Khulna University of Engineering & Technology

Report On

CSE 3200: System Development Laboratory

Topic: Speech to Sign Language Translator Application

Supervised By:

Dr. Sk. Mohammad Masudul Ahsan

Professor

Department of Computer Science and Engineering
Khulna University of Engineering & Technology, Khulna

Submitted by:

Ashikur Rahman

Roll: 1707090
Year: 3rd Term: 2nd
Department of Computer Science and Engineering, KUET

Samin Yeasar

Roll: 1507099
Year: 3rd Term: 2nd
Department of Computer Science and Engineering, KUET

Date of Submission: June 24, 2021

Approval

This Project Report has been submitted for examination with the approval of our supervisor.

Dr. Sk. Md. Masudul Ahsan

Professor

Department of Computer Science and Engineering

Khulna University of Engineering and Technology, Khulna

Acknowledgement

We would like to express our gratitude to our honorable supervisor, Dr. Sk. Md. Masudul Ahsan Sir. We are highly indebted to him for his guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project.

Table of Contents

Table of Figures

Introduction

Background

According to the World Health Organization (WHO), over 6.1% of the world's population suffers from disabling hearing loss.[i] Six percent may seem like a small number, but that totals over 432 million people across the globe. Sign language is one of the very limited techniques of this huge population can communicate with each other and with normal people.

Sign languages (also known as signed languages) are languages that use the visual-manual modality to convey meaning. Sign languages are expressed through manual articulations in combination with non-manual elements. Sign languages are full-fledged natural languages with their own grammar and lexicon.[ii] Sign languages have developed as useful means of communication for people with hearing loss and speech disability.

There is no better option than sign language to communicate with people with hearing loss. But it is not always possible for normal people to learn and practice sign language. Our application will help normal people to communicate with them by translating their speech into a visual representation of sign language.

Problem Statement

We have to build a software application which can take a speech as input and then translate the meaning of speech into sign language representation understandable for people with hearing loss. This application should be built in such platform which will be compatible with maximum types of devices and should be easy to access and to use.

Main Objective

The main objective of the system is to translate conventional sentences, at least words, into sign language.

Specific Objectives

- To design a system for translating speech to sign language.
- To build the system.
- To use suitable technology.
- To design easy and effective user interface (UI) of the system.
- To test the system and solve regarding problems.
- To make the system available in suitable platforms.

Scope of the study

There are many NLP algorithm available for converting "Sign language to text" conversion. But for proper communication with hearing challenged people, we need a two-way conversion system of sign language. That is why, a text/speech to sign transformation system is necessary.

In this project, we have worked on that part of converting text/speech to sign language. We have to limit the system to only desktop application. But since the framework is a cross platform, the same codes can be used to build mobile or web application with some modifications. The input of the system is limited to only English speech and text and as output visualization, we use sign linguistic properties of American Sign Language (ASL).

Significance of the system

- Normal people can use this system to communicate with people with hearing loss.
- This is the first and only work of speech to sign language translation in Kivy framework of Python language.
- The same code can be used to build the system for other platforms like Linux, OS X, Android, IOS, etc. as Kivy is a cross-platform Python framework.
- The new Python module built solely for this system can be used in other video related programs or projects.[iii]
- This is a very lightweight application, so low usage of system resources is required.
- The system can interpret directly from speech normally, so one does not have to type again and again.
- Easy to use, even for non-tech savvy users.
- The system can be used for educational purposes such as to teach or to explain something in sign language to the students with hearing loss.

Related Works and Software

Previous Developments

In the recent years, there has been several works and developments in speech to sign language interpretation. Khalid et al.[iv] represented "Speech to Sign Language Interpreter System (SSLIS)" to translate uttered English speech into video American Sign Language (ASL) in live mode. The Sphinx 3.5 was manipulated as the speech recognition engine for the SSLIS and for translation ASL syntax was not followed, but rather the Signed English (SE) manual was employed as a manual parallel to English.

Signtel Inc patented and launched "Signtel Interpreter™ [v]" in market in 2010. It can translate speech and text into sign language video with the Signtel Interpreter VR (voice recognition). It translates text from the Internet or Word document.

The most recent work in this area is a mobile application named "Hand Talk"[vi] which has already been downloaded more than one million times, works by either speaking directly into it or manually typing in the words. It then displays an on-screen avatar who translates the words into sign language.

Our Contributions

We built a speech to sign language system on Kivy, a cross-platform Python framework for Natural User Interface (NUI) development focusing on usages of the the system on various platforms like Windows, Linux, OS X, Android, IOS and so on. We use Google Cloud Speech API[vii] for voice recognition purposes. Also, ours is a lightweight system. The system uses less resources.

Workflow

- **Speech recognition:** Speech have to be recognized and converted to corresponding text.
- **Tokenize the Input:** The converted text is divided into necessary tokens.
- **Matching with Database:** Corresponding images of signs available for each token in input are searched in predefined database.
- **Creating sign language video:** Merging the corresponding images, a video is created and shown to the users as output.

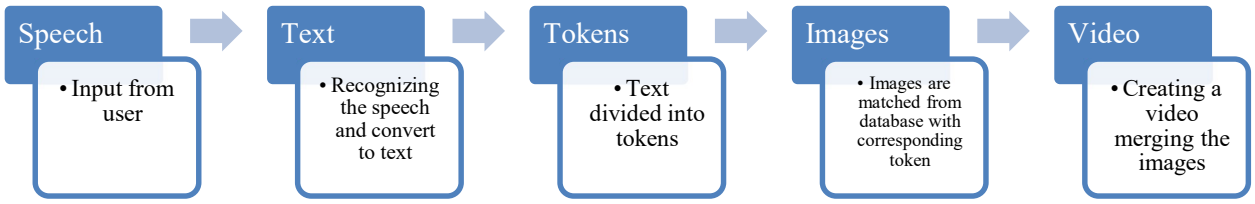

Figure 1: Work flow of the project

System Architecture

Overview

Resources Used in Development:

- **Language:** Python 3.6[viii]
- **Framework:** Kivy[ix]
- **API:** Google Cloud Speech-to-Text API
- **Libraries**: PyAudio v0.2.11[x]
  SpeechRecognition 3.8.1[xi]
- **IDE**: PyCharm 2021.1.1[xii]

Hardware Requirements:

- Desktop/Laptop Computers
- Mobile Phone
- Microphone / Audio input port

Software Requirements:

- **Operating Systems:** Windows 7 or later, macOS, Linux, Android

Other Requirements:

- Internet connectivity for output from speech input

Analysis

Key features:

- Speech collected from audio input of device is converted into sign language and shown in output through a video.
- Deaf people watch the video and understand what the user is trying to express.
- User can save video for using later.
- User can upload recordings as input.
- User can also enter text directly for conversion.
- User can update the database with their own images.

System Flow:

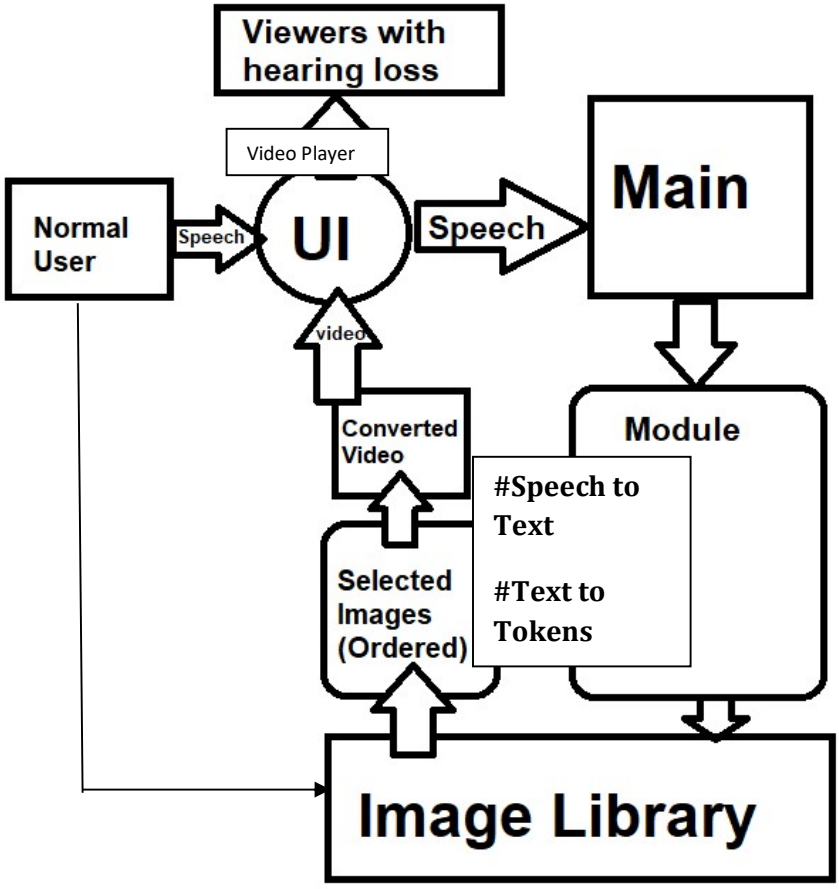

Figure 2: System Control Flow

- Normal user's speech is taken through user interface.
- Speech or input is reached module through main program.
- The module is converted the speech into text.
- The text is processed and converted into tokens excluding punctuation marks and prepositions.
- Corresponding images of tokens from library are selected and saved at a temporary storage in an ordered way.
- These images are merged and turned into a video.

- The video is played in video player of graphical user interface.
- Tasks are controlled and executed by module.
- Main program communicates between module and user interface.
- User can add images in library.

Database:

- **Saved Images:** This folder works as image database storing the predefined sign language images renamed as the corresponding word.
- **Final Images:** This is temporary storage for keeping the fetched images from the sign language image database to be processed for the video.
- **Saved Video:** This folder keeps the converted final output videos.

Design

UI structure:

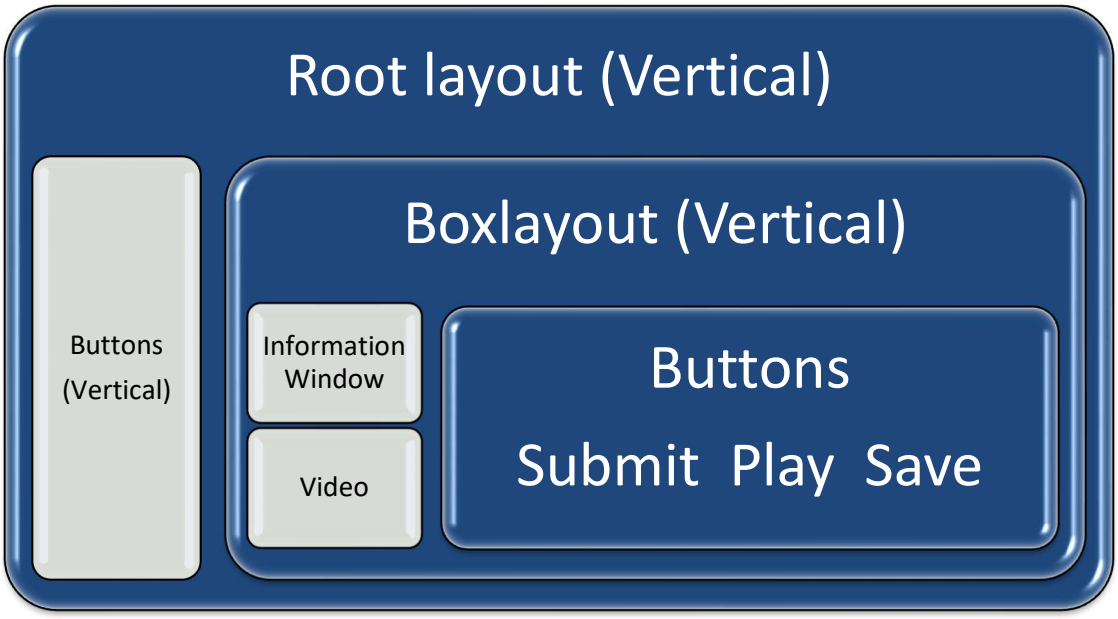

Figure 3: User interface structure

Backend Structure:

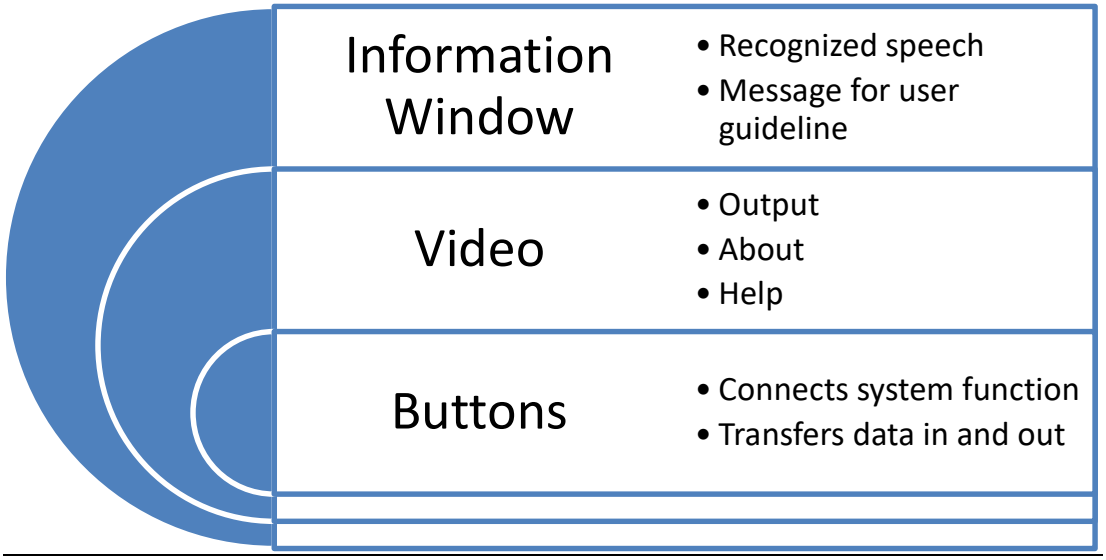

Figure 4: Backend structure

Development

Libraries:

- **PyAudio:** PyAudio is a set of Python bindings for PortAudio, a cross-platform C++ library interfacing with audio drivers. With PyAudio, Python can be easily used to play and record audio on a variety of platforms.
- **SpeechRecognition:** SpeechRecognition is a Python library for performing speech recognition, with support for several engines and APIs, online and offline. This library was imported for recognizing the speech from user and for converting into text.

ASvdo Module:

- **Audio to text:** *audiototext* function converts the audio into text using speech recognition library.
- **Tokenize:** *del_punct* and *tokenize* function divide the text into tokens.
- **Fetching image:** *fetchimg* function takes the tokens and fetch corresponding images from the image database and sort them in order as the tokens. If the image of corresponding

word is not found, then the word will be divided into letters and corresponding images of those letters will be fetched.

- **Video:** *ImgtoVid* function makes a video merging the images one after another in the same order.
- **Clear:** *clear* function cleans the temporary storage for fetched images.

Sign.py:

Sign.py works as the main function. It executes the commands given by users through the interface.

- **Record:** Records the speech from user.
- **Options:** *spinner_clicked* provides various options like upload text file or audio file.
- **Save:** *save* function save the video in user's selected folder.
- **Play:** Plays the output video.
- **Add image:** User adds new images in the database by this function.

User Interface:

An interactive GUI (graphical user interface) was implemented by writing a Kivy script named *speech.kv*. This file provides interactive button to users for commanding and connecting with the system.

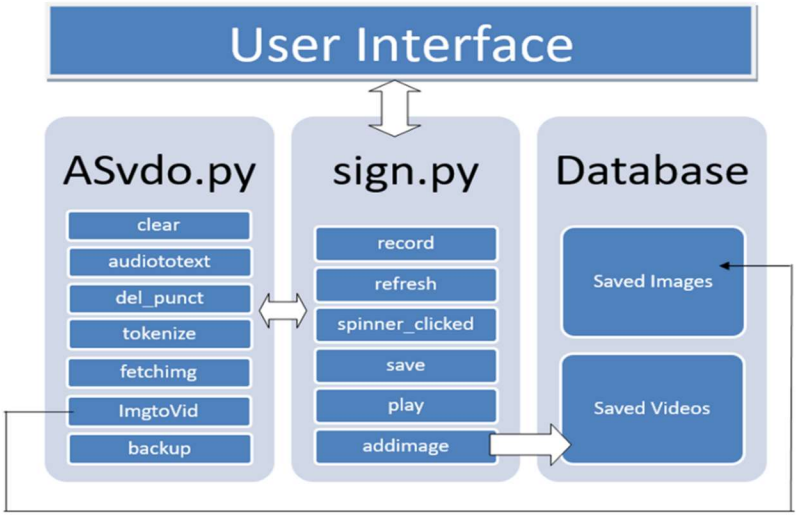

Figure 5: Implementation of the system

User Manual

Duration Selector: This is a drop-down menu where user can select the duration of voice speaking. There are 4 values set in second: 3 s, 5 s, 7 s and 10 s. By default, it is set to 3 s.

Speak Now: This button starts capturing user voice for the duration amount of time set in "Duration Selector". Also it internally converts the voice to text and displays that text on the grey colored notification window.

Options: This a drop-down menu where user can select the alternative method of real time voice input. Usually this is handy when the user is not connected with internet. It provides selection of other ways of converting such as: Upload pre-recorded audio, Upload text file, directly entering text.

Add Image: Clicking this user can add images to the image dataset.

About: Displays an information media about the developers and supervisor.

Contact: Displays contact information with the developers for any further query.

Help: Displays a media manual of common problems and the right way to use this software without facing any trouble.

Refresh: Refreshes the screen. Actually, it restores the UI as the starting window.

Submit: When user chooses an alternative way of converting speech to sign such as Uploading pre-recorded audio or Uploading text or directly entering text via "Options", the system asks for the file address and user writes that. This button sends the address to system function for further processing.

Play: This button starts playing the converted video.

Save: This button triggers functionality for saving the currently playing video as record in a backup folder.

Quit: Closes the window and exit program.

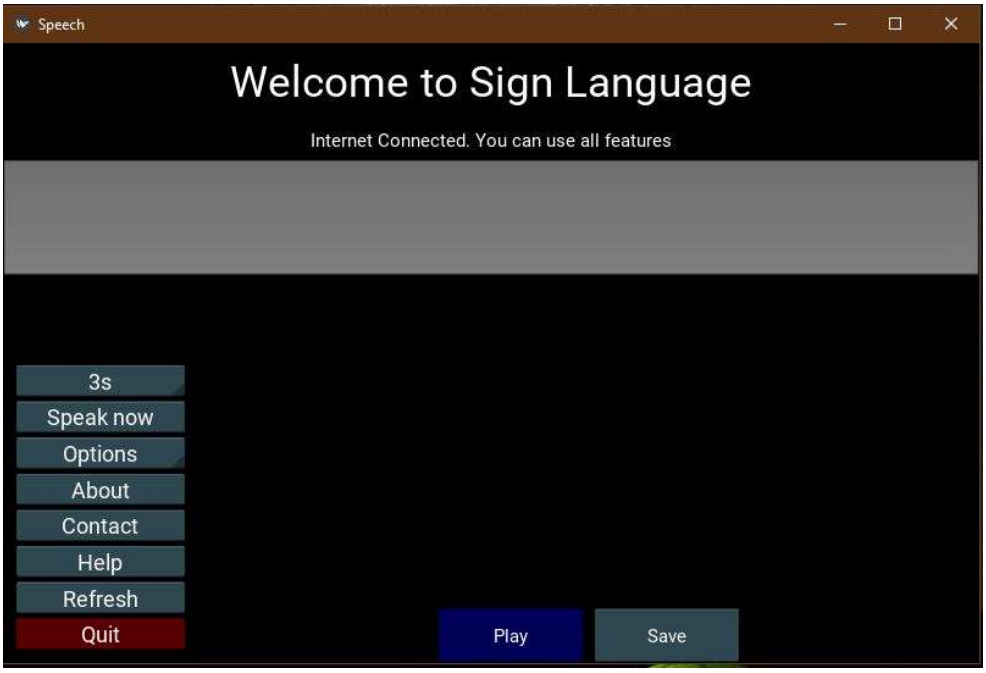

Figure 6: Graphic User Interface

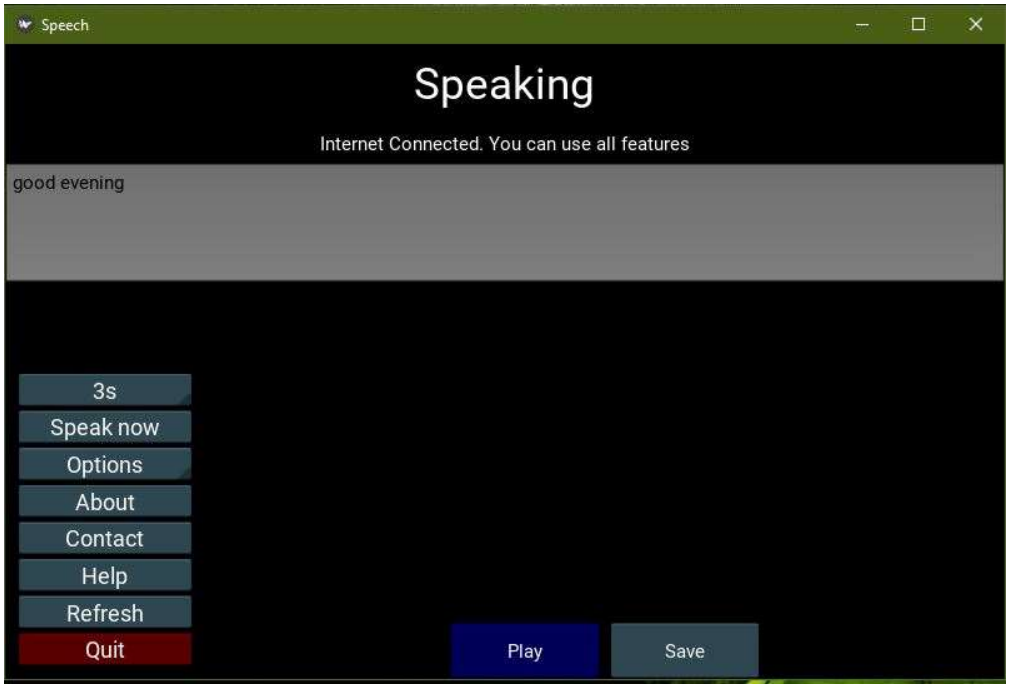

Figure 7: GUI while speaking

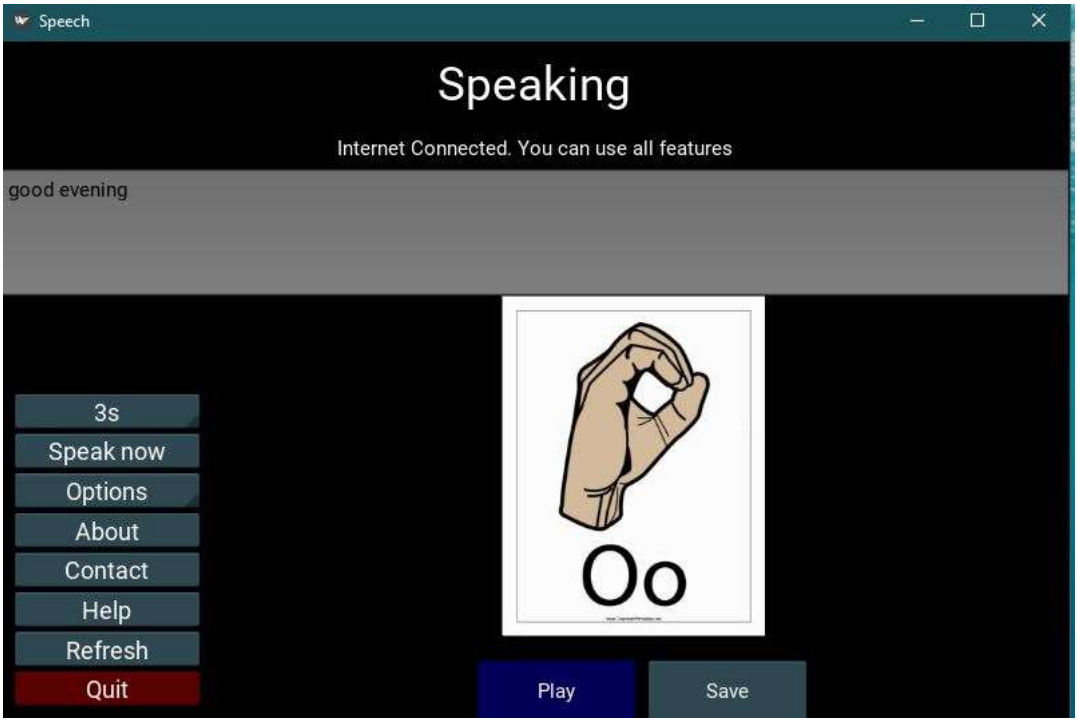

Figure 8: Video is playing in GUI

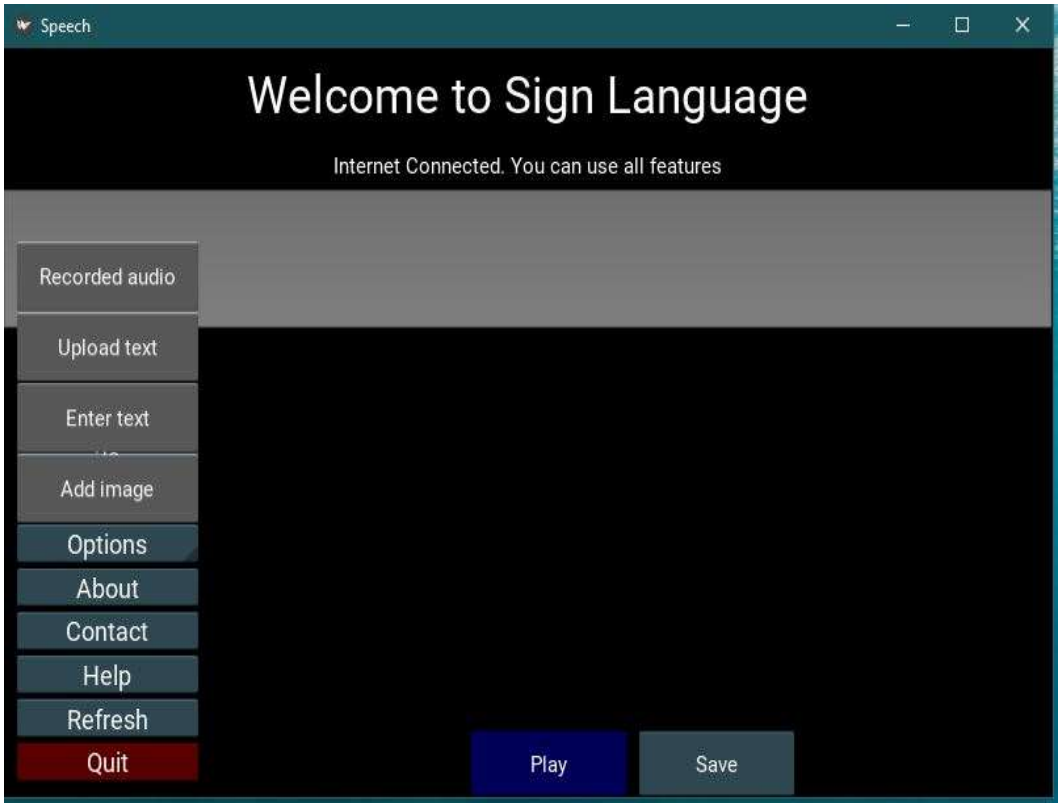

Figure 9 : Different input options

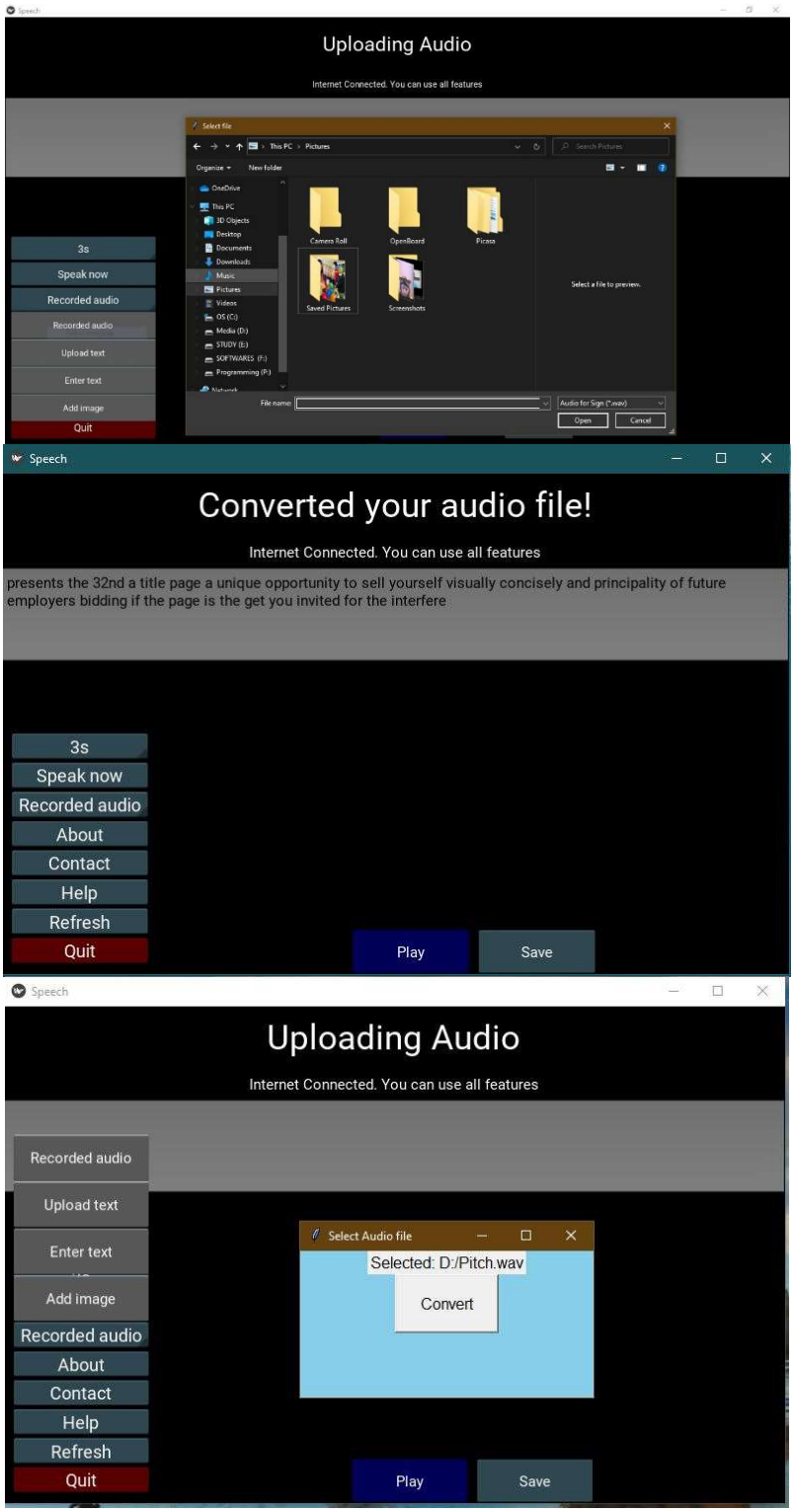

Figure 10 : Uploading audio file

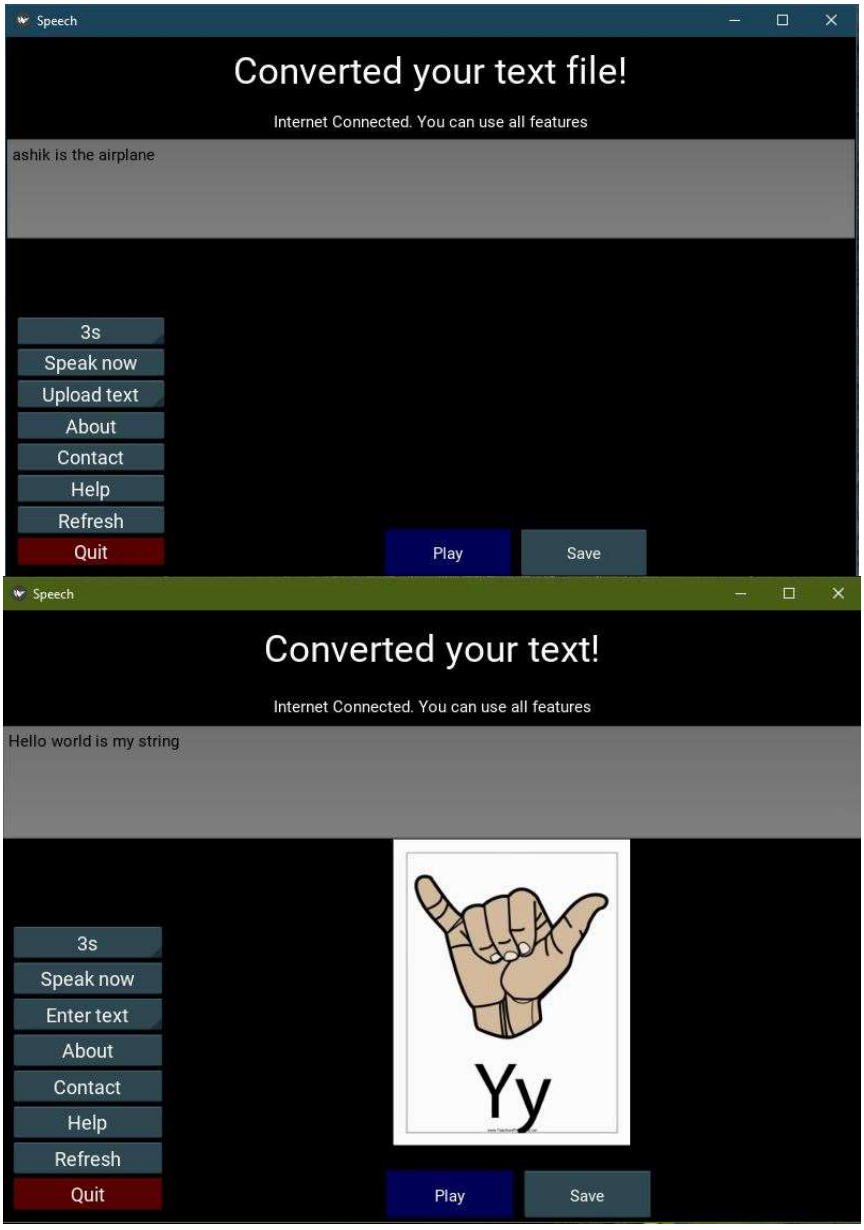

Figure 11: Uploading text file

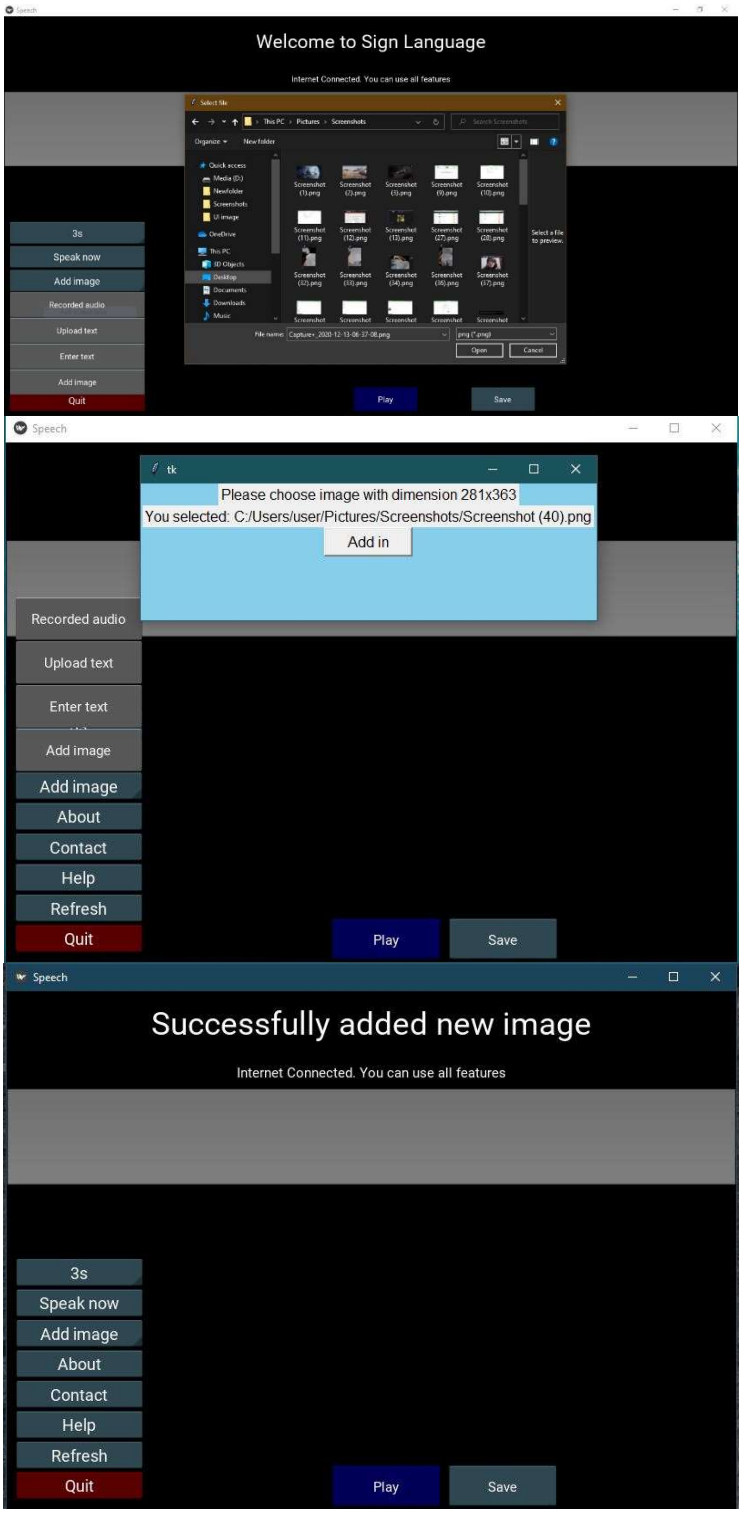

Figure 12: Adding new image to database

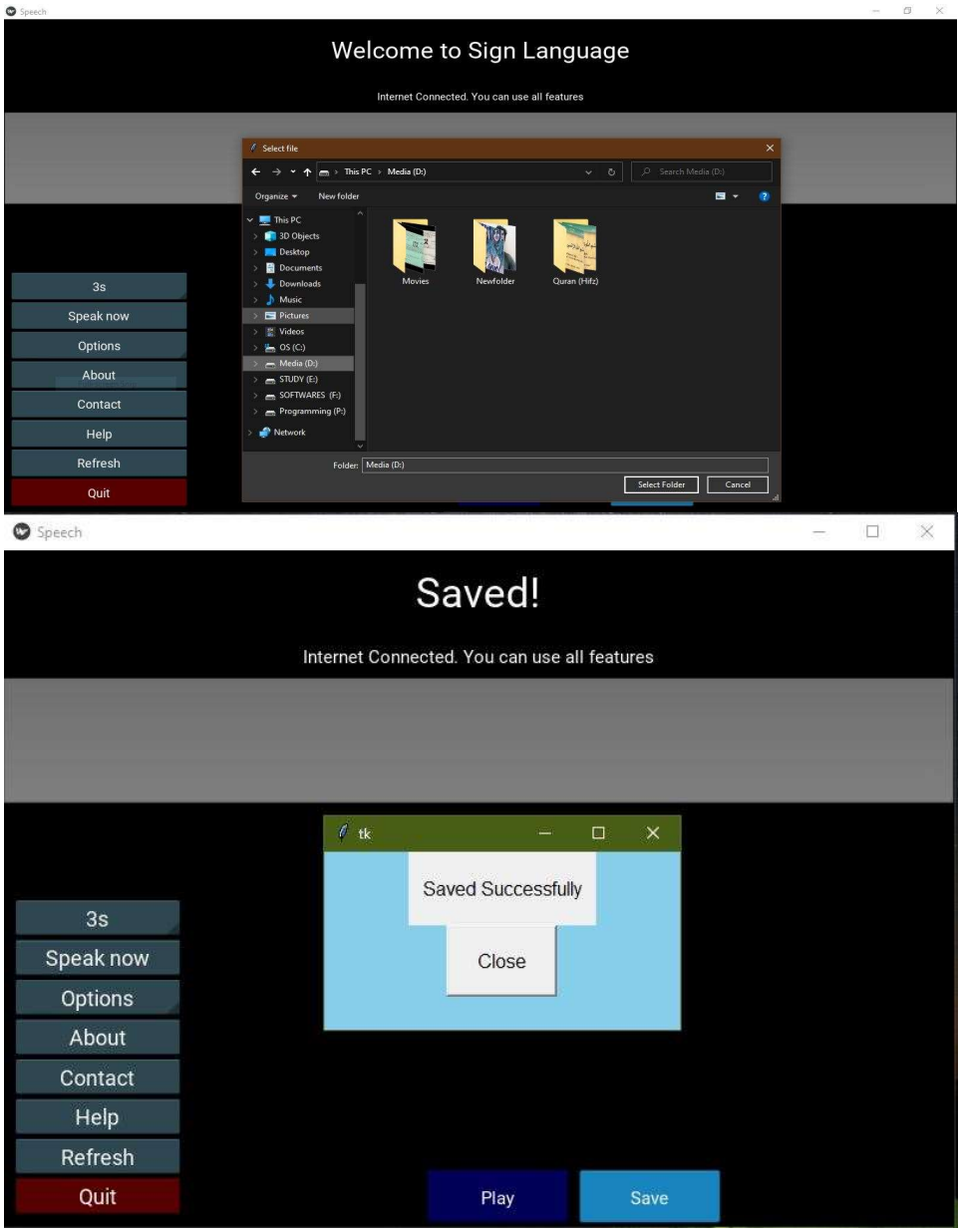

Figure 13: Saving the video in folder

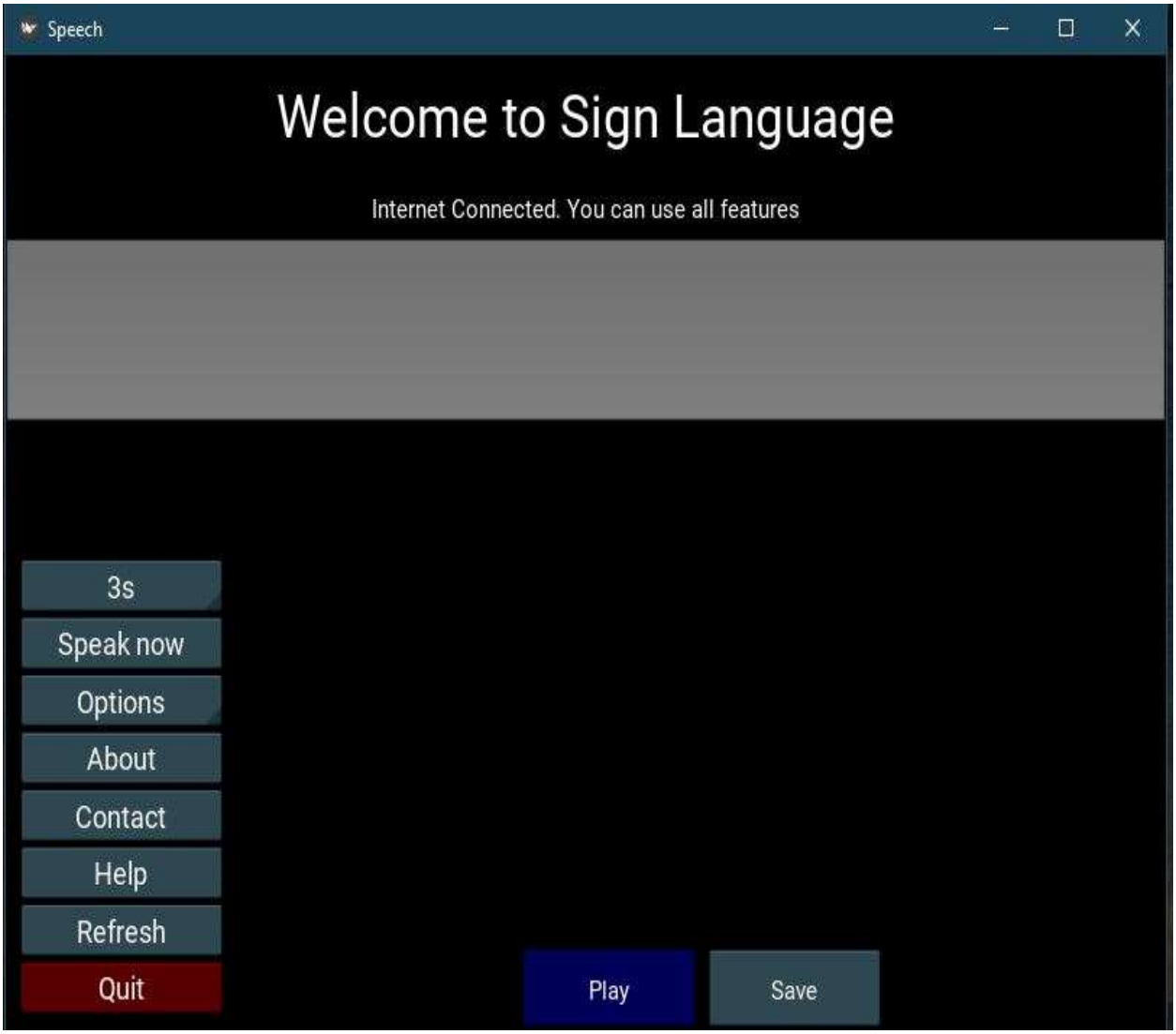

Figure 14: After clicking Refresh button

Limitations and Recommendations

We have to face some obstacles during implementing the system, so there are some limitations remaining in the system. The first limitation is lack of appropriate images of sign equivalent of corresponding words. We could not manage enough images of signs as they are not available in internet or other sources. The second limitation is the system is not supported for Python 3.7 or above because these do not support PyAudio library. Thirdly, there were some limitations while designing the user interface (UI) such as less button and color style options, notifications and hovering limitations, etc. because of limited features and functionality of Kivy as it is a comparably new framework. The limited duration for user to speak the speech is also a very large drawback in the system. Also, we wanted to implement this system for Bangla sign language but we were unable to find a good library for Bangla speech recognition and enough information and resources on Bangla sign language. Even there is no digital records or dictionary of Bangla sign language was not found.

For further development we are suggesting to build a complete image library of sign equivalents of corresponding words. Detailed research and development are required for more smooth real time speech to sign language interpretation so that the online and virtual meeting can use it as a real time translation plugin or tool for communicating with deaf people properly. Also, there is a lot of scopes for research and development in the field of Bangla speech recognition and Bangla sign language. Bangla verbal language needs a speech recognition library and Bangla sign language needs a very well-decorated and digitalized library or at least a documentation for the welfare of people with hearing loss living in Bangladesh.

Conclusion

We completed developing the system based on trial-and-error method. We divided the main problem into small functions and tried to do everyone's part our best. While developing the system we had faced so many limitations and issues due to our inexperience and lack of knowledge, but under strong guidance of our honorable supervisor sir we were able to overcome them. We expect that our system will come with benefits for the people with hearing loss and for the ongoing and following research works in this field.

References

[i] https://www.who.int/home/cms-decommissioning

[ii] Sandler, Wendy; & Lillo-Martin, Diane. (2006). Sign Language and Linguistic Universals. Cambridge: Cambridge University Press.

[iii] https://github.com/Ashik-90/SignLanguage/blob/main/ASvdo.py

[iv] Khalid El0Darymli, Othman O. Khalifa, and Hassan Enemosah, "Speech to Sign Language Interpreter System (SSLIS)", *the IEEE International Conference of Computer and Communication Enginnering (ICCE'06), in Proceedings of*, Kuala Lumpur, Malaysia, 2006.

[v] https://www.signtelinc.com/signtel-interpreter.html

[vi] https://handtalk.me/br

[vii] https://cloud.google.com/speech-to-text/

[viii] https://www.python.org/downloads/release/python-360/

[ix] https://kivy.org/#home

[x] https://pypi.org/project/PyAudio/

[xi] https://pypi.org/project/SpeechRecognition/

[xii] https://www.jetbrains.com/pycharm/