Model 1 Evaluation: Training dataset

Architectures	Avg. Accuracy	Loss	Avg. Precision	Avg. Recall
RNN	0.9457	0.1705	0.9537	0.9443
LSTM	0.9659	0.2115	0.9686	0.9667
GRU	**0.9811**	**0.1632**	**0.9828**	**0.9805**

Table 6.2 Model 1 Evaluation: Testing dataset

Architectures	Avg. Accuracy	Loss	Avg. Precision	Avg. Recall
RNN	0.9293	0.1705	0.9470	0.9336
LSTM	0.9393	0.2115	0.9451	0.9382
GRU	**0.9747**	**0.1632**	**0.9788**	**0.9784**

From the results in Table 6.1, it evident that the GRU outperforms both the RNN and LSTM architectures in terms of performance metrics such as accuracy, loss, precision, and recall.

6.1.2 Conventional Method: Transfer learning with pre-trained model

Table 6.3 Model 2 Evaluation: Testing dataset

	Loss	Avg. Precision	Avg. Recall
MobileNet-SSD	0.7123	0.788	0.8

6.2 Model Testing in Real-time

6.2.1 Novel Method: Training Sequential model assisted with MediaPipe

In the testing phase, the model is employed in real-time using computer vision with OpenCV to detect and recognize sign languages. To enhance visualization, a straightforward user interface (UI) is created, as depicted in Figure 4.1. This UI displays the probability of detecting sign language on the left side and the prediction on the top. The threshold is adjusted to optimize performance.

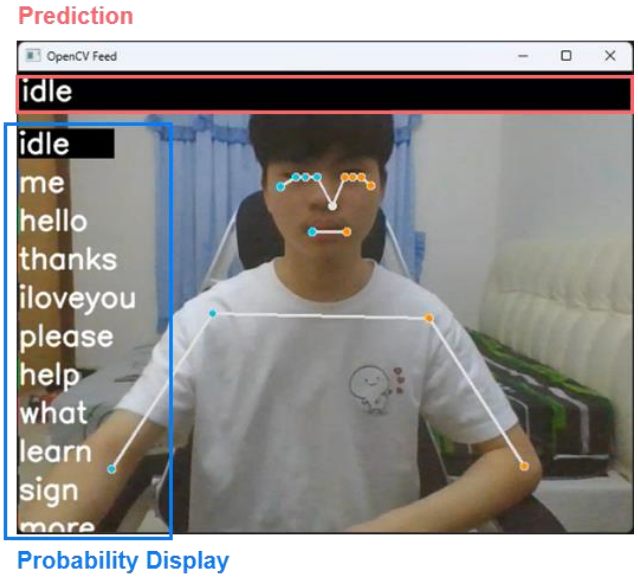


Figure 6.1 Real time testing user interface.

The system's functionality involves capturing a sequence every 30 frames from the camera, using the sequence to make predictions with the trained model, and presenting the results on the UI. To prevent repetitive displays of the same actions, a simple algorithm is implemented. This ensures a smoother and more efficient operation of the system during real-time testing.

Here are the examples of 11 chosen sign languages detected in real-time:

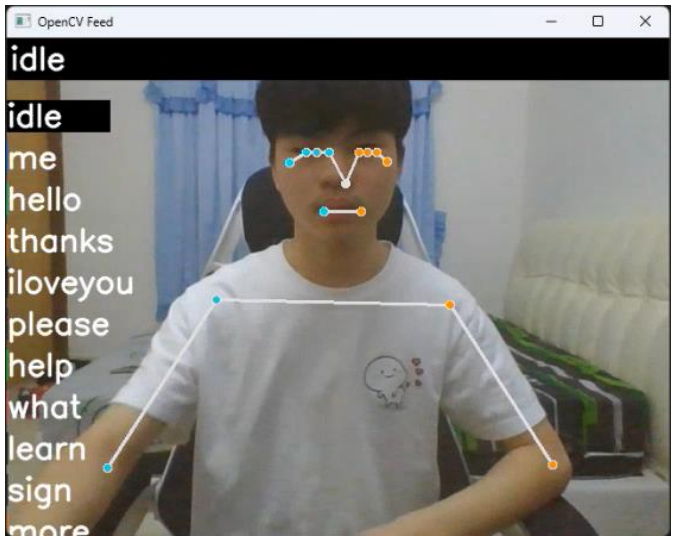


Figure 6.2 Example of "idle".

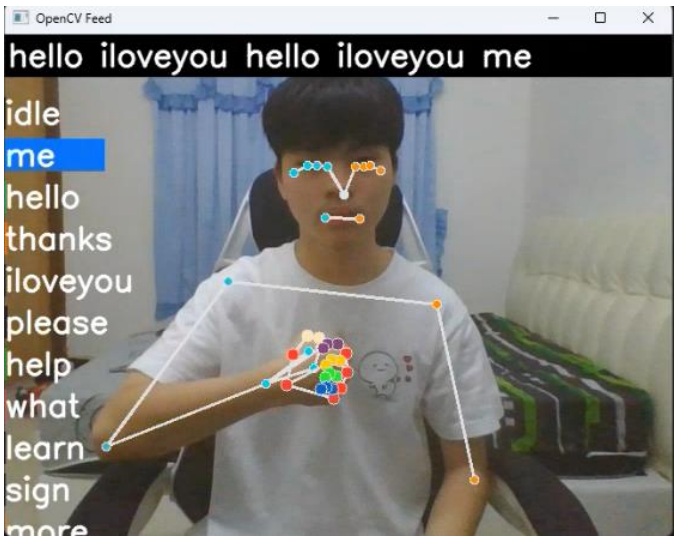


Figure 6.3 Example of sign language "me".

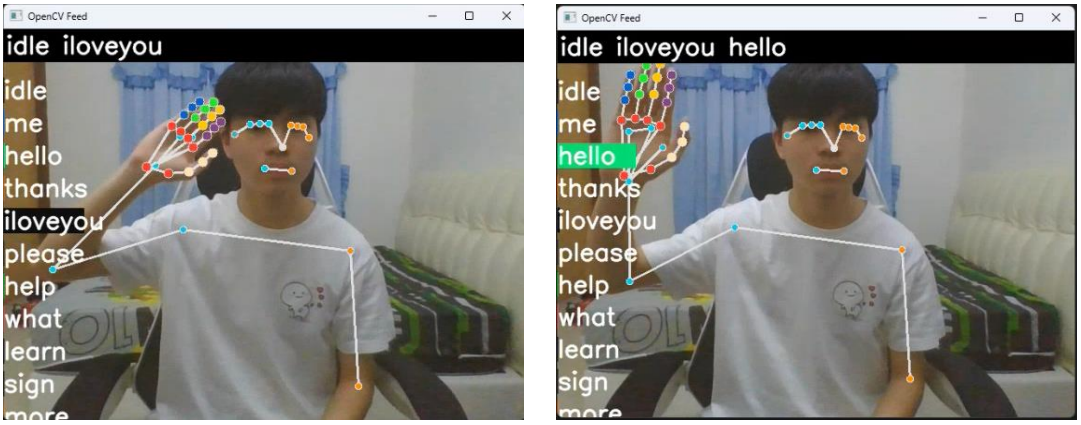


Figure 6.4 Example of sign language “hello”.

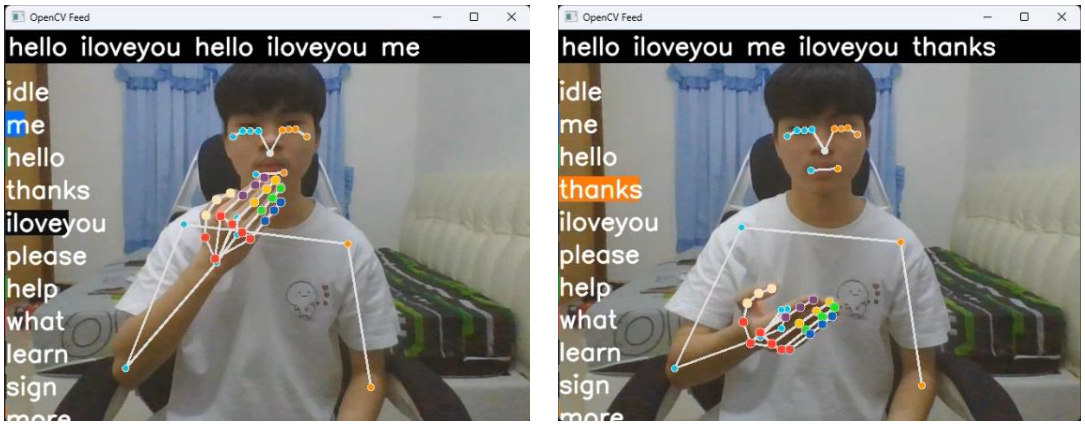


Figure 6.5 Example of sign language “thanks”.

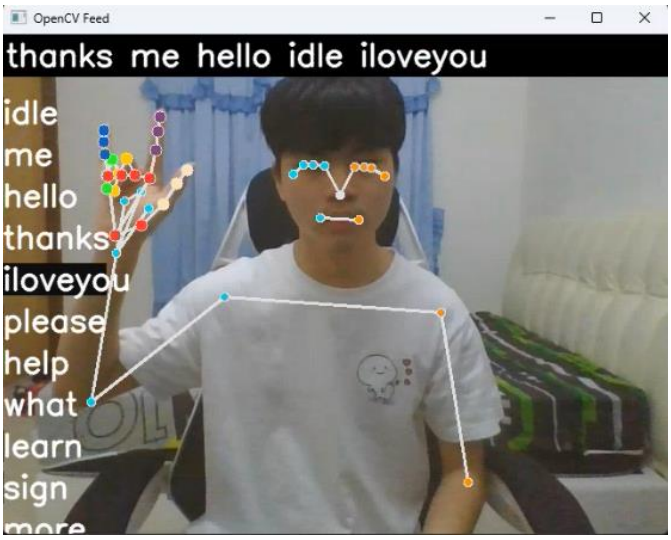


Figure 6.6 Example of sign language “iloveyou”.

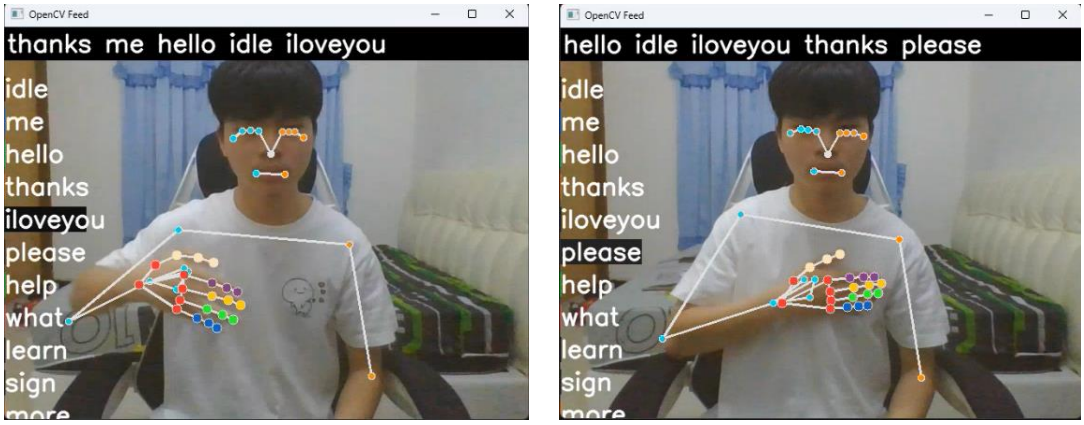


Figure 6.7 Example of sign language “please”.

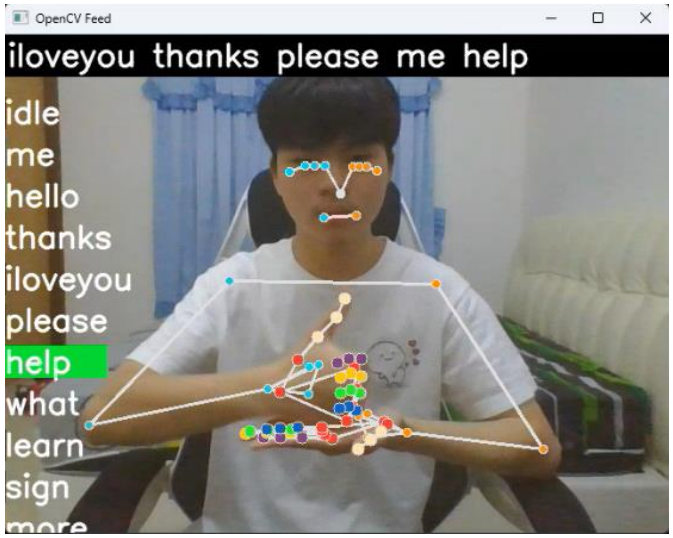


Figure 6.8 Example of sign language “help”.

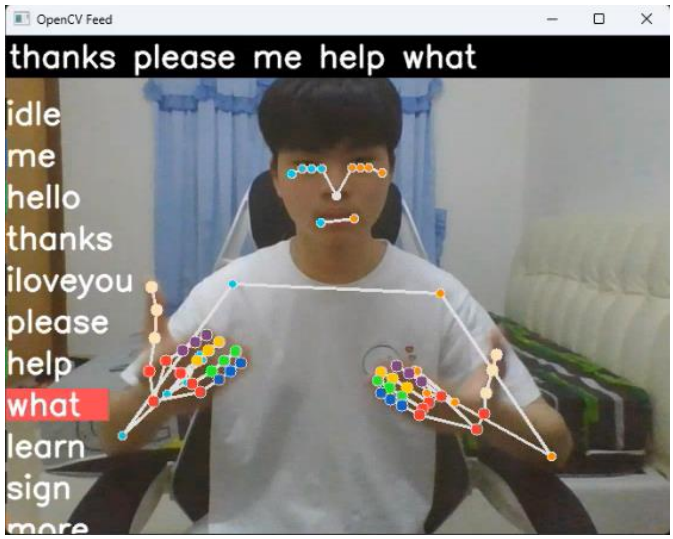


Figure 6.9 Example of sign language “what”.

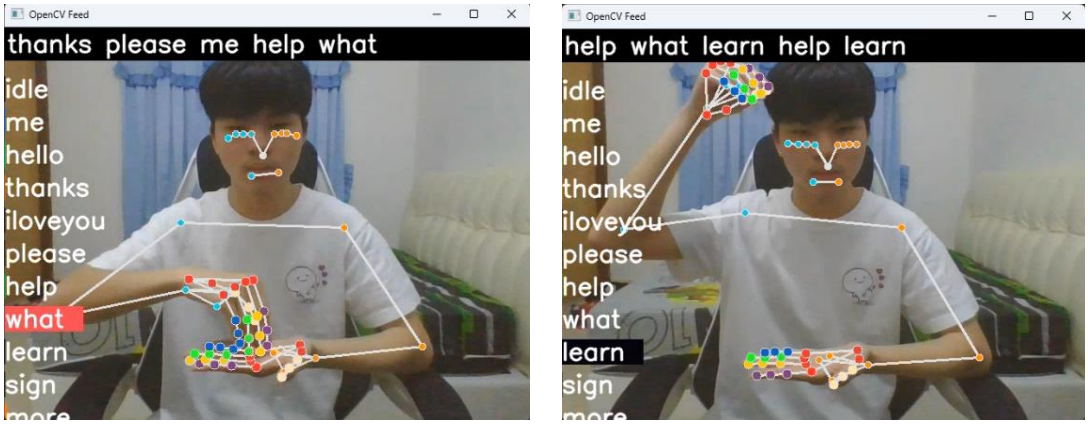


Figure 6.10 Example of sign language “learn”.

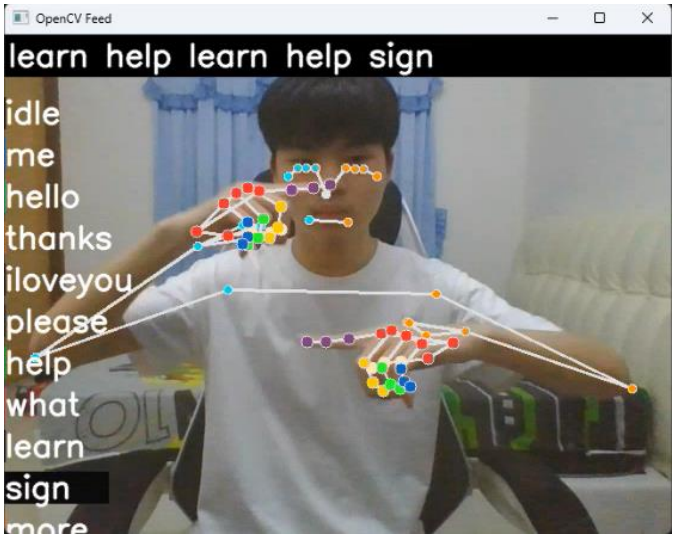


Figure 6.11 Example of sign language “sign”.

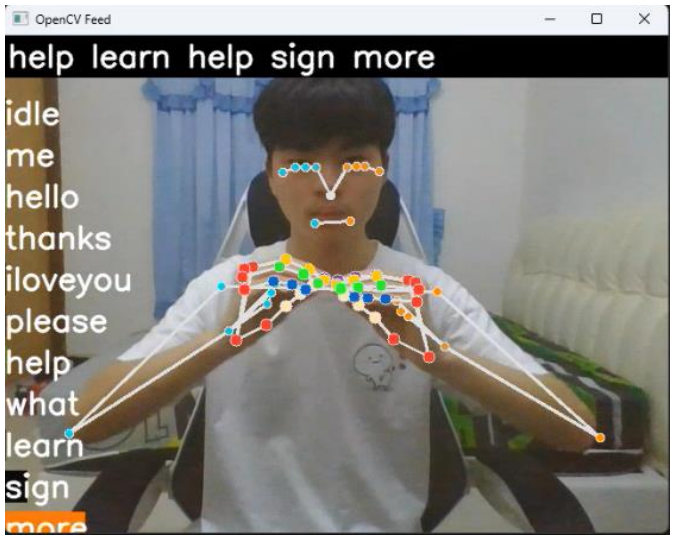


Figure 6.12 Example of sign language “more”.

As a result, it remarkable that the GRU architecture not only demonstrates superior performance in the evaluation metrics but also excels in real-time testing, showing its effectiveness in quickly and accurately recognizing sign language gesture. This underscores the practical applicability and efficiency of the GRU model in real-world scenarios.

6.2.2 Conventional Method: Transfer learning with pre-trained model

Despite the conventional method displaying seemingly high accuracy in performance metrics, real-time testing revels disappointingly low accuracy levels. This discrepancy can be attributed to the limited training data and short training time. During real-time testing, the model only detects actions in a clean, background noise-free environment. However, it struggles in real-world scenarios, frequently misclassifying objects in the camera's surroundings.

6.3 Discussion

Based on the evaluation results of both methods, it evidence that novel approach surpasses the conventional one. In fact, the novel method has been found to better align with our project objectives, enhancing time efficiency, data requirements, and overall performance, thus making it the preferred choice over the conventional method.

The novel method, powered by MediaPipe, not only outperforms but also streamlines the development process by significantly reducing time and cost. Unlike the conventional approach, which requires laborious image labelling for annotation, with MediaPipe eliminates this step, storing all necessary data in numerical values. This not only saves time and frustration but also slashes storage requirements, making it a more efficient and cost-effective solution overall.

In the training process, the conventional method proves to be considerably time-consuming, demanding a significantly longer duration compared to the novel method. Moreover, training the model using the conventional approach necessitates a substantial

volume of training data, unlike the novel method, which operates efficiently with a smaller dataset. Given our project's objective of developing a hand gesture recognition system with high accuracy using minimal data in a short timeframe, leveraging MediaPipe facilitates achieving this goal efficiently.

6.4 Testing on Mobile Application

At last, the deep learning model has been integrated into the mobile application, allowing for real-world testing of hand gesture recognition using the smartphone camera. Below are some examples showcasing the operation of the mobile application:

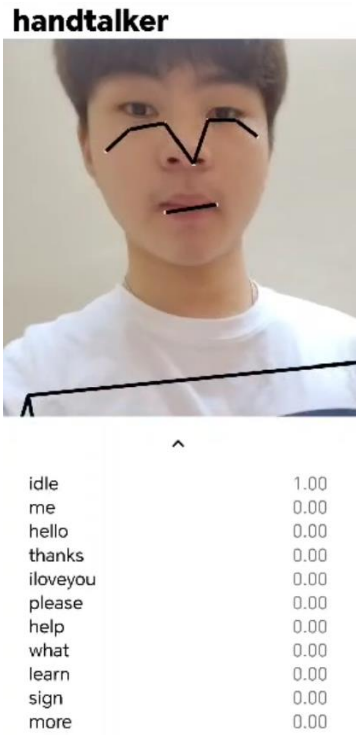


Figure 6.13 Mobile app recognized "idle".

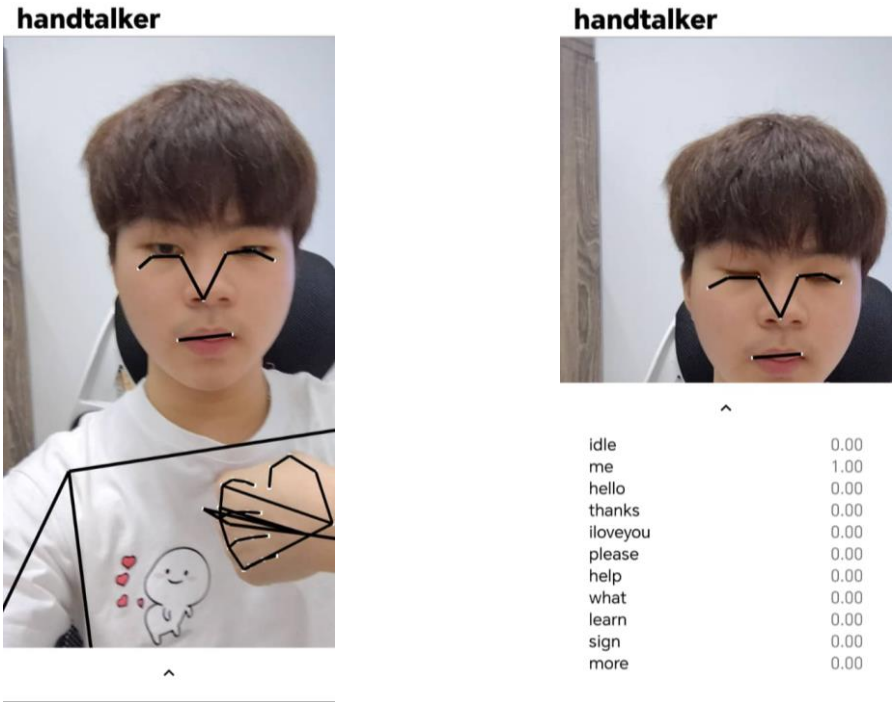


Figure 6.14 Mobile app recognized “me”.

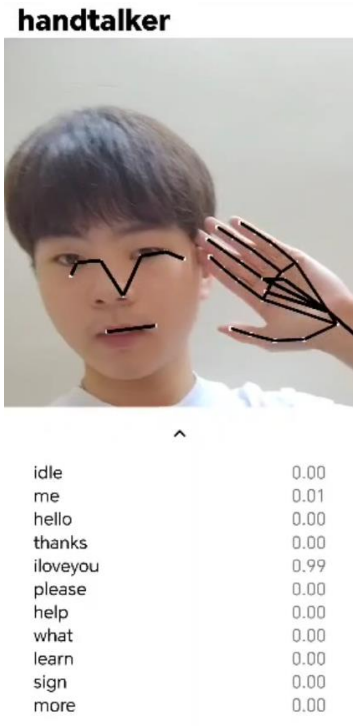


Figure 6.15 Mobile app misinterpreted “hello” as “iloveyou”.

Figure 6.16 Mobile app recognized “thanks”.

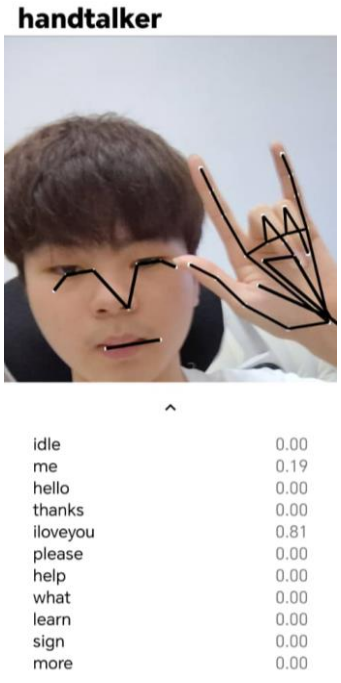


Figure 6.17 Mobile app recognized “iloveyou”.

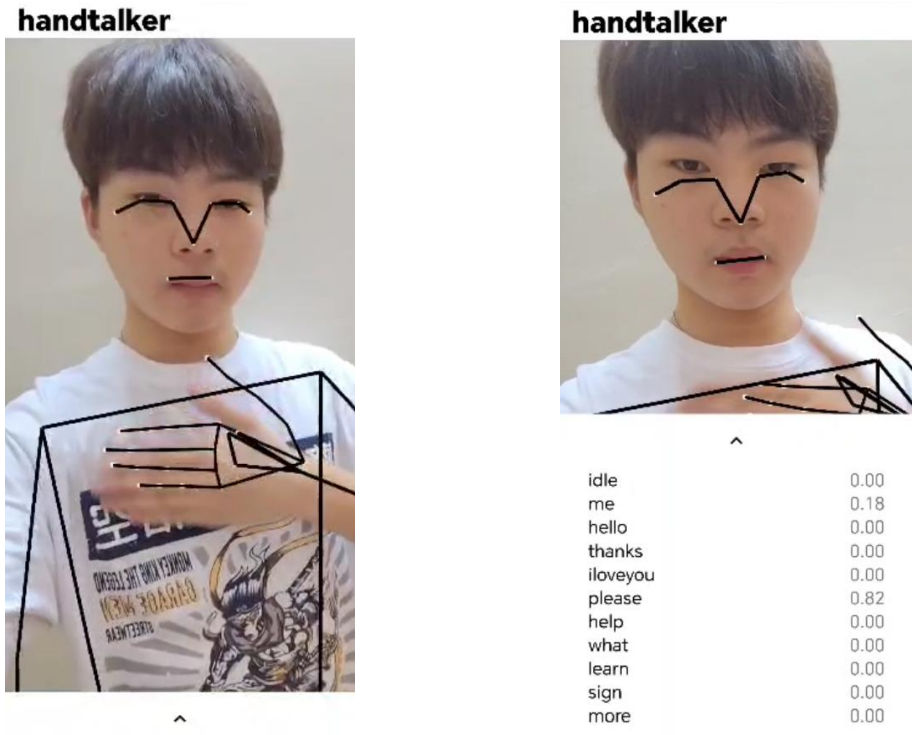


Figure 6.18 Mobile app recognized “please”.

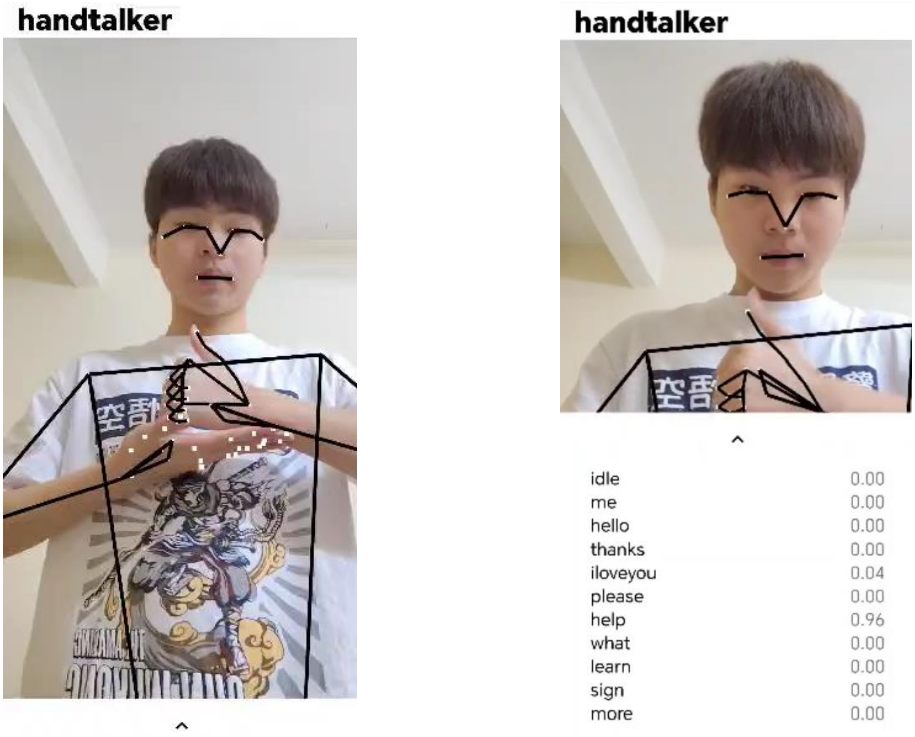


Figure 6.19 Mobile app recognized “help”.

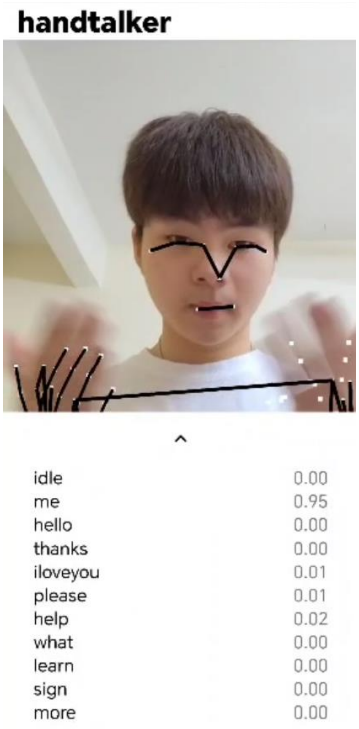


Figure 6.20 Mobile app failed to recognize “what”.

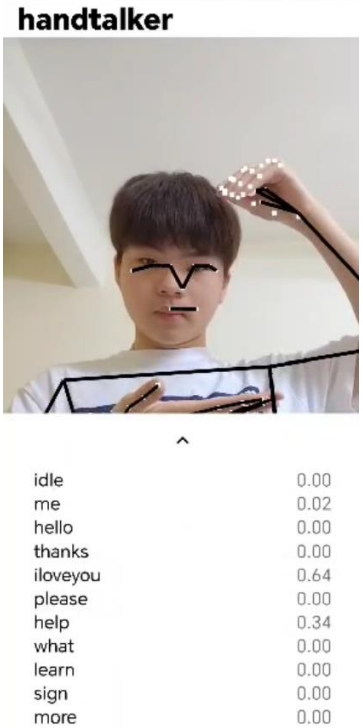


Figure 6.21 Mobile app failed to recognize “learn”.

Figure 6.22 Mobile app failed to recognize “sign”.

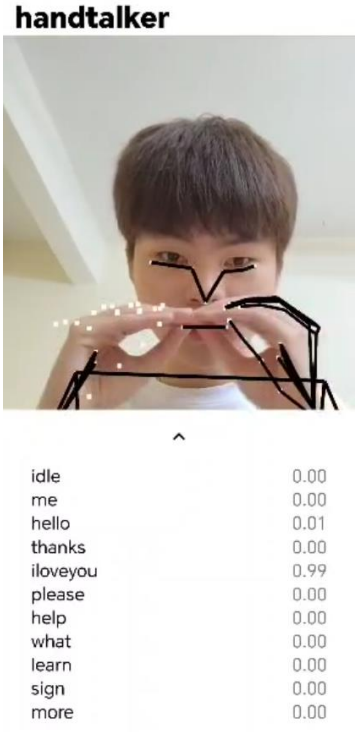


Figure 6.23 Mobile app failed to recognize “more”.

6.5 Project Challenges

When it comes to implementing in mobile applications, we encounter a significant challenge: certain gestures often fail to be recognized. For example, gestures like “hello” and particularly complex two-handed gestures such as “what”, “learn”, “sign” and “more” present notable difficulties. The failure of the “hello” gesture may be attributed to its similarity with another gesture, “iloveyou”. When a user performs the “hello” gesture, it may inadvertently be recognized as “iloveyou”. This issue likely stems from a limited training dataset. To improve real-time accuracy, additional samples may be necessary.

6.6 Objectives Evaluation

Our project unequivocally fulfils its objectives. Firstly, the goal of training a deep learning model capable of accurately recognizing hand gestures in real-time has been achieved with remarkable success, achieving a very high level of accuracy. Secondly, the objectives of developing an augmented reality (AR) mobile application have also been realized. A mobile application has been created to leverage smartphone cameras, enabling users to sketch and display hand and pose landmarks. Ultimately, the integration of both the deep learning model and the mobile application seamlessly captures user hand gestures and effectively recognizes sign language.

6.7 Concluding Remark

After rigorous testing, it became evident that the novel approach, utilizing a sequential model trained with assistance from MediaPipe, outperformed conventional methods in terms of both model accuracy and real-time testing. Crucially, the hand gesture recognizer model seamlessly integrated into the mobile application and underwent successful testing.

CHAPTER 7

Conclusion and Recommendation

7.1 Conclusion

In conclusion, the project has effectively fulfilled its objectives of training a sequential deep learning model capable of recognizing hand gestures and developing a mobile application to seamlessly integrate with this model. By enabling real-time detection and recognition of American Sign Language (ASL) gestures, the mobile application has laid a solid groundwork for tackling communication barriers. This accomplishment marks a significant step forward in bridging gaps and fostering inclusivity in communication. Moving forward, this project sets a promising precedent for further advancements in leveraging technology to enhance accessibility and communication for all.

7.2 Recommendation

While the current progress is commendable, there are opportunities for enhancement. Expanding the dataset to encompass a wider range of complex scenarios, diverse hand positions, and varied frame speeds is imperative. Additionally, augmenting the gesture repertoire beyond the existing 10 American Sign Language (ASL) gestures can enrich the model's capabilities. Continuous refinement through retraining and fine-tuning holds promises for optimizing performance further. Looking ahead, leveraging MediaPipe Holistic to consolidate pose and hand landmark detection can streamline operations, potentially reducing latency and improving overall efficiency.

REFERENCES

[1] “What is deep learning?: How it works, techniques & applications,” How It Works, Techniques & Applications - MATLAB & Simulink, https://www.mathworks.com/discovery/deep-learning.html (accessed Aug. 27, 2023).

[2] “A vision for making deep learning simple,” KDnuggets, https://www.kdnuggets.com/2017/09/databricks-vision-making-deep-learning-simple.html (accessed Aug. 27, 2023).

[3] “Computer vision: Understanding its meaning, examples, and applications,” Master Computer Vision Courses Online With Augmented Startups, https://www.augmentedstartups.com/blog/computer-vision-understanding-its-meaning-examples-and-applications (accessed Aug. 27, 2023).

[4] “Deafness and hearing loss,” World Health Organization, https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss (accessed Aug. 27, 2023).

[5] S. Reifinger, F. Wallhoff, M. Ablassmeier, T. Poitschke, and G. Rigoll, “Static and dynamic hand-gesture recognition for augmented reality applications,” *Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments*, pp. 728–737, 2007. doi:10.1007/978-3-540-73110-8_79

[6] S. S. Rautaray, “Real time hand gesture recognition system for dynamic applications,” *International Journal of UbiComp*, vol. 3, no. 1, pp. 21–31, 2012. doi:10.5121/iju.2012.3103

[7] G. S. Behera, “Face detection with Haar Cascade,” Medium, https://towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd08 (accessed Aug. 28, 2023).

[8] N.-U.-N. Soogund and M. H. Joseph, “Signar: A sign language translator application with augmented reality using text and image recognition,” *2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*, 2019. doi:10.1109/incos45849.2019.8951322

[9] K. Bantupalli and Y. Xie, “American Sign Language Recognition using Deep Learning and Computer Vision,” *2018 IEEE International Conference on Big Data (Big Data)*, 2018. doi:10.1109/bigdata.2018.8622141

[10] R. Radkowski, “Interactive Hand Gesture-based Assembly for Augmented Reality Applications,” *The Fifth International Conference on Advances in Computer-Human Interactions*, 2012. Accessed: Aug. 30, 2023.

[11] A Comparative Study on Human Action Recognition Using Multiple Skeletal Features and Multiclass Support Vector Machine - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Microsoft-Kinect-camera_fig1_326266889 [accessed 28 Aug, 2023]

[12] R. Rastgoo, K. Kiani, and S. Escalera, "Sign language recognition: A deep survey," *Expert Systems with Applications*, vol. 164, p. 113794, 2021. doi:10.1016/j.eswa.2020.113794

[13] E. Forson, "Understanding SSD multibox - real-time object detection in deep learning," Medium, https://towardsdatascience.com/understanding-ssd-multibox-real-time-object-detection-in-deep-learning-495ef744fab (accessed Aug. 30, 2023).

[14] R. Gour, "The Essential Guide to Learn Tensorflow Mobile and Tensorflow Lite," Medium, https://towardsdatascience.com/the-essential-guide-to-learn-tensorflow-mobile-and-tensorflow-lite-a70591687800#:~:text=TensorFlow%20supports%20a%20set%20of,open%20source%20platform%20serialization%20library. (accessed Aug. 30, 2023).

[15] "Build from source : tensorflow," TensorFlow, https://www.tensorflow.org/install/source (accessed Dec. 3, 2023).

[16] "MediaPipe | google for developers," Google, https://developers.google.com/mediapipe/solutions (accessed Dec. 3, 2023).

[17] J. Allibhai, "Building a deep learning model using Keras," Medium, https://towardsdatascience.com/building-a-deep-learning-model-using-keras-1548ca149d37 (accessed Dec. 3, 2023).

[18] "LabelImg," PyPI, https://pypi.org/project/labelImg/ (accessed Apr. 22, 2024).

[19] K;, K.K.S. (no date) *Efficient approach towards detection and identification of copy move and image splicing forgeries using mask R-CNN with MobileNet V1, Computational intelligence and neuroscience*. Available at: https://pubmed.ncbi.nlm.nih.gov/35035463/ (Accessed: 22 April 2024).

[20] *How single-shot detector (SSD) works?* (no date) *ArcGIS API for Python*. Available at: https://developers.arcgis.com/python/guide/how-ssd-works/ (Accessed: 22 April 2024).

[21] Murthy, C.B., Hashmi, M.F. and Keskar, A.G. (2021) *Optimized MobileNet + SSD: A real-time pedestrian detection on a low-end edge device - International Journal of Multimedia Information Retrieval, SpringerLink*. Available at: https://link.springer.com/article/10.1007/s13735-021-00212-7 (Accessed: 22 April 2024).

[22] "MediaPipe | google for developers," Google, https://developers.google.com/mediapipe/solutions (accessed Apr. 22, 2024).

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	**Study week no.: 4**
Student Name & ID: Lee Teck Junn, 2002030	
Supervisor: Vikneswary a/p Jayapal	
Project Title: Automated Hand Gesture Recognition for Enhancing Sign Language Communication	

1. WORK DONE

The trained deep learning model has been transformed into TensorFlow Lite model to be utilized in the project. Subsequently, in the second phase involving mobile app development, the decision was made to switch from Unity to Android Studio for app creation. Presently, only 5% progress has been made, primarily focusing on setting up the Android Studio environment.

2. WORK TO BE DONE

The upcoming focus will be entirely on mobile app development, encompassing both frontend and backend aspects. It is anticipated that by week 6, more than half of the work will be completed.

3. PROBLEMS ENCOUNTERED

No issues have been encountered so far.

4. SELF EVALUATION OF THE PROGRESS

Progress is positive, with initial setup completed and focus shifting to mobile app development. Aim to exceed 50% completion by week6.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	**Study week no.: 6**
Student Name & ID: Lee Teck Junn, 2002030	
Supervisor: Vikneswary a/p Jayapal	
Project Title: Automated Hand Gesture Recognition for Enhancing Sign Language Communication	

1. WORK DONE

The mobile app has been developed, including a basic interface and setup of the MediaPipe model. Users can now use the mobile app, which utilizes the device's camera to detect hand movements.

2. WORK TO BE DONE

The sole remaining task is the integration of the mobile app with the trained deep learning model, which is the crucial aspect yet to be completed.

3. PROBLEMS ENCOUNTERED

A significant issue arose during the development of the mobile app, it was discovered that the MediaPipe holistic model is not yet supported for Android. Consequently, a decision must be made on how to address this problem.

4. SELF EVALUATION OF THE PROGRESS

We are currently in the final stage of the project, with about 80% of the mobile app completed. The remaining crucial step is the integration process.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	**Study week no.: 8**
Student Name & ID: Lee Teck Junn, 2002030	
Supervisor: Vikneswary a/p Jayapal	
Project Title: Automated Hand Gesture Recognition for Enhancing Sign Language Communication	

1. WORK DONE

The mobile app's development has been finalized, with a decision to exclusively utilize hand landmark for detection. Integration of the deep learning model and MediaPipe hand landmark into app has been successful. Consequently, the app is now capable of detecting user hand movements and interpreting sign language effectively.

2. WORK TO BE DONE

Further work is required to refine the mobile app and streamline the integration process for improved efficiency.

3. PROBLEMS ENCOUNTERED

The issue arises due to the absence of pose landmark detection, which impacts the accuracy of sign language recognition. Finding a solution to address this challenge is imperative.

4. SELF EVALUATION OF THE PROGRESS

Progress is on track, and the project is nearing completion with everything proceeding in the right direction.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	**Study week no.: 10**
Student Name & ID: Lee Teck Junn, 2002030	
Supervisor: Vikneswary a/p Jayapal	
Project Title: Automated Hand Gesture Recognition for Enhancing Sign Language Communication	

1. WORK DONE

A solution has been devised to address the previous issue. Now the mobile app will undergo pose landmark detection first, followed by hand landmark detection. Subsequently, utilizing the results from both pose and hand landmark detection, we can employ the gesture recognizer model to predict the sign language.

2. WORK TO BE DONE

Wrap up the project and proceed to finalize the report.

3. PROBLEMS ENCOUNTERED

An issue has arisen where certain sign language gestures cannot be detected accurately. This might be attributed to the heaviness of the process as the mobile app undergoes three integrated models.

4. SELF EVALUATION OF THE PROGRESS

The objectives have been achieved: the deep learning model has been successfully trained, and the mobile app has been fully developed. Furthermore, the integration of the mobile app with the deep learning model has been accomplished successfully.

Supervisor's signature

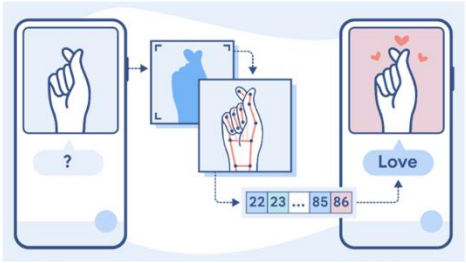
Student's signature

POSTER

AUTOMATED HAND GESTURE RECOGNITION
FOR ENHANCING SIGN LANGUAGE COMMUNICATION

UTAR
UNIVERSITI TUNKU ABDUL RAHMAN

Project Developer: Lee Teck Junn
Project Supervisor: Dr Vikneswary a/p Jayapal

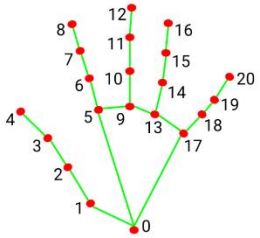


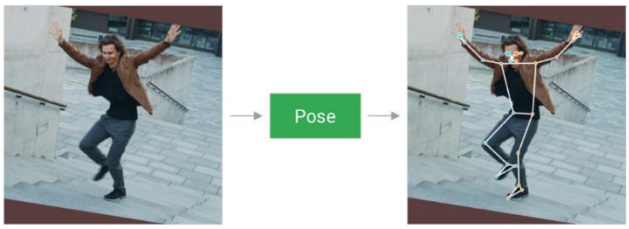
Introduction

The project aims to overcome communication barriers faced by individuals who are deaf when interacting with those unfamiliar with sign language. To achieve this, the project focuses on the **development of a mobile application integrating a sequential deep learning model** capable of real-time recognition and interpretation of sign language.

Method

The method's concept is straightforward: the project will employ MediaPipe Holistic to extract keypoint values from detected hand and pose movements. These collected values will then be utilized to train a sequential deep learning model from scratch.





Result

dynamic

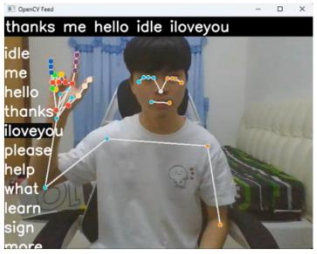
11 classes

0.97

Avg. Accuracy

GRU

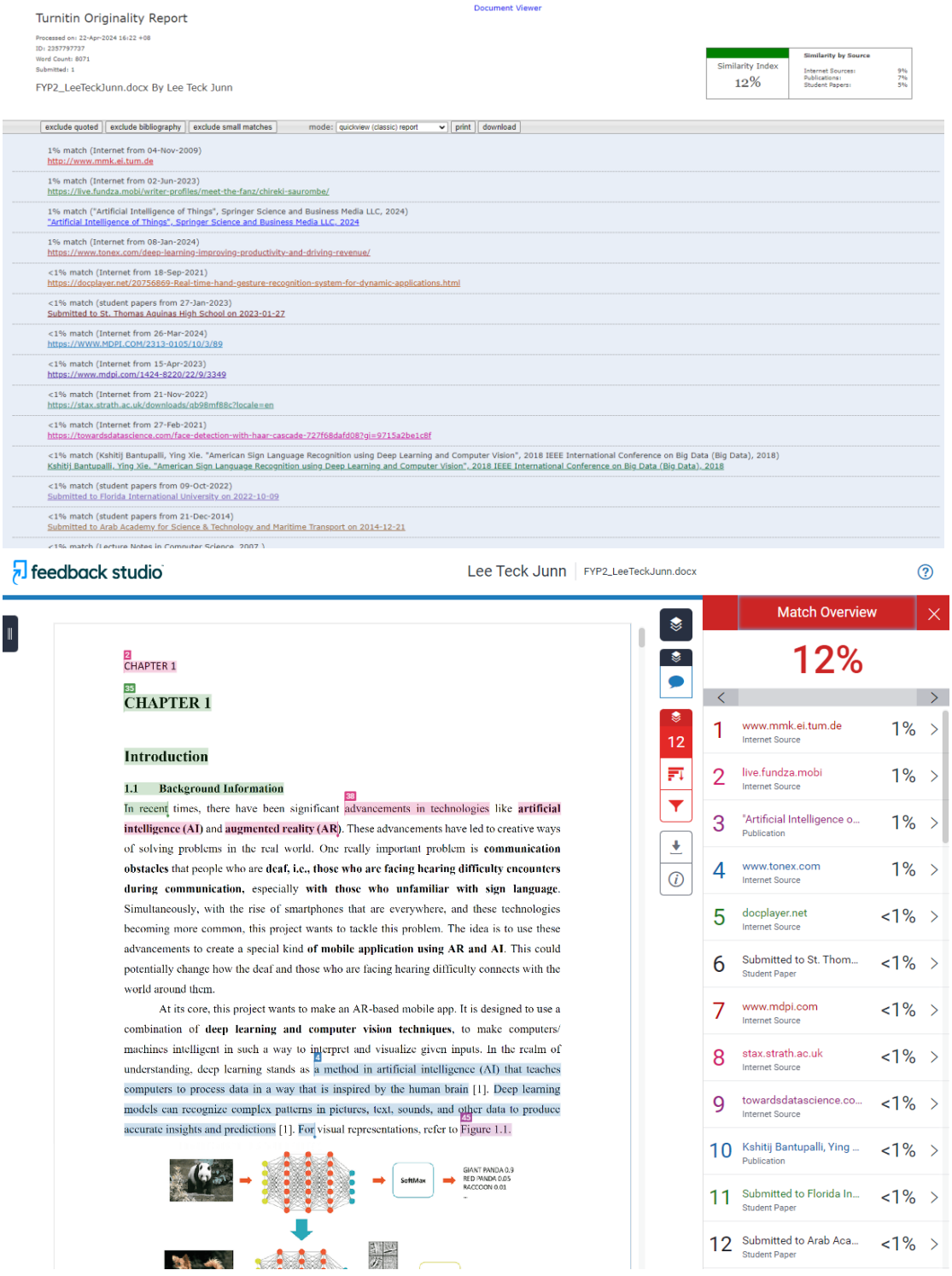
15 epochs



Implementation

A mobile application is developed to integrate with the trained model, empowered the smarthphone camera for seamless recognition processes.

PLAGIARISM CHECK RESULT



Turnitin Originality Report

Document Viewer

Processed on: 22-Apr-2024 16:22 +08
ID: 2357797737
Word Count: 8071
Submitted: 1

FYP2_LeeTeckJunn.docx By Lee Teck Junn

Similarity Index 12%

Similarity by Source
Internet Sources: 9%
Publications: 7%
Student Papers: 5%

exclude quoted | exclude bibliography | exclude small matches | mode: quickview (classic) report | print | download

1% match (Internet from 04-Nov-2009)
http://www.mmk.ei.tum.de

1% match (Internet from 02-Jun-2023)
https://live.fundza.mobi/writer-profiles/meet-the-fanz/chireki-saurombe/

1% match ("Artificial Intelligence of Things", Springer Science and Business Media LLC, 2024)
"Artificial Intelligence of Things", Springer Science and Business Media LLC, 2024

1% match (Internet from 08-Jan-2024)
https://www.tonex.com/deep-learning-improving-productivity-and-driving-revenue/

<1% match (Internet from 18-Sep-2021)
https://docplayer.net/20756869-Real-time-hand-gesture-recognition-system-for-dynamic-applications.html

<1% match (student papers from 27-Jan-2023)
Submitted to St. Thomas Aquinas High School on 2023-01-27

<1% match (Internet from 26-Mar-2024)
https://WWW.MDPI.COM/2313-0105/10/3/89

<1% match (Internet from 15-Apr-2023)
https://www.mdpi.com/1424-8220/22/9/3349

<1% match (Internet from 21-Nov-2022)
https://stax.strath.ac.uk/downloads/qb98mf88c?locale=en

<1% match (Internet from 27-Feb-2021)
https://towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd08?gi=9715a2be1c8f

<1% match (Kshitij Bantupalli, Ying Xie. "American Sign Language Recognition using Deep Learning and Computer Vision", 2018 IEEE International Conference on Big Data (Big Data), 2018)
Kshitij Bantupalli, Ying Xie. "American Sign Language Recognition using Deep Learning and Computer Vision", 2018 IEEE International Conference on Big Data (Big Data), 2018

<1% match (student papers from 09-Oct-2022)
Submitted to Florida International University on 2022-10-09

<1% match (student papers from 21-Dec-2014)
Submitted to Arab Academy for Science & Technology and Maritime Transport on 2014-12-21

feedback studio | Lee Teck Junn | FYP2_LeeTeckJunn.docx

CHAPTER 1

CHAPTER 1

Introduction

1.1 Background Information

In recent times, there have been significant advancements in technologies like **artificial intelligence (AI)** and **augmented reality (AR)**. These advancements have led to creative ways of solving problems in the real world. One really important problem is **communication obstacles** that people who are **deaf, i.e., those who are facing hearing difficulty encounters during communication**, especially **with those who unfamiliar with sign language**. Simultaneously, with the rise of smartphones that are everywhere, and these technologies becoming more common, this project wants to tackle this problem. The idea is to use these advancements to create a special kind **of mobile application using AR and AI**. This could potentially change how the deaf and those who are facing hearing difficulty connects with the world around them.

At its core, this project wants to make an AR-based mobile app. It is designed to use a combination of **deep learning and computer vision techniques**, to make computers/ machines intelligent in such a way to interpret and visualize given inputs. In the realm of understanding, deep learning stands as a method in artificial intelligence (AI) that teaches computers to process data in a way that is inspired by the human brain [1]. Deep learning models can recognize complex patterns in pictures, text, sounds, and other data to produce accurate insights and predictions [1]. For visual representations, refer to Figure 1.1.

Match Overview

12%

1	www.mmk.ei.tum.de Internet Source	1%
2	live.fundza.mobi Internet Source	1%
3	"Artificial Intelligence o... Publication	1%
4	www.tonex.com Internet Source	1%
5	docplayer.net Internet Source	<1%
6	Submitted to St. Thom... Student Paper	<1%
7	www.mdpi.com Internet Source	<1%
8	stax.strath.ac.uk Internet Source	<1%
9	towardsdatascience.co... Internet Source	<1%
10	Kshitij Bantupalli, Ying ... Publication	<1%
11	Submitted to Florida In... Student Paper	<1%
12	Submitted to Arab Aca... Student Paper	<1%

PLAGIARISM CHECK RESULT

Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Lee Teck Junn
ID Number(s)	20ACB02030
Programme / Course	FICT – CS
Title of Final Year Project	Automated Hand Gesture Recognition for Enhancing Sign Language Communication

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)**
Overall similarity index: 12 **%** **Similarity by source** Internet Sources: 9 % Publications: 7 % Student Papers: 5 %	
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required, and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Name: Dr. Vikneswary Jayapal

Date: 24/4/2024

Signature of Co-Supervisor

Name: _____

Date: _____

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	2002030
Student Name	Lee Teck Junn
Supervisor Name	Dr. Vikneswary a/p Jayapal

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

(Signature of Student)
Date: 22/4/2024