Charades

Final Year Project

Session 2016-2020

A project submitted in partial fulfilment of the
COMSATS University Degree
of
BS in Computer Science (CUI)

Department of Computer Science

COMSATS University Islamabad, Lahore Campus

30 July 2020

Project Detail

Type (Nature of project)	[ ] **D**evelopment [ ] **R**esearch [*] **R&D**			
Area of specialization	Computer vision and AI			
Project Group Members				
Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	FA16-BCS-127	Ramsha Rasheed	ramsha4rasheed@gmail.com	
(ii)	FA16-BCS-097	Aisha	aishaqureshi39@gmail.com	
(iii)	FA16-BCS-121	Bilal Ahmed	bilal11.official@gmail.com	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I am __Ramsha Rasheed__ D/o __Muhammad Rasheed__, group leader of FYP under registration no CIIT/FA16-BCS-127_/LHR at Computer Science Department, COMSATS Institute of Information Technology, Lahore. I declare that my FYP proposal is checked by my supervisor and the similarity index is _14__% that is less than 20%, an acceptable limit by HEC. Report is attached herewith as Appendix A.

Date: __28-July-2020___ Name of Group Leader: __Ramsha Rasheed___ Signature:

Name of Supervisor: Dr. Usama Ijaz Bajwa
Designation: Associate head of department
Signature: ______________________

Co-Supervisor (if any): ______________________
Designation: ______________________
Signature: ______________________

HoD: ______________________
Signature: ______________________

Abstract

As stated by World Health Organization(WHO), according to an estimation more than 466 million people around the world are experiencing hearing impairment that can be disabling [1]. These include people that are totally deaf and those who are hard of hearing i.e. having little or no functional hearing. Deaf people perform sign language to communicate while the hard of hearing can use spoken language with aid or else sign language. Sign languages is used as first language by these people. Deaf people always require a sign language translator to converse on daily basis. Hence, they are dependent on an interpreter. Similarly, abled people have hardly any knowledge of sign language too. Having a sign language translator available all the time is difficult and expensive. To aid deaf people we propose an application. This mobile application is easy to use and economical to have. The functionalities that the application can perform are two way. Firstly, it translates the sign language that a deaf person performs to text. Secondly, it converts the text to sign language for deaf people. This will give them a chance to live life to its fullest.

Acknowledgement

This project could not have been possible without the support and guidance provided by COMSATS university Islamabad, Lahore campus and constant supervision of Dr. Usama Ijaz Bajwa. I am grateful to my team members, colleagues and every other person who helped and made this project possible. I extend my gratitude to all of them.

Table of C0ntents

List of Tables

List of Figures

Chapter 1

Introduction

Chapter 1

1 Introduction

1.1 Introduction:

As stated by World Health Organization(WHO), according to an estimation more than 466 million people around the world are experiencing hearing impairment that can be disabling [1]. In those 466 million people some are completely deaf that basically have little or no functional sense of hearing at all and the other are called “hard of hearing” who have mild-to-moderate hearing loss. Deaf people perform sign language to communicate while the hard of hearing can either communicate using sign language or spoken language with assistance. Deaf communicate in sign language as their first language. Abled people perhaps have hardly any knowledge of sign language at all. Deaf individuals all over the globe are facing issues due to this communication gap in daily life. Due to slight disability, many people are being deprived of experiencing their life to its fullest.

According to World health organization 34 million deaf people out of those 466 million deaf are children [1]. Most of the time it’s the hearing parents that bear deaf children. Among those parents, there are many who have no knowledge of sign language and they might not even learn. So deaf children get deprived of the basic education that a child gets in his early years at home. These children can’t go to normal schools as they can’t hear or speak. So, as a result they can’t read or write. Special education is required for these people that can be expensive. Most people belonging to poor families might not be able to afford.

With no education or treatment at early ages these deaf people remain uneducated. These people are deprived of a good living standard and are discriminated against in hiring and recruitment. Because recruiters find it hard for them to accommodate a deaf person.

One solution is that a sign language interpreter may be used. But a sign language interpreter is expensive to have accessible all the time, and it becomes inconvenient too. As there are 466 million people who would require a sign language interpreter to fully function in the society on daily basis. A professional sign language interpreter in United States (US) on average costs $24.00 per hour [2]. The expense of having a sign language interpreter is around $50,000 per annum in the US which is expensive for a median waged person in the US.

A solution to this problem is needed, that is economical and easy to use. As, this disability has isolated the deaf community around the globe. It's their right to enjoy every aspect of life like other abled people.

Our project proposes a solution to this problem that is convenient to have and use too. The solution wished-for is an application that works both ways. First module acts as a listener i.e. it detects the signs performed by the deaf person using Smart phone/tablets camera or a device such as Kinect and translate them to native spoken language in form of text on the Smart Phone's screen. Second module acts as a speaker i.e. it takes the text in natively spoken language typed by the abled person through the keyboard and converts it into a series of signs which are implemented on the screen of the Smart phone by a 2D stick figure.

Objectives:

The project includes development of software that acts as a voice for deaf community who face many challenges in communication daily. The objectives of the application are:

- Using Smartphone Camera to detect the signs performed by the deaf person.
- Translating those signs into text in the English language of the abled person on the Smartphone.
- The abled person typing an answer to the former query on the phone's keyboard.
- The texted answer is then translated into sign language.
- The series of sign gestures, motions are displayed on the screen.

Keeping in mind that every human being in all normal cases has the same hand shapes consisting of 4 fingers and a thumb. This project aims at creating a real-time system that recognizes the meaningful shapes made by using hands.

1.2 Problem Statement:

To design an application that aids deaf individual's communication with abled individuals. The application intended will be two-tiered. Firstly, it detects signs and gets them translated to its text form using trained CNN. Secondly, it translates the text to sign using first an NMT to convert text to gloss and then a lookup table is used to convert gloss to sign. This application will be implemented on android operating system and models trained in python.

1.3 Assumptions and constraints:

For this project, it is assumed that the major part of sign language communication is performed by the gestures performed by hands and arms. We are not paying much attention to other details like facial expression and overall body language.

There are 135 different sign languages around the planet. Sign languages are the least developed languages of the world with no specific international standards. For now, we are working on American sign language (ASL) as it is assumed to be the most popular among deaf community. Among the 135 sign languages ASL is most widely used.

As sign languages are least developed languages worldwide. People are still working on ASL development but on a small scale. We are using videos of ASL for recognition part. There are not many datasets available which puts a constraint on the model training in first module.

Sign language in its entirety is a distinct language that has its own well-defined grammar. ASL's grammar and sentence structure are very different from English language's grammar. So, for second module we require a dataset for English to Gloss conversion. Due to lack of development, we are facing difficulties in finding a good dataset.

1.4 Project scope:

As a human being we always wanted to work for the betterment of the society. Having the knowledge of computer vision and interest in artificial intelligence, we were provided with an opportunity to develop something. So, we proposed this system that will certainly add value to these people's lives by bridging the communication gap for deaf community.

Our targeted audience is 466 million deaf and hard of hearing people globally that makes it 6.1% of the world population. The application will be introduced to students at special schools and others too so that they can get basic education like other normal children. The application will be used by adults for inclusion in all aspects of life too.

Furthermore, this product with integration of other sign languages can be used globally. People can use this application with the variation of sign language according to the region. If in future an international sign language is developed, then the application can integrate that and be general for all users.

Chapter 2

Requirement Analysis

Chapter 2

2 Requirement Analysis

2.1 Practical Work

This section is further partitioned into two sub-parts. Work already done related to this project's application based on trained neural network and other based on applications. Each sub section briefly describes the working of their model/product.

2.1.1 Related Work based on Trained Models

Technology has always been the most effective and fastest medium for communication in recent years. Plenty of work is being performed for the prosperity of sign language and interpreters are being developed using various approaches. The work being done is for one-way communication and a few of the systems are implemented and being used by public as we speak. A few working systems, prototypes and techniques used to make those prototypes are discussed below with critical analysis.

1. **Alphabet recognition of American sign language using Convolutional Neural Networks** [3]**:**

   This paper illustrates how Microsoft Kinect is used to capture images that are used to recognized American Sign Language (ASL) alphabets. A Convolutional Neural Network (CNN) integrated with Multiview augmentation plus inference fusion is being used.

   Firstly the 3D data of an image is retrieved by Multiview augmentation which then generates more data for different perspective views, which results in more efficient training, and odds of potential overfitting are reduced.

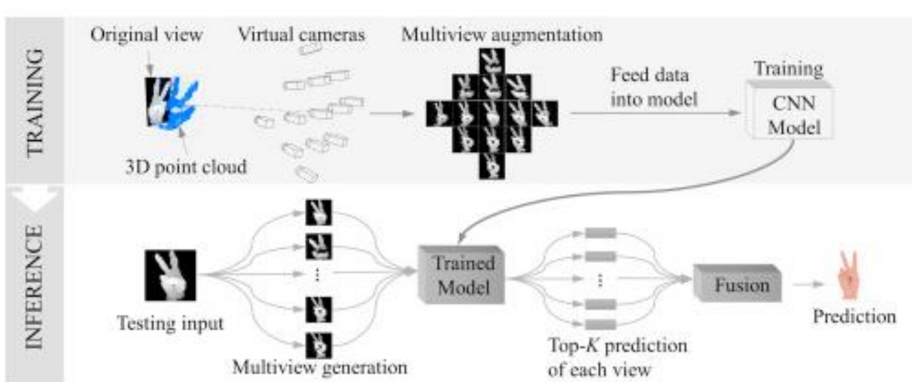

Figure 1: Multiview augmentation strategy and Inference fusion

The above Figure 1 illustrates the training and inference fusion of the CNN model.

2. **Sign Language Fingerspelling translations through Image Processing** [4]**:**

   This research paper discusses a way of translating Fingerspelling of sign language to equivalent text. This method is carried out using Image Processing. The video is captured with internal or external webcam. Image frame from the video are extracted. This extracted image is then processed in order to extract gestures. The images are first smoothened by applying mean filter and later the image is converted to grayscale. Iterative thresholding is used for converting the given grayscale image to binary format. The largest BLOB from this binary image is obtained. The Images are resized using bilinear interpolation algorithm. The images stored in the database undergo the same process. The input processed images we get are compared with the images in the database and the most matched image with its corresponding alphabet is retrieved. Figure 2 shown below, describes the procedure of processing the input images.

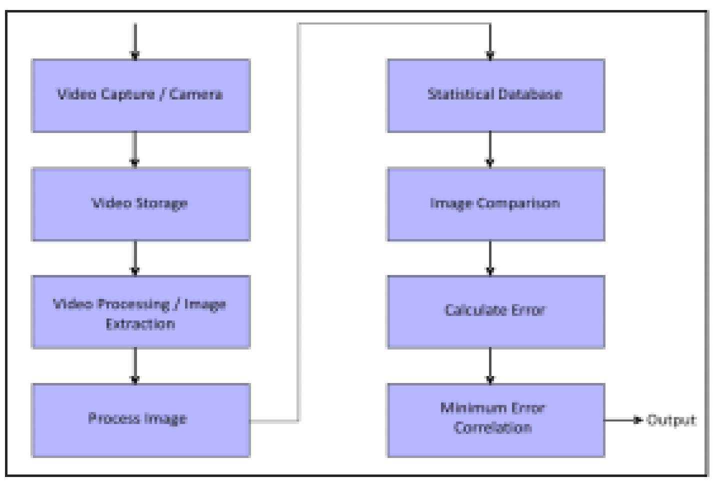

Figure 2: Overview of processing of input image.

Aforge.Net library is used for implementing the image processing task [5].

3. **Sign Language Recognition using Neural Networks** [6]**:**

   This paper describes the means by which the process of sign language recognition can be automated by the application of CNN and ANN. Two major steps were followed in this process, at first feature extraction was done followed by classification of the actions. CNN was used for extraction and then classification was done using an Artificial Neural Network (ANN). Videos were recorded using Microsoft Kinect. The dataset acquired is used for development purposes and comprises a total of 6600 images. Training set had approximately 4400 out of those 6600 images and the rest 2200 were used for validation. The captured image dataset was pre-processed to crop so that only higher part of the hand and torso are left and focused. Using Max pooling, features were extracted and classified respectively. The model consisted of 2 CNN's and one ANN module. The dataset used in this paper was gathered from ChaLearn dataset 2014.

4. **Static hand gesture recognition using Neural networks** [7]**:**

   This article supports hand movement recognition based on the analysis of their shape. Neural Network based approach was used for classification of different stationary hand gestures. They have also used a unique multi-layer perception of neural network for classification using back-propagation algorithm. They achieved 86.38% accuracy on their trained model.

5. **Spanish text to Spanish sign language translation system** [8]**:**

   This Paper explains how the Spanish text to Spanish sign language system was developed. This system was developed to help deaf people at the time of driver's license renewal. The system was deployed in Toledo at Local Traffic Office. The application is made of three parts: first a speech recognizer is used to get the speech, a natural language translator to translate speech to its corresponding signs and then for the animation of signs a 3D avatar module like eSIGN [9] and VISICAST [10]. For natural language translation part three technological approaches were applied and evaluated. Figure 3 shows the architecture used for the system.

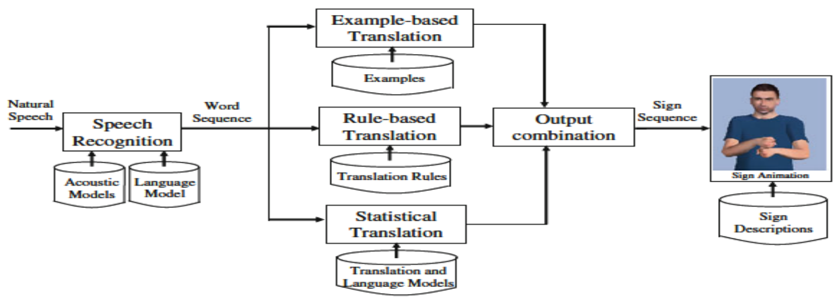

Figure 3: Spanish text to sign translation

The system is restricted in vocabulary. It only caters the sentences used in conversation for license renewal. The methods used for natural machine translation are not that efficient and intelligent. The 3D animation modules are efficient but require text conversion to first Hamburg Notation system (HamNoSys) and then to sign language mark-up language (SIGML) which makes the use of it cumbersome.

6. **English speech to American sign language translator** [11]**:**

This paper converts English speech to American sign language. The system proposed translated English speech to ASL gloss using "Attention is all you need" [12] mechanism. The data set used for training this is ASLG-PC12 [13]. Then the animation is done using video generator that puppet interprets the language.
The system lacks a good animator. The animators used are not available free of cost to public.

7. **German Text to German sign language using natural language processing methods and Generative Adversarial Networks** [14]**:**

The paper explained a well-developed pipeline for text to sign language conversion. The pipeline proposed by paper includes three parts. The text goes through a neural machine translation model that converts it into sign language gloss. The gloss is used to acquire the corresponding 2D skeletal pose from a lookup table. The 2D skeletons were extracted using Open Pose and stored in lookup table. The poses are then used to generate a real-life human animator using generative adversarial network (GAN) as shown in Figure 4. The dataset used is in German PHOENIX14T. The pipeline proposed is as follow:

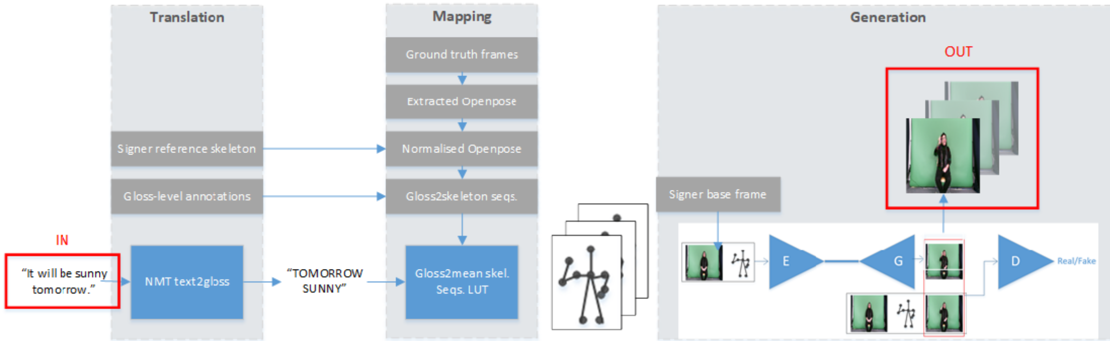

Figure 4:Sign language translation using GAN

First, the paper proposes German to German sign language translation. GAN is a new and heavy and requires more computational power to operate. So, with limited mobile resources GAN is difficult to deploy on mobile phones and does not work properly.

2.1.2 Related work based on applications:

1. **Augmented reality app that converts sign to speech and vice versa:**

New York University students created a working prototype using machine learning and augmented reality. The application translated American sign language into spoken English words and vice versa. The app has a narrow scope of just making an appointment at a clinic presently. This application is limited in the number of signs it can detect and translate [15].

This project was just a pilot project and never was available to public. The sign language used for this project is dialect specific and is American sign language which makes the project region specific too. And as the project uses very limited vocabulary of sign language. Thus, project domain is restricted too.

2. **Sign language to English language conversion through a wearable device:**

The researchers working at Texas A&M University manufactured a wearable gadget that interprets American sign language to English language. This device uses sensors to detect hand movement of the user. The device recognizes 40 ASL signs. There are two sensors which are used to detect the hand movement and are worn by user on their right wrist. The data received from sensors is sent to a laptop through Bluetooth. On the laptop an algorithm runs that interprets the sign sensed and displays the corresponding English word [16].

This device just gives a one-way translation from sign language to text but does not provide a way for the other half of the communication. This also requires hardware that a person cannot wear all the time. If one of the sensors is not working the whole device will provide false interpretation. The hardware will be expensive to replace.

3. KinTrans:

In KinTrans a camera is used to trace the movement of a person's hands and body as they are signing the words. A deaf individual will make contact with a bank teller by signing to the KinTrans camera. This gadget can then translate those signs into English or Arabic language corresponding to the words for the teller to read. The translation works both ways. A machine learning algorithm is used by KinTrans that converts for each sign that is made, and for conversion into grammatically correct sentence another algorithm is used. The lack of data on sign language has made the training of data difficult [17] [18].

The device is limited to English and Arabic users and as sign language like spoken language has many dialects, so its use is really limited too.

4. SignAll:

SignAll's system comprises four cameras. One of the cameras records in 3D and capture data from a signer's face as well as their hands and body. The SignAll technology translates ASL into written English, and then displays it as a chat dialogue. This system comprises of two monitors: one is used by the Deaf person, and other one is used by the hearing person. The deaf person must wear a colourful glove. The coloured gloves assist the equipment to distinguish among the fingers. The hearing person uses voice that is picked by a speech recognition system [19].

It requires a booth installation for the setup of this technology. The deaf user must use clear ASL, avoiding regional signs. The users must avoid colourful clothing, excessive clothing, oversized jewellery (bracelets) etc.

5. **Live sign language to caption translation system:**

   The system has three components: it has a pocket-sized microphone that the user can clip onto his/her clothing, a smartphone-sized Raspberry Pi/Adafruit-powered microcomputer that user can keep in his/her pocket, and a Google Glass-like for display. Mic is attuned so that it can pick up human dialogue, even in environments with considerable background noise. The human dialogues received by microphone are processed by the computer, which are then translated to text and then the text is wirelessly transmitted to be displayed. The display that is clipped to an existing pair of third-party glasses. The user sees text on that display, that is placed in front of their view of the speaker. Allegedly the lag between the words being spoken and being displayed is small. [20]

   A lot of gadgets are required for this to work. Their working for a layman to understand can be difficult and it can be hard to carry everywhere. It would not work if any of the gadgets become defective. Replacing gadgets can be expensive. This system works one way i.e. it's a way that enables a deaf person to hear. The deaf person requires to have intermediate level of written English knowledge.

2.2 Stakeholders:

- Sign language researchers.
- Data Analyst.
- Project team.
- Project supervisor.
- Professional sign language interpreters.

2.3 Requirement Elicitation:

Requirements were gathered using qualitative research methods.

2.3.1 Functi0nal requirements (FR):

The systems functional Requirements denotes the requirements that the system is supposed to fulfil. The tasks and functions that must be performed by the system are defined in it. Functional requirements help you to capture the proposed behaviour of the system.

Functional Requirement-01: Login/Dashboard

Table 1:Functional Requirements of Login/Dashboard

Req. No.	**Functional Requirements**
FR-01-01	User should be allowed to login or sign-up by the application.
FR-01-02	The application shall maintain separate account for each user with user's usage history.
FR-01-03	The application intends to allow user to choose between the options displayed on the dashboard.

In the above Table 1, the functional requirements of login page/dashboard are mentioned.

Functional Requirement-02: Sign-to-Text

Table 2: Functional Requirements of Sign-to-Text Interface.

Req. No.	**Functional Requirements**
FR-02-01	The application shall accurately acquire each sign performed by the signer.
FR-02-02	The application shall precisely predict the correct conversion to text.
FR-02-03	The application shall display text in correct grammatical sequence or structure of the language being used.

Table 2 describes the tasks the application must perform in order to operate properly.

Functional Requirement-03: Text-to-Sign

Table 3: Functional Requirements of Text-to-Sign Interface.

Req. No.	**Functional Requirements**
FR-03-01	The application shall read text entered by the user.
FR-03-02	The application shall generate corresponding Sign Language gloss.
FR-03-03	The application shall be able to obtain matching Sign pose of the gloss.
FR-03-04	The application shall be capable of displaying the sequence of poses to the users in the form of animation.

Table 3 describes the functional requirements of Text-to-Sign interface.

2.3.2 N0n- Functional requirements (NFR):

These requirements entail how the system should perform certain function. It allows to impose limitations or constraints on the design of the system.

Non-Functional Requirement-01: Performance

Table 04: NFR based on Performance

Req. No.	**Non-Functional requirements**
01	Application should be competent for operating on any Android device with minimum 8GB of RAM and Android version 7.0 or above.
02	The start-up of application must not be more than 10 seconds.

03	The application must be able to classify the user's provided information corresponding to the performance of the video and text classifier in no more than 1 minute.

Table 4 describes the NFR that the application needs to have based on performance.

Non-Functional Requirement -02: Reliability

Table 05: NFR based on Reliability

Req. No.	**Non-Functional Requirement**
01	The application must be workable on all types of supported hardware.
02	The application must always acquire the Signing accurately in Sign-to-Text module.
03	The application must always animate accurately in Text-to-Sign module.

Table 5 describes the NFR of the application which is based on reliability of the application.

Non-Functional Requirement -03: Robustness

Table 06: NFR based on Robustness

Req. No.	**Non-Functional Requirement**
01	The application's mean time to crash should not be more than 10 minutes within 1 month.

Table 6 describes the NFR of the application which is based on robustness.

Non-Functional Requirement -04: Security

Table 7: NFR based on Security

Req. No.	Non-Functional Requirement
01	Only the Logged in/Signed up User shall be able to gain access to functionalities.

Table 7 describes the NFR of the application which is based on security.

Non-Functional Requirement -05: Extensibility

Table 8: NFR based on Extensibility

Req. No.	Non-Functional Requirement
01	The application must be 50% extensible to support new additional features and future developments.

Table 8 describes the NFR of the application which is based on extensibility of the application.

2.4 Use case descriptions:

2.4.1 Use case descriptor: Sign-Up

Table 9: Use case ID:1

Use Case ID:	1
Use Case Name:	Sign-Up
Actors:	Deaf user, normal user
Description:	The user wants to register or signs up to use the application.
Pre-Condition:	User ought to be on the sign-up page.
Post-Condition:	The user will be registered/signed up successfully.
Normal Flow of Events:	1. User will open the sign-up page. 2. User will fill the form. 3. User will be signed up after properly filling the form.
Alternatives Flow:	1. User opens the sign-up page. 2. User do not fill the password and confirm password fields properly. 3. Error will be displayed.
Exceptions:	None

Table 9 shows the Use Case ID:1, which is Sign-Up. This is the first main module every user needs to view the complete application. Users cannot view the dashboard until or unless they register their account. For this purpose, they have to fill the form very carefully.

2.4.2 Use case descriptor: Login

Table 10:Use case ID:2

Use Case ID:	2
Use Case Name:	Login
Actors:	Deaf user, normal user
Description:	Access to the dashboard and then respective functions are being requested by user.
Pre-Condition:	Login credentials should be known.
Post-Condition:	Access to the dashboard is provided to the user.
Normal Flow of Events:	1. The user opens the application. 2. The user adds his/her login ID and password. 3. The user receives the access to the application.
Alternatives Flow:	1. The user enters the incorrect login information. 2. Access to the application is denied.
Exceptions:	None

Table 10 shows the Use Case ID:2. In this module, the user enters the login credentials. The software compares the information provided by user with the one stored at the signup in the database. The user is granted access if the information matches.

2.4.3 Use case descriptor: Sign-to-Text module

Table 11:Use case ID: 3

Use Case ID:	3
Use Case Name:	Sign-to-Text module
Actors:	Deaf user, normal user, trained model
Description:	The user using phones camera gets the signs detected. The detected signs are then using model identified. The corresponding text is displayed on the screen.
Pre-Condition:	Deaf user should be able to perform correct signs.
Post-Condition:	Normal user can see the correct translation of signs to text.
Normal Flow of Events:	1. The normal user using camera captures the signs. 2. The normal user presses the convert button for conversion. 3. The normal user can successfully read the translation on the screen.
Alternatives Flow:	The user is unable to acquire the signs performed correctly.
Exceptions:	None

Table 11 shows the Use Case ID:3. In this module, the user using phones camera gets the signs detected. The detected signs are then identified using model. On the screen, the predicted text is displayed.

2.4.4 Use case descriptor: CNN training module

Table 12:Use case ID:4

Use Case ID:	4
Use Case Name:	CNN training module
Actors:	CNN, Mobile application
Description:	The model is training from the spatial features of the provided dataset. The model is used to make predictions for each frame to obtain a series of pool layer outputs for each video.
Pre-Condition:	Dataset should be available for the training.
Post-Condition:	The mobile application will have a trained model for classification of spatial features.
Normal Flow of Events:	1. Data from dataset is used to go through all layers of training of CNN. 2. After successful training, the model is passed on for prediction.
Alternatives Flow:	1. The dataset is not accurate. 2. The model trained is not accurate.
Exceptions:	None

Table 12 shows the Use Case ID:4. In this module, the model is training from the features of the provided dataset. At the end the trained model is used to make predictions for each frame to obtain a sequence of pool layer outputs for each video.

2.4.5 Use case descriptor: Text-to-Sign module

Table 13: Use case ID: 5

Use Case ID:	5
Use Case Name:	Text-to-Sign module
Actors:	Primary: Normal user, deaf person. Secondary: NMT model, Lookup table
Description:	Normal user types the text for conversion. The text acquired is converted to the sign language gloss using a neural machine translation model. The converted gloss is then used to retrieve the corresponding 2D skeleton pose. The poses are used to generate the series of sign which is then animated on the screen.
Pre-Condition:	The normal person should be able to type correctly i.e. no spelling mistakes.
Post-Condition:	Deaf person will be able to observe the converted signs.
Normal Flow of Events:	1. The normal user types the text to be converted. 2. The deaf user can then observe the corresponding signs being displayed on the screen.
Alternatives Flow:	None
Exceptions:	None

Table 13 shows the Use Case ID:5. In this module, normal user types the text for conversion. The text acquired is converted to the sign language gloss using a neural machine translation model. The converted gloss is then used to retrieve the corresponding 2D skeleton pose. The poses are used to generate the series of sign which is then animated on the screen.

2.4.6 Use case descriptor: Neural machine translation model

Table 14:Use case ID:6

Use Case ID:	6
Use Case Name:	Neural Machine translation (NMT) model
Actors:	Database, mobile application.
Description:	The neural machine translation model using a dataset with English and its corresponding ASL Gloss is trained. During training the English sentence is tokenized and encoding vectors are formed. These encoding vectors are used by decoder for decoding to gloss. Then when the model is trained its deployed on mobile application.
Pre-Condition:	The dataset should be relevant and accurate with at least 1000 instances.
Post-Condition:	The accuracy of model should be reasonably good.
Normal Flow of Events:	1. The dataset is provided for training. 2. The model gets trained after tokenizing, encoding and decoding. 3. The model is deployed on mobile successfully.
Alternatives Flow:	None
Exceptions:	None

Table 14 shows the Use Case ID:6. In this section, the neural machine translation model using a dataset with English and its corresponding ASL Gloss is trained. During training the English sentence is tokenized and encoding vectors are formed. These encoding vectors are used by decoder for decoding to gloss. Then when the model is trained it is deployed on mobile application.

2.5 Use case designs:

2.5.1 Use case: Sign-Up

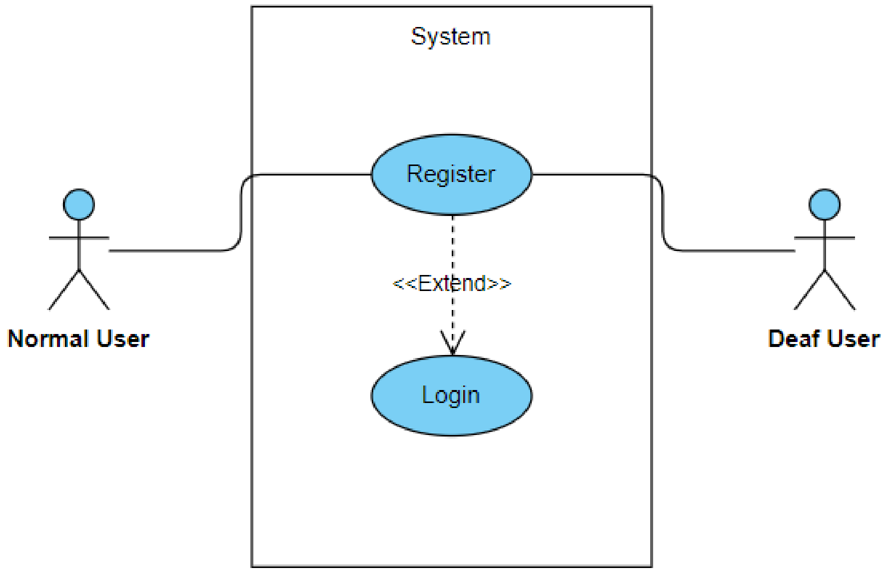

Figure 5: Use Cases: Sign-Up

Figure 5 shows the description of use case Sign-Up. The actors in this use case are Normal user and Deaf user. To login to the application, users must first register themselves at sign up page.

2.5.2 Use case: Log in

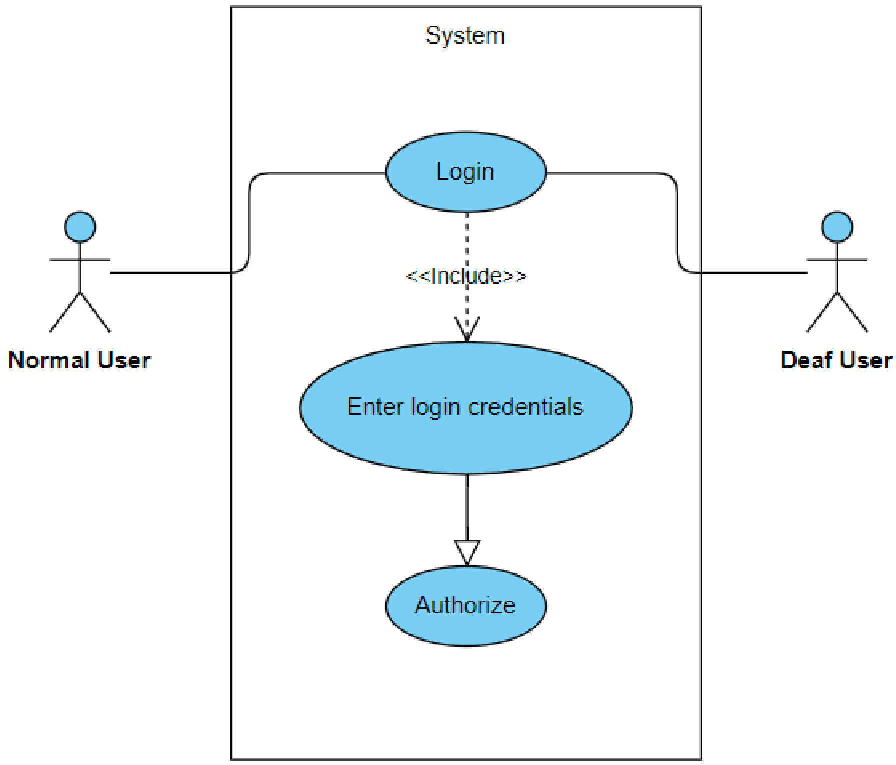

Figure 6: Use Cases: Login

Figure 6 shows the description of use case Login. The actor in this use case are Normal user and Deaf user. They provide the right login credentials to login into the application or the access is denied.

2.5.3 Use case: Sign-to-Text module

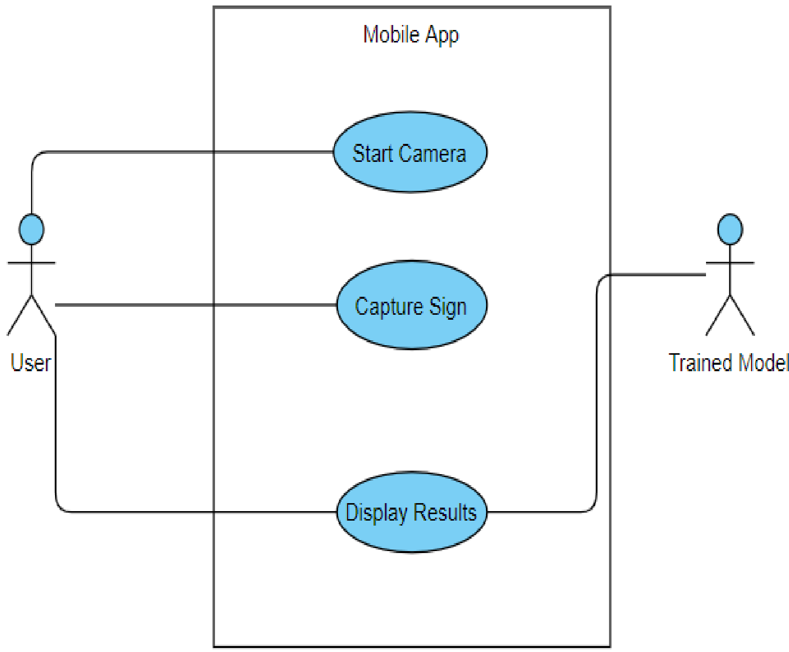

Figure 7: Use Cases: Sign-to-Text

Figure 7 is a use case for Sign-to-Text. The primary actor is the "user" while the secondary actor is the "trained model". The User may perform two actions: start camera and capture sign. Whereas the trained model displays the results back to the user.

2.5.4 Use case: CNN training

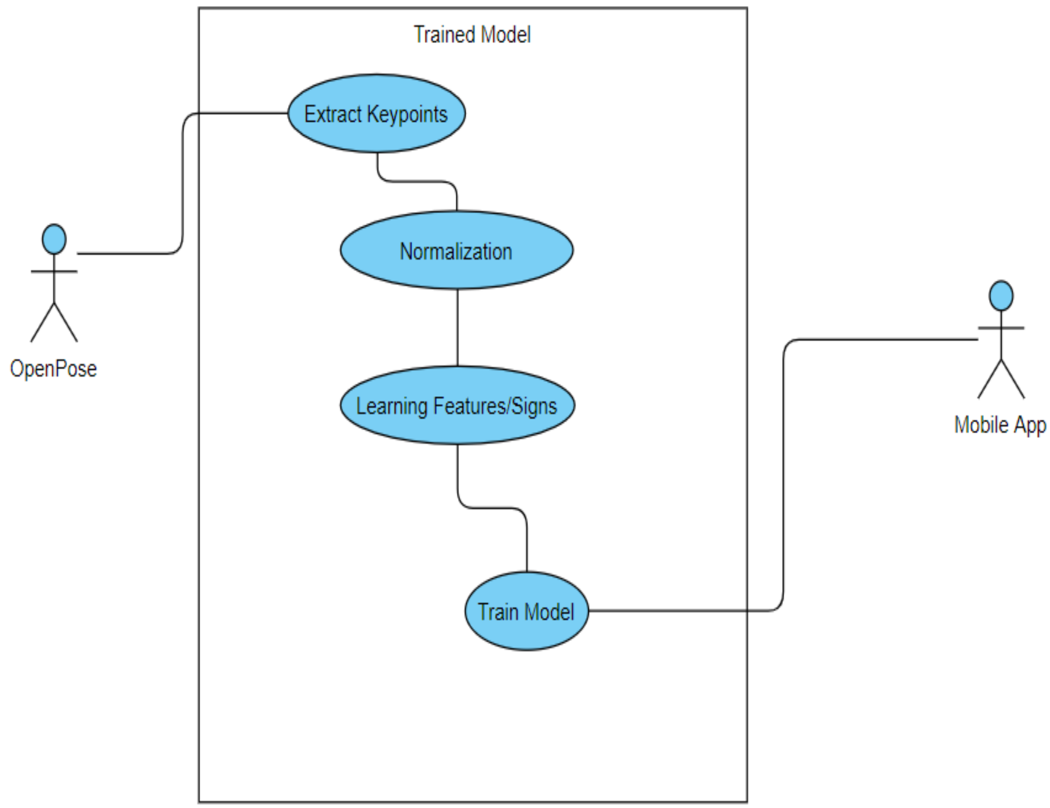

Figure 8:Use Cases: CNN training

Figure 8 displays a use case for Model training. The primary actor is the mobile app which trains the model and the secondary actor is the OpenPose. This OpenPose learns gestures, extracts keypoints, normalizes the data and recognizes the features for output.

2.5.5 Use case: Text-to-Sign module

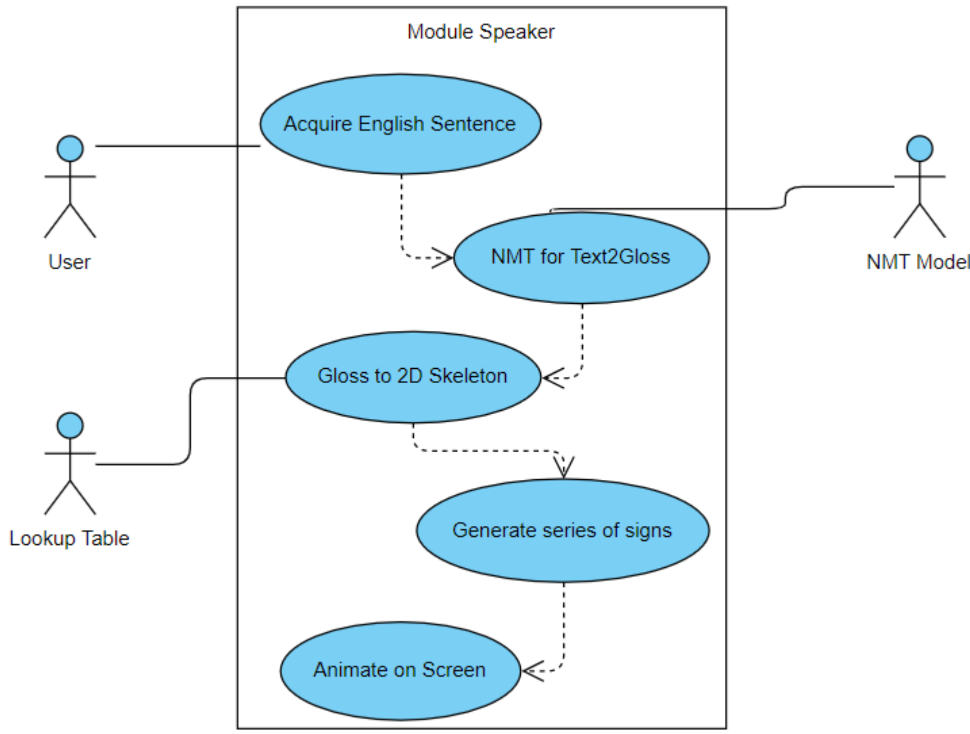

Figure 9:Use Cases: Text-to-Sign

Figure 9 shows the use case for Text-to-Sign. In this use case, the primary actor: "User" interacts with "Acquire English sentence" function. This function further uses "NMT for Text2Gloss" function. The secondary actor: "NMT Model" interacts with NMT for Text2Gloss. The other secondary actor: "Lookup table" interacts with "Gloss to 2D Skeleton" function. Gloss to 2D Skeleton is connected to "Generate Series of Signs" which is further connected to "Animate on screen".

2.5.6 Use case: NMT training module

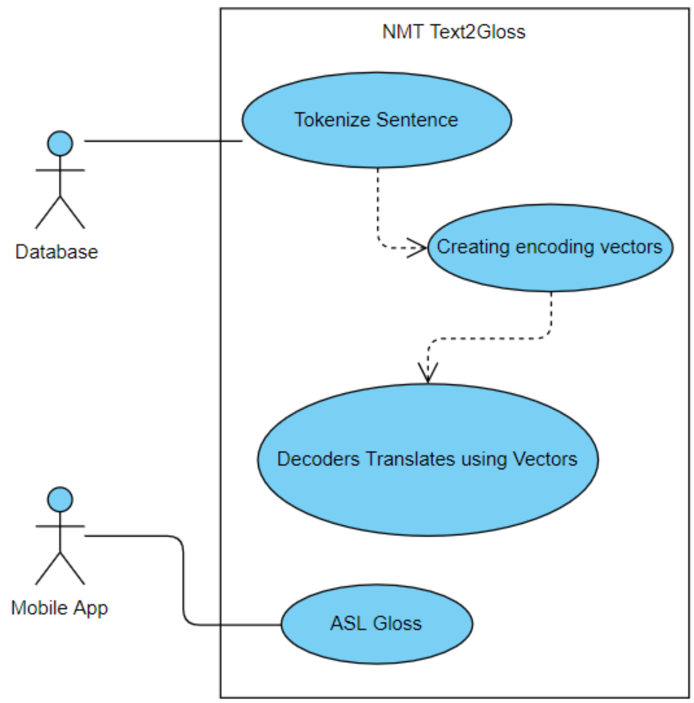

Figure 10: Use Cases: NMT training

Figure 10 is the use case for NMT training. There are two primary actors: Data base and Mobile application. Database is connected to “Tokenize Sentence” which is further attached to “Creating encoding vectors”. This is also joined with “Decoders translates using vectors”. The Mobile app is connected to “ASL Gloss”.

2.6 Software development life cycle:

Software process model also known as software development life cycle. This procedure has stages that a software goes through while its development. The basic stages are initiation, planning, designing, building, coding, testing and deployment. A model fundamentally includes a specific set of tasks with a workflow and definite milestones and outcome of each task. The result is the required product.

Incremental process model is iterative in nature. It divides the requirements into many distinct segments of the software development cycle. Every iteration passes through the requirement gathering, designing, coding and testing phase. At the end of each iteration a partial system is built, and every following release of the system has some added functions to the earliest release until a desired final product is produced.

We are using incremental process model because of our proposed project's nature. Our project is mainly divided into two modules. Both modules require revisions to them until we receive our final project as this project works on sign language. As in the incremental process model, the plan is just made for the next increment; so, the modification would be easier to implement as per need of the users.

Chapter 3

System Design

Chapter 3

3 System Design

3.1 Work breakdown system:

Figure 11:Work breakdown System

Figure 11 describes the overall approach of completing the application.

3.2 Sequence Diagram:

3.2.1 Sequence diagram: Login

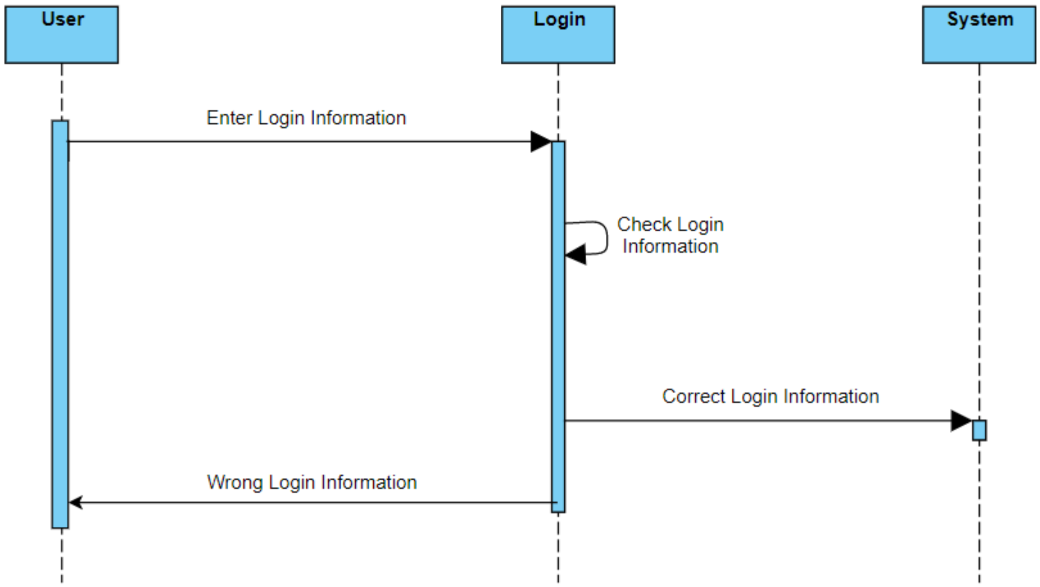

Figure 12:Login sequence diagram

Figure 12 shows the sequence diagram of Login module. In this module, user must enter the correct login credentials which he has chosen at the time of registration, in order to get access of the application or dashboard.

3.2.2 Sequence diagram: Sign-Up

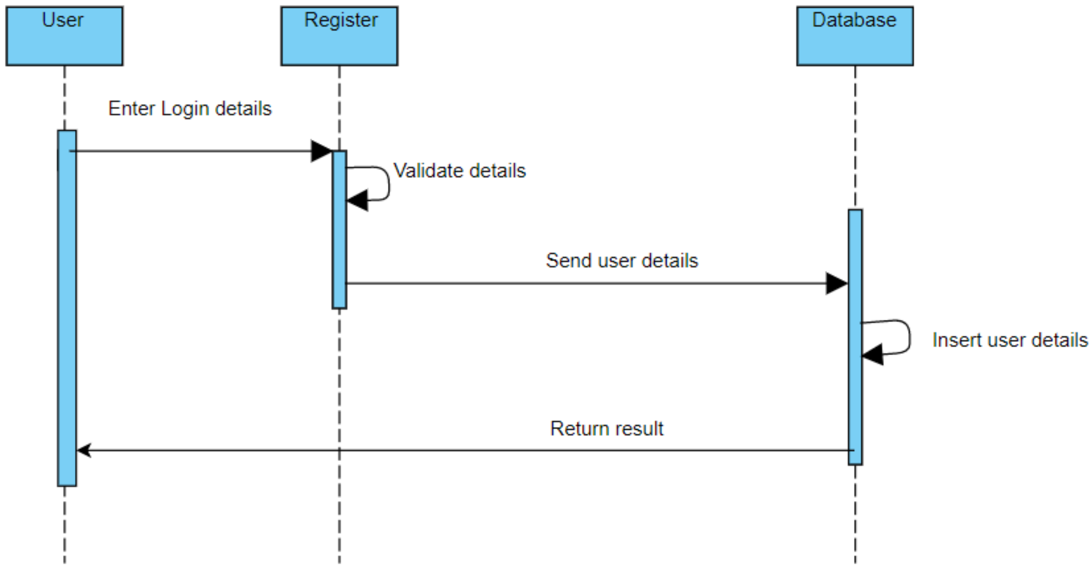

Figure 13:Sign-Up sequence diagram

Figure 13 shows the Sign-Up module's sequence diagram. In this module, user must register himself/herself in order to use the application. There is always some validation process after which credentials of the user are stored in the database.

3.2.3 Sequence diagram: Sign-to-Text

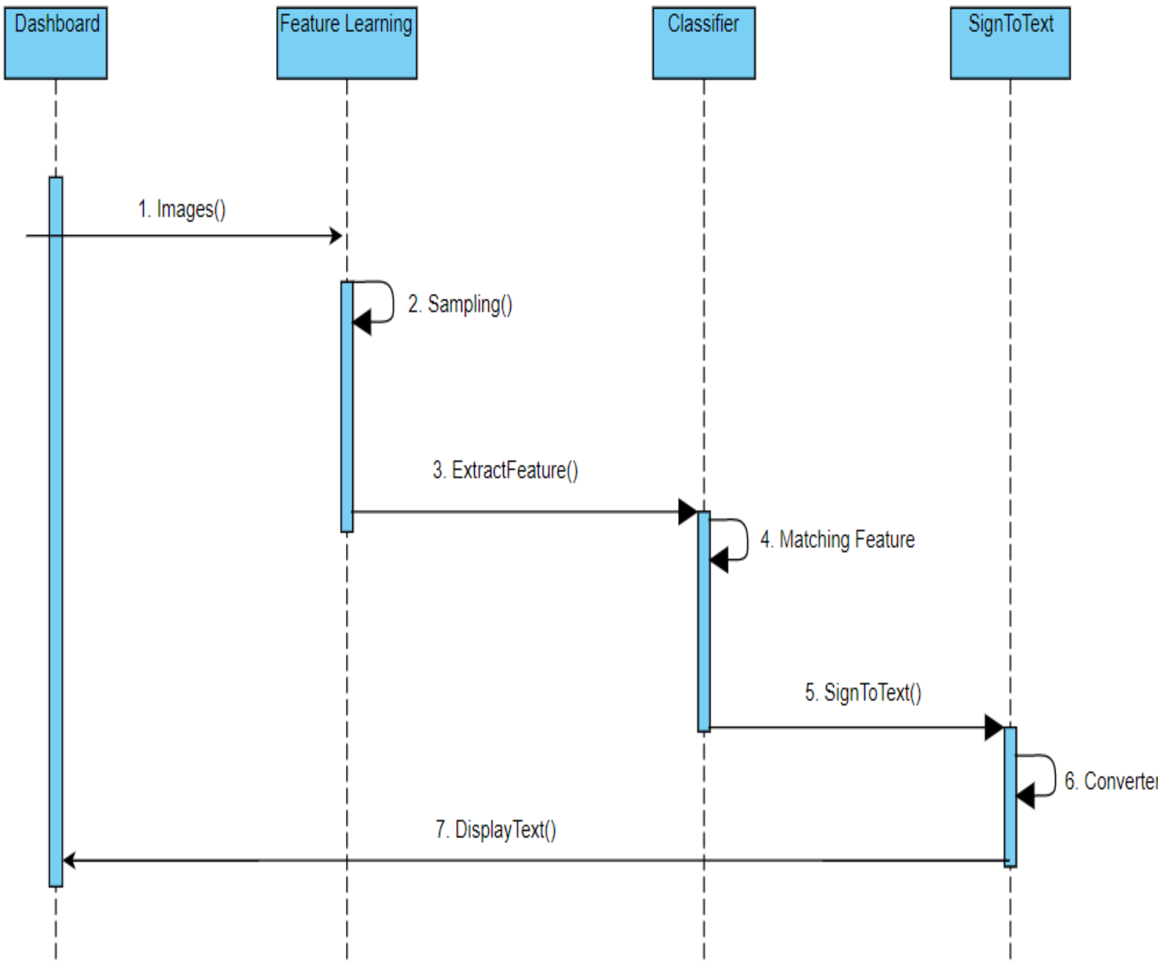

Figure 14: Sign-to-Text sequence diagram

Figure 14 displays the sequence diagram of Sign-to-Text module. The image is fed to the CNN model where the process of sampling is performed. It extracts features after sampling. The extracted features are matched with the trained signs. The matched signs are translated to text which is displayed on the dashboard for the user.

3.2.4 Sequence diagram: Text-to-sign

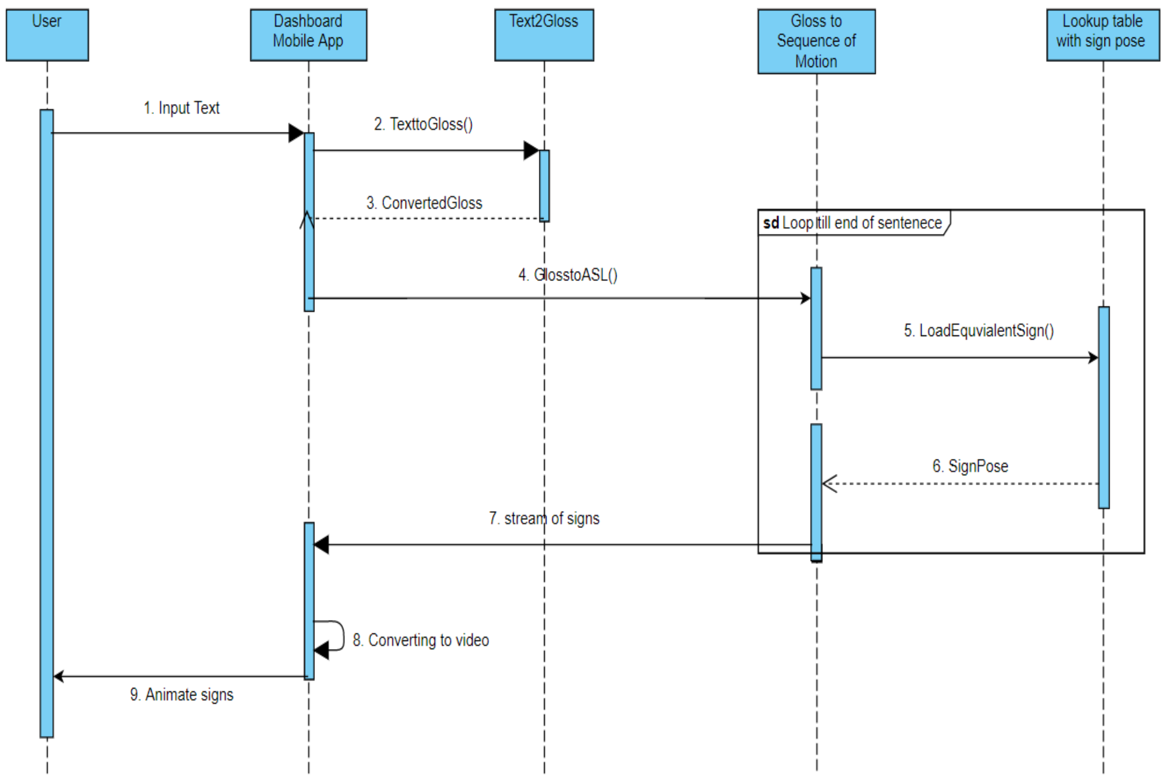

Figure 15: Text-to-Sign sequence diagram

Figure 15 shows Text-to-Sign module's sequence diagram. The user enters the text on the screen, which is converted into gloss. The converted gloss is then compared with its equivalent sign. The sign is generated through corresponding 2D skeleton pose. The poses are used to generate the series of sign which is then animated on the screen.

3.3 Software Architecture:

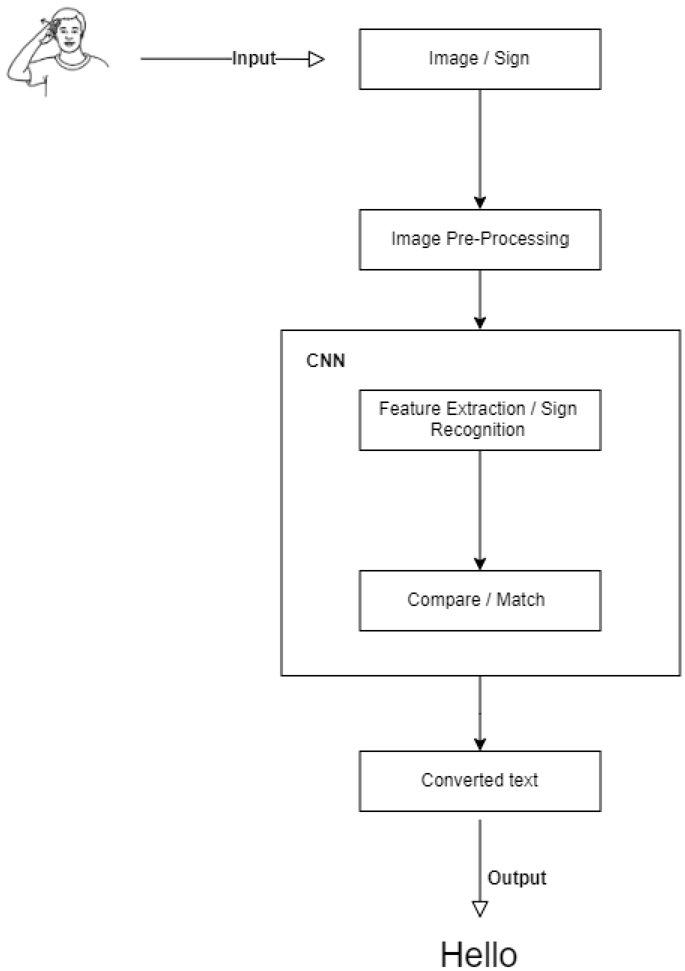

Figure 16: Sign-to-text system architecture

Figure 16 shows the basic architecture of Sign-to-Text. The image is captured and pre-processed. Later the pre-processed image in fed the model, in which the signs are extracted and compared with the trained dataset. The identified image is converted into text and displayed to the user.

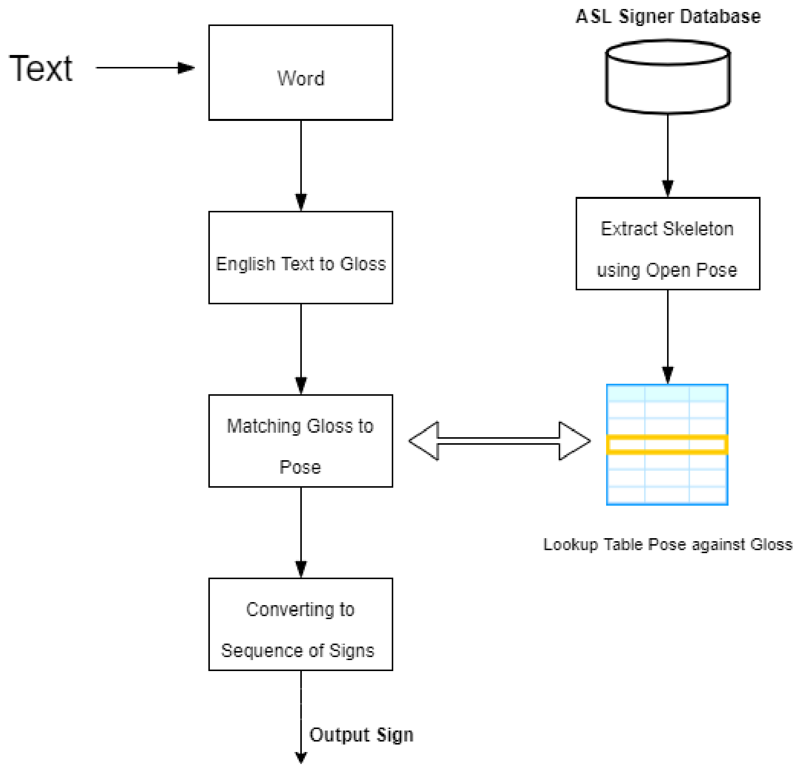

Figure 17: Text-to-Sign system architecture

Figure 17 explains the architecture of Text-to-Sign module. The text is inputted by the user which is converted into gloss. The converted gloss is then used to retrieve the corresponding 2D skeleton pose from the lookup table. The lookup table extracts skeleton using open pose from the database. The poses are used to generate the series of sign which is then animated on the screen.

3.4 Class Diagram:

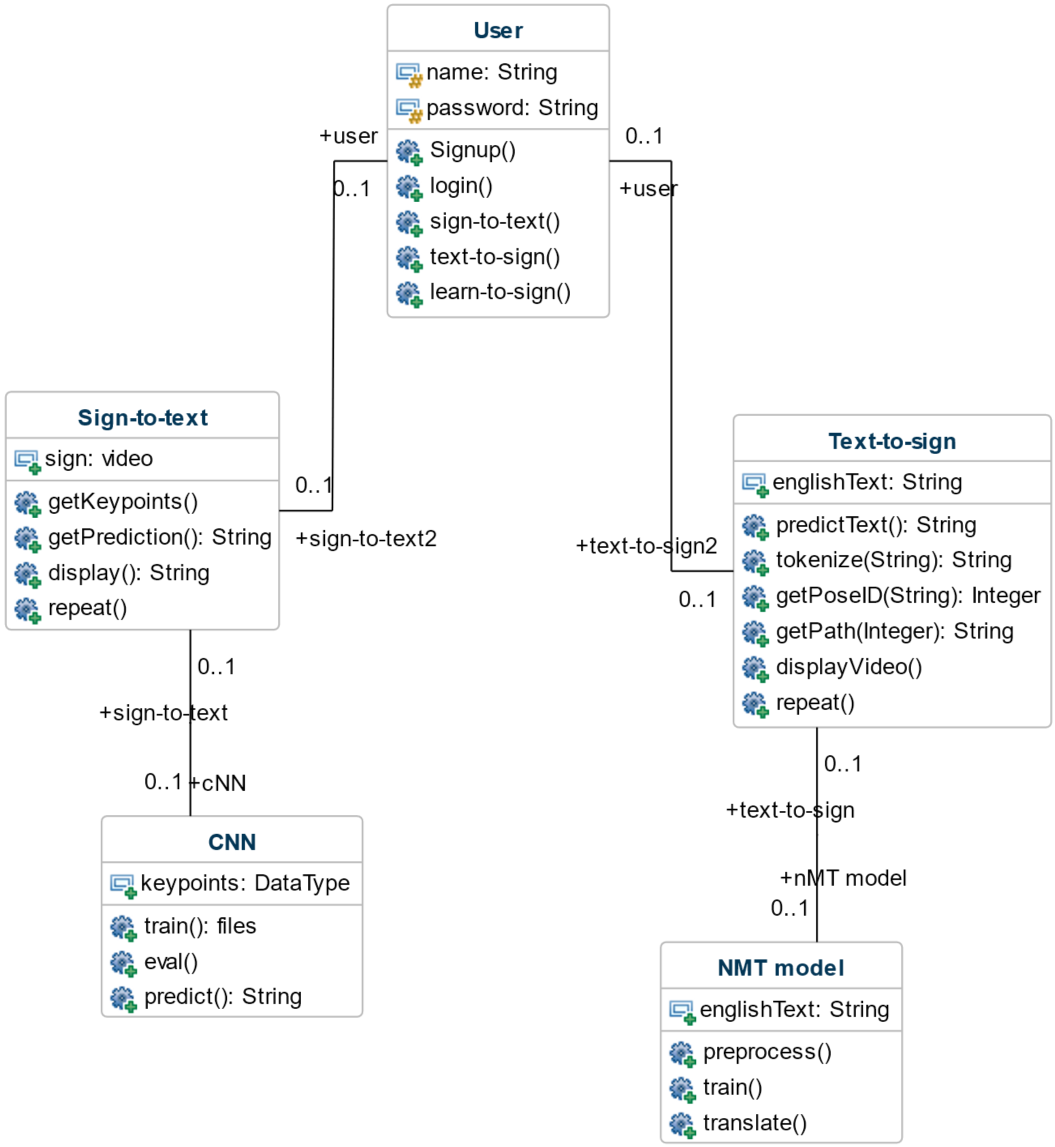

Figure 18: Class diagram

Figure 18 shows the project's class diagram and main functions and classes used to code the project's functionalities.

3.5 Network diagram (Gantt Chart):

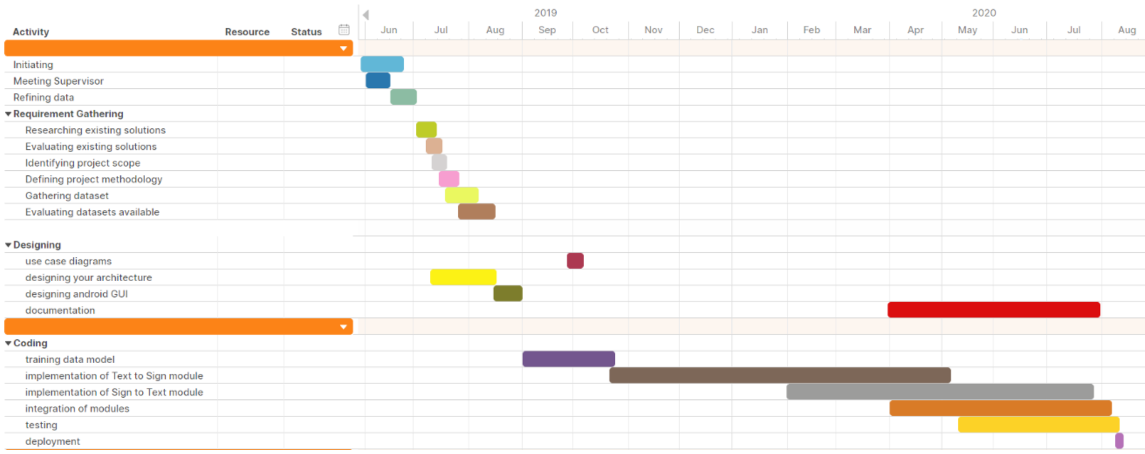

Figure 19: Gantt chart

Figure 19 shows the network diagram that exemplifies overall schedule of the project. It describes the plan according to the whole year and divides the tasks in the months.

Chapter 4

System Testing

Chapter 4

4 System testing

4.1.1 Login:

Table 15: Test case ID: 1

Test case ID 1: Tc-01	
Application Name:	Charades
Use Case(s):	Login
Input Summary:	User will be asked to provide their login details.
Output Summary: **If success:** The user will be provided the access. **Else:** Displayed error message.	
Pre- C0nditions:	The application should be turned on.
Post-conditions:	The user will be navigated to the dashboard screen.

Table 15 explains how the user can login in the application using correct login credentials.

4.1.2 Sign up:

Table 16:Test case ID: 2

<table>
<tr><td colspan="2">Test case ID 1: Tc-02</td></tr>
<tr><td>Application Name:</td><td>Charades</td></tr>
<tr><td>Use Case(s):</td><td>Sign up</td></tr>
<tr><td>Input Summary:</td><td>User should open sign up screen.</td></tr>
<tr><td colspan="2">0utput Summary:

If success:

New profile will be created and saved in the database.

Else:

Error message will be displayed.</td></tr>
<tr><td>Pre- conditions:</td><td>The application should be turned on.</td></tr>
<tr><td>Post-conditions:</td><td>New profile will be created.</td></tr>
</table>

Table 16 explains how the new user can register himself to use the application.

4.1.3 Sign-to-text module:

Table 17: Test case ID:3

<table>
<tr><td colspan="2">Test case ID 1: Tc-03</td></tr>
<tr><td>Application Name:</td><td>Charades</td></tr>
<tr><td>Use Case(s):</td><td>Sign-to-text module</td></tr>
<tr><td>Input Summary:</td><td>The user will open this module and capture a video of a person performing signs. then press the translate button for translation.</td></tr>
<tr><td colspan="2">0utput Summary:

If success:

Translations are displayed to the user.

Else:

Error message will be displayed.</td></tr>
<tr><td>Pre- C0nditions:</td><td>Deaf user should be able to perform correct signs.</td></tr>
<tr><td>Post-conditions:</td><td>Normal user can see the correct translation of signs to text.</td></tr>
</table>

Table 17 explains how the user can use the sign to text part of the application for the translation from capturing the signs performed by the signer the user is in communication with.

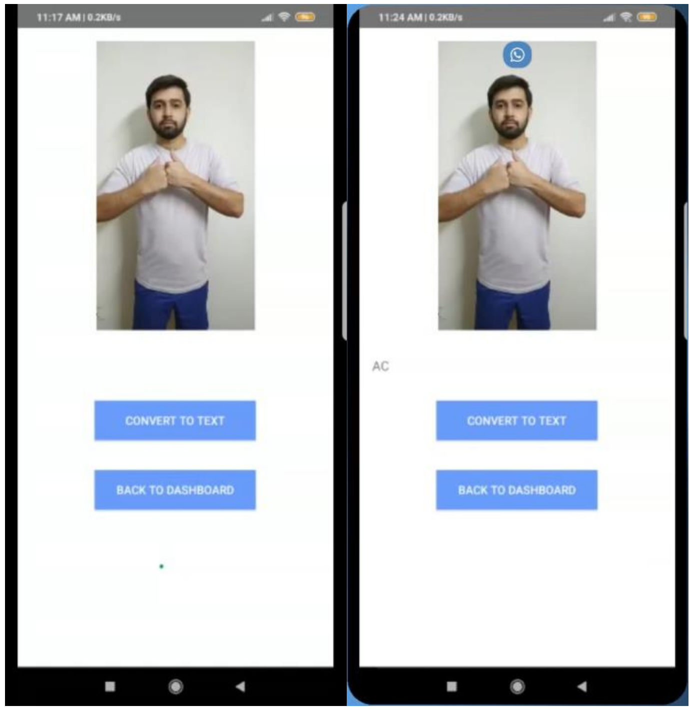

Figure 20: working of Sign-to-text module

Figure 20 shows the working of the sign-to-text module where the signers is performing a sign and application is translating it.

4.1.4 CNN training module:

Table 18: Test Case ID: 4

Test case ID 1: Tc-04	
Application Name:	Charades
Use Case(s):	CNN training module
Input Summary:	The video dataset will go for training the model for recognition.
Output Summary: **If success:** The translations to English sentence in output will be accurate**.** **Else:** The translations generated will be wrong.	
Pre- conditions:	Accurately labelled dataset is available.
Post-conditions:	The mobile application will have a trained model for classification.

Table 18 explains how the CNN model is trained on the video dataset of the American sign language for sign language recognition.

```
[07.21.20|20:30:39] Evaluation Start:
[07.21.20|20:30:56]     mean_loss: 5.3066218480831235
[07.21.20|20:30:56]     Top1: 18.67%
[07.21.20|20:30:56]     Top5: 39.03%
[07.21.20|20:30:56] Done.
```

Figure 21: Evaluation results of CNN training

Figure 21 shows the evaluation score of our CNN model on test dataset. The top evaluation accuracy is 18.67% and mean loss is 5.3066. The top-5 is the accuracy given against the most likely responses provided by the model.

4.1.5 Text-to-sign module:

Table 19: Test Case ID:5

Test case ID 1: Tc-05	
Application Name:	Charades
Use Case(s):	Text-to-sign module
Input Summary:	The text by the user that needs translation in English.
Output Summary: **If Success:** The correct sequence of videos to be displayed performing corresponding signs. **Else:** Error message for words not found or wrong sequence of signs.	
Pre- conditions:	The normal person should be able to type correctly i.e. no spelling mistakes.
Post-conditions:	Deaf person will be able to observe the converted signs.

Table 19 describes the other part of the application that is text to sign part. The text was given to the application part responsible to translate text to signs. A text is given to translate and in result a series of signs is generated.

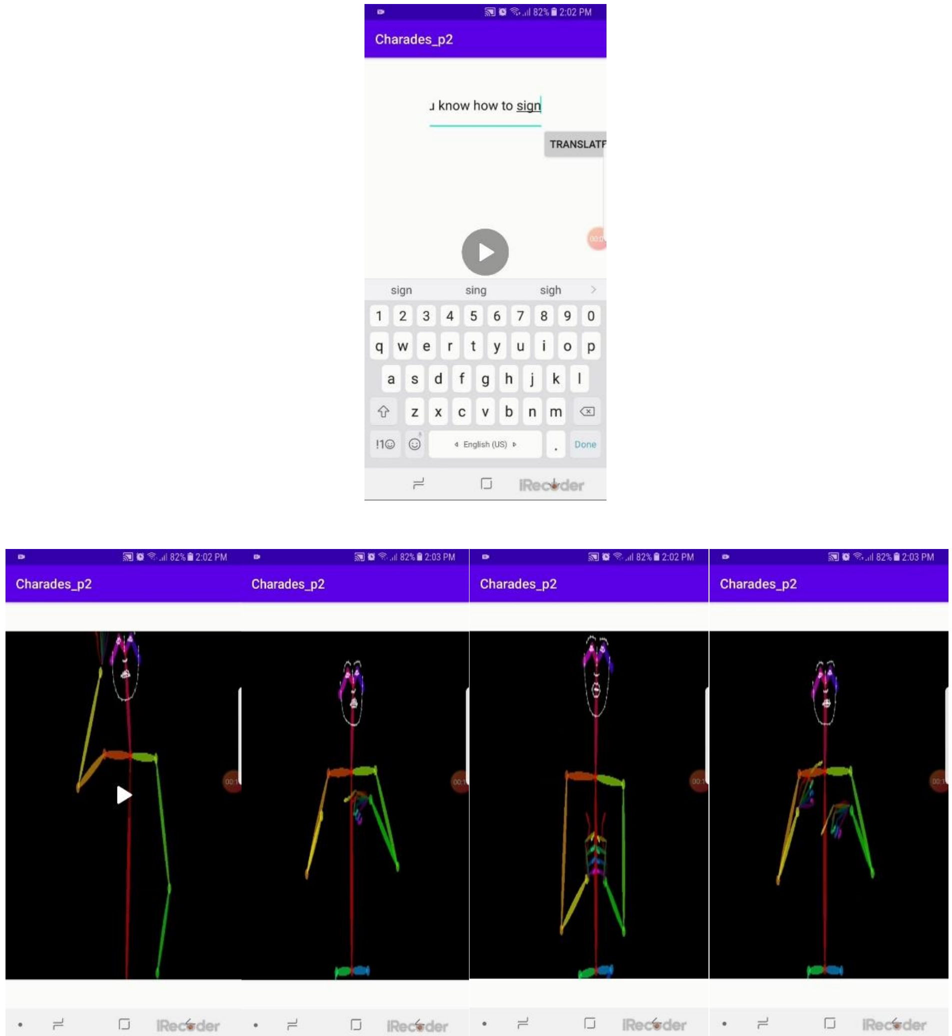

Figure 22: The working state of text-to-sign module

Figure 22 shows how the sentence “Do you know how to sign?” is converted to a series of corresponding signs in American sign language.

4.1.6 Neural machine translation model:

Table 20: Test Case ID:6

Test case ID 1: Tc-06	
Application Name:	Charades
Use Case(s):	Neural Machine translation (NMT) model
Input Summary:	A dataset of correctly synchronized parallel corpus.
Output Summary: **If Success:** Correct translations. **Else:** Wrong translations.	
Pre- conditions:	The dataset should be relevant and accurate with around 80000 instances and vast vocabulary.
Post-conditions:	The accuracy of model should be reasonably good, with a good BLEU score around 1.

Table 20 describes how the NMT model is trained on synchronized parallel dataset for translations.

The model was trained on around 87000 parallel sentences. The best model was found at step 2900 and the accuracies and perplexities are as follow:

```
[2020-05-06 12:18:05,376 INFO] Step 2900/100000; acc:  93.35; ppl:  1.47;
```

Figure 23: Training accuracy and perplexity at step 2900

Figure 23 shows the training accuracy of 93.35 and perplexity of 1.47 at step 2900.

```
[2020-05-06 12:18:18,679 INFO] Validation perplexity: 1.63272
[2020-05-06 12:18:18,679 INFO] Validation accuracy: 93.3846
```

Figure 24: Validation accuracy and perplexity

Figure 24 shows the validation accuracy of 93.3846 and validation perplexity of 1.63272 at step 2900 of model training.

For evaluation of the model was done by using BLEU (Bilingual evaluation understudy) metrics as follow:

```
BLEU = 86.27, 97.2/90.1/85.2/80.9
```

Figure 25:BLEU score

Figure 25 shows the results of BLEU score from left to right, average BLEU score then BLEU 1, BLEU 2, BLEU 3, BLEU 4, respectively.

The translation results achieved are:

```
input.txt ×    pred.txt ×
hello, my name is Ramsha
we had a meeting yesterday
call me at five
i am eating lunch
thank you so much for this favor.
where is library?
can i borrow this book?
i am a teacher.
this is not right.
Do you know how to sign?
```

Figure 26:Input file snippet given to NMT model

input.txt pred.txt

```
my name barbara
we have meeting yesterday
call i at five
i be lunch eat eat eat eat eat eat lunch
thank you so much for this imposture ?
where be hardwire ?
i can borrow this imposture ?
i be hardwire ?
this be not tftp ?
m ?
```

Figure 27: Predicted file snippet given by NMT model

Figure 26 and 27 shows the sample of 10 sentences given to the NMT model and its prediction to the input sentences. The results are not very good as the model was trained on a dataset which was not rich in vocabulary.

Chapter 5

Conclusions

Chapter 5

5 Conclusion

5.1 Problems faced and lessons learned:

5.1.1 Selecting sign language to target:

The first problem faced for this project was the sign language to target, The number of sign languages used around the world exists between 138 to 300 [21]. We wanted to either target the international official sign language or Pakistani sign language (PSL). There is no official sign language declared yet and we could not find a proper Pakistani sign language dataset.

5.1.2 Researching for datasets:

Our project comprises of two parts. Both parts required dataset of different nature. Firstly, a dataset of signs was required for recognition in sign-to-text part and displaying signs in text-to-sign part. Then a sign language gloss dataset was required for text-to-sign part. So, we selected American sign language which has more sign language users amongst others whose datasets were available.

In sign-to-text part, dataset was hard to find as most datasets available were of either picture of alphabets or a handful of videos. Deep neural networks require large datasets. Then a video dataset was found containing 9747 videos of 4-5 different signers per word with ~2500 categories of words [22].

The ASL gloss dataset found was not vast in vocabulary although with 87000 sentences in parallel corpus.

5.1.3 Issues while training and testing model:

In sign-to-text part the input we will acquire is in form of a series of signs that are captured on video so a CNN model trained on image dataset did not result as expected so we transferred to video dataset. After being transferred on video dataset a lot of technical difficulties were faced due to processing requirements this part of the project requires. Hardware wise we had to increase our capacities as processing and training a deep neural network on videos require extremely powerful graphic card.

For text-to-sign part required a neural machine translation model that translated English sentences to ASL gloss sentences. The ASL gloss dataset files after pre-processing had some issues that needed to be resolved i.e. the articles were removed which resulted in word ‘the’ being removed from words like whether and there and then etc. So, the whole dataset was scanned to remove such mistakes. Dataset was also skimmed several times to make sure the sentences were parallel and correct in sequence. Then the sequence to sequence model was trained using this dataset. The BLEU metric was used for evaluation and the scores are as follow:

BLEU 1: 97.2

BLEU 2: 90.1

BLEU 3: 85.2

BLEU 4: 80.9

5.1.4 Issues with android app:

In text-to-sign part the videos are being displayed at the end as the output. There is a codec issue with android so the whole 8747 videos were converted from .mov to .mp4. Then the resultant output video was too fast so the whole dataset speed was reduced by half.

Deployment of Sign language recognition model and NMT model along with the video dataset was extremely computationally challenging so we are using a database server.

5.2 Project summary:

For this project, we have investigated the problems faced by the deaf community all around the globe. Along with the problems faced by IT community to develop a system to overcome these problems. The systems available lack vast vocabulary and do not support both ways communication. Lack of data available on sign language and its gloss makes it difficult to develop smart interpreters. Using the limited and inadequate data available on sign language we are developing the sign language interpreter that is portable and easy to use. The application is capable of both way communication. i.e. Sign-to-Text and Text-to-Sign.

5.3 Future work:

There are numerous improvements that can be implemented in this project to further enhancement. Some of the possible suggestions are mentioned below:

5.3.1 Idea 1

The application can also integrate the knowledge of facial expressions and body language more exclusively so that there is further better understanding of the context and tone of the input speech.

5.3.2 Idea 2

The application can be extended to incorporate multiple sign languages and native languages so the solution can become universal. This can also help in deaf person to person communication of different regions also easier.

5.3.3 Idea 3

In the existing application more AI could be incorporated to make the interpreter more reliable and efficient for unseen or similar sign language data. New technologies could be incorporated like virtual reality or augmented reality for more real time experience.

5.4 GITHUB repository link:

The following repository link has all the code from training of models to server and front end of the application.

https://github.com/mishu45/CharadesFYP.git

5.5 Project description link:

The project's brief description is available at MPVIR site.

https://sites.google.com/view/mpvir/projects/charades

Chapter 6

References

6 References

[1] "World health organization," 15 march 2018. [Online]. Available: https://www.who.int/news-room/facts-in-pictures/detail/deafness. [Accessed 19 July 2020].

[2] “CareerExplorer,” 2018. [Online]. Available: https://www.careerexplorer.com/careers/sign-language-interpreter/salary/. [Accessed 19 July 2020].

[3] W. Tao, M. C. Leu and Z. Yin, “American Sign Language Alphabet Recognition Using Convolutional Neural Networks with Multiview Augmentation and Inference Fusion,” *Engineering Applications of Artificial Intelligence,* vol. 76, 2018.

[4] K. Modi and A. More, “Translation of Sign Language Finger-Spelling to Text,” *International Journal of Computer Applications (0975 – 8887),* vol. 77, 2013.

[5] “Used for Aforge.NET library and framework details,” [Online]. Available: http://www.aforge.net/.

[6] L. Pigou, S. Dieleman, P.-J. Kindermans and B. Schrauwen, “Sign Language Recognition Using Convolutional Neural Networks,” in *European conference on Computer Vision*, 2015.

[7] H. Haitham and S. Abdul-Kareem, “Static hand gesture recognition using neural networks,” Artificial Intelligence Review41, 2014.

[8] R. San-Segundo, J. M. Montero, R. Co´rdoba, V. Sama, F. Ferna´ndez, L. F. D’Haro, V. Lo´pez-Luden˜a, D. Sa´nchez and A. Garcı´a, “Design, development and field evaluation of a Spanish into sign,” Springer-Verlag London Limited 2011, London, 2011.

[9] “Essential Sign Language Information on Government Networks,” [Online]. Available: https://www.sign-lang.uni-hamburg.de/esign/. [Accessed 20 July 2020].

[10] “University of East Anglia,” [Online]. Available: https://www.uea.ac.uk/computing/visicast. [Accessed 20 July 2020].

[11] Manzano and D. Moreno, “ENGLISH TO ASL TRANSLATOR FOR SPEECH2SIGNS,” 2018.

[12] A. Vaswani, N. Shazeer, N. Parmar and J. Uszkoreit, “Attention Is All You Need,” arXiv, 2017.

[13] A. Othman and M. Jemni, “English-ASL Gloss Parallel Corpus 2012: ASLG-PC12,” in *5th Workshop on the Representation and Processing of Sign Languages LREC12*, 2012.

[14] S. Stoll, N. C. Camgoz, S. Hadfield and R. Bowden, “Sign Language Production using Neural network and generative adversarial network,” in *British machine vision conference*, Newcastle, 2018.

[15] K. Rogers, “vice,” 27 March 2018. [Online]. Available: https://www.vice.com/en_us/article/zmgnd9/app-to-translate-sign-language. [Accessed 19 July 2020].

[16] E. Matcher, 6 October 2015. [Online]. Available: https://www.smithsonianmag.com/innovation/wearable-device-translates-sign-language-english-180956827/. [Accessed 19 July 2020].

[17] “KinTrans,” [Online]. Available: https://www.kintrans.com/. [Accessed 19 July 2020].

[18] M. Reynolds, “newscientist,” 2 June 2017. [Online]. Available: https://www.newscientist.com/article/2133451-automatic-sign-language-translators-turn-signing-into-text/. [Accessed 19 July 2020].

[19] “SignAll,” [Online]. Available: https://www.signall.us/. [Accessed 19 July 2020].

[20] B. Coxworth, “New Atlas,” 28 October 2015. [Online]. Available: https://newatlas.com/live-time-closed-captioning-system/40078/. [Accessed 19 July 2020].

[21] R. Brooks, “k-international,” 10 May 2018. [Online]. Available: https://k-international.com/blog/different-types-of-sign-language-around-the-world/. [Accessed 5 May 2020].

[22] [Online]. Available: https://www.cin.ufpe.br/~cca5/asllvd-skeleton/index.html. [Accessed 16 July 2020].

Turnitin Originality Report

CharadesFYP2_thesisFinal.docx by Anoniim

From Research Papers 2 (Research Papers 2)

- Processed on 28-Jul-2020 3:58 AM PDT
- ID: 1363198421
- Word Count: 8441

Similarity Index
14%
Similarity by Source

Internet Sources:
4%
Publications:
3%
Student Papers:
13%

sources:

1 1% match (student papers from 07-Aug-2012)
Submitted to CSU, San Jose State University on 2012-08-07

2 1% match (student papers from 22-Dec-2016)
Submitted to Higher Education Commission Pakistan on 2016-12-22

3 1% match (student papers from 15-May-2018)
Class: OS
Assignment: Assignment 1
Paper ID: 964039549

4 1% match (student papers from 25-May-2020)
Submitted to Study Group Australia on 2020-05-25

5 < 1% match (student papers from 11-Jun-2019)
Submitted to Higher Education Commission Pakistan on 2019-06-11

6 < 1% match (student papers from 16-May-2018)
Class: OS
Assignment: Assignment 1
Paper ID: 964524992

7 < 1% match (Internet from 24-Jan-2013)
http://ijsaa.org/15ICRASE121112.pdf