Bangla Sign Language Recognition Using Leap Motion Sensor

by

Tamkin Mahmud Tan
15301040
Anna Mary Mondol
15301056
Noshin Nawal
15301077
Sabbir Ahmed
15301079

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
August 2019

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Tamkin Mahmud Tan
15301040

Anna Mary Mondol
15301056

Noshin Nawal
15301077

Sabbir Ahmed
15301079

Approval

The thesis titled "Bangla Sign Language Recognition Using Leap Motion Sensor" submitted by

1. Tamkin Mahmud Tan (15301040)
2. Anna Mary Mondol (15301056)
3. Noshin Nawal (15301077)
4. Sabbir Ahmed (15301079)

of Summer, 2019 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on August 28, 2019.

Examining Committee:

Supervisor:
(Member)

Dr. Jia Uddin
Associate Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Jia Uddin
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Interim Chair)

Dr. Mahbub Alam Majumdar
Professor
Department of Computer Science and Engineering
Brac University

Abstract

Sign language is used by hearing and speech impaired people to transmit their messages to other people but it is difficult for a regular people to understand this gesture based language. Instantaneous responses on sign language can significantly enhance the understanding of sign language. In this paper, we propose a system that detects Bangla Sign Language using a digital motion sensor called Leap Motion Controller. It is a sensor or device which can detect 3D motion of hands, fingers and finger like objects without any contact. A Sign Language Recognition system has to be designed to recognize a hand gesture. In sign language system, gestures are defined as some specific patterns or movement of the hands to give an expression. There has to be a library which includes all the datasets to match with the user given gestures. We have to compare the sequences of data we get from Leap Motion and our datasets to get an optimal result which is basically the output. It will then show the output as text in the display. For our system, we choose to use $P Point-Cloud Recognizer algorithm to match the input data with our datasets. This recognition algorithm was designed for rapid prototyping of gesture-based UI and can deliver an average over 99% accuracy in user-dependent testing. Our proposed model is designed in a way so that the hearing and speech impaired people can communicate easily and efficiently with common people.

Keywords: Bangla Sign Language; Leap Motion Controller; $P Point-Cloud Recognizer; Machine Learning; HCI; Greedy Cloud Match; Gesture Recognition

Dedication

We would like to dedicate our paper to our beloved parents and our honorable Supervisor who helped us with his contineous support Dr. Jia Uddin.

Acknowledgement

All praises goes the Almighty who blessed us with our well being for completing this thesis and always guiding to choose the right path.
We are forever grateful to our Supervisor, Dr. Jia Uddin for his continuous support guide and helping us whenever we face any problem.
Finally, our deepest gratitude to parents, siblings and friends for their relentless support in every step of our life
We are eternal grateful to all of them.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ANFIS Adaptive Network Based Fuzzy Inference System

ANN Artificial Neural Network

API Application Programming Interface

ArSL Arabic Sign Language

ASL American Sign Language

BdSL Bangladeshi Sign Language

BSL British Sign Language

DTW Dynamic Time Warping

GUI Graphical User Interface

HCI Human Computer Interface

HMM Hidden Markov Model

HSV Hue Saturation Value

KNN K-Nearest Neighbor

LMC Leap Motion Controller

MLP Multilayer Perception

MLP Multilayer Perception

NCL Negative Correlation Leaning

NNE Neural Network Ensemble

PCA Principal Component Analysis

RGB Red Green Blue

SDK Software Development Kit

SVM Support Vector Machine

UI User Interface

WHO World Health Organization

Chapter 1

Introduction

1.1 Motivation and Goals

From the very beginning, human beings have tried to communicate and express their thoughts and emotions. Thus, they start expressing them through native languages and gradually they have invented signs or alphabets to fulfill their intentions. With the change of time, the number of population of world increased and people started to divide among themselves having different identity, living in different places etc. Again, there are also people who have physical disabilities such as deaf, mute, visual impaired etc. and they have to communicate following certain ways for example, through making gestures with hands. If we consider a deaf child, it is not easy to learn language since from the very beginning of life they are deprived of voices speaking meaningful words. That is when the use of sign language and standardization comes in people's mind. Different countries make their own gestures to express. As a result, standard form of sign language is not designed yet. In Bangladesh, sign language are different for the deaf and mute people of this country. We have ifferent form of sign language. For making easy communication with this group of people, such services and systems which can detect or help to understand their signs and gestures are highly encouraged.
Since English is the first international language, there are several works regarding British sign language. In our country, currently people have started working with Bangla sign language. Moreover, in 1974 a book named 'Bengali Sign Language Dictionary' which was published by National Centre for Special Education Ministry of Social Welfare [1]. Another book named "Ishara Bhasay Jogajog" was reprinted in 2015 for learning purpose of deaf children in Bangladesh. Being motivated in such field, we have decided to work with Bangla Sign Language detection. Thus our purpose is to learn about the gesture interpretation using "Leap motion" device which is a computer 3D based sensor. Although there is scarcity of resources, we have decided to use this device for its very fine accuracy. We will be working with Bangla Sign Language alphabets which a speech or hearing impaired person will be able to express through hand gestures and which will be detected by the sensor. Although there are two types of gestures- (one-handed and both-handed), we will be more focusing on one-handed gestures. Sole purpose of our research is to help the speech and hearing impaired people so that they can communicate easily and efficiently.

1.2 Project Scope

There has not been many works on Bengali sign language let alone a real-time sign language recognition system. That is why we attempted to build a real-time Bangla sign language recognition system. So users of our system will be able to make gesture and see the text as output of given gesture at the same time. The system will first collect and store sign language gestures performed by a user using a Leap Motion Controller . Then, it recognizes and differentiates between the characters of the Bangla Sign Language alphabet acording to the dataset. Finally it shows output as identified matching gesture with its matching proportion as in terms of accuracy.

1.3 Overview

The research started out with in depth studies in multiple distinctive strategies and techniques that are associated with sign language recognition for both Bangla sign language and other sign languages. For our project we have used a new approach and methodologies to recognize Bangla Sign Language in real-time
In chapter 3 we discussed implementation process of our model and why & how $P recognizer and leap motion were used to build a real-time Bangla Sign Language System.
In chapter 4 we gave an overview of the system and how it is working.
In chapter 5 we discussed the result and analysis of our model and why it is better than the previously researched models.
Lastly in chapter 6 we concluded our research and gave a brief overview of the future works of our model.

Chapter 2

Literature Review

2.1 Bangla Sign Language

Bangla Sign language has been playing a significant role in the method of teaching for hearing impaired children. It has been in practice for more than three decades ago. Here, the use of hand gestures, lip reading and speech training are parts of the whole teaching method. In different regions and countries, the basis, teachings and techniques of sign language are different from each other. Moreover, children from different countries use gestures in sign language according to their own local conditions and needs. Thus it can be said that the techniques and hand gestures which are being used for British language is not the same as Bangla language. Shear steps are taken to improve this Bangla sign language in our country[2].

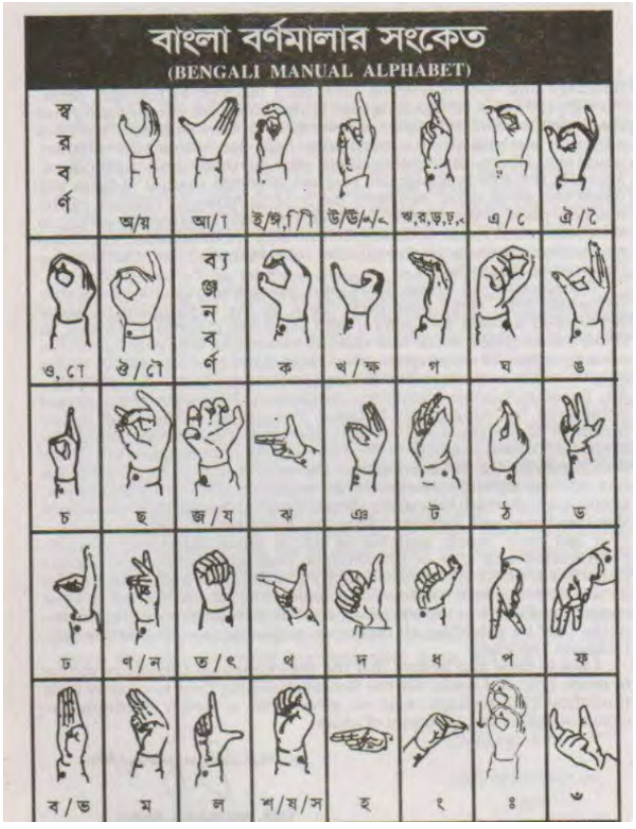

Figure 2.1: Symbols of Bangla Sign Language

In Bangladesh about 2.6 million people are deaf. Again, among the minority communities based on the language, deaf community is the largest community in Bangladesh and about 2.6 million people are deaf in this country[3]. In Bangladesh, Bengali language is the most widely used written language and total there are 49 alphabets. Among which, 11 are vowels- called "sôrôbôrnô and 36 are consonants which are called bænjônbôrnô. For all these alphabets there are different gestures and meaningful words[4]. Figure 2.1 shows the standard gestures for Bengali alphabet.
Again there are two types of Bengali sign language- one type can be represented with one hand and another type can be represented by two hands. Here we will work with mainly gestures which are represented with one hand.

2.2 Gesture Recognition Approaches

Gesture recognition is recognized as a type of perceptual user interface through computer. That user interface works to capture human gesture and interprets as given command. Gesture Recognition is the ability of understanding gestures by computer and execute commands based on those gestures[5]. If we want to understand how gesture recognition works, it is important to understand the general sense of "Gesture". Any non-verbal communication intended to exchange any information specifically can be called a gesture[5]. Gesture includes any large or small physical movement that can be recognized any motion sensor. Gesture recognition works as an alternative user interface for providing real-time data to a computer. To talk about our system, the most important part is the gesture recognition. As we are using real-time data for our system, a suitable algorithm should be used for matching the result which will work very fast and will not affect the user while gesture recognition. If we look at most of the approaches for gesture recognition, those apply machine learning on the frame data of Leap Motion. We cannot say that with confidence that a certain algorithm can show better result than other ones as the condition of testing, variant in sign language and input data can be different. Some algorithms are discussed in subsections 2.2.1 to 2.2.10 which were used in similar work previously. Following the discussion a summary of our understanding is covered in subsection 2.2.11.

2.2.1 Dynamic time Warping

In their paper Vikram, Et al.[6] discussed about the use of dynamic time warping (DTW). The concept of Dynamic Time Warping that it is not dependent on taken time or speed of every input data while make a comparison with them precisely. The gestures which are made at different speed – the DTW can be very functional with those gestures. The authors explained about 2D handwriting gestures and the application of DTW. They suggested after doing research that it could be used with 3D data with extension of earlier version. They conclude with the decision that DTW approach is satisfactory for real-time data input but they opened the scope to further research to explore the increased complexity of gestures comparing with only handwriting. In another paper Santiago[7] said that, in pattern recognition got high versatility with DTW algorithm because the recognition parameters are set as

the requirements for user. They said though the gestures are made in varying speed, DTW can achieve the high accuracy. This is the advantage of DTW.

2.2.2 K-Nearest Neighbor and Support Vector Machines

Chuan, Et al.[8] proposed K-Nearest Neighbor (KNN) and Support Vector Machines (SVM) algorithms for gesture recognition in his paper. He proposed these algorithms for American Sign Language (ASL) using the Leap Motion Sensor. They did the research with 26 letters of English language, analyzed and found out that these two algorithms can initiate the accuracy rate of 72.78% and 79.83% respectively. The authors showed the reason behind low accuracy working with these two algorithms. The authors mentioned, in BSL two hands are using for the gesture creation. Comparing to BSL, we can see that ASL using only one hand. As per the fact, few alphabets are very close to each other for representation which cause to failure of classification of exact gesture with Leap Motion Data. If the authors would use BSL, they could get improved accuracy as BSL uses two hands for gesture creation and they are more different therefore.

2.2.3 Fuzzy Inference System and K-Nearest Neighbor

Fuzzy Interface System and K-Nearest Neighbor (KNN) were proposed by Mufarroha, Et al.[9] in their paper where Adaptive Network Based Fuzzy Interface System (ANFIS) worked to divide the hand gesture of ASL into three categories- gripped fingers, upward facing fingers and sideways facing fingers. After categorization, KNN was used for faster recognition process of the grouped gesture. The authors find it easy to extract the features of the gestures first by giving the sophistication of the ANFIS algorithm. The computation time was ease by using KNN method. The result of gesture recognition got the aggregate accuracy of 69.30%, 80.77% and 70.77% for three different scenarios in their research by using both of the methods.

2.2.4 Gabor filter, Kernel PCA and SVM

In their paper Uddin, Et al.[10] proposed a model on recognition of hand gestures. The authors reported that they converted the RGB images of hand gestures into HSV. The desired feature extraction was done by Gabor filter which resulted in a high dimension output. The Gabor Filter can perform in feature extraction properly nevertheless of the condition of light. Kernel PCA method was used here to reduce the dimension of extracted output. Finally, they used SVM for training and classification of the images. The training and accuracy testing part were done with equal weight of data. The authors got a very high accuracy in testing part- 100% in standard condition and significantly high accuracy by changing in different angle – 98.6%.

2.2.5 $P Point-Cloud Recognizer

Vatavu, Et al.[11] introduced and designed the $P recognizer ($P) as a gesture recognition algorithm. The authors mentioned in the paper, the $P recognizer is “a 2-D gesture recognizer designed for rapid prototyping of gesture-based user interfaces”. The $P treats the gesture as groups or “clouds” of points and then the method assess

each groups or clouds one by one. Thus the method can evaluate the complex task of comparing gestures. The authors described the $P as a very accurate recognizer as it removes any ambiguity from the signs by the help of creating point cloud. This fact makes it simpler while doing comparison and recognition. According to the authors, $P delivers more than 99% accuracy in user-dependent gesture recognition.

2.2.6 Hidden Markov Model

Chen, Et al.[12] proposed the use of Hidden Markov Model (HMM) to support the motion recognition of 2D and 3D data. Hidden Markov Models are generally recognized for using as pattern-matching method in typing prediction or speech recognition. Markov Model is defined as a state-network where the states are connected with each other having a certain weight. In this method, the following state will be reliant only on the current state and there is a link of probability within these states. The difference between Markov Model and HMM is that its states are partially hidden. In speech recognition waveform can be observed without revealing the spoken words. The same way can be used in gesture recognition system where the movement of data is given but the real gesture is not revealed. The authors mentioned that they found the recognition accuracy with 91.9% in user-dependent testing and in user-independent testing they got 96.9% accuracy with this HMM algorithm.

2.2.7 Artificial Neural Networks

Mohandes, Et al.[13] proposed the use of Artificial Neural Networks (ANN) in Arabic Sign Language (ArSL). They showed the use of ANN in particular Multilayer Perception neural network (MLP). ANN is a machine learning algorithm where we can find the similarity to human brain. Human brain is composed of neurons which work like central processing unit and they are interconnected and having a determined weight. The provided test data works as experience for the network and ANN learns from experience. The method then calculates outputs specifically for the inputs. The authors found the accuracy over 99% in classification done with Artificial Neural Network algorithm. Though the accuracy level is very high with this method, the testing resulted in little erroneous result due to similarity between some gestures and the obstruction of fingers by palm of the hand or other fingers. The authors suggested placing another Leap Motion device beside the user. This method can resolve the problem theoretically. Another research was done by Potter[14] where ANN is used to recognize Australian Sign Language with Leap Motion Controller. They also trained the network before using it as a recognizer. Naglot, Et al.[15] worked with Indian Sign Language (ISL) using the Leap Motion Controller which was based on ANN algorithm. The authors used normalized dataset before providing to the network for training and testing. For classification they used MLP neural network with Back Propagation algorithm. Their proposed System was improved in performance and accuracy with ANN algorithm.

2.2.8 Artificial Neural Network and Support Vector Machine

Chowdhury, Et al.[16] used Microsoft Kinect to collect the dataset where they performed hand gesture in front of the camera. The Kinect camera can detect wrists and joints with the help of contours. SVM method was used for classification of the gesture after the extraction of contour features. For Recognition they first used the convex hull method as contour detection to extract the features of those contours. Then, the features were sent to the SVM for recognition. After getting an input as RGB image, they converted it to a grayscale image and finally to a threshold image. Then the joints and bends and angles were detected by the convex hull algorithm. Then SVM collect the data of joint detection and match with the datasets. After that the final match is announced. Their proposed system resulted with an accuracy of 84.11% with these algorithms.

2.2.9 Key maximum curvature points and 3D chain codes

Key maximum curvature and 3D chain codes were used by Geetha, Et al.[17] in their research as a unique approach for recognition of Indian Sign Language (ISL) characters. To understand a word in its entire motion, they considered the facial expression along with the stroke based recognition system. The key frames were extracted only with efficiency thus the accuracy had increased with a lower time complexity. The authors used Key maximum curvature points algorithm for identification of key frames with most prominent curvature. After recognition of keystrokes using Stroke Sequence Identification method, error-detection and rectification part was done with HMM. Using the stroke decision tree, the keystroke sequence was matched with the words finally and the resulted output was shown.

2.2.10 Artificial Neural Network, Canny Edge Detection

Ahmed Et al.[18] used 518 still images of 37 Bangla sign language alphabets for their datasets and used Artificial Neural Network, Canny Edge Detection and Back propagation algorithm to recognize the sign gestures. They recognized the letters based on fingertip position. The best test set classification accuracy is observed 98.99% for BSL-FTP for 100 hidden neurons. Only two patterns of 198 test patterns were misclassified in their model.

2.2.11 Research Summary

The above discussion shows that there are number of studies have been engaged in sign language recognition with different country's' sign language. Some these researches have been done with Leap Motion Controller and others with similar devices. Using different algorithms and methods theses research got varied output and showed their level of accuracy as well as success.
As we are doing similar research in Bangla Sign Language with Leap Motion Sensor, we got plenty of ideas from these related researches. As these algorithms showed specific accuracy in different sign language, we can say that those algorithms and methods will also work similarly in Bangla Sign Language. So, repeating those

Table 2.1: Summary of Literature Review

Author Name	**Dataset**	**Methods**	**Accuracy**	**Real-time Detection**
Chuan, C.H. Et al. [8]	26 English Alphabets of 2 Sets	KNN, SVM	72.78% & 79.83%	Yes
Murfarroha, F. Et al. [9]	26 English Alphabets of 5 Sets	ANFIS, KNN	86.92% & 80.77%	No
Uddin, M. A. Et al. [10]	2400 images of Bangla Sign Language	Gabor Filter, Kernel PCA, Morphological Operation, SVM	99%	No
Chen M. Et al. [12]	20 set of hand gestures	HMM	91.9%, 96.9%	No
Mohandes, M. Et al. [13]	10 gesture sets of 28 Arabic Alphabets	ANN	99%	No
Chowdhury A. R. Et al. [16]	800 iterations of 38 BdSL alphabets	ANN, SVM, Convex Hull Method	84.11%	No
Naglot, D. & Kulkarni, M.[19]	26 English Alphabets of 20 sets	MLP neural network with Back Propagaton algorithm	96.15%	Yes
Ahmed, S. T. Et al. [18]	518 images of 37 BDSL alphabets	ANN, Canny Edge Detection, Back Propagation	98.99%	No
Rahman, M. A. Et al. [20]	3600 images of 6 Bengali vowels & 30 Consonants	KNN	96.46%	No
Karmokar, B. C. Et al. [21]	265 samples of 47 BdSL Signs	NNE, NCL	93%	No
Deb, K. Et al. [22]	80 Hand Signs	RGB Color Model, Color Segmentation & Template Matching	96%	No
Rao, G. A. Et al. [23]	200 signs of 5 persons in different angle & different environment	CNN	92.88%	No
Elbadawy, M. Et al. [24]	25 Hand Gestures of Arabic Sign Languages	CNN	98%	No

methods would be time consuming and nothing would be explored in a fresh manner. There are already many recognized algorithms available in wide range so we did not think of creating any entirely new machine learning algorithm. It would be beyond the scope of our research aim. Instead, we considered finding what areas and methods are not yet approached in a wide range and explored fully. A summary of our findings is shown in Table 2.1.
We found that $P recognizer, studied in subsection 2.3.5, is not incorporated in real-time Bangla Sign Language recognition yet. The algorithm is yet to be approached in a wide range in gesture identification. We thought that it would be more beneficial in this specific area if the result, analysis and accuracy could be recorded and performance and capabilities shown in more detailed manner. The characteristics of $P recognizer recommends about the credibility of modification in 3D gestures. The effectiveness is to be seen in rest of the work with $P Point-Cloud Recognizer.

Chapter 3

System Implementation

3.1 Proposed model

Our proposed model recognizes Bangla Sign Language in real time. It has two major features. First one is **Data Collection** and the second one is **Recognition**. In data collection part, we collect the data from the users. Here, users can be any person who can make the gestures of Bangla Sign Language accurately. These data are used for training our model. We have built an easy and simple application to collect the data and for recognition purpose.
At first, datasets are collected using the leap motion sensor. Leap motion controller captures the frame and store the points from the frame in a gesture array list as the provided gesture. This gesture represents a specific alphabet. These points of the gestures are 3 dimensional vector points or co-ordinates. Before recognition, these points become normalized using the $P Point-Cloud Recognizer algorithm for matching with the given gesture[11].
In recognition part, sign language or gesture is shown by the user who wants to communicate via the model. When a user shows a specific alphabet, leap motion controller detects the frame and converts it into gesture-point array. Then, it runs $P Point-Cloud Recognizer algorithm on the gesture for recognition. We have modified the algorithm for 3D points since it was introduced for 2D gestures only. The $P Recognizer algorithm normalizes the point array by re-sampling, scaling and translation. After that, it finds the Euclidean distance between the saved gesture points and the shown gesture points using greedy cloud match method. Later, the distance is converted to the accuracy of the given gesture in comparison to the gestures from our dataset. $P Point-Cloud Recognizer is used in our model as it has a lower time complexity- $O(n^{2.5})$ and for faster recognition of a gesture (only 32 milliseconds)[11]. Finally, if the accuracy is above 60%, we have considered it as the matched alphabet and shown the particular alphabet as output.
Our proposed model identifies the gestures in real time. As we have used $P Point-Cloud Recognizer, it takes a less significant amount of time which helps the users to communicate efficiently without any delay. The flowchart given in Fig. 3.1 shows the steps of our proposed model:

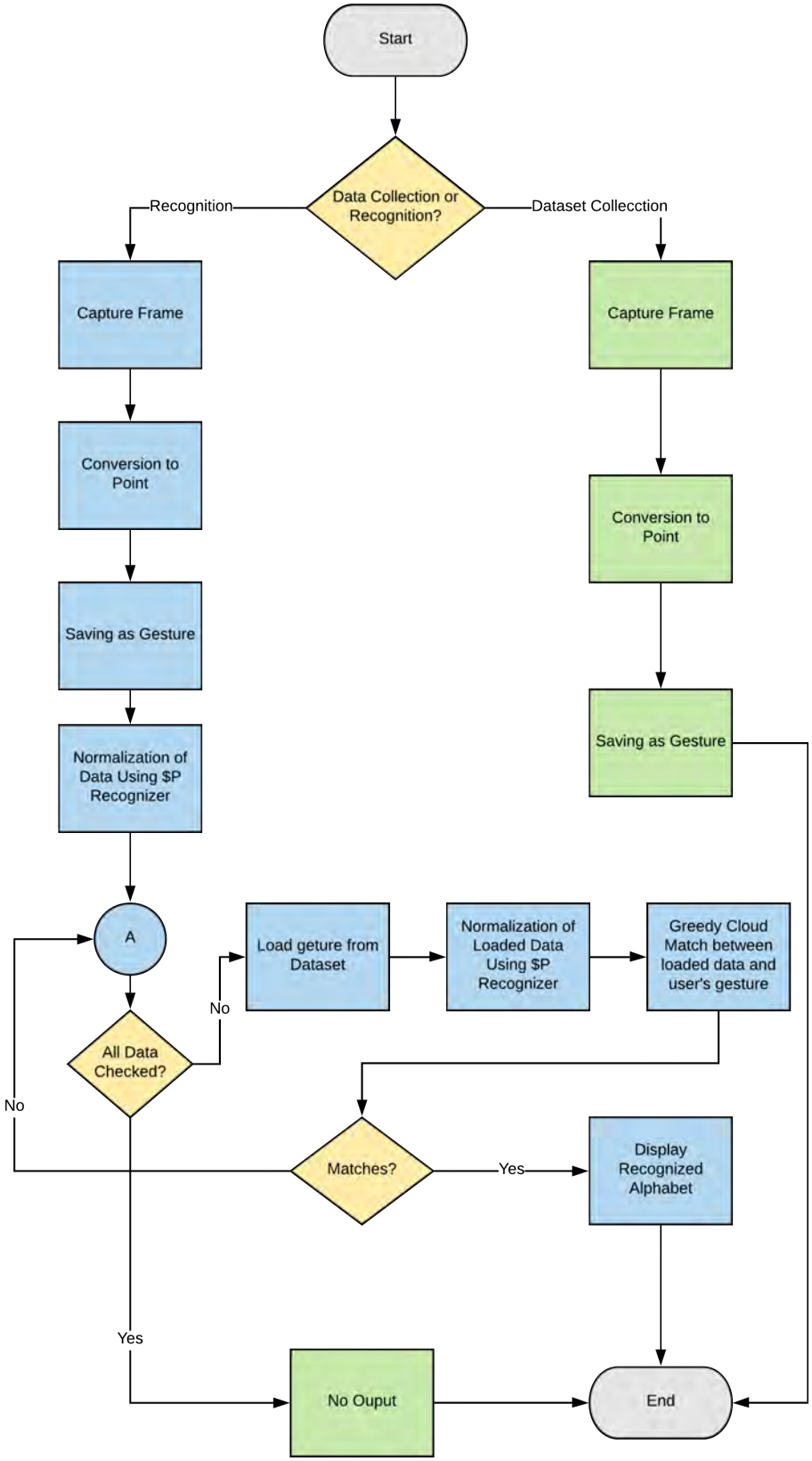

Figure 3.1: Flowchart of the Proposed Model

3.2 Application Design

To make the data collection and recognition easier, we have built an application. As we have built the application for our research purpose, we have made it as simple as possible. This application has 3 activities. We have discussed the application using Use Case diagram for better understanding. Use case diagram of the application is shown in Fig. 3.2.
At first, leap motion visualizer shows the hand with gesture which is provided by the user. Here, User and Visualizer work as primary actor and secondary actor respectively. User wants to display the hand using the application. So, he has to show his hand in front of leap motion device within the allowed radius because, leap motion sensor cannot sense any gesture beyond the Leap Motion's capture radius.

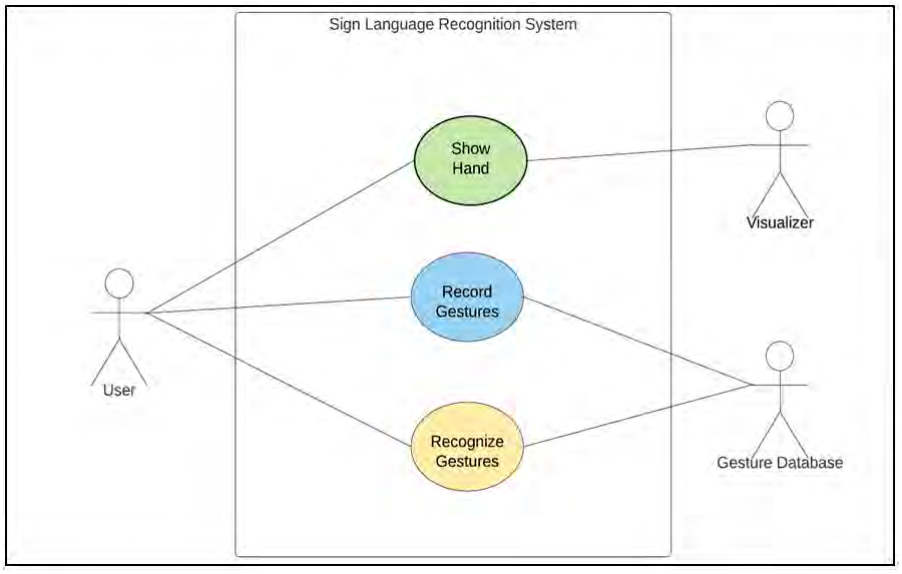

Figure 3.2: Use Case diagram of the application

Again, user can view two options in the main menu. Either he can choose record gestures or recognize gestures. In record gestures, the data provided by the user will be saved as the dataset in gesture database for future recognition. Basically, we train our model through this activity. Here, User who is a trainer or sign language expert is the primary actor and Gesture Database is the secondary actor. User has to holds his hand still for a particular period of time. He has to move his hand in a specific velocity range. Most importantly, as we have said earlier, user has to show his hand in front of leap motion device within the allowed radius as leap motion sensor cannot sense any gesture beyond the Leap Motion's capture radius.
The last and final activity is to recognize gesture. In the main menu, the second option is Recognition. After training the model, when we want to check whether our proposed model works or not, we select the recognition option. In recognition, user acts as primary actor and gesture database as secondary actor. Here, user is the common people who know about sign language or the person with hearing and speech disability. Again, user has to follow the preconditions like recording for successful use of the application like maintaining specific velocity of hand movement, showing the hand for a particular amount of time and within the allowed radius.

3.3 Leap Motion Controller

Human Computer Interaction- HCI is becoming more important day by day. Main focus of HCI is to ease the difficulties faced by human being in an efficient manner. Motion sensing devices are part of HCI. There are various motion sensors which are used for different purposes. Kinect is a famous motion sensing device produced by Microsoft. It can detect the full human body.
In our model, we have used Leap Motion Sensor which is shown in Figure 3.3.

Figure 3.3: Leap Motion Controller

Though it is not famous like Kinect, we have chosen this device for its low cost and better accuracy. Leap Motion Controller (LMC) is a small USB device. It has the size of $1.2 \times 3 \times 7.6$ centimeters [25]. Leap Motion Inc. introduced this device in 2013. This device is different from Kinect as it can detect the motion of human hands only. It cannot detect full body motion like Kinect. This device can detect up to 200 frames data of hand gesture per second[26]. Besides, it can detect 150°field of view with around 0.227 cubic meter of 3D plane[27]. Leap Motion Device can detect various kind of hand gestures like moving hand, moving one hand over another, pinching, stroking, crossing fingers etc. This device can detect and track hands, fingers, bones, joint of the bones and palm position accurately[19]. LMC recognizes a gesture using right handed Cartesian coordinate system which is shown in Figure 3.4[19][28]. At the top of the device, origin is centered. X-axis of it is situated in the horizontal plane and goes parallel to length of the controller. The Y-axis is in vertical plane, it goes upwards with positive direction. The Z-axis has positive values increasing toward the user and it is situated in the horizontal plane[29][30].

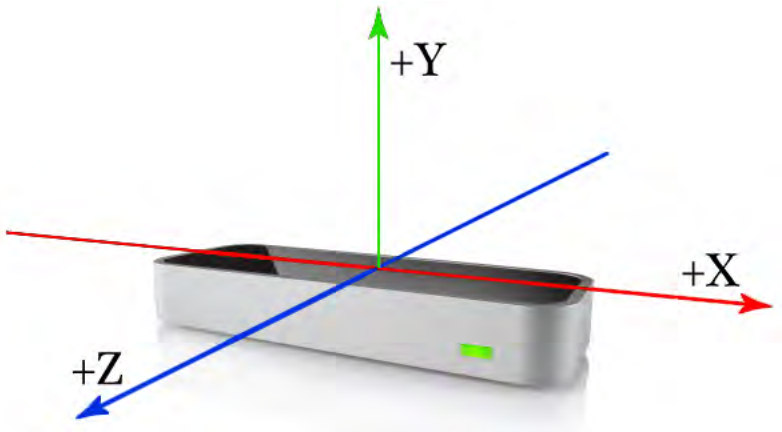

Figure 3.4: Leap Motion Co-ordinate System.)

Leap Motion Sensor can be considered as a high functional camera. It has two monochromatic Infrared (IR) cameras and three LED lights. Internal architecture of the controller is shown in Fig. 3.4 (a) and (b). Using these two IR cameras and 3 LED Lights, leap motion sensor generates 3D patterns of IR lights[26].

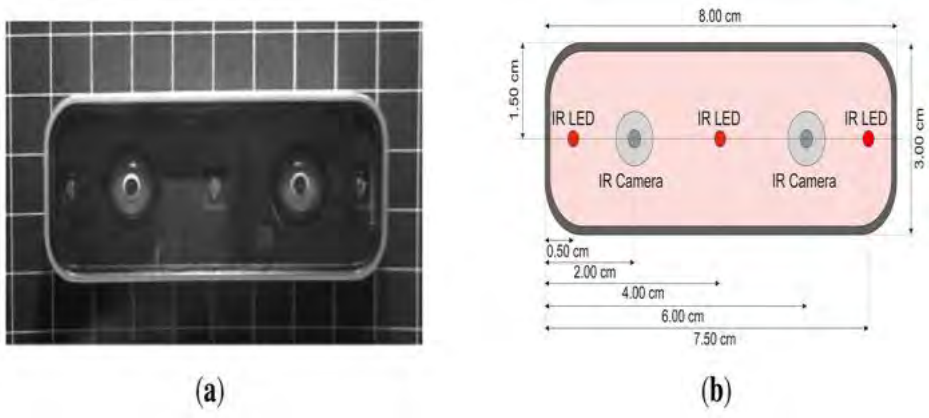

Figure 3.5: a) Leap Motion Sensor with 2 IR Cameras, (b) Internal Architecture of the Controller[19]

These patterns then go to the dedicated computer. Through the internal mechanism of Leap Motion Controller, Leap Motion visualizer then shows the hand gestures in 3D shape as the output which is shown in Figure 3.5.

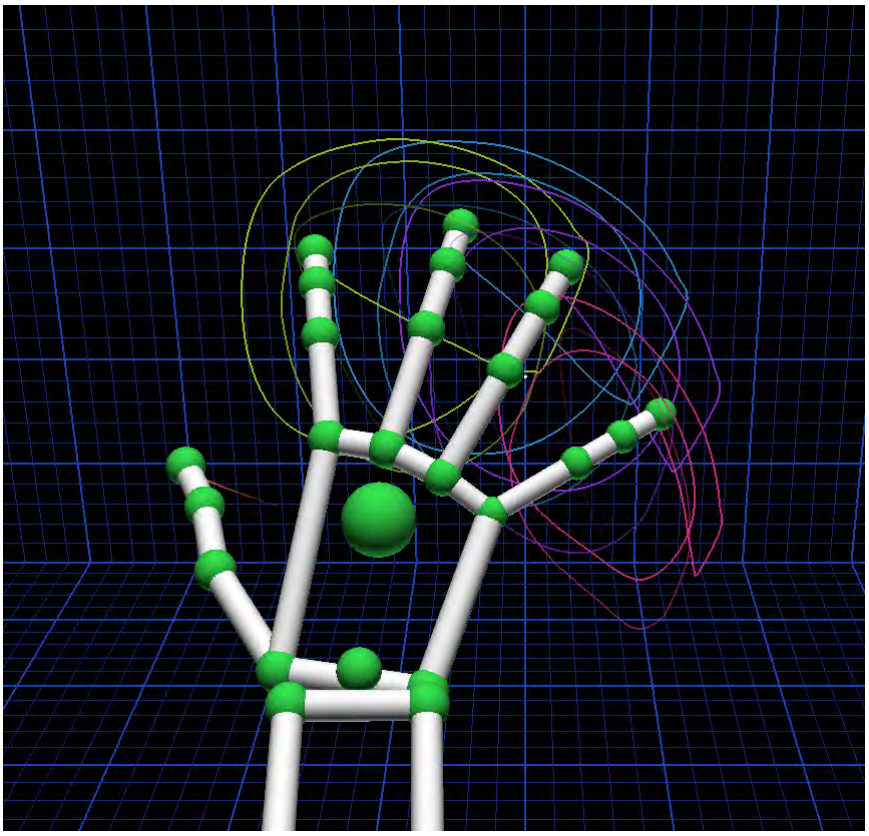

Figure 3.6: Leap Motion Visualizer

3.4 Leap Motion Integration

Leap Motion Controller has to be integrated with our application in order to record and recognize gestures. Leap Motion Controller captures data and saves them in a series of frame. A particular frame contains specific gesture or position of the Hand. To get the frame, we can use any of the two processes. One is polling the device and another is calling method from the listener. Leap Motion Controller builds new threads for every new frame in the second process. Then, it will wait until the current thread finishes execution. This process prevents thread-flooding. Thread-flooding happens when threads are built faster than the device's ability to process. For this reason, we have chosen the second option for our leap motion integration. This process is very helpful as it allows us to record the gestures without considering the poll rate. If this process did not check thread-flooding, some frame could be missed or became duplicated

Leap Motion API helps the user to access the raw information or data of the gestures. These raw data are used for recognizing the gesture accurately by the gesture recognizer. Leap Motion API supports various programming languages like Python, Java, C#, C, JavaScript etc. It makes the device more users friendly. SDKv2 of the leap motion controller has special skeletal view which can provide the view of hand with all fingers in 3D space. It can show rotation of open hand, hand grabbing, pinch strength etc.[25]. We have used Skeletal Model as visualizer and SDK version 2.3 in our model.

We have used Java as the platform for our application. Leap Motion API is connected to our application by listener and controller class. Controller class is basically built for controlling or using the leap motion controller. It can manipulate the frame by polling but as we have used the second process for integration, we have built a listener class with the controller class. The listener class contains various methods which is used to manipulate the controller class. For example, onConnect() method is called when a listener is connected to a controller. onFrame() method is used for frame detection. When LMC finds a new frame of hand gesture, onFrame() method will be called. onExit() method is called when a listener is disconnected from the controller.

JavaFX is used for the Graphical User Interface (GUI) of our application. It is an open source library of graphics and various media packages written in Java programming language. JavaFX allows our application to easily integrate with the leap motion controller.

3.5 $P Point-Cloud Recognizer for Gesture Recognition

There are various gesture recognition techniques, for example Hidden Markov Models, graph theory, statistical classifiers which is feature-based or even mixture of classifiers. To work with these algorithms, we need technical knowledge and understanding to use in a new platform. Among several algorithm approaches, we have decided to use $P Point-Cloud Recognizer algorithm designed by Vatavu, Et al.[11] which is a member of $-family recognizers. This algorithm can be considered as low-cost, high performing and easy to understand approach. In fact, the "$" symbol

in the name is given for the low cost functionality simplicity and as its implementation is easier. The "P" symbol denotes "Point-clouds." So, the term "$P" means point-based recognizer in place of stroke based recognizer which were used earlier with simple, low cost and easy implantation ability. The $-family has got simple geometric computations and clear internal representations and the techniques that are accessible for example $1 and $N have pseudo codes which developer can code in JavaScript, C# and object- C. Although this $-family have got effective approaches, there are some limitations. The $1 can handle only 'unistroke' gestures which are a set of x-y sample points created in single gesture by fingers on digitalized surface. These gestures represent characters, symbols etc.[31][32]. $N and $N-Protractor are introduced to handle multistrokes, which takes multistrokes as unistroke by connecting individual strokes so that they can be seen as a single stroke which is obtained by $1. Following this approach, we can get different stroke order and direction by drawing same symbol (shown in Figure 3.7) due to which $N generates all possible number of permutations of a set of given multistrokes that might cause huge problems in execution time and storing[33][34]. However, to overcome this problem $P algorithm was introduced.

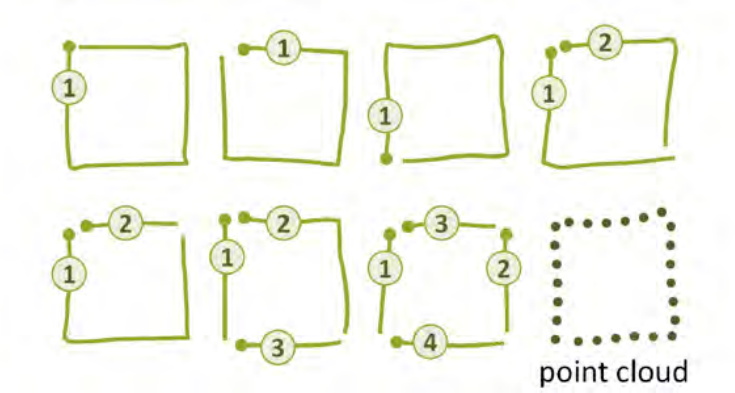

Figure 3.7: Strokes Point

$P algorithm is an approach which ignores variable user behavior in case of stroke order and direction and it represents strokes as cloud points. This algorithm has got low complexity, high accuracy rate and low barriers of adoption. $P algorithm also has 98% of average accuracy and above 99% of accuracy for user dependent tests (experiment based on 10 participants)[11].
$P algorithm does not follow any particular order for strokes or points. For example suppose strokes represented in the form of $U = (X_i, Y_i)|i = 1..n$. Here, U_i follows neither U_{i-1} nor U_{i+1}. Again, U_1 does not necessarily have to be starting point nor U_n has to be the ending point. Here the gestures are unordered sets grouping together popularly known as clouds[11]. If we closely observe Figure 3.7, it is not necessary to categorize whether its unistroke or multistrokes. It becomes necessary to match the point cloud of the candidate gesture (which should be recognized) - "C" to the point cloud of each dataset "T" in training set and find out the matching distance. If we follow the Nearest Neighbor approach, we will select the dataset with the smallest distance from candidate gesture and find the result[1]. If the distance between C and T is represented in terms of a function M where any point $C_i \in C$ with exactly one point $T_j \in T$, relation between, C, M and T becomes $T_j = M \times (C_i)$. We can say minimum matching distance; M is the sum of Euclidean distances for

all the pairs of points from M:

$$\sum_{i=1}^{n}(C_i - T_j) = \sum_{i=1}^{n}\sqrt{(C_i \cdot x - T_j \cdot x)^2 + (C_i \cdot y - T_j \cdot y)^2} \tag{3.1}$$

If we observe Figure 3.8 (a) and (b), we can see that time ordered alignment fails when one spiral drawn backwards in $1 and $N as $N makes all permutations of all orders and directions in training set. On the other hand, in Figure 3.8 (c) $P approach alignments of point clouds avoids execution details and operates correctly[11].

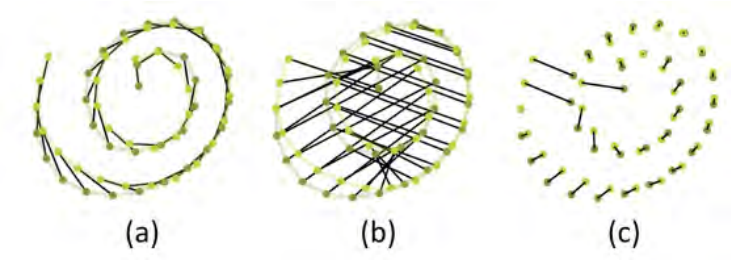

Figure 3.8: Spiral shape drawn in order by $1 and $N (a) in backward order it fails to align (b) disordered $P approaches aligns correctly

Though Hungarian recognizer has better accuracy of 98% accuracy, it is not suitable for simple and user friendly user interface prototypes. Its complexity is O(n^3) which is not good for faster recognition.
The $P algorithm was invented after several experiments and approaches by the authors[11]. Vatavu, Et al. have explained each and every detail which they have compared with another algorithm known as Hungarian Recognizer. For matching the cloud points, some heuristic approaches were done by them in which Greedy-5 approach was accepted for $P recognizer[11].
In Greedy-1 approach, all Euclidean distances between the n points of cloud C and template T are computed and sorted in ascending order where the first n pair of matched points is taken. Complexity- $O(n^2 \times log(n))$.
In Greedy-2, if C_i is the first cloud, it will find its closest and unmatched second cloud. Once it is matched, it will continue with C_{i+1} until all points of C are matched ($i = 1..n$). Complexity- $O(n^{2+\epsilon})$ was found for this approach, which is not suitable as it is like Hungarian Recognizer as value of $\epsilon > 1$. After running the approach, the result varied with the order, thus the author decided to run with multiple times from different starting points. Taking C_k as starting point, it follows the following equation (2) for getting the minimum match:

$$\sum_{i}(C_i - T_j) = \sum_{i=k}^{n}(C_i - T_j) + \sum_{i=1}^{k-1}(C_i - T_j) \tag{3.2}$$

Here i represents all the points of C particularly.
Finally, the author comes up with a Euclidean distance with weights to encode the confidence in each pair (C_i, T_j) computed during greedy run (Greedy-5 approach).

This comes with run-time $O(n^{2+\epsilon})$, where the value of ϵ differs according to experiments. It follows the same equation as equation (2) but weight is added here,

$$\sum_{i} w_i \cdot (C_i - T_j) \tag{3.3}$$

The author weighted first match with $w_1 = 1.0$ (meaning high confidence). Here the first point has got all data in order to make a decision for the closest match. Running the algorithm, we get few points remaining for the rest of the points of second cloud. To recognize the rest of the points, the author weighted them with confidence values $[0 - 1]$ which are of low confidence. Thus the last point of first cloud will have only one option left which is the last point of the second cloud and these all will be done based on confidence values assigned on them[11]. The following equation will be followed in case of weighing the points:

$$w_i = 1 - \frac{i-1}{n} \tag{3.4}$$

Here, $i = 1..n$ means current step of the approach.
Analyzing the last approach (Greedy-5), we can observe that, for user dependent tests the performance was $\epsilon \geq 0.5$ which is similar to Hungarian Recognizer. Again in case of user- independent tests, significant differences were obtained between two recognizers while performing the last approach. Here the author have found the highest accuracy with value $\epsilon = 0.5$[11]. Thus, the last approach was named as "$P" by the author which has got the best performance and execution time compared to any approaches he has performed before. This $P Recognizer algorithm has only seventy lines of code.
This approach was the best considering three performances. First one is effect of number of training samples. The author stated that for user dependent testing, the $P reached 98.5% accuracy with three training samples per gesture type in terms of effect of training samples quantity. When more than five samples were used, the accuracy rate reached to above 99%. Again, second performance consideration was effect of number of training participants. For user independent testing, $P achieved 90% recognition from just one training participant compared to $N has got 86% and Hungarian 88%[11]. For 10 participants the accuracy rate reached to 99.5%. The final performance consideration was the execution time. $P has $O(n^{2.5})$ complexity and can categorize the gesture of candidate within 32ms. This is ten times faster than Hungarian approach. Considering these performance issues, we have selected $P Recognizer as the recognition process in our proposed model.

3.6 Feature Extraction

In the above, we have discussed briefly about the works of Vatavu, Et al. and how they have come up with $P Recognizer. Our application is build using the adapted version of $P algorithm. Since the pseudo code is available, we have implemented

the algorithm in Java programming language and have modified the 2D version og the algorithm into three dimensional one so that we can recognize 3D gestures using Leap Motion Sensor.

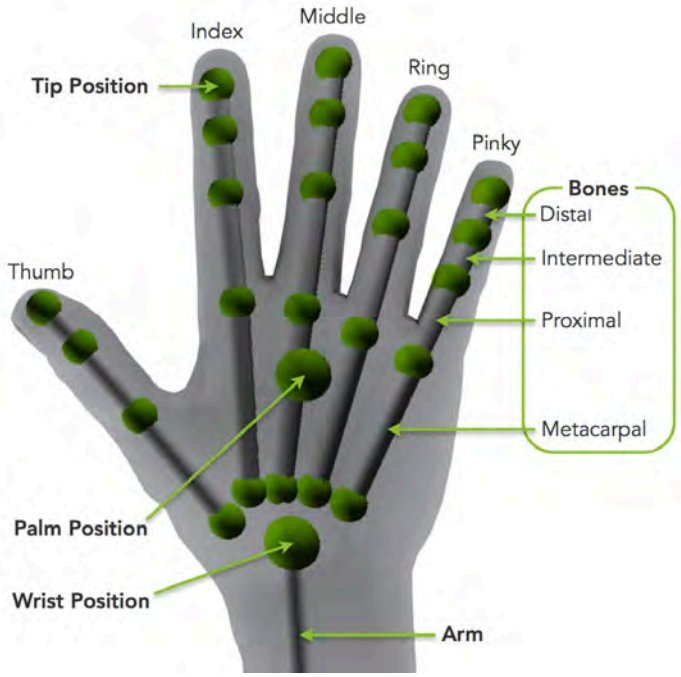

Figure 3.9: Features provided by LMC API[35]

In our work, $P algorithm is used to find the possible type of for an object. This is done by measuring the euclidean distances between the candidate object and all available points of the object from the dataset to determine the closest match. $P algorithm is "instance- based nearest neighbor classifier with Euclidean scoring function" because it compares a unknown object with the recorded known object. Here, objects are categorized based on features which are characteristics or property of an object to make it unique.

Table 3.1: Used Hand Features

Palm Features	**Finger Features**
Stabilized palm position	Finger direction
Palm normal	Metacarpal start position
Palm direction	Metacarpal end position
	Proximal end position
	Intermediate end position
	Distal end position

Here we have used some features of palm and fingers shown in Table 3.1 are described below:

1. **Stabilized palm position:** This is the distance between the leap motion controller's origin and the center point of the palm position of a hand.

2. **Palm normal:** It is a vector pointing in same direction as palm's normal means 90 deg with the plane. It points in downward position.

3. **Palm direction:** It is a vector that shows direction from palm towards the fingers.

4. **Finger direction:** It shows direction of the finger pointing at.

5. **Finger start position:** The starting point of metacarpal - closest bone near the wrist.

6. **Finger end position:** The end of distal bone which is closest to the fingertip position.

Velocity of each hand is set within the frame. The velocity of all detected hands (previously given) are measured with each newly found image data by LMC. By computing the velocity of palm and maximum velocity of each finger on hand, the velocity of whole detected hand is obtained. There is a threshhold or minimum velocity range set in the system which is by default 300 ms^{-1}. Leap Motion considers a gesture if velocity of it is above the minimum range. If it recognizes something below the minimum range, LMC will record it as a 'pose'. When the leap motion device detects a gesture of hand, it waits for 50 frames. When the device gets the motion velocity threshold for 50 frames, it captures a single frame to show as the pose. Then, the pose is saved in memory as a gesture which contains all features discussed above.

3.7 Modification of $P Point-Cloud Recognizer for Gesture Recognition

Studying about the $-family and the origin of $P algorithm, we have come to know that it is designed to improve the recognition of handwritten two dimensional gestures. Analyzing about its use in the area of recognition of gestures, we have decided to use this algorithm for our 3D gestures captured by Leap Motion device.
$P algorithm takes a 2D gesture as series of evenly spaced points on flat plane (popularly known as point- cloud among authors). Here with the first point we have assigned a stroke ID of 1 for all points it created. In case of 3D gestures, we have adapted the same technique for z axis as we have for x and y axis. The modified algorithm has a Z dimension for 3D gesture recognition which is given below:

$$\sum_{i=1}^{n}(C_i - T_j) = \sum_{i=1}^{n}\sqrt{(C_i \cdot x - T_j \cdot x)^2 + (C_i \cdot y - T_j \cdot y)^2 + (C_i \cdot z - T_j \cdot z)^2} \quad (3.5)$$

The problem that we have faced is we are no longer taking a single stroke, we have to

take points from multiple hands, fingers, bones etc. thus here idea of stroke id cannot be applied. Again only recording fingertip in 2D gesture would not give a good result and will cause high gesture comparisons. Thus we must consider more factors such as- stabilized palm position, palm normal, palm direction, finger direction finger start and end point as additional features for 3D gesture recognition. So instead of

assigning to a single stroke we have assigned same stroke ID to all generated points of a gesture. We could get better accuracy for the 3D gesture recognition or go further if sufficient sources of code of the device were available.

3.8 Normalization

For fair and accurate result, normalization of both a given gesture and a saved gesture with which the given gesture needs to be compared is needed. This brings fair and accurate comparisons. All stored gestures are normalized from being loaded from file till storing in the memory and new gestures are normalized before they are passed to the algorithm for comparison.
First of all, the gestures will not perform always in the same position. Thus the feature values which are framing each gesture are translated to an origin of (0, 0, 0). This process is known as translating to a known origin. Besides, the size of hands along with the feature values might vary from person to person which might result various gesture sizes. To solve this problem, the points are re-scaled to range [0-1]. Finally, the obtained gestures are re-sampled into specific number of points.Resampling is a well known process in gesture pre-processing. It helps to maintain equality and consistency of input data for recognizers. This part helps in case of gestures which are in motion. For example- comparing gestures that last for 5 seconds would not be appropriate if we compare with the gesture which last for 10 seconds. Moreover, gestures that are created of a single frame of hand data are re-sampled to same number of points although essentially they are kept unchanged. Considering the frames in which gestures are captured can contain three features for palm and six features for each finger, together they create 33 featured points for a single hand. After capturing the gestures into frames comes the main part. Here we have performed a machine learning approach to evaluate the gestures.

3.9 Machine Learning

The sum of Euclidean distance between each of their feature points helps evaluating gestures. From a given point, the distance is calculated to all the other points until the nearest one is found in a given unknown gesture. This process continues repeatedly until the distance of all points are found and recorded. After the process has been repeated the gesture with the closest match to the saved one is considered with 60% of minimum accuracy. Machine learning is an approach to make the algorithm work better and give more accuracy. The set of training data contains number of features also specific label which identifies an object. Thus running the algorithm over unknown gestures can categorize based on known training data and the process is called supervised learning.
If we consider an example- for Bengali letter "Ka (ক)" saved with feature points valued X, Y and Z, we need to give gesture having same features which is different from other Bengali gestures. Again if we add an extra recording of "Ka (ক)" with feature values $X + 1, Y + 1, Z + 1$, there would be wider degree of tolerance for any provided gesture to match it. Here we might give gestures which can be slightly different from previous one in that case new match could be generated. As the number of attainable category increases it helps to extend the number of testing

samples and improve the accuracy of recognition.

Chapter 4

System in Operation

4.1 Initial State

When we start the application, a window will open displaying the main menu of the application. The menu contains two different 3D buttons which has access to two different screens. The users can access to these screens by creating a faucet motion within the Leap Motion sensor's line of vision with their hand, as if they were truly pressing the button. The main menu UI of the application is shown in Figure 4.1. The UI is made simple intentionally for the ease of the users.

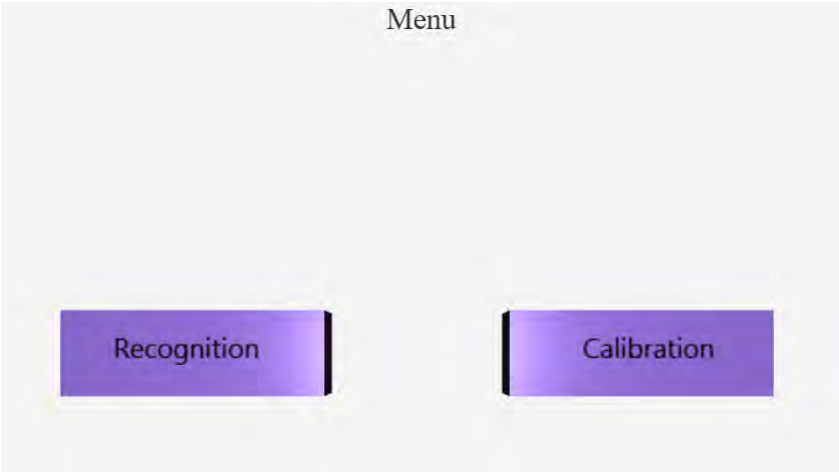

Figure 4.1: Main menu UI of the application

4.2 Skeletal View of Hands

When the Leap Motion sensor detects hands in it's line of vision, a skeletal representation renders in the window. The skeletal model mirrors the movement of the user's hand like a clone. So if there remains any defect in the Leap Motion's capturing, it will affect the application too. For instance, if Leap Motion fails to detect a hand properly, the model may appear disfigured in the application. This is in one way helpful as it features situations where the Leap Motion is to blame, instead of the user. A skeletal model is shown in Figure 4.2.

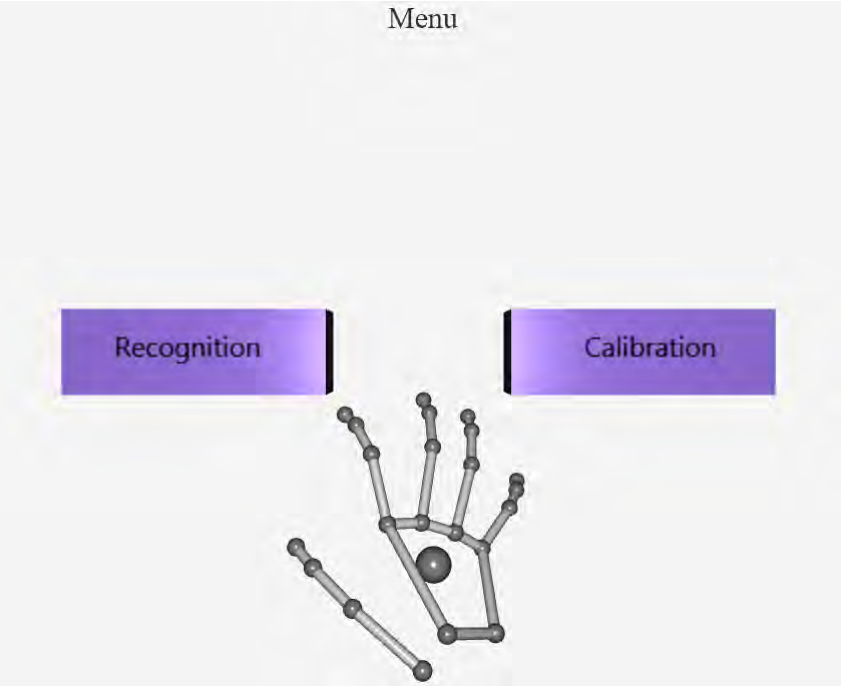

Figure 4.2: Skeletal View of Hand in the Main Menu

4.3 Dataset Collection

When the user presses the 'Calibration' button in the main menu a screen similar to figure 4.3 will appear. With this process, we can improve the recognition system of the application with the help of various gesture sets. These gesture sets are categorized by their alphabetical names. For each alphabet, an image will appear in the top-center position of the screen. This image contains the standard sign language pose as per dictionary of Bengali sign language. The user will give the gesture according to the shown image but in one's own manner. This process is mainly our training process for the system. We have 50 gesture sets from 20 users and each set contains 10 Bengali alphabets. So we have in total of 500 character information stored for our dataset. This improves the accuracy of the recognition algorithm as there are more extensive types of objects to choose for each given classification. The 10 Bengali Alphabet's gesture that we used in our project is shown in Figure 4.3.

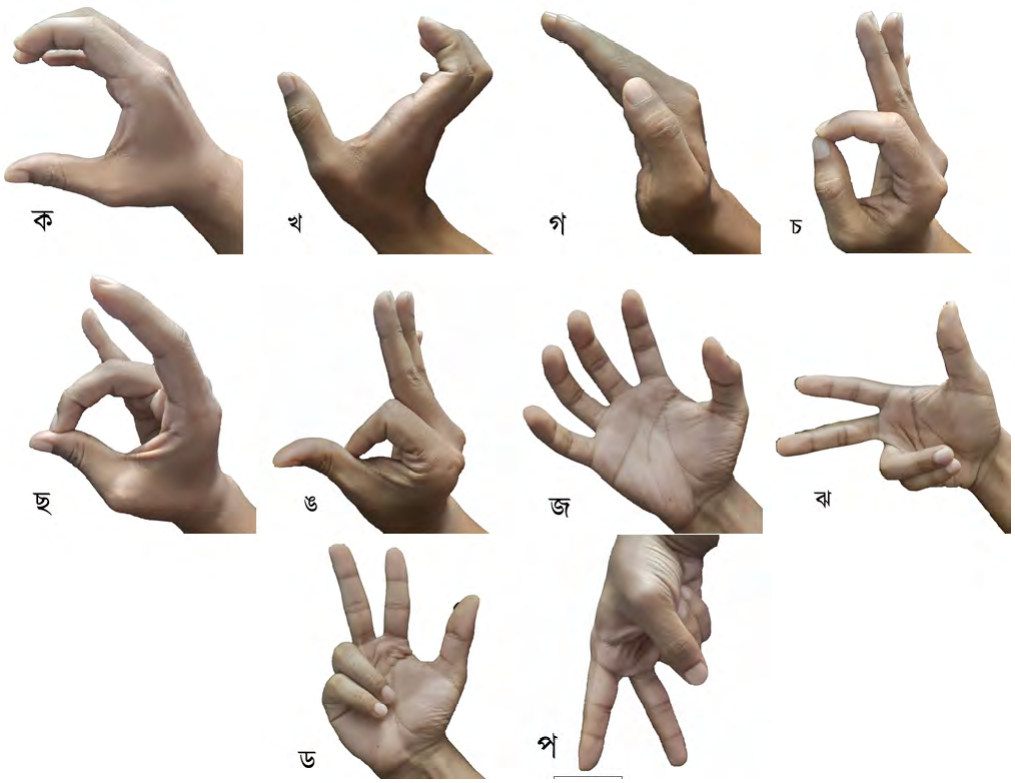

Figure 4.3: Bengali Hand Gestures from our Dataset

There will be an 8-second countdown before storing data. This delay is given for the ease of the users so that they make the proper gesture to store the gesture set. When a gesture is successfully stored, it will move on to the next alphabet and the image will update accordingly. Upon the successful storage of all letters, it will automatically head back to the main menu.
This application is adaptable to our need and users can add additional gesture sets to the system following a dictionary. But its only incompetence is Leap Motion not having the aptitude for recognizing face or body gestures as it was built for dealing with hand gestures. Other than that Leap motion is the most accurate motion sensor which makes the system more effective.

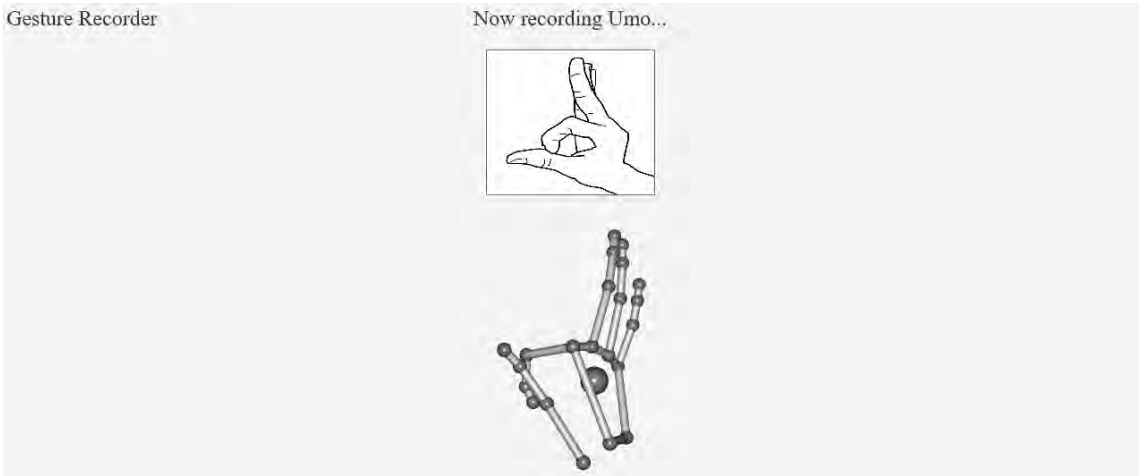

Figure 4.4: Data Collection of Umo (ঙ) Alphabet's gesture

4.4 Recognition

The screen shown in Figure 4.5 appears if the user touches the “Recognition” button. In the recognition window, the user will perform any of the given gestures from the Bengali alphabet. When a valid sign language is given, the closed match to that gesture along with the matching percentage will exhibit on the screen. As we have mentioned we used $P recognizer as a recognition algorithm. We set the accuracy parameter to 60% to be exhibit on the screen. So that any gesture having accuracy above 60% will be recognized as a valid gesture and will be shown on the screen. This process is mainly the testing process of the system.

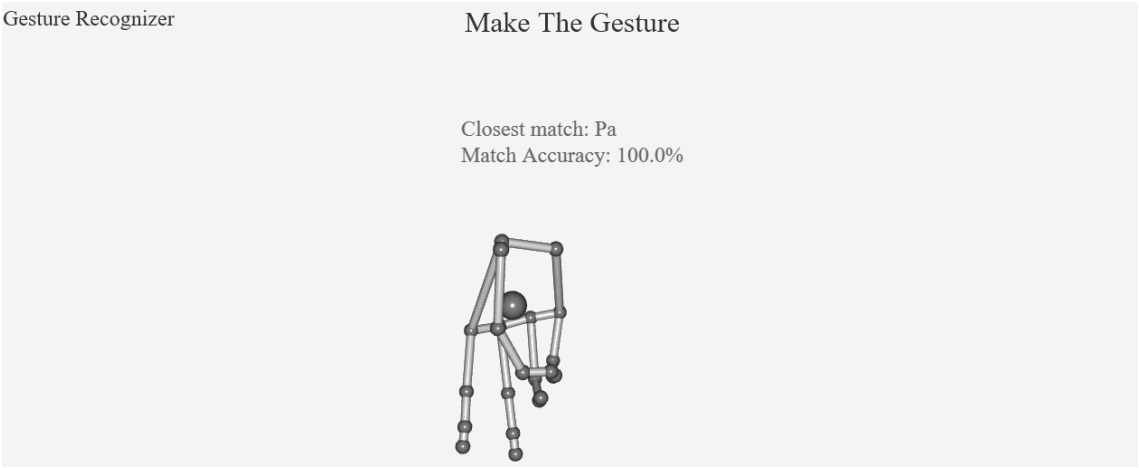

Figure 4.5: User Making a Gesture in ’Recognition’ State

In Figure 4.5 we can see that the user is making a particular gesture and on the screen, it is recognized as ”Pa (পা)” with the matching percentage of 100%.
The application was built to be set out as a template to use real-time gesture recognition system to integrate a Bengali Sign language Recognition application. The accuracy is changeable based on orientation of Hand.

Chapter 5

Experimental Analysis and Results

5.1 Test and Training Sets

We have 50 gesture sets from 20 users and each set contains 10 Bengali alphabets. As a result, in total we have 500 gestures' information for our training set. Our model is trained with these data for recognition purpose. This improves the accuracy of the recognition algorithm as there are more extensive types of objects to choose for each given classification based on the features used. We have already discussed about the data collection and recognition process in the previous Chapter 4. After collecting the data from the users, We test our system by 10 random users who gave gesture in the recognition panel through which we have checked the accuracy of our proposed model. When a user makes any gesture in front of the Leap Motion Controller, that gesture is converted into a Point Array and simultaneously checked with our collected data (which are saved in the training phase). Our model matches the shown gesture by $P Point Cloud Recognizer which mainly finds the nearest matched neighbour points by calculating euclidean distance between two points. Finally, the matched gesture is shown in the screen with the match accuracy in percentage value.

5.2 Results

Main goal of our proposed model is recognizing Bangla Sign Language accurately and within the least possible time. As we have made our model for the benefit of hearing and speech impaired people, without high accuracy rate we cannot say that our model will be helpful for them. As they are physically challenged, our model should be fast enough to recognize the gestures correctly when they use them for communication in real time. After calculating the average accuracy rate, we have found the results for our 10 alphabet classes which are listed in the Table 5.1.

Table 5.1: Test set accuracy for each alphabet class

Alphabet Class	**Average Accuracy**	**Alphabet Class**	**Average Accuracy**
Ka (ক)	100%	Chho (ছ)	100%
Kha (খ)	100%	Ja (জ)	93%
Ga (গ)	100%	Jha (ঝ)	100%
Umo (ঙ)	99%	Do (ড)	100%
Cho (চ)	99%	Pa (প)	100%

From the accuracy rate of each alphabet class, we can see that while most of the alphabet can be recognized with 99-100% accuracy, only "Ja (জ)" has an average accuracy of 93%. It happens sometimes as the sign "Ja (জ)"shown using the back side of the hand and sometimes Leap Motion Controller cannot recognize the position of the bones in back view correctly. Figure 5.1 shows the Recognition of "Ja (জ)" in our model.

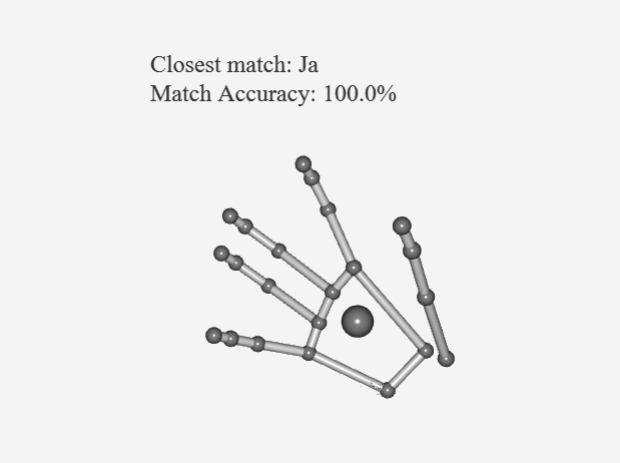

Figure 5.1: Recognition of Alphabet Ja (জ)

After calculating all average accuracy of the 10 alphabets, we have found that our model has an average accuracy rate of 99.1%. This accuracy rate is fairly high enough not only because of the accuracy value itself but also for its real time detection ability. In Literature review part from Chapter 2, we have shown the summary of various models. If we compare their results with our proposed one, we can see that our model is better in terms of accuracy and real-time detection ability. In Subsection 5.4 we have discussed comparative analysis in details.
In Figure 5.2, we have shown how accuracy rate of the model is varying with the number of used training sets. The graph shows that recognition accuracy of our proposed model is increasing linearly with the number of used training sets. If we increase the number of training sets, features domain increases which can support a wide range of values for the features in time of recognition. Then, there is less chance of making mistakes. For this reason, users get more flexibility in time of orientation of a particular gesture. So, it increases the accuracy accordingly.

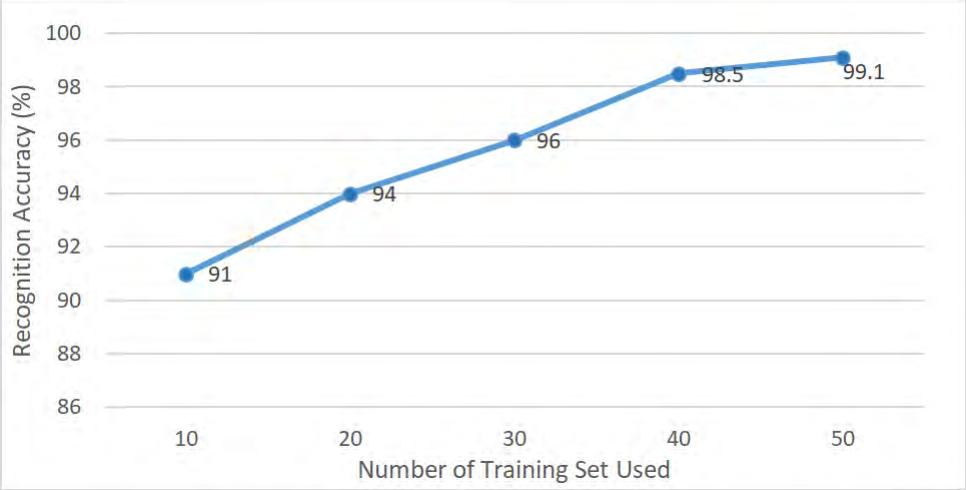

Figure 5.2: Recognition Accuracy in terms of Number of Training Sets Used

Form the graph in Figure 5.2, we can see that when we used 40 datasets, we got 98.5% accuracy. Finally, for our 50 datasets, we have got accuracy of 99.1%.
Figure 5.3 shows the relation between "Time taken to recognize an alphabet" and "Number of used datasets". Again, we can notice that when we increase the number of datasets, recognition time increases. It happens because, with the increasing number of datasets, our proposed model has to check more feature values and these comparisons take more time. However, when we use 50 datasets, our recognition accuracy becomes 99.1% with only 77.1 milliseconds recognition time. As time is a big factor in machine learning and recognition, less recognition time makes the proposed model more credible.

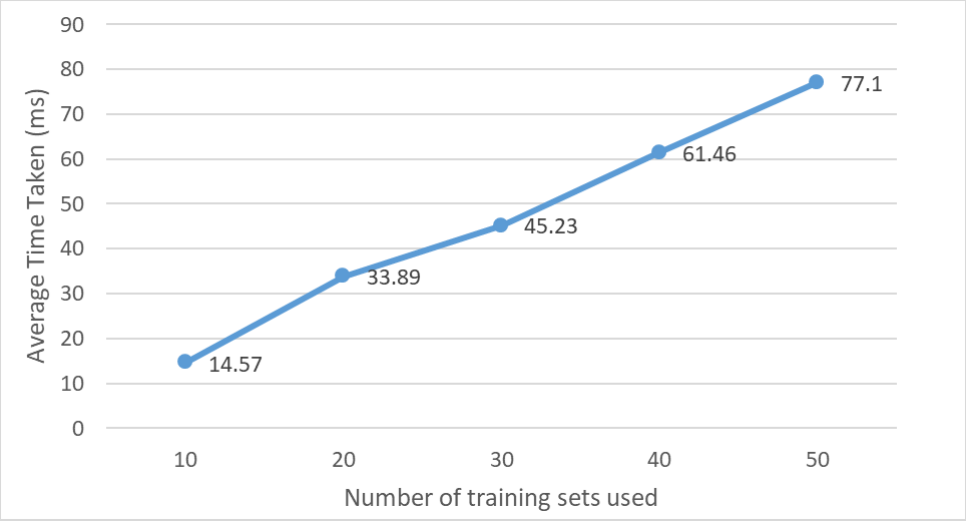

Figure 5.3: Average Time Taken for Recognition

In addition, when we increase the number of split points for the features, it has non-linear impact on the recognition accuracy which is shown in Figure 5.4. From this graph, we can notice that if we use 4 to 8 points for recognition purpose, we can get only 45-58% accuracy but 16 points provide 96% accuracy while 100% accuracy is provided by 32 points which is maximum recognition accuracy.

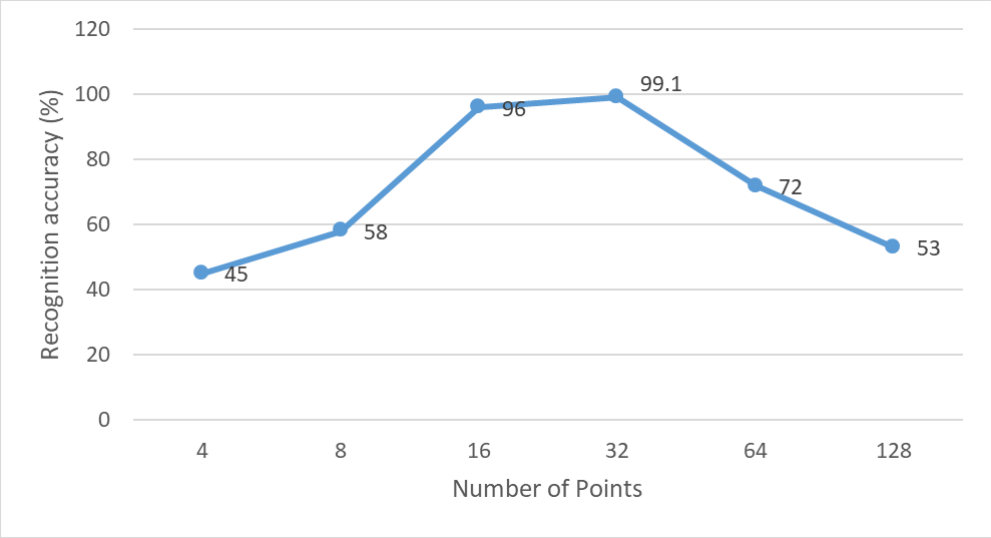

Figure 5.4: Recognition Performances with Variable Number of Points

However, if we use more than 32 points, recognition accuracy decreases. Recognition accuracy of our proposed model mainly depends on optimal number of points. Less than the optimal number of points provide inaccurate recognition. More than the optimal number of points also provides inaccurate recognition, because in that case, our algorithm has to match more split features for recognition and so, user do not get enough flexibility in case of making gestures. So, splitting a feature into more points cannot provide accurate result in our model.

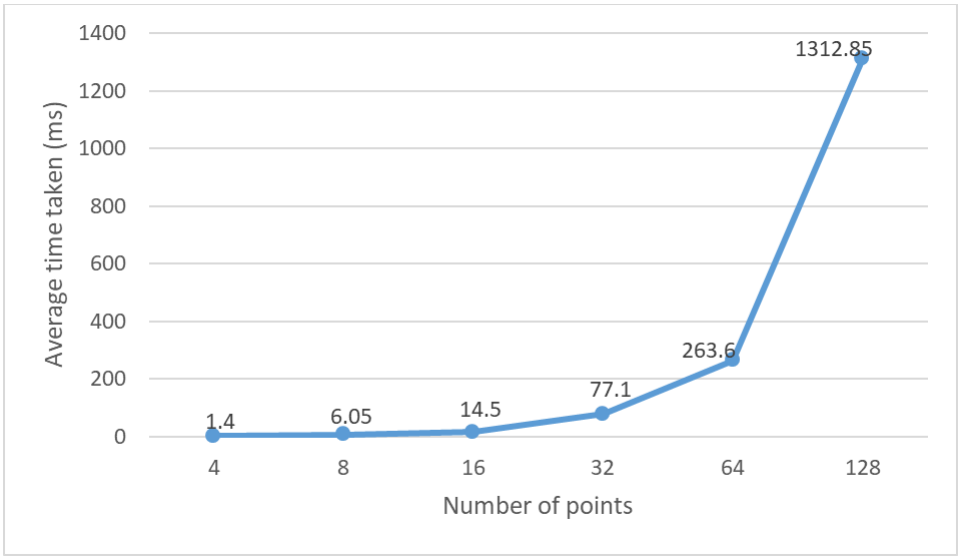

Figure 5.5: Average Time Taken with Varying Number of Points

Finally, Figure 5.5 shows that Number of split points has an exponential increasing rate with the Recognition time. It happens because, if we increase split points for a feature, it creates more data for a single feature. In that case, recognizer has to manipulate more data in time of recognition. To manipulate more data, recognizer has to take more time and thus, recognition time increases with the increasing number of split points.

5.3 Analysis Mismatch Gesture

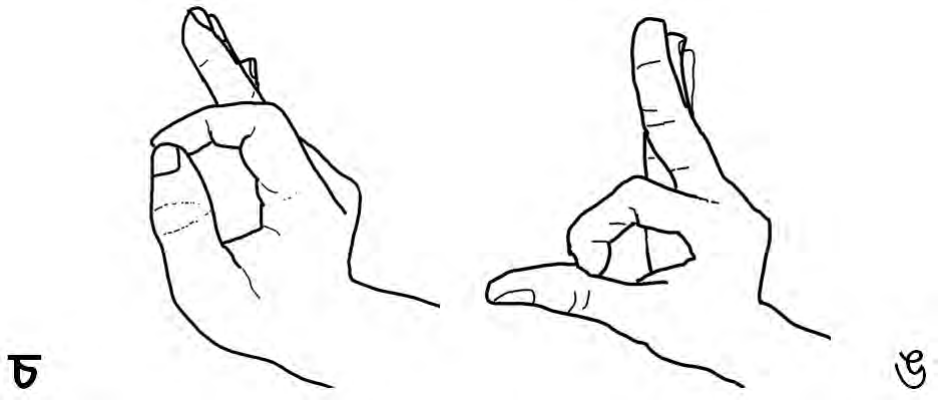

Figure 5.6: Actual gestures of Cho (চ) and Umo (ঙ)

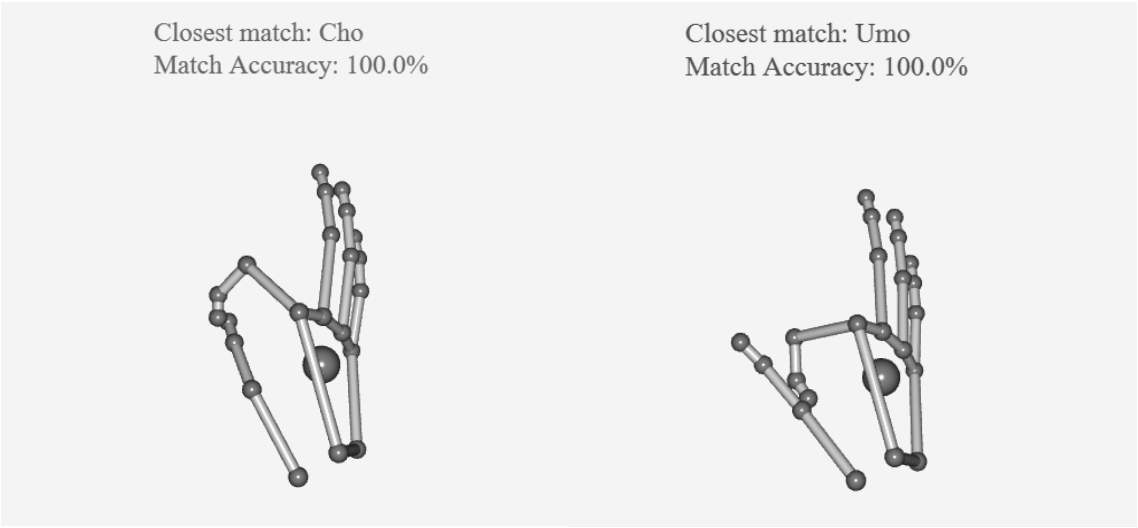

Figure 5.7: Skeletal views of Cho (চ) and Umo (ঙ)

There are some gestures in Bangla sign language which are almost identical except for a little change for example "Cho" and "Umo". In the Figure 5.6 and Figure 5.7 we can see the visual representation of both gestures. In 'Cho' both index's and thumb's tip are connected and in 'Umo' index finger's tip is connected with the intermediate bone of thumb finger. So there were few cases where Leap motion recognized 'Umo' as 'Cho' but given a proper gesture for Umo it displayed the proper output.

5.4 Comparative Analysis of Different Models

The Table 2.1 refers to some of the previous research works similar to our research area. This table includes the methodologies, types of dataset they have used as well as their average accurate rate from their proposed models. All the approaches have their advantage and disadvantages.

Comparing the techniques of all these projects, almost all of them used still images while only one of them used video footages and one of them used Leap Motion for collection of data. The number of dataset used varying from 244-2400 images of characters. Though the most common method used in these projects is Support Vector Machine, there are other methods like MLP NN, ANN, CNN, KNN and Convex hull method etc.
If we are to compare the techniques with ours, for our model we used $P Point-Cloud Recognizer whose sole purpose is to compare or recognize gesture based UI. We stored the gestures as 3D coordinates using Leap Motion Controller for real-time comparison. $P treats the gesture as groups or "clouds" of points and then the method assess each groups or clouds one by one. With this technique we have got up-to 100% accuracy rate and on an average 99.1% from our proposed model.
As we can see only two of the previous models supports real-time recognition like our model. That too has a accuracy of 72.78% & 96.15% respectively with KNN & SVM techniques and MLP neural network with Back Propagation algorithm. So, it is safe to conclude that our model is by far the most accurate and unique as it supports the real-time recognition of gestures for Bangla sign language and with the average accuracy of 99.1%.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

Sign Language Recognition system is essential for the speech and hearing impaired people for communicating with others. According to WHO the numbers of hearing and speech impaired people are more than 5% in the world so it is safe to say we have a big number of these people in our country too. These big numbers of people are neglected most of the time but our project can remove the complication of the communication gap between regular people and hearing or speech impaired people. The research we have carried out using the $P recognizer gave us a solid result that can ensure the intigrity of our model.From our model we received an average accuracy of 99.1% and the model is a real-time recognition system whereas all the existing techniques do not work on real-time, some only works on static still images of hand gestures.

6.2 Future Scope

For future work, we will try to make our system more efficient. As discussed earlier we only used 10 alphabets as we had to make our own dataset. So, in the future we intend to implement it for all the alphabets and possibly for word and sentences too.Then, we can measure the accuracy. Our model is already more efficient than most other models as it measures accuracy in real-time. We built the system for our research purpose only but it can be implemented to business purpose. Nevertheless, our foremost goal remains to help the speech and hearing impaired people.

References

[1] M. S. Islam, S. S. S. Mousumi, N. A. Jessan, A. S. A. Rabby, and S. A. Hossain, "Ishara-lipi: The first complete multipurposeopen access dataset of isolated characters for bangla sign language", in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, IEEE, 2018, pp. 1–4.

[2] B. S. L. Committee *et al.*, "Bengali sign language dictionary", *Sutrapur, Dhaka: National Centre for Special Education, Ministry of Social Welfare*, 1994.

[3] M. Alauddin and A. H. Joarder, "Deafness in bangladesh", in *Hearing Impairment*, Springer, 2004, pp. 64–69.

[4] F. N. Arko, N. Tabassum, T. R. Trisha, and F. Ahmed, "Bangla sign language interpretation using image processing", PhD thesis, BRAC University, 2017.

[5] S. Schechter, *What is gesture recognition? gesture recognition defined*, 2018. [Online]. Available: https://www.marxentlabs.com/what-is-gesture-recognition-defined/.

[6] S. Vikram, L. Li, and S. Russell, "Writing and sketching in the air, recognizing and controlling on the fly", in *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, ACM, 2013, pp. 1179–1184.

[7] S. Riofrío, D. Pozo, J. Rosero, and J. Vásquez, "Gesture recognition using dynamic time warping and kinect: A practical approach", in *2017 International Conference on Information Systems and Computer Science (INCISCOS)*, IEEE, 2017, pp. 302–308.

[8] C.-H. Chuan, E. Regina, and C. Guardino, "American sign language recognition using leap motion sensor", in *2014 13th International Conference on Machine Learning and Applications*, IEEE, 2014, pp. 541–544.

[9] F. A. Mufarroha and F. Utaminingrum, "Hand gesture recognition using adaptive network based fuzzy inference system and k-nearest neighbor", *International Journal of Technology*, vol. 8, no. 3, pp. 559–567, 2017.

[10] M. A. Uddin and S. A. Chowdhury, "Hand sign language recognition for bangla alphabet using support vector machine", in *2016 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, IEEE, 2016, pp. 1–4.

[11] R.-D. Vatavu, L. Anthony, and J. O. Wobbrock, "Gestures as point clouds: A $ p recognizer for user interface prototypes", in *Proceedings of the 14th ACM international conference on Multimodal interaction*, ACM, 2012, pp. 273–280.

[12] M. Chen, "Universal motion-based control and motion recognition", PhD thesis, Georgia Institute of Technology, 2013.

[13] M. Mohandes, S. Aliyu, and M. Deriche, "Arabic sign language recognition using the leap motion controller", in *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, IEEE, 2014, pp. 960–965.

[14] L. E. Potter, J. Araullo, and L. Carter, "The leap motion controller: A view on sign language", in *Proceedings of the 25th Australian computer-human interaction conference: augmentation, application, innovation, collaboration*, ACM, 2013, pp. 175–178.

[15] D. Naglot and M. Kulkarni, "Ann based indian sign language numerals recognition using the leap motion controller", in *2016 International Conference on Inventive Computation Technologies (ICICT)*, IEEE, vol. 2, 2016, pp. 1–6.

[16] A. R. Chowdhury, A. Biswas, S. F. Hasan, T. M. Rahman, and J. Uddin, "Bengali sign language to text conversion using artificial neural network and support vector machine", in *2017 3rd International Conference on Electrical Information and Communication Technology (EICT)*, IEEE, 2017, pp. 1–4.

[17] M. Geetha and M. Kaimal, "A 3d stroke based representation of sign language signs using key maximum curvature points and 3d chain codes", *Multimedia Tools and Applications*, vol. 77, no. 6, pp. 7097–7130, 2018.

[18] S. T. Ahmed and M. Akhand, "Bangladeshi sign language recognition using fingertip position", in *2016 International Conference on Medical Engineering, Health Informatics and Technology (MediTec)*, IEEE, 2016, pp. 1–5.

[19] D. Naglot and M. Kulkarni, "Real time sign language recognition using the leap motion controller", in *2016 International Conference on Inventive Computation Technologies (ICICT)*, IEEE, vol. 3, 2016, pp. 1–5.

[20] M. A. Rahaman, M. Jasim, M. H. Ali, and M. Hasanuzzaman, "Real-time computer vision-based bengali sign language recognition", in *2014 17th International Conference on Computer and Information Technology (ICCIT)*, IEEE, 2014, pp. 192–197.

[21] B. C. Karmokar, K. M. R. Alam, and M. K. Siddiquee, "Bangladeshi sign language recognition employing neural network ensemble", *International journal of computer applications*, vol. 58, no. 16, pp. 43–46, 2012.

[22] D. Kaushik Deb, M. I. Khan, H. P. Mony, and S. Chowdhury, "Two-handed sign language recognition for bangla character using normalized cross correlation", *Global Journal of Computer Science and Technology*, 2012.

[23] G. A. Rao, K. Syamala, P. Kishore, and A. Sastry, "Deep convolutional neural networks for sign language recognition", in *2018 Conference on Signal Processing And Communication Engineering Systems (SPACES)*, IEEE, 2018, pp. 194–197.

[24] M. ElBadawy, A. Elons, H. A. Shedeed, and M. Tolba, "Arabic sign language recognition with 3d convolutional neural networks", in *2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)*, IEEE, 2017, pp. 66–71.

[25] R. McCartney, J. Yuan, and H.-P. Bischof, "Gesture recognition with the leap motion controller", 2015.

[26] F. R. Khan, H. F. Ong, and N. Bahar, "A sign language to text converter using leap motion", *International Journal on Advanced Science, Engineering and Information Technology*, vol. 6, no. 6, pp. 1089–1095, 2016.

[27] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the accuracy and robustness of the leap motion controller", *Sensors*, vol. 13, no. 5, pp. 6380–6393, 2013.

[28] C. Chen, L. Chen, X. Zhou, and W. Yan, "Controlling a robot using leap motion", in *2017 2nd International Conference on Robotics and Automation Engineering (ICRAE)*, IEEE, 2017, pp. 48–51.

[29] *How does the leap motion controller work?*, Jan. 2017. [Online]. Available: http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/.

[30] *Leap motion*. [Online]. Available: https://www.leapmotion.com/.

[31] Y. Li, "Protractor: A fast and accurate gesture recognizer", in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2010, pp. 2169–2172.

[32] J. O. Wobbrock, A. D. Wilson, and Y. Li, "Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes", in *Proceedings of the 20th annual ACM symposium on User interface software and technology*, ACM, 2007, pp. 159–168.

[33] L. Anthony and J. O. Wobbrock, "A lightweight multistroke recognizer for user interface prototypes", in *Proceedings of Graphics Interface 2010*, Canadian Information Processing Society, 2010, pp. 245–252.

[34] ——, "$N-protractor: A fast and accurate multistroke recognizer", in *Proceedings of Graphics Interface 2012*, Canadian Information Processing Society, 2012, pp. 117–120.

[35] W. Lu, Z. Tong, and J. Chu, "Dynamic hand gesture recognition with leap motion controller", *IEEE Signal Processing Letters*, vol. 23, no. 9, pp. 1188–1192, 2016.