INFORMATICS INSTITUTE OF TECHNOLOGY

In Collaboration with

UNIVERSITY OF WESTMINSTER

Sri Lankan Sign Language Mobile Application with Real-Time Motion Recognition

A Literature Review By

Mr. Naveen Grero

Supervised By

Mr. Nathindu Himansha

Submitted in partial fulfilment of the requirements for the BEng (Hons) Software Engineering degree at the University of Westminster

ABSTRACT

The deaf community in Sri Lanka faces major communication barriers due to the dearth of reliable and easily available translation resources for Sri Lankan Sign Language (SLSL). Current mobile applications frequently have issues with real-time motion detection, hand and face gesture identification, and two-way translation.

To fill these gaps, our research uses cutting-edge computer vision and machine learning techniques to create a smartphone application that can translate SLSL in real time. In addition to using CNN-LSTM models for gesture detection and MediaPipe for motion tracking, the software facilitates two-way translation (text/voice to sign language and vice versa). Throughout the development process, user input including surveys and interviews—has been incorporated to guarantee usability and cater to the particular requirements of the intended audience. Early findings show increased processing speed and accuracy when translating intricate motions, indicating the app's potential to improve social inclusion and remove obstacles to communication for the deaf community.

Typical machine learning metrics appropriate for classification tasks were used to assess the trained CNN-LSTM model. Model performance was evaluated using F1-Score, Accuracy, Precision, and Recall. On the validation set, the system's overall classification accuracy was more than 80%. Specifically, the F1-Score stayed high across most gesture classes, suggesting that recall and precision were balanced. The model's practical use for live gesture detection was confirmed by its minimal latency in real-time prediction on mobile devices.

Keywords: Sri Lankan Sign Language, real-time motion recognition, gesture translation, computer vision, two-way communication, machine learning, CNN-LSTM, MediaPipe, sign language app, accessibility.

Subject Descriptions:

- **Human-Computer Interaction (HCI)** -> Assistive technologies -> Accessibility tools > Sign language translation.
- **Computing methodologies** -> Machine learning -> Neural networks -> Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks.

- **Information systems** -> Mobile applications -> Real-time processing systems -> Motion recognition.
- **Computer vision** -> Gesture recognition -> Facial recognition -> Hand tracking.
- **Natural Language Processing (NLP)** -> Language translation -> Text-to-Sign and Sign-to-Text conversion.

DECLARATION

I guarantee that I am the author of this thesis and that it is entirely my original work. It has never before been submitted to another university for a degree or test. In this paper, every source that was used has been properly cited and acknowledged. This thesis is an example of my personal contributions and research efforts, unless otherwise noted. I further declare that all research was carried out in compliance with the institution's ethical standards and guidelines. This statement reaffirms my dedication to honesty and integrity in the production of this academic work.

Name: K.N.S Grero

IIT Student ID: 20200546

UoW ID : w1830258

…………………………..
Signature   Date 01/04/2025

ACKNOWLADGEMENT

First of all, I want to thank God for strengthening me to do my project during the hard times of this journey. Also, I express my gratitude to my beloved parents and brother who have supported and stayed with me in any manner to make my studies possible. I want to begin by thanking my supervisor, Mr. Nathindu Himansha, and expressing my sincere gratitude for all his support, counsel, and expertise. His commitment to education made it easy for me to complete my research, and each of his suggestions improved its performance. Following that, I would like to express my gratitude to everyone who supported me in any way during my research period, as well as to all of the IIT lecturers who educated me and provided the knowledge I required for my study.

Table of Contents

List of Tables

List of Figures

LIST OF ABBREVIATIONS

Acronym	Description
ANN	Artificial Neural Network
API	Application Programming Interface
CNN	Convolutional Neural Network
GPU	Graphics Processing Unit
HCI	Human-Computer Interaction
LSTM	Long Short-Term Memory
ML	Machine Learning
NLP	Natural Language Processing
PoC	Proof of Concept
RCNN	Region-based Convolutional Neural Network
RNN	Recurrent Neural Network
SLSL	Sri Lankan Sign Language
TFLite	TensorFlow Lite
TTS	Text-to-Speech
UI	User Interface
UX	User Experience

1. INTRODUCTION

1.1 Chapter Overview

In Sri Lanka, people frequently fall short of overcoming the challenges faced by deaf people daily and vice versa. A few instances are using a phone, comprehending spoken language in noisy settings, taking part in meetings at work, following a basic discussion with multiple people, and not having subtitles.

1.2 Problem Background

The project's subject is assistive technology and accessibility, with a particular focus on the communication difficulties that the deaf and hearing communities in Sri Lanka experience. Despite being a basic human need, communication can be extremely difficult for the deaf community because there aren't many useful resources available to help them connect with hearing people who utilize Sri Lankan Sign Language (SLSL). SLSL is a separate language from popular sign languages like American or British Sign Language, with its own grammar and gestures. The deaf minority in Sri Lanka is frequently isolated due to its distinctiveness since current resources do not meet their particular requirements.

The domain of sign language translation research emphasizes the necessity of accurate and real-time recognition of dynamic motions, such as body postures, facial expressions, and hand gestures, all of which are essential to SLSL. On the other hand, dynamic motions cannot be efficiently interpreted by current mobile applications, which are primarily made for static sign recognition. Additionally, these technologies frequently don't have two-way translation capabilities, which restricts communication to one-way flows for example, from sign language to text or speech—without providing the opposite. The deaf community faces additional difficulties in obtaining education, participating in the workplace, and interacting with others in daily life due to the lack of real-time, bidirectional communication technologies.

Accessibility technologies have grown in Sri Lanka's digital ecosystem, but significant research and development deficit still exists in the form of assistive solutions for SLSL. Research shows that although gesture recognition and language translation have advanced globally, these technologies are still not entirely tailored to the linguistic and cultural nuances of SLSL. Gesture recognition has shown promise with technologies like convolutional neural networks (CNNs), long short-term memory networks (LSTMs), and frameworks like MediaPipe. However, there is room for creativity as their use in SLSL translation is underutilized.

Beyond just technological constraints, the issue also affects the social and educational spheres. Teachers, students, and members of the deaf community have expressed unhappiness with current technologies in surveys and interviews because of their unidirectional communication skills, lack of real-time processing, and inaccuracy. In order to enable smooth communication between SLSL users and the hearing community, a solution that makes use of cutting-edge machine learning and computer vision techniques is urgently needed.

By developing a mobile application that is specifically suited to the needs of the deaf community in Sri Lanka, our initiative seeks to close these gaps. The suggested approach aims to improve accessibility and promote social inclusion by integrating cutting-edge motion-tracking technologies and bidirectional translation features, ultimately enabling members of the deaf community to interact successfully and independently.

1.3 Problem Definition

Due to the lack of accurate and easily accessible sign language translation resources, the deaf community in Sri Lanka confronts substantial communication obstacles. While helpful, the sign language smartphone applications now available are unable to accurately translate between spoken/written English and Sinhala and accurately recognize the unique hand and facial movements of Sri Lankan Sign Language (SLSL). To close this communication gap, a strong mobile application that can reliably decipher SLSL using cutting-edge motion gesture technology and smoothly convert it into spoken or written language, and vice versa.

1.4 Research Motivation

A complete understanding of the communication obstacles that the deaf community in Sri Lanka faces serves as the driving force behind this study. Being raised in a culture where spoken and written language predominates, I have direct experience with the difficulties that people with hearing impairments face on a daily basis, ranging from difficulty communicating with non-signers to exclusion from school and the workforce. This gap has grown as a result of the lack of an efficient, real-time Sri Lankan Sign Language (SLSL) translation tool, which has limited the deaf community's ability to smoothly integrate into society. The dynamic aspect of sign language is not captured by current systems, which either concentrate on static gesture identification or do not provide bidirectional translation. The motivation behind this project is a personal dedication to using technology to close this gap by creating a cutting-edge, AI-powered smartphone application that allows for real-time SLSL translation. In order to promote meaningful interactions between the hearing and deaf communities and ultimately advance assistive communication technology and social equity, this research will integrate motion detection, computer vision, and machine learning to develop an accessible, inclusive, and empowering communication tool.

1.5 Research Gaps

● **Gap 1: Dynamic Motion Recognition**

The majority of current sign language translation technologies concentrate on analyzing static images or video frames, which restricts their capacity to identify and decipher the dynamic body, facial, and hand motions that are inherent to SLSL. There is a gap in accurately reading real-time gestures due to the severe under-exploration of advanced motion detection algorithms that take temporal and spatial changes into consideration.

● **Gap 2: Accurate real-time translation of written language into sign language** The accuracy and speed of translation between spoken/written language and sign language is a common problem with current sign language apps. This restriction makes it difficult for deaf people to communicate and puts up obstacles for them.

● **Gap 3: SLSL translator mobile apps do not have 2-way translations in it(text to sign language and vice versa)**

One-way translation is provided by the majority of sign language apps, usually from sign language to text or speech. Bidirectional translation, which enables users to transition between text, speech, and sign language with ease, is severely lacking. The accessibility and usability of the software are restricted by this limitation.

These gaps in the research show how much more work has to be done to improve communication for deaf people through improvements in real-time translation, sign language recognition technology, and user-friendly features.

1.6 Contribution to Body of Knowledge

1.6.1 Contribution to Problem Domain

The study focuses on human-computer interaction (HCI) and assistive technologies, aiming to create a system for real-time translation of spoken or written English into sign language. The method will use sophisticated motion recognition techniques for hand and facial motions, improving accessibility for the deaf community. This research will promote inclusivity and create smarter, compassionate assistive technologies that cater to user needs.

1.6.2 Contribution to Research Domain

An emergency solution based on research on instantaneously converting SLSL motions to voice allows SLSL speakers to communicate using signs. Wadhan employs models from several research domains, including speech recognition, natural language processing, and gesture detection, to achieve the desired outcomes (Dewasurendra et al., 2020).

Additionally, D. Manoj Kumar built an RCNN-based model and used an API based on machine learning to translate hand image recognition into spoken language; however, the responses are too slow, and the processing time is too long. (Manoj Kumar et al., 2020)

Thus, developing a mobile app for Sri Lankan sign language translation that incorporates hand and facial motion identification and offers accurate real-time translation from spoken or written language into sign language would be a useful addition to the domain of research.

1.7 Research Challenge

Challengers are expected to emerge during the study after an assessment of the previously indicated research areas. Below is a list of some of them.

- **Lack of Sri Lankan Sign Language (SLSL) Labeled Datasets**

The lack of high-quality, labeled datasets for SLSL was one of the main obstacles in our study. It is challenging to build an appropriate machine-learning model for SLSL because it lacks large digital datasets, in contrast to sign languages that are well studied, such as American Sign Language (ASL) or British Sign Language (BSL). It was necessary to capture real-time sign gestures, extract crucial frames, and manually label them to construct a custom dataset. To guarantee data diversity and accuracy, this procedure took a long period and a lot of work.

- **Accuracy for Real-Time Motion Recognition**

Sign language involves intricate hand, face, and body gestures and is very dynamic. A significant technological obstacle was achieving high-accuracy real-time recognition because the models that were in use had trouble distinguishing between minute differences in gestures. To process sequential movement patterns efficiently and preserve real-time responsiveness, the CNN-LSTM model had to be optimized. Accuracy and dependability had to be improved by adjusting hyperparameters, diversifying the dataset, and utilizing MediaPipe's motion-tracking features.

1.8 Research questions

RQ1: Which machine learning models are being applied to SLSL (Sri Lankan Sign Language) at the moment?

RQ2: How can SLSL (Sri Lankan Sign Language) signals be reliably recognized and translated using computer vision and machine learning?

RQ3: How can an app for SLSL translation be created that accurately recognizes and translates words using SLSL, even ones with complex grammatical structures?

RQ4: In an SLSL translator app, what is the ideal ratio between offline functionality and realtime translation?

1.9 Research Aim

The purpose of this study is to create, develop, and assess a novel mobile application that facilitates precise and instantaneous translation of Sri Lankan Sign Language (SLSL), thereby bridging the communication gap between the hearing and the deaf communities in Sri Lanka. By using cutting-edge motion detection technologies like MediaPipe and the CNN-LSTM model, the project aims to improve the accuracy of gesture recognition for hand, face, and body movements while addressing the shortcomings of current tools. Additionally, by offering a bidirectional translation mechanism, the application hopes to provide smooth communication between sign language and text/voice and vice versa. The solution will guarantee inclusion and accessibility for its intended audience by concentrating on the unique linguistic and cultural quirks of SLSL. The final objective is to develop a mobile platform-optimized application that is effective, lightweight, and easy to use. This will empower members of the deaf community and encourage social inclusion by improving accessibility.

1.10 Research Objectives

Objective	**Research Objective**	**Description**	**Learning Outcome**	**Research Questions**

Problem Identification	Undertake a thorough investigation into an issue that requires resolution. **R1:** Examination of issues found in the field of social services (creating more userfriendly mobile app for deaf community). **R2:** Determining the shortcomings of the current SLSL translators **R3:** Determining the significance of proper SLSL translators in the social services domain. **R4:** Investigating the algorithms that are now in use to gain insight into SLSL translator mobile app technological techniques.	Perform a thorough analysis to comprehend and pinpoint problems in the social services industry, paying special attention to improving mobile technologybased user-friendly solutions for the community. Th evaluating employed technological algorithms of the investigating shortcomings SLSL translat are currently in	**LO2,** **LO4**	**RQ2**

Literature Review	Evaluation and analysis of earlier research in the field. **R5:** Examining and evaluating the motion gesture analysis models that are currently in use **R6:** Examine and evaluate the CNN, RNN, and AI frameworks presently employed in this domain. **R7:** Examine and identify the existing research gaps in the problem and research domains.	Conduct a thorough investigation of previous research on motion gesture analysis and AI frameworks like CNN and RNN. Examine existing technological constraints and research needs in the field of SLSL translation.	**LO1, LO4, LO5**	**RQ1, RQ2**

Data Gathering and Analysis	Obtaining the project's requirements and evaluating the data acquired. **R8:** Recording performances in sign language to train and assess models.	Record sign language performances, examine user comments and utilize statistical techniques to analyze user surveys to gather and assess data. The purpose of this step is to learn from the user	**LO1,** **LO2,** **LO4,** **LO5**	**RQ1,** **RQ2,** **RQ4**
	R9: Gathering user opinions on the features, contentment, and usefulness of the app. **R10:** Using statistical techniques to examine user survey data, usage trends, and performance indicators. **R11:** Patterns, themes, and user insights can be found in interview and group transcripts by using thematic	experiences and thematic trends in the data that has been gathered.		

Design	Creating a working prototype and system elements for a mobile app that translates SLSL. **R12:** Easy navigation and accessibility for users with different levels of tech skill should be guaranteed by the design. **R13:** Users should receive clear and educational feedback from the app while it is translating. **R14:** To find areas for development and enhance the user experience, test usability.	Create a mobile app prototype that is accessible to users with varying degrees of technical proficiency. To improve usability, make sure the design encourages simple navigation, intuitive user interfaces, and feedback during translations.	**LO2,** **LO5**	**RQ2,** **RQ3**

Implementation	Putting into practice a system that can address the main shortcomings of current systems in order to improve the real-time translation accuracy of the SLSL translator mobile app and include motion sensor detection. **R15:** Constructing a strategy that integrates all goals.	Improve the correctness of real-time SLSL translation by integrating and implementing a mechanism that fixes known flaws. During this phase, a strategic solution for motion sensor detection and model integration is developed and tested.	**LO2,** **LO5,** **LO7**	**RQ3,** **RQ4**

Testing and Evolution	Testing, assessing, and contrasting the application with the models and systems that are being used. **R16:** Evaluating the precision and effectiveness of various models for sign language recognition and translation algorithms. **RO17:** Provide a working prototype and ask experts in the field for input. **RO18:** Assessing how the produced app affects hearing and deaf people's ability to communicate.	To evaluate the accuracy of the model and the efficacy of the translation, compare the new app with the current systems and get professional input. Assure ongoing development and assess how the software affects hearing and deaf users' ability to communicate.		**RQ3,** **RQ4**

Table 1. 1: Research Objectives

1.11 Chapter summary

In our daily routines, Sri Lankan Sign Language (SLSL) is a very helpful language to communicate with the deaf community. However, there are situations where hand gestures and facial expressions can be complex and challenging to figure out, facing numerous challenges even for the deaf community, which cannot understand standard languages. A mobile application for translating sign language that tracks facial expressions, has better dataset accuracy, and is easy to use could be a big help.

2. LITERATURE REVIEW

2.1 Chapter Overview

An extensive review of the research on mobile applications for Sri Lankan Sign Language (SLSL) that use real-time motion recognition is provided in this chapter. It points out important research gaps, such as the absence of real-time translation capability and motion sensor capture detection. For the Deaf population in Sri Lanka to benefit from increased accessibility, more effective communication, and social inclusion, these gaps must be filled. The most recent developments in AI and image/video processing methods for Sri Lankan Sign Language Mobile Applications with Real-Time Motion Recognition will be covered in this chapter. This chapter will detail state-of-the-art instruments, technological advancements, and methods that can be applied to implement the suggested solutions.

2.2 Concept Map

A concept map displays the technologies, techniques, and other crucial project-related components of the study field, including computer vision, artificial intelligence, and deep learning. Following a thorough literature review of the project's domain and scope, the graph provides a visually appealing, yet educational summary of the data acquired. **Appendix A – Concept Map** provides the foundation for the concept graph.

2.3 Problem Domain

2.3.1 Introduction to Deafness

A wide range of hearing loss, from modest impairment to complete deafness, is included in the complicated condition of deafness. Making the distinction between deafness and hearing loss is crucial. A partial loss of hearing is referred to as hearing loss, whereas a severe or total loss of hearing is usually indicated by deafness.

Deafness has a wide range of reasons, from environmental variables to genetic issues. Some people are born deaf (congenital deafness), while others get hearing loss later in life as a result of disease, trauma, or aging-related degeneration. People's capacity to perceive sounds at various pitches and volumes can be greatly impacted by the type and degree of hearing loss. It is important to understand that being deaf is a cultural and linguistic identity in addition to a medical condition.

Deaf culture, with its own history, language (Sri Lankan Sign Language, for example), and social conventions, has become a potent force in promoting the interests and rights of the deaf community. The effects of deafness on a person's life are complex. Having trouble speaking becomes a major problem when spoken language is not supported appropriately. This may have an impact on social connections, work, and education. However, many people with hearing loss can enjoy happy lives because of technological developments like cochlear implants, assistive listening devices, and hearing aids. Furthermore, the advancement of sign languages and the increasing recognition of deaf culture have enabled deaf people to achieve success.

2.3.2 Sri Lankan Sign Language (SLSL)

Sri Lanka's deaf community uses Sri Lankan Sign Language (SLSL), a visual language. It's a sophisticated communication system that interprets meaning using hand gestures, body language, facial expressions, and spatial orientation. Although SLSL is comparable to other sign languages, it has a distinct vocabulary and syntax of its own. In the past, Sri Lanka had a diverse linguistic environment, with distinct sign languages originating in various parts of the country. The lack of uniform sign language and the isolation of deaf groups were reflected in these regional variances. However, attempts to harmonize these diverse sign systems began with the founding of deaf schools during the colonial era. The modern recognition of SLSL is a very recent development that emerged during this standardization phase.

The deaf community in Sri Lanka is making a concerted effort to highlight SLSL as an essential component of their identity and culture. They hope to change society by advocating for a more accepting and respectful vocabulary.

2.3.3 Mobile App for Sri Lankan Sign Language

Sri Lankan Sign Language (SLSL) mobile apps are a kind of digital technology that lets hearing and deaf people communicate with each other. It attempts to make SLSL practice, learning, and communication easier. Community forums, interactive lectures, real-time translation, and sign language dictionaries are just a few of the things that such an app might provide. Through technological accessibility, SLSL may foster inclusion and enable successful communication and interaction between hearing and deaf people.

2.3.4 Motion Recognition and Gesture Recognition

- **Motion Recognition**

  The foundation of a strong Sri Lankan Sign Language (SLSL) recognition system is motion recognition. To extract relevant information, it entails recording and analyzing human body motions. This approach makes use of cutting-edge technology like deep learning, machine learning, and computer vision. The dynamic motions that are the foundation of SLSL can be recognized and understood by using methods like optical flow, skeleton tracking, and posture estimation.

- **Gesture Recognition**

  A subclass of motion recognition called gesture recognition is especially concerned with hand and face gestures. These are essential SLSL elements that communicate complex meanings and nuance. To accurately read the intended signs, methods such as fingertip tracking, hand form analysis, and face recognition of expressions are used. It has been demonstrated that Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are capable of accurately simulating the intricate patterns found in sign language gestures. It is possible to create complete systems that can correctly translate SLSL by combining motion and gesture detection, providing the deaf community in Sri Lanka with new opportunities for accessibility and communication.

2.4 Existing Work

2.4.1 Emergency Communication Application for Speech and Hearing

The innovative research conducted by (Dewasurendra et al.,2020) A smartphone application called "Wadhan" was created to help Sri Lankan Sign Language (SLSL) users communicate in an emergency by translating SLSL motions into voice messages and vice versa, enabling direct connection with emergency personnel. It incorporates technologies like text-to-speech conversion MaryTTS, speech recognition CMUSphinx, animation creation for displaying SLSL motions, and gesture recognition EfficientNet-Lite0. The software allows for real-time emergency requests without the need for middlemen, thereby addressing the communication hurdles posed by the speech and hearing challenged. Although the test dataset yielded excellent accuracy (99.76%) in gesture recognition, the accuracy decreases to 68.21% for unseen data, partially because of COVID-19 restrictions on data acquisition. Other difficulties include varying processing periods that can affect response and less than ideal speech clarity and recognition accuracy. In order to improve emergency communication for SLSL users, the app's future development will concentrate on growing the SLSL vocabulary, enhancing model performance, and minimizing processing delays.

2.4.2 A Multimodal Approach for Real-Time Sinhala Sign Language Translation

The work of (Gamage et al., 2023) to improve communication between the deaf and hearing people in Sri Lanka, the study describes a smartphone application that enables real-time translation of Sinhala Sign Language (SLSL). Three primary functions of the program are supported: translating text from Sinhala to English, converting English text to a 3D SLSL avatar, and translating text from SLSL motions to text. For real-time gesture identification and translation, the system makes use of a multimodal approach and machine learning models, such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Time Sequence Neural Networks. While the frontend is created with Unity and Blender for the Android platform, the backend is constructed with Python, Flask API, TensorFlow, and Keras. For speech translation, Google Speech-to-Text API is used for both Sinhala and English.

The goal of the study is to improve communication accessibility for the hearing challenged in Sri Lanka by introducing a multimodal approach for interpreting Sinhala Sign Language in real-time. It closes the gap in the current SLSL translation technologies and offers a full translation solution with speech, sign-to-text, and text-to-sign capabilities.

The accuracy and generalizability of the translation models may be impacted by the application's reliance on a small dataset. Furthermore, the diversity and quality of training data determines the system's efficacy, and enhancements are required to handle complicated gestures and dialectal variations in SLSL.

2.4.3 A Static Sinhala Sign Language Translation using Media-pipe and SVM Compared with Hybrid Model of KNN, SVM and CNN Algorithms

The innovative research conducted by (Gedaragoda et al.,2023) using machine learning and computer vision techniques, the research provides a translation system for Sinhala Sign Language (SLSL) that focuses on static sign recognition. Three models are presented: a CNN-based VGG16 model, a hybrid model incorporating SVM, KNN, and Random Forest, and a hand model that uses MediaPipe to detect hand landmarks. With the Hand model, the research outperformed the Hybrid model with an accuracy of 99.69%, exceeding 86.57%. In order to overcome the lack of resources, a new dataset for SLSL was also produced, concentrating on numbers and specific words. The technologies employed are machine learning methods for categorization, OpenCV for image processing, and MediaPipe for landmark identification. Among the contributions are the creation of a new dataset for research purposes and a highly accurate static SLSL recognition system. The small dataset size, the CNN model's high processing requirements, and the CNN model's exclusive focus on static signs which leave out dynamic or compound signs are among its limitations. The objective of the next research is to augment the dataset and include dynamic sign recognition to improve accessibility for the deaf populace.

2.4.4 Multi-modal Deep Learning Approach to Improve Sentence level Sinhala Sign Language Recognition

In order to improve communication between Sri Lankans who use Sri Lankan Sign Language (SSL) and those who do not, the article presents "EasyTalk," a web-based translator. It uses animated visuals to translate English text into SSL and translates SSL motions into text or speech. Hand Gesture Detector, Image Classifier, Text and Audio Generator, and Text to Sign Converter are the four primary parts of the system. CNN for image classification, Faster RCNN for gesture recognition, TensorFlow for model training, and Natural Language Processing (NLP) for text processing are important technologies. With a detection accuracy of more than 91% in a variety of settings, EasyTalk efficiently facilitates real-time translation. It needs high-resolution photos for best accuracy, though, and has trouble recognizing similar sign gestures, which might result in detection errors. Future enhancements include adding 5G technology to improve communication efficiency and creating a mobile application for faster reaction times.

2.4.5 Multi-modal Deep Learning Approach to Improve Sentence level Sinhala Sign Language Recognition

In the study (Akshit Tayade and Halder, 2021) order to help the deaf-mute community communicate, the paper presents a real-time sign language recognition system that uses MediaPipe and machine learning. It uses Support Vector Machine (SVM) for categorization and MediaPipe's architecture for accurate hand gesture tracking, attaining a high accuracy of 99% across several sign languages, including Italian, Indian, and American sign languages. The technology offers a portable and affordable substitute for conventional, sensor-based recognition techniques since it is lightweight and compatible with smart devices. SVM for classification, MediaPipe for feature extraction, and TensorFlow for model deployment are important technologies. The system needs more work to properly handle dynamic sign language and full sentence translation because it is now only capable of handling static motions, despite its great accuracy and versatility.

2.5 Technology Review

2.5.1 Dataset

This project's dataset was specially created to address the recognition of Sri Lankan Sign Language (SSL). Videos of sign language motions were captured and subsequently transformed into structured training data because there aren't many publicly accessible datasets for SSL. Multiple sign gestures executed by many people make up the dataset, guaranteeing diversity in hand gestures, facial expressions, and signing pace variations. To capture movement-based patterns, each movie was divided into 30-frame segments while preserving temporal consistency. Furthermore, a variety of signing settings, including varying lighting conditions, angles, and speeds, were included in the dataset to improve model generalization.

2.5.2 Pre-processing

To guarantee high-quality input for model training, preprocessing was essential. To retrieve 30 consecutive frames for each sign instance, the captured movies were first transformed into image frames. To enhance motion recognition, position landmarks (33 keypoints), hand landmarks (21 each), and facial landmarks (468 points) were extracted using MediaPipe's holistic model. To guarantee positional consistency across frames, all the keypoints were then scaled and normalized. Furthermore, data augmentation techniques such as horizontal flipping, brightness modifications, and small rotations were applied to improve robustness against environmental perturbations.

2.5.3 CNN Architecture and Training

Long Short-Term Memory Networks (LSTM) and Convolutional Neural Networks (CNNs) are specifically designed for real-time sign language detection, using large datasets to train on differentiating characteristics of sign motions. The CNN-LSTM architecture learns from labeled sign frames through its feature extraction and classification layers. CNN-LSTMs are a strong tool for dependable real-time sign language detection since the model's performance and accuracy of recognizing sign motions can be enhanced by iterative improvement.

2.5.5 Feature Engineering

Real-time feature engineering Due to processing efficiency requirements and computational limitations, SLSL mobile applications pose special obstacles. To successfully represent hand shapes, movements, and postures, it entails extracting useful information from raw video footage. Feature engineering for a real-time SLSL mobile application usually involves the following steps:

Data acquisition: involves gathering a wide range of SLSL videos, making sure that the lighting, background, and signer characteristics are all different.

Preprocessing video: Segmenting videos into separate signs or sign phrases is known as video segmentation. Extracting individual frames from every video segment is known as frame extraction. Using methods such as object detection or background subtraction, the hands are separated from the backdrop.

2.5.6 Feature Extraction

Feature extraction is a crucial step in a machine learning project that helps a system evaluate more significant and pertinent portions of the dataset being used with greater accuracy and efficiency.

In the thesis *Multi-modal Deep Learning Approach to Improve Sentence-level Sinhala Sign Language Recognition* by Haputhanthri et al. (2023), the feature extraction process incorporates both manual and non-manual features to enhance the accuracy of continuous Sinhala Sign Language recognition. Manual features primarily involve the extraction of hand gesture movements using a two-stream 3D convolutional neural network (3D-CNN), which captures spatial and temporal information from video sequences. Non-manual features focus on extracting lip movements by employing a separate lip-reading module based on the LipNet architecture. The extracted features from both streams are then fused using a late fusion strategy to improve the recognition performance. This combination allows the system to interpret the complex interactions between hand gestures and lip motions, enabling sentence-level understanding in Sinhala Sign Language recognition (Haputhanthri et al., 2023).

In another thesis *Convolutional Neural Networks: A Novel Approach for Sinhala Sign Recognition System* by Dilakshan and Priyadarshana (2020), feature extraction is achieved through a series of image-processing techniques aimed at isolating hand gestures for accurate recognition. The process begins with color segmentation using a Gaussian Mixture Model (GMM) to filter out the background and focus on skin color regions. Following segmentation, morphological transformations and smoothing techniques such as Gaussian blur and median blur are applied to enhance the quality of the extracted hand region. The resulting binary images are then used as input for a Convolutional Neural Network (CNN), which performs automated feature learning. The CNN extracts hierarchical features from the pre-processed images through convolutional, pooling, and activation layers, enabling effective classification of 12 distinct Sinhala sign language gestures (Dilakshan and Priyadarshana, 2020).

2.5.7 Evaluation Metrices

Evaluating the overall effectiveness of the real time sign language detection. It necessitates evaluating important measures including F1-score, recall, accuracy, and precision, among others. Furthermore, evaluating these models' accuracy and dependability in producing results requires comparing them to the state-of-the-art techniques for diagnosis and emotion recognition.

2.6 Evaluation and Benchmarking

This chapter examines benchmarking and evaluation techniques used by different studies in this field.

EasyTalk: A Translator for Sri Lankan Sign Language using Machine Learning and Artificial Intelligence This research has conducted user surveys indicating a strong need for accessible sign language translators, particularly emphasizing the benefits of web-based systems. However, the study acknowledged certain limitations, such as difficulties in distinguishing similar sign gestures in real-time scenarios. Future work includes extending the application to mobile platforms and enhancing system responsiveness with the advent of 5G technology *(Kumar et al., 2020)*.

Translation of Sri Lankan Sign Language to Sinhala Text: A Leap Motion Technology-based Approach this thesis introduces an innovative sensor-based method using Leap Motion technology combined with geometric template matching and Natural Language Processing (NLP) to interpret Sri Lankan Sign Language (SSL) into Sinhala text. The evaluation of the system demonstrated an average accuracy of 80% for static signs and 77% for dynamic signs, highlighting its capability to effectively recognize and translate complex gestures. The benchmarking was conducted using datasets that included signs with multiple meanings and combined signs. Results indicated that the system achieved a detection ratio of 0.8 and a reliability score of 1 for signs with multiple meanings, while for combined signs, the detection ratio was slightly lower at 0.7, with a reliability score of 0.63. The word error rate (WER) for the multiple-meaning signs was minimal (0.1%), whereas the WER for combined signs was higher at 12%. Although the system showed strong performance in interpreting static gestures, dynamic gestures presented more challenges due to their complexity. The study suggested that integrating additional technologies, like Artificial Neural Networks (ANN) and external cameras, could enhance the system's future accuracy and robustness *(Rishan et al., 2022)*.

Multi-modal Deep Learning Approach to Improve Sentence-Level Sinhala Sign Language Recognition the give thesis presents a comprehensive and innovative evaluation of continuous Sinhala Sign Language recognition (CSLR) using a multi-modal deep learning architecture. The study addresses significant limitations in prior Sinhala SLR research, which focused mainly on isolated word-level recognition and primarily utilized manual features such as hand gestures. By integrating both manual (hand movements) and non-manual (lip movements) features, the proposed model demonstrates improved recognition accuracy at the sentence level. The evaluation uses standard metrics such as Word Error Rate (WER) and Specific Word Accuracy (SWA). The multi-modal fusion model achieves a notable reduction in WER to 12.70%, outperforming both the baseline (17.41%) and lip-only (29.69%) models. The authors further benchmark their model on various test sets, confirming its generalization capability, although performance gaps remain when applied to signer-specific variations due to dataset limitations. The proposed model's performance is validated through practical implementation in a prototype application, highlighting its potential to bridge communication barriers for the hearing-impaired community in Sri Lanka. Overall, the benchmarking and evaluation underscore the advantages of multi-modal deep learning and late fusion strategies in enhancing CSLR systems *(Haputhanthri et al., 2023).*

Convolutional Neural Networks: A Novel Approach for Sinhala Sign Recognition System this thesis demonstrate the effectiveness of their CNN-based gesture recognition model for Sinhala sign language. The system was tested using a dataset of 26,000 images per gesture type, capturing 12 basic hand signs. During evaluation, the proposed CNN model achieved an average accuracy of 89.9%, outperforming other baseline models such as ResNet-50 and Fully Convolutional Networks (FCN), which achieved accuracies of 86.3% and 85.3%, respectively. The error rate of the CNN model was reported as 10.1%, and a confusion matrix was used to further validate classification performance, showing minimal misclassifications across the gesture categories. This benchmarking illustrates the superiority of the CNN approach in terms of accuracy and efficiency when compared to traditional machine learning models and highlights its potential for practical applications in Sinhala sign language recognition *(Dilakshan and Priyadarshana, 2020).*

2.7 Chapter Summary

The chapter addresses some main challenges associated with Brain Magnetic Resonance Imaging (MRI), which are recognizing the issues due to image quality, interpretation difficulty, and time consumption for radiologists. The chapter proposes two contemporary technical solutions: Convolutional Neural Networks (CNN) and Explainable AI (XAI). These solutions aim to improve the effectiveness of MRI by employing advanced neural networks and providing interpretability in the diagnostic process. Chapter starts by providing the concept map which graphically represents the finding related to the research, after a critical review of literature by referring to research papers, journals and articles of existing domain related work. Next the chapter itself describes the insights of the technologies, datasets and feature engineering procedures to be used for the proposed novel approach of automated MRI image analyzing system.

3. METHODOLOGY

3.1 Chapter Overview

To effectively achieve our goals, this chapter outlines the essential methods the author used for this study endeavor. The writer will outline the methods, approaches, and instruments utilized throughout this chapter to gather, examine, and evaluate data to answer the research questions and meet the objectives of the overall project.

3.2 Research Methodology

Layer	What is being using	Why you are using it
Philosophy	Pragmatism	The study aims to create a practical SLSL mobile app translator, utilizing pragmatism and combining quantitative and qualitative research approaches to address real-world communication issues.
Approach	Deductive	Finding current theories and models in the fields of machine translation, human-computer interaction, and sign language recognition would be the first step in the research process.
Methodological Choice	Mixed Method	This study explores the relationship between technology and human behavior in sign language translation, combining quantitative and qualitative research methods for comprehensive understanding and improved validity and consistency.
Strategy	Surveys Official Reviews	The author uses surveys and official interviews to enhance usability in common processes, focusing on new outcomes and addressing developed research questions.
Time Horizon	Cross-sectional	The author determined that the cross-sectional time frame is the best time horizon for this

		project because data is collected often, and the dataset is modified during the project.
Data Collection and Analysis	Interview	Give detailed explanations of user requirements and experiences.

Table 3. 1: Research Methodology

3.3 Development Methodology

The author has concluded that both prototyping, and the agile methodology can be helpful at different phases of development after weighing the benefits and drawbacks of both approaches. Therefore, the author approaches the project using a combination of agile methods and prototyping.

1. **Prototyping:** Developing a functioning model of the product (mobile app) in order to assess its usability and functionality is known as prototyping. This can be used to find and address any issues before the system is finished developing, as well as to gather input from domain experts. In the beginning phases of the project, this is an excellent strategy.

2. **Agile Scrum Methodology:** The author can switch to the One-person agile technique once the system requirements and revisions have been determined, and they are satisfied with the prototype. Even when an author is working alone, this strategy is more effective and appropriate for producing and distributing software in a timely and effective manner.

3.4 Project Management Methodology

The author chose the SCRUM Agile methodology for a machine learning project due to its versatility, adaptability, and compatibility with iterative data analysis, model improvement, and deaf user demands. This approach ensures a high-quality project end, balancing process control and flexibility, despite changing objectives and technological hurdles.

3.4.1 Gantt Chart

Appendix B - Gantt Chart has a Gantt chart that details the project plan along with the dates and deadlines.

3.4.2 Deliverables

Deliverable	**Date**
Initial Project Proposal	30th July 2024
Literature Review Document	1st September 2024
Final Project Proposal along with finalized LR	30th August 2024
Project Proposal and Requirement Specification	4th November 2024
Interim Progress Demonstration	2nd February 2025
Final Project Report (Thesis) and Demo Complete documentary including all project related details including findings, analysis, and conclusions of a research project.	1st April 2025

Table 3. 2: Deliverables

3.5 Resources

3.5.1 Hardware Resources

AMD RYZEN 5-3600	Used AMD RYZEN 5-3600 processor for this project
16GB RAM	RAM is needed to manage the execution of machine learning models and image processing.
1TB Disk Space	Space for storing necessary files and datasets.
GPU	Improved speed in machine learning model-related mathematical calculations and image/video processing operations.
DSLR Camera (Sony Alpha a7 3)	Used camera to collect and record dataset

Table 3. 3: Hardware Resources

3.5.2 Software Resources

Operating System (Windows 10)	The Windows 10 operating system is likewise user-friendly for the project
Python Programming Language	The most practical models for ML projects are because of their abundant libraries that make data science processors easier to use.
Keras/TesnsorFlow/ Mediapipe/OpenCV	Framework for managing and creating models in data science.
MS Office 2018 or above	For the purpose of documentation and the research thesis
Google Drive	Store and create a backup on files and documents.

Pycharm	To create the Mobile App
Jupyter Notebook	To train the CNN-LSTM model and to collect data

Table 3. 4: Software Resources

3.5.3 Technical skills

- Computer vision requires proficiency in image processing, video analysis, and deep learning methods (CNNs, LSTMs).
- Robust base in model training, evaluation, and algorithm development.
- Proficiency in text processing, computer translation, and language modeling.
- Proficiency in developing mobile applications using native or cross-platform frameworks (such as Kivy).

3.5.4 Data Requirements

It is crucial to capture different signers in varying lighting circumstances using high-quality image and video captures. To train translation models, precise and synchronized text or speech data is essential. The information is further enhanced by metadata pertaining to signers, recording environments, and sign classifications. User data and performance metrics are optional, but they can offer insightful information on how to make the program better. Throughout the data collecting and preparation process, it is critical to maintain data consistency, privacy, and quality.

3.6 Risks and Mitigation

Risk	Probability of Occurrence	Magnitude of the loss	Mitigation Plan
Time management problems	4	2	Create a deadline in advance and operate quickly and efficiently

(Top Risk) may arise for the other course tasks and examinations.			according to a solid work plan.
Technical problems can happen and cause delays in the project's deadlines.	4	5	Keeping accurate code versions and having backups for project resources at all times.
Unexpected events like health issues or outside problems that are out of control	4	2	To reduce the likelihood of uncontrollable situations, it is essential to set early deadlines, work quickly, and have a backup plan in place.
Resource Limits In order for machine learning-based processing to function, specific computer resources may be required.	4	5	Making the best use of resources by employing cloud-based solutions like Google and Microsoft.

Table 3. 5: Risks and Mitigation

3.7 Chapter Summary

The chapter includes a description of the proposed project's working approach and work plan. It covers the methods for developing and resolving difficult situations that will be used throughout the project, in addition to the project's deliverables and deadlines. The section that follows goes into additional detail about the project's research goals and scope, as well as an example prototype.

4. SOFTWARE REQUIREMENTS SPECIFICATION

4.1 Chapter Overview

The features, functionality, and technical specifications of the web application under development are thoroughly examined in this chapter. The chapter itself assesses the internal and external environments that could have an impact on the process of developing and implementing the system. This chapter also makes use of graphic representations to offer high-level explanations of its topics. In the end, this chapter concentrates on helping researchers and developers clearly comprehend the needs for the system.

4.2 Rich Picture

The relationship and interactions between the system environment's benefits and stakeholders are depicted in the rich picture diagram. This helps connected parties collaborate effectively and gives a clear understanding of the system and system environment.

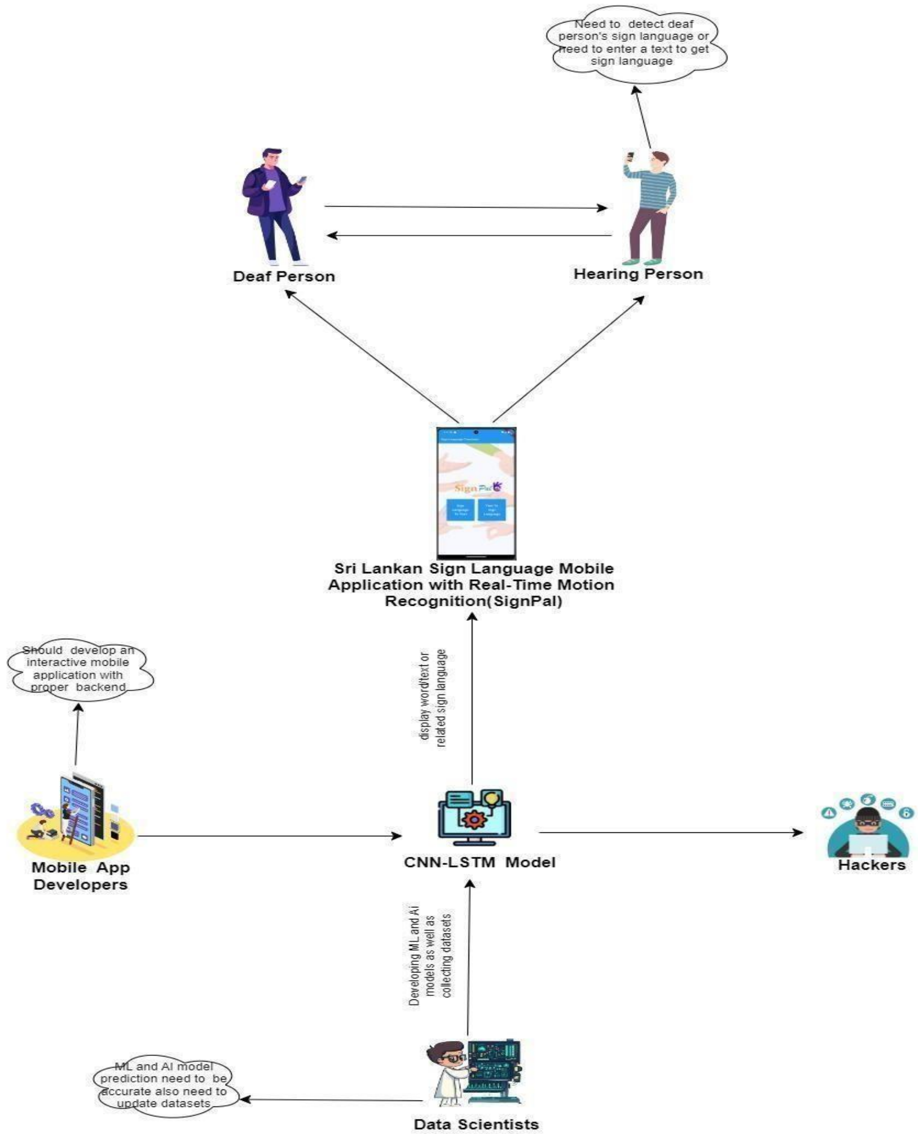


Figure 4. 1: Rich Picture Diagram

4.3 Stakeholder Analysis

When undertaking a project, it's critical to recognize and comprehend the major stakeholders who are particularly interested in its conception and execution. A wide range of people are involved in the proposed project, and everyone has a distinct viewpoint. Involving stakeholders at every stage of the project will guarantee that their demands are taken into account and suitably met, increasing the project's likelihood of success.

4.3.1 Onion Model

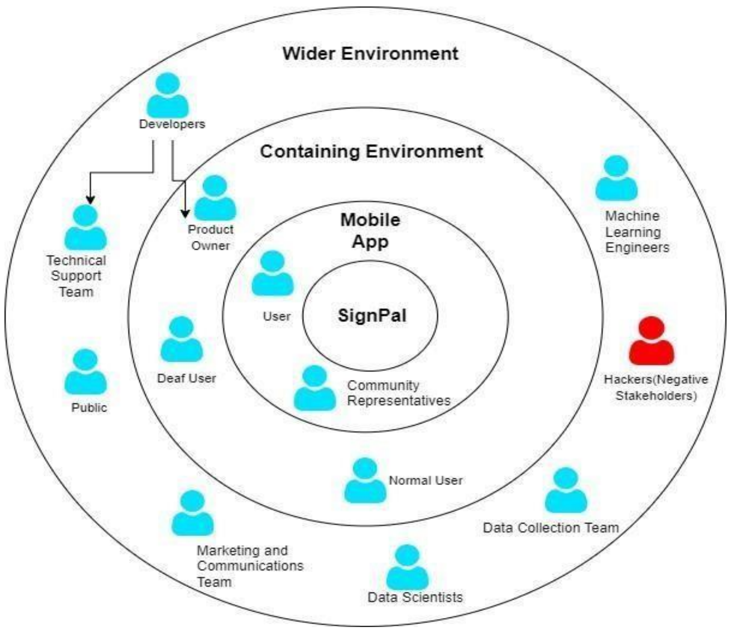


Figure 4. 2: Onion Model Diagram

4.3.2 Stakeholder Viewpoints

Stakeholder	Role	Description
Software Developers Machine Learning and Data Science Engineers Data Collection Team	Developer	Putting the system in place in accordance with its specifications. In charge of bug fixes and further system updates.
Product Owner	Product Owner	Serve as a liaison between the development team as well as stakeholders. In charge of defining the final product
Signers (Deaf Community) Non-Signers (Normal Community)	Users	The system's users, both in terms of simplifying their jobs and reaping the benefits of improved and effective system results.

System Management Operators	Operational Stakeholders	Keep the system running. Monitor system performance and notify relevant parties of any need.
Hackers	Negative Stakeholders	Who can access the system illegally and act in a malicious way.
Technical Support Team Marketing and Communications Team	Support	Provide necessary guidance for the user whenever needed.

Table 4. 1: Stakeholder Viewpoints

4.4 Selection of Requirement Elicitation Methodologies

Technique	**Justification for Selecting the approach**
Survey	The author conducted a survey with deaf school and public members to gather requirements and communicate with potential users, ensuring the project aligns with community expectations.
Literature Reviews	The author conducted a literature review on transfer learning techniques in image/video classification to identify motions in sign language. This assessment helped identify existing problems and constraints and propose solutions to improve results.
Self- Evaluation	The author regularly evaluated and prioritized requirements in addition to clearly describing the project's scope. This was a crucial procedure to follow in order to make sure the project stays on track and achieves its goals.
Brainstorming Sessions	Brainstorming sessions are also carried out through discussions with people who possess a variety of backgrounds, experiences, and topic skills. It was crucial to have a diversity of viewpoints among the

	participants in order to enable a more in-depth examination of potential solutions and factors.

Table 4. 2: Selection of Requirement Elicitation Methodologies

4.5 Findings

4.5.1 Findings from Survey

The author presents here the key survey findings that are most crucial for the verification and for defining the project needs. You may find the full survey form questionnaire in **Appendix C – Survey Questions**.

<table>
<tr><td>Question 01</td><td>How familiar are you with Sri Lankan Sign Language? (Rate 1 to 5)</td></tr>
<tr><td>Aim of the Question</td><td>To understand how familiar the society with Sri Lankan Sign Language</td></tr>
<tr><td colspan="2">

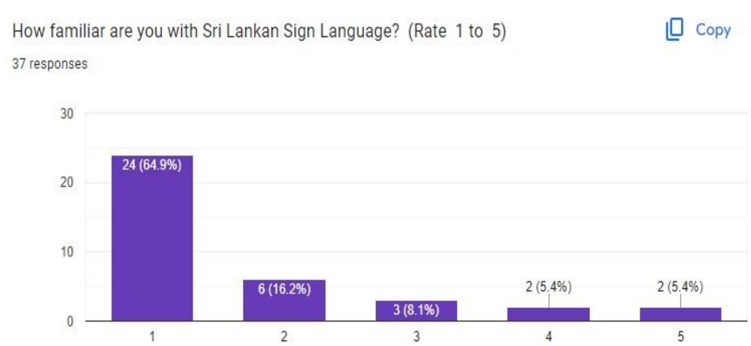


Figure 4. 3: Survey Finding Question 1

Observation

Most people in the community are not familiar with Sri Lankan Sign Language.

</td></tr>
<tr><td>Conclusion</td><td>Most people in the community are not familiar with Sri Lankan Sign Language.

This shows the use of providing a Sri Lankan Sign Language mobile application with real-time motion recognition to the community.</td></tr>
</table>

Table 4. 3: Survey Finding Question 1

Question 02	Do you think there is a need for an SLSL mobile application?

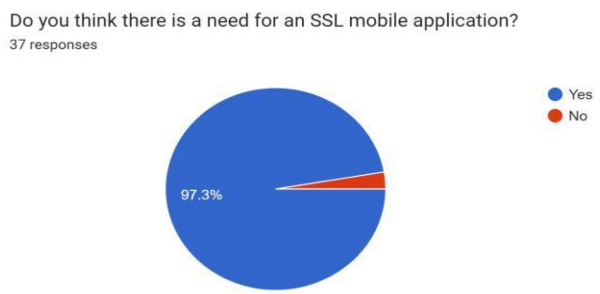
Aim of the Question	To understand the need for a Sign Language mobile application in the community.
Observation	 4: *Figure 4. 4: Survey Finding Question 2* 97.3% from the community is in need of a Sign Language Translation Mobile Application
Conclusion	This shows the use of providing a Sri Lankan Sign Language mobile application with real-time motion recognition to the community. .

Table 4. 4: Survey Finding Question 2

Question 03	Currently how do you communicate with those that don't sign?
Aim of the Question	To understand how deaf community communicates with people who don't sign.

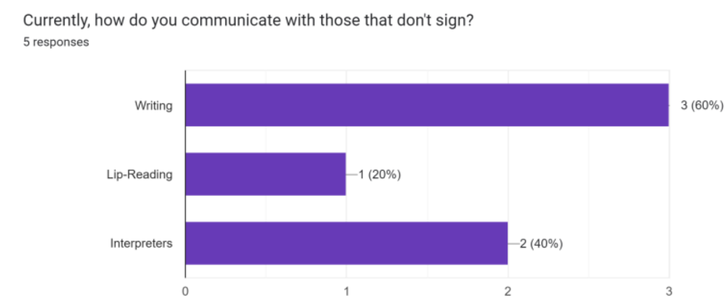


Figure 4. 5: Survey Finding Question 3

Observation

Writing is the most used method to communicate with non-signers by the deaf community. Also, a significant amount of deaf community uses Lipreading and interpreters as well.

Conclusion	These 3 ways are the only communication methods deaf community uses when they communicate with the normal community in their day today life and that's where Sri Lankan Sign Language mobile application with real time motion recognition uses. This app has the feature text to sign language which input text converts to sign language skeleton performance.

Table 4. 5: Survey Finding Question 3

Question 04	How would a realtime Sign Language translation mobile app benefit you personally or professionally?
Aim of the Question	To understand how familiar the community is with realtime sign language detection translators

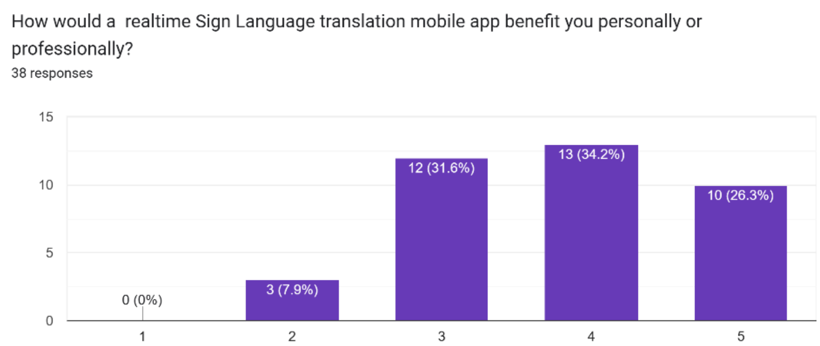


Figure 4. 6: Survey Finding Question 4

Observation Most of the community thinks it will be a benefit to use a real-time sign language translation mobile app.	
Conclusion	The community is prepared to embrace the mobile app that provides real-time sign language translation. When offering new mobile applications to the community, they should be simpler, easier to use, and less complicated. However, this demonstrates the value of giving users access to a mobile app.

Table 4. 6: Survey Finding Question 4

4.5.2 Findings from Literature Review

Finding	Citation
[Accurate and Light-weight Sign Language Translation Techniques:] To efficiently use depth information for sign language translation, we create a set of methods (such as a human gesture emphasis module, a lightweight 3DCNNbased translation model, and data augmentation). In comparison to the RGB based model, our system demonstrates that the depth information is useful to, i) Improving the accuracy of the sign language translation when compared to using RGB datasets, and ii) Demonstrating improved robustness under a variety of lighting conditions (up to about 40% in dark environments and about 10% with typical indoor lighting). Although the depth information can be blended with other modalities to provide an even greater translation accuracy, particularly when significant networking and computing resources are available, our work focuses on investigating the potential of using it as the only sensing modality.	(Park, 2021)

Table 4. 7: Findings from Literature Review

4.5.3 Findings from Self Evaluation

Criteria	Finding
Project Scope Defining	The project's goals are clearly stated, making it easy to grasp what the initiative is trying to accomplish. By clearly defining what is and is not included, the scope aids in keeping the project on track.

Well defined deliverables	The project's deliverables, which should include particular features, functionalities, or results, should be clearly stated.
Traceability Matrix	To make sure that every demand contributes to the goals, a traceability matrix is created that links requirements to project objectives.
Feasibility Assessment	The viability of carrying out the specified criteria while taking operational, financial, and technical aspects into account.

Table 4. 8: Findings from Self Evaluation

4.5.4 Findings from Brainstorming

Criteria	**Finding**
Inclusion of Sign Language Professionals	Experts in sign language offer insightful contributions to the project that are helpful for the CNN-LSTM model's detection process.
Constant system enhancement during the development phase	In order to promote ongoing improvement and enhance the capabilities and features of the mobile app, participants are urged to offer feedback on the brainstorming process iteratively. by letting later meetings expand on and improve the concepts developed in earlier ones.
Recording Concepts and Understandings	To act as a guide and reference for the project's later stages, the concepts and insights produced during the meetings are recorded.

Table 4. 9: Findings from Brainstorming

4.6 Summary of Findings

ID	Finding	Literature Review	Self-Evaluation	Survey	Brainstorming
1	Identified research gaps in sign language mobile application with realtime motion recognition and giving outputs in an interpretable way	X		X	X
2	The use and the preference of deaf and normal community of a Sri Lankan Sign Language Mobile App with real time motion detection.		X	X	X
3	The interpretability of real-time motion recognition is crucial, particularly for non-technical people who need to understand the model's predictions.	X	X	X	X
4	Experts in the subject area who lack technical expertise ought to feel at ease utilizing the mobile application.	X		X	X
5	Utilizing popular programming languages that support machine learning models.	X		X	X
6	Should have a user friendly and clear UI for the mobile app.		X		X

Table 4. 10: Summary of Findings

4.7 Context Diagram

A simplified visual depiction of a system's interactions with outside entities is called a context diagram. It offers a high-level summary of the system's limitations and relationships with the outside environment. The figure illustrates how outside parties use a smartphone app that uses real-time Sign Language. External entities, including developers, hearing users, and deaf users, are represented in the environment by their interactions with the program, which is portrayed as a central entity. Without a thorough technical design, a context diagram's goal is to assist stakeholders in rapidly identifying the system's scope and important interactions.

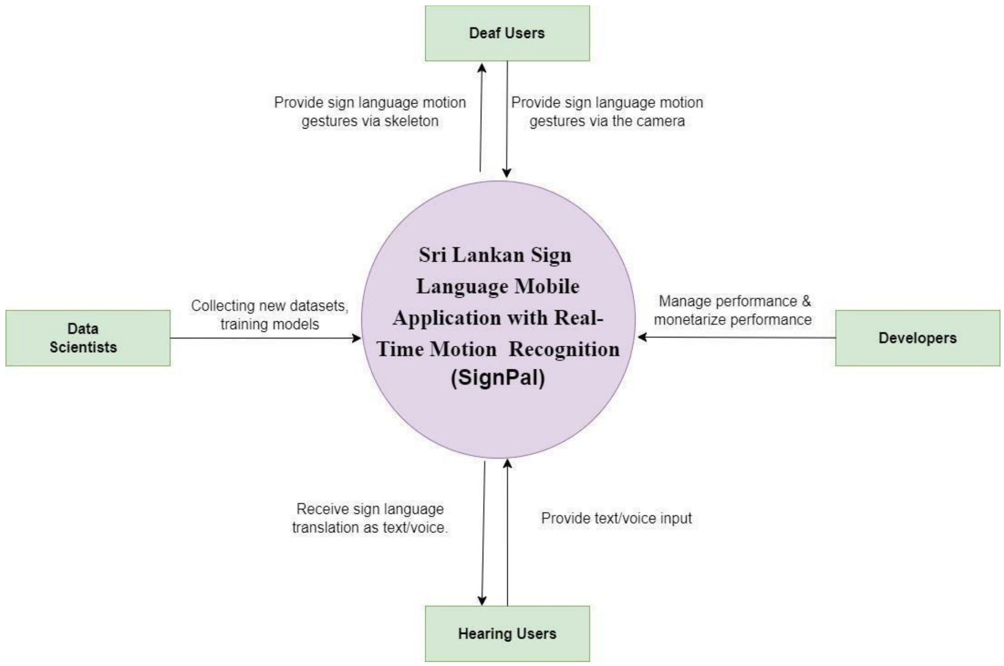


Figure 4. 7: Context Diagram

4.8 Use Case Diagram and Descriptions

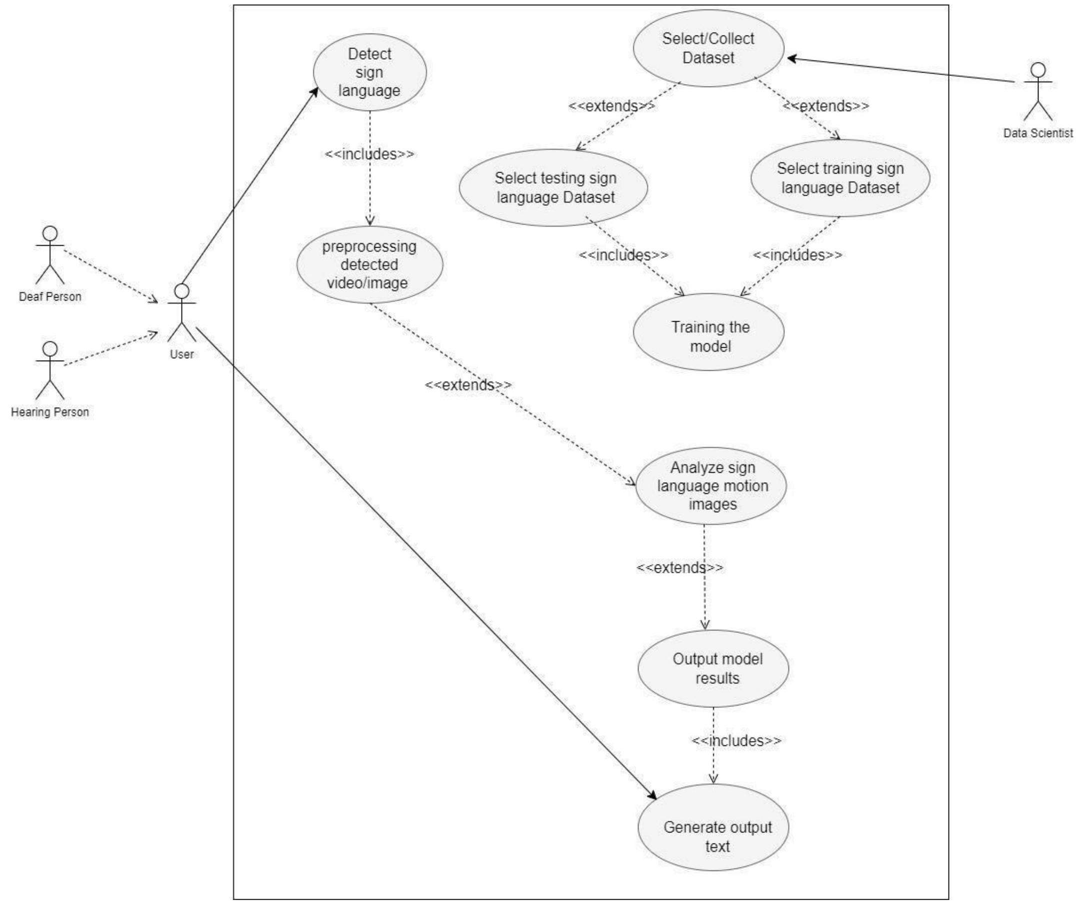


Figure 4. 8: Use Case Diagram

4.9 Use Case Description

Use Case Name	UC 1: Detect Sign Language
Description	The system records video or still photos of the user using sign language, preprocesses the information as needed, and then uses the trained model to identify the sign language.
Participating Actors	User (Deaf Person)

Preconditions	The user is in front of the camera, and the model is trained using a dataset of sign language.
Postconditions	A sign language dataset is used to train the model, and the user is in front of the camera.
Main Flow	1. Pre-processed sign language video frames are added to the CNN-LSTM model to analyze the process. 2. Start analyzing the frames and identifying its features. 3. Get the model predictions as its output.
Alternative Flows	None
Exceptional Flows	Errors due to training of the model. Display error message. Ask system admins for a new proper input.

Table 4. 11: Use Case Description 1

Use Case Name	UC 2: Generate Output Text
Description	The user sees sign language once it has been detected and translated into text.

Participating Actors	User (Hearing Person).
Preconditions	The sign language detection process is complete, and there are no errors in image/frames processing.
Postconditions	The text output is generated and shown to the hearing person.
Main Flow	4. Pre-processed sign language video frames are added to the CNNLSTM model to analyze the process. 5. Start analyzing the frames and identifying its features. 6. Get the model predictions as its output.
Alternative Flows	None
Exceptional Flows	Errors due to training of the model. Display error message. Ask system admins for a new proper input.

Table 4. 12: Use Case Description 2

4.10 Requirements

The suggested system's functional and non-functional requirements are described, along with the MoSCoW prioritization technique used to classify and rank these requirements.

M - Must Have	Requirements that are critical for success and to get the expected result of the project.
S - Should Have	Requirements that are important but not indispensable to the project's success.
C - Could Have	Requirements that are desirable but not necessary for the project's success.
W - Won't Have	Requirements that are explicitly not desirable to the project scope at its current development phase.

4.10.1 Functional Requirements

Table 4. 13: Functional Requirements

FR ID	**Requirement**	**Priority Level**
FR01	Users should able to open sign to text screen with camera on.	M
FR02	The user should be able to detect human poses effectively.	M
FR03	Analyzing the sign language pose images using CNNLSTM and identifying relevant signs in the images to predict the word or text.	M
FR04	The mobile app should generate a word or text and give the skeleton output that performed sign language	M
FR05	Sign language learning guide page with videos.	S

4.10.2 Non-Functional Requirements

NFR ID	Title and Description	Priority Level (MoSCoW)
NFR1	Accuracy	M
NFR2	Speed	M
NFR3	Reliability	M
NFR4	Security	C
NFR5	Usability	S
NFR6	Scalability	C

Table 4. 14: Non-Functional Requirements

4.11 Chapter Summary

This chapter covered role description, thorough stakeholder identification, and efficient requirement collection techniques. A context diagram that shows the boundaries and interactions of the system is included in the chapter itself. A use case diagram and thorough use case descriptions were also supplied, offering a thorough explanation of the system's main features. The chapter ends by gathering functional and non-functional needs and defining them in accordance with the MoSCoW convention. This serves as a roadmap for the project's upcoming phases.

5. SOCIAL, LEGAL, ETHICAL AND PROFESSIONAL ISSUES (SLEP)

5.1 Chapter overview

This chapter has looked at the project's possible results as well as the ethical standards that must be adhered to during the study process. plays a significant role in monitoring research ethics, as the BSC Code of Conduct states.

5.2 SLEP issues and mitigation

Issue	Description
Social issues	No personal information about the research participants was provided in the thesis because they all requested anonymity in order to express their perspectives.
Legal issues	Users' real-time identified videos won't be saved once detection is complete. All the open-source licenses apply to the chosen and used frameworks and technologies.
Ethical issues	Survey participants and other requirement gathering processors are provided with sufficient details regarding the study being done, as well as an explanation of their participation. The basic theory is unique, even in the face of plagiarism. Every citation and reference to the sources listed in the thesis is correct. Academic integrity concerns, such as plagiarism and falsification, were taken into account.
Professional issues	The project's limits were thoroughly described in the thesis, and during the feedback gathering phase, they were prominently underlined and underscored for the reviewers' attention. Throughout the project's implementation, best coding techniques, academic standards, and industry standards were applied.

Table 5. 1: SLEP issues and mitigation

5.3 Chapter summary

The social, legal, ethical, and professional difficulties associated with this endeavor were thoroughly examined in this chapter, which also described the author's plans to mitigate any possible hazards.

6. DESIGN

6.1 Chapter Overview

The first system architecture and design for the SLSL mobile app (Signpal) are covered in this chapter. The development of a suitable mobile app prototype is also discussed in this chapter, as it may be able to help the system achieve its intended goals. In order to create an ideal system architecture, the chapter also includes low-level design diagrams such as the data flow diagram, sequence diagram, algorithm design diagram, UI design diagram, and process flow chart diagram.

6.2 Design Goals

Design goals	Description
Accuracy	The project's top priority is real-time motion sign language recognition accuracy. For a mobile application to be useful, it must ensure that the results it generates are extremely accurate. This is especially true when the application supports sign language translation.
Correctness	It is crucial that the approaches and algorithms used are accurate. The project makes sure that while converting sign language into text format, the mobile app does it with a high degree of accuracy and dependability.
Performance	One essential component that may lead to optimizing the mobile application's speed and responsiveness is efficiency. This could provide prompt and effective outcomes, enhancing user experience and facilitating prompt decision-making all of which are critical for sign language translation in real time.
Usability	A crucial design objective is usability, which focuses on developing user-friendly interfaces and workflows for mobile applications. Ensuring user navigation and interaction with mobile applications is of utmost importance.

Scalability	The ability to handle the increasing volume of data and user demands is referred to as scalability. Additionally, the mobile application was designed to be easily scaled up to meet growing demand without sacrificing accuracy or performance.

Table 6. 1: Design Goals

6.3 System Architecture Design

6.3.1 Layard / Tiered Architecture Diagram

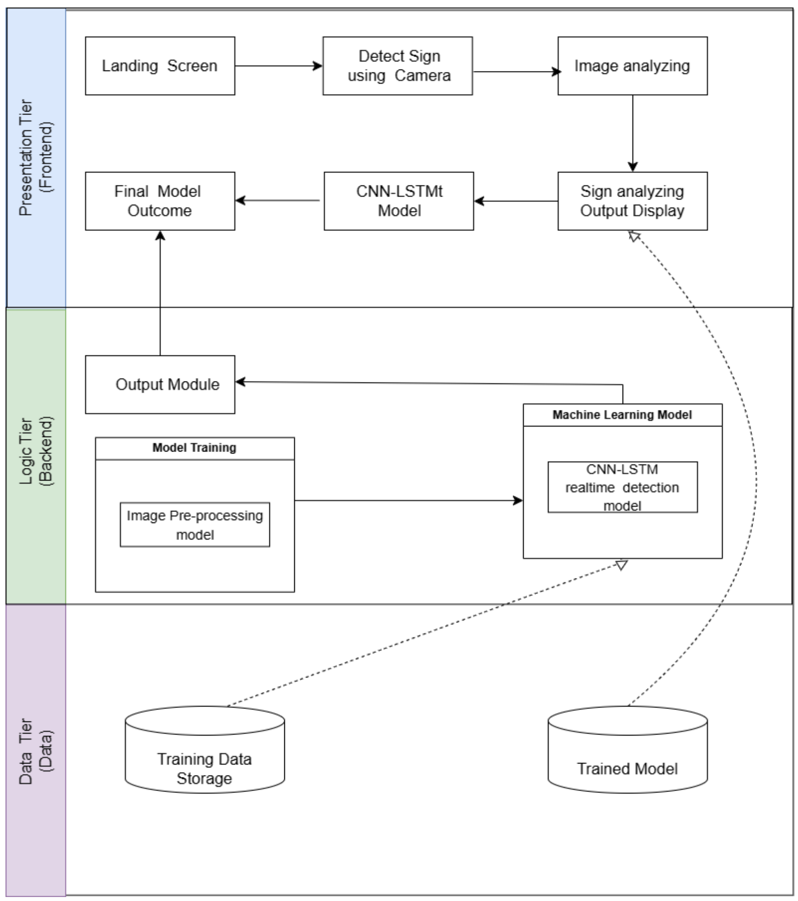


Figure 6. 1: Layered/Tiered Architecture Diagram

6.3.2 Tiered Architecture Discussion

1 Presentation Tier

The Sri Lankan Sign Language mobile application's presentation tier functions as its user-facing interface, offering a simple and intuitive experience. It consists of the following parts,

- **Homepage -** The first screen users see when they activate an application is the home page. It offers four primary buttons for navigating to the app's key features:
- **Text to Sign -** Users can enter text on the "Text to Sign" page, and the system will convert it into real-time sign language motions.
- **Sign to Text** - This feature opens a page where users can do sign language using their live camera. The program recognizes and instantly converts the sign language into text.
- **Text to Speech** - This tool helps with smooth communication by translating user-input text into spoken speech.
- **Learn Sign** - Directs users(non-deaf) to a page with interactive video lessons and resources for learning sign language.
- **Quiz Game** – Helps non-deaf users to learn and question their knowledge about sign language through a Quiz game.

2 Application Tier

The backend processing layer of the Sri Lankan Sign Language mobile application is called the Application Tier. It manages all computational tasks, such as preparing data, running the model, and sending the finished products to the presentation layer. The Application Tier is composed of the following elements,

- **Module for Data Preprocessing -** In charge of using MediaPipe to process raw video input and extract key features (hand, face, and position landmarks). Transforms user-uploaded video frames or live video streams into numerical landmark data that may be saved in npy format. Makes sure the data is formatted correctly so the CNN-LSTM model can analyze it.
- **Module for CNN-LSTM -** To predict sign language motions, the trained CNN-LSTM model examines the preprocessed landmark data. Enables precise sign-to-text or text-to-sign translation by classifying motions in real time. Contains features for effectively carrying out predictions and loading the trained model.

3 Data Tier

All necessary datasets, models, and application outputs must be stored, managed, and maintained by the Data Tier of the Sri Lankan Sign Language mobile app. It consists of the following parts,

- **Training Data Storage** – A location where all the datasets used to train the CNN-LSTM model is kept.
- **Trained Model Storage**

  Where the refined CNN-LSTM model is stored once it has been trained. The stored model is tuned for precise sign-to-text translations and real-time motion detection. This guarantees that the model can be readily retrieved for incorporation into the application for tasks involving inference.

6.4 Detailed Design

6.4.1 Choice of Design Paradigm

Object-Oriented Analysis and Design Methodology (OODAM) and Structure System Analysis and Design (SSADM) are the two primary design paradigms that can be applied to system design. Because of its modular structure, reusability, and support for multiple algorithms, the author has selected SSADM as the design paradigm for the Sri Lankan Sign Language mobile application, taking into account the programming paradigms and technologies that are utilized in the suggested system. In order to ensure flexibility to new datasets or changing requirements, SSADM makes it easier to systematically break down capabilities like gesture detection, translation, and learning modules into separate components. From pre-processing sign language markers to providing real-time translation outputs, the sequential approach enhances the app's productivity. The app's capabilities for effective sign language learning and translation are enhanced by the MediaPipe integration and Python-based backend.

6.4.2 Component Diagram

6.4.2.1 Level 1 Data Flow Diagram

Because the author uses the Structured Systems Analysis and Design Methodology (SSADM) as the design paradigm, the system's processes are represented using a Data Flow Diagram (DFD). The main elements of the real-time motion detection system are described in the Level 1 DFD.

Real-time video input from the user is captured, motion is analyzed using MediaPipe to identify hand, facial, and body landmarks, and the data is processed using a CNN-LSTM model to predict related sign language gestures. These are the main procedures. For two-way communication, the predictions are subsequently converted into text or shown as a skeleton-based animation. In order to facilitate future use and reference for ongoing development, the system additionally saves motion data and translation outcomes in databases. For both developers and consumers, this methodical approach improves interpretability and guarantees a clear breakdown of the system's functioning.

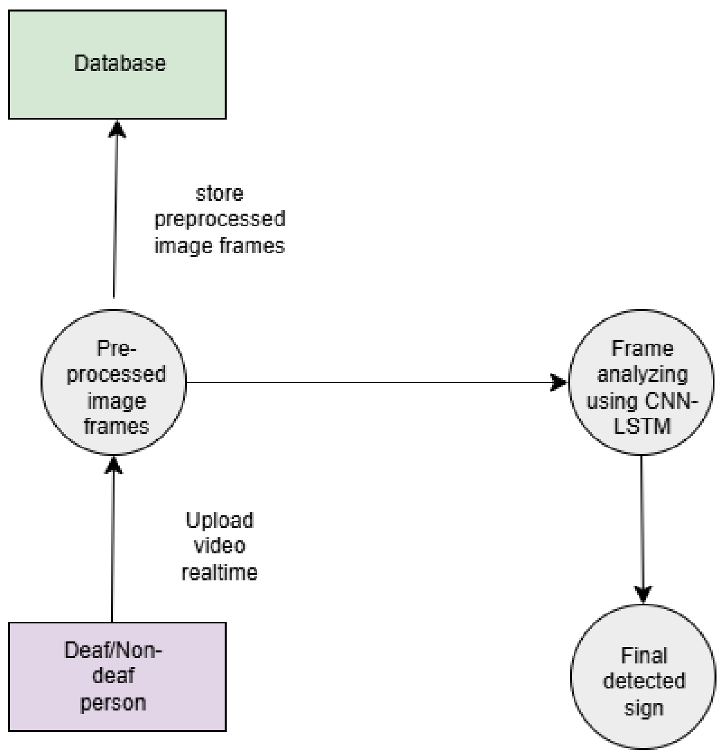


Figure 6. 2: Level 1 Data Flow Diagram

6.4.2.2 Level 2 Data Flow Diagram

The Level 2 Data Flow Diagram provides a more detailed breakdown of the system's processes for real-time SLSL motion detection and translation. The user initiates the system by performing gestures in front of the camera, which captures real-time video input. The system validates the input by ensuring the captured frames are compatible with gesture recognition processing. The validated input is then passed to MediaPipe's motion detection module, where key landmarks of hands, face, and body are identified and extracted. These features are sent to the CNN-LSTM model, which analyzes the spatial and temporal patterns to predict the corresponding sign or text.

The predictions are further processed to ensure accuracy and displayed to the user as text or a skeleton-based animation, facilitating bidirectional communication. The system also saves the processed data and results in a database for future improvements and analysis, ensuring scalability and continuous enhancement of the application's capabilities.

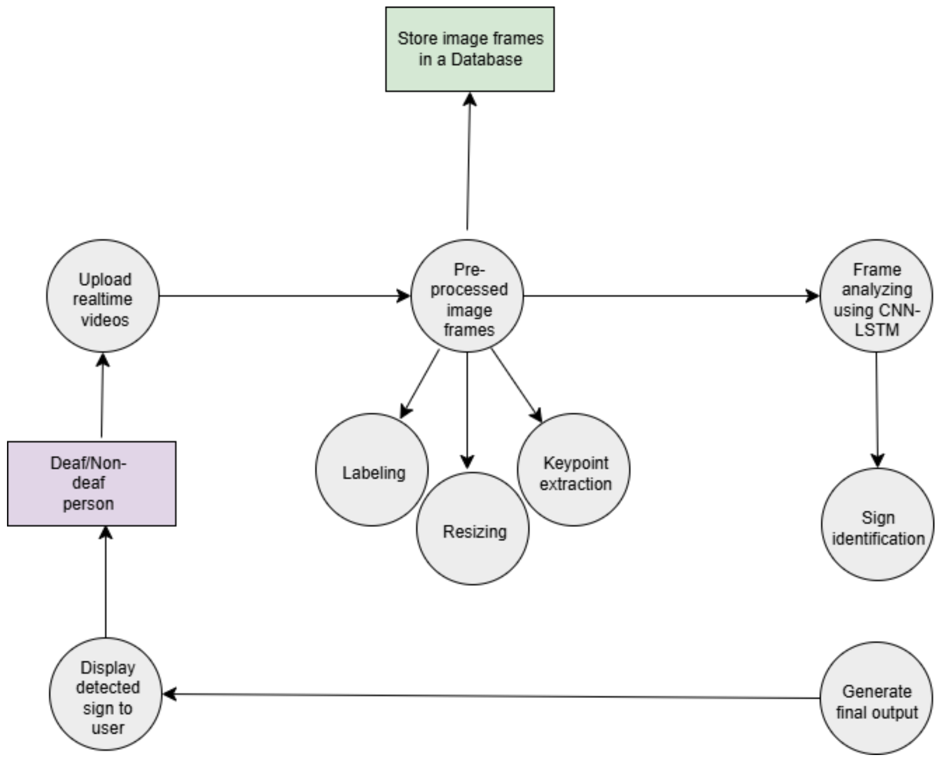


Figure 6. 3: Level 2 Data Flow Diagram

6.4.3 Algorithm Design

This model uses MediaPipe, OpenCV, and TensorFlow to build a real-time sign language recognition system. It detects and displays body landmarks using Matplotlib, NumPy, and Python tools. The dataset is produced by recording 7-actions, 30-frame video sequences, identifying important details, and creating labeled datasets. Prob_viz is used to illustrate the TensorFlow model architecture's predictions after it has been trained across 2000 epochs. Mechanisms for identifying user input, anticipating movements, and creating cohesive phrases are also included in the script. The goal of this approach is to help the deaf community communicate more effectively.

6.4.4 UI Design

A low-fidelity prototype diagram provides a basic depiction of the suggested application's user interface, basic user experiences, and mobile app features. The fidelity diagrams are in **Appendix D – Low Fidelity UI Frame Diagrams**.

6.4.5 System Process Workflow

Organized processes make up the system process flow for the real-time motion detection project in Sri Lankan Sign Language. First, real-time video input of users making SLSL motions is provided by live camera feeds. By processing this information, MediaPipe can recognize dynamic motion by identifying and tracking important hand, facial, and body landmarks. To forecast the relevant sign or word, the pre-processed data is then fed into a CNN-LSTM model, which extracts spatial and temporal information. Lastly, the application ensures smooth communication for users by displaying the translated text or sign output in real-time. Modern technologies are integrated into this organized process to provide a solution that is both effective and easy to use.

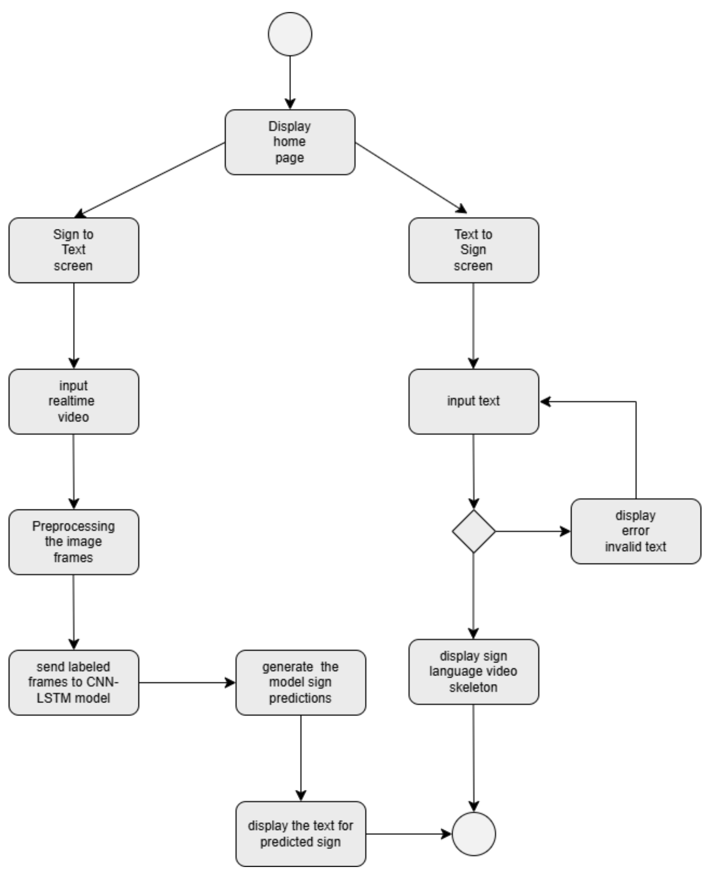


Figure 6. 4: System Process Workflow Diagram

6.5 Chapter Summary

The chapter concluded with an overview of the essential elements of the suggested mobile application. The development phases will have a strong base thanks to the well-articulated design goals. With an emphasis on scalability and adaptability, the tiered architectural diagram and its thorough explanation provide a thorough understanding of the mobile app's structural organization. The low-level system design concentrates on the particulars of every module to guarantee a clear and effective implementation. A deeper knowledge of the project's deliverables is made possible by the addition of fidelity diagrams, system process flow charts, and comprehensive design diagrams, which further improve and clarify the mobile application's design.

7. IMPLEMENTATION

7.1 Chapter Overview

The author will provide a thorough overview of the study prototype's basic implementation in this chapter. The tools, programming languages, and technologies that make up the core of the development process will be explained by the author.

7.2 Technology Selection

7.2.1 Technology Stack

The technical stack that the author will employ across all project development tiers is depicted in the following diagram.

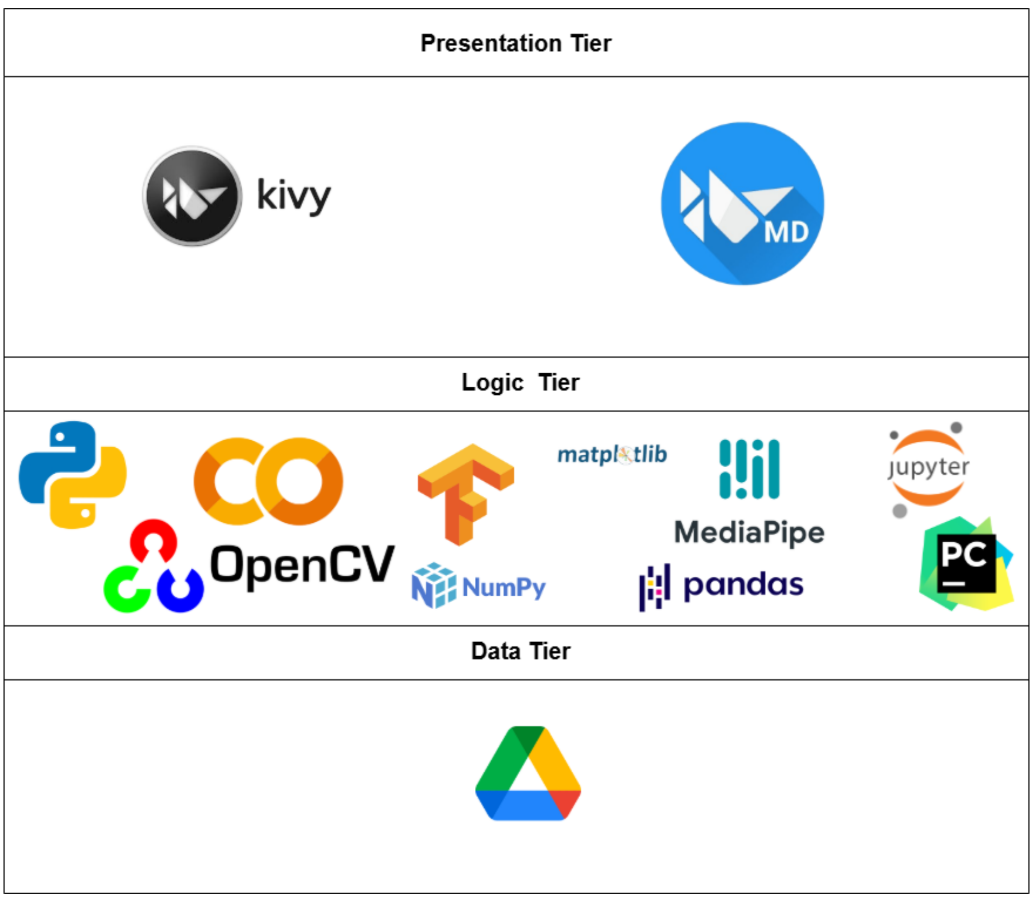


Figure 7. 1: Technology Stack

7.2.2 Frontend Technology Stack

KivyMD, a framework for making visually appealing and responsive mobile application interfaces, is used to construct the project's front end. Using KivyMD preserves the flexibility needed for developing mobile applications while guaranteeing consistency in UI/UX design. Python is used to implement the frontend, which easily integrates with middle-tier and backend technologies. With its strong support for Python and Kivy-based applications, PyCharm is used as the Integrated Development Environment (IDE) for front-end development. Google Collab is used in the development of the APK for Android device deployment, allowing for a resource-efficient and collaborative process to complete the program.

7.2.3 Middle-tier technology stack

The project's middleware layer integrates several reliable technologies to manage real-time processing and prediction activities. Key landmarks like hands, faces, and bodies may be identified and tracked in real-time thanks to MediaPipe's motion detection feature. This is enhanced with OpenCV, which offers frame capture and video input preprocessing capabilities. Accurate predictions are made by analyzing the spatial and temporal properties of movements using a CNN-LSTM model. PyCharm is the main development environment, while Python is the fundamental programming language for combining these elements. The middleware supports the application's real-time functionality by facilitating seamless communication between the frontend and backend.

7.2.4 Backend Technology Stack

Python is used in the development of the backend to control data processing, storage, and machine learning model interaction. Lightweight and effective data management is ensured by using npy files as the main storage type for landmark data. To prepare data for model predictions, OpenCV was used for frame preprocessing and video input handling. User inputs from the camera stream were quickly analyzed and converted into text outputs thanks to the backend architecture's optimization for real-time interactions. This backend stack guarantees the application's reliability, scalability, and ease of maintenance while also offering a strong basis for future improvements. The backend is built in PyCharm for local development.

7.2.5 Development Framework

Framework	Description
Tensorflow	Offers a strong foundation for putting Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNNs) into practice. enhances Keras's potential by integrating with it seamlessly. Python-compatible, matching the primary language used for the project.

Table 7. 1: Development Framework

7.2.6 Programming Language

Taking into account the benefits of utilizing **Python** and the project's context, the author chose to make it the project's programming language.

Advantage	Description
Cross-Platform Compatibility	Python code can run on a variety of operating systems since it is platform-independent. Python guarantees consistency in execution, which lowers deployment complexity whether you're developing on Windows, Linux, or macOS.
Rich Ecosystem of Libraries and Frameworks	Python provides a wealth of libraries for backend development and machine learning. Model training is made easier by libraries like TensorFlow, PyTorch, and Scikit-learn, while backend programming is made easier by Flask, FastAPI, and Delphi.
Ease of Learning and Readability	Python's straightforward syntax facilitates code writing and comprehension.This saves developers time on difficult syntax and enables them to rapidly prototype machine learning models and implement backend services.

Table 7. 2: Programming Language

7.2.7 Libraries and Tools

Library / Tool	Description
Mediapipe	Provides and tracks down the face, hands and pose keyframes to detect the sign gestures in skeleton form.
NumPy	allows for effective numerical calculations for applications involving machine learning also saves dataset in .npy format.
Matplotlib & Seaborn	Make educational visualizations to examine data and comprehend the functioning of the model such as graphs.
Pandas	Helps with the pre-processing of data for the creation of models.
Scikit-learn	provides a range of benchmarking machine learning algorithms.
Google Drive	Offers an easily accessible place to store datasets.
OpenCV	A library for image processing and computer vision is called OpenCV (Open Source Computer Vision). It has extensive features for jobs including object recognition, feature detection, and image enhancement.

Table 7. 3: Libraries and Tools

7.2.8 Frameworks

Framework	Description
KivyMD	Kivy can help with the rapid creation of applications. Code organization: Compared to other frameworks, Kivy makes code authoring more systematic and accurate. Clarity: The features are as obvious as Python.

Table 7. 4: Frameworks

7.2.9 IDEs

The most widely used IDEs were chosen by the author to construct each individual application component.

IDE	Description
Jupyter Notebook	A robust IDE made especially for Python programming. It has features that increase coding efficiency and preserve excellent code quality, such as intelligent code completion and sophisticated debugging.
Pycharm	Jupyter Notebooks, a popular component, enabling users to run code interactively and create code visualizations as well as to train models.

Table 7. 5: IDEs

7.2.10 Summary of Technology Selection

Component	Tool
Programming Language	Python
Development Framework	Tensorflow
Libraries	OpenCV,NumPy,Matplotlib,Scipy,,Mediapipe,Pandas,Matplotlib, Seaborn, Scikit-learn,
UI Framework	KivyMD
IDE s	PyCharm,Jupyter Notebook

Table 7. 6: Summary of Technology Selection

7.3 Implementation of Core Functionalities

7.3.1 Loading, Labelling, and Splitting the Dataset

The load data function was used to load the dataset from the local directories and associated data labels. Load, label, and partition a machine learning dataset to get it ready for training. The steps involved in loading are importing raw data into memory, assigning target labels to it, and dividing it into training and testing sets. This guarantees prediction accuracy and consistency. These procedures guarantee a structured dataset for efficient assessment and training.

```
from sklearn.model_selection import train_test_split
from tensorflow.keras.utils import to_categorical
```

```
label_map = {label:num for num, label in enumerate(actions)}
```

```
label_map
```

```
{'Afternoon': 0,
 'Ayubowan': 1,
 'Good': 2,
 'Month': 3,
 'Morning': 4,
 'Night': 5,
 'Noon': 6,
 'Evening': 7}
```

```
# sequences, labels = [], []
# for action in actions:
#     for sequence in np.array(os.listdir(os.path.join(DATA_PATH, action))).astype(int):
#         window = []
#         for frame_num in range(sequence_length):
#             res = np.load(os.path.join(DATA_PATH, action, str(sequence), "{}.npy".format(frame_num)))
#             window.append(res)
#         sequences.append(window)
#         labels.append(label_map[action])
sequences, labels = [], []
for action in actions:
    for sequence in range(no_sequences):
        window = []  # This line should be properly aligned with the inner for-loop
        for frame_num in range(sequence_length):
            res = np.load(os.path.join(DATA_PATH, action, str(sequence), "{}.npy".format(frame_num)))
            window.append(res)
        sequences.append(window)
        labels.append(label_map[action])
```

Figure 7. 2: Loading, Labelling, and Splitting the Dataset

7.3.2 Dataset Collection

Using MediaPipe's solutions, real-time video recordings of motion, posture, and facial landmarks are part of the dataset gathering process. After processing, the data is saved as numerical features and the coordinates of important landmarks are extracted. For the purpose of recognizing sign language gestures, the data is labeled. A rich, varied, and well-structured dataset is guaranteed for precise model training thanks to the process' efficiency and interactivity. To facilitate easy retrieval for preprocessing and model training, the gathered samples are kept in an organized directory in .npy format.

```python
cap = cv2.VideoCapture(0)
# Set mediapipe model
with mp_holistic.Holistic(min_detection_confidence=0.5, min_tracking_confidence=0.5) as holistic:

    # NEW LOOP
    # Loop through actions
    for action in actions:
        # Loop through sequences aka videos
        for sequence in range(no_sequences):
            # Loop through video length aka sequence length
            for frame_num in range(sequence_length):

                # Read feed
                ret, frame = cap.read()

                # Make detections
                image, results = mediapipe_detection(frame, holistic)

                # Draw landmarks
                draw_styled_landmarks(image, results)

                # NEW Apply wait logic
                if frame_num == 0:
                    cv2.putText(image, 'STARTING COLLECTION', (120,200),
                               cv2.FONT_HERSHEY_SIMPLEX, 1, (0,255, 0), 4, cv2.LINE_AA)
                    cv2.putText(image, 'Collecting frames for {} Video Number {}'.format(action, sequence), (15,20),
                               cv2.FONT_HERSHEY_SIMPLEX,0.8, (0, 225, 0), 2, cv2.LINE_AA)

                    # Resize and show the image
                    image = cv2.resize(image, (1000,800))
                    # Show to screen
                    cv2.imshow('SignPal', image)
                    cv2.waitKey(2000)
                else:
                    cv2.putText(image, 'Collecting frames for {} Video Number {}'.format(action, sequence), (15,20),
                               cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 225, 0), 2, cv2.LINE_AA)

                    # Resize and show the image
                    image = cv2.resize(image, (1000,800))
                    # Show to screen
                    cv2.imshow('SignPal', image)

                # NEW Export keypoints
                keypoints = extract_keypoints(results)
                npy_path = os.path.join(DATA_PATH, action, str(sequence), str(frame_num))
                # # Ensure directory exists
                # os.makedirs(os.path.dirname(npy_path), exist_ok=True)
```

```python
                # Save keypoints
                np.save(npy_path, keypoints)

                # Break gracefully
                if cv2.waitKey(10) & 0xFF == ord('q'):
                    break

    cap.release()
    cv2.destroyAllWindows()
```

Figure 7. 3: Dataset Collection

7.3.3 Splitting the dataset as training and testing sets

A machine learning dataset must be loaded, labeled, and divided to be ready for training. It must first be imported into memory, labeled with goal labels, and then divided into training and testing sets to load data. Prediction accuracy and consistency are thus guaranteed. Training typically uses

80% of the data, with testing using the remaining 20%. Following these procedures guarantees a well-structured dataset for efficient training and assessment.

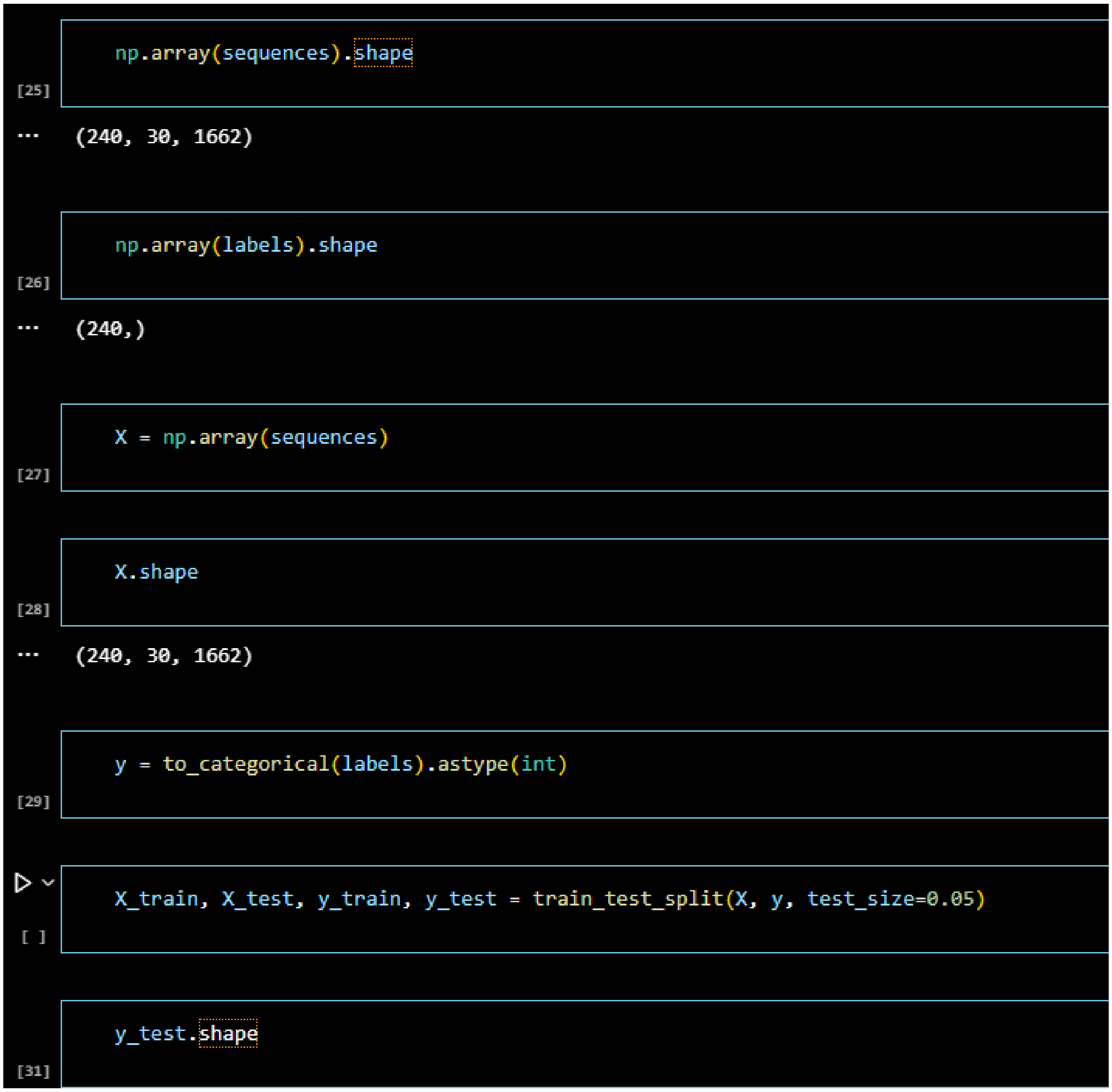


Figure 7. 4: Splitting the dataset as training and testing

7.3.4 Pre-processing

One of the most important steps in getting raw data ready for machine learning models is pre-processing. Outliers must be identified, duplicates must be eliminated, missing values must be handled, features must be scaled, and categorical labels must be encoded into numerical values. This procedure guarantees that learning is founded on significant and consistent patterns, increases model performance, and improves the quality of the dataset.

```
label_map = {label:num for num, label in enumerate(actions)}
```

```
label_map
```

```
{'Afternoon': 0,
 'Ayubowan': 1,
 'Good': 2,
 'Month': 3,
 'Morning': 4,
 'Night': 5,
 'Noon': 6,
 'Evening': 7}
```

```
sequences, labels = [], []
for action in actions:
    for sequence in range(no_sequences):
        window = []  # This line should be properly aligned with the inner for-loop
        for frame_num in range(sequence_length):
            res = np.load(os.path.join(DATA_PATH, action, str(sequence), "{}.npy".format(frame_num)))
            window.append(res)
        sequences.append(window)
        labels.append(label_map[action])
```

```
def extract_keypoints(results):
    pose = np.array([[res.x, res.y, res.z, res.visibility] for res in results.pose_landmarks.landmark]).flatten() if results.pose_landmarks else np.zeros(33*4)
    face = np.array([[res.x, res.y, res.z] for res in results.face_landmarks.landmark]).flatten() if results.face_landmarks else np.zeros(468*3)
    lh = np.array([[res.x, res.y, res.z] for res in results.left_hand_landmarks.landmark]).flatten() if results.left_hand_landmarks else np.zeros(21*3)
    rh = np.array([[res.x, res.y, res.z] for res in results.right_hand_landmarks.landmark]).flatten() if results.right_hand_landmarks else np.zeros(21*3)
    return np.concatenate([pose, face, lh, rh])
✓ 0.0s
```

Figure 7. 5: Preprocessing

7.3.5 Model Training

Pre-processing the dataset, normalizing body landmark features, dividing the dataset into training and testing sets, and applying a CNN-LSTM are all steps in the model training process. The accuracy of the model's iterative learning of sign language movements is monitored throughout ten simulated epochs. After training, the model is saved as a .h5 file for real-time gesture recognition and tested on the test set.

```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from tensorflow.keras.callbacks import TensorBoard
```

```python
log_dir = os.path.join('Logs')
tb_callback = TensorBoard(log_dir=log_dir)
```

```python
model = Sequential()
model.add(LSTM(64, return_sequences=True, activation='relu', input_shape=(30,1662)))
model.add(LSTM(128, return_sequences=True, activation='relu'))
model.add(LSTM(64, return_sequences=False, activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(actions.shape[0], activation='softmax'))
```

```python
model.compile(optimizer='Adam', loss='categorical_crossentropy', me
```

```python
model.fit(X_train, y_train, epochs=2000, callbacks=[tb_callback])
```

Figure 7. 6: Model Training

7.4 User Interface

Your real-time sign language detection smartphone app's user interface is made to be simple and easy to use, meeting the needs of both functional communication and learning. Five separate buttons are displayed on the home page**: Learn Sign, Play Quiz, Text to Sign, Text to Voice, and Sign to Text**. These choices take users to distinct interactive sites with special features.

Through the use of the camera, the Sign to Text Page makes it possible to recognize movements in real-time and translate them into legible text for smooth communication. To improve comprehension of particular movements, users can enter text on the Text to Sign Page and see a dynamic skeleton making the appropriate signs. By producing a speech output from user-input text, the Text-to-speech Page makes communication accessible by providing auditory feedback. For new learners, the Learn Sign Page offers instructional resources in the form of video tutorials for teaching Sri Lankan Sign Language. Users can guess the correct sign meanings from video clips in this entertaining and engaging game on the Play Quiz Page. A motivational message is shown beside the final score. For both the hearing and the deaf communities, this well-designed interface creates a comprehensive platform that facilitates everyday communication and learning by smoothly integrating educational and practical components.The fidelity diagrams are in **Appendix E – Screenshots of the UI**.

7.5 Chapter Summary

The technical stack for implementing the suggested research prototype, including programming languages and other tools and technologies used, was covered in this chapter. The chapter included a detailed exploration of the project's essential features, and the best technical stack was selected.

8. TESTING

8.1 Chapter Overview

This chapter describes the testing process that was carried out throughout the project to guarantee that the end product satisfies the anticipated system outcomes of the study. The chapter will assess the functional and non-functional characteristics, testing objectives, and testing approaches. Lastly, this chapter also includes the evaluation of the benchmark testing.

8.2 Objectives and Goals of Testing

Verifying that the suggested system satisfies the anticipated essential product requirements is the main goal of the testing phase. This ensures that the generated model and the system will meet the necessary expectations of the research, maintaining the quality and effectiveness of the final product outcome. Some of the system's main objectives are,

- Confirm that the algorithm for real-time sign language detection operates correctly and produces the desired outcomes.
- Confirm that the CNN-LSTM model offers satisfactory model insights.
- Check to see if the mobile application's user interface offers improved user experience.
- Confirm that all functional and non-functional requirements are met by the system.
- Confirm that the code complies with the best practices and quality requirements.
- Find any mistakes, glitches, or flaws that were overlooked during the development process.

8.3 Testing Criteria

Reducing the discrepancy between predicted and observed results is the primary goal of testing. It is a crucial stage in the software development process that guarantees developers have access to the tools, processes, and environments they need to create reliable software. Functional and non-functional quality assessments are the two main approaches used in this testing.

- **Functional Testing**: Functional specification aims to verify that the system complies with the suggested system design and achieves the intended results and objectives.
- **Non-functional Testing**: Non-functional testing is done to see if the system complies with performance standards, efficiency benchmarks, and established norms.

8.4 Model Testing (for ML projects)

The trained model's ability to recognize and identify SLSL is assessed through a model testing procedure. The F1 Score, Confusion Matrix, Training, and Validation Loss Curve, and other widely used assessment matrices were utilized by the author in

In order to evaluate the model's performance, which might have ensured that the model performed accurately when tested on both visible and unknown input.

8.4.1 Training and Validation Loss Curve

In order to obtain the most generalized model for unknown data, the training validation loss curves show the model's learning process, search for indications of overfitting or underfitting, and identify the ideal epoch to halt training.

8.4.1.1 Training and Validation Loss Curve for Sign Prediction

The category accuracy and loss trends for the sign language detection model across roughly 900 epochs are depicted in the training graphs that are supplied. The accuracy in the first graph, which represents epoch_categorical_accuracy, starts out low and progressively rises to a value of about 0.9 near the conclusion. Effective learning is indicated by the generally favorable trend, although minor oscillations. As training goes on, the second graph, epoch loss, shows a more progressive fall in loss after a steep initial decline that indicates quick learning in the early stages. Although the overall downward trajectory validates the model's better performance with time, the presence of variations implies unpredictability in model convergence. With rising accuracy and falling loss, these findings imply that the model is learning efficiently, while some fine-tuning would still be required to stabilize learning.

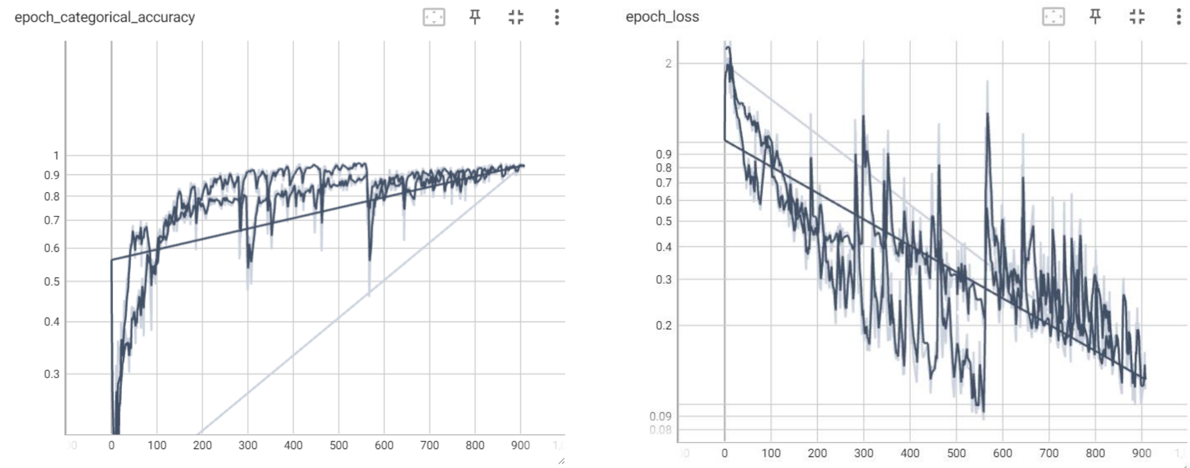


Figure 8. 1: Training and Validation Loss Curve

8.4.2 Confusion Matrix

By contrasting real and expected labels, the given confusion matrix graphically illustrates how well a classification model is performing. Whereas off-diagonal elements signify incorrect classifications, diagonal elements show accurate predictions. In this instance, the majority of predictions match the actual labels; the "Evening" class had the best accuracy, with three accurate predictions. Some misclassifications do exist, though, like the two cases where class "3" was predicted rather than "Afternoon." The blue shading's intensity corresponds to the frequency of predictions; larger counts are indicated by darker hues. Although the model performs rather well in classification overall, it could do better at distinguishing between particular classes. The model performs well in spite of these small misclassifications, but additional analysis utilizing metrics like accuracy, recall, and F1-score may yield a more thorough review.

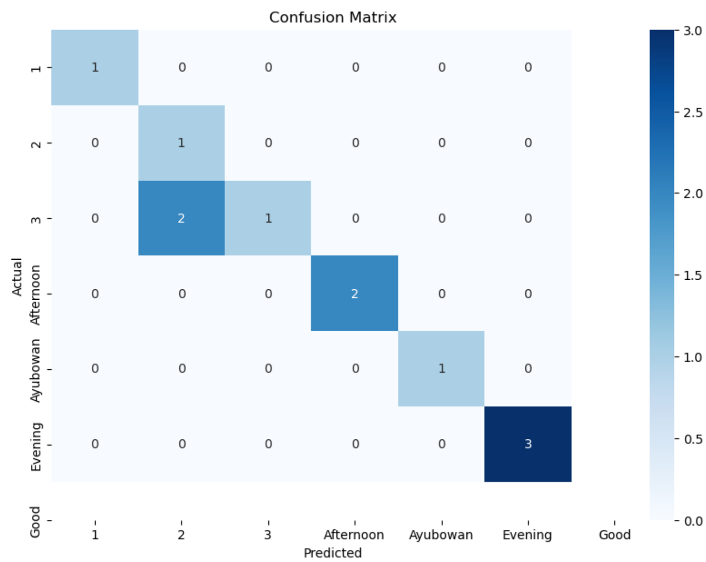


Figure 8. 2: Confusion Matrix

8.4.3 Classification report

Key performance indicators, including as precision, recall, and F1-score for every class, are included in the classification report. Due to a lack of data support, some courses, including class 1 and class 2, exhibit inferior recall and precision even while some classes receive perfect scores (1.00). Although the overall accuracy of 82% indicates respectable performance, the model's capacity to generalize is hampered by the class imbalance. These disparities are highlighted by the macro and weighted averages, which show that some classes predominate in the predictions. Performance can be enhanced by addressing this problem with strategies like class weighting or data augmentation.

```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         1
           1       0.33      1.00      0.50         1
           2       1.00      0.33      0.50         3
           3       1.00      1.00      1.00         2
           4       1.00      1.00      1.00         1
           6       1.00      1.00      1.00         3

    accuracy                           0.82        11
   macro avg       0.89      0.89      0.83        11
weighted avg       0.94      0.82      0.82        11
```

Figure 8. 3 :Classification report

8.4.3.1 Accuracy

Measures the overall proportion of correctly classified samples.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} = \frac{TP + TN}{TP + TN + FP + FN}$$

The model's learning progress over epochs is displayed via the accuracy curve. Effective learning is indicated by the curve's overall upward trend and high category accuracy (~90%). The accuracy discrepancies, however, point to either overfitting or batch-wise training changes. Abrupt declines could be a sign of learning instability, which could be brought on by difficult samples, changes in the dataset, or tweaks to the learning rate. Methods like learning rate scheduling, batch normalization, or larger dataset sizes could produce a more stable curve.

8.4.3.2 Precision

Precision quantifies how well optimistic predictions work. It is calculated as the ratio of genuine positives to the total of false positives and true positives.

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

The accuracy of the model is demonstrated by the extraordinarily high precision rates in the trained model for sign language detection, which are 1.00 for the 0,2,3,4 and 6 indexes. 0.33 for index 1.

8.4.3.3 Recall

Recall, also known as sensitivity or true positive rate, gauges how well the model can identify positive examples.

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

The trained model's recall for index 0, 2,3,4 and 6 is 1.00, while the recall for index 1 is 0.33 . The model's great accuracy in detecting real positive situations is demonstrated by these recall values.

8.4.3.4 F1-score

The F1-Score offers a single statistic to evaluate the balance between precision and recall since it is the harmonic mean of the two.

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

The strong detection capabilities of the proposed model are demonstrated by the F1-score of at least 0.5 for each type of sign index.

8.4.3.1 AUC/ROC Curve

A multiclass classification model's performance is shown by the ROC curve in the diagram. The Area Under the Curve (AUC) score measures the model's capacity to discriminate between positive and negative examples for each class, which is represented by each curve. The model's capacity to distinguish between classes is assessed by the ROC-AUC curve. The AUC values for the majority of classes are near 1.0, indicating excellent discrimination. However, the model's dependability for those particular gestures may be impacted by the presence of NaN for some

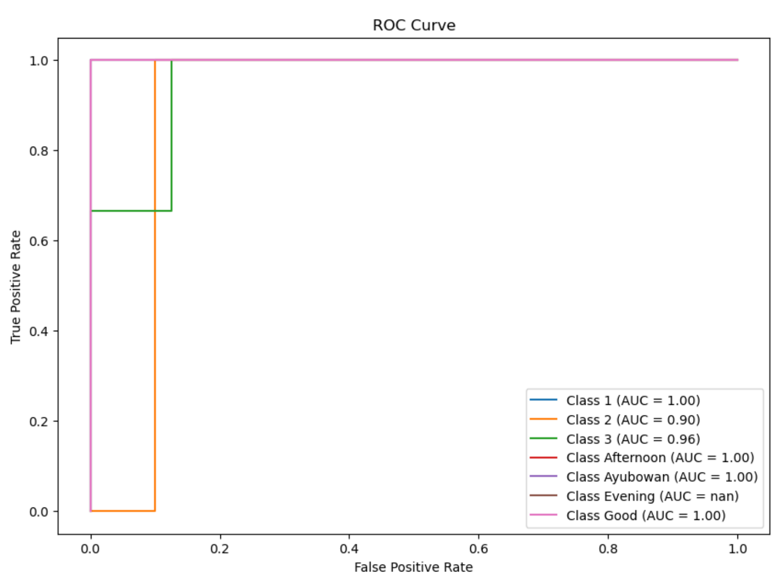


Figure 8. 4: AUC/ROC Curve

classes (such as "Evening"), which suggests missing or inadequate positive samples. A robust model is indicated by a well-defined ROC curve with few gaps; nonetheless, to enhance detection in practical applications, it is crucial to provide adequate data representation for every class.

8.5 Functional Testing

FR ID	User Action	Expected Results	Actual Result	Status
FR01	Users should be able to open sign to text screen with the camera on.	Users will be able to see the camera successfully	Users will be able to see the camera successfully	Passed
FR02	The user should be able to detect human poses effectively.	Users will be able to see landmarks and connection skeletons on the camera of a person	Users will be able to see landmarks and connection skeletons on the camera of a person	Passed
FR03	Analyzing the sign language pose videos using CNN-LSTM and identifying relevant signs in the videos to predict the word or text.	Detects sign language that is performed and gives the output text in real time	Detects sign language that is performed and gives the output text in real time	Passed
FR04	The mobile app should generate a word or text and give the skeleton output that performed sign language	Users enter a word/sentence and will be able to see a skeleton perform relevant sign languages	Users enter a word/sentence and will be able to see a skeleton perform relevant sign languages	Passed

FR05	Sign language learning guide page with videos.	This page will display sign language learning videos	This page will display sign language learning videos	Passed

Table 8. 1: Functional Testing

8.6 Module and Integration Testing

Module	Expected Result	Actual Result	Status
When User detects the deaf person using sign to text feature camera	Gives the detected sign language output in text format.	Provides the detected sign language output in text format below camera screen.	Passed
When a normal user needs to express something using text to sign feature	User enters a word or a sentence and after generating sign language must perform for the relevant word/sentence	User enters a word or a sentence and after generating sign language it performs a skeleton based gif for the relevant word/sentence	Passed
When a user needs to communicate with a mute person can use the text to speech feature	User enters a word or a sentence and after generating the relevant voice for the input text will be heard.	User enters a word or a sentence and after generating the relevant voice for the input text will be heard.	Passed

Table 8. 2: Module and Integration Testing

8.7 Non-Functional Testing

NFR ID	Requirement	Description	Priority Level (MoSCoW)	Status

NFR1	Accuracy	for easier communication between the deaf and the general public, the accuracy of the real-time sign language-detecting mobile app is essential. The system must be able to provide exact sign predictions that demonstrate the model's precision and the forecast's quality.	M	Implemented
NFR2	Performance	The performance of the real-time sign language detecting program was assessed through non-functional testing. Motion tracking and translation were handled by the system without any noticeable lag during testing for responsiveness and efficiency. Response time, resource consumption, and performance tests of the program showed difficulties in preserving high accuracy and maximizing computing efficiency.	M	Implemented
NFR3	User-friendly user interface	Both the deaf community and the general public with non-technical backgrounds utilize the system, so the user interface	M	Implemented

		needs to offer a straightforward and user-friendly experience.		
NFR4	Security	User data that pertains to private and personal criteria is not handled by the system.	C	Not Implemented

Table 8. 3: Non-Functional Testing

8.8 Limitations of Testing Process

The lack of high-quality datasets for Sri Lankan Sign Language (SLSL) gestures, computational difficulties processing dynamic motion, environmental factors like background clutter and lighting, and user variations in sign execution styles were all problems for the real-time sign language detection application. Extensive testing and performance and scalability optimization were limited by resource constraints. The processing requirements of CNN-LSTM models made it challenging to achieve minimal latency during real-time gesture-to-text translations. For better real-world performance, these constraints necessitate more optimization, extensive user testing, and dataset improvements.

8.9 Chapter Summary

The chapter explores a thorough assessment of the system. After outlining the testing criteria and objectives, it analyses the efficacy of the data science model using metrics like accuracy and F1 score, which are displayed using ROC curves and confusion matrices. It also talks about benchmarking, functional and non-functional tests, including security and performance, and admits the limitations of the testing procedure. Code snippets and illustrations are also used to accompany the chapter, which ends with a summary of the results.

9. EVALUATION

9.1 Chapter Overview

This chapter examines the input provided by domain experts and target users, concept, and prototype evaluators on the research, along with a thorough examination of each component's functionality and recommendations for enhancements. It incorporates self-evaluation to pinpoint areas of strength and improvement. This thorough analysis aids in identifying paths for next improvements.

9.2 Evaluation Methodology and Approach

The chapter's evaluation methodology combines qualitative and quantitative techniques to assess a project's performance. Quantitative analysis was used to evaluate the model's effectiveness, while qualitative analysis involved thematic analysis of comments. This approach provides insights into the project's wider effects, including research challenges, design, implementation, and user interface. Subject-matter experts' comments and focused questionnaires enhance the study.

9.3 Evaluation Criteria

Criteria	Evaluation Purpose
Selecting the concept and primary focus of the project	This criterion was used to confirm that the chosen domain and study concept were sufficiently tentative to be considered research.
Research Gap	The author has determined various limitations and their scope in the chosen research topic by consulting prior work in this field.
Proposed Solution	The identified gaps will be filled, and the problem domains will be solved by evaluating the proposed solution and the system functionality.
UI & UX	To confirm if the user experience and user interface have pertinent standards.

Contribution to the research domain	To evaluate the technological innovations introduced to the regions of the proposed system as well as the research contributions made to the field.

Table 9. 1: Evaluation Criteria

9.4 Self-Evaluation

The author's research, which was conducted based on the study themes, is given in this part. This self-evaluation was carried out in accordance with the previously mentioned evaluation criteria.

Criteria	Evaluation Purpose
Selecting the concept and primary focus of the project	The author is particularly interested in finding a way to bridge the communication gap between the deaf and the general public. After careful consideration, the author identified a few concepts that needed improvement.
Research Gap	A few of the drawbacks of Sinhala Sign Language (SLSL) translation technologies are their inability to recognize dynamic motion, their poor accuracy in real-time translation, and their inability to translate in both directions. These problems impact the general usability and adaptability of these applications by impeding communication for the deaf community and preventing smooth transitions between text, speech, and sign language modes.
Proposed Solution	It goes without saying that decisions made in the social arena must always be more precise and reliable. The mobile application would give precise results when it came to detecting signs, but it was never clear how these models arrived at their conclusions.

System architecture	The system architecture was developed using industry standards as well as the SignPal mobile application's appearance.
UI & UX	The user design approach for mobile applications has adopted industry standards user experience and interface.

Table 9. 2: Self-Evaluation

9.5 Selection of the Evaluators

The system's assessors were chosen based on their professional skills and particular research interests, with an emphasis on those with extensive expertise in the pertinent fields. The assessors fall under the following categories:

Evaluator	**Description**
Users	To test the system in the actual world, test users have been selected situations among deaf community and normal community
Domain Experts	The chosen domain expert works for globally renowned IT firms and as well as in Sign Language Teaching.

Table 9. 3: Selection of the Evaluators

9.6 Evaluation Result

Evaluator	**Area**	**Feedback**
Domain Experts	Concept	According to Domain experts, a number of new studies have been conducted utilizing a variety of technologies that are being launched in the social domain of sign language on a daily basis. Thus, the selected angel is has a novel approach to problem-solving.
	Solution	This approach can be successfully applied in the community since SignPal offers a dependable means of real-time bidirectional sign detection. This approach can help the deaf community and the general public communicate more easily.

End Users	Prototype Features	According to test users, the SignPal mobile application's capabilities are the most dependable and helpful for lowering communication barriers.
	Usability	According to users, this would be a very useful tool for public communication. They claimed that if the mobile application was available, they wouldn't have to deal with the difficulties they had previously had in comprehending and communicating.

Table 9. 4: Selection of the Evaluators

9.6.1 Domain Experts

Designation	Evaluation of Feedback on Concept	Evaluation Feedback on Solution
Mrs. Anusha Fernando The Principal The School for the Deaf 521, Galle Road, Ratmalana Sri Lanka	A more excellent concept for improving the communication barrier between the deaf and the normal community.	This is an excellent solution to one of the problems in the social sector.
Mr. Buddhika Gunathilaka Teacher The School for the Deaf 521, Galle Road, Ratmalana Sri Lanka	The concept of real-time sign language detection is a very promising strategy.	Making sign detection faster and adding and providing text output also providing text to sign features is a big step forward in the social sector.
Mrs. Malani Alexander Teacher The School for the Deaf 521, Galle Road,	Well approached	This mobile app is an excellent innovation for the communication barrier

Ratmalana Sri Lanka		between the deaf and the normal community.

Table 9. 5: Domain Experts

9.6.2 Technical Experts

Designation	Architecture of Solution	Implementation of the Solution
Mr. M. Kalana Pathum Silva Senior Software Engineer Mitra Innovation	A more excellent concept for improving the communication barrier between the deaf and the normal community.	This is an excellent solution to one of the problems in the social sector.
Mr. Uvindu Sri Dharmawardana Senior Site Reliability Engineer WSO2	An innovative software architecture and well-executed mobile app.	A mobile application that makes use of cutting-edge technologies
Mr. Nimesha Lakruwan Wijeratne Consultant Business Applications IFS	Excellent with the system's architectural layout. UI that is easy to use.	Well done on the planning. Nice use of deep learning.

Table 9. 6: Technical Experts

9.7 Limitations of Evaluation

Reviews were the primary method of gathering feedback on the prototype; however, due to hectic schedules, the majority of interviews were conducted both in-person and online. Since evaluators do not have the opportunity to physically examine the prototype, the virtual nature of these interviews makes it difficult to conduct a complete assessment of the project. Furthermore, the majority of the specialists knew very little about the developments in CNN-LSTM technology for real-time sign language identification.

9.8 Evaluation of Functional Requirements

FR ID	Title and Description	Priority Level	Status
FR01	Users should be able to open sign to text screen with the camera on.	M	Implemented
FR02	The user should be able to detect human poses effectively.	M	Implemented
FR03	Analyzing the sign language pose videos using CNN-LSTM and identifying relevant signs in the videos to predict the word or text.	M	Implemented
FR04	The mobile app should generate a word or text and give the skeleton output that performed sign language	M	Implemented
FR05	Sign language learning guide page with videos.	S	Implemented

Table 9. 7: Evaluation of Functional Requirements

9.9 Evaluation of Non-Functional Requirements

NFR ID	Title and Description	Priority Level	Status
NFR1	Accuracy	M	Implemented
NFR2	Security	C	Not- Implemented

NFR3	Reliability	M	Implemented
NFR4	Speed	C	Implemented
NFR5	Scalability	C	Not- Implemented
NFR6	Usability	S	Implemented

Table 9. 8: Evaluation of Non-Functional Requirements

9.10 Chapter Summary

This chapter provides a thorough review of the structure and evaluation technique used in this study, segmenting the project into discrete parts for in-depth examination. In comparison to these standards, the author offered a thoughtful self-evaluation, which was supported by insights obtained from technical and domain specialists' questionnaires and interviews. The input from the evaluators was methodically grouped into categories to enhance comprehension of the project's efficacy and impact. Together with graphic representations of the quantitative data analysis, the results of this theme study were carefully assembled.

10. CONCLUSION

10.1 Chapter Overview

The technical stack for implementing the suggested research prototype, including programming languages and other tools and technologies used, was covered in this chapter. The chapter included a detailed exploration of the project's essential features, and the best technical stack was selected.

10.2 Achievements of Research Aims & Objectives

By developing a mobile application that can translate Sri Lankan Sign Language (SLSL) in real-time, the project was able to accomplish its goals. By combining CNN-LSTM models with MediaPipe, dynamic motion detection was made easier, guaranteeing precise and instantaneous SLSL gesture identification. An important communication tool gap for the deaf community was filled with the introduction of bidirectional translation between text/voice and sign language. Additionally, the application's intuitive UI made it accessible to users with different degrees of technological expertise. The accuracy and usability assessments showed how well the program worked to bridge communication gaps for Sri Lanka's deaf community.

10.3 Utilization of Knowledge from the Course

Knowledge from several school modules, such as software engineering, machine learning, and human-computer interface, was heavily used in the project. The ideas of accessible user interface design, real-time motion detection with MediaPipe, and CNN and LSTM for gesture recognition were all successfully implemented. In order to ensure that the system satisfied user needs while staying within project schedules, the Agile methodology was also used to oversee iterative development and integrate user feedback. All of these abilities worked together to make the application's development and implementation successful.

10.4 Use of Existing Skills

This project offered a fantastic chance to put the abilities acquired during the course of the academic program to use. The backend for sign language recognition was built mostly using Python programming, a fundamental skill learned in coursework. The CNN-LSTM architecture for gesture categorization was designed and trained with the use of knowledge of machine learning, particularly deep learning principles. The interface was constructed and integrated with the real-time model output using ideas from mobile application development. Additionally, managing the development workflow and guaranteeing modular and legible code were made possible by familiarity with frameworks like TensorFlow and Keras, as well as version control procedures like Git.

10.5 Use of New Skills

Throughout this investigation, several new technical capabilities that went beyond the course curriculum were established. Learning how to translate trained models into formats like TensorFlow for on-device inference was necessary for integrating computer vision models into a mobile context. An expertise in effective image pipeline building was necessary for real-time frame capture and pre-processing using OpenCV and MediaPipe. Additionally, a new issue that improved knowledge of hybrid neural network designs was synchronizing the flow between LSTM-based temporal sequence learning and CNN-based spatial feature extraction. Lastly, one crucial ability acquired via trial and error and incremental improvement was performance optimization for mobile devices, which included lowering model size and latency.

10.6 Achievement of Learning Outcomes

The project effectively fulfilled the academic program's learning objectives. To create a workable, real-time system that could recognize hand gestures used in sign language, theoretical knowledge had to be applied. Strong analytical and problem-solving abilities were demonstrated during the implementation, especially in creating a smooth model deployment pipeline. The phases of planning, frequent testing, and thorough documentation demonstrated, time management, and technical communication. The author's proficiency with an end-to-end machine learning application is demonstrated by the iterative development process, which includes everything from dataset creation to mobile integration.

10.7 Limitations of Research

A few issues were noted during testing, despite the system's promising real-time sign language detection findings. There is a need for improved generalization because the accuracy of gesture classification tends to decline in dimly lit areas or with crowded backgrounds. Another drawback was the challenge of differentiating movements with little visual variation, particularly when they were executed at different speeds. Furthermore, too-fast motions caused misclassification since the LSTM sequence recognition difficulty. These results imply that although the model works well in controlled circumstances, more training and more datasets could increase its robustness in a variety of real-world situations.

10.8 Achievement of the Contribution to Body of Knowledge

By showcasing a practical, mobile-based solution for real-time sign language detection, this project adds to the expanding field of assistive technologies. It confirms the effectiveness of CNN-LSTM hybrid architecture for portable devices' temporal gesture recognition. The study also sheds light on whether it is feasible to implement deep learning models for real-time applications in settings with limited resources. Additionally, the research establishes the foundation for upcoming developments in accessibility-focused AI solutions and gesture-based communication tools by tackling issues like effective on-device inference and gesture sequence modeling.

10.9 Problems and Challenges Faced

Throughout its development, the project ran into several difficulties. The scarcity of labeled SLSL datasets was a major problem, requiring the development of a bespoke dataset to properly train the models. Optimizing the CNN-LSTM model for real-time performance on mobile devices presented another difficulty, requiring precise changes to strike a compromise between computational efficiency and accuracy. Aligning disparate needs was also complicated by the need to collect and integrate varied user feedback from the hearing and deaf communities. Creative problem-solving techniques were used to overcome these obstacles, such as setting up iterative feedback sessions to improve the program and utilizing MediaPipe for effective tracking.

10.10 Deviations from the Original Plan

Although the project stayed true to its main goals, there were a few detours during the development phase. The original idea to use an existing dataset was changed to generate a bespoke dataset that guaranteed improved recognition accuracy and better depicted SLSL motions. To increase performance and dependability, some of the tools in the intended technological stack were also swapped out with more effective substitutes, like MediaPipe for real-time motion detection. To guarantee the overall quality of the application and its applicability to the intended user population, these deviations were warranted.

10.11 Future Enhancements

The project establishes a solid framework for upcoming improvements. The application's adaptability would be increased by adding more intricate gestures and regional SLSL dialects to the dataset. While improving the application for low-end devices would assure usability across varied user groups, adding other languages could increase its accessibility. Including cutting-edge AI models, like transformer-based topologies, could enhance functionality and accuracy even further. The focus is to expand the dataset for future work. Additionally, an animation avatar must be created for the text-to-sign feature and add Sinhala text-to-voice feature to make the app more user-friendly.

10.12 Concluding Remarks

In conclusion, the project demonstrates the potential of real-time motion detection and translation technology to enhance communication for the deaf community in Sri Lanka significantly. The continuing development's initial test results offer insightful information about the project's status, showing both its successes and its shortcomings. The system needs to be improved to be fine-tuned, problems must be resolved, and performance must be maximized. This chapter provides a clear roadmap for the project's future implementation while also summarizing its status.

REFERENCES

- Agrawal, T. and Urolagin, S. (2020). 2-way Arabic Sign Language Translator using CNNLSTM Architecture and NLP. *Proceedings of the 2020 2nd International Conference on Big Data Engineering and Technology*. 3 January 2020. Singapore China: ACM, 96–101. Available from https://doi.org/10.1145/3378904.3378915 [Accessed 3 November 2024].

- Akshit Tayade and Halder, A. (2021). Real-time Vernacular Sign Language Recognition using MediaPipe and Machine Learning. Available from https://doi.org/10.13140/RG.2.2.32364.03203 [Accessed 3 November 2024].

- Asthana, O. et al. (2022). Sign Language Recognition Based on Gesture Recognition/Holistic Features Recognition: A Review of Techniques. *2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM)*. 23 February 2022. Gautam Buddha Nagar, India: IEEE, 713–718. Available from https://doi.org/10.1109/ICIPTM54933.2022.9754140 [Accessed 3 November 2024].

- Attili, V.S. et al. (2024). Real-time Sign Language Translation using Computer Vision and Machine Learning. *2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC - ROBINS)*. 17 April 2024. Coimbatore, India: IEEE, 703–709. Available from https://doi.org/10.1109/ICCROBINS60238.2024.10533962 [Accessed 3 November 2024].

- Bhat, C. et al. (2023). Two-way Communication: An Integrated System for American Sign Language Recognition and Speech-to-Text Translation. *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. 6 July 2023. Delhi, India: IEEE, 1–7. Available from https://doi.org/10.1109/ICCCNT56998.2023.10307541 [Accessed 3 November 2024].

- Bora, J. et al. (2023). Real-time Assamese Sign Language Recognition using MediaPipe and Deep

Learning. *Procedia Computer Science*, 218, 1384–1393. Available from https://doi.org/10.1016/j.procs.2023.01.117.

- Choudhary, D.K., Singh, R. and Kamthania, D. (2021). Sign Language Recognition System. *SSRN Electronic Journal*. Available from https://doi.org/10.2139/ssrn.3832151 [Accessed 3 November 2024].

- Dewasurendra, D. et al. (2020). Emergency Communication Application for Speech and Hearing-Impaired Citizens. *2020 From Innovation to Impact (FITI)*. 15 December 2020. Colombo, Sri Lanka: IEEE, 1–6. Available from https://doi.org/10.1109/FITI52050.2020.9424899 [Accessed 3 November 2024].

- Fernando, P. and Wimalaratne, P. (2016). Sign Language Translation Approach to SinhaleseLanguage. *GSTF Journal on Computing (JoC)*, 5 (1), 9. Available from https://doi.org/10.7603/s40601-016-0009-8.

- Gamage, S. et al. (no date). A Multimodal Approach for Real-Time Sinhala Sign Language Translation.

- Gedaragoda, G.H.M.S.C. et al. (2023). “Hand Model” – A Static Sinhala Sign Language Translation Using Media-Pipe and SVM Compared with Hybrid Model of KNN, SVM and Random Forest Algorithms. *2023 5th International Conference on Advancements in Computing (ICAC)*. 7 December 2023. Colombo, Sri Lanka: IEEE, 77–82. Available from https://doi.org/10.1109/ICAC60630.2023.10417384 [Accessed 3 November 2024].

- Haputhanthri, H.H.S.N. et al. (2023). Multi-modal Deep Learning Approach to Improve Sentence level Sinhala Sign Language Recognition. *International Journal on Advances in ICT for Emerging Regions (ICTer)*, 16 (2), 21–30. Available from https://doi.org/10.4038/icter.v16i2.7264.

- Ivanska, L. and Korotyeyeva, T. (2022). Mobile real-time gesture detection application for sign language learning. *2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT)*. 10 November 2022. Lviv, Ukraine: IEEE, 511–514. Available from https://doi.org/10.1109/CSIT56902.2022.10000440 [Accessed 3 November 2024].

- Manoj Kumar, D. et al. (2020). EasyTalk: A Translator for Sri Lankan Sign Language using Machine Learning and Artificial Intelligence. *2020 2nd International Conference on Advancements in Computing (ICAC)*. 10 December 2020. Malabe, Sri Lanka: IEEE, 506– 511. Available from https://doi.org/10.1109/ICAC51239.2020.9357154 [Accessed 3 November 2024].

- Optimised Machine Learning-based Translation of Indian Sign Language to Text. (2023). *International Journal of Intelligent Engineering and Systems*, 16 (4), 398–408. Available from https://doi.org/10.22266/ijies2023.0831.32.

- Park, H., Lee, Y. and Ko, J. (2021). Enabling Real-time Sign Language Translation on Mobile Platforms with On-board Depth Cameras. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5 (2), 1–30. Available from https://doi.org/10.1145/3463498.

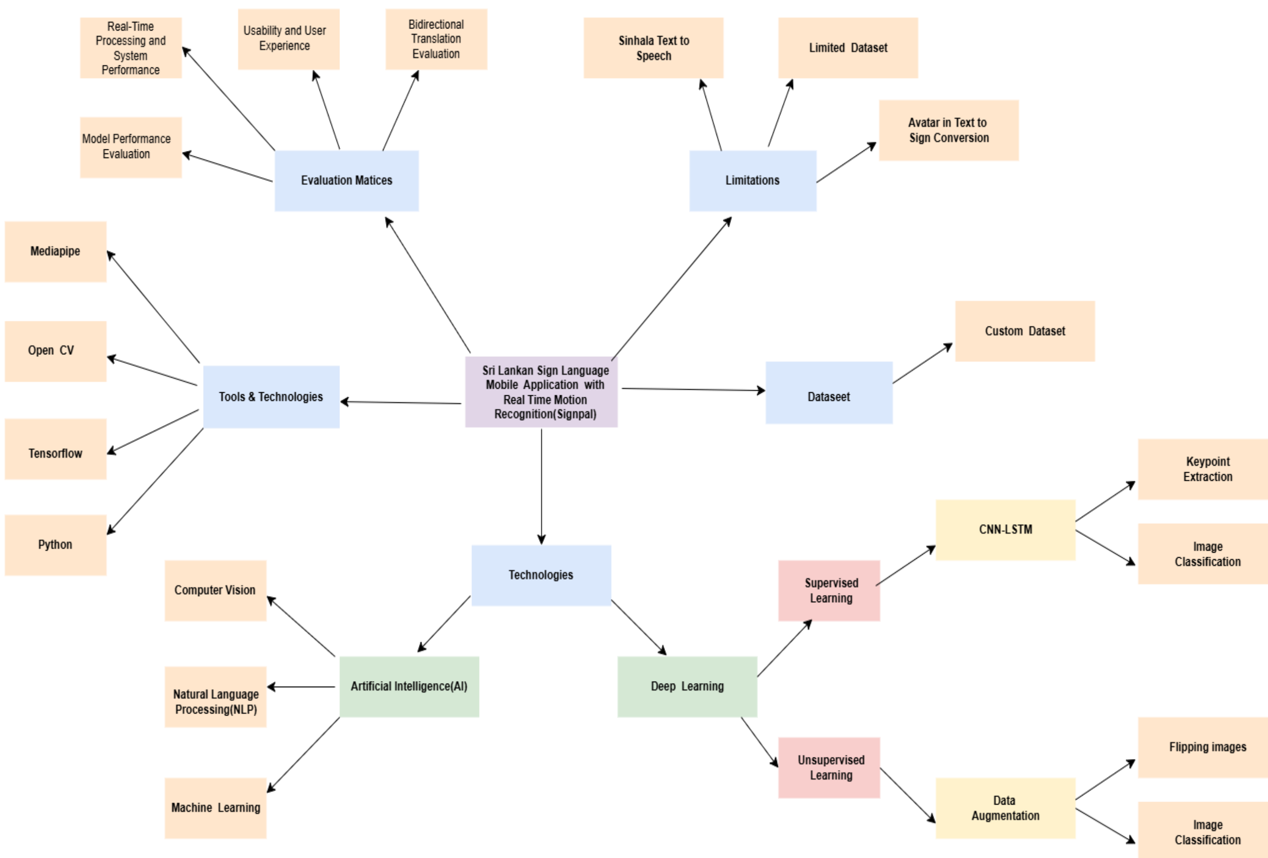
- Priyankara, M., Gunasekara, A. and Ilmini, K. (2023). Sign Language Translation Techniques Using Artificial Intelligence for the Hearing Impaired Community in Sri Lanka: A Review. *2023 7th SLAAI International Conference on Artificial Intelligence (SLAAI-ICAI)*. 23 November 2023. Kuliyapitiya, Sri Lanka: IEEE, 1–6. Available from https://doi.org/10.1109/SLAAIICAI59257.2023.10365012 [Accessed 3 November 2024].

- Punsara, K.K.T. et al. (2020). IoT Based Sign Language Recognition System. *2020 2nd International Conference on Advancements in Computing (ICAC)*. 10 December 2020. Malabe, Sri Lanka: IEEE, 162–167. Available from https://doi.org/10.1109/ICAC51239.2020.9357267 [Accessed 3 November 2024].

- Rishan, R.M., Jayalal, S. and Wijayasiriwardhane, T.K. (2022). Translation of Sri Lankan Sign Language to Sinhala Text: A Leap Motion Technology-based Approach. *2022 2nd International Conference on Advanced Research in Computing (ICARC)*. 23 February 2022. Belihuloya, Sri Lanka: IEEE, 218–223. Available from https://doi.org/10.1109/ICARC54489.2022.9754050 [Accessed 3 November 2024].

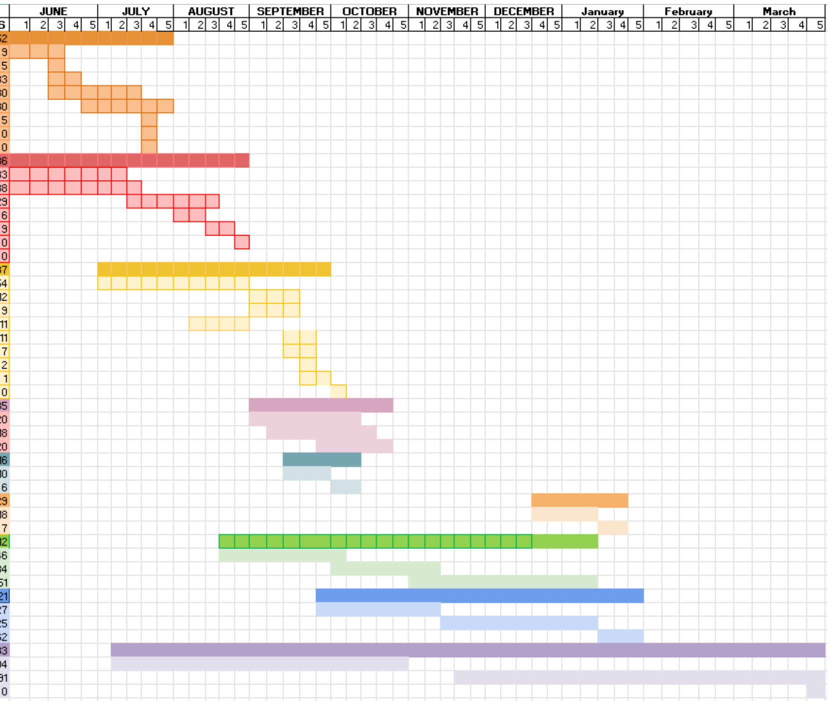
- Snehalatha, N. et al. (2024). Sign Language Detection Using Action Recognition LSTM Deep Learning Model. *2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON)*. 9 August 2024. Bengaluru, India: IEEE, 1–6. Available from https://doi.org/10.1109/NMITCON62075.2024.10699301 [Accessed 3 November 2024].

Appendix A – Concept Map

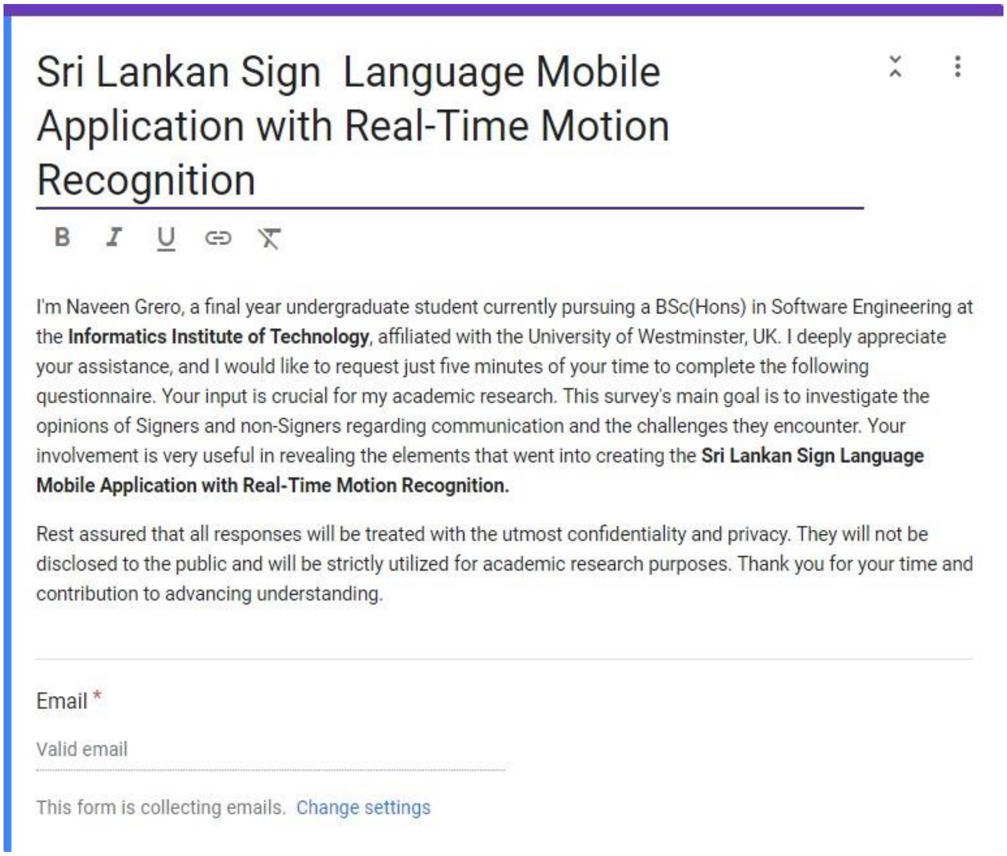


Appendix B - Gantt Chart

Start Date			
TASK	**START**	**END**	**TOTAL DAYS**
1. Project Initiation	6/1/2024	7/23/2024	52
1.1 Identify Research Domain, Research Gap	6/1/2024	6/10/2024	9
1.2 Approval of the gap	6/10/2024	6/15/2024	5
1.3 Reading and collecting Research papers	6/15/2024	7/18/2024	33
1.4 Get Domain Knowledge	6/15/2024	7/15/2024	30
1.5 Work on initial PP	6/30/2024	7/30/2024	30
1.6 Identifying aim,requirements and objectives	7/23/2024	7/28/2024	5
1.7 Mentor review on progress and initial PP	7/23/2024	7/23/2024	0
1.8 Submitting Initial PP	7/31/2024	7/31/2024	0
2. Literature Review	6/5/2024	8/30/2024	86
2.1 Gathering Research papers	6/7/2024	7/10/2024	33
2.2 Reading and summerizing the research papers	6/7/2024	7/15/2024	38
2.3 Reviewing gathered research data	7/15/2024	8/13/2024	29
2.4 Categorize the LR componenets for documenta	8/3/2024	8/9/2024	6
2.5 Preparing draft LR	8/13/2024	8/22/2024	9
2.6 Mentor review on LR	8/23/2024	8/23/2024	0
2.7 LR submission	8/30/2024	8/30/2024	0
3. Requirement Gathering	7/5/2024	9/30/2024	87
3.1 Analyze prevailing procedures	7/5/2024	8/28/2024	54
3.2 Preparing questionnaires	9/1/2024	9/13/2024	12
3.3 Distributing questionnaires to target audiences	9/1/2024	9/10/2024	9
3.4 Conducting interviews	8/8/2024	8/19/2024	11
3.5 Collecting data from questionnaires and Intervie	8/9/2024	8/20/2024	11
3.6 Analyze and prepare recived data for the SRS	9/10/2024	9/17/2024	7
3.7 Mentor review on draft SRS	9/20/2024	9/22/2024	2
3.8 Self-evaluation on SRS	9/25/2024	9/26/2024	1
3.9 SRS submission	9/30/2024	9/30/2024	0
4. System Design	9/2/2024	10/7/2024	35
4.1 Create High Level Architecture	9/2/2024	9/22/2024	20
4.2 Create System Design Diagrams	9/10/2024	9/28/2024	18
4.3 Prepare the Document	9/17/2024	10/7/2024	20
5. Tool and Technology Selection	9/18/2024	10/4/2024	16
5.1 Compare and Contrast Tools	9/18/2024	9/28/2024	10
5.2 Select the suitable Tools	9/28/2024	10/4/2024	6
6. Interim Progress Report	12/22/2024	1/20/2025	29
6.1 Prepare the Document	12/22/2024	1/9/2025	18
6.2 Finalize and submit Document	1/13/2025	1/20/2025	7
7. Prototype and Implementation	8/20/2024	12/10/2024	112
7.1 Implementation the Prototype	8/20/2024	10/5/2024	46
7.2 Implement the Classification Model	10/6/2024	11/9/2024	34
7.3 Finalize the Prototype	10/20/2024	12/10/2024	51
8. Testing and Evaluation	9/28/2024	1/27/2025	121
8.1 Create Test Case	9/28/2024	10/25/2024	27
8.2 Excute test Cases	10/26/2024	11/20/2024	25
8.3 Prepare and Submit the Test Report	11/26/2024	1/27/2025	62
9. Documentation	7/8/2024	2/26/2025	233
9.1 Prepare thesis	7/8/2024	10/20/2024	104
9.2 Refine the Document	11/26/2024	2/25/2025	91
9.3 Submission of final thesis	2/26/2025	2/26/2025	0



Appendix C – Survey Questions



Sri Lankan Sign Language Mobile Application with Real-Time Motion Recognition

I'm Naveen Grero, a final year undergraduate student currently pursuing a BSc(Hons) in Software Engineering at the **Informatics Institute of Technology**, affiliated with the University of Westminster, UK. I deeply appreciate your assistance, and I would like to request just five minutes of your time to complete the following questionnaire. Your input is crucial for my academic research. This survey's main goal is to investigate the opinions of Signers and non-Signers regarding communication and the challenges they encounter. Your involvement is very useful in revealing the elements that went into creating the **Sri Lankan Sign Language Mobile Application with Real-Time Motion Recognition.**

Rest assured that all responses will be treated with the utmost confidentiality and privacy. They will not be disclosed to the public and will be strictly utilized for academic research purposes. Thank you for your time and contribution to advancing understanding.

Email *

Valid email

This form is collecting emails. Change settings

Who are you, a signer of Sri Lankan Sign Language (SSL) or a non-signer? *

- ◯ Signer
- ◯ Non-Signer

Which age group do you belong to? *

- ◯ Below 18
- ◯ 18 - 30
- ◯ 30 - 50
- ◯ Above 50

Which profession do you hold? *

Short answer text

In your daily life, how frequently do you come across obstacles to communication? (Rate 1 to 5) *

1	2	3	4	5
◯	◯	◯	◯	◯

What difficulties do you frequently encounter while speaking with non-signers/signers? *

Long answer text

Currently, how do you communicate with those that don't sign/sign? *

- [ ] Writing
- [ ] Lip-Reading
- [ ] Interpreters

Have you ever used a sign language translator mobile app? *

- ◯ Yes
- ◯ No

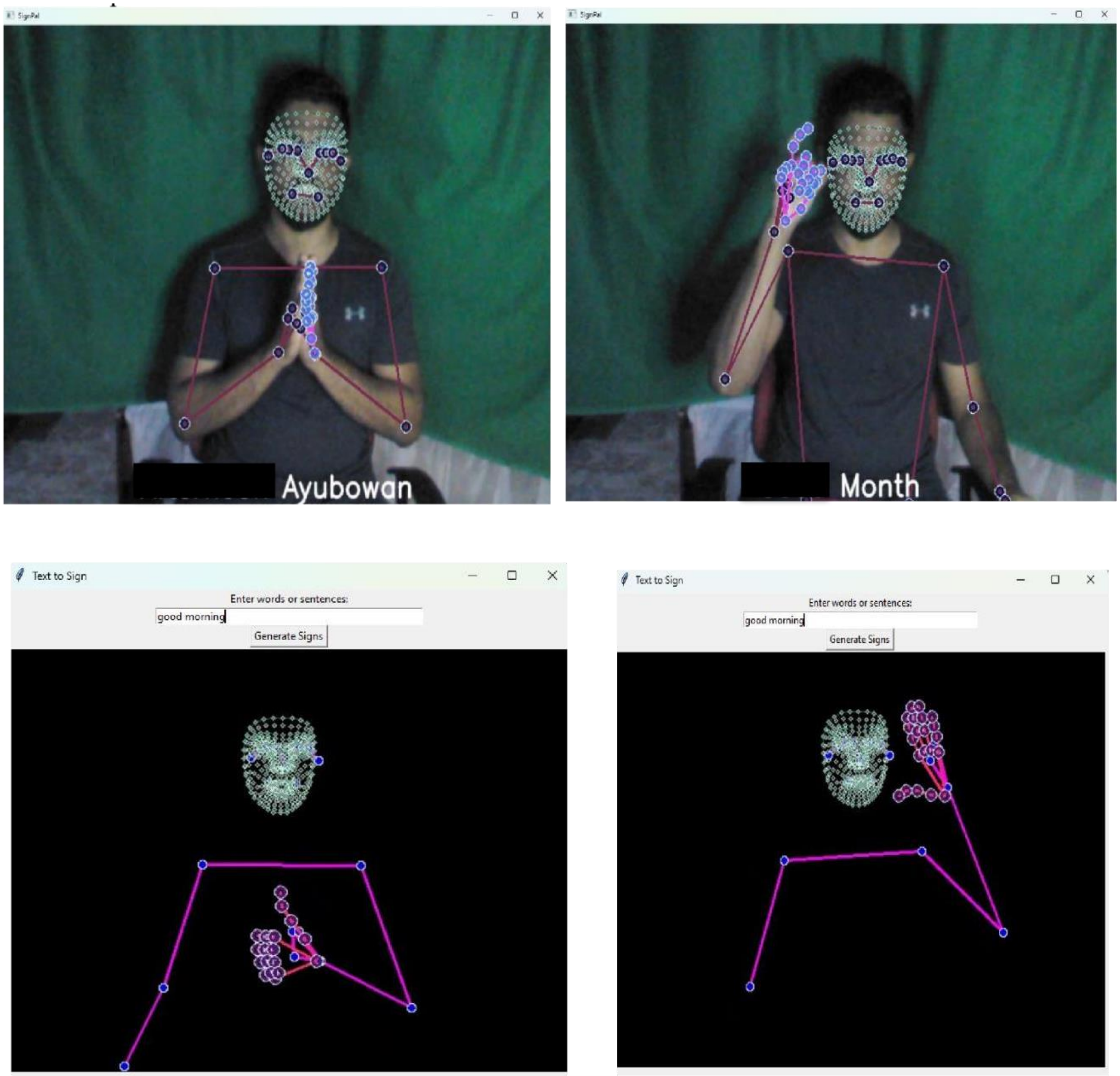
If Yes, What features would you like to see in an SSL(Sri Lankan Sign Language) mobile application? *

Long answer text

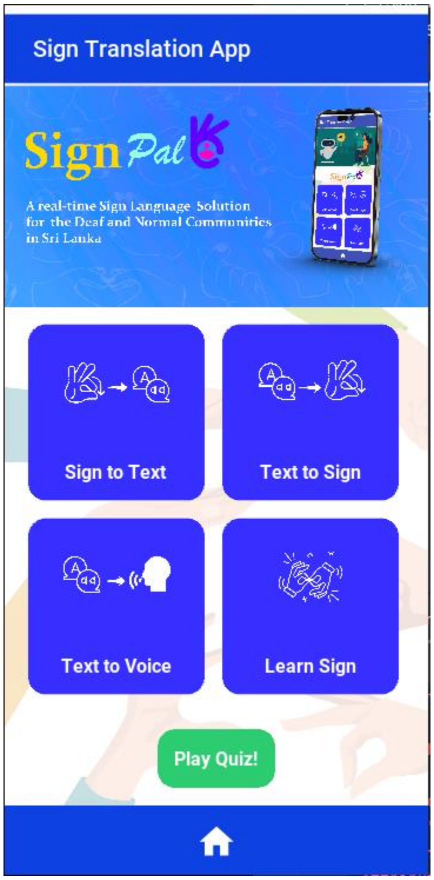
Any other comments or suggestions?

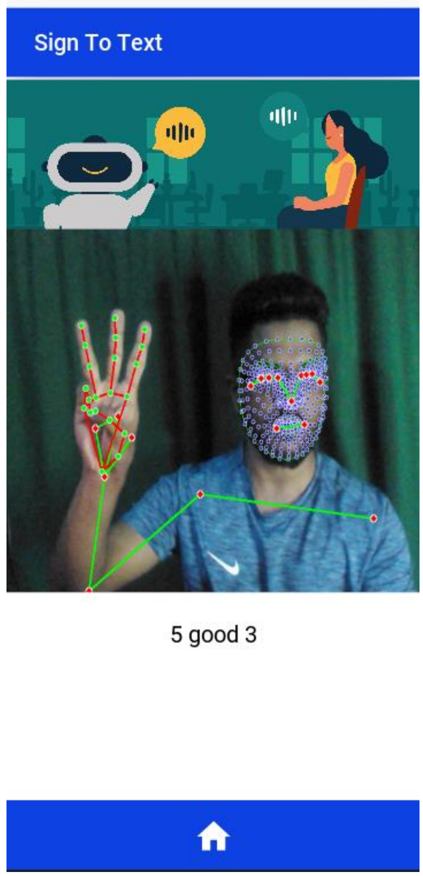
Long answer text

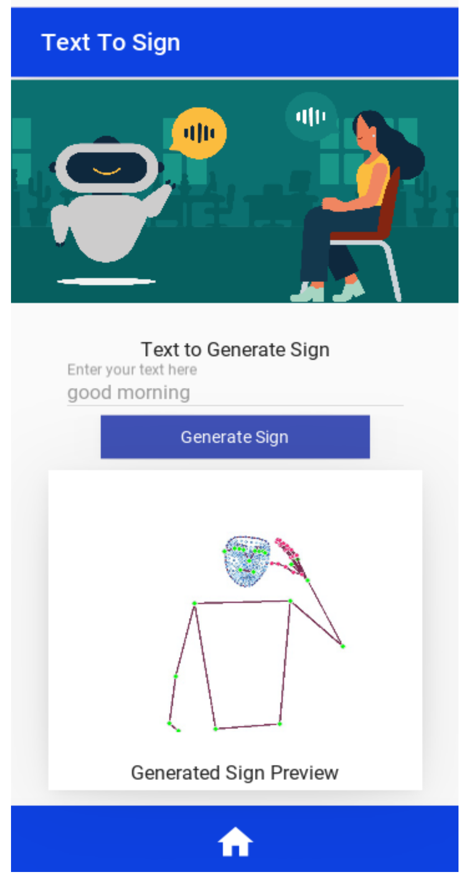
Appendix D – Low Fidelity UI Frame Diagrams

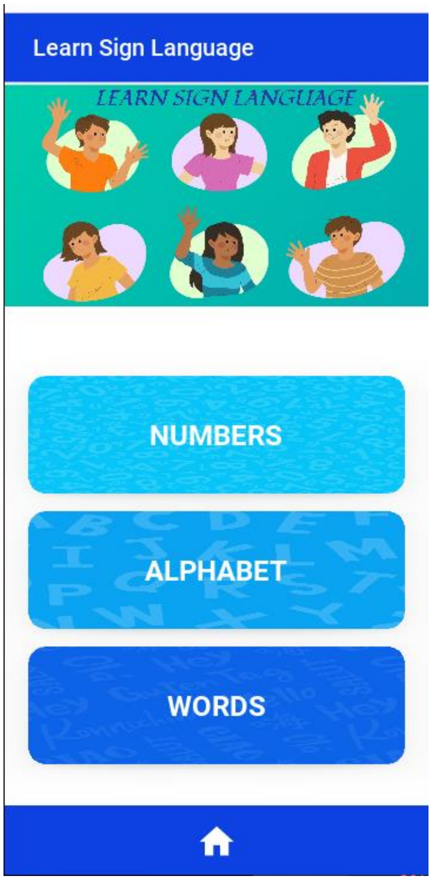


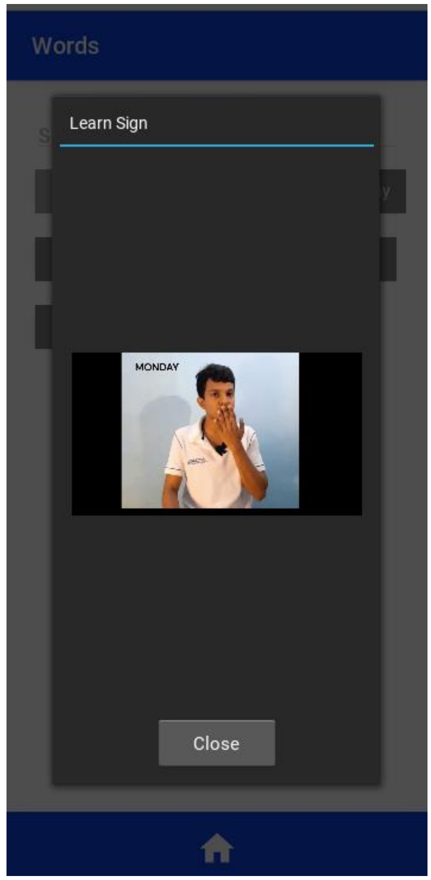
Appendix E – Screenshots of the UI

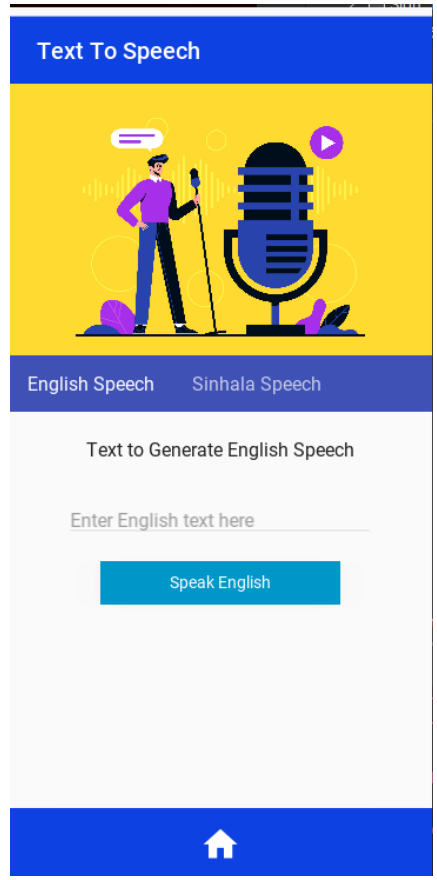







Learn Sign Language
LEARN SIGN LANGUAGE
NUMBERS
ALPHABET
WORDS


Words
Learn Sign
MONDAY
Close


Text To Speech
English Speech
Sinhala Speech
Text to Generate English Speech
Enter English text here
Speak English