03-134142-106 M. FURQAN KHAWAJA

03-134142-045 MUHAMMAD SHAHRUKH

Mute Life Envoy

In partial fulfilment of the requirements for the degree of

Bachelor of Science in Computer Science

Supervisor: Summaira Nosheen

Department of Computer Sciences

Bahria University, Lahore Campus

June 2018

Certificate

We accept the work contained in the report titled

“MUTE LIFE ENVOY”,

written by

M. FURQAN KHAWAJA

MUHAMMAD SHAHRUKH

as a confirmation to the required standard for the partial fulfilment of the degree of Bachelor of Science in Computer Science.

Approved by:

Supervisor: Ms. Summaira Nosheen

____ (Signature)

June 4th, 2018

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations, which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-134142-106	M. FURQAN KHAWAJA	
03-134142-045	MUHAMMAD SHAHRUKH	

Date : ___________________________

Specially dedicated to
my beloved mother and father

(M. Furqan Khawaja)

my beloved mother and father

(Muhammad Shahrukh)

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express my gratitude to my research supervisor, Madam Summaira Nosheen for her invaluable advice, guidance and her enormous patience throughout the development of the research.

In addition, we would also like to express my gratitude to our loving parent and friends who had helped and given me encouragement.

M. Furqan Khawaja
Muhammad Shahrukh

MUTE LIFE ENVOY

ABSTRACT

Improving human computer interaction emphasized the research in gesture recognition field. Researchers have done a lot of research work in Gesture Recognition field for different languages. And many gesture recognizers are already in the market for different languages. But unfortunately International Sign Language has received less attention. In Pakistan, India and Middle East, there are more than hundred million people, which are deaf and mute, and having difficulty for communicating with normal people. As literacy rate is very low in these areas, there is barrier of technological learning/awareness; therefore, we intend to propose Gesture Recognizer.

In Gesture recognition, Gesture is converted to text that makes it easy to communicate. This project deals with the study of processing techniques of International Sign Language and conversion of gestures with the computing methodologies to achieve a gesture recognizing system for communication. This system would help user to communicate with any one, which is deaf or mute without getting dependent on any interpreter or person who translate their signs. Our overall goal is the implementation of existing gesture recognition techniques and develops a gesture recognizer for International Sign Language (Alphabets) using existing technologies. Recommendations for future development and conclusions are also included in the report.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

DNM	Deaf and mute
MLE	Mute Life Envoy
LMC	Leap Motion Controller
IS	International Signs
ASL	American Sign Language
PSL	Pakistani Sign Language
HCI	Human Computer Interaction
SDK	Software Development Kit
API	Application Programming Interface
PUN	Photon Unity Networking
SQL	Structured Query Language
OOP	Object Oriented Programming

CHAPTER 1

INTRODUCTION

1.1 Background

Hand gesture is the most effective, dominant and basic way to interact or communicate with DNM and general public. They commonly interact with each other by using their region sign language only. This is the most uncomfortable and lesser easy for communication with the people of other region having their own language(s). They prefer to interact with general public and computer just like normal people. Hence, expanded enthusiasm of researchers in enhancing human computer interaction underlined exploration in field of hand gesture recognition as the Deaf and Mute (DNM) communication menu-driven user interface with machines are presently getting to be well known.

1.2 Problem Statement

Nowadays, gesture recognition applications are becoming beneficial. Various gesture intuitive applications are available in market but unfortunately, majority focus on American Sign Language (ASL) or other pidgins. MLE handle with the processing techniques of ISL and incorporates the semantics to achieve hand gesture acknowledgment framework that will translate gestures to words. The user input gestures will be interpreted leading to transmission of interpretation to other end user. General objective is to implement existing gesture acknowledgment techniques to develop hand gesture recognition system for International Sign Language (ISL) using existing technologies.

1.3 Aims and Objectives

Project manages with study of leap motion controller's infrared sensors processing techniques along hand motion for static gestures and incorporates the semantics along figuring approaches to achieve a hand gesture detection framework to interact through peer-to-peer communication. The aim is to apply gesture recognition approaches to develop a recognizer framework for International Sign Language (ISL). The objectives for project are as below:

i) To define and characterize the static hand gestures based on IS language using leap motion controller.
ii) To characterize the interpreted gesture into English text words/sentences.
iii) To study the principles responsible for reading and recognising techniques using leap motion controller.
iv) To study peer-to-communication interface for chat.
v) To study machine learning techniques for hand gesture validation.

1.4 Scope of Project

There is no application related to hand gesture recognition for International Signs language (ISL). By observing the expanding interest of hand gesture recognition in industry, a hand gesture recognizer is being proposed to dictate and transmit gestures from ISL to English words/sentences. Our product is fundamentally for DNM individuals to communicate with each other and general public independently without any physical interpreter. The proposed framework benefits DNM individuals and they can use MLE system in various perspectives to get ease in their day-by-day life. The system will also allow general public to communicate with DNM effectively and efficiently excluding their pidgin issues. Hence, MLE as gesture recognizer for ISL is proposed to defeat all the issues by lowering the rate of error as well.

CHAPTER 2

LITERATURE REVIEW

2.1 Background

Hand gesture is the most effective, dominant and basic way to interact or communicate with DNM and general public. They commonly interact with each other by using their region sign language only. This is the most uncomfortable and lesser easy for communication with the people of other region having their own language(s). They prefer to interact with general public and computer just like normal people. So, expanded enthusiasm of researchers in enhancing human computer interaction underlined the exploration in the field of hand gesture recognition. The DNM communications through menu-driven user interface with machines are presently getting to be well known where gestures are transformed in sequence of words.

2.2 Need for Hand Gesture Recognition

Gesture recognition alludes to the procedure of transformation of human hand gestures in a sequence of words (English). These words are perceived and after that can be utilized as command(s) to operate/control systems, for typing documents and chatting purposes. Gesture recognition also alludes to the association amongst individuals and computer machines by utilizing gestures as contribution to framework causing the system to perform specific tasks in the wake of perceiving the input.

Expanded enthusiasm of researchers in enhancing human computer interaction underlined the exploration in the field of hand gesture recognition. The DNM communications through menu-driven user interface with machines are presently getting to be well known. There is a set of number of individuals who know how to get to operate computers and access them hence utilizing more efficient interface for communication using sign language. Extensive number of individuals would not be able to communicate with others without the advancement in this field so, gesture recognition framework advantage DNM individuals.

2.3 Application of Hand Gesture Recognition

Hand gesture recognition also said as processing of gesture-to-text. Generally, gesture-to-text processing involves the below steps:

i) Gesture Read
ii) Data Processing
iii) Gesture Recognition
iv) Interpreted Gesture Transmission

2.3.1 Gesture Read

Some data has to be prepared before the hand gesture recognition process can be started. The hand's interior structural details are received through the hardware by perusing hand gesture in digital machine language for computer that is converted for the understanding to recognizer.

2.3.2 Data Processing

A DNM individual when makes static hand gesture within range of the infrared sensors, the gesture recognition process starts. Information gestures are converted by infrared sensor(s) to digital language that can be cached (after verification) in computer. Recognizer, using some information for conversion of digital input signals, converts digital signals to generate some meaningful strings of English text and caches the sequence in preferences before transmitting text

2.3.3 Gesture Recognition

The input sequences are matched with computer stored fragments. It looks through all potential outcomes produced using gesture data processing phase thus this process finds the counterpart for our input by applying most proficient algorithm.

2.3.4 Interpreted Gesture Recognition

In MLE, the interpreted gestures, which have been translated into English text, are stored in the database before transmission. These are then transmitted through Internet services and protocols from one device to another. To accomplish this there must be tenets for every transmission convention.

2.4 Challenges To Mute Life Envoy System

Developing a static hand gesture recognition system is a challenging task. That includes following challenges:

- Substantial vocabulary could be problem.
- Is there a continuous signal capacity in our framework?
- Do we have constrained environmental conditions?
- Hardware limitation could be problem
- Do any algorithm has 100% accuracy?

Researchers in the gesture recognition field are trying to resolve the above mentioned challenges.

Advancement of computational techniques for the transformation of an information signal into set of words is the reason for MLE framework. At the moment, no generic recognition framework exists that acknowledges user-invented sign language of different district or the International Sign (IS) Language. In this way, below factors are limited for a hand gesture recognition framework to handle issues.

2.4.1 Language

Gesture recognition systems are language specific i.e. they are trained for a certain language. Dialects in any language lead to different pronunciations. So particular dialect should be defined in a gesture recognition system. Furthermore, different people have different dialects, which may vary from each other. So MLE system should be trained as pidgin sign language for different people having different dialects.

2.4.2 Environment

Predetermined positioning for gesture, as there is a wide variation in hands and fingers.

2.4.3 Vocabulary Size

Vocabulary measure fluctuates for various frameworks. Some systems utilize small vocabulary; some have extensive vocabulary thus size of vocabulary must be resolved for MLE framework to ensure precision

2.5 Gesture Recognition Classification

Based on following gesture approaches, speaker's class and vocabulary size, gesture recognition systems can be differently categorized. In development of hand gesture recognition are numerous challenges. These are described briefly as below:

2.5.1 Gesture Recognition Approaches

Hand gesture recognition systems are built on the basis of following types of approaches (R.Pradipa):

i) Template Matching
ii) Feature Extraction Analysis
iii) Active Shapes Model
iv) Principal Component Analysis
v) Linear Fingertip Models
vi) Casual Analysis

2.5.1.1 Template Matching

"The simplest method for recognizing hand postures is through Template matching. The template matching is a method to check whether a given data record can be classified as a member of a set of stored data records. Recognizing hand postures using template matching has two parts. The first is to create the templates by collecting data values for each posture in the posture set. The second part is to find the posture template most closely matching the current data record by comparing the current sensor readings with the given set" (R.Pradipa) [6].

2.5.1.2 Feature Extraction Analysis

"The low-level information from the raw data is analysed in order to produce higher-level semantic information and is used to recognize postures and gestures are defined as Feature Extraction and Analysis. The system recognized these gestures with over 97% accuracy. It is a robust way to recognize hand postures and gestures. It can be used to recognize both simple hand postures and gestures and also complex ones as well" [6].

2.5.1.3 Active Shapes Model

"A technique for locating a feature within a still image is called Active shape models or "smart snakes". A contour on the image that is roughly the shape of the feature to be tracked is used. The manipulation of contour is done by moving it iteratively toward nearby edges that deform the contour to fit the feature" [6].

"Active shape model is applied to each frame and use the position of the feature in that frame as an initial approximation for the next frame" [6].

2.5.1.4 Principal Component Analysis

"A statistical technique for reducing the dimensionality of a data set in which there are many interrelated variables is called Principal Component Analysis where retaining variation in the dataset. Reduction of data set is by transforming the old data to a new set of variables that are ordered so that the first few variables contain most of the variation present in the original variables. By computing the eigenvectors and eigenvalues of the data set's covariance matrix the original data set is transformed. When dealing with image data is that it is highly sensitive to position, orientation, and scaling of the hand in the image" [6].

2.5.1.5 Linear Fingertip Models

"This is a model that assumes most finger movements are linear and comprise very little rotational movement. The model uses only the fingertips as input data and permits a model that represents each fingertip trajectory through space as a simple vector. Once the fingertips are detected, their trajectories are calculated using motion correspondence. The postures themselves are modelled from a small training set by storing a motion code, the gesture name, and direction and magnitude vectors for each of the fingertips. The postures are recognized if all the direction and magnitude vectors match (within some threshold) a gesture record in the training set. System testing showed good recognition accuracy (greater than 90%), but the system did not run in real time and the posture and gesture set should be expanded to determine if the technique is robust" [6].

2.5.1.6 Casual Analysis

"A vision-based recognition technique that stems from work in scene analysis is known as Causal Analysis. The technique extracts information from a video stream by using high-level knowledge about actions in the scene and how they relate to one another and the physical environment. The gesture filters normalize and combine the features and use causal knowledge of how humans interact with objects in the physical world to recognize gestures. The system captures information on shoulder, elbow and wrist joint positions in the image plane. From these positions, the system extracts a feature set that includes wrist acceleration and deceleration, work done against gravity, size of gesture, area between arms, angle between forearms, nearness to body, and verticality. Gesture filters normalize and combine the features and use causal knowledge of how humans interact with objects in the physical world to recognize gestures such as opening, lifting, patting, pushing, stopping, and clutching. There is no clarity how accurate this method is. This system also has the disadvantage of not using data from the fingers. More research needs to be conducted in order to determine if this technique is robust enough to be used in any nontrivial applications" [6].

2.6 Hand Gesture Types

Following are the hand gesture types on the basis of their motion:

i) Static Hand Gesture: The gestures that have relative position to the body.
ii) Dynamic Hand Gesture: The gestures that have motion speed or direction to the body.

2.7 Brief Overview of International Sign Language

"Deaf people in the Western and Middle Eastern world have gathered together using sign language for 2,000 years and communication variety of sign language arose from this span, whether it is in an informal personal context or in a formal international context. Deaf people have therefore used a kind of auxiliary gestural system for international communication at sporting or cultural events since the early 19th century. The need to standardize an international sign system was discussed at the first World Deaf Congress in 1951, when the WFD was formed. In the following years, a pidgin developed as the delegates from different language backgrounds communicated with each other, and in 1973, a WFD committee ("the Commission of Unification of Signs") published a standardized vocabulary. They selected "naturally spontaneous and easy signs in common use by deaf people of different countries" to make the language easy to learn. A book published by the commission in the early 1970s, Gestuno: International Sign Language of the Deaf, contains a vocabulary list of about 1500 signs. The name "Gestuno" was chosen, referencing gesture and oneness" [27].

"However, when Gestuno was first put into service, at the WFD congress in Bulgaria in 1976, it was incomprehensible to deaf participants. Subsequently, it was developed informally by deaf and hearing interpreters, and came to include more grammar — especially linguistic features that are thought to be universal among sign languages, such as role shifting and the use of classifiers. Additionally, more iconic signs and loan signs from different sign languages gradually replaced the vocabulary" [27].

"The first training course in Gestuno was conducted in Copenhagen in 1977 to prepare interpreters for the 5th World Conference on Deafness. Sponsored by the Danish Association of the Deaf and the University of Copenhagen, the course was designed by Robert M. Ingram and taught by Betty L. Ingram, two American interpreters" [27].

"The name Gestuno has fallen out of use, and the phrase "International Sign" is now more commonly used in English to identify this sign variety. Indeed, current IS has little in common with the signs published under the name 'Gestuno' " [27].

A parallel development has been occurring in Europe in recent year, where increasing interaction between Europe's deaf communities has led to the emergence of a pan-European pidgin [27].

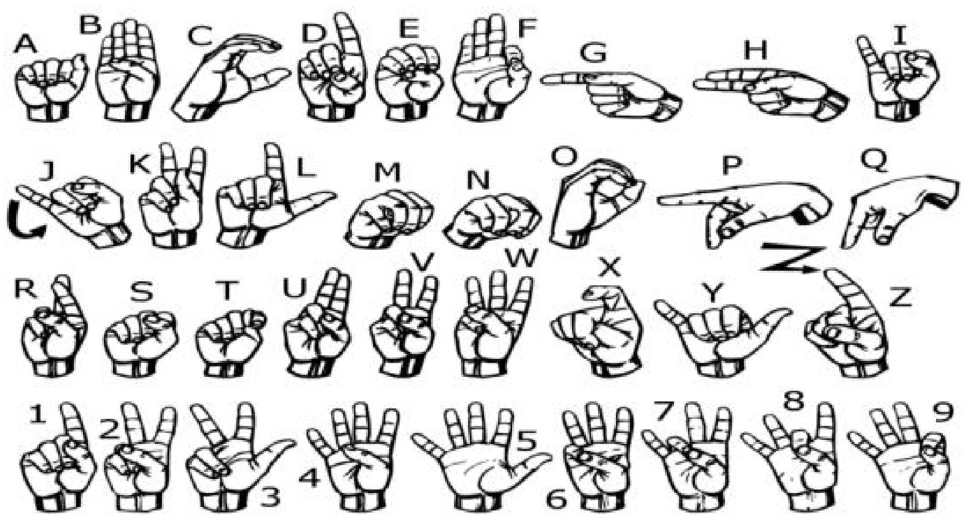


Figure 1: Internal Signs Language [28]

2.8 List of Hand Gesture Recognition

Expanded enthusiasm of researchers in enhancing human computer interaction underlined the exploration in the field of hand gesture recognition. The DNM communications through menu-driven user interface with machines are presently getting to be well known. The milestone of hand gesture recognition system is as follows:

i) Sign Language Recognition (SLARTI)
ii) Hand Motion Understanding (HMU)
iii) Enable Talk
iv) ASL Translator
v) iCommunicator
vi) Coffee for Yawns
vii) Samsung Switching Channels
viii) Building Automaton System
ix) Motion Savvy

2.9 Comparison of Hand Gesture Recognition Applications

➢ Hand Gesture Recognition – Analysis of Various Techniques, Methods and Their Algorithms [6].

Author:	R.Pradipa, Ms. S.Kavitha (Mar, 2014)
Recognition technique:	Template Matching, Active Shapes Model, Linear Fingertip Models (>90%), Casual Analysis, Principal Component Analysis, Feature Extraction Analysis (97% accuracy)
Models:	Hidden Markov Model and camshift algorithm

➢ Static And Dynamic Hand Gesture Recognition In Depth Data Using Dynamic Time Wrapping [7].

Author:	Guillaume Plouffe and Ana-Maria Cretu (Feb, 2016)
Recognition technique:	Dynamic Time Wrapping
Models:	
Language:	American Sign Language
Accuracy:	92.4%

➢ Real-time, Static and Dynamic Hand Gesture Recognition for Human-Computer Interaction [8].

Author:	S.M. Hassan Ahmeda, T. C. Alexander and G. C. Anagnostopoulos
Recognition technique:	Features from Accelerated Segment Test (FAST)
Models:	
Language:	American Sign Language
Accuracy:	75%

➢ Two-Handed Sign Language Recognition for Bangla Character Using Normalized Cross Correlation [10].

Author:	Kaushik Deb, Helena Parvin Mony & Sujan Chowdhury
Recognition technique:	Template Matching Technique
Models:	
Language:	Bangla sign language
Accuracy:	96%

➢ Real-time American Sign Language Recognition with Convolutional Neural Networks [11].

Author:	Brandon Garcia, Sigberto Alarcon Viesca
Recognition technique:	Convolutional Neural Network
Models:	
Language:	American Sign Language
Accuracy:	74%

➢ Real-Time and Robust Method for Hand Gesture Recognition System Based on Cross-Correlation Coefficient [12].

Author:	Reza Azad, Babak Azad and Iman tavakoli kazerooni
Recognition technique:	Image segmentation, Morphological Filtering, Cross-correlation based feature extraction and matching
Models:	
Language:	American Sign Language
Accuracy:	98.34%

➢ Visual Recognition of American Sign Language Using Hidden Markov Models [13].

Author:	Thad Eugene Starner
Recognition technique:	Hidden Markov Method
Models:	
Language:	American Sign Language
Accuracy:	95%

➢ Artificial Neural Network Based Method for Indian Sign Language Recognition [14]

Author:	Adithya V. Vinod P. R, Usha Gopalakrishna (2013)
Recognition technique:	Artificial Neural Network
Models:	
Language:	Indian Sign Language
Accuracy:	91.11%

➢ An Artificial Neural Network Approach For Sign Language Vowels Recognition [15]

Author:	V.V. NABĐYEV, C. KÖSE, S. BAYRAK (2013)
Recognition technique:	Artificial Neural Network
Language:	Sign Language Vowels
Accuracy:	96%

➢ Pakistan Sign Language Recognition Using Statistical Template Matching [16]

Author:	Aleem Khalid Alvi, M. Yousuf Bin Azhar, Mehmood Usman, Suleman Mumtaz, Sameer Rafiq, Razi Ur Rehman, Israr Ahmed
Recognition technique:	Artificial Neural Network
Language:	Pakistan Sign Language

➢ Recognition of Tamil Sign Language Alphabet Using Image Processing To Aid Deaf-Dumb People [9]

Author:	P. Subha Rajam, G. Balakrishnan
Recognition technique:	Pattern Recognition, Feature Point Extraction
Models:	
Language:	Tamil Sign Language
Accuracy:	96.87% (static), 98.75 (dynamic)

➢ Sign Language To Text Converter Using Leap Motion Controller [1]

Author:	P. Subha Rajam, G. Balakrishnan
Recognition technique:	Geometric Template Matching, Artificial Neural Network, Cross-Correlation
Models:	
Language:	American Sign Language
Accuracy:	52.56% (GTM), 44.87% (ANN), 35.90% (Cross-Correlation)

➢ Dynamic Hand Gesture Recognition With Leap Motion Controller [17]

Author:	W.Lu, Z.Tong, J. Chu
Recognition technique:	Hidden Conditional Neural Field (HCNF)
Models:	
Language:	American Sign Language
Accuracy:	

➢ Gesture Recognition With The Leap Motion Controller [18]

Author:	R.McCartney, J.Yuan, Hans-Peter. Bischof
Recognition technique:	Convolutional Neural Network
Models:	
Language:	American Sign Language
Accuracy:	89.5%

➢ Hand Movement And Gesture Recognition Using Leap Motion Controller [19]

Author:	Lin Shao
Recognition technique:	LMC API
Models:	
Language:	
Accuracy:	Medium

➢ Using The Leap Motion Controller To Translate Sign Language To Speech [4]

Author:	T.C. Yan, L.K. Chun, C.Y. Laam and T.W.Yin
Recognition technique:	Dynamic Time Wrapping
Models:	
Language:	
Accuracy:	Medium

- Transforming Indian Sign Language Into Text Using Leap Motion [20]

Author:	P. Karthick, N. Parthiba, V.B. Rekha, S. Thanalaxmi
Recognition technique:	Dynamic Time Wrapping
Models:	
Language:	
Accuracy:	Medium

- Research on Chinese-American Sign Language Translation [21]

Author:	Z. Xiaomei, D. Shiquan, W. Hui
Recognition technique:	Dynamic Time Wrapping
Models:	
Language:	Chinese-American Sign Language
Accuracy:	

Pakistan Sign Language Recognition and Translation System Using Leap Motion Controller [22]

Author:	N. Raziq, S. Latif
Recognition technique:	Artificial Neural Network
Models:	
Language:	Pakistan Sign Language
Accuracy:	92.5%

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 Development Plan

Mute Life Envoy consists of two development stages i.e. Gesture Recognition and Interpretation Transmission; both are further divided into phases. The Iterative and Incremental development practices model will be used throughout the workflow to meet the goals and deadlines. The development stages are discussed as below:

The first stage is of Gesture Recognition that consists of four phases. The first phase “Gesture Read” prepares the data from taking input to converting it into computational language. Second phase “Gesture Verification” verifies the input data as recognized IS gesture. Third phase “Gesture-to-Text Interpretation” converts the recognized input gesture into English format text and create a sequence of words and sentences. Fourth phase “Text Storage” is the phase that stores the interpreted text into database for secure transmission to the other user.

The second stage is Interpretation Transmission that consists of three phases. First phase is “Text Read”. It reads the stored text in database and encrypts the data. Second phase “Text Transmission” that transmit the encrypted data. Third phase is “Text Display” that displays the transmitted text over the other user interface.

Once all the components will be developed, the integration of all the components will be focused after the above mentioned phase and will be followed by acceptance testing of the deliverable.

3.2 System Design

3.2.1 Use Case

3.2.1.1 Level_0

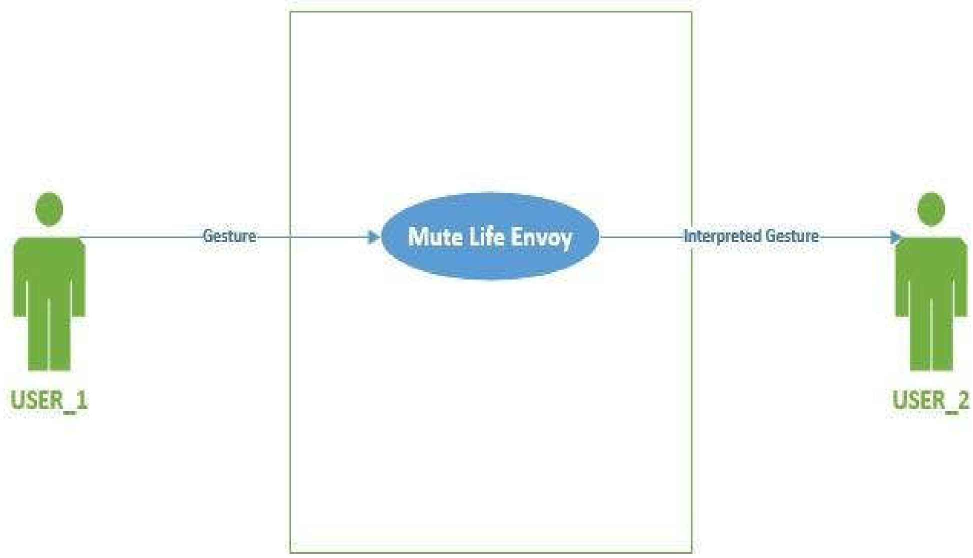


Figure 2: Use Case - System View

3.2.1.2 Use Case Description

Brief Description:

The user will start the application and will perform the gestures and the interpreted gestures will be displayed to the other user.

Preconditions:

The application must be installed on the computer

Basic Flow:

1. User_1 starts applications
2. User_1 establish connection
3. User_1 provides input

Alternative Flows:

1.a. Application didn`t start

2.a. Connection didn't established

3.a. User changes input method

3.b. User closes application

Post Conditions:

The user will provide input.

3.2.1.3 Level_1

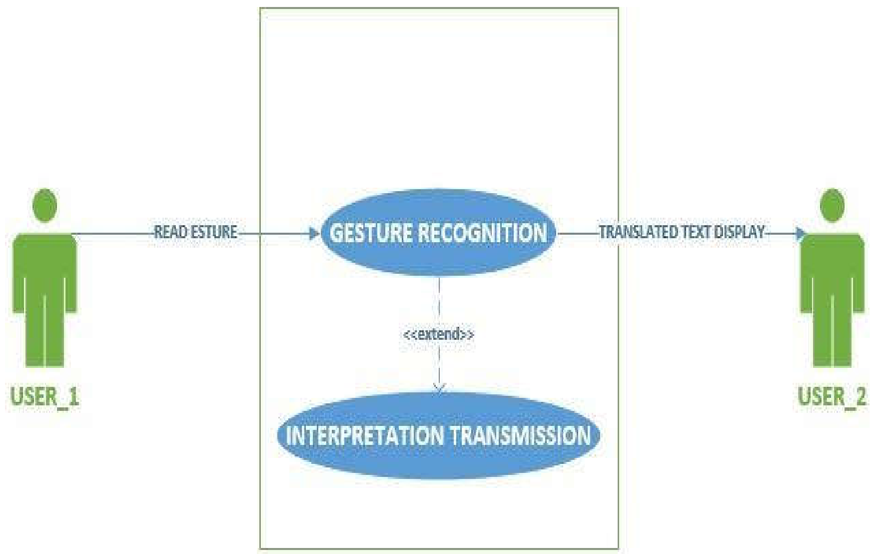


Figure 3: Use Case - System Inner View

3.2.1.4 Use Case Description

Brief Description:

The user will start the application and will perform the gestures. These gestures will be taken as input by the application where they will be interpreted as text which will be displayed to the other user.

Preconditions:

User record gesture in consideration with particular height and angle as input

Basic Flow:

i) User input gesture
ii) Gesture recognize

iii) Gesture interpretation into text form
iv) Text display to the other user

Alternative Flow:

i) An incorrect input
ii) A Gesture did not recognize
iii) A Gesture did not interpret into text form
iv) A Text didn't display on the other end

Post Conditions:

Input converted into text form

Notes/Issues:

The user must record the gesture in consideration with particular height and angle as input else failed attempt will occur.

3.2.1.5 Level_2

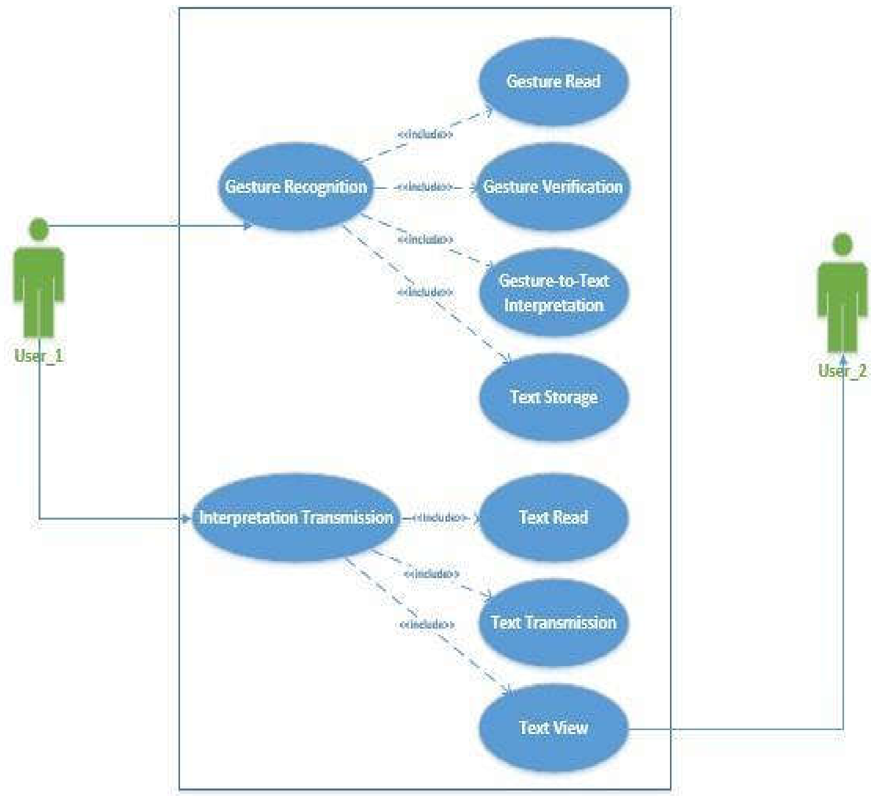


Figure 4: Use Case - System Detailed View

3.2.1.6 Use Case Description

Brief Description:

The user will start the application and will perform the gestures. These gestures will be taken as input by the application, where they will be verified, interpreted into text form and stored in the database. These interpreted gestures will be displayed to the other user.

Preconditions:

User must enter/scan input

Basic Flow:

1. User starts application
2. User establish connection
3. User provide input
4. Gesture read
5. Gesture verified after read
6. Gesture interpreted into text form
7. Storing of text
8. Getting text from database
9. Transmission of text
10. Text display to the other user

Alternative Flow:

1.a. Application didn`t start

2.a. Connection didn’t established

3.a. User changes input method

3.b. User closes application

4.a. Gesture didn’t read

5.a. Gesture did not verify after read

6.a. Gesture did not interpret into text

7.a. Text didn’t store

8.a. Text didn’t get from storage

9.a. Text didn’t transmit

10.a Text didn’t display to the other end

Post Conditions:

Input converted into text form and then display to the other user.

3.2.2 Work Breakdown Structure

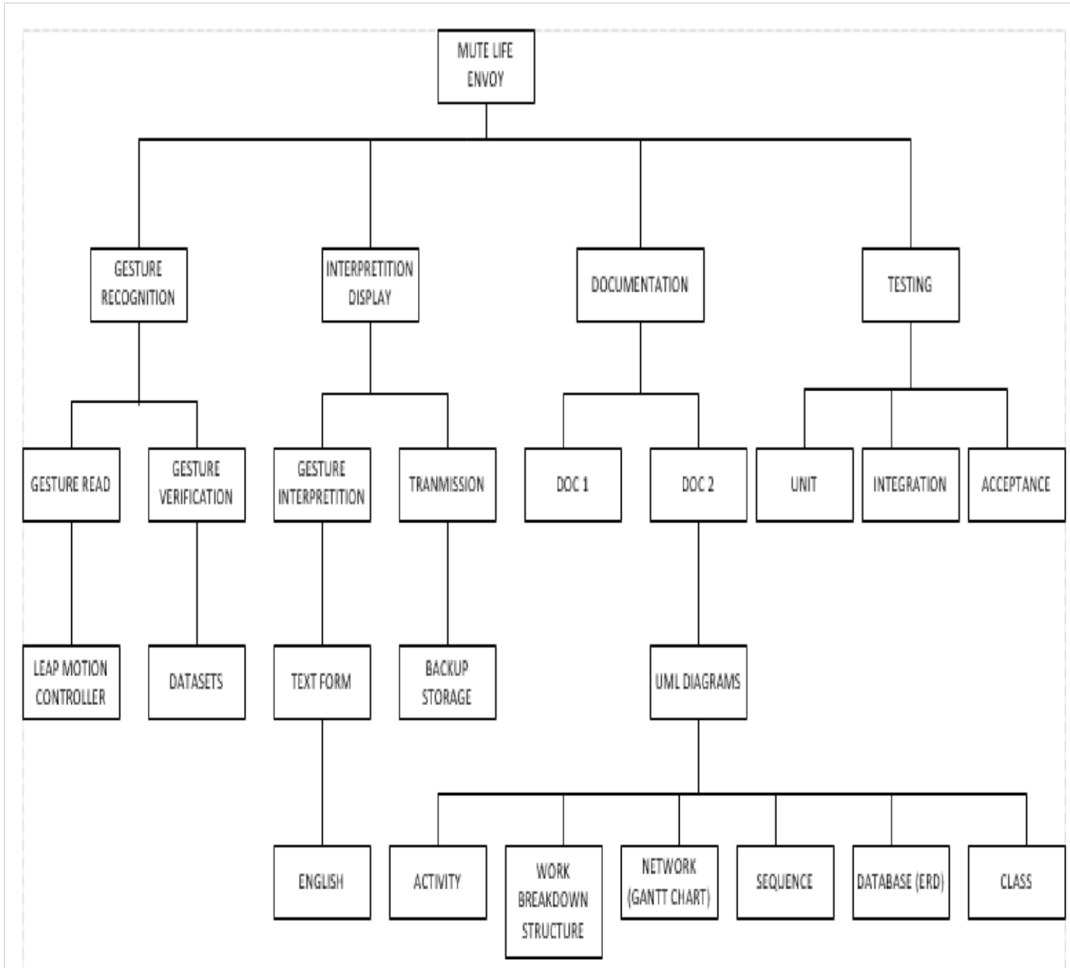


Figure 5: Work Breakdown Structure

3.2.3 Sequence Diagram

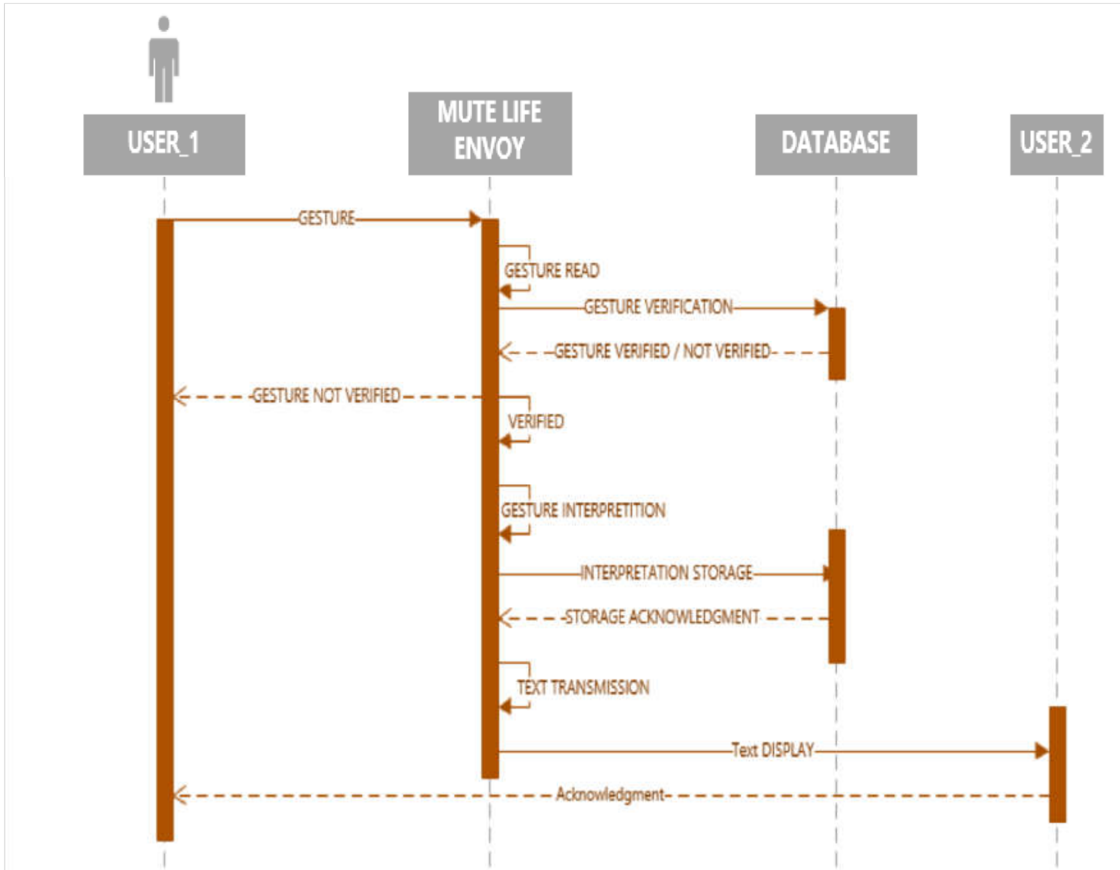


Figure 6: MLE Sequence Diagram

3.2.4 Block Diagram

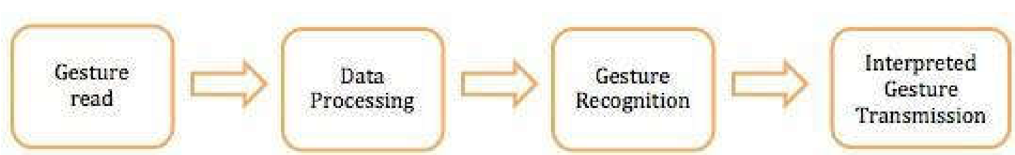


Figure 7: Block Diagram Showing Steps Involved in MLE System

3.2.5 Data Flow Diagram

3.2.5.1 System View

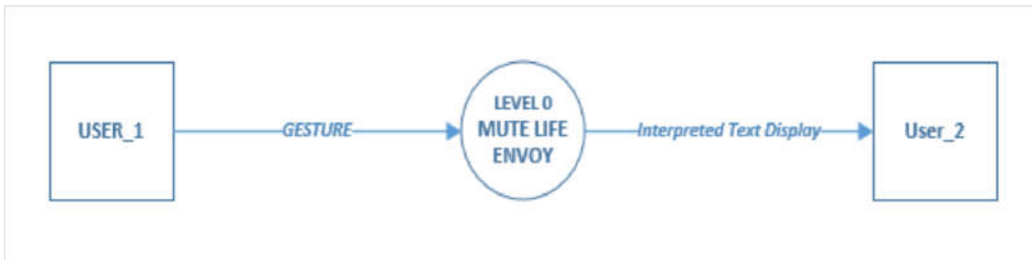


Figure 8: Dataflow Diagram - Level 0

3.2.5.2 System Inner View

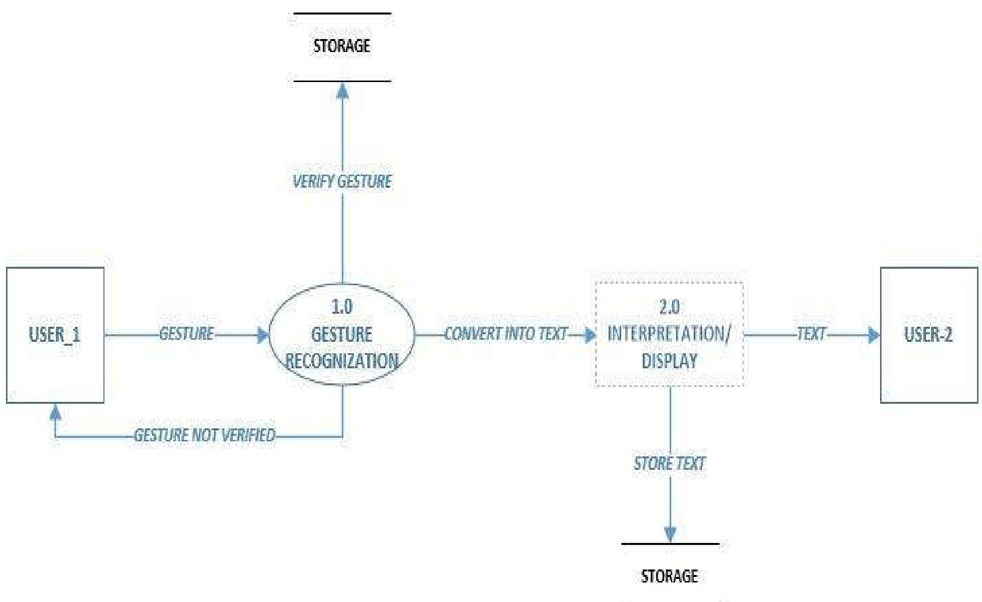


Figure 9: Dataflow Diagram - Level 1

3.2.5.3 System Detailed Overview

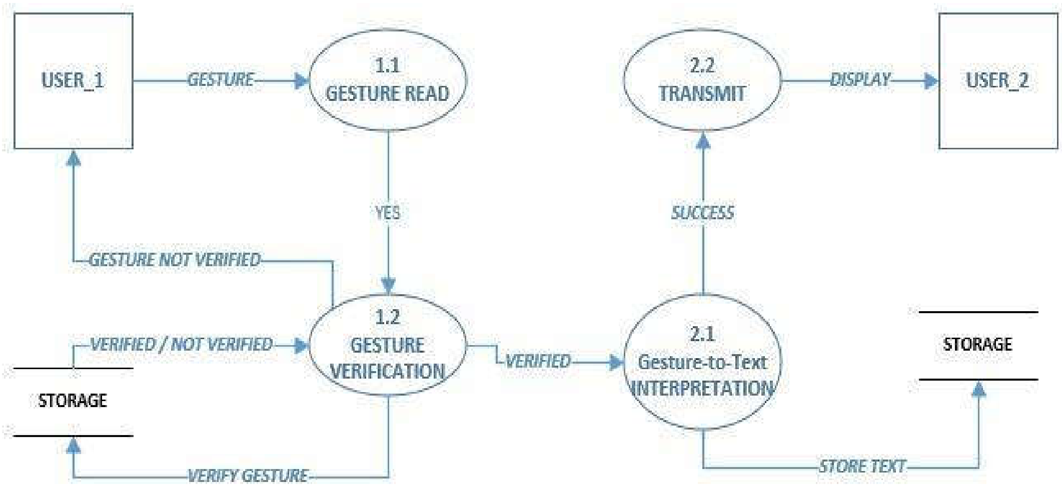


Figure 10: Dataflow Diagram - Level 2

3.2.6 Activity Diagram

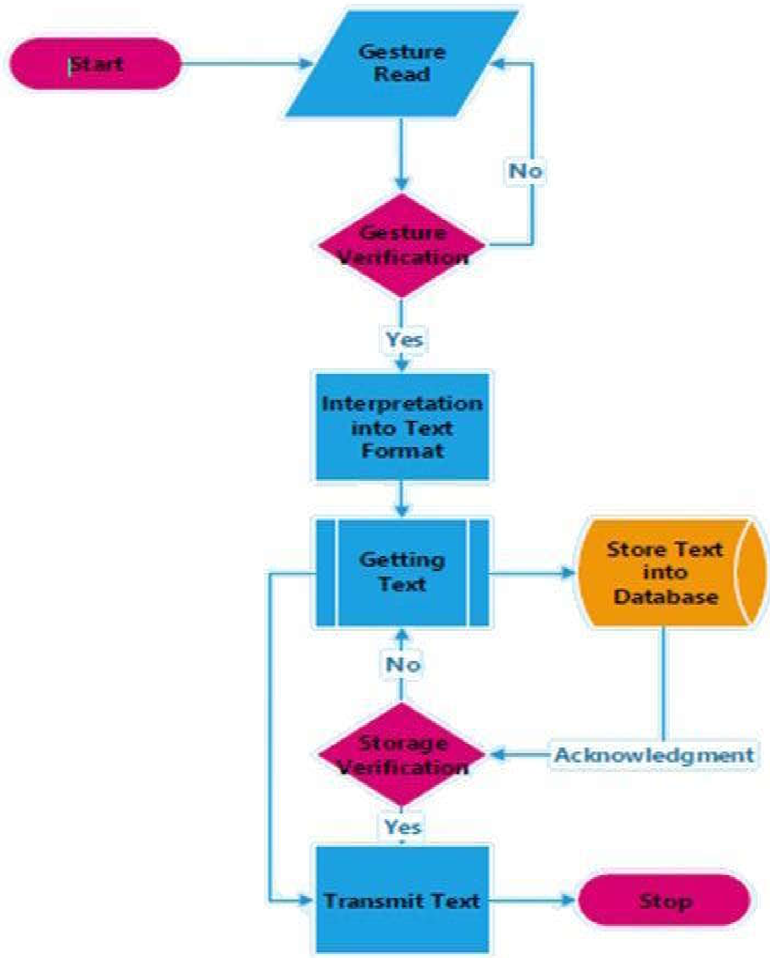


Figure 11: Activities Flow Diagram Showing Steps Involved In MLE System

3.2.7 Gesture Read

3.2.7.1 Sequence Diagram

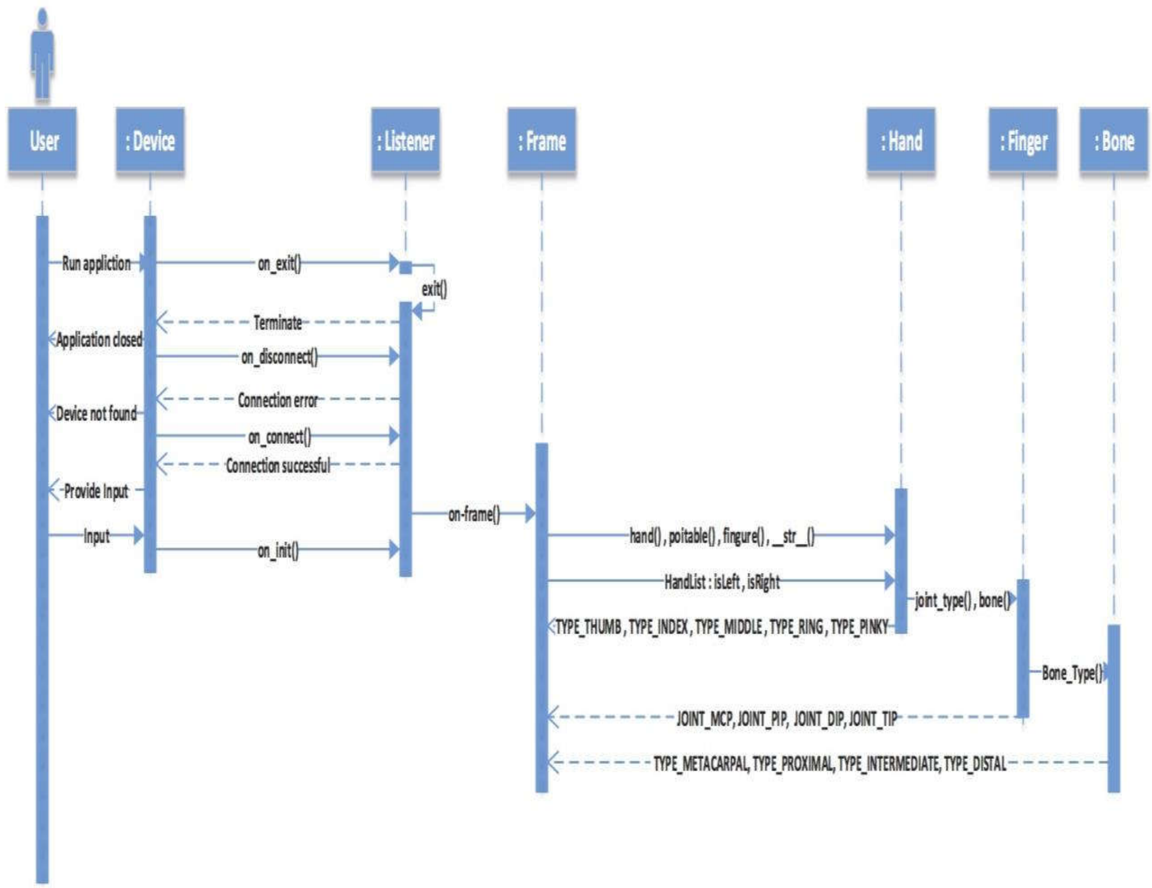


Figure 12: Collaboration Sequence - Gesture Read

3.2.7.2 Operation Contract

Name	Gesture Read
Responsibilities	Input read and encryption
Cross Reference	Uc-0
Exceptions	Device didn’t detect input or device connection failure
Pre-conditions	Input command should be received
Post-conditions	Device will stop and verify the input

Table 1: Operation Contract - Gesture Read

3.2.8 Gesture Verification

3.2.8.1 Sequence Diagram

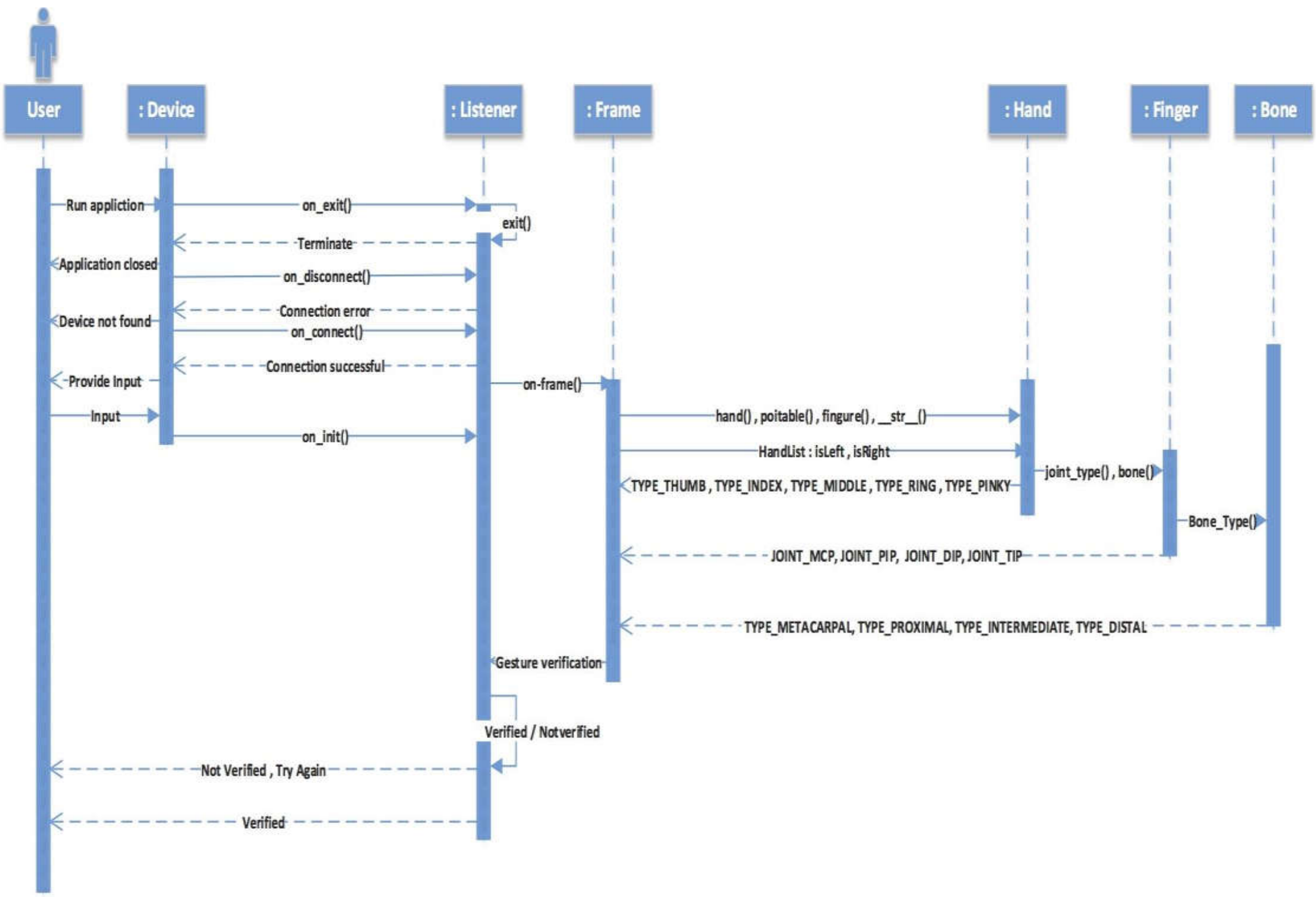


Figure 13: Collaboration Sequence - Gesture Verification

3.2.8.2 Operation Contract

Name	Gesture Verification
Responsibilities	System will select either gesture recognized as International Sign or not recognized
Cross Reference	Uc-1
Exceptions	Gesture Not Recognized
Pre-conditions	Gesture must be read and encrypted
Post-conditions	Verified gesture interpretation into English text

Table 2: Operation Contract - Gesture Verification

3.2.9 Gesture-to-Text Interpretation

3.2.9.1 Sequence Diagram

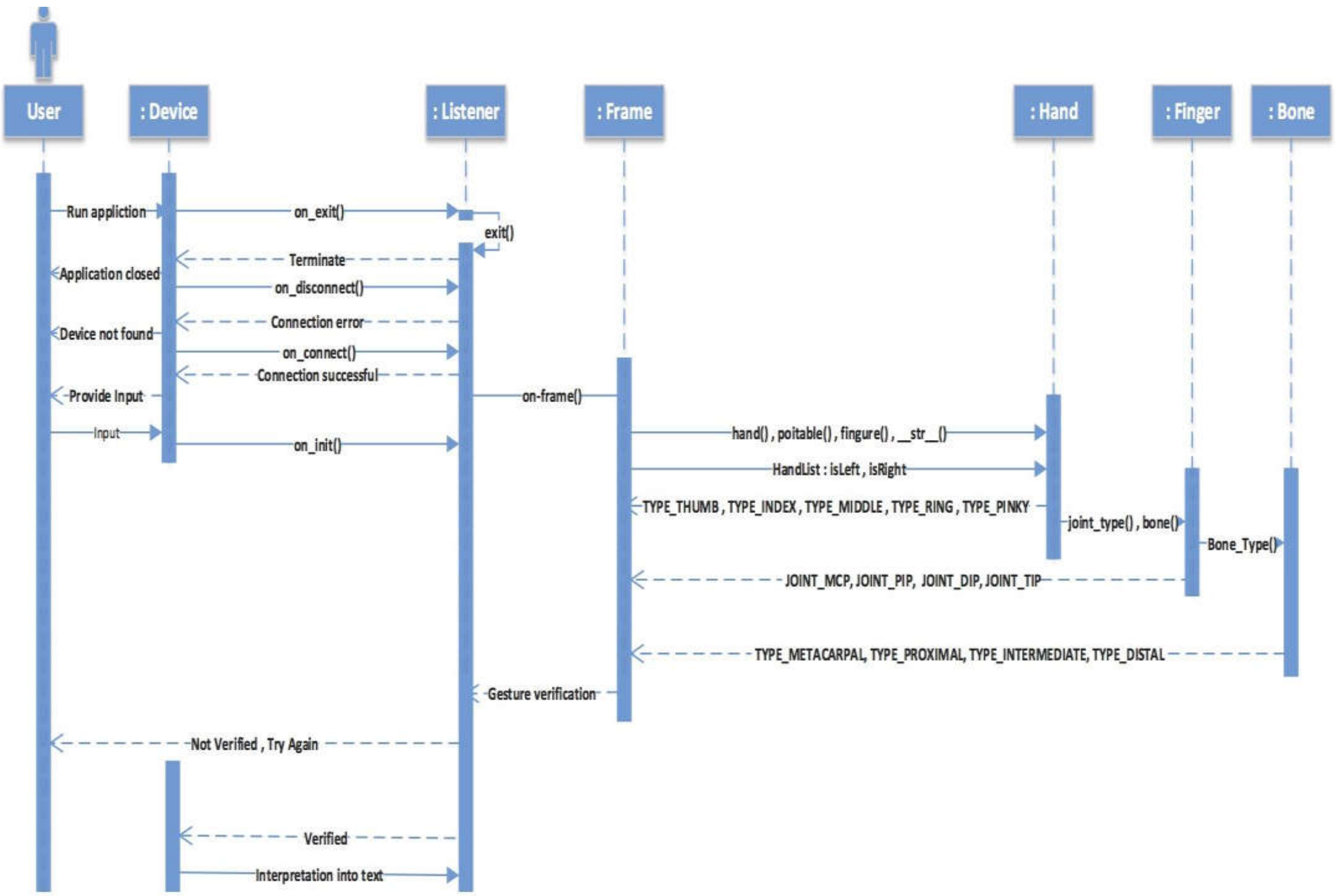


Figure 14: Collaboration Sequence Diagram - Gesture-to-Text Interpretation

3.2.9.2 Operation Contract

Name	Gesture-to-Text Interpretation
Responsibilities	Voice input will be read and interpreted
Cross Reference	Uc-2
Exceptions	-
Pre-conditions	Gesture Verification
Post-conditions	Store interpreted gesture in database

Table 3: Operation Contract - Gesture-to-Text Interpretation

3.2.10 Text Storage

3.2.10.1 Sequence Diagram

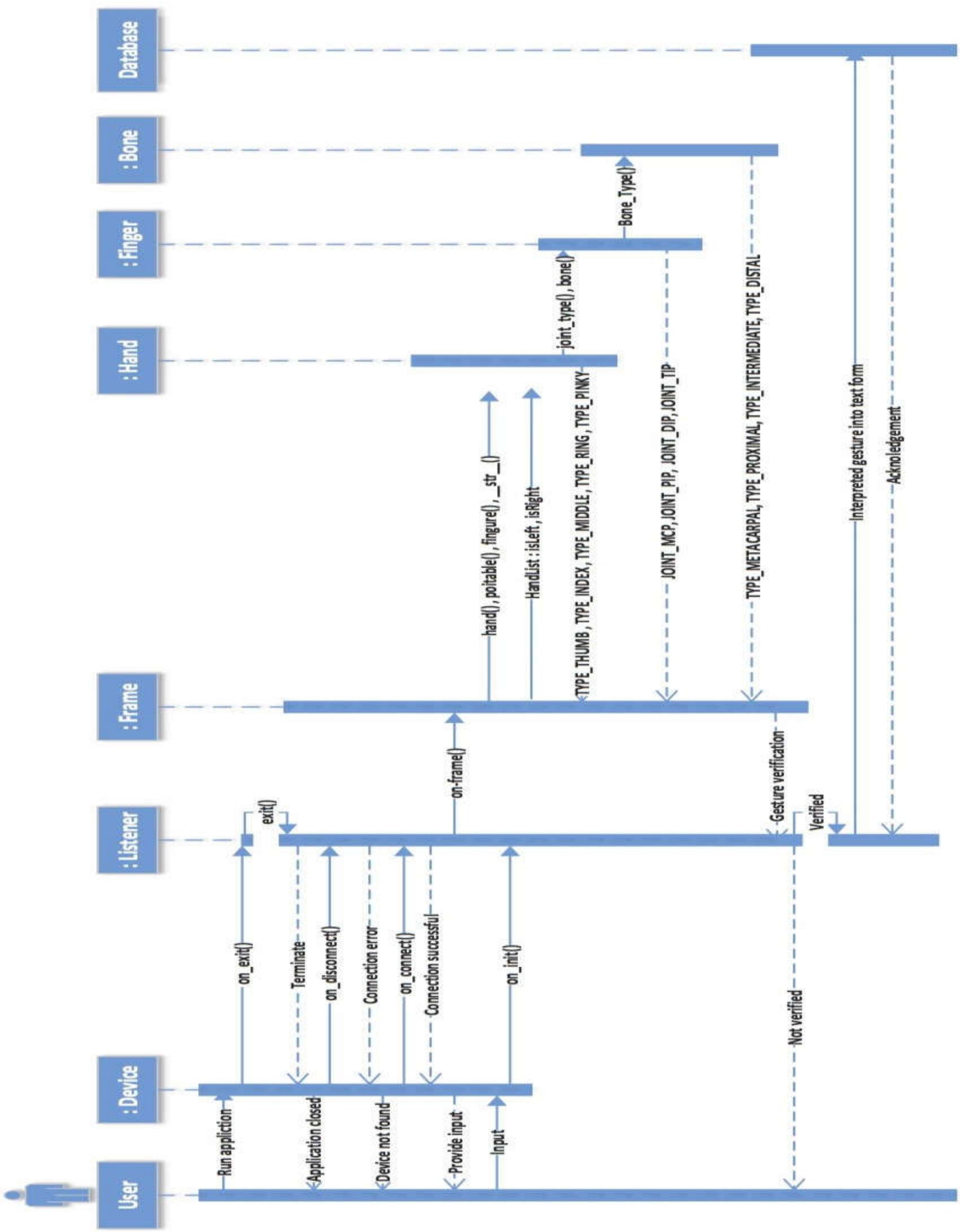


Figure 15: Collaboration Sequence - Text Storage

3.2.11.1 Operation Contract

Name	Text Storage
Responsibilities	Interpreted input gesture in English format will be stored in database in sequence of words and sentences.
Cross Reference	Uc-3
Exceptions	Application didn't stored text
Pre-conditions	Verified gesture must be interpreted
Post-conditions	Text Storage acknowledgment before transmission

Table 4: Operation Contract - Text Storage

3.2.11.2 Class Diagram

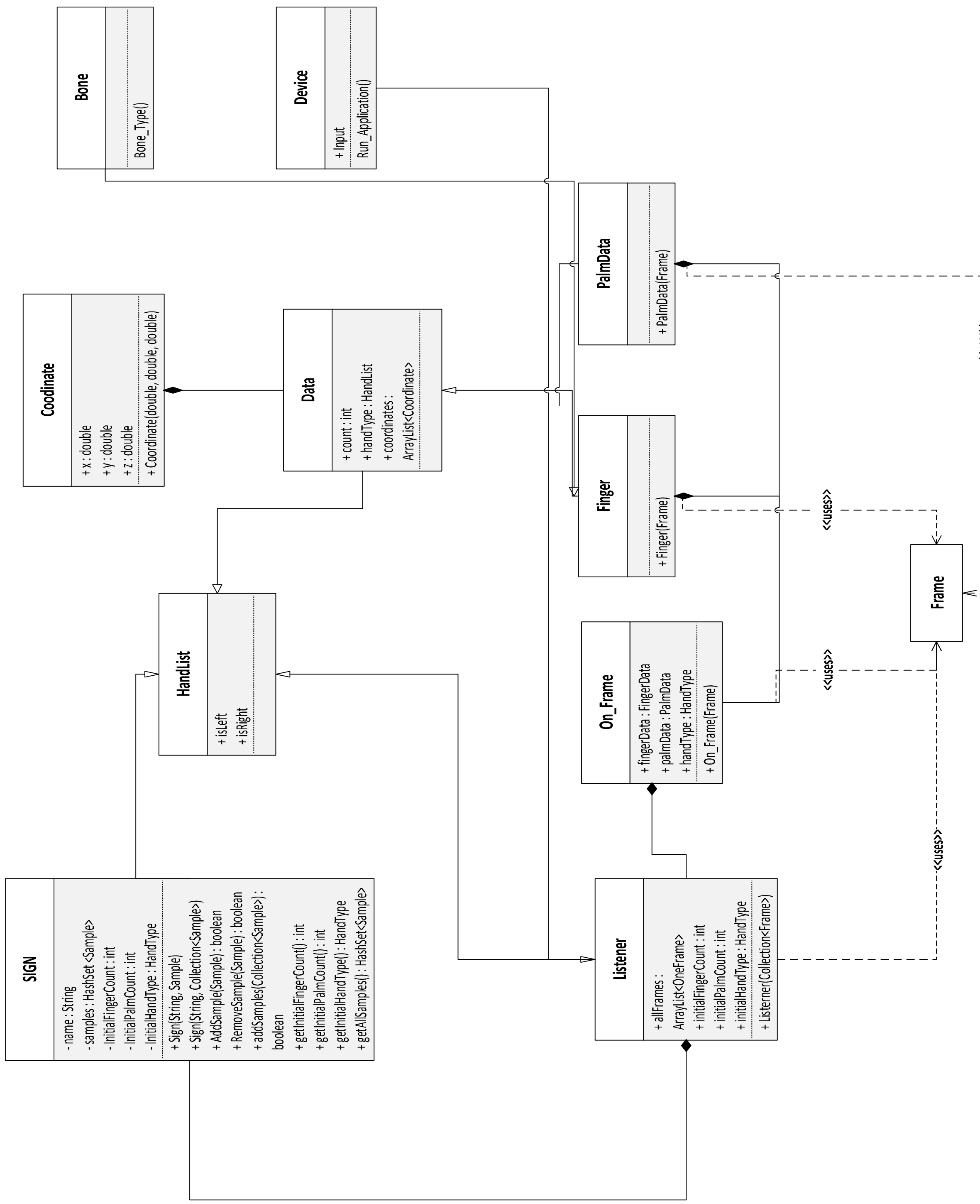


Figure 16: MLE Class Diagram

3.2.11.3 Class Table

Class	Purpose	Overview
Sign	• Gets hand type • Gets finger index • Gets palm index • Gets name	The Sign class is responsible for getting hand type, finger index, palm index, name and verify input sign
Device	• Gets input from user	The Device class is responsible for getting input from user
Bone	• Gets bone data	The Bone class is responsible for getting data of bones
Data	• Gets hand type and coordinates of the device	The Data class is responsible for getting the data of hand of the user and sets the coordinates of gesture on the device
Coordinate	• Getting device coordinates	The Coordinate class is responsible for getting coordinates of the device
HandList	• Gets data of hand type	The HandList class is responsible for getting data of Hand
Listener	• Gets finger data • Gets palm data • Gets hand type • Gets frame	The Listener class is responsible for getting data of fingers, palm, hand type, frame and verify input of user
On_Frame	• Gets finger data • Gets palm data • Gets hand type	The On_Frame class is responsible for getting data of fingers, palm, bones, and hand type
Finger	• Gets finger data in frame	The Finger class is responsible for getting data of fingers
PalmData	• Gets palm data	The PalmData class is responsible for getting data of palm in frames

Table 5: UML Class Table

CHAPTER 4

IMPLMENTATION

4.1 Implementation

The MLE is a real-time application with the integration of software with the hardware to achieve the efficiency within the system. MLE consists of two development stages i.e. *Gesture Recognition* and *Interpretation Transmission*; those both are further divided into sub-phases. The Iterative and Incremental development practices model is used throughout the workflow to meet the goals and deadlines. The development stages are discussed as below:

4.1.1 Gesture Recognition

The first stage is of Gesture Recognition that consists of four sub-phases i.e. *Gesture Read*, *Gesture Verification*, *Gesture-to-Text Interpretation* and *Text Storage*. The implementation to achieve the above discussed four milestones are described as following:

4.1.1.1 Gesture Read

The first phase "Gesture Read" prepares the data from taking input to converting it into computational language. The gesture data is received with the integration of MLE and hardware names as Leap Motion Controller (LMC) that provides the feature of detecting and reading the hands including the bones, joints and tip positions of them with the help of infrared sensors.

LMC *SDK* (*version 3.2*) is utilized to read a hand and extracted the data using the type of hand (left/right), extended fingers (*Thumb*, *Index*, *Middle*, *Ring*, *Pinky*), bones (*Metacarpal*, *Proximal*, *Intermediate*, Distal) and properties of the above discussed features (classes) provided in the LMC SDK.

4.1.1.2 Gesture Verification

The second phase "Gesture Verification" verifies the input data as recognized IS gesture. This milestone is achieved with the help of LMC SDK (version 3.2) and extracted data later at which by using the type of hand (left/right), extended fingers (Thumb, Index, Middle, Ring, Pinky), bones (Metacarpal, Proximal, Intermediate, Distal) and sub-properties of the above features (classes) discussed - provided in the LMC SDK - the IS gesture achieve the validation and inform the MLE to further interpret IS Language gesture into text format based upon *English alphabets*.

4.1.1.3 Gesture-to-Text Interpretation

The third phase "Gesture-to-Text Interpretation" converts the recognized input gesture into *English* format text and create a sequence of words and sentences. This milestone is achieved with the help of custom defined methods.

4.1.1.4 Text Storage

The fourth phase "Text Storage" is the phase that stores the interpreted text into database using *SQL* queries or preferences using pointer technique for secure transmission to the other user. This milestone is suggested to eliminate the loss of text due to power failure or the desktop pc crash.

4.1.2 Interpretation Transmission

The second stage is Interpretation Transmission that consists of three sub-phases i.e. *Text Read*, *Text Transmission* and *Text Display*. The implementation to achieve the mentioned four milestones are described as following:

4.1.2.1 Text Read

The first phase "Text Read" reads the stored text in database with the help of SQL queries or using preferences in order to encrypt the data and prepare it before the it's transmission to other user interface.

4.1.2.2 Text Transmission

The second phase "Text Transmission" transmits the encrypted data from one user to the other user over the Internet protocol using the cloud services provided by the PUN SDK that is also used by the multinational software developing companies and other companies i.e. *Microsoft, Coca Cola, CODEGLU, Fathom Interactives, M2H Game Studio, Fish Labs,* etc.

4.1.2.3 Text Display

The third phase is "Text Display" that displays the transmitted text over the other user interface after decrypting it into English Text format and receiving from database using SQL queries/preferences and displayed it with the help of PUN SDK to the other user interface.

4.2 Experiments

Many experiments were performed throughout the development phase, at various modules in different platforms and with the help of different tools, to achieve a suitable and in time result of different project modules. The experiments are discussed below:

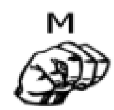
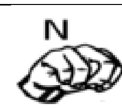
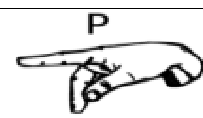
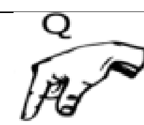
Python development was initially performed for the integration of LMC with the MLE to read the static hand gestures via LMC SDK for python development. The milestone to integrate the hardware with MLE and the initial development was successful but later the SDK showed so many flaws with python support as sometimes LMC SDK doesn't load up in the custom defined scripts of python files (.py). We also utilized PyQT for the front-end development of MLE at python platform.

.Net framework of Microsoft was used with it's component, *Windows Form,* was utilized for the development using C# programming language with the help of OOP skills. The front-end was developed with the help of .Net Windows Form using C# scripts. The issues were faced in the integration of LMC SDK with .Net framework of C# development.

Unity3D was utilised to perform the integration of LMC with it to perform the following milestones:

- The accuracy and support of LMC with IS Language.
- A-Z alphabets recognition (IS Language).
- Gestures reading through Leap Motion Controller
- Signs Verification.
- Input interpretation into sequence of words or sentences.
- Text Transmission from one medium to the other.

The developments to achieve the above milestones are completed with the usage of C# programming language. Some issues with the hand gesture's reading via LMC and hand gesture recognition were faced. The LMC faces issues with the reading of hand gestures that are not created when the hand palm is not facing the IR sensor so LMC doesn't even recognise those hand gestures. Furthermore, the issues faced within the confliction of few gestures with each other as the LMC reads the hand gesture using the hand's fingers and their bones, joints & fingertips states. The hand gestures that are not being read by the LMC are as following:

Alphabets	**Gesture**
• E	
• J	
• M	
• N	
• P	
• Q	
• R	
• S	
• T	

• X	
• Z	

Table 6: Not Readable Hand Gestures For MLE

The hand gestures of IS that shows confliction with each other in LMC are as following:

C	G
C	O
O	G
K	U
K	H
K	V
U	H
U	V
H	V

Table 7: Confliction Occurrence Between Hand Gestures

The gestures defined for the working product are divided in both hands to ignore the conflictions between the hand gestures and are mentioned as following:

Left Hand	**Right Hand**
C	A
G	B
K	D
L	F
U	H
W	I
Y	O
Delete	V
	Space

Table 8: MLE Hand Gestures Input

The accuracy of an alphabet is 0.5 to 1.5 seconds per gesture per frame. The word like bad, dad, ball etc. takes time of 4.5 seconds per word of 3-4 gestures. Whereas, sentence containing 40 gestures (having 25 alphabets) all are either used once or repeated, took a time of 3.41 minutes per 40 gestures sentence.

The sentence that is used and accuracy tested as mentioned above is “ I had a ball I did a goal and I did bad foul ”

The outcomes of journal [1], that was using IS Language with some other techniques, are compared with the outcomes of our research of language using LMC. The results among the cumulative accuracy are as below:

Alphabets	**Test 1**	**Test 2**	**Test 3**	**Average**
A	1	1	1	100.00%
B	1	1	1	100.00%
C	1	0	0	33.33%
D	1	1	1	100.00%
E	1	1	0	66.67%
F	1	1	0	66.67%
G	1	0	0	33.33%
H	1	1	0	66.67%
I	1	1	1	100.00%
J	1	0	1	66.67%
K	1	0	0	33.33%
L	1	1	0	66.67%
M	1	0	0	33.33%
N	1	0	0	33.33%
O	1	1	0	66.67%
P	0	0	0	0.00%
Q	1	1	0	66.67%
R	0	0	0	0.00%
S	0	1	0	33.33%
T	0	0	0	0.00%
U	0	0	1	33.33%
V	1	1	0	66.67%
W	1	1	0	66.67%
X	1	0	0	33.33%
Y	1	1	0	66.67%
Z	1	0	0	33.33%
Cumulative Average				**52.56%**

Table 9: Results of Alphabet Recognition Using Geometric Template Matching [1]

Alphabets	Test 1	Test 2	Test 3	Average
A	1	1	1	100.00%
B	1	1	1	100.00%
C	0	0	0	0.00%
D	1	1	0	66.67%
E	0	0	0	0.00%
F	1	0	0	33.33%
G	1	0	0	33.37%
H	1	0	0	33.37%
I	1	1	1	100.00%
J	1	0	0	33.33%
K	1	1	0	66.67%
L	1	0	0	33.33%
M	1	1	0	66.67%
N	0	0	0	0.00%
O	1	1	0	66.67%
P	1	0	0	33.33%
Q	1	1	0	66.67%
R	1	0	0	33.33%
S	0	0	0	0.00%
T	0	0	0	0.00%
U	1	0	1	66.67%
V	1	1	0	66.67%
W	1	1	0	66.67%
X	1	0	0	33.33%
Y	1	0	0	33.33%
Z	1	0	0	33.33%
Cumulative Average				**44.87%**

Table 10: Results Of Alphabet Recognition Using Artificial Neural Network [1]

Alphabets	**Test 1**	**Test 2**	**Test 3**	**Average**
A	1	1	0	66.67%
B	1	0	1	66.67%
C	1	0	0	33.33%
D	0	1	0	33.33%
E	0	0	0	0.00%
F	1	0	0	33.33%
G	0	0	0	0.00%
H	1	0	0	33.37%
I	1	1	1	100.00%
J	1	1	0	66.67%
K	0	0	0	0.00%
L	1	0	0	33.33%
M	1	0	0	33.33%
N	0	0	0	0.00%
O	1	1	1	100.00%
P	0	0	0	0.00%
Q	0	1	0	33.33%
R	0	0	0	0.00%
S	0	0	0	0.00%
T	0	0	0	0.00%
U	1	0	0	33.33%
V	1	1	0	66.67%
W	1	0	0	33.33%
X	1	1	0	66.67%
Y	1	1	0	66.67%
Z	1	0	0	33.33%
Cumulative Average				**35.90%**

Table 11: Results Of Alphabet Recognition Using Cross Correlation [1]

The testing performed in the MLE to determine the cumulative accuracy using the LMC is as below:

Alphabets	Test 1	Test 2	Test 3	Test 4	Test 5	Average of 3 Tests	Average of 5 Tests
A	1	1	1	1	1	100.00%	100.00%
B	1	1	1	0	1	100.00%	80.00%
C	1	0	1	1	1	66.67	80.00%
D	1	1	1	1	1	100.00%	100.00%
E	0	0	0	0	0	0.00%	0.00%
F	1	1	1	0	1	100.00%	80.00%
G	0	1	1	0	0	66.67%	40.00%
H	0	0	0	0	0	0.00%	0.00%
I	1	1	0	1	1	66.67%	80.00%
J	0	0	0	0	0	0.00%	0.00%
K	1	1	0	0	0	66.67%	40.00%
L	1	1	1	1	0	100.00%	80.00%
M	0	0	0	0	0	0.00%	0.00%
N	0	0	0	0	0	0.00%	0.00%
O	1	1	1	0	0	100.00%	60.00%
P	0	0	0	0	0	0.00%	0.00%
Q	0	0	0	0	0	0.00%	0.00%
R	0	0	0	0	0	0.00%	0.00%
S	0	0	0	0	0	0.00%	0.00%
T	0	0	0	0	0	0.00%	0.00%
U	1	0	1	1	0	66.67%	60.00%
V	1	0	0	0	1	33.33%	40.00%
W	1	1	1	1	1	100.00%	100.00%
X	0	0	0	0	0	0.00%	0.00%
Y	1	0	1	1	1	66.67%	80.00%
Z	0	0	0	0	0	0.00%	0.00%
Cumulative Accuracy						**43.59%**	**39.00%**

Table 12: Results Of Alphabet Recognition Using LMC

The IS language testing is further executed by using the knowledge of Convolutional Neural Network (CNN). The Convolutional Neural Network (CNN) has shown excellent performance in many computer vision and machine learning problems [23]. In CNN, a 2D convolutional layer is used for the image pixels' matrix in convolutional layer. The convolutional layer reads the pixels of the image (input data) for training and generates convolutions within a specified matrix dimension to generate filters (also called kernels) for the feature-mapping process. The feature map process is achieved with the help of kernels that overlap itself over the input image pixel's convolution matrix to identify each feature and their number of occurrence. The featured-map matrix is then bypassed in the max-pooling layer, which gathers the maximum pooling of features in featured-map matrix using the same dimensions of kernel and creates a max-pooled matrix. The max-pooling layer forwards the information (matrix) to the fully connected layer, where the full connections of the neural network are generated and the max-pooled matrix is then connected with each perceptron. The fully connected layer then provides the gathered and analysed information to the actual output perceptron, that in turn gives the trained class, the input image (data) belongs to.

Epochs are used for the optimization of the CNN; which runs the network with forward propagation and backward propagation but takes both as one step i.e. epoch. The validation steps are used to tell the network how many images (data) should be trained/read from the dataset i.e. 50 validation steps inform the network to read/train the images (data) in a sequence of 50. So, in general if we have 500 images (data), the dataset would be read/trained in chunks of 50 images at a time and if the last iteration of validation step is less than 50 images then they are trained/read within single iteration.

CNN trained feeds the image for prediction containing the dimensions of 64x64 and coloured image (i.e. containing 3 channels). The activation function "softmax" is used for the gestures classification.

The CNN model is illustrated below:

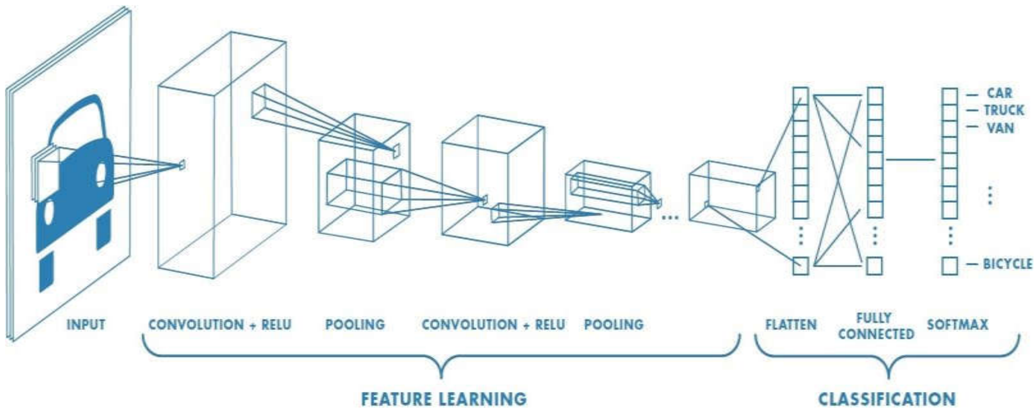


Figure 17: CNN Model Illustration [24]

The architecture of CNN model is as below:

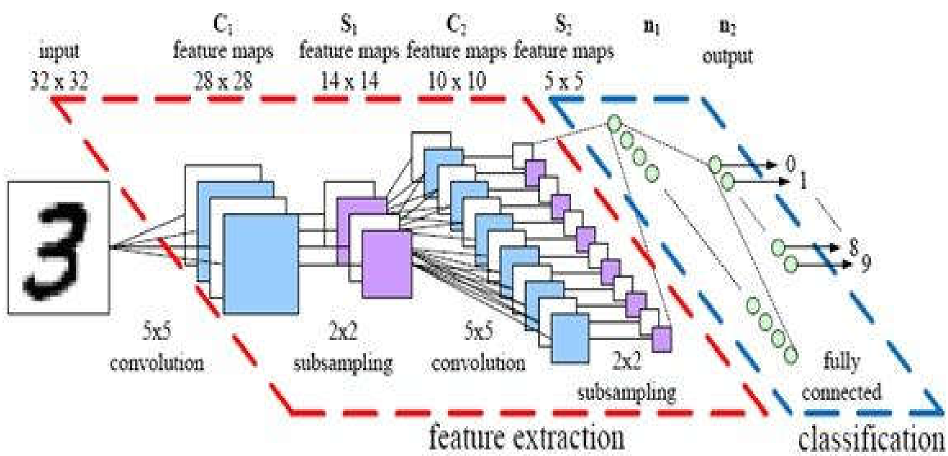


Figure 18: CNN Model Architecture [25]

The CNN of binary classification over the datasets of hand gesture A and B and gathered the following model accuracy, loss rate, training data accuracy and testing dataset (value) accuracy as below:

```
Found 7279 images belonging to 24 classes.
Found 1793 images belonging to 24 classes.
Epoch 1/10
4536/4536 [==============================] - 1940s 428ms/step - loss: 0.2291
- acc: 0.9321 - val_loss: 2.5447 - val_acc: 0.7526
Epoch 2/10
4536/4536 [==============================] - 2072s 457ms/step - loss: 0.0317
- acc: 0.9903 - val_loss: 2.9923 - val_acc: 0.7619
Epoch 3/10
4536/4536 [==============================] - 2022s 446ms/step - loss: 0.0174
- acc: 0.9951 - val_loss: 3.1276 - val_acc: 0.7540
Epoch 4/10
4536/4536 [==============================] - 2404s 530ms/step - loss: 0.0140
- acc: 0.9961 - val_loss: 3.2173 - val_acc: 0.7472
Epoch 5/10
4536/4536 [==============================] - 1816s 400ms/step - loss: 0.0103
- acc: 0.9972 - val_loss: 3.3543 - val_acc: 0.7490
Epoch 6/10
4536/4536 [==============================] - 1758s 388ms/step - loss: 0.0091
- acc: 0.9978 - val_loss: 3.5105 - val_acc: 0.7566
Epoch 7/10
4536/4536 [==============================] - 1742s 384ms/step - loss: 0.0087
- acc: 0.9979 - val_loss: 3.2102 - val_acc: 0.7519
Epoch 8/10
4536/4536 [==============================] - 1732s 382ms/step - loss: 0.0082
- acc: 0.9979 - val_loss: 3.6077 - val_acc: 0.7487
Epoch 9/10
4536/4536 [==============================] - 1763s 389ms/step - loss: 0.0068
- acc: 0.9982 - val_loss: 3.4473 - val_acc: 0.7598
Epoch 10/10
4536/4536 [==============================] - 1719s 379ms/step - loss: 0.0064
- acc: 0.9985 - val_loss: 3.5336 - val_acc: 0.7530
```

Figure 19: Train and Test Dataset Accuracy

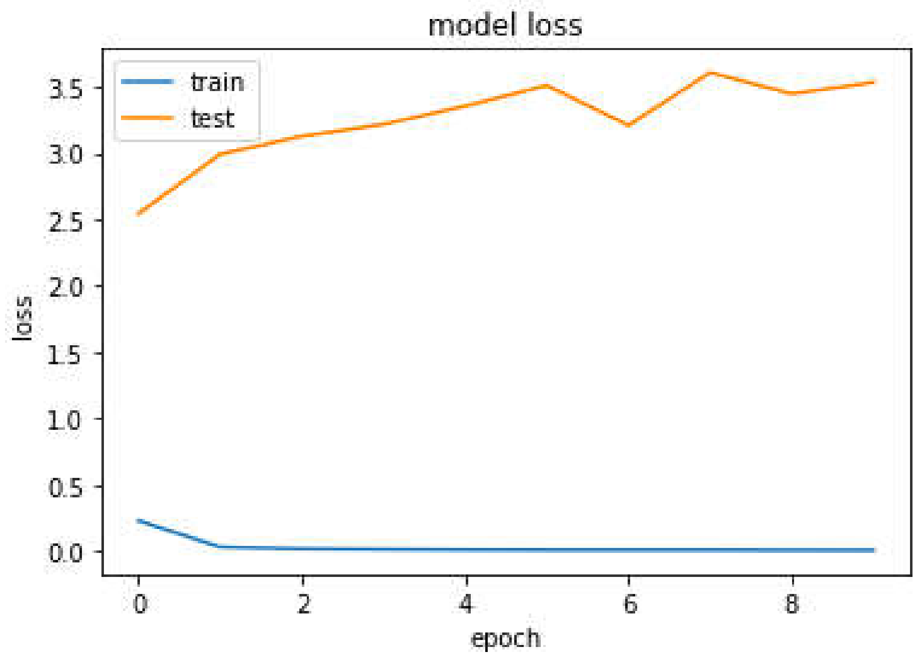


Figure 20: Model Loss

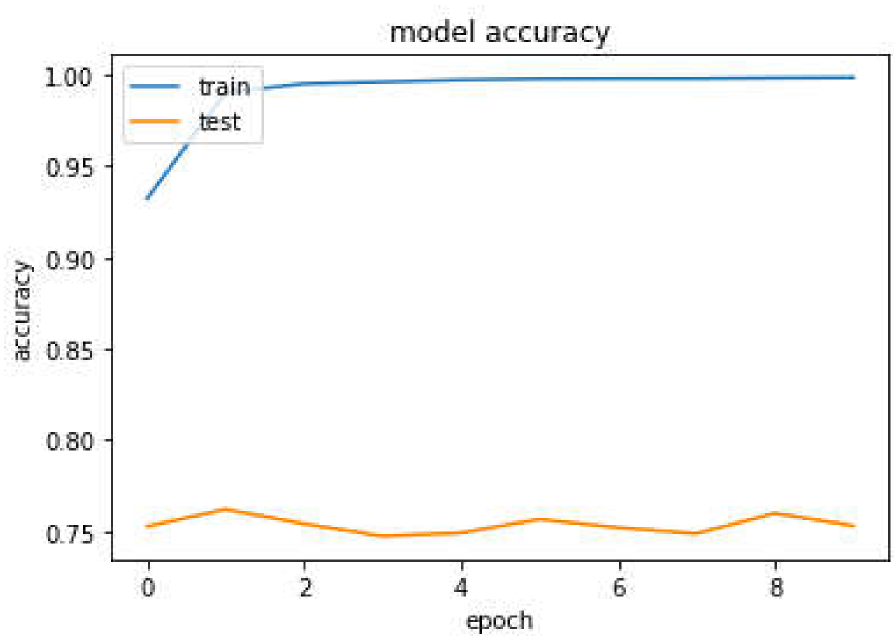


Figure 21: Model Accuracy

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
<class 'numpy.ndarray'>
(64, 64, 3)
[[0]]
```

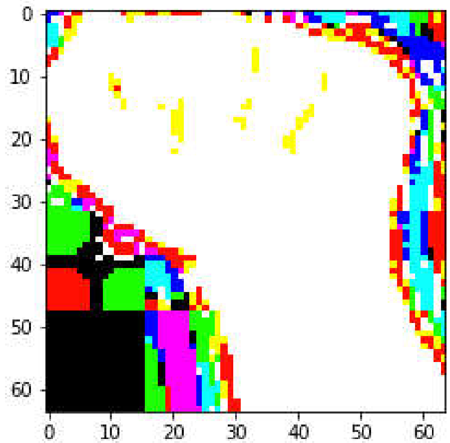


Figure 22: Prediction of Gesture A (0)

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
<class 'numpy.ndarray'>
(64, 64, 3)
[[1]]
```

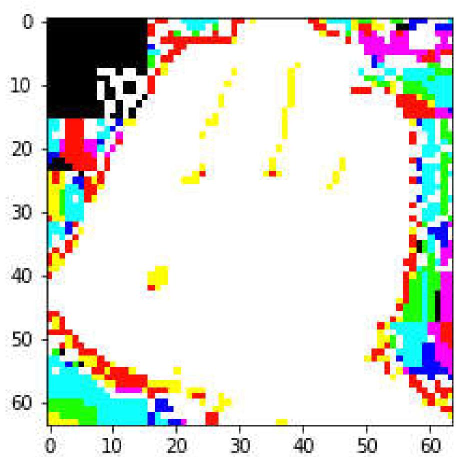


Figure 23: Prediction of Gesture B (1)

```
Using TensorFlow backend.
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats
or [0..255] for integers).
<class 'numpy.ndarray'>
(64, 64, 3)
[12]
```

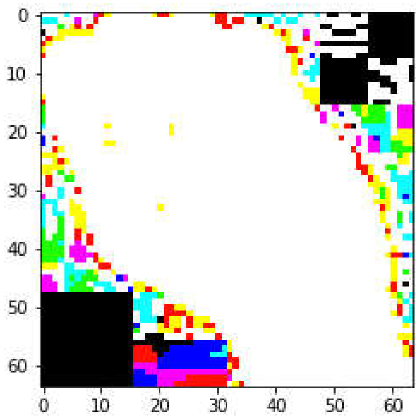


Figure 24: Prediction of Gesture E (12)

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats
or [0..255] for integers).
<class 'numpy.ndarray'>
(64, 64, 3)
[18]
```

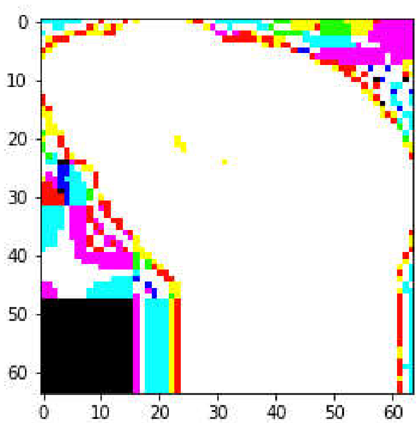


Figure 25: Prediction of Gesture S (18)

CHAPTER 5

USER MANUAL

5.1 Safety and Emergency Requirements

i) Handle the loss of power to secure database and transmission flow with power-backup via generators or UPS.
ii) Handle the migration and maintenance of database by never taking the entire system offline at once but achieve it through redundancy of essential components.
iii) Handle the loss of Internet connection for proper encrypted data keep getting stored in database for transmission.
iv) Connect the Leap Motion Controller before executing the application.
v) Make sure Leap Motion Controller indicate green light and the device is working properly especially the infrared sensors (three red lights).
vi) Make Sure device is clean and protect it from liquid substances.
vii) Leap Motion Controller cannot read hand (palm) at specific angle and height. The improper reading will affect results.
viii) USB cable has been known to cause trouble; in such case change USB cable/port whatever is required to be replaced.
ix) Use Leap Visualizer the check the device is working properly.

5.2 Constraints

The constraints are as below:

i) Leap Motion Controller communication with the system.
ii) MLE components communication with each other.
iii) Leap Motion Controller latest SDK is required for the compatibility with the system.
iv) The palm should be placed in front of Leap Motion Controller for proper readings and avoid miscellaneous results.
v) Leap Motion Controller has a range limitation and cannot read hand exceeding to 25cm range.
vi) Leap Motion Controller frame can be controlled at minimum 1.3 seconds per frame only.
vii) Leap Motion Controller takes last 60 frames and destroys them to get the next 60 frame after releasing them from memory.
viii) Leap Motion Controller do not read any input outside the range of IR sensor spectrum that have the 850nm range.
ix) Leap Motion Controller lost it's reading in the sunlight
x) Convolutional Neural Network needs high processing units for the training of datasets within a day or minutes.
xi) CNN are Translation Invariant means Convolutional Networks are unable to identify the position of one object relative to another. For example, CNN predict a Face to a bunch of randomly assembled face parts because all the key features are there. [26]
xii) CNN require a lot of data to generalize because it has to learn different filters for each different viewpoint. [26]

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

The aim of this project is to develop a gesture recognizer for dictation of ISL. So many applications are introduced but no one has yet focused over ISL that was standardized by WFD. This system is also beneficial for people with disabilities. For visually impaired people, for people who are deaf or they can hardly hear; this type of gesture recognizer system would be helpful to overcome their problems. This system may also assist general public in our community who want to interact with DNM but due to pidgin issues they cannot interact.

6.2 Recommendation

We recommend the avoiding of leap motion controller for hand gesture recognition system as it has so many flaws with hand reading and it's validation at each frame per second. Convolutional Neural Network is so far a good choice for hand gesture recognition as it has lesser flaws and the hand gesture data could be trained through it as per needs. We will continue our work for MLE with the help of Capsule Network in the future.

REFERENCES

Books:

[1] (Barkat), *Signal Detection and Estimation*, 2nd Edition, Artech House , Boston, MA., 2005 (F.R. Khan; R.Pradipa; Kaushik Deb; Reza Azad; P. Karthick; W.Lu; R.McCartney; Shao; A.Sarkar; S P. Subha Rajam)

[2] P. Peebles, *Probability, Random Variables and Random Signal Processing*, 4th Ed., , McGraw Hill, New York, 2001.

Journal Papers:

[1] F.R. Khan, H.F. Ong and N. Bahar, 'A Sign Language to Text Converter Using Leap Motion'', International Journal on Advance Science Engineering and Information Technology, vol. 6.6, pp. 1089-95, Dec 2016.

[2] M.M. Hasan and P.K. Mishra, "Hand Gesture Modeling and Recognition Using Geometric Features: A Review", Canadian Journal on Image Processing and Computer Vision, vol. 3(1), pp.12-26, March 2102.

[6] R.Pradipa, Ms S.Kavitha, "Hand Gesture Recognition – Analysis of Various Techniques, Methods and Their Algorithms" International Journal of Innovative Research in Science, Engineering and Technology Volume 3, Special Issue 3, March 2014

[10] Kaushik Deb, Helena Parvin Mony & Sujan Chowdhury (Feb 2012), "Two-Handed Sign Language Recognition for Bangla Character Using Normalized Cross Correlation" Global Journal of Computer Science & Technology, Volume 12 Issue 3 Version 1.0 Fabruary 2012.

[12] Reza Azad, Babak Azad and Iman tavakoli kazerooni, "Real-Time and Robust Method for Hand Gesture Recognition System Based on Cross-Correlation Coefficient" IEEE Member, Electrical and Computer Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran.

[20] P. Karthick, N. Parthiba, V.B. Rekha, S. Thanalaxmi, "Transforming Indian Sign Language into Text Using Leap Motion", International Journal of Innovative Research in Science, Engineering and Technology, Vol:3, Issue:4, April 2014.

[7] P. Karthick, N. Parthiba, V.B. Rekha, S. Thanalaxmi, " Transforming Indian Sign Language into Text Using Leap Motion", International Journal of Innovative Research in Science, Engineering and Technology, Vol:3, Issue:4, April 2014.

[17] W.Lu, Z.Tong, J. Chu, "Dynamic Hand Gesture Recognition With Leap Motion Controller", IEEE Signal Processing Letters, Vol:23, No:9, September 2016

[18] R.McCartney, J.Yuan, H.-P. Bischof, "Gesture Recognition With The Leap Motion Controller", International Conference IP, Computer Vision and Pattern Recognition, IPCV'15, 2015

[19] L. Shao, "Hand Movement and Gesture Recognition Using Leap Motion Controller", Stanford EE 267, Virtual Reality Course Report.

Conference Papers:

[3] A.Sarkar, G.K. Capoor, K.A. Patel and G. Ram R.k. "Gesture control of drone using a motion controller", In Industrial Informatics and Computer Systems (CIICS), International Conference on (pp. 1-5) IEEE, March 2016.

[9] S P. Subha Rajam, G. Balakrishnan (2011), "Recognition of Tamil Sign Language Alphabet using Image Processing to aid Deaf-Dumb People" International Conference on Communication Technology and System Design 2011, Procedia Engineering 30 (2012) 861 – 868.

[14] Adithya V. Vinod P. R, Usha Gopalakrishna (2013), "Artificial Neural Network Based Method for Indian Sign Language Recognition", IEEE Conference on Information and Communication Technologies (ICT 2013)

[21] Z. Xiaomei, D. Shiquan, W. Hui, "Research on Chinese-American Sign Language Translation", The 14th IEEE International Conference on Computational Science and Engineering.

Technical Reports and Theses:

[4] T.C. Yan, L.K. Chun, C.Y. Laam and T.W.Yin, "Using The Leap Motion Controller To Translate Sign Language To Speech", MS Thesis, Department of Computer Science, HKST, Clear Water Bay, Hong Kong, 2016.

[8] S.M. Hassan Ahmed, Todd C. Alexander and Georgios C. Anagnostopoulos, "Real-time, Static and Dynamic Hand Gesture Recognition for Human-Computer Interaction" Electrical Engineering, University of Miami, Miami, FL; Electrical & Computer Engineering, Florida Insitute of Technology, Melbourne, FL

[11] Brandon Garcia, Sigberto Alarcon Viesca, "Real-time American Sign Language Recognition with Convolutional Neural Networks" Stanford University

[13] Thad Eugene Starner, "Visual Recognition of American Sign Language Using Hidden Markov Models" Massachusetts Institute of Technology, February 1995

[15] V.V. NABĐYEV, C. KÖSE, S. BAYRAK, "An Artificial Neural Network Approach For Sign Language Vowels Recognition" Department of Computer Engineering, Faculty of Engineering, Karadeniz Technical University, Trabzon, Turkey

[22] N. Raziq, S. Latif, "Pakistan Sign Language Recognition and Translation System Using Leap Motion Controller", Springer International Publishing AG, F. Chafa et al. (eds.), Advances on P2P, Parallel, Grid, Cloud and Internet Computing, Lecture Notes on Data Engineering and Communications Technologies 1, 2017

Book Papers:

[5] A.Chan, T. Halevi and N. Memon, "Leap motion controller for authentication via hand geometry and gestures", Human Aspects of Information Security, Privacy and Trust, pp.13-22.

Electronic Sources from Internet:

[9] Aleem Khalid Alvi, M. Yousuf Bin Azhar, Mehmood Usman, Suleman Mumtaz, Sameer Rafiq, Razi Ur Rehman, Israr Ahmed, "Pakistan Sign Language Recognition Using Statistical Template Matching" World Academy of Science, Engineering and Technology, Vol:1, No:3, 2007, Available at: https://waset.org/publications/15125/pakistan-sign-language-recognition-using-statistical-template-matching

[23] J. Wu, "Introduction to convolutional neural networks", National Key Lab for Novel Software Technology, Nanjing University. China, 2017), Available at: https://pdfs.semanticscholar.org/450c/a19932fcef1ca6d0442cbf52fec38fb9d1e5.pdf

[24] https://hackernoon.com/deep-learning-cnns-in-tensorflow-with-gpus-cba6efe0acc2

[25] https://www.kdnuggets.com/2016/04/deep-learning-vs-svm-random-forest.html

[26] https://www.quora.com/How-is-Capsule-networks-different-from-CNN

[27] https://en.wikipedia.org/wiki/International_Sign

[28] https://propelsteps.wordpress.com/2014/06/27/know-sign-languages