**Chapter 1**

**Introduction**

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-mutual elements.Sign languages are full-fledged natural languages with their own grammar and lexicon. Sign languages are not universal and they are not mutually intelligible with each other, although there are also striking similarities among sign languages

So far, we had the opportunity of using Speech Recognition to convert it to text. The new idea that we shall be proposing in this project will be the incorporation of Speech to Text, Text-Sign Language using various applications and combining each independent function to together form a better application in totality.

**Chapter 2**

**Literature Survey**

Paper 1:-V2S: Voice to Sign Language Translation System for Malaysian Deaf People

Authors:Oi Mean Foong, Tang Jung Low, and Wai Wan La

Computer & Information Sciences Department, Universiti Teknologi PETRONAS,

Bandar Sri Iskandar, 31750 Tronoh, Malaysia

Summary: This paper proposes a solution to this problem by providing a voice (English Language) to sign language translation system using Speech and Image processing technique. This project uses template-based recognition as the main approach in which the V2S system first needs to be trained with speech pattern based on some generic parameter set.

In this paper the proposed method is such that, first with the help of microphone the voice is recorded and this voice is further sent to the digital signal processing unit, now in the DSP stage, MFCC(Mel Frequency Cepstral Coefficients) algorithm is used in formant extraction process.After the extraction process, the next step is the vector quantization step, this is one of the best matching techniques used for speech recognition.The basic concept of vector quantization is to compress any vector of a speech/voice feature into one scalar vector. These one scalar vectors are sent to the video database for matching purposes, one its matched, the video for the sign language is played.

Paper 2:- Speech to Sign Language Interpreter System (SSLIS)

Authors:Khalid Khalil El-Darymli , Othman O. Khalifa and Hassan Enemosa

International Islamic University Malaysia, ECE Dept., Faculty of Engineering

Summary: In this paper in the most basic form we see that the speech is recognized

and matched with the ASL(American Sign Language) and the matching text with

the ASL translation is displayed. In this again speech recognition tools are used

which covert the speech/voice to recognized text which is further compared to the

ASL database. In the speech recognition stage, the model is first trained and

then tested, they have used the ASR(Automatic Speech Recognition ) technique, in

ASR the acoustic Signal is converted to textual message.

Paper 3:- Sign Language Translator using Machine Learning

Authors: Vishwas S, Hemanth Gowda M, Vivek Chandra H N, Tannvi Students, Department of Computer Science & Engineering, Vidyavardhaka College of Engineering, P.B. No. 206, Kannada Sahithya Parishath Road, III Stage, Gokulum, Mysuru, Karnataka-570002

Summary : In this paper if the proper gestures are made, the system lets out the corresponding words. And though there are some minor differences in movements, the device will produce good results. Variations of various types of people performing the movements can vary. The machine identifies several expressions one after the other, and lets out the words in question. The model keeps on improving and goes on to give near perfect results. This sytem is based on indian sign language.

Paper 4:- Translating Indian Sign Language to text and voice messages using flex sensors

Authors: Sachin Bhat , Amruthesh , Ashik , Chidanand Das2 , Sujith

Shri Madhwa Vadiraja Institute of Technology, Udupi, Karnataka, India

Summary: In this paper a more robust, user-independent, and portable device is built to convert sign language to text message type that consumes less power due to low-power microcontroller AT89S52. Using a simple mobile app this text message can be converted into speech. This helps solve the poor contact with the rest of the world between the dumb / deaf people. The flex sensors detect the movements of the fingers and are able to deduce the sign language which is then transmitted over the system

Paper 5 :-Machine Learning Techniques for Speech Recognition using the

Magnititudes

Authors:Angeline Valentina Sweety A., Gopala Krishnan C., Mukesh Krishnan M.

Summary: In this paper, we have tried to introduce a simple technique which could be used to recognize connected speech and the person concerned. We have tried to develop a device which will enable to find the presence of a particular data from the cluster of dataset using python.The speech features extracted are compared with already saved speeches in the database for matching. we have implemented only Artificial Neuron Network (ANN) and accordingly input data’s have to be fed and output results probably the exact one is expected to be derived. By this means considerable time saving is expected and financial implications too can be minimized.

Paper 6:- Speech Translation System for Vernacular Languages

Authors: Ms. A. H. Utgikar,1 Mr. A. S. Deshpande2 , Mr. K. S. Ambulgekar3 , Mr. K. R. Joshi4 Asst. Prof., E & TC Dept., Maharashtra Institute of Technology,

Summary: Voice to Voice Language Translation system is a device that is designed to bridge the language gap between individuals and foreigners when traveling in our country. The need arises from the inability of dictionaries and human translators to suit our needs for better communication. At present we need „Personalized Interpreters‟ which will reduce our dependence on dictionaries and human interpreters. This will reduce the hindrance posed by the language barrier. A prototype which uses a speech processing hardware and on translators to provide the user with real time translation. Speech processing hardware works on the principle of „compare and forward‟, i.e., a database is already stored in the unit which is used for comparing with the input speech and the result is forwarded for further processing.

(Conclusion to why we go with our model)

We, as a group have collectively decided to go ahead with our model by using the Google API for

Speech recognition and the text processing is done via NLP(Natural Language Processing)

**Chapter 3**

**Problem Statement**

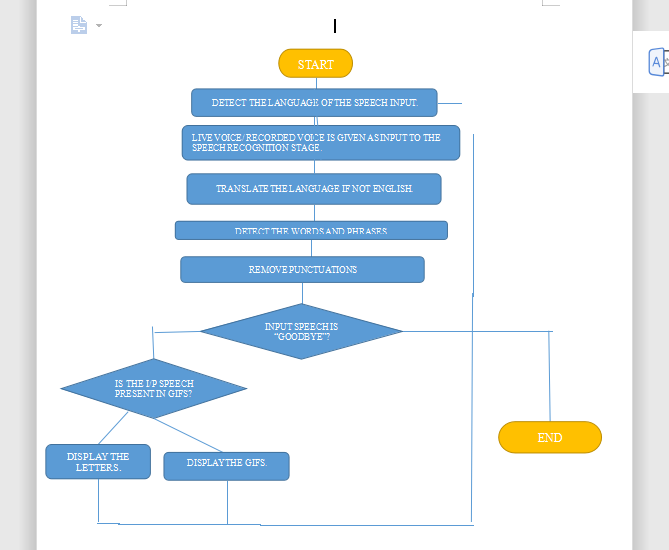
Sign language is a language that uses manual communication methods such as facial expressions, hand gestures and bodily movements to convey information.

To implement a project that translates foreign languages into English , makes use of Gifs and images for words to translate the text language into Indian sign language.

**Chapter 4**

**Project Design**

**4.1 Flowchart**

****

**4.2 Algorithm**

Step 1 : Import all necessary libraries

Step 2 : Define Functions and Initialize array of alphabets and phrases.

Step 3 : Accept audio sample using the PyAudio Library.

Step 4 : Translate into English Language using google translate library .

Step 5 : Detect words and Phrases using speech recognition library. If goodbye , then exit program .

Step 6 : Remove punctuation

Step 7 : If part of phrases display Gif from array, else spell the word out using the jpeg files of letter and Pyplot.

Step 8 : Jump to Step 3.

Step 9 : End.

**Chapter 5**

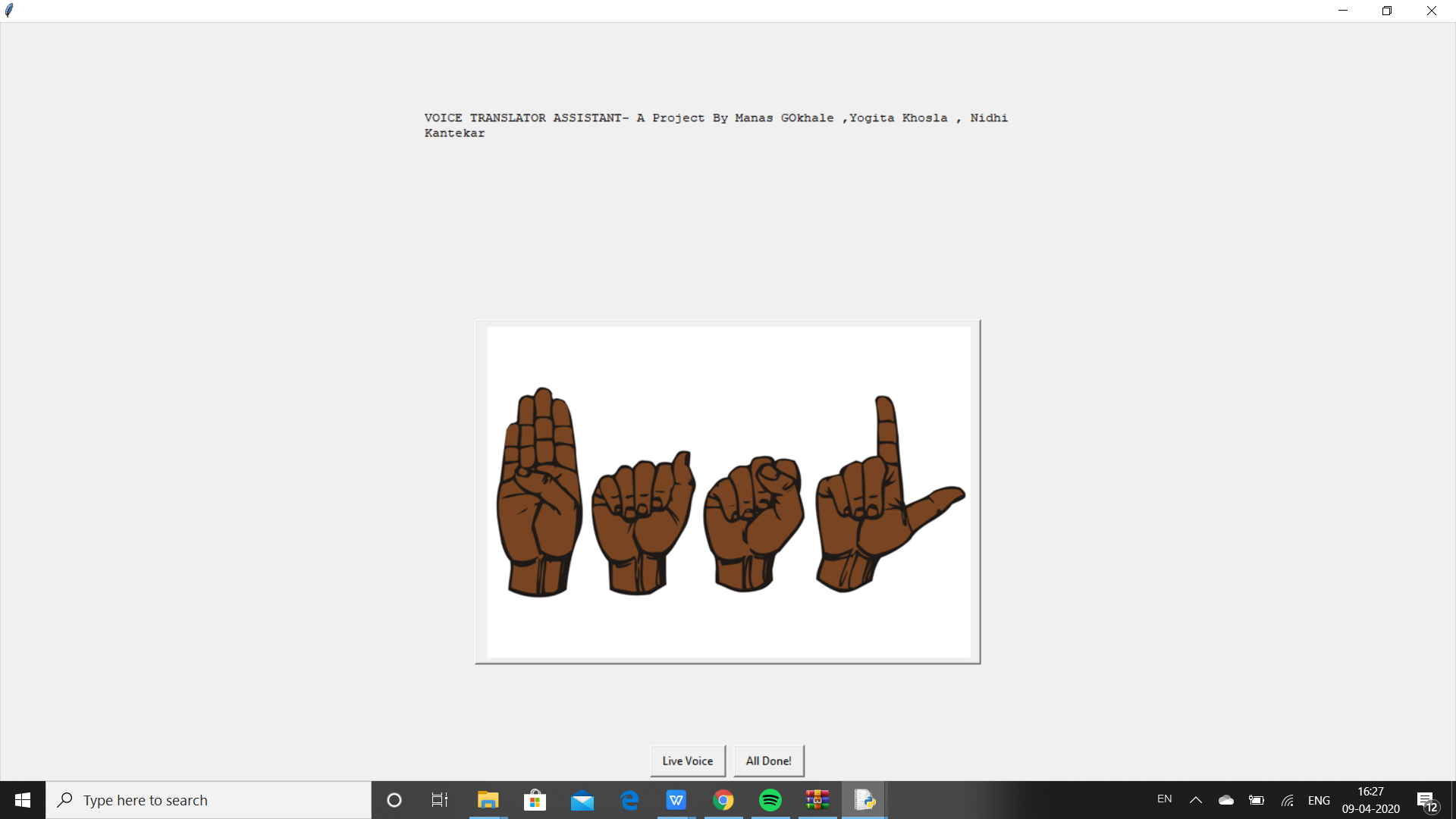
**Implementation Details**

This is an application that takes live voice as an input and converts it into text and then displays the relevant sign language or the gifs.

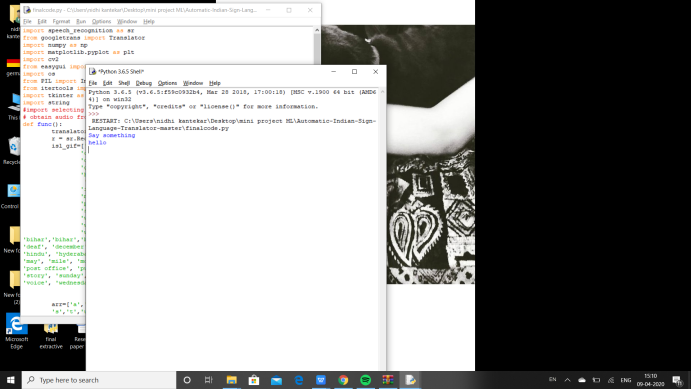
The modules we have used are:

1) EasyGui is used for the front end.

EasyGUI is a module for very simple, very easy GUI programming in Python. EasyGui provides an easy-to-use interface for simple GUI interaction with a user.



2) PyAudio is used to take the speech input through the microphone PyAudio provides Python bindings for PortAudio, the cross-platform audio I/O library. With PyAudio, you can easily use Python to play and record audio on a variety of platforms.

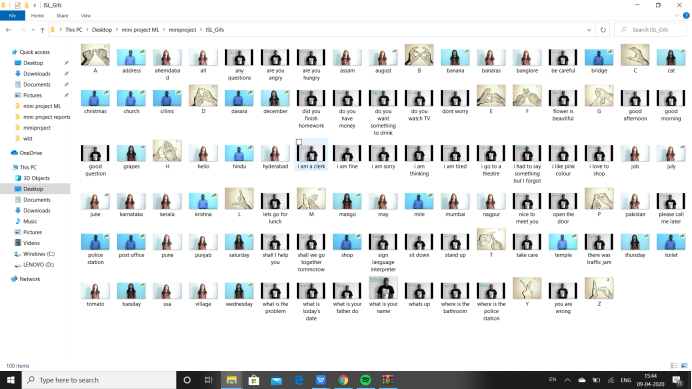


3) Speech recognition is done using Google API an the python speech recognition module “SpeechRecognition” is used. This module even configures the microphone, so we need to give the speech input via the microphone.

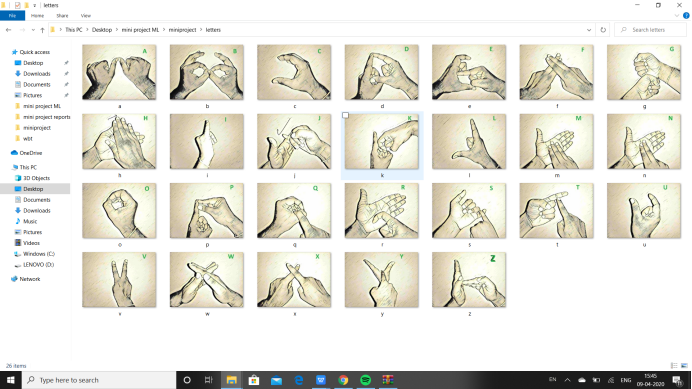
4) Text processing is done using NLP(Natural Language Processing). NLP is a subfield of computer science and artificial intelligence concerned with interactions between computers and human (natural) languages. It is used to apply machine learning algorithms to text and speech.

5) Once the speech is converted into text, we have to match it with our dataset to either display the gifs or the letters.

The dataset of gifs :



The dataset of letters:



\* Modules/ Libraries to be imported:

1) SpeechRecognition : pip install SpeechRecognition .

2) Googletrans: pip install Googletrans .

3) Numpy: pip install numpy

4) Matplotlib : pip install matplotlib.pyplot

5) CVS: pip install opencv-python .

6) EasyGui: pip install easygui

7) OS: pip install os-win

8) PL: pip install Pillow

9) Itertools: pip install itertools

10) Image: pip install image

11) Tkinter: pip install tkinter

12) String: pip install string

SOURCE CODE:

import speech\_recognition as sr #speechrecognition library is used to listen and convert it to text

from googletrans import Translator #used for translating

import numpy as np #used for arrays and mathematical

import matplotlib.pyplot as plt #use for image displaying

import cv2 #to convert image space, basically used to resize image according to our window

from easygui import \*

import os # used for file interaction

from PIL import Image, ImageTk #support for opening, manipulating, and save img file format

from itertools import count

import tkinter as tk #Python with **tkinter** is the fastest and easiest way to create the GUI applications

import string

#import selecting

# obtain audio from the microphone

def func():

translator=Translator()

r = sr.Recognizer()

isl\_gif=['all the best', 'any questions', 'are you angry', 'are you busy', 'are you hungry', 'are you sick', 'be careful',

'can we meet tomorrow', 'did you book tickets', 'did you finish homework', 'do you go to office', 'do you have money',

'do you want something to drink', 'do you want tea or coffee', 'do you watch TV', 'dont worry', 'flower is beautiful',

'good afternoon', 'good evening', 'good morning', 'good night', 'good question', 'had your lunch', 'happy journey',

'hello what is your name', 'how many people are there in your family', 'i am a clerk', 'i am bore doing nothing',

'i am fine', 'i am sorry', 'i am thinking', 'i am tired', 'i dont understand anything', 'i go to a theatre', 'i love to shop',

'i had to say something but i forgot', 'i have headache', 'i like pink colour', 'i live in nagpur', 'lets go for lunch', 'my mother is a homemaker',

'my name is john', 'nice to meet you', 'no smoking please', 'open the door', 'please call an ambulance', 'please call me later',

'please clean the room', 'please give me your pen', 'please use dustbin dont throw garbage', 'please wait for sometime', 'shall I help you',

'shall we go together tommorow', 'sign language interpreter', 'sit down', 'stand up', 'take care', 'there was traffic jam', 'wait I am thinking',

'what are you doing', 'what is the problem', 'what is todays date', 'what is your age', 'what is your father do', 'what is your job',

'what is your mobile number', 'what is your name', 'whats up', 'when is your interview', 'when we will go', 'where do you stay',

'where is the bathroom', 'where is the police station', 'you are wrong','address','agra','ahemdabad', 'all', 'april', 'assam', 'august', 'australia', 'badoda', 'banana', 'banaras', 'banglore',

'bihar','bihar','bridge','cat', 'chandigarh', 'chennai', 'christmas', 'church', 'clinic', 'coconut', 'crocodile','dasara',

'deaf', 'december', 'deer', 'delhi', 'dollar', 'duck', 'febuary', 'friday', 'fruits', 'glass', 'grapes', 'gujrat', 'hello',

'hindu', 'hyderabad', 'india', 'january', 'jesus', 'job', 'july', 'july', 'karnataka', 'kerala', 'krishna', 'litre', 'mango',

'may', 'mile', 'monday', 'mumbai', 'museum', 'muslim', 'nagpur', 'october', 'orange', 'pakistan', 'pass', 'police station',

'post office', 'pune', 'punjab', 'rajasthan', 'ram', 'restaurant', 'saturday', 'september', 'shop', 'sleep', 'southafrica',

'story', 'sunday', 'tamil nadu', 'temperature', 'temple', 'thursday', 'toilet', 'tomato', 'town', 'tuesday', 'usa', 'village',

'voice', 'wednesday', 'weight'] #gifs files avaiilabale

arr=['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r',

's','t','u','v','w','x','y','z'] #letters files availabe

with sr.Microphone() as source:

r.adjust\_for\_ambient\_noise(source) #filtering out the voice

i=0

while True:

print('Say something')

audio = r.listen(source)

# recognize speech using Sphinx

try:

aa=r.recognize\_google(audio)

print(aa) #this could be in any language

b=translator.translate(aa)

a=(b.text) #tanslated statement

print("you said " + a.lower()) #translated statement will be printed

for c in string.punctuation:

a= a.replace(c,"") #remove punctuations

if(a.lower()=='goodbye'):

print("oops!Time To say good bye")

break

elif(a.lower() in isl\_gif):

class ImageLabel(tk.Label):

"""a label that displays images, and plays them if they are gifs"""

def load(self, im): #this for subtitles

if isinstance(im, str):

im = Image.open(im)

self.loc = 0

self.frames = []

try:

for i in count(1):

self.frames.append(ImageTk.PhotoImage(im.copy()))

im.seek(i)

except EOFError:

pass

try:

self.delay = im.info['duration']

except:

self.delay = 100

if len(self.frames) == 1:

self.config(image=self.frames[0])

else:

self.next\_frame()

def unload(self):

self.config(image=None)

self.frames = None

def next\_frame(self):

if self.frames:

self.loc += 1

self.loc %= len(self.frames)

self.config(image=self.frames[self.loc])

self.after(self.delay, self.next\_frame)

root = tk.Tk()

lbl = ImageLabel(root)

lbl.pack() #does the work of adding the image or video onto the window for displaying

lbl.load(r'/home/manas/project/ISL\_Gifs/{0}.gif'.format(a.lower()))

root.mainloop() #root.mainloop is for lopping the gif #this is used for playing the gifs and displaying the subtitles

else:

for i in range(len(a)): #now for displaying letters

#a[i]=a[i].lower()

if(a[i] in arr):

ImageAddress = 'letters/'+a[i]+'.jpg'

ImageItself = Image.open(ImageAddress)

ImageNumpyFormat = np.asarray(ImageItself)

plt.imshow(ImageNumpyFormat) #since we are using matplot we use the function plt(plot) to plot the image properly

plt.draw()

plt.pause(0.8) # pause how many seconds

#plt.close()

else:

continue

except:

print("Could not listen")

plt.close()

#func()

while 1:

image = "signlang.png"

msg="VOICE TRANSLATOR ASSISTANT- A Project By Manas GOkhale ,Yogita Khosla , Nidhi Kantekar"

choices = ["Live Voice","All Done!"]

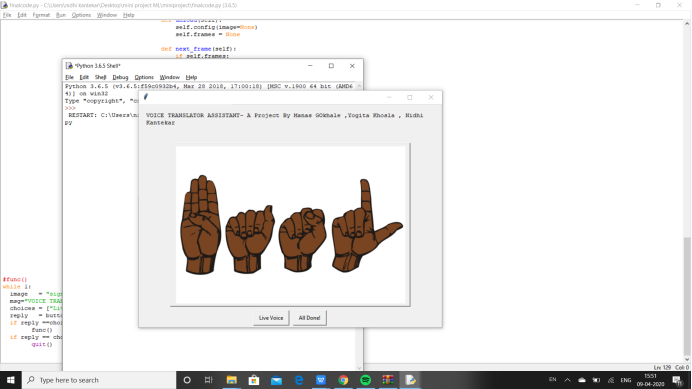
reply = buttonbox(msg,image=image,choices=choices)

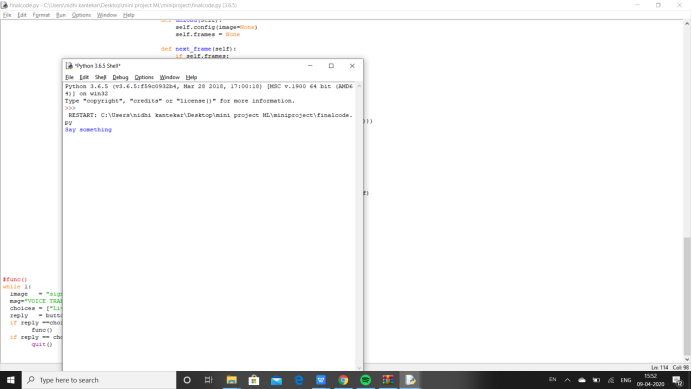
if reply ==choices[0]:

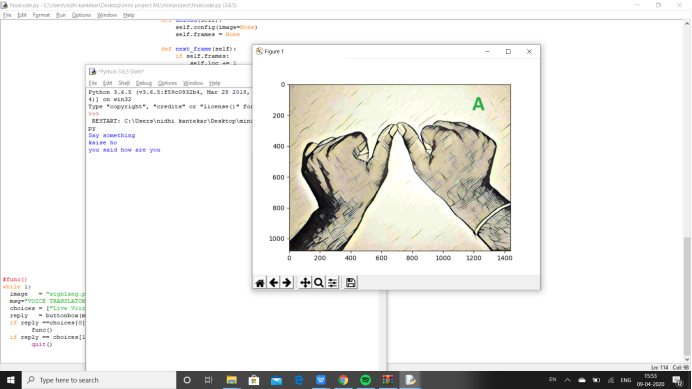
func()

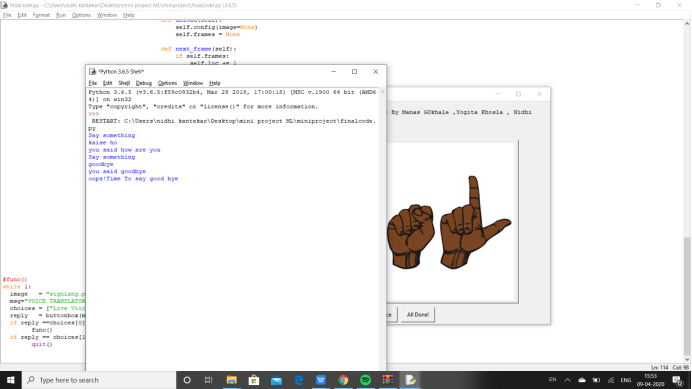
if reply == choices[1]:

quit()









Result video:

<https://drive.google.com/drive/folders/1Umlsk3GtoLkfOn5L35JBGklGjX8a0P74?usp=sharing>

(Please use SAKEC ID)

**Chapter 6**

**Result & Analysis**

Our approach to the use of lightweight videos improves overall efficiency by pre-processing the input text using NLP techniques and using the appropriate word from the input sentence. In addition, we've used visual representation for the entire input sentence and not for a single word that is represented in many of the previous projects.

This means that during live interaction our device becomes more stable and efficient. Videos and Gifs give the performance a realistic appeal. This method is readily accessible and requires less advanced knowledge of sign language.

We have observed that small phrases that have been pre defined into the system and are identified instantly. Sentences that are not part of the predefined data are displayed letter by letter and take time to be communicated. This can be improved by extending the amount of pre fedefined phrases.

**Chapter 7**

**7.1 Conclusion and Future Scope**

We have tried in this project to create a system that would be beneficial to people with

disabilities having communication problems by designing a system that would allow them to express themselves clearly and easily.

Our model successfully converts the entire input sentence into a single visual instead of

depicting different words by means of a picture which gives the model a very realistic and lively appeal.

We have made sure that the user doesnt have to keep changing applications to translate foreign language to engllish and further to sign language by combining and making this an effective one stop system that is easy to use .

Much more development on this track can be done as the ISL dictionary is still small and needs to grow eventually. In the Future we can add support for the Sign Languages of other countries and making it more portable by mitiating it to web and application based platforms. We can also add the functonality of making the system offline which can make the app suitable for rural areas.

The Speech to Sign Language Translator can bring a huge revolving change in the lives of many.

It’s has gone through the process of trail and tests to obtain a smooth working model which can be implemented for socials causes and for any other necessary occasions.

Implemented and tested successfully

Thus we have sucessfully implemented a Global to ISL translator.

**7.2 Appendix**

**7.3 References**

**7.4 Acknowledgements**