Sign Language Recognition for Sentence Level Continuous Signings

I.K Godage

Sign Language Recognition for Sentence Level Continuous Signings

I.K Godage
Index No: 14000407

Supervisor: Dr. A.R Weerasinghe

January 2019

Submitted in partial fulfillment of the requirements of the B.Sc in Computer Science Final Year Project (SCS4124)

UCSC

Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Candidate Name: I.K Godage

………………………………………………

Signature of Candidate Date:

This is to certify that this dissertation is based on the work of

Mr. I.K Godage

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Supervisor's Name: Dr. A.R Weerasinghe

………………………………………………

Signature of Supervisor Date:

Co-Supervisor's Name: Dr. K.D Sandaruwan

………………………………………………

Signature of Co-Supervisor Date:

Abstract

It is no doubt that communication plays a vital role in human life. Most people consider communication as important as breathing for humans. However, there is a significant communication gap between hearing impaired people and others, because they use different techniques for their communication purposes which others cannot understand. These techniques are based on sign language, the main communication protocol among hearing impaired people.

In this research, we propose a method to bridge the communication gap between hearing impaired people and others which translates sign language sentences into text. Most of the existing solutions, based on technologies such as Kinect, Leap Motion, Computer vision, EMG and IMU try to recognize and translate individual signs of hearing impaired people. The few approaches to sentence-level sign language recognition suffer from not being user-friendly or even practical owing to the devices they use.

The proposed system was therefore designed to provide full freedom to the user. For this purpose, we employ two Myo armbands for gesture-capturing. Using signal processing and supervised learning based on a vocabulary of 49 words and 346 sentences for training with a single signer, we were able to achieve 75-80% word-level accuracy and 45-50% sentence level accuracy using gestural (EMG) and spatial (IMU) features for our signer-dependent experiment.

Preface

The purpose of this research was to recognize and translate sentence level continuous signing into a natural language. This research project is an extension of previous work and in that research project they only interested in recognition and translate individual signs. Because of that, it was unable to use the previous data set to continue this work. In order to continue this research project, I had to create a dataset which has sentence level continuous signing. Since I used Myo armband wearable device to capture the data and dataset was consist with signals. The analysis of the data is entirely my own work.

Then we proposed a method to collect the data and it helped segment the dataset in a proper way. Best of my knowledge this is the first time that Myo armband wearable device used to recognize sentence level continuous signings. I tried out few machine learning models and I carried out the analytical calculation to find the best model for this research. Then I observed how do the feature reduction and feature selection methods effect to the accuracy of the classifier and how we can employ feature reduction or selection methods to improve the real-time signs prediction time. Finally, we were able to show the promising result of the study.

Acknowledgement

I would first like to express my sincere gratitude to my supervisor Dr. A.R Weerasinghe and my co-supervisor Dr. K.D Sandaruwan of University of Colombo School of Computing. The doors to their offices were always open whenever I ran into a trouble spot or had a question about my research or writing. They consistently allowed this paper to be my own work but steered me in the right direction whenever they thought I needed it.

I would also like to thank linguistic Mr. Chamila and Sign Language interpreter Mr. Rajitha Hewage who involved in the signs selection and data collection processes respectively. Without their passionate participation and input, the signs selection and data collection process could not have been successfully conducted.

Another special thank should go to my colleagues and my brother who participated in the data collection process of my research. They dedicated their valuable time for me. Furthermore, I would like to thank my fellow colleagues for their feedback, cooperation and of course friendship.

Finally, I must express my very profound gratitude to my parents, my brother and my sister for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Table of Contents

Declaration .... **i**

Abstract .... **ii**

Preface .... **iii**

Acknowledgement .... **iv**

Table of Contents .... **v**

List of Figures .... **x**

List of Tables .... **xii**

List of Acronyms .... **xiii**

Chapter 1 - Introduction **1**

1.1 Background to the Research 3

1.1.1 Sign language recognition for word level signings 4

1.1.2 Sign language recognition for sentence level continuous signings 4

1.2 Research Problem and Research Questions 4

1.3 Justification for the research 5

1.4 Methodology 7

1.4.1 Approach 7

1.4.2 Methodology 7

1.5 Outline of the Dissertation 8

1.6 Definitions 8

1.6.1 Sign Language [26] 8

1.6.2 Hearing loss, Deafness and Profound deafness [27] 8

1.7 Delimitations of Scope 9

1.8 Conclusion 10

Chapter 2 - Literature Review **11**

2.1 Introduction 11

2.2 Theoretical Background 11

2.2.1 Hearing loss and deafness 11

2.2.2 The Different Types of Hearing Loss [24] 12

2.2.3 How does a deaf person communicate? [25] 12

2.2.3.1 Lip reading 12

2.2.3.2 Sign language 13

2.2.4 Sri Lankan sign language (SLSL) 13

2.2.5 Electromyography (EMG) and Initial Measurement Units (IMU) 14

2.2.5.1 Electromyography (EMG) 14

2.2.5.2 Initial Measurement Unit (IMU) 14

2.3 Related work 15

2.3.1 Word level sign language recognition systems. 15

2.3.1.1 Kinect device-based solution [5] 15

2.3.1.2 Data Glove device-based solution [6] 16

2.3.1.3 Leap Motion device-based solution [7] 16

2.3.1.4 Image/Video based solutions (Vision Based) [8] 16

2.3.1.5 EMG and IMU based solutions [9] 16

2.3.2 Sentence level sign language recognition systems 17

2.3.2.1 Kinect device-based solution [10] 17

2.3.2.2 Data Glove device-based solution [11] 17

2.3.2.3 Image/Video based solutions (Vision Based) [12] 17

2.3.2.4 EMG and IMU based solutions [13] 18

2.3.3 Word level sign language recognition systems using Myo gesture control armband 18

2.3.4 Sentence level sign language recognition systems using Myo gesture control armband 19

2.3.5 Previously conducted related research projects in Sri Lanka 19
2.4 Summary 19
Chapter 3 - Design 21
3.1 Introduction 21
3.2 Main Study-1 21
3.2.1 Sri Lankan sign language and making sentences 21
3.2.2 Device Selection 24
3.2.3 Data Collection 27
3.2.4 Data Preprocess 30
3.2.5 Data Segmentation 30
3.2.6 Feature Extraction 31
3.2.7 Feature reduction and Feature selection methods 31
3.2.7.1 Feature reduction vs Feature selection 32
3.2.8 Machine learning model training 32
3.2.9 Evaluation plan 33
3.3 Pre-Study 33
3.3.1 Select Signs 34
3.3.2 Data Collection 34
3.3.3 Data Preprocessing 35
3.3.3.1 EMG Data 35
3.3.3.2 IMU Data 35
3.3.4 Data Segmentation 35
3.3.5 Extract Features 35
3.3.6 Model Training 36
3.4 Main Study -2 36
3.4.1 Real-Time Gesture Classification 36
3.4.2 Capture the gestures using two Myo armbands 37

3.4.3 Data Preprocess/ Data Segment/ Extract Features.. 37
3.4.4 Predict the gestures using previously trained classifier 37
3.4.5 Display the predicted sentence.. 37
3.4.6 Evaluation of the real time gesture classifier 37
Chapter 4 - Implementation... 38
4.1 Introduction.. 38
4.2 Software Tools 38
4.3 Main Study-1 – Experiment 1.. 38
4.3.1 Data Collection 38
4.3.2 Data Preprocess... 41
4.3.3 Data Segmentation 42
4.3.4 Feature Extraction.. 43
4.3.5 Machine learning model training 44
4.4 Main Study-1 – Experiment 2.. 46
4.4.1 Feature reduction and Feature selection methods... 46
4.5 Main Study-2 47
4.5.1 Real-Time gesture classification... 47
Chapter 5 - Results and Evaluation 50
5.1 Introduction.. 50
5.2 Pre-study 50
5.3 Main Study-1 51
5.3.1 Experiment-1.. 51
5.3.2 Experiment-2.. 59
5.4 Main Study-2 64
5.5 Comparison of results of the proposed solution and related work...................... 65
Chapter 6 - Conclusions 67
6.1 Introduction.. 67

6.2 Conclusions about research questions (aims/objectives) 67

6.3 Conclusions about research problem 68

6.3.1 Contributions of this research 68

6.3.1.1 Recognize and translate sentence level continuous signings 68

6.3.1.2 Recognize and translate sentence level continuous signings in real time 69

6.3.1.3 Created a new dataset 69

6.4 Limitations 70

6.5 Implications for further research 70

References **71**

Appendix A: Diagrams **75**

Appendix B: Code Listings **76**

List of Figures

Figure 1.1: Communication Methods ... 2
Figure 1.2: Communication method of the proposed solution ... 3
Figure 3.1: Flow of the research project ... 21
Figure 3.2: Created a sentence with signs (Father goes home) ... 23
Figure 3.3: Created a sentence with signs (Friend learns English) ... 23
Figure 3.4: Signs Histogram ... 24
Figure 3.5: Myo armband ... 26
Figure 3.6: Data types of Myo armband ... 27
Figure 3.7: Data collection design ... 29
Figure 3.8: How to wear a Myo armband. ... 29
Figure 3.9: Sign 1 ... 34
Figure 3.10: Sign 2 ... 34
Figure 3.11: Sign 3 ... 34
Figure 3.12: Flow diagram of real time classification study ... 36
Figure 4.1: Graphical user interface of the python Application. ... 49
Figure 5.1: Sign 1 ... 50
Figure 5.2: Sign 2 ... 50
Figure 5.3: Sign 3 ... 50
Figure 5.4: Confusion matrix for the pre-study (ANN classifier) ... 51
Figure 5.5: Confusion matrix for the pre-study (Naïve Bayes classifier) ... 51
Figure 5.6: Confusion matrix main study 1 ... 53
Figure 5.7: ROC Chart ... 56
Figure 5.8: Accuracies of each classifier vs Feature reduction method vs Number of features ... 61
Figure 5.9: Reduce the features to 40 and the cross-validation accuracies of each model ... 62
Figure 5.10: Reduce the features to 80 and the cross-validation accuracies of each model ... 62

Figure 5.11: Reduce the features to 100 and the cross-validation accuracies of each model....63
Figure 5.12: Example output of a real-time gesture classification 65

List of Tables

Table 1.1: Break down of Sri Lankan hearing disabilities by gender 6
Table 1.2: Usability problems of existing solutions 6
Table 3.1: Selected words (signs) for sentences creation 22
Table 3.2: Comparison of EMG/IMU based technique with other existing approaches 25
Table 3.3: Specification of Myo armband 26
Table 5.1: Accuracies of the pre-study 51
Table 5.2: Average 10-fold cross-validation score 52
Table 5.3: Category of each class 52
Table 5.4: Precision, Recall and F1-score values of the main study 1 53
Table 5.5: Category of each class which has the F1-score less than 0.60 55
Table 5.6: AUC-ROC Values 56
Table 5.7: AUC-ROC results according to the academic point system 57
Table 5.8: Contribution of sign's positions for sentences misclassification 58
Table 5.9: Comparison of the results of the proposed solution and two main references 66

List of Acronyms

WHO	World Health Organization
EMG	Electromyography
IMU	Initial Measurement Units
ANN	Artificial Neural Network
NB	Gaussian Naïve Bayes
LDA	Linear Discriminant Analysis
RFC	Random Forest Classifier
LR	Logistic Regression
RC	Ridge Classifier
PCA	Principal Component Analysis
US	Univariate Selection
SVD	Singular Value Decomposition
RFE	Recursive Feature Elimination
RF	Random Forest

Chapter 1 - Introduction

According to Wikipedia, communication is the act of conveying meanings from one entity or group to another through the use of mutually understood signs and semiotic rules. There are many approaches to communication. Such as voice and speech, writing, manual signs, and gestures etc. These communication methods can be divided into two different forms. First one is verbal communication methods and the second one is non-verbal communication methods. Verbal communication describes the processes of communicating with words, whether written or spoken. Non- verbal communication is defined as the process of using the wordless message to generate meaning. Examples of nonverbal communication include haptic communication, chronemic communication, gestures, body language, facial expressions, eye contact, etc.

"*Communication is the essence of human life*"

-Janice Light-

It is no doubt that communication plays a vital role in human life. Most of the people think that communication is very important as same as breathing. Communication helps to share information and knowledge. As well as it helps to make new relationships, expression of ideas, feelings, emotions, thoughts.

There are two conditions to be satisfied for successful communication.

1. There must be at least two parties who involves for the communication
2. Each party must use a common communication platform

Most of the ordinary people (without any hearing/speaking disability) use verbal communication methods (E.g.-: voice and speech) for their communication purposes. By the way, deaf and speaking-impaired people use non-verbal communication methods (mostly signs and gestures) for their communication purposes. Both parties (ordinary people and deaf and speaking impaired people) use different platforms for their communication. Because of this problem, there is a huge communication barrier between ordinary people and hearing/speaking impaired people when they are communicating with each other.

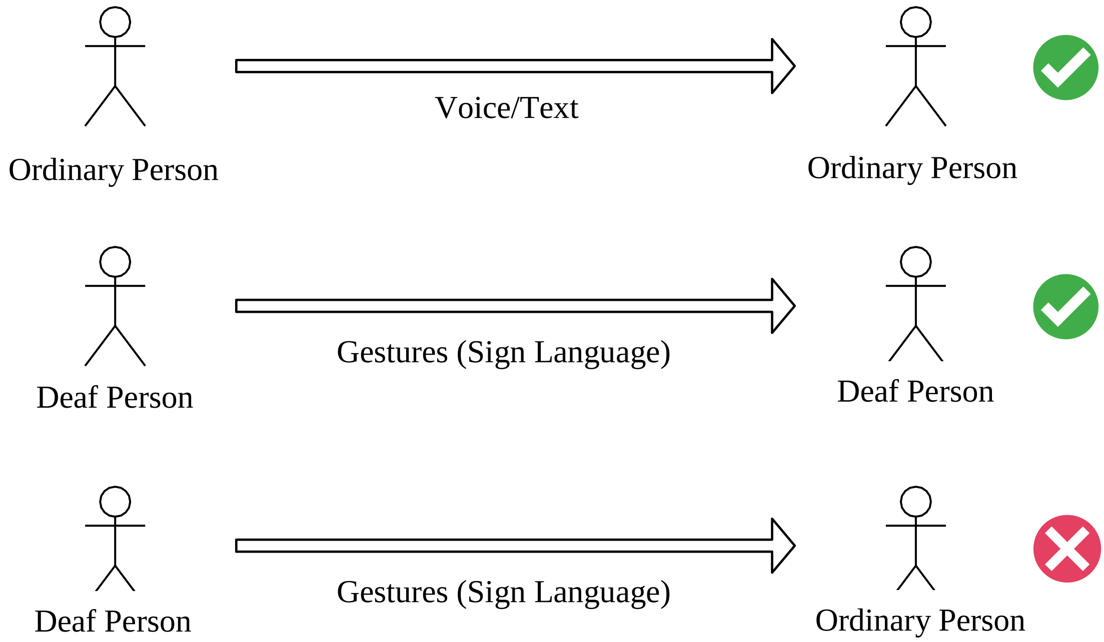


Figure 1.1: Communication Methods

Figure 1.1 demonstrates the communication barrier between a deaf person and an ordinary person. A deaf person uses sign language and the ordinary person uses voice or text. As mentioned previously, there are two conditions to be satisfied for successful communication. Ordinary person to ordinary person and deaf person to deaf person communications satisfy those conditions. However, deaf person to ordinary person communication does not satisfy the second condition which is both parties use a common communication platform. Here, they use sign language as their communication platform which deaf person can understand but an ordinary person cannot understand. Since they do not use a common communication platform, the communication is failed.

In the proposed solution, we created a sign language translator which can recognize sentence level continuous signings and translate them into a natural language. While it translates signs into text/voice, it improves the practical usability of the system by using a simple wearable device to capture the signs.

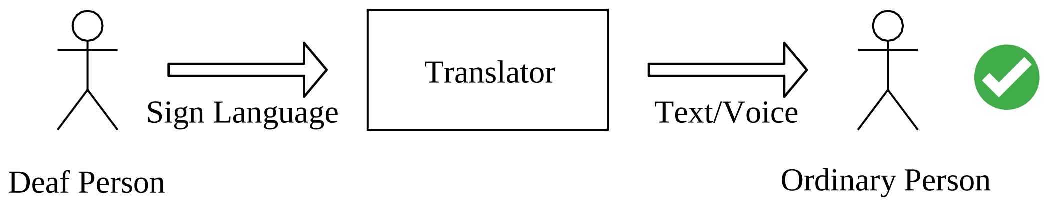


Figure 1.2: Communication method of the proposed solution

Figure 1.2 elaborates how does our solution simulate a common platform. It captures the signs of a sign language and translates them into text/voice. Afterward, the ordinary person can understand the signs which the deaf person has performed. In our solution, we capture sentence level continuous signings in Sri Lankan sign language and translate them into Sinhala natural language.

1.1 Background to the Research

As mentioned in the previous section communication in between hearing/speaking impaired and ordinary people is very difficult. Because the majority of the hearing/speaking impaired people use sign language as their first language and very few normal people good at communicating with them using sign language. Therefore, there is a huge communication gap between hearing/speaking impaired people and normal people.

In order to adhere this problem, many research were conducted in all-around the world to recognize different sign languages related to each country such as American Sign Language, Arabic Sign Language, Chinese Sign Language, Sri Lankan Sign Language, etc. Researchers used many devices and techniques to capture signs such as Kinect, Data Glove, Leap Motion, Vision-based techniques, and EMG/IMU based techniques. As well as they used different machine learning methodologies to recognize and translate sign language into a natural language. Such as Naïve Bayes Classifier, Artificial Neural Networks, Hidden Markov Models, Support Vector Machines etc.

Existing solutions can be divided into two classes.

1. Sign language recognition for word level signings.
2. Sign language recognition for sentence level continuous signings.

1.1.1 Sign language recognition for word level signings

Researchers were initially trying to recognize word level signings using different signs capturing devices and different methodologies. Those works can be compared with the user-friendliness of the signs capturing devices, the techniques and the accuracies of each solution. There are advantages as well as disadvantages in each solution. However, sign language recognition for word level signings is not sufficient for the practical usability of the sign language translator. Therefore, the researchers identified the necessity of recognition of sentence level continuous signings. Therefore, they started to recognize sentence level continuous signings.

1.1.2 Sign language recognition for sentence level continuous signings

Recognition of sentence level continuous signings is a tricky problem. Because, when performing a sentence level signing, a signer has to perform more than one sign in a continuous manner. In order to recognize each sign in that continuous signings, researchers followed different approaches. Approaches are depended on the devices that used for the signs capturing. However, the works which have been conducted so far to recognize sentence level continuous signings are not sufficient and there is huge potential in this research problem.

1.2 Research Problem and Research Questions

The existing sign language recognition and translation systems are trying to recognize signs using different techniques. Such as Image and vision-based techniques, Data Glove based techniques, EMG/IMU-based techniques. However, Most of these solutions are not user-friendly and limit the user's freedom of using the device. For an example, sometimes the user has to perform signs in front of the device (E.g. Kinect) or user has to wear cumbersome devices which are full of wires (E.g. Data Glove).

Most of the existing solutions cannot identify the sentence level continuous signings and those are trying to identify individual signs (word level signings). Even though some works are being tried to identify sentence level continuous signings, there are some practical issues of identifying the boundary of each individual signs of the continuous signings and the usability of those solutions was very minimal. In order to identify the sign boundary, there should be a way to identify the end of the 1st sign and start of the 2nd sign which is technically called identification of the movement epenthesis. Making pauses in between every two signs [12] and use a third-party device (E.g.: camera) to synchronize the signs [11] are two techniques of identifying the sign boundaries in previous studies. However, those techniques are very unnatural things in communication.

The proposed solution tries to address the following problems.

1. Sign language recognition for sentence level continuous signings.
2. Sign language recognition for sentence level continuous signings in a real-time manner.
3. The identification method of the sign boundary of the stream of signs should not be an unnatural one.
4. How to facilitate a hearing/speaking impaired people to communicate with ordinary people via assistive technology?
5. There are some solutions which are able to translate sign language into text or speech. But there are usability problems in those solutions. Such as the user has to wear bulky devices which are full of wires (E.g. Data Glove), the user has to perform in front of the device (E.g. Kinect). We try to improve the usability of the solution. (We use Myo armband which is lightweight, wireless and wearable device).

1.3 Justification for the research

Approximately 466 million people around the world have hearing/speaking disabilities. It is over 5% of the world's population. 34 million of them are children. According to WHO's statistics, they estimated that there will be over 900 million people have hearing/speaking disabilities by 2050 [1]. In Sri Lanka, there are around 400,000 people are suffering from hearing/speaking disabilities [2] and 1000 of them are totally deaf. Table 1.1 shows that the breakdown of hearing disabilities of Sri Lanka by gender. [2].

Table 1.1: Break down of Sri Lankan hearing disabilities by gender

Gender	Population	Rate per 1000 persons
Male	169,201	19
Female	219,876	23
Total	389,077	21

Most of the existing solutions try to recognize individual characters or individual words of particular Sign Language. Recognition of an individual character or word is not practical and insufficient in common usage. Because, when having a proper verbal communication people speak in sentence level. Hence, recognition of sentence level continuous signings is useful than recognition of word level signs. In non-verbal communication context, deaf people perform signs in a continuous manner, which is more similar to the speaking of ordinary people. Therefore recognition of an individual character or word is not sufficient in proper translator and there should be a way of recognition sentence level continuous signings. Most of the existing solutions cannot recognize sentence level continuous signings. Even though some works are being tried to identify sentence level continuous signings such as using vision-based techniques, data glove-based techniques, etc. there are some usability problems as explained in Table 1.2.

Table 1.2: Usability problems of existing solutions

Existing Approach/Solution	Usability Problem
Vision-based solutions	The sign should be performed in front of the device
Data Glove based solutions	Cumbersome device
Leap motion-based solutions	The sign should be performed in front of the device

Most of the existing solutions try to translate American Sign Language (ASL), Arabic Sign Language (ArSL), Chinese Sign Language (CSL) into a natural language [11, 13, 15]. In the Sri Lankan context, the works [16 - 20] which were conducted in this domain are insufficient. Therefore, there is a huge space to fill.

1.4 Methodology

1.4.1 Approach

The Myo gesture control armband which is a commercial-off-the-shelf device was used for this research project and it was used to capture the hand gestures. It gives EMG (Electromyography) data and IMU (Initial measurement units) data. Those are just integers (-127 to +127). In this project, we had to work with numbers. As well as different experiments were carried out. Such as different signal processing techniques, different machine learning models and different framework evaluation methods. Because of the above reasons, a quantitative research approach was used and the experimental study was performed.

1.4.2 Methodology

In this research, what we are going to perform is recognizing sentence level continuous signings and translate them into a text.

As explained in the research questions section, high usability devices were used for the data collection. Because the usability of the existing solutions is very minimal. In this project, Myo armband was used as the gesture acquisition device. Because research study [9] and [13] show good results for Myo armband and it also preserves the usability aspects.

For this research, Sri Lankan Sign Language (SLSL) was used as the Sign Language. In Sri Lankan sign language, there are around 2000 signs are practically used [21] by the hearing/speaking peoples for their communication purposes. For this study, it is not practical to use all signs in Sri Lankan sign language. Because of that, some of them were selected and using those words (signs) [22], sentences (sequence of signs) were created for the data collection process.

A professional sign language interpreter was chosen and he was asked to perform each sign for a particular sentence. Data were saved in CSV files separately.

After the data collection process, by using digital signal processing techniques and machine learning techniques, each sentence level continuous signings were recognized and finally, each model was evaluated.

1.5 Outline of the Dissertation

The dissertation is structured as follows. Existing approaches related to the domain of sign language translation is detailed in Chapter 2. We then describe the research design and methodology adopted in this research in Chapter 3, detailing potential approaches in addressing the problem. Next, we demonstrate the implementation details of the proposed methodology (Chapter 4). We then present the evaluation model and the interpretation of the results of the proposed approaches in Chapter 5. Finally, Chapter 6 presents our conclusions based on the research carried out and outlines interesting future work to be done.

1.6 Definitions

1.6.1 Sign Language [26]

Sign languages (also known as signed languages) are languages that use the visual-manual modality to convey meaning. Language is expressed via the manual sign stream in combination with non-manual elements. Sign languages are full-fledged natural languages with their own grammar and lexicon. This means that sign languages are not universal and they are not mutually intelligible, although there are also striking similarities among sign languages. Sign Languages are used by deaf people as their communication method.

1.6.2 Hearing loss, Deafness and Profound deafness [27]

Hearing loss: This is a reduced ability to hear sounds in the same way as other people.
Deafness: This occurs when a person cannot understand speech through hearing, even when sound is amplified.

Profound deafness: This refers to a total lack of hearing. An individual with profound deafness is unable to detect sound at all.

The severity of hearing impairment is categorized by how much louder volumes need to be set at before they can detect a sound. Some people define profoundly deaf and totally deaf in the same way, while others say that a diagnosis of profound deafness is the end of the hearing spectrum.

1.7 Delimitations of Scope

Currently, there are around 300 sign languages all around the world [4]. For this research study, Sri Lankan sign language (SLSL) was used as the sign language. In Sri Lankan sign language, there around 2000 signs [21]. When we create sentences, it is not a practical task to create sentences using all the signs. Fifty (49) words which are common and useful in our day to day life were selected.

Sentences were created using selected signs. Three signs (words) were selected for a particular sentence. The structure of a sentence is SOV (Subject + Object + Verb) and adverb, adjectives were not used for the sentences. The total number of sentences is 346.

Sri Lankan sign language users follow below techniques to perform signs [21].

1. Hand gestures
2. Lip movements
3. Facial Expressions

However, in this research project, we only concern on the hand gestures because hand gestures are used as the main representation technique. We are not interested in lip movements and facial expressions for this research.

Sri Lankan sign language utilizes the movement of both hands and a mirror movement has the same meaning. Therefore, a right-handed signer was used for the data collection purpose and both hands were considered when the signer performs the signs. For this data collection purpose, single signer was used throughout the project (user dependent study).

This research for only recognizing and translating Sri Lankan sign language into Sinhala natural language. However, no other way around (Sinhala natural language translate into Sri Lankan Sign Language).

1.8 Conclusion

This chapter laid the foundations for the dissertation. It introduced the research problem and research questions and hypotheses. Then the research was justified, definitions were presented, the methodology was briefly described and justified, the dissertation was outlined, and the limitations were given. On these foundations, the dissertation can proceed with a detailed description of the research.

Chapter 2 - Literature Review

2.1 Introduction

In this chapter, a review of related work is provided. Section 2.2 discusses the related theoretical details of this research study. Subsection 2.2.1 discusses the hearing loss and deafness, 2.2.2 discusses the different types of hearing loss, 2.2.3 discusses the communication methods of a deaf person. Sign language is one of the main communication methods for a deaf person. Since we considered Sri Lankan sign language, subsection 2.2.4 discusses the Sri Lankan sign language. Our selected device (Myo armband) for the data capturing gives two types of signals which are EMG and IMU signals, these two types of signals are discussed in subsection 2.2.5.

Section 2.3 discusses the previously conducted research attempts and results. Subsection 2.3.1 discusses the existing research of word level signs recognition methods and their accuracies. Subsection 2.3.2 discusses existing research of sentence level continuous signings recognition methods and results. The subsection 2.3.3 and 2.3.4 discuss existing research of sign language recognition for word level and Sentence level continuous signings using Myo armband. In subsection 2.3.5, existing studies related to this domain which are conducted in Sri Lanka are listed down. Finally, a summary is provided in section 2.4.

2.2 Theoretical Background

2.2.1 Hearing loss and deafness

A person who is not able to hear as well as someone with normal hearing – hearing thresholds of 25 dB or better in both ears – is said to have hearing loss. Hearing loss may be mild, moderate, severe, or profound. It can affect one ear or both ears and leads to difficulty in hearing conversational speech or loud sounds.

'Hard of hearing' refers to people with hearing loss ranging from mild to severe. People who are hard of hearing usually communicate through spoken language and can benefit from hearing aids, cochlear implants, and other assistive devices as well as captioning. People with more significant hearing losses may benefit from cochlear implants.

'Deaf' people mostly have profound hearing loss, which implies very little or no hearing. They often use sign language for communication. [1]

2.2.2 The Different Types of Hearing Loss [24]

There are 3 categories of hearing loss.

- **Conductive hearing loss** happens when sound waves cannot reach the inner ear due to a blockage of some kind, such as fluid or earwax buildup. This type of hearing loss can usually be treated.
- **Sensorineural hearing loss** occurs when there is damage to the inner ear structure or the nerves that relay information from the ears to the brain. Unfortunately, sensorineural hearing loss is permanent.
- **Mixed hearing loss** occurs when you have compounding factors of both conductive and sensorineural hearing loss

2.2.3 How does a deaf person communicate? [25]

Deaf people have two main ways of communicating with others. Deaf people may not be able to hear what you're saying, but that doesn't mean they can't understand you.

2.2.3.1 Lip reading

This is a technique to understand speech by visually interpreting the movements of the lips and tongue, using facial expression and body language to help.

Lip readers also use the information they have from:

- The context (or topic) of the conversation – this helps narrow down the possible vocabulary they might be lip reading
- The knowledge they have about the language and its lip patterns.

- Any residual hearing, they may have (with or without a hearing aid).

It is used by many deaf people who do not sign; especially those who were born hearing and have either gradually or suddenly lost their hearing during their life.

2.2.3.2 Sign language

There are around 300 sign languages in the world [4]. These languages are different from each other. Word order in sentences can differ between these languages as well as from written text. Sign language is a visual language that incorporates gestures, facial expressions, head movements, body language and even the space around the speaker. Hand signs are the foundation of the sign language. Many signs are iconic, meaning the sign uses a visual image that resembles the concept it represents. Actions are often expressed through hand signals that mimic the action being communicated.

2.2.4 Sri Lankan sign language (SLSL)

In Sri Lanka, there are around 400,000 people are suffering from hearing or speaking disabilities. 1000 of them are totally deaf. Majority of those people use Sri Lankan sign language as their mother tongue. Most of the ordinary people who do not have any hearing/speaking disabilities cannot understand this language. Sri Lankan Sign Language is a visual-gestural Language based on hand movements and the body (including facial expressions, lip moments, head movement). In Sri Lankan sign language, it can represent alphabets of normal languages (Sinhala, English) and it can represent other sings for each word. Currently, Sri Lankan sign language contains around 2000 signs [21,22]. It also has regional signs across Sri Lanka.

British introduced the sign language to Sri Lanka. Hence, Sri Lankan sign language has been developed for years with the influence of British Sign Language (BSL). Because of that, there are some similarities between Sri Lankan sign language and British sign language.

2.2.5 Electromyography (EMG) and Initial Measurement Units (IMU)

Myo armband is selected as the gesture capturing device for this research. That device gives two types of signals which are EMG and IMU. These two types of signals are discussed in the following.

2.2.5.1 Electromyography (EMG)

Electromyography (EMG) is the detection and recording of the electrical signal produced by muscle tissue as it contracts. The anatomy and physiology of a muscle can be modeled as follows. A muscle is composed of a set of overlapping motor units. A motor unit is a set of many fibers innervated by a single motor neuron. The ends of the fibers are connected to the tendons.

One end of the motor neuron connected to the spin code and other end connected to the fibers of the muscle through the neuromuscular junction. As we know, nerves use electrical impulses to coordinate muscle movement in our bodies. When a motor neuron fires, an electrical impulse propagates to the tendons through neuromuscular junction. Then that electrical impulse propagates through the fibers and reaches the tendons. Then the movement of the relevant body part happens. Since EMG is designed to record the electrical activity produced by the muscles during the rest and contraction, we can capture the data.

EMG depends on several factors. Such as the thickness and temperature of the skin, the thickness of the fat between the muscle and the skin, the velocity of the blood flow, and the location of the sensors.

Depending on the device that we are going to capture the EMG data, EMG signals can be divided into two categories. Such as surface EMG and intramuscular EMG. To capture the surface EMG signals, sensors must be placed on the skin. To capture the intramuscular EMG signals, sensors (E.g. needle) must be placed inside the muscle. In this research, surface EMG signals were captured and sensors were placed on the skin.

2.2.5.2 Initial Measurement Unit (IMU)

According to the Wikipedia an IMU is an electronic device that measures and reports a body's specific force, angular rate, and sometimes the magnetic field surrounding the body, using a combination of accelerometers and gyroscopes, sometimes also magnetometers. Therefore, in this research project, we used the following sensors for the IMU data acquisition.

1. Accelerometer – Use for detecting linear acceleration.
2. Gyroscope – Use for detecting rotational rate.
3. Magnetometer – Used as a heading reference and magnetometer data to calculate the roll, pitch, yaw angles.

2.3 Related work

There are studies, that have been conducted on this topic and some solutions have been developed as well. Most of the worldwide research were conducted based on the American Sign Language (ASL) or limited to other native sign Languages such as Indian, Chinese sign languages. Most of the existing solutions are using many devices and techniques to address the problem. We can categorize those techniques and devices as follows.

1. Kinect device-based solutions
2. Data Glove device-based solutions
3. Leap Motion device-based solutions
4. Image/Video based solutions (Vision Based)
5. EMG and IMU based solutions

2.3.1 Word level sign language recognition systems.

2.3.1.1 Kinect device-based solution [5]

Kalin Stefanov and Jonas Beskow proposed a method for automatic recognition of isolated Swedish Sign Language signs for the purpose of educational signing-based games. Two datasets consisting of 51 signs have been recorded from a total of 7 (experienced) and 10 (inexperienced) adult signers. Signer-dependent recognition rate is 95.3% for the most consistent signer. HMM have been used as the model. Signerindependent recognition rate is on average 57.9% for the experienced signers and 68.9% for the inexperienced.

2.3.1.2 Data Glove device-based solution [6]

Wu jiangqin et al proposed a simple sign language recognition system based on data glove. In this paper, the process of building a simple word-level sign language recognition system is presented, and the method for recognizing sign language word is also proposed. 26 sign language words were used for this experiment. This is a Chinese sign language recognition system and there are primarily three methods used for sign language recognition. Such as template matching, neural networks, Hidden Markov Model. The Recognition rate of testing samples is over 90%.

2.3.1.3 Leap Motion device-based solution [7]

Deepali Naglot and Milind Kulkarni proposed a system for recognition of 26 different alphabets of American Sign Language using Leap Motion Controller. LMC is 3D non-contact motion sensor which can track and detects hands, fingers, bones and finger-like objects. Multi-Layer Perceptron (MLP) is executed on a dataset of total 520 samples and Recognition rate of the proposed system is 96.15%.

2.3.1.4 Image/Video based solutions (Vision Based) [8]

Manar et al introduce the use of different types of neural networks in human hand gesture recognition for static images as well as for dynamic gestures. A static gesture is a particular hand movement represented by a single image, while a dynamic gesture is a moving gesture represented by a sequence of images. This work focuses on the ability of neural networks to assist in Arabic Sign Language (ArSL) hand gesture recognition. This work focuses on the 28 letters of the Arabic alphabet. Fully recurrent architecture has had a performance with an accuracy rate of 95% for static gesture recognition.

2.3.1.5 EMG and IMU based solutions [9]

Jian Wu et al proposed a real-time American SLR system leveraging fusion of surface electromyography (sEMG) and a wrist-worn inertial sensor at the feature level. A feature selection is provided for 40 most commonly used words and for four subjects. SVM was used as the classifier model. Their system achieves 95.94% recognition rate.

2.3.2 Sentence level sign language recognition systems

2.3.2.1 Kinect device-based solution [10]

Edon Mustafa and Konstantinos Dimopoulos developed a system which uses SigmaNIL framework to recognize Alphabet, Number, Word, and Sentence of Kosova sign language. The recognition of sentence "HELLO DAUGHTER" is done by starting the timer when the HELLO sign is performed and if within five seconds if sign DAUGHTER happens, it is concluded that sentence "HELLO DAUGHTER". was performed. They divided the body into few regions and both "HELLO" and "DAUGHTER" signs are performed in two different regions. After "HELLO" performs, System is looking for another five seconds and within that time period, if "DAUGHTER" sign performed, they identify it as a sentence. The last word identifies using region changing. The recognition rate for one sentence from three testers is 73%.

2.3.2.2 Data Glove device-based solution [11]

Noor Tubaiz et al proposed a glove-based Arabic sign language recognition system using a novel technique for sequential data classification. The dataset contains 40 sentences using an 80-word lexicon. Data labeling is performed using a camera to synchronize hand movements with their corresponding sign language words. Modified k-Nearest Neighbor (MKNN) approach is used for classification. The proposed solution achieved a sentence recognition rate of 98.9%.

2.3.2.3 Image/Video based solutions (Vision Based) [12]

Daniel Kelly et al presented a multimodal system for the recognition of manual signs and non-manual signals within continuous Irish sign language sentences. In this paper, they proposed a multichannel HMM-based system to recognize manual signs (hand gestures) and non-manual signals (E.g. facial expressions, head movements, body postures, and torso movements). Signer has to make pauses between words, to segment the words in a sentence. They have considered about 8 words. Using 4 words at a time they have created sentences. Their system achieved a detection ratio of 95.7%.

2.3.2.4 EMG and IMU based solutions [13]

Xu Zhang et al presented a framework for hand gesture recognition based on the information fusion of a three-axis accelerometer (ACC) and multichannel electromyography (EMG) sensors. In this framework, the start and end points of meaningful gesture segments are detected automatically by the intensity of the EMG signals. 72 Chinese Sign Language (CSL) words and 40 CSL sentences are classified using a decision tree and multi-stream hidden Markov models. overall word recognition accuracy is 93.1% and a sentence recognition accuracy is 72.5%.

We observed that EMG and IMU based solutions have sufficient accuracy, they can be enhanced as mobile solutions and they improve the practical usability of the system. Therefore, we planned to use EMG and IMU based device for this research. Instead of using electrodes, we chose Myo gesture control armband which is a commercial-off-the-shelf device for this research as the data capturing device [23].

2.3.3 Word level sign language recognition systems using Myo gesture control armband

Celal Savur and Ferat Sahin proposed a system [14] to identify recognize the American Sign Language alphabet letters (26) and a one for the home position. As a classification method, Support Vector Machine and Ensemble Learning algorithm were used. Accuracies are 80% and 60.85% respectively. Only one hand use to perform gestures.

Prajwal Paudyal et al proposed SCEPTRE [15] which utilizes two non-invasive wrist-worn devices (Both arms were used) to decipher gesture-based communication. The system uses a multitiered template-based comparison system for classification on input data from accelerometer, gyroscope, and electromyography (EMG) sensors. They tried to identify 20 signs of American sign language and the system was able to achieve an accuracy of 97.72 % for ASL gestures.

2.3.4 Sentence level sign language recognition systems using Myo gesture control armband

Best of our knowledge, we were unable to find literature which tries to recognize sentence level continuous signing using Myo gesture control armband. However, the proposed system used Myo gesture control armband to recognize sentence level continuous signings.

2.3.5 Previously conducted related research projects in Sri Lanka

1. A Sinhala Finger Spelling Interpretation System Using Nearest Neighbor Classification [16] 2002.
2. Image-Based Sign Language Recognition System for Sinhala Sign Language [17] 2013.
3. "The Rhythm of Silence" - Gesture Based Intercommunication Platform for Hearing-impaired People (Nihanda Ridma) [18] 2014.
4. Sign Language Translation Approach to Sinhalese Language [19] 2016.
5. Framework for Sinhala Sign Language Recognition and Translation Using a Wearable Armband [20] 2016.

2.4 Summary

The first part of this section we discussed the theoretical background which is related to this research project. So, we have discussed Following things,

- Hearing loss and deafness
- The Different Types of Hearing Loss
- How does a deaf person communicate?
- Electromyography (EMG)
- Initial Measurement Units (IMU)

As the second part, we discussed the related work.

Under the existing solutions for word-level sign language recognition systems. We observed that EMG & IMU based solutions have good accuracy, mobility, and user-friendliness. Even though other solutions have good accuracy, their usability and mobility are questionable. Therefore, we chose MYO armband [23] which is an EMG/IMU based device as our data capturing device.

We were able to observe the method used for the identification of movement epenthesis in each solution. However, those methods are very unnatural in communication. Moreover, we observed that existing solutions have usability and mobility issues in the existing solutions for sentence level sign language recognition systems section.

Under the word level sign language recognition systems using Myo gesture control armband section, we observed that existing solutions used both single and both hands for gesture recognition and the accuracy of each method was significant.

Best of our knowledge, we were unable to find existing solutions for sentence level sign language recognition systems using Myo gesture control armband. However, some of the existing solutions [15,20] mentioned that sentence level sign language recognition is their future work.

Finally, existing work which were conducted at different universities in Sri Lanka were listed down. We were able to observe that, research work that were conducted in Sri Lanka which relates to this domain is very minimal. However, the proposed research project is an extension of a previous research project which was titled as “Framework for Sinhala Sign Language Recognition and Translation Using a Wearable Armband” [20].

Therefore, we can conclude that there is enough space in this research domain both locally and globally.

Chapter 3 - Design

3.1 Introduction

This chapter explicates the proposed solutions to the research problem. It consists of three major sections namely Main study-1, Main study-2, and Pre-study.

3.2 Main Study-1

In this section, it will be discussed the flow of the research project as shown in Figure 3.1.

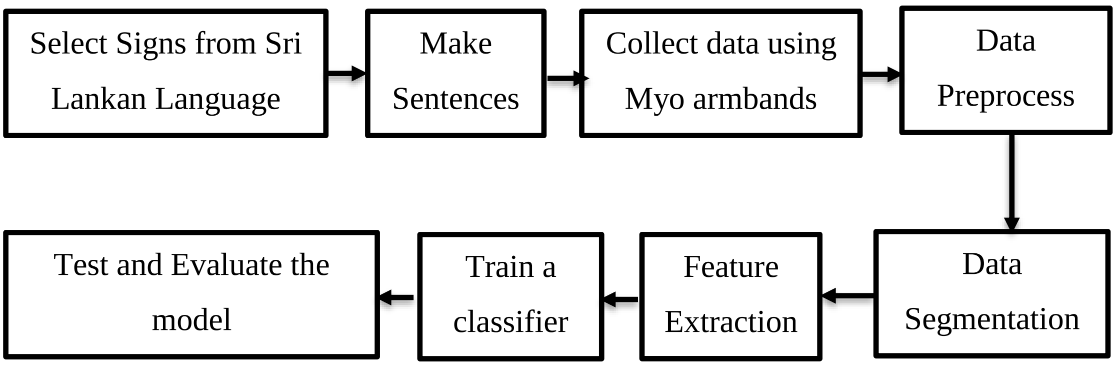


Figure 3.1: Flow of the research project

3.2.1 Sri Lankan sign language and making sentences

There are around 300 sign languages all around the world. Most of the existing solutions try to recognize American, Arabic or Chinese sign languages and translate them into other natural languages. However, studies which have been done so far in this research domain in the Sri Lankan context are very minimal. Therefore, Sri Lankan sign language was selected as the Sign Language for this research project.

There are around 2000 signs in Sri Lankan sign language [21, 22]. When we creating sentences using those signs, it is not practical to use all the signs available in Sri

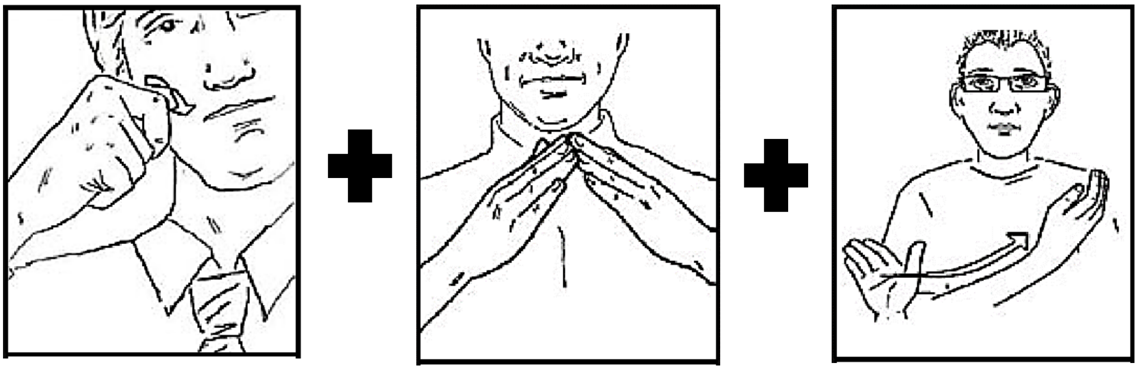
Lankan sign language. Because of that reason, a subset from the total sign set was selected. Selected Signs are common and useful signs in our day to day life. 49 Signs were selected. These 49 signs include nouns, pronouns nouns, and verbs only.

In this sentences creation process, we used SOV (Subject + Object + Verb) structure as the structure of the sentences. Each sentence consists of three words. Those were subject, object, and verb. We did not use adverbs, adjectives, etc. 346 sentences were created using those 49 selected signs. There are 1038 individual signs in the dataset. Table 3.1 shows that 49 words which were selected as subjects, objects, and verbs. Figure 3.4 shows the frequencies of each sign which are selected for this study. Figure 3.2 and Figure 3.3 shows that two example sentences which were created.

Table 3.1: Selected words (signs) for sentences creation

Class	**Subject**	**Class**	**Object**	**Class**	**Verb**
1	මම / I	19	මේසය / Table	36	අදිනවා / Pull
2	අපි / We	20	පුටුව / Chair	37	අඳිනවා / Draw
3	ඔහු / He	21	බර / Weight	38	අරිනවා / Open
4	ඇය / She	22	චිත්‍රය / Painting	39	ඉගෙනගන්නවා / Learn
5	ඔවුන් / They	23	දොර / Door	40	ඉරනවා / Tear
6	අම්මා / Mother	24	ජනෙලය / Window	41	උණුකරනවා / Boil
7	තාත්තා / Father	25	ඉංග්‍රීසි / English	42	උයනවා / Cook
8	අක්කා / Elder Sister	26	පොත / Book	43	එල්ලනවා / Hang
9	අයියා / Elder Brother	27	පත්තරය / News Paper	44	පලනවා / Split
10	නංගී / Younger Sister	28	කඩදාසිය / Paper	45	යනවා / Go
11	මල්ලී / Younger Brother	29	වතුර / Water	46	බොනවා / Drink
12	දුව / Daughter	30	මාලු / Fish	47	ලියනවා / Write
13	පුතා / Son	31	එළවළු / Vegetable	48	විසිකරනවා / Throw
14	නැන්දා / Aunt	32	රෙදි / Clothes	49	සිටුවනවා / Plant
15	මාමා / Uncle	33	දර / Firewood		
16	ආච්චි / Grand	34	ගෙදර / Home		

	Mother				
17	සීයා / Grand Father	35	ගස / Tree		
18	යාලුවා / Friend				



තාත්තා / Father　　ගෙදර / Home　　යනවා / Go

Figure 3.2: Created a sentence with signs (Father goes home)

යාලුවා / Friend　　ඉංග්‍රීසි / English　　ඉගෙනගන්නවා / Learn

Figure 3.3: Created a sentence with signs (Friend learns English)

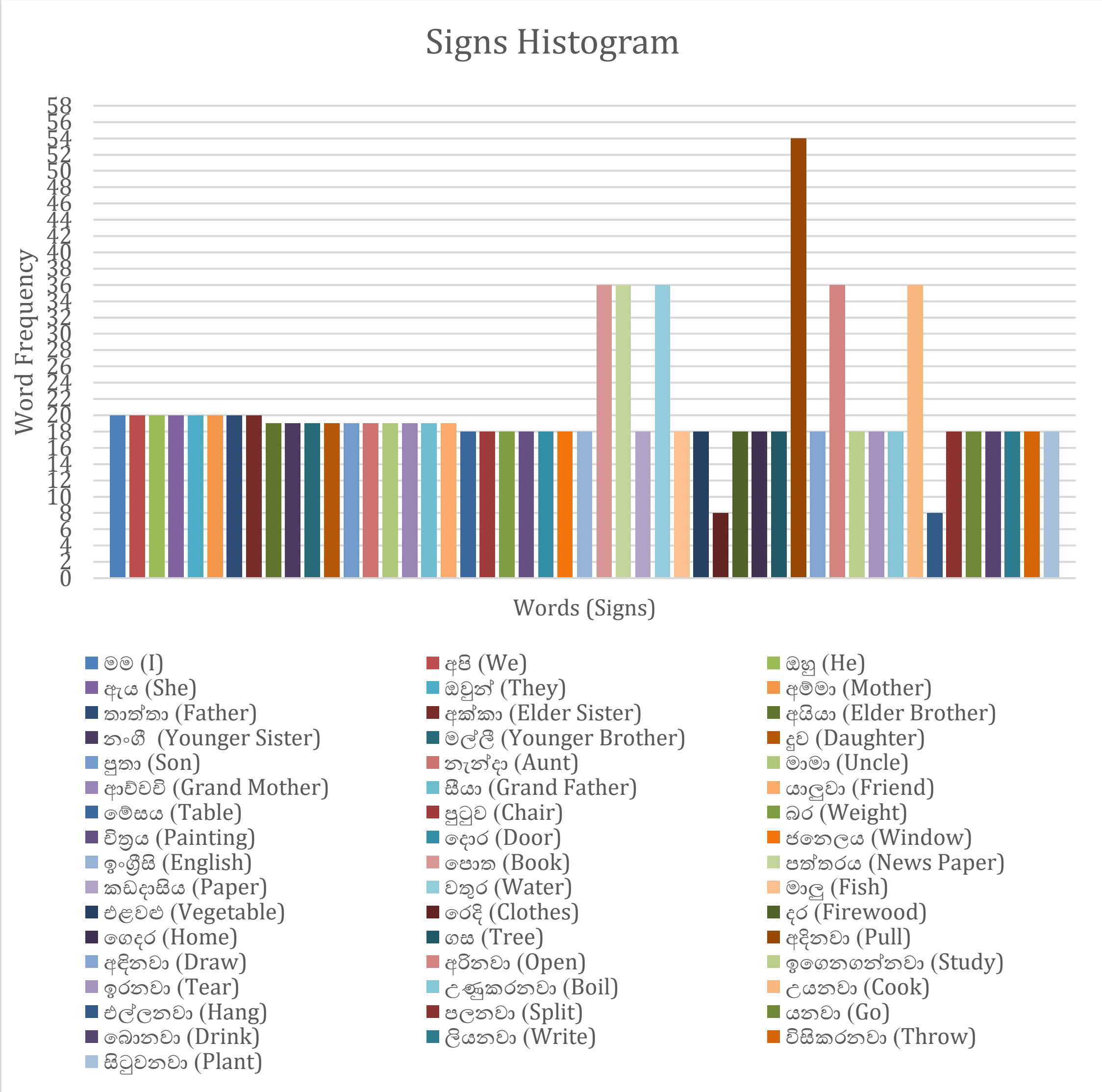


Figure 3.4: Signs Histogram

3.2.2 Device Selection

In this research, Myo gesture recognition armband was selected as our data capturing device. This device was developed and introduced by Thalmic Labs Inc as a new way of using hand gestures to interact with computers and mobile devices (especially as an input/controlling device). Before Myo armband was selected as the data capturing device we had to consider three things.

1. Sign Recognition Accuracies of EMG and IMU based techniques
2. The mobility of the device
3. User convenience of the device

Table 3.2 shows the comparison between EMG/IMU based techniques and other approaches.

Table 3.2: Comparison of EMG/IMU based technique with other existing approaches

Device/ Technology	**Vocabulary size**	**Accuracy**	**Mobility**	**User convenience**
Kinect [5]	51	95.3%	Not a mobile device	The sign should be performed in front of the device, not suitable for day to day usage
Leap Motion [6]	26	90%	Can be attached to a mobile device	The sign should be performed in front of the device, not suitable for day to day usage
Data Glove [7]	26	96.15%	Most are wired to a computer	Cumbersome device
Image Processing [8]	28	95%	Not a mobile solution	The sign should be performed in front of the cameras, not suitable for day to day usage
EMG/IMU solution [9]	40	95.94%	Can be enhanced as mobile solutions	The user should stick the pods or wear the device as an armband

As mentioned in the above Table 3.2, EMG/IMU solutions have a competitive accuracy compared to the other solution. However, EMG/IMU solution can be enhanced as a mobile solution and user convenience is higher than others. Therefore, we have chosen the EMG/IMU method for this research and Myo armband as the data capturing device.

Figure 3.5 shows that the Myo armband and Table 3.3 shows that the specification of the Myo armband.

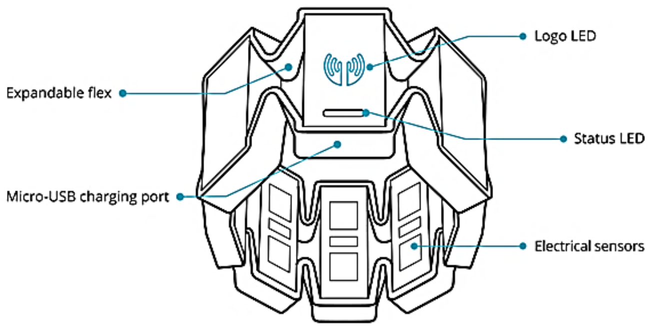


Figure 3.5: Myo armband

Table 3.3: Specification of Myo armband

Sizing, Weight, and Dimensions	
Arm size	Expandable between 7.5 - 13 inches (19 - 34 cm) forearm circumference
Weight	93 grams
Thickness	0.45 inches
Compatible devices	
WINDOWS	7, 8, 10 (with included USB Bluetooth® adapter and OpenGL 2.1 or higher)
MAC	OS X 10.8 (Mountain Lion) and above (with included USB Bluetooth® adapter)
IOS	7.0 and higher
ANDROID	Android 4.3 (Jelly Bean) and up (device must have Bluetooth® radio that supports Bluetooth® 4.0 LE)
Hardware	
Sensors	Medical Grade Stainless Steel EMG sensors, highly sensitive nine-axis IMU containing three-axis gyroscope, three-axis accelerometer, three-axis magnetometer
LEDs	Dual Indicator LEDs
Processor	ARM Cortex M4 Processor
Haptic Feedback	Short, Medium, Long Vibrations
Communication	
Media	Bluetooth® Smart Wireless Technology

Battery	
Power and Battery	Micro-USB charging, Built-in rechargeable lithium-ion battery, one full day use out of a single charge

A Myo armband gives EMG and IMU data. There are 8 EMG signals and 10 IMU signals. Myo armband has 8 EMG sensors, 1 accelerometer sensor, 1 gyroscope sensor, and 1 magnetometer sensor. Figure 3.6 shows all data types of Myo armband.

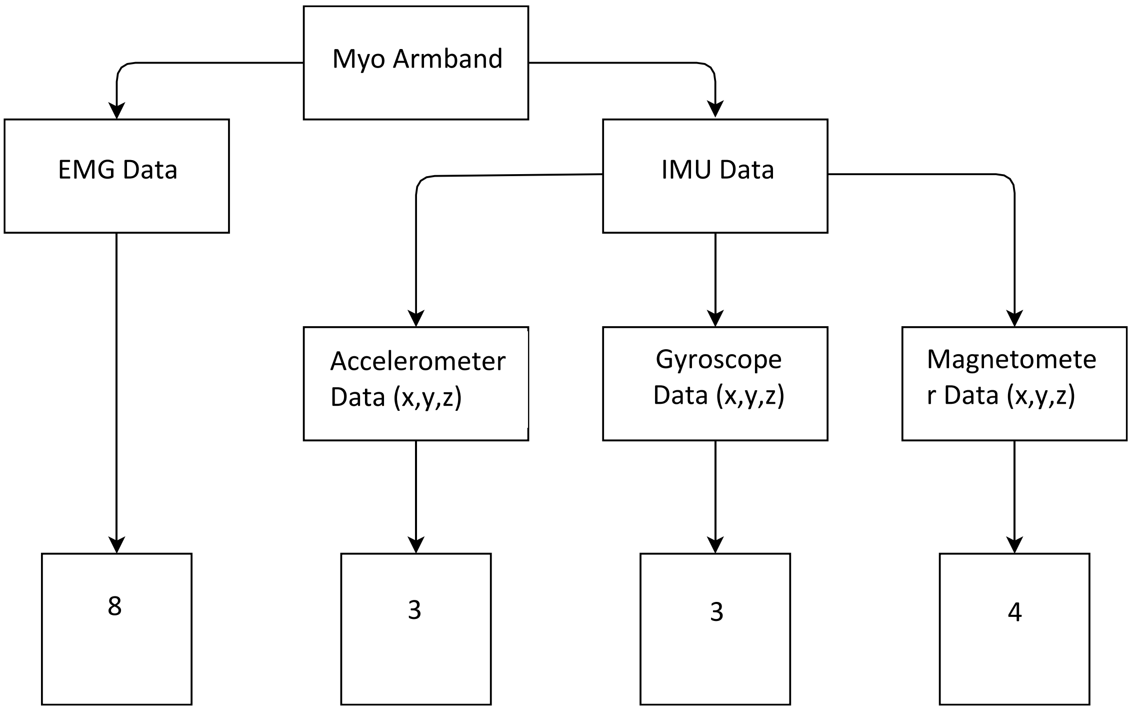


Figure 3.6: Data types of Myo armband

3.2.3 Data Collection

As part of this research project, a deadest was created. Data collection is the most import part in a research project unless having an existing dataset. Because finally, the output is depending on the dataset. After creating the sentences, data were collected using a sign language interpreter. Myo armband was used as our data collection device. Since we use both arms to perform signs, two Myo armbands were used.

It is possible to connect two Myo armbands to a single computer using a single Bluetooth adapter. However, there is an issue when getting EMG data using two armbands which were connected into a single computer using a single Bluetooth adapter. Since the bandwidth of the Bluetooth is small, it is impossible to capture the EMG data using a two Myo armband and a single computer. At the same time, IMU data can be captured without any issue.

Because of the above issue, two Myo armbands were connected to two different computers using two Bluetooth adapters. After connecting armbands with the computers via Bluetooth, by running a C++ program with the help of Myo SDK, EMG and IMU data were captured and stored in CSV files separately.

Correctly position and wear the Myo armband where the 8 sensors must directly touch the skin in order to provide accurate data. The signer has to wear the armband as shown in Figure 3.8 and he had to wear the Myo armbands in the same place on both hands.

To avoid the speed variations when performing signs, a metronome was used as a supporting tool. A metronome is a device that produces an audible click or another sound at a regular interval that can be set by the user, typically in beats per minute. thus, then the signer performs all the sentences (346) in the same rhythm. In the metronome, 5seconds were considered. In each second the signer performed the particular sign according to the sentence. Rest sign was performed in 1^{st} and 5^{th} seconds. The first sign, second sign and third sign in the sentence were performed in 2^{nd}, 3^{rd} and 4^{th} seconds respectively. The metronome was screened in a separate display while performing signs. Figure 3.7 depicts the data collection design.

Moreover, it’s necessary to have a common starting and ending point for all the sentence to recognize a particular sentence when it gets started or ended.

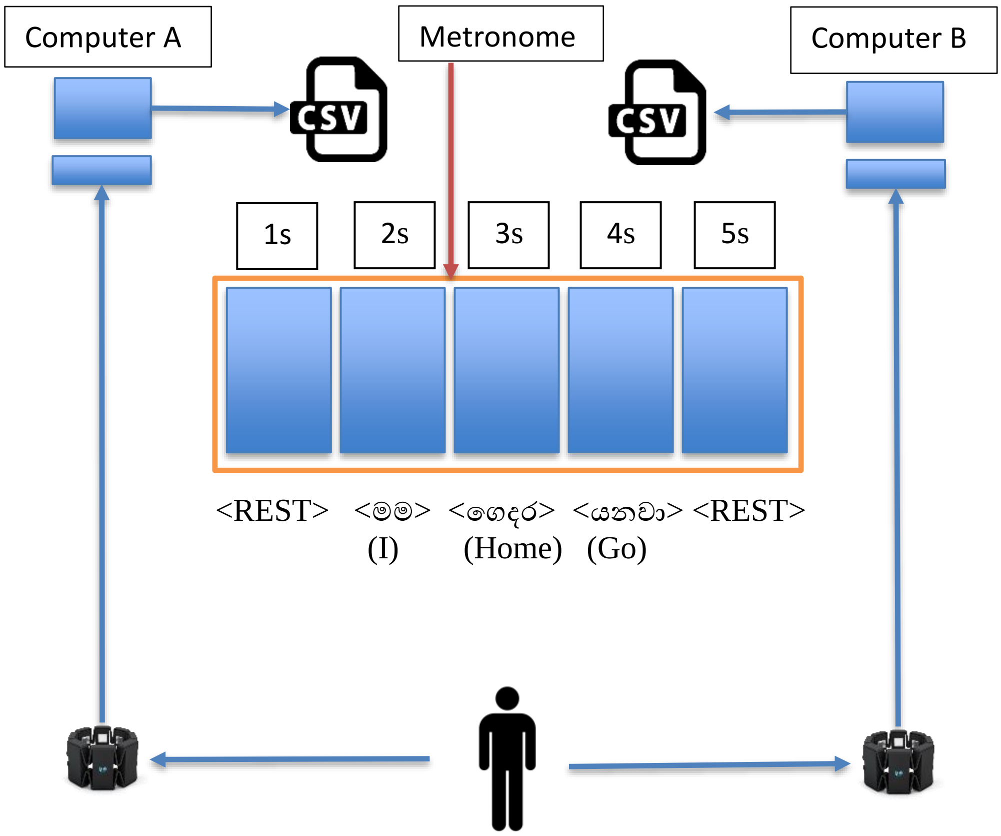


Figure 3.7: Data collection design

Figure 3.8: How to wear a Myo armband.

3.2.4 Data Preprocess

It is not a better idea to use raw data as it is for the classification process. Because there are many issues with the raw data such as unwanted data are there (noise), some data is incomplete, inconsistent etc. After data preprocessing, it is possible to obtain noisy less, complete consistent data. Data goes through a series of steps during preprocessing such as data dleaning, data integration, data transformation, data reduction, data discretization. Here we had EMG data and IMU data. Therefore, we had to use digital signal processing (DSP) techniques to preprocess the collected raw data.

Preprocessing methods of EMG data

1. Resampled the signals
2. Removed the DC offset
3. Applied full wave rectification
4. Used Butterworth Filter
5. Conducted zero-phase digital filtering

Preprocessing methods of IMU data

1. Used Moving average filter

3.2.5 Data Segmentation

After doing data preprocessing, the next most important step was data segmentation. Each sentence has 3 words and each signal contains 3 signs. The aim of this step was to segment each sign separately. As a result of segmentation, there will be 3 segments per sentence. For a single sentence, one Myo armband gives 18 signals (8 EMG and 10 IMU). Since two armbands were used for the data collection, we had to segment 36 signals and saved the segmented signs separately. We can improve the accuracy of the framework by performing a proper segmentation.

We carried out a manual segmentation method. Since we used a metronome as a supporting tool, we knew that the length of a sentence which is 5 seconds and rest sign was performed in the 1^{st} and the 5^{th} seconds. First, second and third signs in the sentence were performed in the 2^{nd}, 3^{rd} and 4^{th} seconds respectively.

Since the 2^{nd}, 3^{rd} and 4^{th} seconds contain the valid signs of a particular sentence. All signals were segmented within that each time period.

3.2.6 Feature Extraction

After segmenting each sentence, there are signal portions for each sign (word). It is not worth, if we input these 36 signals directly into the model for the training. The feature is the single value that represents that whole segment of data. Therefore, features were selected and then, those features can be input into the model.

For this research, below features were extracted from each segmented signal. Features were selected according to the existing work [20].

1. Mean Absolute Value

$$MAV = \frac{1}{N}\sum_{n=1}^{N} |x_n|$$

2. Variance

$$VAR = \frac{1}{N-1}\sum_{n=1}^{N} x_n^2$$

3. Standard Deviation

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

Since we are interested in 3 features, there are 108 (=36*3) features for each sign.

3.2.7 Feature reduction and Feature selection methods

Feature engineering is one of the main tasks of traditional machine learning techniques. Since features are the input for the particular machine learning model. Feature reduction and feature selection methods are some techniques to do this feature engineering. Basically, what it does is, identifying the most important features.

Sometimes there may be correlated features in the main feature set or the dimension of the feature set may be large. Following are some of the reasons to do feature reduction and feature selection.

1. It enables the machine learning algorithm to train faster.
2. It reduces the complexity of a model and makes it easier to interpret.
3. It improves the accuracy of a model if the right subset is chosen.
4. It reduces overfitting.

3.2.7.1 Feature reduction vs Feature selection

Feature selection approaches and feature reduction are very close. However, feature selection allows selecting features among a certain objective function to be optimized without transforming the features.

Feature reduction approaches allow representing features in another space, so the features are transformed.

In this study, we have done below feature reduction and feature selection techniques.

1. PCA - Principal Component Analysis
2. US- Univariate Selection
3. SVD - Singular Value Decomposition
4. RFE - Recursive Feature Elimination
5. RF - Random Forest

3.2.8 Machine learning model training

In this research project, we used supervised learning techniques. Because this is a classification problem. We selected 5 classifiers and trained all the classifiers using the training data. We got the 10-fold cross-validation accuracy of all the models and selected the highest accuracy given classifier as the final classifier for this study.

We selected the following classifiers and got the 10-fold cross-validation accuracies.

1. NB – Gaussian NB
2. LDA – Linear Discriminant Analysis
3. RFC – Random Forest Classifier

4. LR – Logistic Regression
5. RC – Ridge Classifier

3.2.9 Evaluation plan

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. In this research project, supervised learning techniques were conducted. This research problem is a classification problem and few classification models were trained. After that selected the best model by comparing the cross-validation accuracy of each model. Then we had to evaluate the best model. There are three parts of evaluating a classification model.

1. **Accuracy Evaluation**
   A confusion matrix was used for the evaluating the accuracy of the classification model.
2. **Performance Evaluation**
   Receiver Operating Characteristic (ROC) Chart was used to evaluate the performance of the model.
3. **Quality Evaluation**
   Area Under the Curve (AUC) – ROC was used to evaluate the quality of the model.

3.3 Pre-Study

Prior to the main study, we have conducted this pre-study. The main purpose of this pre-study was to identify the research pipeline. Therefore, we have followed all the steps as stated below.

1. Select signs
2. Data collection
3. Data preprocess
4. Data segmentation
5. Extract features

6. Train a few machine learning models
7. Evaluate the models

3.3.1 Select Signs

Since this is a pre-study, three signs were chosen for this study. Figure 3.9, 3.10 and 3.11 show the particular signs. For this study single hand signs were considered due to reducing the complexity of the study and there is no meaning of the signs.

Figure 3.9: Sign 1

Figure 3.10: Sign 2

Figure 3.11: Sign 3

3.3.2 Data Collection

Myo armband was selected as the data collection device. Selected a subject and asked him to perform above three signs in a continuous manner. The subject is not a sign language interpreter thus, he is an ordinary person. The right hand was the dominant arm of that subject. All the signs were performed using his right hand.

Twenty (20) samples were collected. While collecting those signs metronome was used as the supporting tool to preserve the same speed of performing signs. One sample was collected in 5 seconds. In 1st and 5th seconds no signs were performed and in 2nd, 3rd and 4th seconds; sign1, sign2, and sign3 were performed respectively.

- Second 1: No sign performed.
- Second 2: Performed sign 1.
- Second 3: Performed sign 2.

- Second 4: Performed sign 3.
- Second 5: No sign performed

3.3.3 Data Preprocessing

3.3.3.1 EMG Data

Since IMU data is streamed at 50Hz and EMG data is streamed at 200Hz, the EMG data were resampled. After resampling EMG data were smoothed using low pass Butterworth filter.

3.3.3.2 IMU Data

IMU signals were smoothed using moving average filter.

3.3.4 Data Segmentation

A manual segmentation method was performed in this segmentation step. Because signs were performed within 5 seconds and each sign performed in a one second.

- Second 1: No sign performed.
- Second 2: Performed sign 1.
- Second 3: Performed sign 2.
- Second 4: Performed sign 3.
- Second 5: No sign performed.

A MATLAB program was used for the data segmentation process.

3.3.5 Extract Features

Extracted following features for all the segmented signs.

- Mean absolute value
- Standard deviation
- Variance

3.3.6 Model Training

For this pre-study two models were selected.

- Artificial Neural Network (ANN)
- Naïve Bayes Classifier

3.4 Main Study -2

3.4.1 Real-Time Gesture Classification

Real-time hand gesture recognition is one of the most challenging research areas in the human-computer interaction field. As explained in the main study 1, we did the offline training and offline testing. However, in this experiment, our hand gesture recognition system consists of two steps:

1. offline training
2. online testing.

As stated in the main study 1, we can train a model with offline and save that model as a pickle file. Therefore, we don't need to train a classifier again and we can reuse the classifier which has the highest recognition accuracy in the main study 1. However, in the real-time classification scenario, we had to test the model in a real time manner. In order to satisfy that condition, we had to capture the hand gestures in real time. Figure 3.12 shows the flow diagram of the real-time classification study.

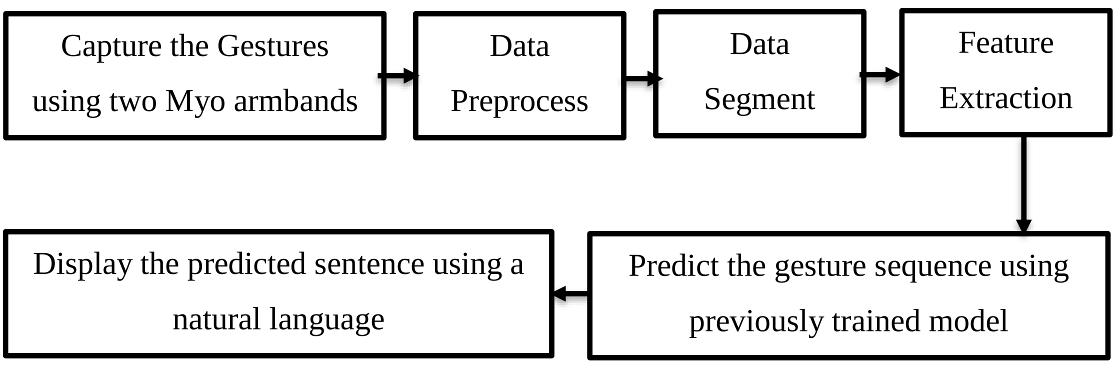


Figure 3.12: Flow diagram of real time classification study

3.4.2 Capture the gestures using two Myo armbands

As the first step of the real-time gesture classification scenario, we had to capture the hand gestures using two Myo armbands. As mentioned in the main study1, we had to use two computers for the data collection process and metronome as a supporting tool.

3.4.3 Data Preprocess/ Data Segment/ Extract Features

After capturing the data, we had to preprocess, segment and extract the features from the raw data. We are using the same techniques which are used for the main study 1.

3.4.4 Predict the gestures using previously trained classifier

After the previous step, we can input the features for the previously trained classifier and we can get the predicted classes for each input. In this step, we used two different classifiers.

1. The classifier which has been trained using all the features and which has the highest testing accuracy.
2. The same classifier which has been trained after feature reduction.

3.4.5 Display the predicted sentence

After getting the names of the predicted classes. Final sentence will be displayed using a word mapping to the screen.

3.4.6 Evaluation of the real time gesture classifier

In this stage, we evaluated the sentence prediction time. Since we use two classifiers to predict the classes, we can compare the effect of reducing feature with the prediction time.

Chapter 4 - Implementation

4.1 Introduction

This chapter elaborates the implementation details of each step of the proposed system. Used software tools are discussed in section 4.2. Section 4.3 and 4.4 discusses the implementation details of the main study-1. Section 4.5 discusses the implementation details of the main study-2.

4.2 Software Tools

The proposed system was created using below software tools and programming languages.

1. C++ programming language and Visual Studio 2017 IDE
2. MATLAB R2017
3. Signal Processing Toolbox™
4. Python 2.7 programming language and jupyter notebook
5. Scikit learn, Skplot python libraries

4.3 Main Study-1 – Experiment 1

4.3.1 Data Collection

Myo armband was selected as the data collection device. Two Myo armbands were used for the data collection process. Each Myo armband was connected to a single computer using a Bluetooth adapter and Myo connect application. It is possible to get the raw data from Myo armband using Myo SDK and using a C++ program. Here, Myo Connect for Windows 1.0.1 version and Windows SDK 0.9.0 were used. Custom C++ program was used to gather the data and Visual Studio 2017 was used as the IDE.

```cpp
void onEmgData(myo::Myo* myo, uint64_t timestamp, const int8_t* emg)
{
    emgFile << timestamp;
    for (size_t i = 0; i < 8; i++) {
        emgFile << ',' << static_cast<int>(emg[i]);

    }
    emgFile << std::endl;
}

void onOrientationData(myo::Myo *myo, uint64_t timestamp,
                        const myo::Quaternion< float > &rotation)
{
    orientationFile << timestamp
        << ',' << rotation.x()
        << ',' << rotation.y()
        << ',' << rotation.z()
        << ',' << rotation.w()
        << std::endl;
}

void onAccelerometerData(myo::Myo *myo, uint64_t timestamp,
                        const myo::Vector3< float > &accel)
{
    accelerometerFile << timestamp
        << ',' << accel.x()
        << ',' << accel.y()
        << ',' << accel.z()
        << std::endl;
}


void onGyroscopeData(myo::Myo *myo, uint64_t timestamp,
                        const myo::Vector3< float > &gyro)
{
    gyroFile << timestamp
        << ',' << gyro.x()
        << ',' << gyro.y()
        << ',' << gyro.z()
        << std::endl;
}
```

The above four code snippets for the collection of EMG, Accelerometer, Orientation and Gyroscope data with the time stamps respectively and save the data in separate CSV files in different folders.

Following openFiles( )function is to create all the files to store the data and each sample stores in a separated folder inside the folder name “Data” with the name of the time stamp. After executing this data collection C++ program, it automatically stops after 5 seconds. Because data is collected for 5 seconds.

```
void openFiles() {
    time_t timestamp = std::time(0);
    start_timestamp = timestamp;

    // Create a Folder
    std::string output_folder = "data";

    std::stringstream ts;
    ts << timestamp;
    std::string output_subfolder = ts.str() ;
    system(("mkdir "+ output_folder + "\\" + output_subfolder).c_str());

    // Open file for EMG log
    if (emgFile.is_open()) {
        emgFile.close();
    }
    std::ostringstream emgFileString;
    emgFileString << output_folder +"/"+ output_subfolder +
                                    "/emg-" << timestamp << ".csv";
    emgFile.open(emgFileString.str(), std::ios::out);
    emgFile << "timestamp,emg1,emg2,emg3,emg4,emg5,emg6,emg7,emg8"
                                    << std::endl;
    // Open file for gyroscope log
    if (gyroFile.is_open()) {
        gyroFile.close();
    }
    std::ostringstream gyroFileString;
    gyroFileString << output_folder + "/" + output_subfolder +
                                    "/gyro-" << timestamp << ".csv";
    gyroFile.open(gyroFileString.str(), std::ios::out);
    gyroFile << "timestamp,x,y,z" << std::endl;

    // Open file for accelerometer log
    if (accelerometerFile.is_open()) {
        accelerometerFile.close();
    }
    std::ostringstream accelerometerFileString;
    accelerometerFileString << output_folder + "/" + output_subfolder +
                                "/accelerometer-" << timestamp << ".csv";
    accelerometerFile.open(accelerometerFileString.str(), std::ios::out);
    accelerometerFile << "timestamp,x,y,z" << std::endl;

    // Open file for orientation log
    if (orientationFile.is_open()) {
        orientationFile.close();
    }
    std::ostringstream orientationFileString;
    orientationFileString << output_folder + "/" + output_subfolder +
                                "/orientation-" << timestamp << ".csv";
    orientationFile.open(orientationFileString.str(), std::ios::out);
    orientationFile << "timestamp,x,y,z,w" << std::endl;

}
```

4.3.2 Data Preprocess

Raw data is noisy, incomplete, and inconsistent. Therefore, data should be preprocessed. In order to preprocess both EMG and IMU data, a MATLAB program was created.

EMG data is streamed at 200Hz but, IMU data is streamed at 50Hz. Because of that EMG data were resampled as the initial step. After resampling a few steps were conducted in order to smooth the EMG signal. Those steps can be listed down as follows.

1. Removed the DC offset
2. Conducted full wave rectification
3. Used low pass Butterworth filter
4. Conducted zero-phase digital filtering

Finally, store the smoothed data in a sperate CSV file. Below MATLAB function shows the implementation of the above steps.

```
function smoothEMG(emgRawDirectory, smoothedDirectory,emgFileName)
    EMG_RAW = csvread(emgRawDirectory ,1,1);

    EMG_RAW = resample(EMG_RAW,1,4); % resamling 200Hz ----> 50Hz 200*(1/4)

    % Step 1 : Detect any DC offset in the collected data set and Remove
    EMG_DC = detrend(EMG_RAW);

    % Step 2 : Full wave rectification
    EMG_REC= abs(EMG_DC);

    % Step 3 :Use low pass filter Butterworth filter
    [b, a] = butter (3, 3/(50/2), 'low');

    % Step 4 :Filter the EMG signal
    EMG_FILTERED = filtfilt (b, a, EMG_REC);

    EMG_IN=EMG_FILTERED;

    dlmwrite(smoothedDirectory+"/"+emgFileName,EMG_IN);

end
```

In order to preprocess the IMU data, a MATLAB program was created. Moving average filter is used for the preprocessing the IMU data and finally smoothed IMU data is stored in separate CSV files.

```
function smoothIMU(imuRawDirectory, smoothedDirectory, imuFileName)
    IMU_RAW = csvread(imuRawDirectory ,1,1);
    IMU_MA= movmean(IMU_RAW,10);

    dlmwrite(smoothedDirectory+"/"+imuFileName,IMU_MA);

end
```

4.3.3 Data Segmentation

Below MATLAB function is responsible for the segmentation of the data. It’s manual segmentation process and the user has to give the segmentation points as parameters of the function sign_segment. Finally, segmented data were stored in separately.

```
function sign_segment(flag, sourceDirectory, destinationDirectory,
            segmentFolder, filename, start_point, end_point, timeStamp)

    smoothed_data = csvread(sourceDirectory);
    subFolder=fullfile(destinationDirectory, segmentFolder);

    if 0==exist(subFolder,'dir')
        mkdir(subFolder);
    end

    if(flag==1)
        segmentedSmoothedData = smoothed_data(start_point:end_point, 1:3);
    elseif(flag==2)
        segmentedSmoothedData = smoothed_data(start_point:end_point, 1:8);
    elseif(flag==3)
        segmentedSmoothedData = smoothed_data(start_point:end_point, 1:3);
    elseif(flag==4)
        segmentedSmoothedData = smoothed_data(start_point:end_point, 1:4);
    end

    segmentedDirectory=fullfile(subFolder, filename);
    dlmwrite(segmentedDirectory,segmentedSmoothedData);

    getFeatures(flag, segmentedDirectory, timeStamp);
end
```

4.3.4 Feature Extraction

In order to extract the features from smoothed and segmented EMG and IMU data, MATLAB program was created. Below features have been selected as the features for this study.

1. Mean Absolute Value
2. Standard Deviation
3. Variance

Below MATLAB function extract the features of all the segmented signs of EMG and IMU data. After extracting features from each signal, the extracted features were stored in separated CSV files.

```
function getFeatures(flag, segmentedDirectory, timeStamp)

    EMG_SEG = csvread(segmentedDirectory);

    feature1 = getmavfeat(EMG_SEG);

    feature2=std(EMG_SEG);

    feature3=var(EMG_SEG);

    n = str2num(timeStamp);

    feature = [ones(1,1)*n feature1 feature2 feature3];

    if 0==exist('acc_features','dir') .
        mkdir('acc_features');
    end
    if 0==exist('emg_features','dir')
        mkdir('emg_features');
    end
    if 0==exist('gyro_features','dir')
        mkdir('gyro_features');
    end
    if 0==exist('ori_features','dir')
        mkdir('ori_features');
    end
```

```
    if(flag==1)
        dlmwrite("acc_features/feature_mav_std_var.csv",feature,
                    'delimiter',',','-append','precision', 10);

    elseif(flag==2)
        dlmwrite("emg_features/feature_mav_std_var.csv",feature,
                    'delimiter',',','-append','precision', 10);

    elseif(flag==3)
        dlmwrite("gyro_features/feature_mav_std_var.csv",feature,
                    'delimiter',',','-append','precision', 10);

    elseif(flag==4)
        dlmwrite("ori_features/feature_mav_std_var.csv",feature,
                    'delimiter',',','-append','precision', 10);
    end

end
```

4.3.5 Machine learning model training

In order to train the model a python script was used and Jupiter notebook was used as the IDE. In that Python script, 5 models were trained and 10-fold cross validation was calculated and finally, it displays the average cross-validation accuracies of each model.

Then selected the classifier which gave the highest cross-validation accuracy as the final classification model for the study. Then using that final model, word level and sentence level accuracies were taken. Relevant code snipped for the model training is shown in the following figures.

```
x_reader = csv.reader(open("all_features_in_a_row.csv", "rb"), delimiter=",")
x = list(x_reader)
result_x = numpy.array(x).astype("float")

y_reader = csv.reader(open("all_classes_in_a_row.csv", "rb"), delimiter=",")
y = list(y_reader)
result_y = numpy.array(y).astype("float")

# Combined the features and labels
c = numpy.c_[result_x, result_y]

# Shuffle the combined matrixes
numpy.random.shuffle(c)

# all Dataset
train = c[:242, :327]
test =c[242:, :327]
```

```
#==============================================================================
#training Dataset
result_x_1_train=train[:, :108]
result_x_2_train=train[:, 108:216]
result_x_3_train=train[:, 216:324]

result_y_1_train=train[:, 324:325]
result_y_2_train=train[:, 325:326]
result_y_3_train=train[:, 326:327]

X_train=numpy.vstack((result_x_1_train,result_x_2_train,result_x_3_train))
Y_train=numpy.vstack((result_y_1_train,result_y_2_train,result_y_3_train)).ravel()

#Combined the features and labels
b = numpy.c_[X_train, Y_train]

#Shuffle the combined matrixes
numpy.random.shuffle(b)

X_train=b[:, :108]
Y_train=b[:, 108:]
#==============================================================================
#testing Dataset
result_x_1_test=test[:, :108]
result_x_2_test=test[:, 108:216]
result_x_3_test=test[:, 216:324]

result_y_1_test=test[:, 324:325]
result_y_2_test=test[:, 325:326]
result_y_3_test=test[:, 326:327]

X_validation=numpy.vstack((result_x_1_test,result_x_2_test,result_x_3_test))
Y_validation=numpy.vstack((result_y_1_test,result_y_2_test,result_y_3_test)).ravel()

#==============================================================================
seed = 7
scoring = 'accuracy'

# Spot Check Algorithms
models = []

models.append(('NB', GaussianNB()))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('RFC', RandomForestClassifier()))
models.append(('LR', LogisticRegression()))
models.append(('RC', RidgeClassifier()))

# evaluate each model in turn
results = []
names = []
for name, model in models:
        kfold = model_selection.KFold(n_splits=10, random_state=seed)
        cv_results = model_selection.cross_val_score(model, X_train,
                                                Y_train, cv=kfold, scoring=scoring)
        results.append(cv_results)
        names.append(name)
        msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
        print(msg)
```

4.4 Main Study-1 – Experiment 2

4.4.1 Feature reduction and Feature selection methods

In order to evaluate the effect of the number of features, we have done some feature engineering techniques. Such as feature reduction and feature selection. We have performed 5 feature reduction and selection techniques.

1. PCA - Principal Component Analysis
2. US- Univariate Selection
3. SVD - Singular Value Decomposition
4. RFE - Recursive Feature Elimination
5. RF - Random Forest

Below code depicts the feature reduction and selection methods which have been performed in this study.

```
# load data
x_reader = csv.reader(open("all_features_no_class_all.csv", "rb"), delimiter=",")
x = list(x_reader)
X = numpy.array(x).astype("float")

y_reader = csv.reader(open("classes_labels_all.csv", "rb"), delimiter=",")
y = list(y_reader)
Y = numpy.array(y).astype("float")

num_features=20; # checked for 20, 40, 60, 80, 100

# feature reduction uv
uv = SelectKBest(score_func=chi2, k=num_features)
uv_fit = uv.fit(X, Y)
uv_features = uv_fit.transform(X)

# feature reduction RFE
model = LogisticRegression()
rfe = RFE(model, num_features)
ref_fit = rfe.fit(X, Y)
ref_features = ref_fit.transform(X)

# feature reduction PCA
pca = PCA(n_components=num_features)
pca_fit = pca.fit(X)
pca_features = pca_fit.transform(X)
```

```
# feature reduction SVD
LA = np.linalg
U, s, Vh = LA.svd(X, full_matrices=False)
assert numpy.allclose(X, numpy.dot(U, numpy.dot(numpy.diag(s), Vh)))
s[num_features:] = 0
svd_features = numpy.dot(U, numpy.dot(numpy.diag(s), Vh))

# feature reduction RFC
clf = RandomForestClassifier(n_estimators=10000, random_state=0, n_jobs=-1)
clf.fit(X, Y.ravel())
sfm = SelectFromModel(clf, threshold=0.01)
sfm.fit(X, Y.ravel())
rfc_features = sfm.transform(X)
```

4.5 Main Study-2

4.5.1 Real-Time gesture classification

As explained in above sections, we used C++ program to collect the data from the two Myo armbands, used MATLAB scripts for data preprocessing, segmentation and feature extraction and a python script for the train the classifiers and reduce the features using feature reduction techniques. Theses C++, MATLAB and python scripts were used in real-time gesture classification scenario as supporting scripts.

In order to classify gestures in real time and display the translation of the performed gesture, we created a python application with a graphical user interface (GUI). Figure 4.1 shows the GUI of the python application.

This application consists of 6 components. Functionalities of each component can be described as follows.

1. **Output display**

   The translated sentence will be displayed in Sinhala text.

2. **Start the server**

   As I explained in earlier, we used two computers to capture the gestures using two Myo armbands. Therefore, there should be a method for the main computer to get the gesture data from the other computer for the further process. In order to that, we can start a python server by clicking this button and we can get the access to the relevant folder of the other computer where 2nd Myo armband stores its data.

3. **Start Data collection**

   By clicking this button, a .exe application (C++ application) will be started and data will be captured form Myo armband. After 5 seconds the application will be stopped automatically. A metronome should be used in this step as a supporting tool in order to preserve the rhythm of performing signs. In order to capture the data from both armbands, this “Start Data collection” button should be clicked in both machines simultaneously.

4. **Start Sign Recognition**

   After clicking this button, following operations will be taken place.

   i. Get the captured data to the main computer from the 2nd computer using the python server.
   ii. Smooth all the raw data.
   iii. Segment all the smoothed data
   iv. Extract the features from each segmented data.
   v. Predict the classes for each segmented data.
   vi. Map the predicted classes with words.
   vii. Display the output in Sinhala natural language.

   In order to smooth, segment and extract the features, few MATLAB scripts which are explained in the previously will be executed.

5. **Close**

   This is for the close the application.

6. **Logger**

   This is to log the important things. Then the user can easily understand what is going correctly and what is going wrong.

The related code segment for this application is reported in Appendix B.

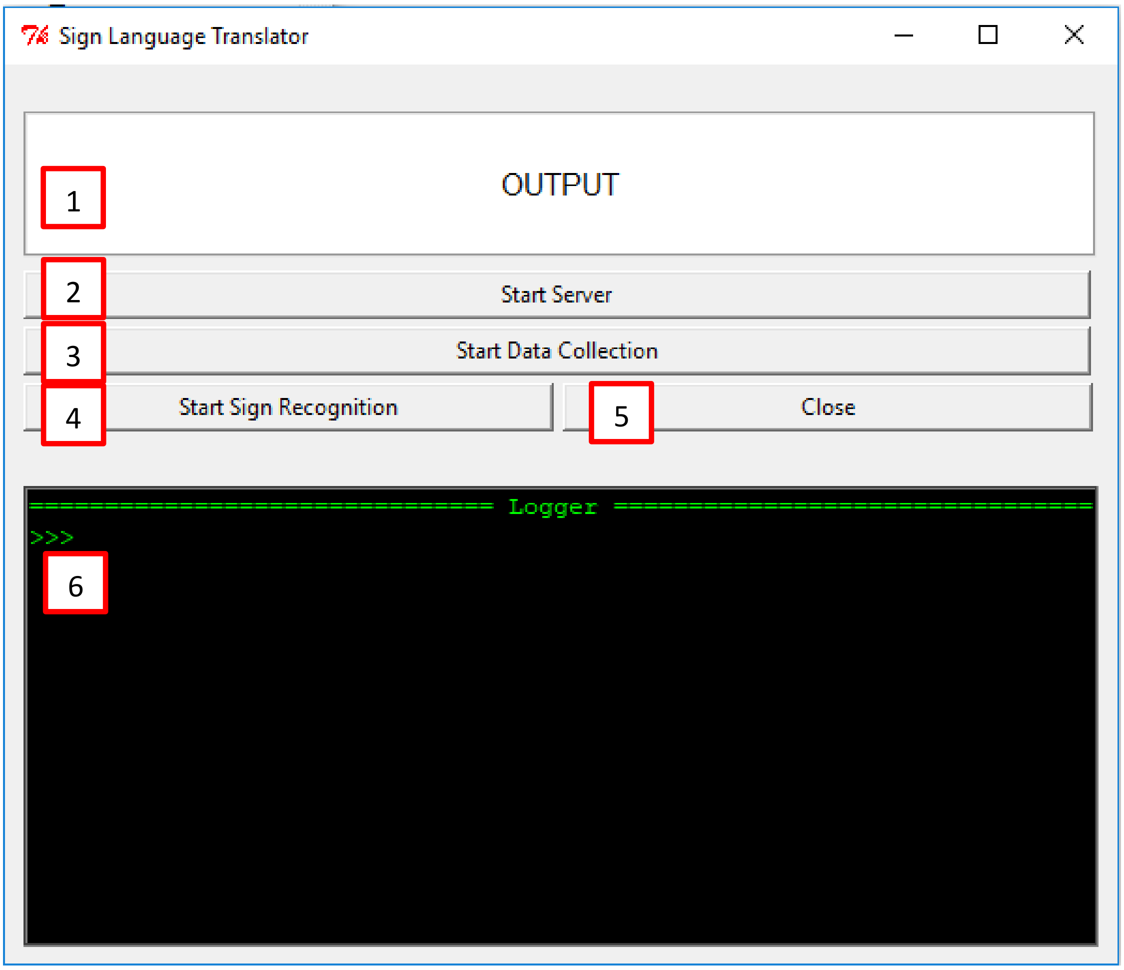


Figure 4.1: Graphical user interface of the python Application.

Chapter 5 - Results and Evaluation

5.1 Introduction

This chapter elaborates how results are evaluated and the success level of the proposed solutions. Section 5.2 discusses the results, evaluation methods and the success level of the pre-study. Section 5.3 and Section 5.4 discuss the results, evaluation methods and success level of main study-1 and main study-2 respectively.

5.2 Pre-study

Before conduct the main study, a pre-study was conducted. The purpose of this pre-study was to identify the research pipeline. Therefore, we collected 20 samples of the three signs which are depicted in below. Each sample was collected by performing these signs (Figure 5.1, 5.2, 5.3) in a continuous manner (At a time, three signs were performed in a continuous manner).

Figure 5.1: Sign 1

Figure 5.2: Sign 2

Figure 5.3: Sign 3

After doing Preprocessing, segmentation and feature extraction, two models were trained. Table 5.1 shows the accuracy of each model. Figure 5.4 and Figure 5.5 show the confusion matrixes of each model.

Table 5.1: Accuracies of the pre-study

Model	Testing accuracy
ANN	20%
Naïve Bayes	85%

	ANN		
	Sign 1	Sign 2	Sign 3
Sign 1	0	0	10
Sign 2	0	0	6
Sign 3	0	0	4

Figure 5.4: Confusion matrix for the pre-study (ANN classifier)

	Naïve Bayes		
	Sign 1	Sign 2	Sign 3
Sign 1	10	0	0
Sign 2	2	4	0
Sign 3	0	1	3

Figure 5.5: Confusion matrix for the pre-study (Naïve Bayes classifier)

5.3 Main Study-1

For the main study-1, 49 signs of Sri Lankan sign language were selected and 346 sentences were created. Then using a single sign language interpreter, data were collected. We used two Myo armbands to capture the signs. Then particular EMG and IMU signals of each sign were smoothed, segmented and extracted the feature. Then experiment-1 and experiment-2 were conducted.

5.3.1 Experiment-1

As the first experiment input all features to each model. As mentioned in the design section, 5 models were trained using 5 Machine Learning algorithms and feature vectors. Then compared the cross-validation accuracies of each model. The cross-validation results were displayed in Table 5.2. Linear Discriminant Analysis classifier has performed the highest average 10-fold cross-validation accuracy.

Table 5.2: Average 10-fold cross-validation score

Model	**Average 10Fold cross-validation score**	**standard deviation**
Logistic Regression (LR)	0.597774	0.047929
Linear Discriminant Analysis (LDA)	**0.761796**	**0.064298**
Ridge Classifier (RC)	0.675114	0.067444
Random Forest Classifier (RFC)	0.603387	0.052306
Gaussian Naïve Bayes (NB)	0.560731	0.067124

Since Linear Discriminant Analysis (LDA) classifier showed the highest cross-validation accuracy (0.761796) and Its standard deviation value (0.064298) is small, It implies that the LDA model is stable. Therefore, we selected LDA as the classifier for the final study. Then we trained the LDA classifier using all the features (108). Finally, we got the word level testing accuracy and it is varying in between 75% - 80%. Sentence level accuracy is varying in between 45% - 50%.

Initially evaluate the accuracy of the LDA classifier. Then evaluate the performance and quality of the LDA classifier. After the evaluation of the classifier, we evaluated how does the position of each sign in the sentences effect to the accuracy of the sentence level gesture recognition.

In order to evaluate the accuracy of the LDA classifier, we used confusion matrix and three measures such as precision, recall, and F1-score. Table 5.3 shows the category (subject, verb or object) of each class label. Figure 5.6 shows the confusion matrix and Table 5.4 shows the precision, recall, and F1-score of each class.

Table 5.3: Category of each class

Class	**Category (subject, verb, object)**
1-18	Subject
19 – 35	Object
36 – 49	Verb

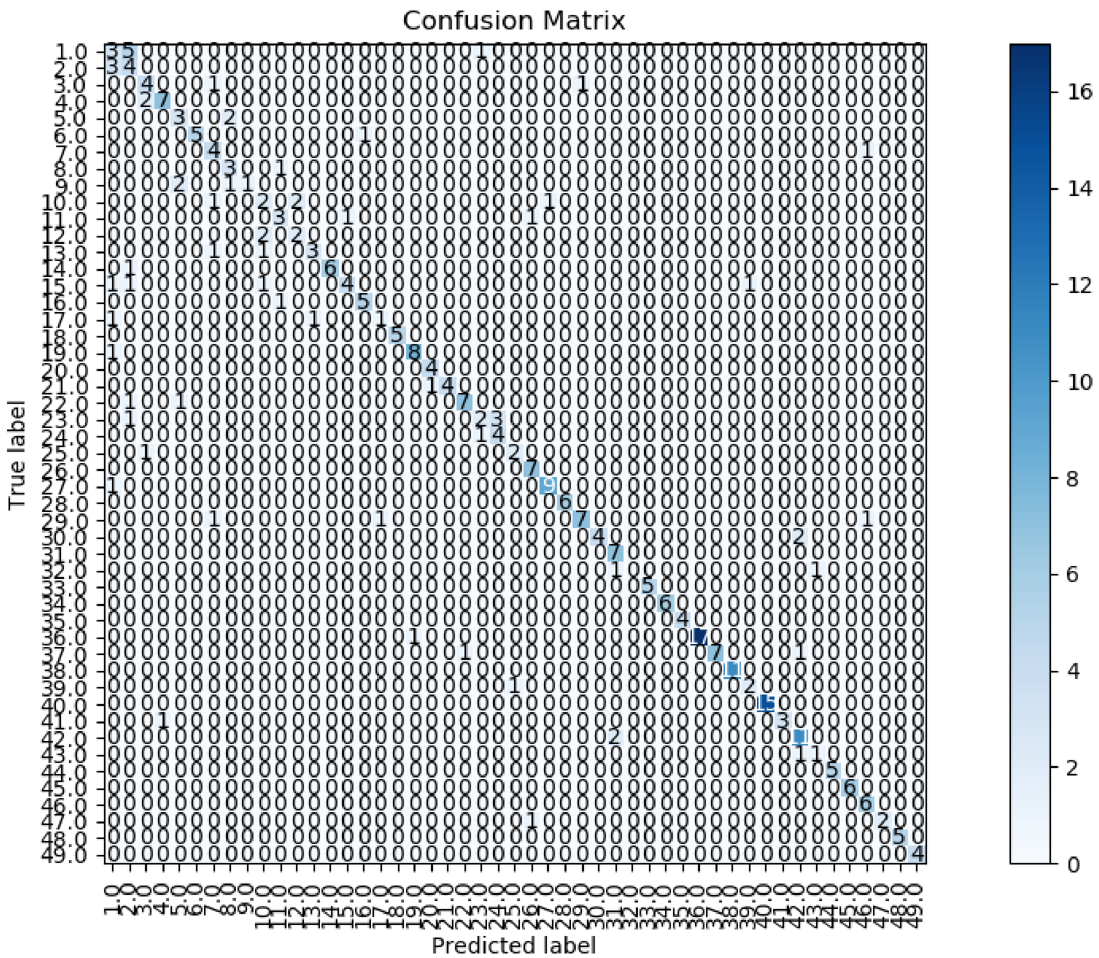


Figure 5.6: Confusion matrix main study 1

Table 5.4**:** Precision, Recall and F1-score values of the main study 1

Class	Precision	Recall	F1-score	Support
1	0.30	0.33	0.32	9
2	0.31	0.57	0.40	7
3	0.57	0.67	0.62	6
4	0.88	0.78	0.82	9
5	0.50	0.60	0.55	5
6	1.00	0.83	0.91	6
7	0.50	0.80	0.62	5
8	0.50	0.75	0.60	4
9	1.00	0.25	0.40	4
10	0.33	0.33	0.33	6
11	0.60	0.60	0.60	5
12	0.50	0.50	0.50	4

13	0.75	0.60	0.67	5
14	1.00	0.86	0.92	7
15	0.80	0.50	0.62	8
16	0.83	0.83	0.83	6
17	0.50	0.33	0.40	3
18	1.00	1.00	1.00	5
19	0.89	0.89	0.89	9
20	0.80	1.00	0.89	4
21	1.00	0.80	0.89	5
22	0.88	0.78	0.82	9
23	0.50	0.33	0.40	6
24	0.57	0.80	0.67	5
25	0.67	0.67	0.67	3
26	0.78	1.00	0.88	7
27	0.90	0.90	0.90	10
28	1.00	1.00	1.00	6
29	0.88	0.70	0.78	10
30	1.00	0.67	0.80	6
31	0.70	1.00	0.82	7
32	0.00	0.00	0.00	2
33	1.00	1.00	1.00	5
34	1.00	1.00	1.00	6
35	1.00	1.00	1.00	4
36	1.00	0.94	0.97	18
37	1.00	0.78	0.88	9
38	1.00	1.00	1.00	11
39	0.67	0.67	0.67	3
40	1.00	1.00	1.00	15
41	1.00	0.75	0.86	4
42	0.73	0.85	0.79	13
43	0.50	0.50	0.50	2
44	1.00	1.00	1.00	5
45	1.00	1.00	1.00	6
46	0.75	1.00	0.86	6
47	1.00	0.67	0.80	3

48	1.00	1.00	1.00	5
49	1.00	1.00	1.00	4
avg / total	**0.81**	**0.79**	**0.79**	**312**

Precision - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all signs that labeled as a correct sign, how many actually correct signs? High precision relates to the low false positive rate. We have got 0.81 average precision value which is pretty good.

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to all observations in actual class. The question recall answers is: Of all the signs that have true class, how many did we label? We have got an average recall of 0.79 which is good for this model as it's above 0.5.

F1-score - F1-score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1-score is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have a similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall. In our case, the average F1-score is 0.79.

However, F1-scores of classes 1,2,5,9,10,12,17,23,32 and 43 are less than 0.60. Especially, F1-score of class 32 is 0.00. The reason would be there are not enough examples (There are only 2 examples). Table 5.5 shows the category of each class which has the F1-score less than 0.60.

Table 5.5: Category of each class which has the F1-score less than 0.60

Classes	**Category (subject, verb, object)**
1, 2, 5, 9, 10, 12, 17	Subject
23, 32	Object
43	Verb

According to the above Table 5.5, we can observe that most of the classes which have F1-score less than 0.60 belong to the subject category. Therefore, we can come to the decision that our model was unable to recognize signs which belong to the subject category. By the way, it is not possible to observe a clear diagonal and values are spread in the class range 1 – 18 in the confusion matrix (Figure 5.6). Therefore, we can confirm that our model was unable to recognize signs which represent the subject category of the sentence compared to other categories by looking at the confusion matrix further.

We evaluated the performance of the system by using a Receiver Operating Characteristic (ROC) Chart. Figure 5.7 shows the ROC chart of the LDA classifier.

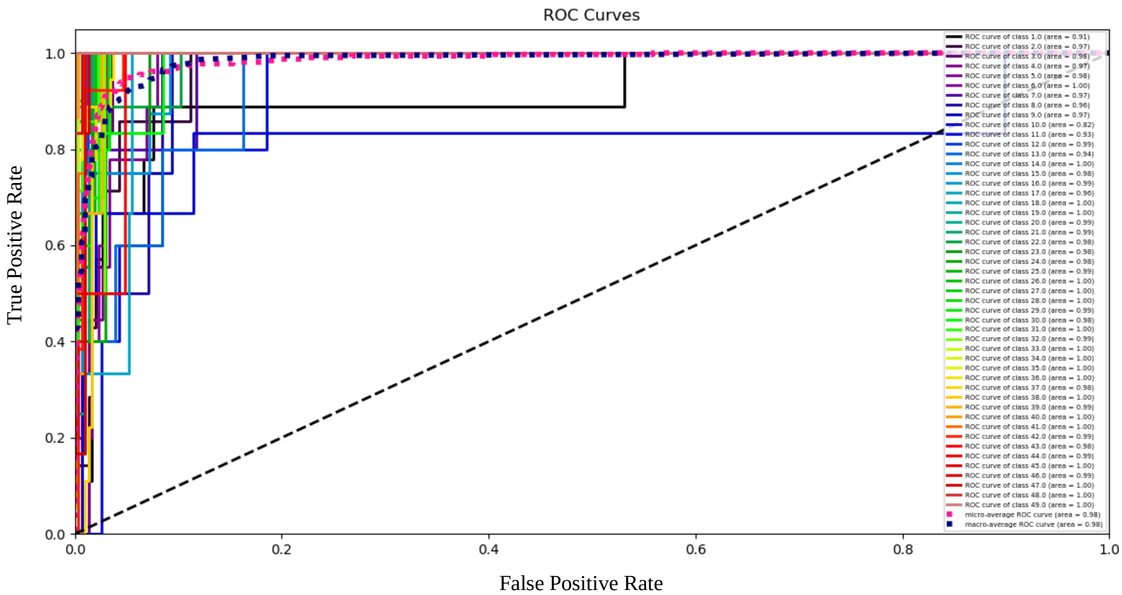


Figure 5.7: ROC Chart

The ROC curve shows the trade-off between sensitivity (or True positive rate) and specificity (or 1 – False positive rate). Classifiers that give curves closer to the top-left corner indicate better performance. In our case, all classes (49 signs) closer to the top-left corner.

Finally, we evaluated the quality of the classifier by using an Area Under the Curve (AUC) – ROC. Table 5.6 shows the AUC-ROC values of LDA classifier.

Table 5.6: AUC-ROC Values

Class	**Area**	**Class**	**Ares**	**Class**	**Area**	**Class**	**Ares**

1	0.91	**2**	0.97	**3**	0.98	**4**	0.97
5	0.98	**6**	1.00	**7**	0.97	**8**	0.96
9	0.97	**10**	0.82	**11**	0.93	**12**	0.99
13	0.94	**14**	1.00	**15**	0.98	**16**	0.99
17	0.96	**18**	1.00	**19**	1.00	**20**	0.99
21	0.99	**22**	0.98	**23**	0.98	**24**	0.98
25	0.99	**26**	1.00	**27**	1.00	**28**	1.00
29	0.99	**30**	0.98	**31**	1.00	**32**	0.99
33	1.00	**34**	1.00	**35**	1.00	**36**	1.00
37	0.98	**38**	1.00	**39**	0.95	**40**	1.00
41	1.00	**42**	0.99	**43**	0.98	**44**	0.99
45	1.00	**46**	0.99	**47**	1.00	**48**	1.00
49	1.00	**micro-average ROC curve**		0.98	**macro-average ROC curve**		0.98

An area of 1 represents a perfect test; an area of 0.5 represents a worthless test. A rough guide for classifying the accuracy of a diagnostic test is the traditional academic point system:

- .90-1 = excellent (A)
- .80-.90 = good (B)
- .70-.80 = fair (C)
- .60-.70 = poor (D)
- .50-.60 = fail (F)

In our case, we have the following statistics (Table 5.7) and our model shows good classification results.

Table 5.7: AUC-ROC results according to the academic point system

Points	**No. of classes (Signs)**
0.90 - 1 = excellent (A)	49 (all signs)
0.80 - 0.90 = good (B)	0
0.70 - 0.80 = fair (C)	0
0.60 - 0.70 = poor (D)	0
0.50 - 0.60 = fail (F)	0

According to Table 5.7, all the AUC-ROC values vary in between 0.90 – 1. Therefore, we can conclude that all classes are closer to the perfect test. According to the traditional academic point system also we can confirm that all the signs belong to the excellent category. Finally, we can conclude that model performed well in word level classification scenario.

As mentioned in previously, the sentence level accuracy is varying between 45% - 50%. A sentence consists of 3 words (subject + object + verb). If at least one of the signs is predicted wrongly, the entire sentence will be classified as a misclassification. Therefore, the meaning of the sentence level accuracy is, 45%-50% of the entire testing sentences are correctly classified and other 55%-50% sentences are misclassified. Table 5.8 shows, how does the position of the sign contribute to the sentence misclassification.

Table 5.8: Contribution of sign's positions for sentences misclassification

1st Sign (Subject)	**2nd Sign (object)**	**3rd Sign (verb)**	**No. of Misclassified Sentences (n)**	**Percentage (n/104)*100%**
Misclassified	Correctly Classified	Correctly Classified	31	29.8%
Misclassified	**Misclassified**	Correctly Classified	5	4.8 %
Misclassified	**Misclassified**	**Misclassified**	1	0.9 %
Misclassified	Correctly Classified	**Misclassified**	2	1.9 %
Correctly Classified	**Misclassified**	Correctly Classified	9	8.7 %
Correctly Classified	**Misclassified**	**Misclassified**	3	2.9 %
Correctly Classified	Correctly Classified	**Misclassified**	3	2.9 %
Number of misclassified sentences (total)			54	51.9 %

According to the above Table 5.8, we can observe that the contribution made by the predicting of the 1^{st} sign to the incorrect veracity of a sentence is higher than all else and it shows a considerable amount (29.8%) compared to other values. Therefore, we can conclude that our model was unable to recognize the 1^{st} sign (Subject of the sentence) correctly relative to the other positions. We have already discussed this issue by looking at the confusion matrix (Figure 5.6) and the F1-score (Table 5.4). Misclassification of one sign directly effects to the sentence level accuracy. We observed that there are two main reasons for the misclassification.

1. **Similarities of signs** -: We noticed that some signs have similar movements. Because of that, some signs are classified incorrectly. This issue can be improved by investigating more distinct attributes for each sign. In Appendix A, few examples of signs which have similar movements in signs are displayed.
2. **Sign segmentation problem**. -: For this study, we assumed that 1^{st} sign, 2^{nd} sign, and the 3^{rd} sign is performed in 2^{nd}, 3^{rd} and 4^{th} seconds respectively with the help of a metronome. With that assumption, we segmented all sentences in each time period. (1^{st}, 2^{nd} and 3^{rd} seconds). However, when we gathering data the sign language interpreter did not perform some signs in the particular time period. Because of that, some signs are not segmented as we expected. We can improve this by using better segmentation methods since we are considering continuous signings.

5.3.2 Experiment-2

As the second experiment, we wanted to experiment, how are feature reduction and feature selection techniques effect to the accuracy of the model. In order to address this question, we selected 5 feature reduction and selection techniques.

1. PCA - Principal Component Analysis
2. US- Univariate Selection
3. SVD - Singular Value Decomposition
4. RFE - Recursive Feature Elimination
5. RF - Random Forest (This method only used at 40 features case.)

We observed the 10-fold cross-validation accuracy of each classifier against with 5 feature reduction and selection techniques.

This experiment was conducted due to observe how feature reduction and feature selection methods effect to the improvement of the machine learning model. In order to do that, we trained 5 machine learning models after reducing the features by 5 feature reduction and selection methods. We reduced the original number of features (108) into 20, 40, 60, 80 and 100 by using the above mentioned 5 methods. However, we were unable to observe any significant improvement of the model, when we trained the model using 20 and 60 features. By the way, models showed significant improvement in 40, 80, 100 features instances. Therefore, we only reported the results of the above mentioned 3 instances in Figure 5.8. Each column represents the average cross-validation accuracy of each model and each row represents a feature reduction method and the number of features which are used to train the model. The last row shows the baseline accuracies of each model.

# Features	Feature Reduction Selection Method	NB Avg	NB SD	LDA Avg	LDA SD	RFC Avg	RFC SD	LR Avg	LR SD	RC Avg	RC SD
40	PCA	0.580723	0.067899	0.628916	0.051059	0.439759	0.058839	0.63253	0.073731	0.536145	0.064937
	SVD	0.53253	0.049909	0.615663	0.060108	0.506024	0.038854	0.589157	0.061776	0.521687	0.067952
	US	0.515663	0.05646	0.590361	0.077709	0.456627	0.040483	0.56988	0.058044	0.512048	0.068844
	RFE	0.626506	0.053067	0.725301	0.032239	0.56988	0.045417	0.655422	0.044822	0.603614	0.048925
	RF - 0.01	0.571084	0.042151	0.757831	0.043256	0.607229	0.048553	0.609639	0.051173	0.546988	0.033735
80	PCA	0.644578	0.037427	0.780723	0.022732	0.422892	0.034269	0.604819	0.060673	0.703614	0.036623
	SVD	0.574699	0.041754	0.768675	0.039521	0.59759	0.051456	0.628916	0.041667	0.703614	0.054203
	US	0.562651	0.035353	0.738554	0.043122	0.572289	0.04005	0.613253	0.045862	0.660241	0.049029
	RFE	0.598795	0.072899	0.762651	0.050559	0.608434	0.078313	0.640964	0.053827	0.681928	0.050025
100	PCA	0.642169	0.043122	0.772289	0.026643	0.384337	0.039022	0.703614	0.052572	0.709639	0.03253
	SVD	0.581928	0.032351	0.791566	0.030976	0.612048	0.049324	0.618072	0.048508	0.712048	0.045225
	US	0.589157	0.031625	0.777108	0.029141	0.590361	0.04572	0.636145	0.045016	0.713253	0.042358
	RFE	0.590361	0.039959	0.789157	0.038948	0.592771	0.039521	0.625301	0.04359	0.684337	0.030384
108	Baseline	0.571084	0.040751	0.763855	0.036623	0.581928	0.051975	0.622892	0.032351	0.685542	0.039022

Figure 5.8: Accuracies of each classifier vs Feature reduction method vs Number of features

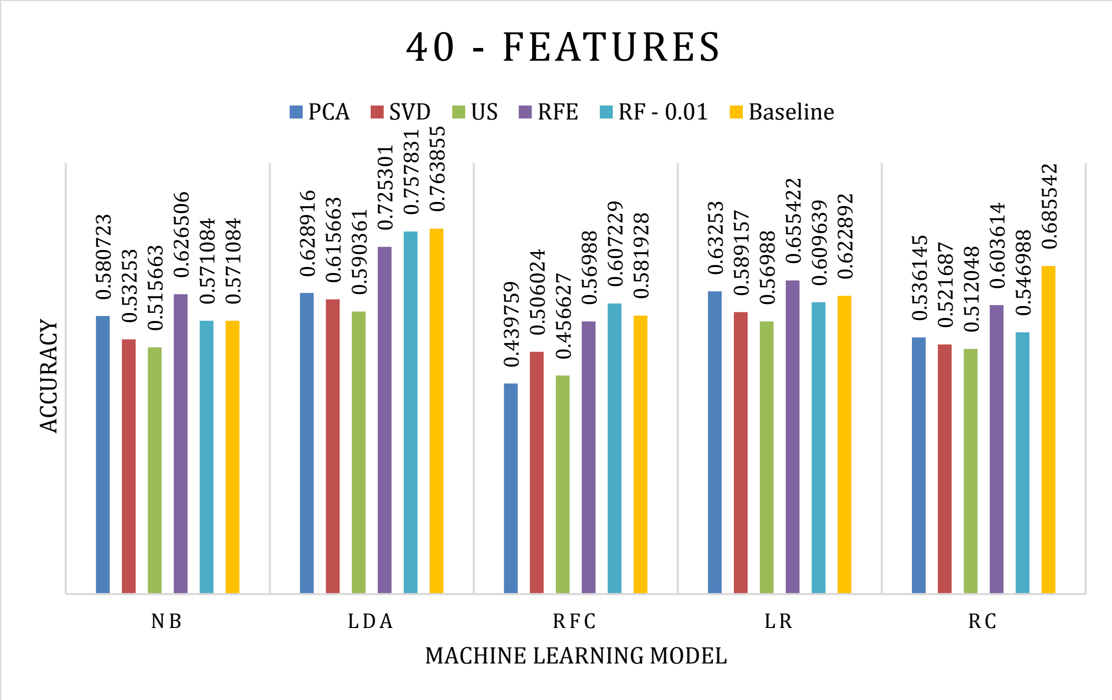


Figure 5.9: Reduce the features to 40 and the cross-validation accuracies of each model

Figure 5.9 shows how does the cross-validation accuracy vary in each model after trained them with 40 features. LDA model shows the highest cross-validation accuracy (0.757831) after reduced the features by random forest method.

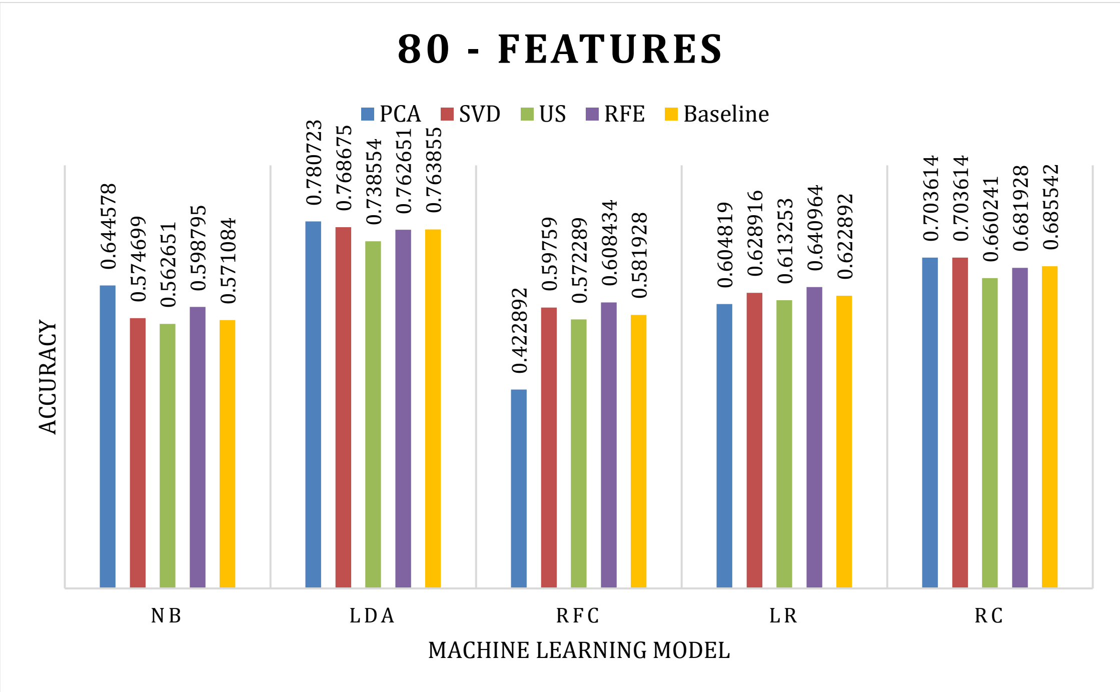


Figure 5.10: Reduce the features to 80 and the cross-validation accuracies of each model

Figure 5.10 shows how does the cross-validation accuracy vary in each model after trained them with 80 features. The LDA model shows the highest cross-validation accuracy (0.780723) after reduced the features by principal component analysis (PCA) method.

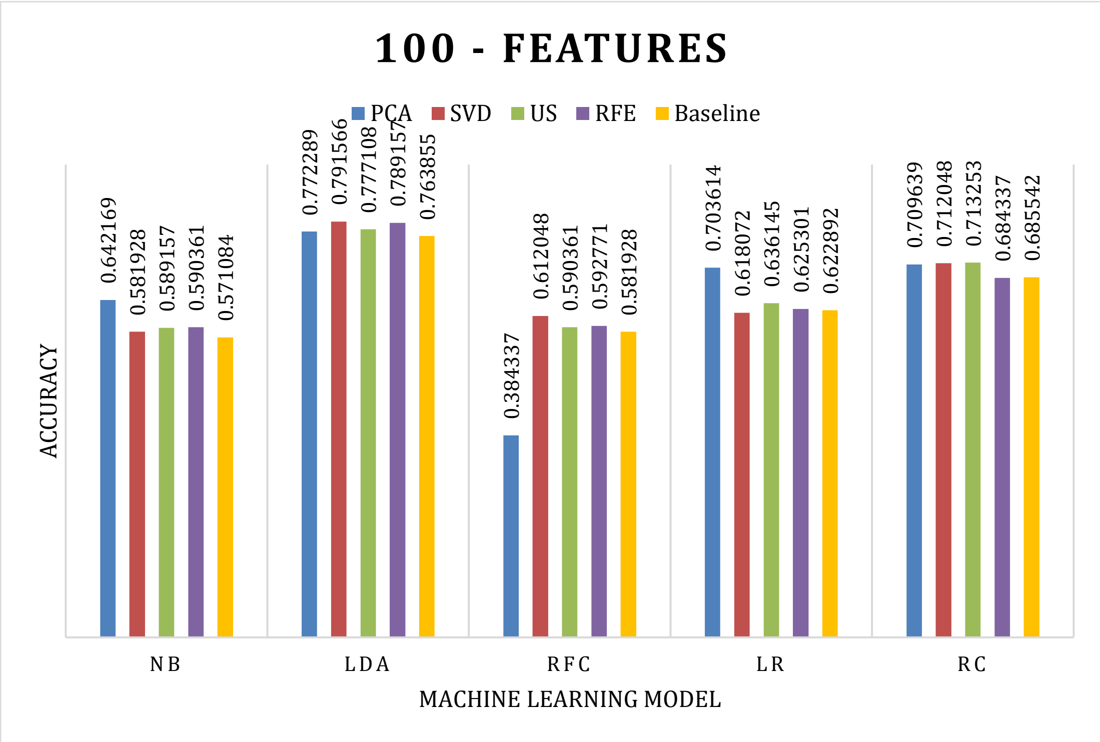


Figure 5.11: Reduce the features to 100 and the cross-validation accuracies of each model

Figure 5.11 shows how does the cross-validation accuracy vary in each model after trained them with 100 features. The LDA model shows the highest cross-validation accuracy (0.791566) after reduced the features by singular value decomposition (SVD) method.

Initially, we had 108 features and accuracy vary between 75-80% (Baseline). However, we observed that the LDA model has the highest cross-validation accuracy even though reduced the features. The accuracies and number of features can be listed down as follows.

- 40 features - 0.757831
- 80 features - 0.780723
- 100 features - 0.791566

However, when we used random forest technique on the same classifier (LDA), we were able to get 75.7831% accuracy by only using 40 features. Even though 80 feature and 100 features instances have slightly high accuracy than 40 feature instances, the number of features is closer to the initial number of features (108). Therefore, we had to consider both model accuracy and the number of features to select the best feature reduction method. Hence, we selected 40 feature reduction method (random forest) and its accuracy as the best feature reduction method and accuracy. Finally, we observed that the training time and the classification time were reduced after the feature reduction by random forest technique.

5.4 Main Study-2

Up to this point, we have done this study as an offline experiment which is all the training data and testing data collected previously. Then preprocess, segment, extract the features from the raw data and finally trained the classifier.

As the main study-2, we wanted to do this study as a real-time classification problem. Because the final goal is to use this system in real-world scenarios. Here, we used two previously trained classifiers.

1. The classifier which has been trained using all the features (108 features) (Classifier-1)
2. The same classifier which has been trained after feature reduction (40 features). (Classifier-2)

After that, we translated the gestures in a real-time manner. Example of a real-time classification output is shown in Figure 5.12. It shows the output as "යාලුවා චිත්‍ර අඳිනවා". (Friend draws paintings).

Average prediction time of a sentence using classifier-1 in real-time -: 17.4 seconds
Average prediction time of a sentence using classifier-2 in real-time -: 13.6 seconds

We can observe that the classifier-2 (LDA) which was trained using 40 features for the sentence prediction, shows less prediction time than classifier-1 (LDA) in a real-time scenario. Even though, that 13.6 seconds of time is not suitable for the real-time scenario, we can observe that the prediction time of a sentence is reduced when we reduced the number of features. Therefore, we can state that there is an effect to the prediction time when we reduced the features.

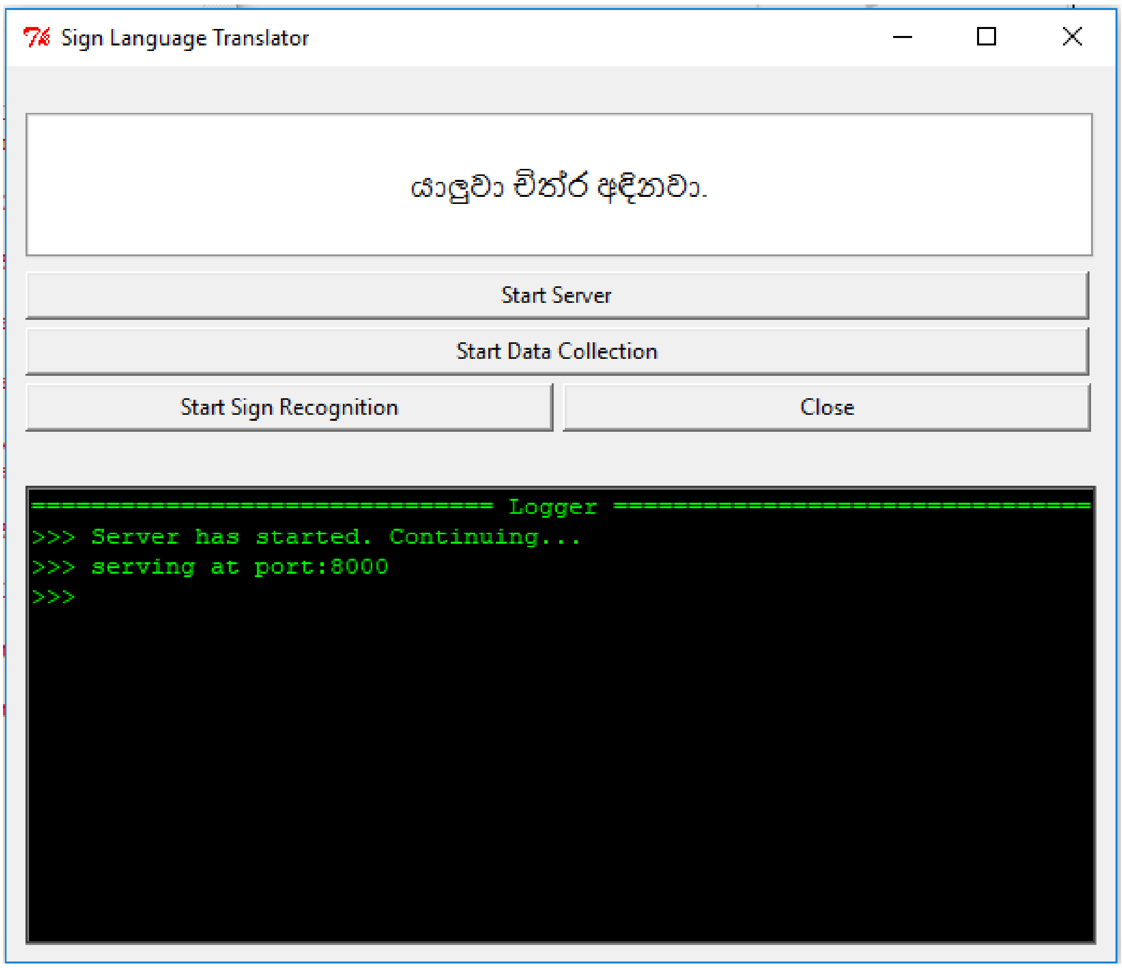


Figure 5.12: Example output of a real-time gesture classification

5.5 Comparison of results of the proposed solution and related work

This research project is an extended version of previous research which has been conducted at University of Colombo School of Computing and title of that publication is “Framework for Sinhala Sign Language Recognition and Translation Using a Wearable Armband” [20] 2016. Prajwal Paudyal et al proposed another work which is SCEPTRE [15] 2016. Table 5.9 shows the comparison between the proposed solution and the above mentioned two main reference research projects ([15], [20]).

Table 5.9: Comparison of the results of the proposed solution and two main references

	Myo Armband [20] (EMG and IMU based solution)	**Myo Armband [15] (EMG and IMU based solution)**	**Proposed Solution**
Sign Language	Sri Lankan Sign Language	American Sign Language	Sri Lanka Sign Language
Word Level	Yes	Yes	Yes
Sentence Level	No	No	Yes
User Dependent	Yes	Yes	Yes
Accuracy)around(	100% (Word Level)	97.72% (Word level)	75%-80% (Word Level) 45%-50% (Sentence Level)
Number of Signs	3	20	49 (Words) 346 (Sentences)
Method	ANN	The multitiered template-based comparison system	Linear Discriminant Analysis
Real-Time	No	Yes	Yes
Real time recognition time	-	0.552 S (Word Level)	13.6 S (Sentence Level)

According to Table 5.9, we can observe that; the proposed solution mainly answers the question which is how to recognize and translate sentence level continuous signing which was not answered by other work. Even though word level accuracy of proposed work is less than the other work, our vocabulary size is greater than other work. Therefore, the word level accuracy of our work is significant. However, the real-time recognition time should be reduced in the proposed work.

Chapter 6 - Conclusions

6.1 Introduction

This chapter includes a review of the research aims and objectives, research problem, limitations of the current work and implications for further research.

6.2 Conclusions about research questions (aims/objectives)

The aim of this research was to bridge the communication gap between hearing/speaking impaired and ordinary people by proposing a framework for recognize sentence level continuous signings of Sri Lankan sing language and translate them into a natural language (Sinhala Language).

Our main research question was to sign language translation for sentence level continuous signings and it is a valid research problem. Because, as explained in Chapter 2, Nobody conducted any research to recognize and translate sentence level continuous signings using an EMG/IMU based wearable device. Even though there are none EMG/IMU based solutions, the usability is minimal and the identification of moment epenthesis is unnatural in natural communication. (E.g.: making pauses between every two signs). Since we used an armband, it improved the usability and we did not use any unnatural method to identify the moment epenthesis as explained in the 3.2.5 subsection.

Followings were the objectives of our research project.

- Study Sri Lankan sign language and its properties to identify the most suitable method and its capability for the recognition and translation.
- Recognize hand and finger gestures with respect to Sri Lankan sign language using a wearable gesture recognition device.
- Propose a framework for translate sentence level continuous signs in Sir Lankan sign language into the natural language (Sinhala Language).

- Implement the proposed framework, obtain a result and evaluate them in order to prove the proposed framework.
- Contribute to the existing body of the knowledge for the progress of this domain.

6.3 Conclusions about research problem

In this research project, we tried to recognize and translate sentence level continuous signings. In order to conduct the research, we created a dataset using a single sign language interpreter and Sri Lankan sign language was selected as the sign language. Then we trained models and got promising results for sign language recognition for both word level and sentence level continuous signings as discussed in Chapter 5.

6.3.1 Contributions of this research

Main contributions of this research project can be listed down as follows.

1. Recognize and translate sentence level continuous signings.
2. Recognize and translate sentence level continuous signings in real time
3. Created a new dataset.

6.3.1.1 Recognize and translate sentence level continuous signings

As explained in early sections, recognition and translate sentence level continuous signings are very challenging problem. Moreover, best of our knowledge nobody has conducted research to recognize and translate sentence level continuous signing using EMG/IMU based wearable devices (Myo armband) locally as well as globally. However, our proposed solution address to this problem using Myo armband which is a wearable device and we selected Linear Discriminant Analysis classifier as the classification algorithm. However, word level accuracy of our system varies between 75-80% and sentence level accuracy varies between 45% - 50%. Our proposed solution shows promising results. Therefore, our contribution to this sign language translation domain is very important.

6.3.1.2 Recognize and translate sentence level continuous signings in real time

Even though recognize and translate sentence level continuous signings in real time is not the main research question, we got the initial step to recognize and translate sentence level continuous signings in real-time manner. We used two classifiers in order to recognize sentence level continuous signings in real time manner which are LDA classifier trained with all features (108) and LDA classifier trained with 40 features (Features reduced by Random Forest method). With all features, it took 17.4 seconds and with 40 features it took 13.6 seconds to predict a sentence. We selected the classifier which trained with 40 features for this real-time scenario. However, that sentence prediction time is quite a high value. We conducted this research using Sri Lankan sign language. Best of our knowledge nobody conducted any research to recognize and translate sentence level continuous signings in real time manner using a Myo armband globally as well. Therefore, our proposed solution contributes to the progress of this domain.

6.3.1.3 Created a new dataset

When we started this research project there was not a proper dataset. Because in this research project we followed a new approach to recognizing and translate sentence level signings using a wearable device. Therefore, a dataset was created and data set consist the signs of Sri Lankan sign language and both hands and single hand used to perform signs. After completing this research project, that dataset will be publicly available. That was a huge contribution to the progress of this domain. The most difficult problem was comparing the accuracies of the previously conducted research with our study. Because each researcher used a different dataset for their own work and the datasets were not publicly available. If a dataset is publicly available, others can use their own methods to increase the accuracy using the same dataset and they can compare the results with other research which use the same data set. Therefore, this contribution will help in the progress of this sign language translation research domain.

The proposed approach for the sign language recognition for the sentence level which was the main research question was answered successfully. Then observed how feature reduction methods effect to the improvement of the classifier accuracy. We employed a feature reduction technique to reduce the sentence prediction time in real time scenario. By the way, the proposed solution improves the usability and mobility, of the system. Because we used the wireless, lightweight wearable device and we did not use any unnatural method to identify the moment epenthesis. Finally, we can conclude that our work answered all the research question as mentioned in section 1.2

6.4 Limitations

For the proposed solution Sri Lankan Sign Language was used as the sign language and this solution cannot be generalized to the other Sign Languages Because syntaxes are different to each Sign Language. However, we only consider the signs which can be performed using hands only. In this research project, three-word sentences are considered. Since a single subject was used for the data collection process, it is not possible to generalize this solution to all users.

6.5 Implications for further research

Our proposed solution showcased a proper outcome based on the scope of the research study which can be extended in several ways.

1. Increase the number of signs.
2. Increase the words per sentence
3. Identify an automatic way to segment the signs.
4. Reduce real-time classification time.

References

[1] "Deafness and hearing loss", *Who.int*, 2018. [Online]. Available: http://www.who.int/en/news-room/fact-sheets/detail/deafness-and-hearing-loss. [Accessed: 01- Jan- 2019].

[2] *Unstats.un.org*, 2016. [Online]. Available: https://unstats.un.org/unsd/demographic-social/meetings/2016/bangkok--disability-measurement-and-statistics/Session-6/Sri%20Lanka.pdf. [Accessed: 01- Jan- 2019].

[3] “KinTrans", *Kintrans.com*. [Online]. Available: http://kintrans.com/#product. [Accessed: 01- Jan- 2019].

[4] "List of sign languages", *En.wikipedia.org*. [Online]. Available: https://en.wikipedia.org/wiki/List_of_sign_languages. [Accessed: 01- Jan- 2019].

[5] K. Stefanov and J. Beskow, "A Real-time Gesture Recognition System for Isolated Swedish Sign Language Signs", *Proceedings of the 4th European and 7th Nordic Symposium on Multimodal Communication (MMSYM 2016), Copenhagen, 29-30 September 2016*, no. 141, pp. 18-27, 2017. [Accessed 1 January 2019].

[6] Wu jiangqin, Gao wen, Song yibo, Liu wei and Pang bo, "A simple sign language recognition system based on data glove", *ICSP '98. 1998 Fourth International Conference on Signal Processing (Cat. No.98TH8344)*. Available: 10.1109/icosp.1998.770847 [Accessed 1 January 2019].

[7] D. Naglot and M. Kulkarni, "Real time sign language recognition using the leap motion controller", *2016 International Conference on Inventive Computation Technologies (ICICT)*, 2016. Available: 10.1109/inventive.2016.7830097 [Accessed 1 January 2019].

[8] M. Maraqa, F. Al-Zboun, M. Dhyabat and R. Zitar, "Recognition of Arabic Sign Language (ArSL) Using Recurrent Neural Networks", *Journal of Intelligent Learning Systems and Applications*, vol. 04, no. 01, pp. 41-52, 2012. Available: 10.4236/jilsa.2012.41004 [Accessed 1 January 2019].

[9] J. Wu, Z. Tian, L. Sun, L. Estevez and R. Jafari, "Real-time American Sign Language Recognition using wrist-worn motion and surface EMG sensors", *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, 2015. Available: 10.1109/bsn.2015.7299393 [Accessed 1 January 2019].

[10] E. Mustafa and K. Dimopoulos, "Sign Language Recognition using Kinect", *Conference: Conference: 9th South East European Doctoral Student Conference*, pp. 271-285, 2014. [Accessed 1 January 2019].

[11] N. Tubaiz, T. Shanableh and K. Assaleh, "Glove-Based Continuous Arabic Sign Language Recognition in User-Dependent Mode", *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 4, pp. 526-533, 2015. Available: 10.1109/thms.2015.2406692 [Accessed 1 January 2019].

[12] D. Kelly, J. Reilly Delannoy, J. Mc Donald and C. Markham, "A framework for continuous multimodal sign language recognition", *Proceedings of the 2009 international conference on Multimodal interfaces - ICMI-MLMI '09*, 2009. Available: 10.1145/1647314.1647387 [Accessed 1 January 2019].

[13] Xu Zhang, Xiang Chen, Yun Li, V. Lantz, Kongqiao Wang and Jihai Yang, "A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors", *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 41, no. 6, pp. 1064-1076, 2011. Available: 10.1109/tsmca.2011.2116004 [Accessed 1 January 2019].

[14] C. Savur and F. Sahin, "American Sign Language Recognition system by using surface EMG signal", *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016. Available: 10.1109/smc.2016.7844675 [Accessed 1 January 2019].

[15] P. Paudyal, A. Banerjee and S. Gupta, "SCEPTRE", *Proceedings of the 21st International Conference on Intelligent User Interfaces - IUI '16*, 2016. Available: 10.1145/2856767.2856794 [Accessed 1 January 2019].

[16] D. M. Wathugala and N. D. Kodikara, "A Sinhala finger spelling interpretation system using nearest neighbor classification", *proc of 4th International infonnation technology coference Envisioning an eNation, Colombo, Sri Lanka*, 2002. [Accessed 1 January 2019].

[17] H. C. M. Herath, W. A. L. V. Kumari, W. A. P. B. Senevirathne and M. B. Dissanayake, "Image based sign language recognition system for Sinhala sign language", *Proceedings of SAITM Research Symposium on Engineering Advancements 2013*, pp. 107-110, 2013. [Accessed 1 January 2019].

[18] N. Kulaveerasingam, S. Wellage, H. M. P. Samarawickrama, W. M. C. Perera and J. Yasas, " The Rhythm of Silence" - Gesture Based Intercommunication Platform for Hearing-impaired People (Nihanda Ridma)", 2014. [Accessed 1 January 2019].

[19] P. Fernando and P. Wimalaratne, "Sign Language Translation Approach to Sinhalese Language", *GSTF Journal on Computing (JoC)*, vol. 5, no. 1, 2016. Available: 10.7603/s40601-016-0009-8 [Accessed 1 January 2019].

[20] A. Madushanka, R. Senevirathne, L. Wijesekara, S. Arunatilake and K. Sandaruwan, "Framework for Sinhala Sign Language recognition and translation using a wearable armband", *2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer)*, 2016. Available: 10.1109/icter.2016.7829898 [Accessed 1 January 2019].

[21] "Sri Lankan Sign Language", *Lankasign.lk*. [Online]. Available: http://www.lankasign.lk/index.html. [Accessed: 01- Jan- 2019].

[22] *Sri Lanka Sign Dictionary*, 2nd ed. SRI LANKA CENTRAL FEDERATION OF THE DEAF, 2012.

[23] "Myo Gesture Control Armband tech specs", *Welcome to Myo Support*. [Online]. Available: https://support.getmyo.com/hc/en-us/articles/202648103-Myo-Gesture-Control-Armband-tech-specs. [Accessed: 01- Jan- 2019].

[24] "What Causes Hearing Loss?", *Verywell Health*. [Online]. Available: https://www.verywellhealth.com/what-causes-hearing-loss-1191926. [Accessed: 01- Jan- 2019].

[25] "How does a deaf person communicate", *Hearingdogs.org.uk*. [Online]. Available: https://www.hearingdogs.org.uk/deafness-and-hearing-loss/how-deaf-people-communicate/. [Accessed: 01- Jan- 2019].

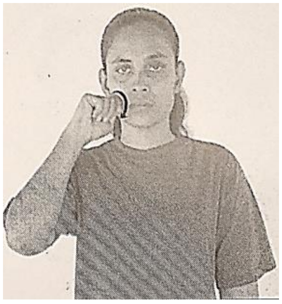
[26] "Sign language", *En.wikipedia.org*. [Online]. Available: https://en.wikipedia.org/wiki/Sign_language. [Accessed: 01- Jan- 2019].

[27] M. Alana Biggers, "Deafness and hearing loss: Causes, symptoms, and treatments", *Medical News Today*. [Online]. Available: https://www.medicalnewstoday.com/articles/249285.php. [Accessed: 01- Jan- 2019].

Appendix A: Diagrams

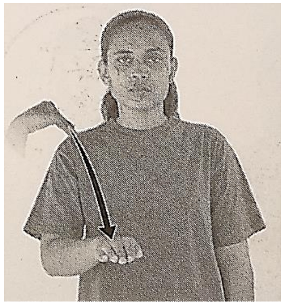
Few examples of signs which have similar movements in signs can be displayed as follows.

අම්මා / Mother



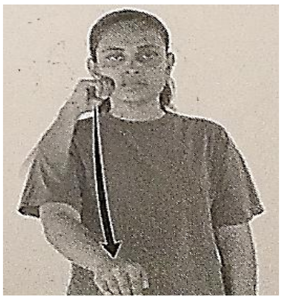
තාත්තා / Father

නංගී / Younger Sister

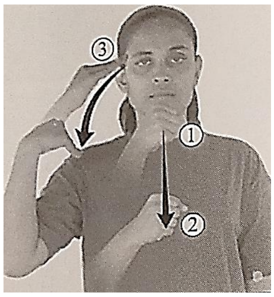


මල්ලී / Younger Brother

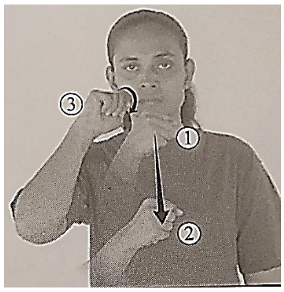
දුව / Daughter



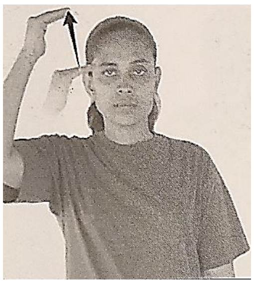
පුතා / Son



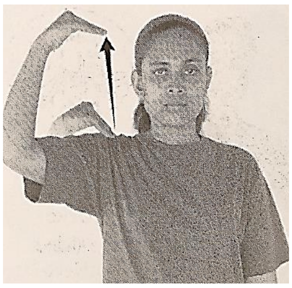
ආච්චි / Grand Mother



සීයා / Grand Father



අක්කා / Elder Sister



අයියා / Elder Brother

Appendix B: Code Listings

A detailed implementation of the real time gesture classification application is provided below.

```
from tkinter import *
top_widget=Tk()
top_widget.geometry('594x480+650+180')
top_widget.title('Sign Language Translator')

label3=Label(top_widget,text='OUTPUT', width=63, height=4,borderwidth=2,
                            relief="groove",bg='white',fg='black',font=(44))
label3.place(x=10,y=25)

T = Text(top_widget, height=15, width=71,borderwidth=2, relief="groove",
                            bg='black',fg='green')
T.place(x=10,y=225)
T.insert(END,
"=============================== Logger ===============================\n>>> ")

#========================== Main Function ==============================

def main():
    #===================== Delete previous files =======================
    import time,os
    import shutil



    if (os.path.isdir('Left/')== True): # if this folder exist -> then remove it.
        shutil.rmtree('Left/')
    if (os.path.isdir('Right/')== True):
        shutil.rmtree('Right/')
    if (os.path.isdir('data/')== True):
        shutil.rmtree('data/')
    if (os.path.exists('all_features_no_class.csv')==True):
        os.remove('all_features_no_class.csv')
    if (os.path.exists('all_features_no_class_left.csv')==True):
        os.remove('all_features_no_class_left.csv')
    if (os.path.exists('all_features_no_class_right.csv')==True):
        os.remove('all_features_no_class_right.csv')
```

```
os.mkdir("Left")
os.mkdir("Right")
os.mkdir("data")
os.mkdir("data/right")
os.mkdir("data/left")
os.mkdir("data/right/1")
os.mkdir("data/left/1")


#==============================================================================
```

```
#===================== Download CSV file from server ========================
import bs4
import requests
import socket
hostname = socket.gethostname()
IPAddr = socket.gethostbyname(hostname)

left_url = "http://192.168.1.3:8000/ServerData/left/1/"
r_l = requests.get(left_url)

files_l=['p','q','r','s','t']

data_l = bs4.BeautifulSoup(r_l.text, "html.parser")
i_l=0
for l_l in data_l.find_all("a"):
    r_l = requests.get(left_url + l_l["href"])
    #print(r_l.status_code)

    with open("data/left/1/"+l_l["href"],'wb') as files_l[i_l]:
        files_l[i_l].write(r_l.content)
    i_l=i_l+1
#-----------------------------------------------------------------------------
right_url = "http://192.168.1.4:8000/ServerData/right/1/"
r_r = requests.get(right_url)

files_r=['p','q','r','s','t']

data_r = bs4.BeautifulSoup(r_r.text, "html.parser")
i_r=0
for l_r in data_r.find_all("a"):
    r_r = requests.get(right_url + l_r["href"])
    #print(r_r.status_code)

    with open("data/right/1/"+l_r["href"],'wb') as files_r[i_r]:
        files_r[i_r].write(r_r.content)
    i_r=i_r+1

#==============================================================================
```

```
#=========================== Run MATLAB script ===============================

import matlab.engine
eng = matlab.engine.start_matlab()

# Left hand
eng.EMG_IMG_alltogether_left(nargout=0)

# Right hand
eng.EMG_IMG_alltogether_right(nargout=0)

# Combine Right and Left
eng.join_right_left(nargout=0)

#==============================================================================


#========================== Predict the sentence ============================
import pickle
import csv
import numpy
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score

filename = 'Lda.sav'

# load the model from disk
loaded_model = pickle.load(open(filename, 'rb'))

# List
Slist = ["මම", "අපි", "ඔහු","ඇය","ඔවුන්","අම්මා","තාත්තා","අක්කා","අයියා",
        "නංගී","මල්ලී","දුව","පුතා","නැන්දා","මාමා","ආච්චි","සීයා","යාලුවා",
        "මේසය","පුටුව","බර","චිත්‍ර","දොර","ජනේලය","ඉංග්‍රීසි","පොත",
        "පත්තරය","කඩදාසිය","වතුර","මාලු","එළවළු","රෙදි","දර","ගෙදර","ගස",
        "අඳිනවා","අඳිනවා","අරිනවා","ඉගෙනගන්නවා","ඉරනවා","උණුකරනවා",
        "උයනවා","එල්ලනවා","ඔසවනවා","පලනවා","යනවා","බොනවා","ලියනවා",
        "විසිකරනවා","සිටුවනවා"]

x_reader = csv.reader(open("all_features_no_class.csv", "rb"), delimiter=",")
xxx = list(x_reader)
result_x = numpy.array(xxx).astype("float")

x=(int)(loaded_model.predict([result_x[0]]))
y=(int)(loaded_model.predict([result_x[1]]))
z=(int)(loaded_model.predict([result_x[2]]))

#Print the predicted stentence
text_T= Slist[x-1].decode('utf-8') +" "+ Slist[y-1].decode('utf-8') +" "+
        Slist[z-1].decode('utf-8') +"."

print  text_T
#============================ main Function END ==============================
```

```python
#========================== Server Function ==============================
from socket import *
import SimpleHTTPServer
import SocketServer
import thread

port = 8000

def create_server():

    handler = SimpleHTTPServer.SimpleHTTPRequestHandler
    httpd = SocketServer.TCPServer(("", port), handler)
    print("serving at port:" + str(port))
    httpd.serve_forever()
def start_server():
    thread.start_new_thread(create_server, tuple())
    print("Server has started. Continuing..")
    T.insert(END, "Server has started. Continuing...\n>>> ")
    T.insert(END, "serving at port:" + str(port)+"\n>>> ")
#========================== Server Function END ===========================

#========================== Stop Function =================================
def stop():
    top_widget.destroy()
#========================== Stop Function END =============================

#==================== Data Collection Function ============================
import sys, string, os
def dataCollect():
    T.insert(END, "Start data collecting...\n>>> ")
    os.system(r'"datacollector\MyoInputDataProject.exe"')
#========================== Stop Function END =============================

btn1=Button(top_widget,text='Start Server', width=80, height=1,
                                    command=start_server).place(x=10,y=110)
btn2=Button(top_widget,text='Start Data Collection', width=80, height=1,
                                    command=dataCollect).place(x=10,y=140)
btn3=Button(top_widget,text='Start Sign Recognition', width=39,
                                            command=main).place(x=10,y=170)
btn4=Button(top_widget,text='Close', width=39,
                                            command=stop).place(x=298,y=170)

top_widget.mainloop()
```