

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

TEAM ID : PNT2022TMID44713

TEAM LEADER : SREENITHI S (732519104027)

TEAM MEMBERS : PARIMALA B (732519104019)
SINDHUJA A (732519104026)
SRIDHARAN S (732519104028)

DEPARTMENT : CSE

**COLLEGE NAME : SHREE VENKATESHWARA
HI-TECH ENGINEERING
COLLEGE, GOBI.**

CONTENTS

1. INTRODUCTION

1.1. Project Overview

1.2. Purpose

2. LITERATURE SURVEY

2.1. Existing problem

2.2. References

2.3. Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

3.1. Empathy Map Canvas

3.2. Ideation & Brainstorming

3.3. Proposed Solution

3.4. Problem Solution fit

4. REQUIREMENT ANALYSIS

4.1. Functional requirement

4.2. Non-Functional requirements

5. PROJECT DESIGN

5.1. Data Flow Diagrams

5.2. Solution & Technical Architecture

5.3. User Stories

6. PROJECT PLANNING & SCHEDULING

6.1. Sprint Planning & Estimation

6.2. Sprint Delivery Schedule

6.3. Reports from JIRA

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1. Feature 1

7.2. Feature 2

7.3. Database Schema (if Applicable)

8. TESTING

8.1. Test Cases

8.2. User Acceptance Testing

9. RESULTS

9.1. Performance Metrics

9.2. Output

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Link.

1. INTRODUCTION

1.1 . PROJECT OVERVIEW

In our society, we have people with disabilities. The technology is developing day by day but no significant developments are undertaken for the betterment of these people. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language. In emergency times conveying their message is very difficult. The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language.

1. 2 . PURPOSE

The project aims to develop a system that converts the sign language into a human hearing voice in the desired language to convey a message to normal people, as well as convert speech into understandable sign language for the deaf and dumb. We are making use of a convolution neural network to create a model that is trained on different hand gestures. An app is built which uses this model. This app enables deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech is given as output.

2. LITERATURE SURVEY

2.1. EXISTING SOLUTION

In our society, we have people with disabilities .The technology is developing day by day but no significant developments are undertaken for the betterment of these people. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people are not trained on hand sign language .In emergency times conveying their message is very difficult.

The existing system, dealt with similar problems has some drawbacks and used video devices such as camera for processing data. But the key limitations of the existing system are lack of feasibility, portability, and accuracy. Further, in existing system author used contact sensors ,image processing are used which makes the project were limited number of letters can be implemented, unable to display or recognize the sentences, special gestures/patterns are not included, and help message is not implemented.

2.2. REFERENCE

S. ZHAO, M. WANG, Z. WEI, 2013, “A NEW TYPE OF DEAF-MUTE SIGN LANGUAGE RECOGNITION SYSTEM BASED ON THE MOBILE COMMUNICATION PLATFORM AND TERMINAL EQUIPMENT” ADVANCED MATERIALS RESEARCH.

GOSLING, JAMES; JOY, BILL;STEELE,GUY;AND BRACHA, GILAD."THE JAVA LANGUAGE SPECIFICATION" 2ND EDITION.

M. MOHANDES, M. DERICHE AND J. LIU, "IMAGE-BASED AND SENSOR-BASED APPROACHES TO ARABIC SIGN LANGUAGE RECOGNITION," IN IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS, AUG. 2014.

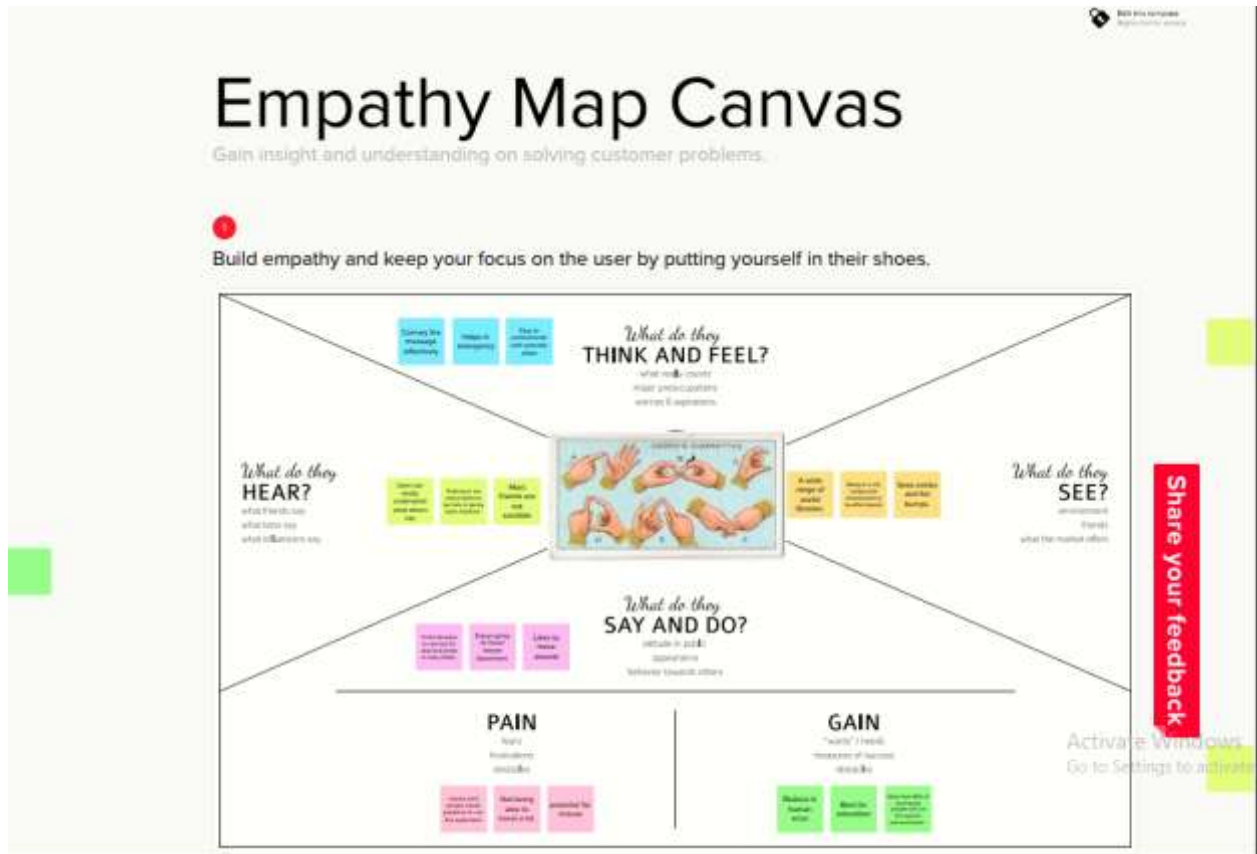
2.3. PROBLEM STATEMENT DEFINITION

In our society, we have people with disabilities. The technology is developing day by day but no significant developments are undertaken for the betterment of these people. Communications between deaf-mute and a normal person has always been challenging a task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language. In emergency times conveying their message is very difficult.

The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language.

3.IDEATION & PROPOSED SOLUTION

3.1. EMPATHY MAP CANVAS



3.2. IDEATION & BRAINSTORMING



3.3.PROPOSED SOLUTION

The portable device is the transduction system and consists of a glove comprising 5 flex sensors connected to a microcontroller through front-end electronics. The sensors are positioned in the glove hence the finger joints are in the middle of the sensitive area this way the sensor covers the finger joint area and follows its movement. Doing sign language our finger will bend, each bending of finger the sensor send signal to our controller. Our controller match the sensor signals with train signals. Recorded audio sound will playback for each sign. Opposite person easily communicate to deaf a keypad and display is used. Hearing impaired users can use text descriptions and icons. This module makes the device handy for the vocally disabled as it enables them to vocalise words by typing it on the screen.

Novelty

There are few mobile applications for Deaf and dumb like Deaf and Dumb through 4G applications. These techniques only enable communication between deaf and dumb through sign language using mobile phones. The mobile application which proposed in helps to make recognition of sign language. Mobile-based Deaf and Dumb Interaction System project in proposed mobile application that enables the needs of deaf and dumb developing a voice-activated mobile which would convert their sign language into messages that may be read by other users, this message can also converted to a voice

Feasibility Of Idea

Without dialling number we can communicate to other like face to face communication. It does not require large amount of storage as it uses the Hand speak support through online. The sign words are signed in the same order as letters appear in English alphabets. This project prepares individuals to work

as interpreter/translators facilitating and mediating communication between Deaf/Dumb and hearing people.

Business Model

The technology used are, Motion gesture, Touchless, Infrared Array, Ultrasonic Technology, 2D camera based technology, 3D vision technology. Lipreading recognition for people with a hearing impairment, Real-time captioning or translations for people with a hearing impairment or even people who don't speak the language. The system contains Real time functioning, Portable, Doesn't damage through use. It provides all the data needed more accurately as it also provides fingers movement data. Most deaf people use a combination of sign language, lip-reading and written communication to go about their daily lives. By using camera is that it removes that needs of sensors gloves and reduce cost from building the system.

Social Impact

Deaf and dumb gesture recognition system is an inexpensive device. It is also very efficient device when compared to the existing devices. The main objective of this project is to achieve communication of deaf-mute people like a normal person. It allows deaf and dumb people to communicate with others. It is a best device for these people to overcome their disability. They can express their views to others.

Scalability Of Solution

GTTS - It is a Python library and CLI tool to interface with Google Translates textto speech API. Writes spoken mp3 data to a file, a file-like object (byte string) for further audio manipulation, or studio. Implementation on Android platform for smart phones and tablets PCs. Read and write images and detection of images for extracting its features. Detection of shapes like

Circle, rectangle etc. in an image, Detection of coin in images. Image processing works with text recognition in images. e.g. Reading Number Plates, Modifying image quality and colour. This describes a project target audience and its user interface, hardware and software requirements. It defines how the client, team and audience see the project and its functionality.

3.4. PROBLEM SOLUTION FIT

Hand gesture recognition system is widely used technology for helping the deaf and dumb people. Human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. We aim for developing an deaf and dumb gesture recognize system for establishing communication between the deaf and the dumb people. Gestures are considered as the most natural expressive way for communications between human and computers in virtual system. Hand gestures which can represent ideas using unique shapes and finger orientation have a scope for human machine interaction.

4. REQUIREMENT ANALYSIS

4.1. FUNCTIONAL REQUIREMENT

Operating System : Windows, Mac, Linux, Android

SDK : OpenCV TensorFlow, Keros, Numpy

4.2. NON FUNCTIONAL REQUIREMENT

Camera : Good quality,3MP

Ram : Minimum 8GB or higher

GPU : 4GB dedicated

Processor : Intel Pentium 4 or higher

HDD : 10 GB or higher

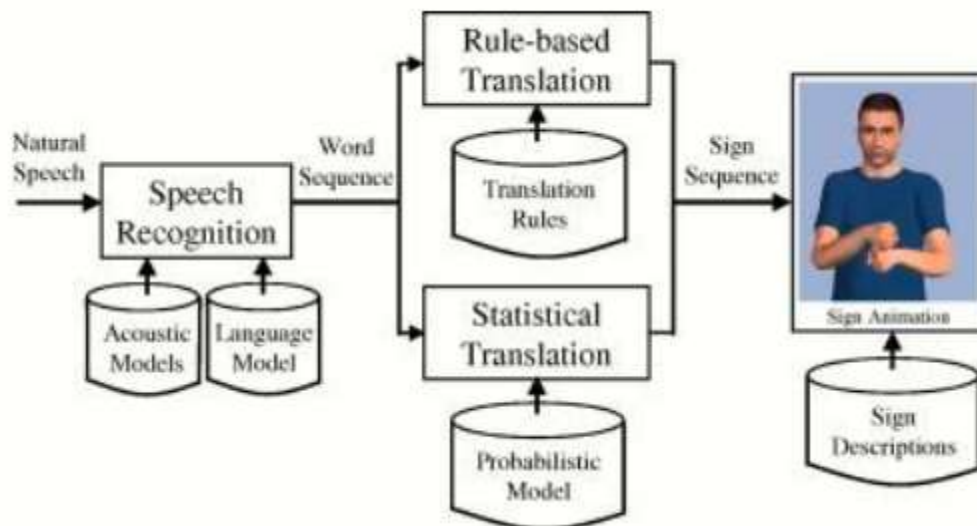
Monitor : 15" or 17" color monitor

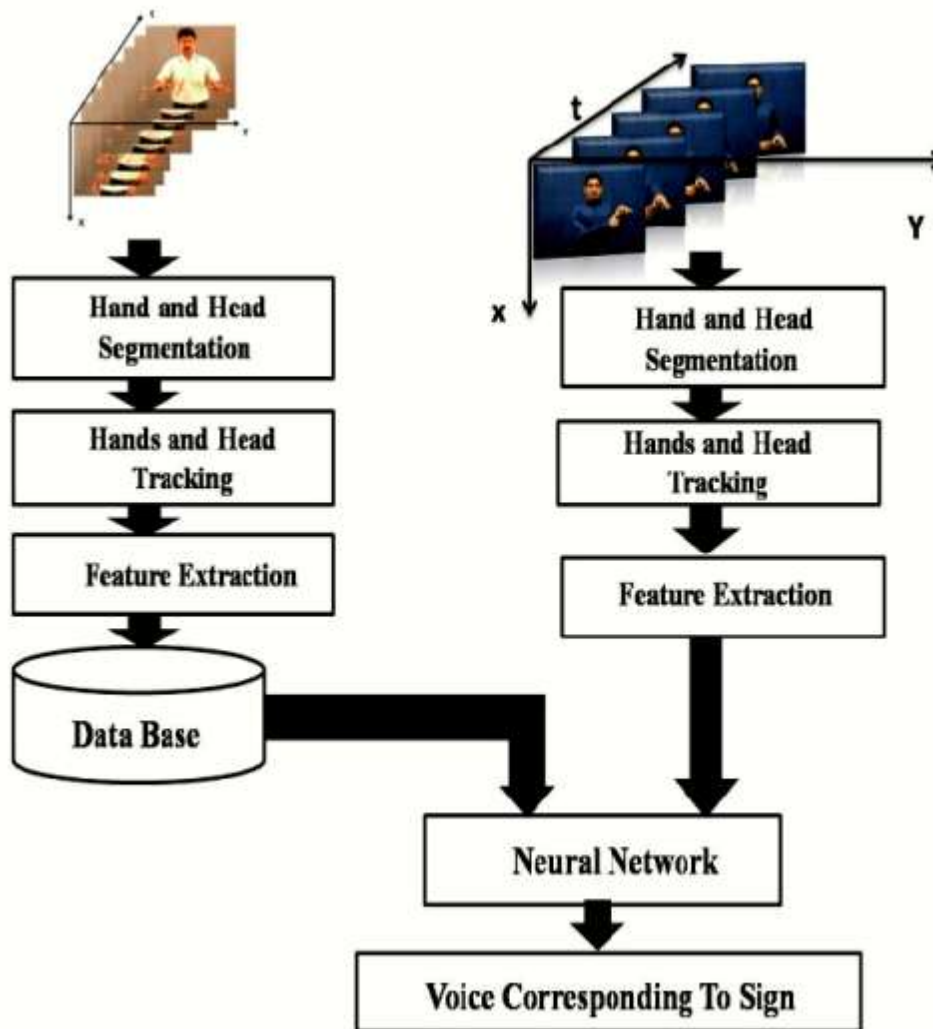
Mouse : Scroll or Optical Mouse or Touch Pad

Keyboard : Standard 110 keys keyboard

5. PROJECT DESIGN

5.1. DATA FLOW DIAGRAMS

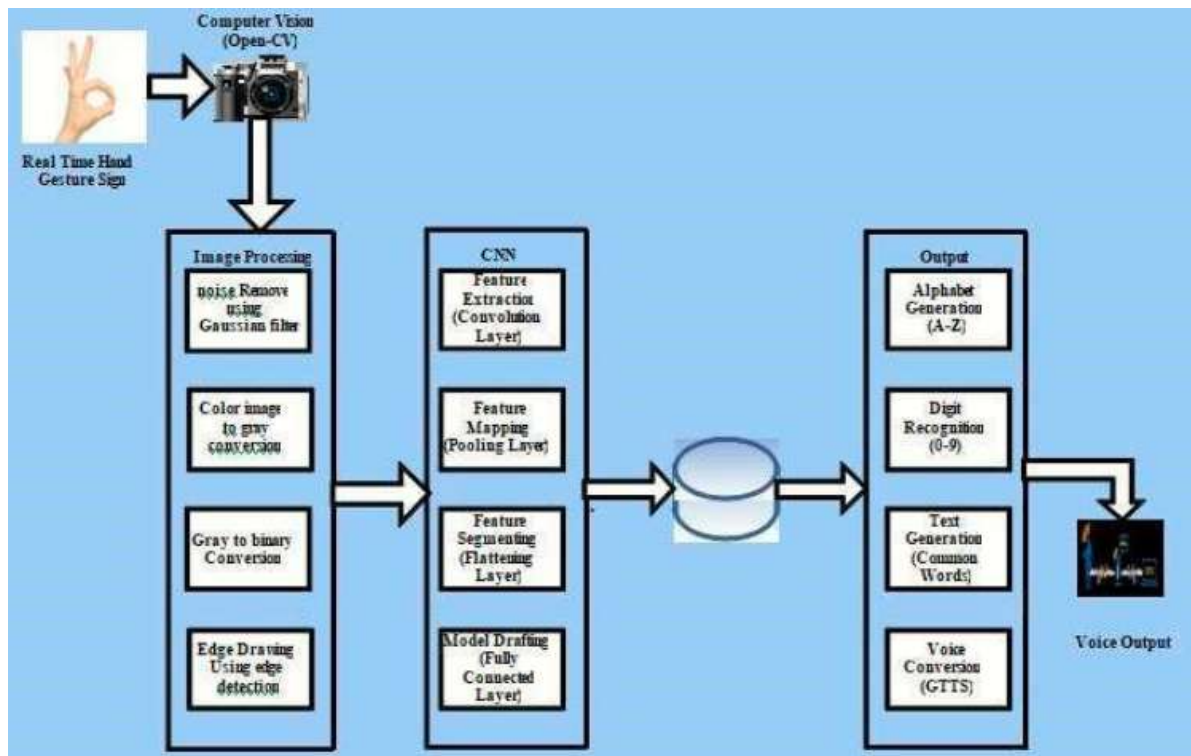




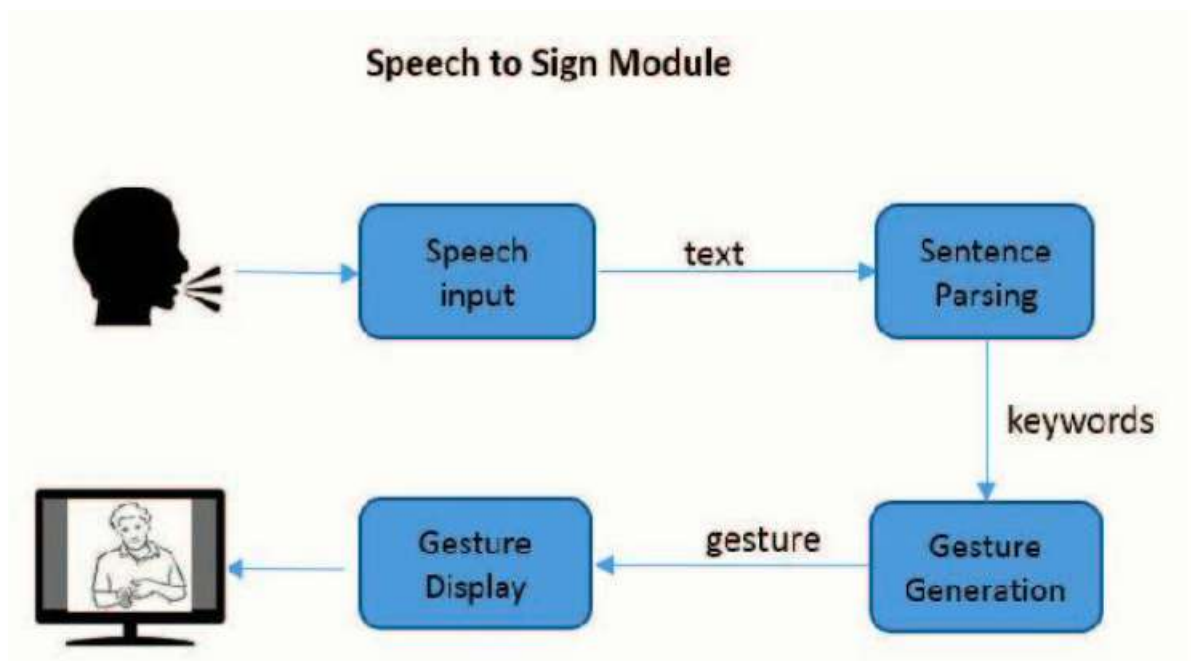
5.2. SOLUTON & TECHNICAL ARCHITECTURE

SOLUTION ARCHITECTURE

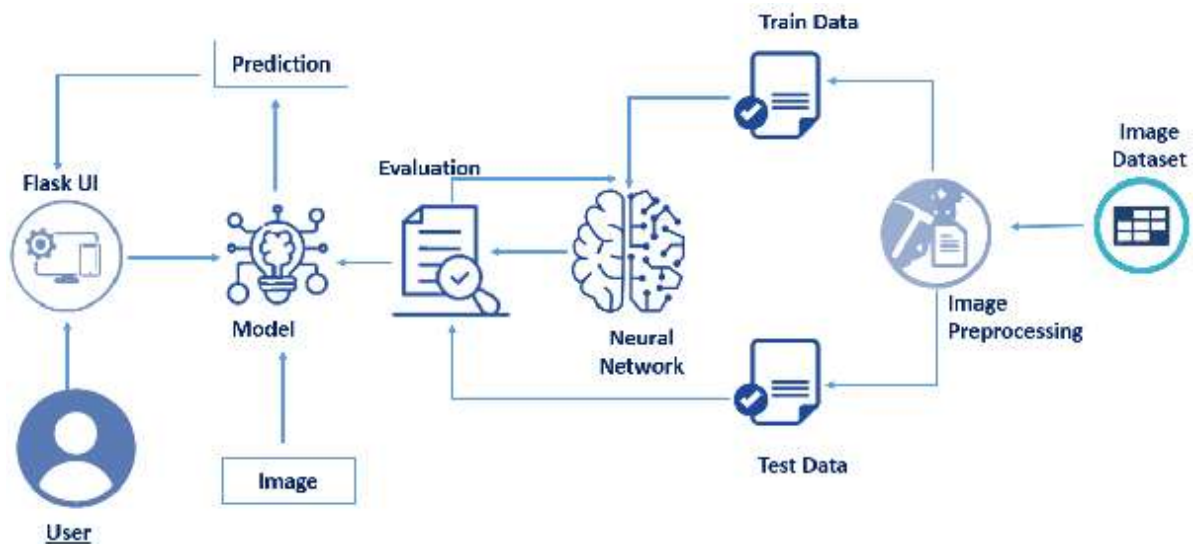
SIGN TO VOICE OUTPUT



SPEECH TO SIGN MODULE



TECHNICAL ARCHITECTURE



5.3. USER STORIES

1. As a User, I can register for the application by entering my email, Password and confirming, my password.
2. As a User, I will receive confirmation email once I have registered for the application.
3. As a user, I can enter the web page ones clicked, which provides be The Guidelines to use the app.
4. As a User , I can give through a read the guidelines to understand the performance and functioning of the app.
5. As a User, I can click the button Convert sign , which directs me towards the Main screen.

6. As a User, I can show my hand sign towards the camera which converts them into text manner.
7. Once the text is obtained, As a User I can click on the voice mode which provides the text in the form of speech.
8. As an Executive, I can provide the Specifications of Camera required, and other factors that are required for smooth functioning of the app.
9. As an Executive, I can check the usage and queries obtained from the end users or customers.
10. As an Admin, I can take the queries from the customer care and perform the testing phase again ,loading the other signs in the dataset, in order to make the customers to use the app effectively.

6. PROJECT PLANNING & SCHEDULING

6.1.SPRINT DELIVERY SCHEDULE

SPRINT	FUNCTIONAL REQUIREMENT	USER STORY NUMBER	USER STORY/ TASK	STORY POINTS	PRIORITY	TEAM MEMBERS
Sprint-1	Registration	USN-1	As a User ,I can register for the application by entering my email, password and confirming my password.	2	High	S.Sreenithi

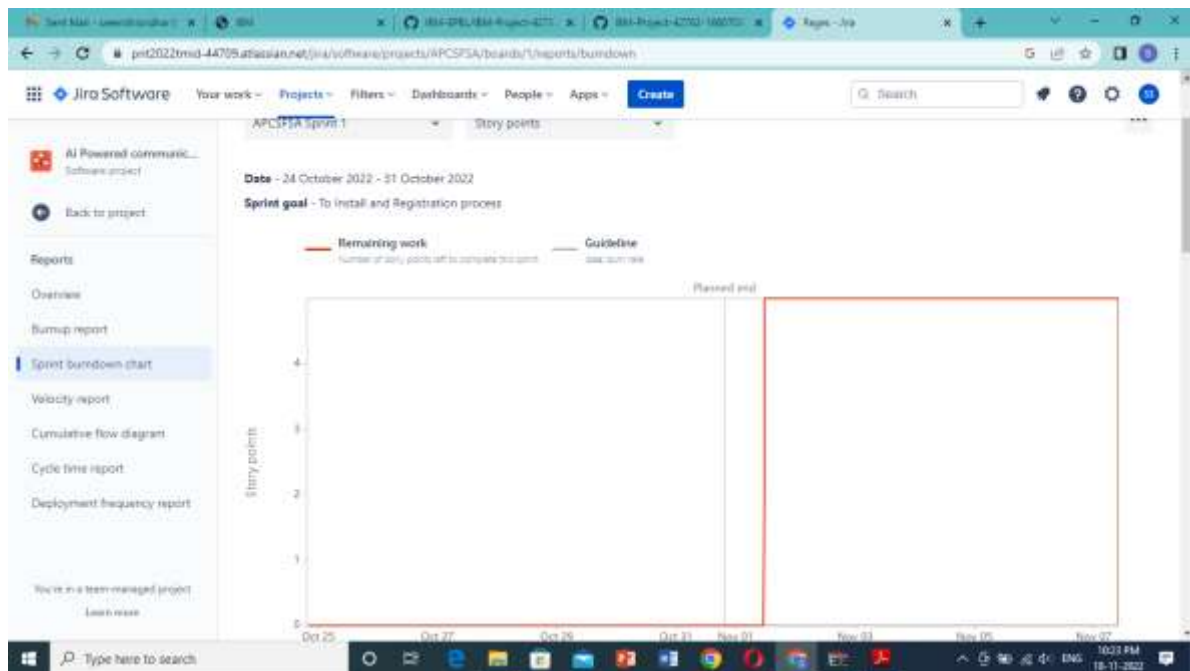
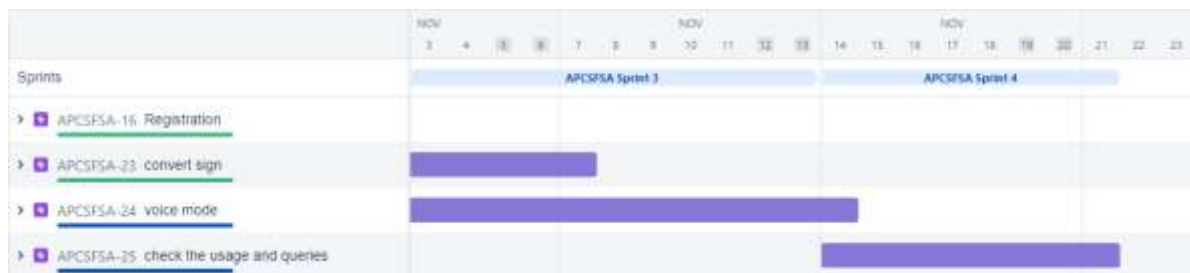
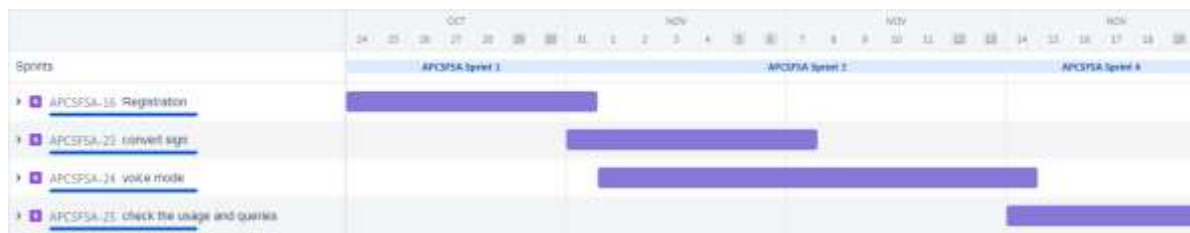
Sprint-1	Login	USN-2	As a User, I will receive confirmation email once I have registered for the application.	1	High	B.Parimala
Sprint-1	Main page	USN-4	As a User, I can enter the web page once clicked, which provides be the Guidelines to use the app	1	Medium	A.Sindhuja
Sprint-1	Guidelines	USN-5	As a User , I can give a read through the guidelines to understand the performance and functioning of the app.	1	Medium	B.Sridharan
Sprint-2	Covert sign	USN-6	As a User, I can click the button Convert sign , which directs me towards the Main screen	4	Medium	S.Sreenithi
Sprint-2	Camera(Hand movement detection)	USN-7	As a User, I can show my hand sign towards the camera which converts them into text manner.	8	High	B.Parimala
Sprint-3	Voice mode	USN-8	Once the text is obtained, As a User I can click on the voice mode which provides the text in the form of speech.	3	High	A.Sindhuja
Sprint-1	Provide the necessary functionalities required to use the app.		As an Executive, I can provide the Specifications of Camera required, and other factors that are required for smooth functioning of the app.	1	Low	B.Sridharan

Sprint-4	Check the performance of the app		As an Executive, I can check the usage and queries obtained from the end users or customers.	1	Medium	S.Sreenithi
Sprint-4	Receive queries based on the usage.		As an Admin, I can take the queries from the customer care and perform the testing phase again , loading the other signs in the dataset, in order to make the customers to use the app effectively.	2	High	B.Parimala

6.2. SPRINT DELIVERY SCHEDULE

SPRINT	TOTAL STORY POINTS	DURATION	SPRINT START DATE	SPRINT END DATE(PLANNED)	STORY POINTS COMPLETED(AS ON PLANNED END DATE)	STORY RELEASE DATE(ACTUAL)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	23	29 Oct 2022
Sprint-2	20	12 Days	31 Oct 2022	5 Nov 2022	26	5 Nov 2022
Sprint-3	20	3 Days	7 Nov 2022	12 Nov 2022	11	12 Nov 2022
Sprint-4	20	4 Days	14 Nov 2022	19 Nov 2022	18	19 Nov 2022

6.3. REPORTS FROM JIRA



7. CODING & SOLUTIONING

7.1. FEATURE 1

Good segmentation process leads to perfect features extraction process and the later play an important role in a successful recognition process. Interesting points on every object which can be extracted to provide a "feature" description of the object. Features vector of these image can be extracted in different ways according to particular application. Under different scene conditions, the performance of different feature detectors will be significantly different.

Feature extraction is a method of reducing data dimensionality by encoding related information in a compressed representation and removing less discriminative data. The selection of which features to deal with and the extraction method are probably the most significant design decisions in hand motion and gesture recognition development.

App.py

```
# -*- coding: utf-8 -*-
```

```
"""
```

Created on Sat Oct 24 00:48:19 2020

```
@author: Tulasi
```

```
"""
```

```
# USAGE
```

```
# import the necessary packages
```

```
from flask import Flask,render_template,request
```

```
# Flask-It is our framework which we are going to use to run/serve our application.
```

```

#request-for accessing file which was uploaded by the user on our application.

import cv2 # opencv library

from tensorflow.keras.models import load_model#to load our trained model

import numpy as np

from gtts import gTTS #to convert text to speech

from skimage.transform import resize

import os

from keras.preprocessing import image

from playsound import playsound

'''

def playaudio(text):

    speech=gTTS(text)

    print(type(speech))

    speech.save("output1.mp3")

    playsound("output1.mp3")

    return

'''

app = Flask(__name__,template_folder="templates") # initializing a flask app

# Loading the model

model=load_model('aslpng1.h5')

print("Loaded model from disk")

vals = ['A', 'B','C','D','E','F','G','H','I']

#app=Flask(__name__,template_folder="templates")

@app.route('/', methods=['GET'])

```

```

def index():
    return render_template('home.html')

@app.route('/home', methods=['GET'])
def home():
    return render_template('home.html')

@app.route('/upload', methods=['GET', 'POST'])
def predict():

    # Get a reference to webcam #0 (the default one)

    print("[INFO] starting video stream...")

    vs = cv2.VideoCapture(0)

    #writer = None

    (W, H) = (None, None)

    # loop over frames from the video file stream
    while True:

        # read the next frame from the file

        (grabbed, frame) = vs.read()

        # if the frame was not grabbed, then we have reached the end
        # of the stream

        if not grabbed:

            break

```

```

# if the frame dimensions are empty, grab them

if W is None or H is None:

    (H, W) = frame.shape[:2]


# clone the output frame, then convert it from BGR to RGB

# ordering and resize the frame to a fixed 64x64

output = frame.copy()

#print("apple")

img = resize(frame,(64,64,1))

img = np.expand_dims(img,axis=0)

if(np.max(img)>1):

    img = img/255.0

result = np.argmax(model.predict(img), axis=-1)

index=['A', 'B','C','D','E','F','G','H','I']

result=str(index[result[0]])

#print(result)

#result=result.tolist()


cv2.putText(output, "It indicates: {}".format(result), (10, 120),
cv2.FONT_HERSHEY_PLAIN,

            2, (0,255,255), 1)


#converts text to speech and plays the audio

speech = gTTS(text = result, lang = 'en', slow = False)

#speech=gTTS(text)

```

```

    print(type(speech))

    speech.save("text.mp3")

    os.system("start text.mp3")


    cv2.imshow("Output", output)

    key = cv2.waitKey(1) & 0xFF

    # if the `q` key was pressed, break from the loop
    if key == ord("q"):

        break


    # release the file pointers
    print("[INFO] cleaning up...")

    vs.release()

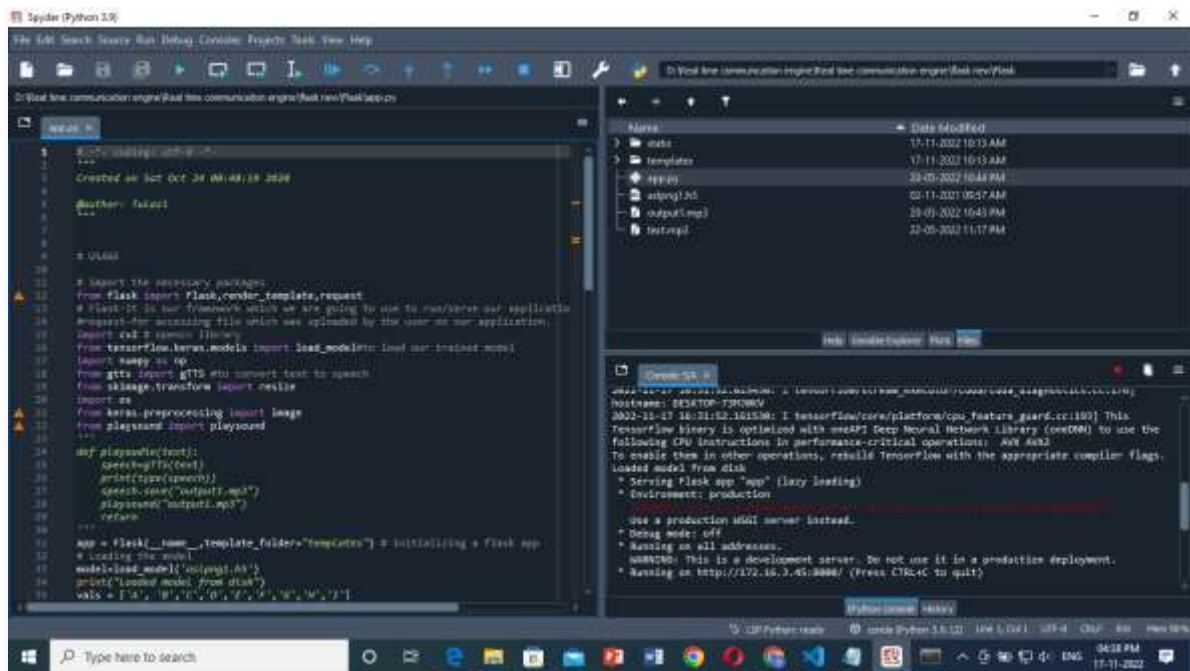
    cv2.destroyAllWindows()

    return render_template("upload.html")

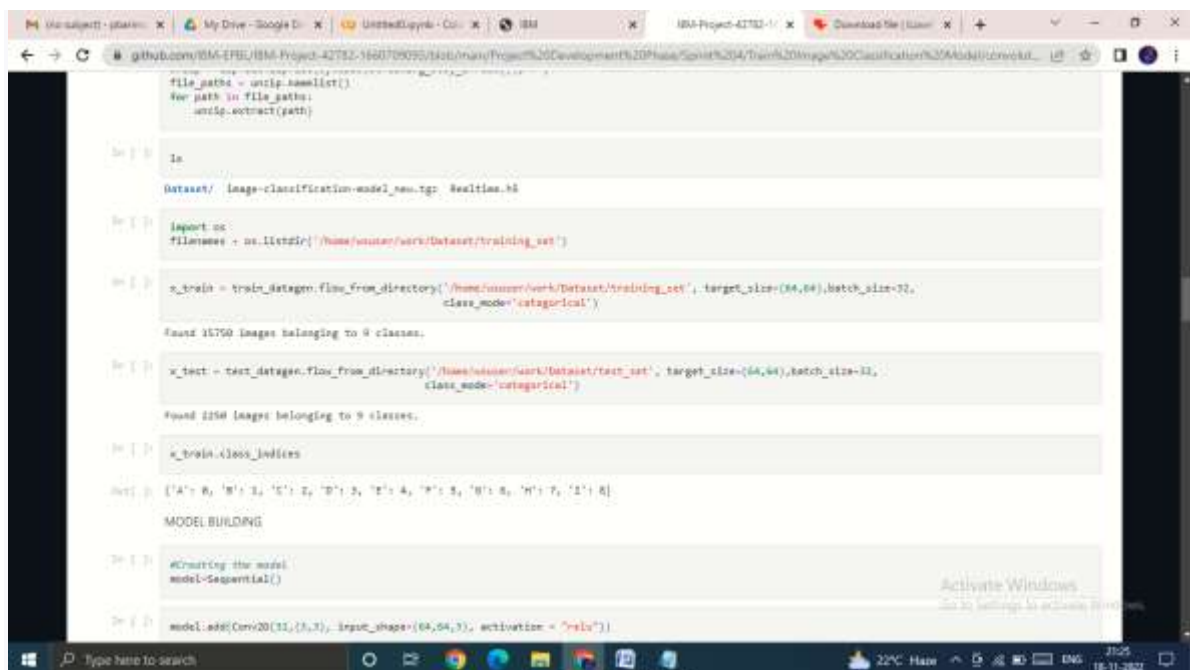

if __name__ == '__main__':

    app.run(host='0.0.0.0', port=8000, debug=False)

```

7.2.Feature 2



```
MODEL BUILDING

In [1]: #Creating the model
model=Sequential()

In [2]: model.add(Conv2D(32,(3,3), input_shape=(64,64,3), activation = 'relu'))

In [3]: model.add(MaxPooling2D(pool_size=(2,2)))

In [4]: model.add(Flatten())

In [5]: #Adding hidden layers
model.add(Dense(units=128, kernel_initializer='random_uniform', activation='relu'))

In [6]: model.add(Dense(units=9, kernel_initializer='random_uniform', activation='relu'))

In [7]: model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

In [8]: model.fit_generator(x_train, steps_per_epoch = 47, epochs = 10, validation_data = x_test, validation_steps = 10)
#steps_per_epoch = no of train images/batch size
#validation_steps = no of test images/batch size

Epoch 1/10
/tmp/tmpnwr7pykrrw/64/4170007720.py:11: UserWarning: "Model.fit_generator" is deprecated and will be removed in a future version. Please use "Model.fit", which supports generators.
  model.fit_generator(x_train, steps_per_epoch = 47, epochs = 10, validation_data = x_test, validation_steps = 10)
67/47 [=====] - 3s 156s/step - loss: 2.1753 - accuracy: 0.8488 - val_loss: 1.2784 - val_accuracy: 0.8188
Epoch 1/10
```

```
Import the library

In [1]: from keras.preprocessing.image import ImageDataGenerator

Arguments for ImageDataGenerator class

In [2]: train_datagen=ImageDataGenerator(rescale=1./255, shear_range=0.1, zoom_range=0.1, horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)

Applying ImageDataGenerator functionality to trainset and testset

In [3]: x_train=train_datagen.flow_from_directory('/content/drive/MyDrive/dataset/data_set/training_set', target_size=(64,64), batch_size=32, class_mode='categorical')
Found 5750 images belonging to 9 classes.

In [4]: x_test=test_datagen.flow_from_directory('/content/drive/MyDrive/dataset/data_set/test_set', target_size=(64,64), batch_size=32, class_mode='categorical')
Found 2250 images belonging to 9 classes.

!pip install opencv.python

In [5]: import cv2

 imread is used to read the image

In [6]: img=cv2.imread('/content/drive/MyDrive/dataset/data_set/test_set/A/10.png')
```

```

In [ ]: ['A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8]

MODEL BUILDING

In [ ]:
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Conv2D
from tensorflow.keras.layers import Conv2D, MaxPooling2D
from keras.layers import Dropout
from keras.layers import Flatten

In [ ]:
#creating the model
model=Sequential()
#adding the layers
model.add(Convolution2D(32, (3,3), input_shape=(64,64,1), activation = 'relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())

#adding hidden layers
model.add(Dense(200, activation='relu'))
model.add(Dense(200, activation='relu'))
model.add(Dense(200, activation='relu'))

#adding the output layer
model.add(Dense(1, activation='softmax'))

In [ ]:
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

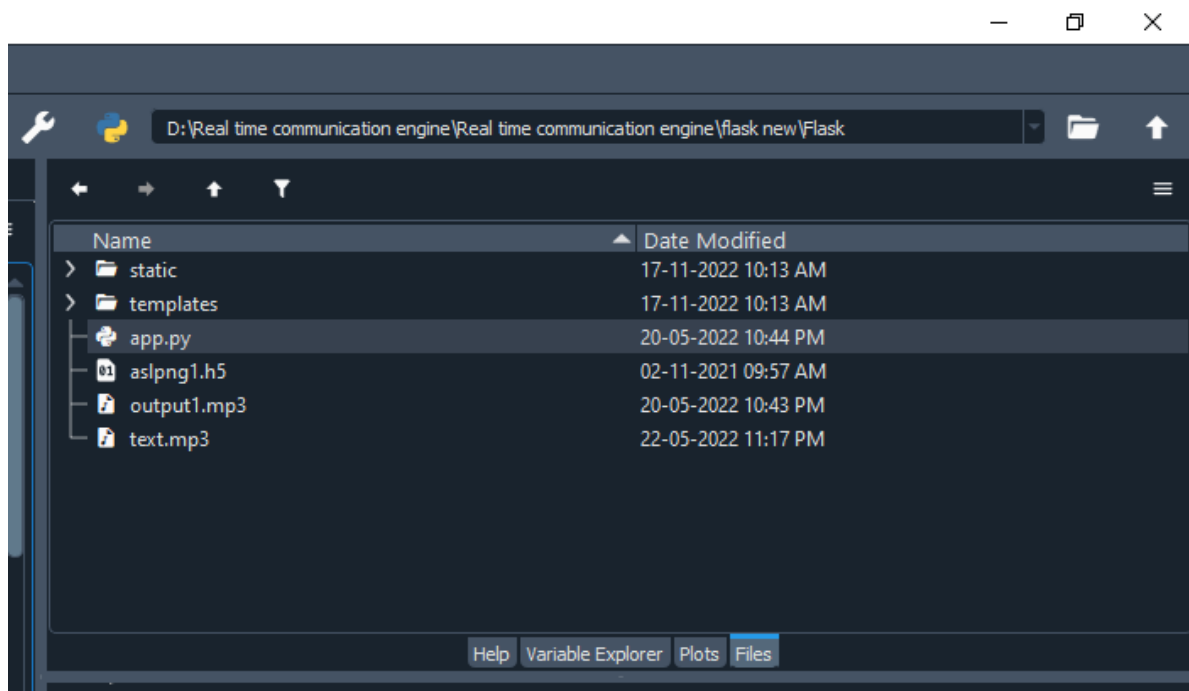
In [15]:
model.fit_generator(x_train, steps_per_epoch=30, epochs=10, validation_data=x_test, validation_steps=50)

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: "model.fit_generator" is deprecated and will be removed in a future version. Please use "model.fit", which supports generators.
***Stack trace for launching on IPython kernel.***
Epoch 1/10

```

8. TESTING

8.1. TEST CASE



8.2. USER ACCEPTANCE TESTING

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [Real Time Communication System powered by AI for specially abled] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	15	4	2	3	25
Duplicate	0	0	3	0	0
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	28	9	9	24	69

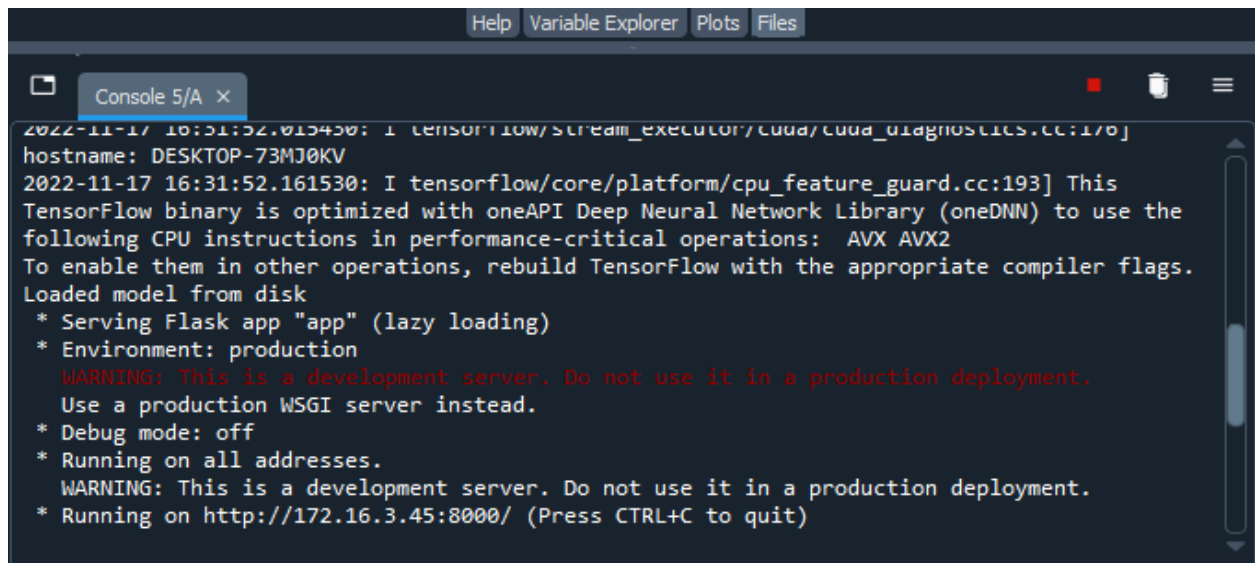
3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested.

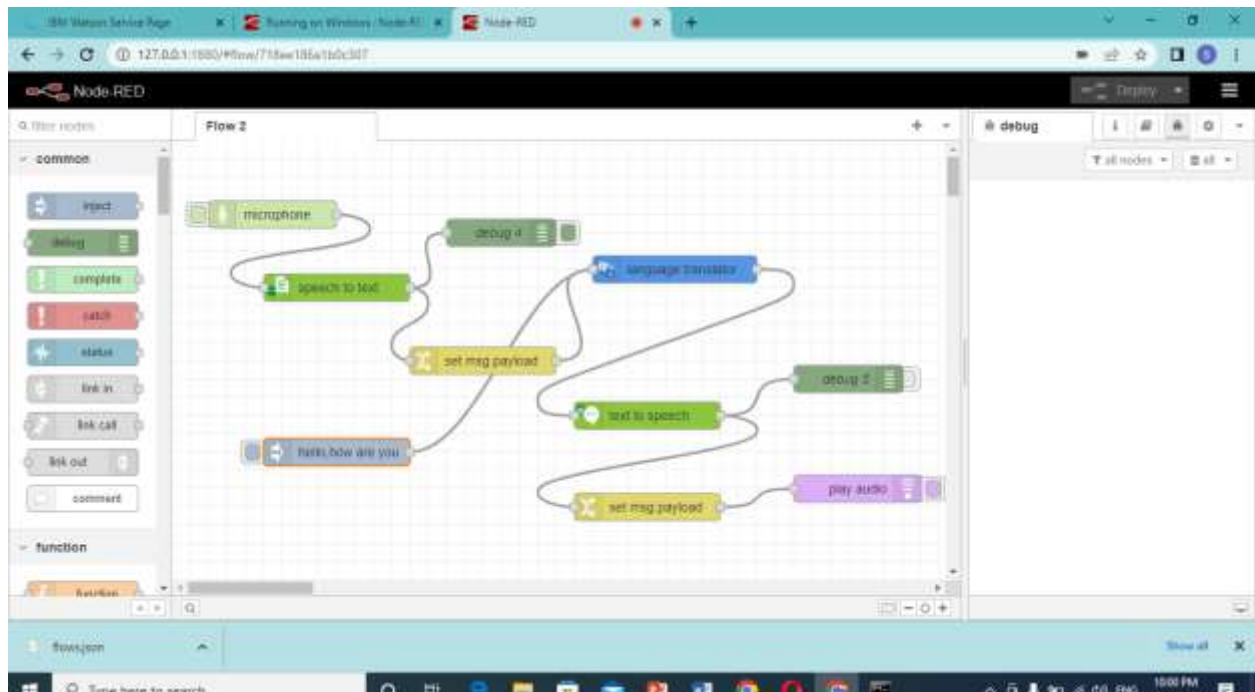
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	0	5
Client Application	15	0	0	15
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	15	0	0	15
Final Report Output	5	0	0	5
Version Control	2	0	0	2

9. RESULTS

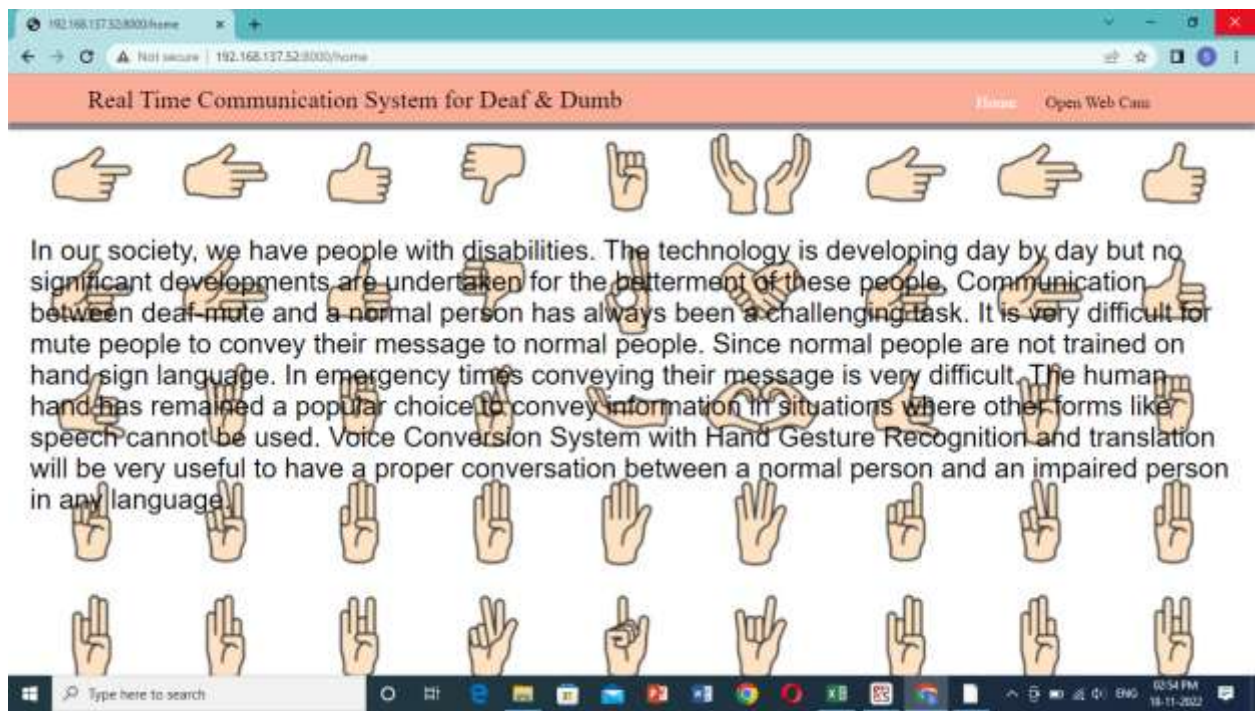
9.1. PERFORMANCE METRICS

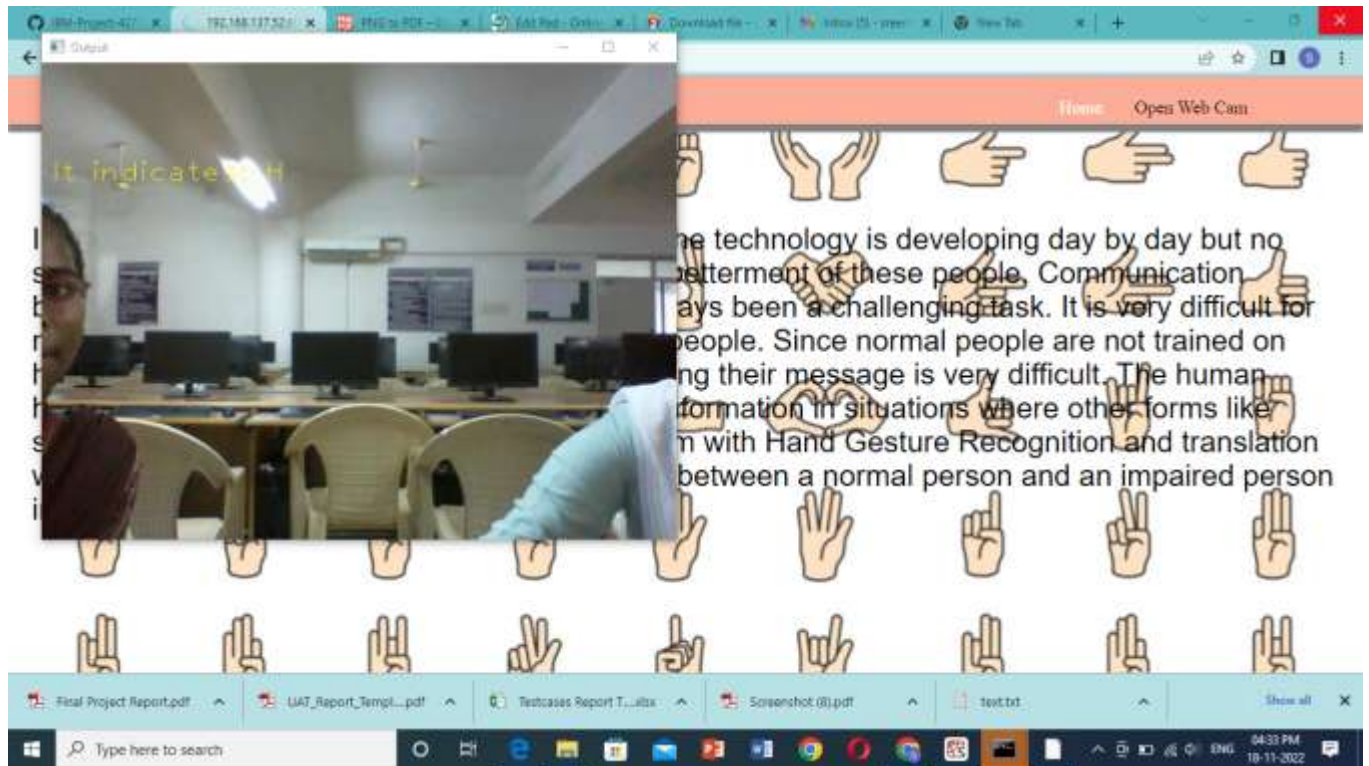


```
Help Variable Explorer Plots Files
Console 5/A x
2022-11-17 16:31:52.015450: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176]
hostname: DESKTOP-73MJ0KV
2022-11-17 16:31:52.161530: I tensorflow/core/platform/cpu_feature_guard.cc:193] This
TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the
following CPU instructions in performance-critical operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Loaded model from disk
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on all addresses.
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://172.16.3.45:8000/ (Press CTRL+C to quit)
```



9.2. OUTPUTS







10. ADVANTAGES & DISADVANTAGES

ADVANTAGES :

- Low cost
- Compact system
- Flexible to users
- It takes less power to operate system
- They can live independently without others help

- Safety is one of the highest priorities for this translator
- The use of Artificial Intelligence is a boon for specially-abled people.
- In future, this technology will become less expensive and hopefully easier to use so that everyone can experience its benefits.
- It is usually a substitute of verbal based communication.

DISADVANTAGES :

- Gestures are difficult in understanding, informal etiquette, information might get distorted, etc.
- One cannot make long explanation or conversation through gesture.
- It is one of the informal types of communication, where it is not suited for official purpose.

11. CONCLUSION

The project aims to develop a system that converts the sign language into a human hearing voice in the desired language to convey a message to normal people, as well as convert speech into understandable sign language for the deaf and dumb. We are making use of a convolution neural network to create a model that is trained on different hand gestures. An app is built which uses this model. This app enables deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech is given as output.

Through this project we are trying to build a flexible system for the physically impaired people that will ease their life. An attempt to create a sign

language to text conversion wireless lightweight system ,through the use of information gestured by physically impaired person which can be effectively conveyed to a normal person .The system will to convert the sign language of the physically impaired person into text that can interpreted by their own genre and also to the rest of the world.

12. FUTURE SCOPE

Artificial intelligence is one of the most rapidly developing fields in the technology industry. AI is now used in a variety of fields, including healthcare, transportation, and security. Multiple sectors demand the knowledge of experienced AI specialists as a result of this increase.

The goal is to create computer intelligence programmes that can handle real-time problems and help organizations and everyday people achieve their goals. Machine games, speech recognition, language detection, computer vision, expert systems, robotics, and other fields have potential.

13.APPENDIX

Source code

App.py

```
# -*- coding: utf-8 -*-
```

```
"""
```

Created on Sat Oct 24 00:48:19 2020

@author: Tulasi

"""

USAGE

import the necessary packages

from flask import Flask,render_template,request

Flask-It is our framework which we are going to use to run/serve our application.

#request-for accessing file which was uploaded by the user on our application.

import cv2 # opencv library

from tensorflow.keras.models import load_model#to load our trained model

import numpy as np

from gtts import gTTS #to convert text to speech

from skimage.transform import resize

import os

```

from keras.preprocessing import image

from playsound import playsound

'''

def playaudio(text):

    speech=gTTS(text)

    print(type(speech))

    speech.save("output1.mp3")

    playsound("output1.mp3")

    return

'''

app = Flask(__name__,template_folder="templates") # initializing a flask app

# Loading the model

model=load_model('aslpng1.h5')

print("Loaded model from disk")

vals = ['A', 'B','C','D','E','F','G','H','I']

#app=Flask(__name__,template_folder="templates")

```

```

@app.route('/', methods=['GET'])

def index():

    return render_template('home.html')

@app.route('/home', methods=['GET'])

def home():

    return render_template('home.html')

@app.route('/upload', methods=['GET', 'POST'])

def predict():

    # Get a reference to webcam #0 (the default one)

    print("[INFO] starting video stream...")

    vs = cv2.VideoCapture(0)

    #writer = None

    (W, H) = (None, None)

    # loop over frames from the video file stream

```

```
while True:

    # read the next frame from the file

    (grabbed, frame) = vs.read()


    # if the frame was not grabbed, then we have reached the end

    # of the stream

    if not grabbed:

        break


    # if the frame dimensions are empty, grab them

    if W is None or H is None:

        (H, W) = frame.shape[:2]


    # clone the output frame, then convert it from BGR to RGB

    # ordering and resize the frame to a fixed 64x64

    output = frame.copy()

    #print("apple")
```

```

img = resize(frame,(64,64,1))

img = np.expand_dims(img,axis=0)

if(np.max(img)>1):

    img = img/255.0

result = np.argmax(model.predict(img), axis=-1)

index=['A', 'B','C','D','E','F','G','H','I']

result=str(index[result[0]])

#print(result)

#result=result.tolist()


cv2.putText(output, "It indicates: {}".format(result), (10, 120),
cv2.FONT_HERSHEY_PLAIN,

2, (0,255,255), 1)


#converts text to speech and plays the audio

speech = gTTS(text = result, lang = 'en', slow = False)

#speech=gTTS(text)

```

```
print(type(speech))

speech.save("text.mp3")

os.system("start text.mp3")


cv2.imshow("Output", output)

key = cv2.waitKey(1) & 0xFF


    # if the `q` key was pressed, break from the loop

    if key == ord("q"):

        break


# release the file pointers

print("[INFO] cleaning up...")

vs.release()

cv2.destroyAllWindows()

return render_template("upload.html")
```



```
if __name__ == '__main__':  
  
    app.run(host='0.0.0.0', port=8000, debug=False)
```

HOME.html

```
<html>  
  
<script>  
  
</script>  
  
<style>  
.header { position: relative;  
top:0;  
margin:0px;  
z-index: 1;  
left: 0px;  
right: 0px;  
position: fixed;  
background-color: #FCAD98 ;  
color: white;  
box-shadow: 0px 8px 2px grey;
```

```
overflow: hidden;
padding-left: 20px;
font-family: 'Josefin Sans';
font-size: 2vw;
width: 100%;
height: 8%;
text-align: center;
}

.topnav {
overflow: hidden;
background-color: #FCAD98;
}

.topnav-right a {
float: left;
color: black;
text-align: center;
padding: 14px 16px;
text-decoration: none;
font-size: 18px;
}

.topnav-right a:hover {
background-color: #FCAD98;
color: black;
```

```

}

.topnav-right a.active {
background-color: #FCAD98;
color: white;
}

.topnav-right {
float: right;
padding-right: 100px;
}

body {
background-image: -webkit-linear-gradient(90deg, skyblue 0%, steelblue 100%);
background-image: url("");
background-size: cover;
background-attachment: fixed;
background-size: 100% 100%;
background-color: ;
background-repeat: no-repeat;
background-size: cover;
background-position: 0px 0px;
}

.button {
background-color: #091425;
border: none;

```

```

color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 12px;
border-radius: 16px;
}

.button:hover {
box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
}

form {border: 3px solid #f1f1f1; margin-left:400px;margin-right:400px;}
input[type=text], input[type=password] {
width: 100%;
padding: 12px 20px;
display: inline-block;
margin-bottom:18px;
border: 1px solid #ccc;
box-sizing: border-box;
}

button {
background-color: #091425;
color: white;

```

```
padding: 14px 20px;
margin-bottom:10px;
border: none;
cursor: pointer;
width: 17%;
border-radius:4px;
font-family:Montserrat;
}

button:hover {
opacity: 0.8;
}

.cancelbtn {
width: auto;
padding: 10px 18px;
background-color: #f44336;
}

.imgcontainer {
text-align: center;
margin: 24px 0 12px 0;
}

img.avatar {
width: 30%;
border-radius: 50%;
```

```

}

.container {
padding: 16px;
}

span.psw {
float: right;
padding-top: 16px;
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
span.psw {
display: block;
float: none;
}

.cancelbtn {
width: 100%;
}
}

.home{
margin:80px;
width: 84%;
height: 500px;
padding-top:10px;

```

```
padding-left: 30px;

}

.login{
margin:80px;
box-sizing: content-box;
width: 84%;
height: 420px;
padding: 30px;
border: 10px solid blue;
}

.left,.right{
box-sizing: content-box;
height: 400px;
margin:20px;
border: 10px solid blue;
}

.mySlides {display: none;}
img {vertical-align: middle;}

/* Slideshow container */
.slideshow-container {
max-width: 1000px;
position: relative;
margin: auto;
```

```
}  
  
/* Caption text */  
.text {  
color: #f2f2f2;  
font-size: 15px;  
padding: 8px 12px;  
position: absolute;  
bottom: 8px;  
width: 100%;  
text-align: center;  
}  
  
/* The dots/bullets/indicators */  
.dot {  
height: 15px;  
width: 15px;  
margin: 0 2px;  
background-color: #bbb;  
border-radius: 50%;  
display: inline-block;  
transition: background-color 0.6s ease;  
}  
  
.active {  
background-color: #FCAD98;
```



```

}

/* Fading animation */

.fade {
  -webkit-animation-name: fade;
  -webkit-animation-duration: 1.5s;
  animation-name: fade;
  animation-duration: 1.5s;
}

@-webkit-keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

@keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

/* On smaller screens, decrease text size */

@media only screen and (max-width: 300px) {
  .text {font-size: 11px}
}

@import url('https://fonts.googleapis.com/css2?family=Poppins&display=swap');

* {

```

```
box-sizing: border-box;
}

body {
min-height: 100vh;
margin: 0;
color: #fff;
font-family: 'Poppins',sans-serif;
display: flex;
align-items: center;
justify-content: center;
background-color: #f5f5f5;
}

.container {
max-width: 1376px;
margin: auto;
padding: 2rem 1.5rem;
}

.cards {
display: flex;
flex-wrap: wrap;
align-items: center;
justify-content: center;
}
```

```
.card {  
  cursor: pointer;  
  background-color: transparent;  
  height: 300px;  
  perspective: 1000px;  
  margin: 1rem;  
  align-items: center;  
  justify-content: center;  
}  
  
.card h3 {  
  border-bottom: 1px #fff solid;  
  padding-bottom: 10px;  
  margin-bottom: 10px;  
  text-align: center;  
  font-size: 1.6rem;  
  word-spacing: 3px;  
}  
  
.card p{  
  opacity: 0.75;  
  font-size: 0.8rem;  
  line-height: 1.4;  
}  
  
.card img {
```

```
width: 360px;
height: 300px;
object-fit: cover;
border-radius: 3px;
}

.card-inner {
position: relative;
width: 360px;
height: 100%;
transition: transform 0.9s;
transform-style: preserve-3d;
}

.card:hover .card-inner {
transform: rotateY(180deg);
}

.card-front,
.card-back {
position: absolute;
width: 360px;
height: 100%;
-webkit-backface-visibility: hidden;
backface-visibility: hidden;
}
```

```
.card-back {  
background-color: #222;  
color: #fff;  
padding: 1.5rem;  
transform: rotateY(180deg);  
}
```

```
.text-block {  
position: absolute;  
bottom: 20px;  
right: 20px;  
background-color: black;  
color: white;  
padding-left: 20px;  
padding-right: 20px;  
}
```

```
p  
{  
color:black;  
font-style:italic;  
font-size:30px;  
}
```

```
</style>
```

```
<body style="background-
```

```
image:url({ { url_for('static',filename='images/bck3.png') } });background-position:
center;background-repeat: no-repeat;
```

```
background-size: cover;">
```

```
<div class="header">
```

```
<div style="width:50%;float:left;font-size:2vw;text-align:left;color:black;
```

```
padding-top:1%;padding-left:5%;">Real Time Communication System for Deaf &
Dumb</div>
```

```
<div class="topnav-right"style="padding-top:0.5%;">
```

```
<a class="active" href="/home">Home</a>
```

```
<a href="/upload">Open Web Cam</a>
```

```
</div>
```

```
</div>
```

```
<div class="container">
```

```
<p>In our society, we have people with disabilities. The technology is developing
day by day but no significant developments are undertaken for the betterment of
these people. Communication between deaf-mute and a normal person has always
been a challenging task. It is very difficult for mute people to convey their
message to normal people. Since normal people are not trained on hand sign
language. In emergency times conveying their message is very difficult. The
human hand has remained a popular choice to convey information in situations
where other forms like speech cannot be used. Voice Conversion System with
Hand Gesture Recognition and translation will be very useful to have a proper
conversation between a normal person and an impaired person in any
language.</p>
```

</div>

</body>

<html>

Upload.html

<html lang="en">

<head>

<title>Conversation Engine</title>

<link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">

<style>

```
.header {    position: relative;
              top:0;
              margin:0px;
              z-index: 1;
              left: 0px;
              right: 0px;
              position: fixed;
              background-color: #F36262 ;
              color: white;
              box-shadow: 0px 8px 4px grey;
              overflow: hidden;
```

```

padding-left:20px;
font-family: 'Josefin Sans';
font-size: 2vw;
width: 100%;
height:8%;
text-align: center;
}

.topnav {
overflow: hidden;
background-color: #FCAD98;
}

.topnav-right a {
float: left;
color: black;
text-align: center;
padding: 14px 16px;
text-decoration: none;
font-size: 18px;
}

.topnav-right a:hover {
background-color: #FCAD98;
color: black;
}

.topnav-right a.active {

```



```
background-color: #FCAD98;
color: white;
}
```

```
.topnav-right {
float: right;
padding-right: 100px;
}
```

```
body {

background-color: ;
background-repeat: no-repeat;
background-size: cover;
background-image:
url("https://i.pinimg.com/originals/b2/1d/c6/b21dc69346915015bc4e19bd502f401
b.gif");
background-size: cover;
background-position: 0px 0px;
}

.button {
background-color: #091425;
border: none;
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
```

```
display: inline-block;
font-size: 12px;
border-radius: 16px;
}
.button:hover {
  box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
}
form {border: 3px solid #f1f1f1; margin-left:400px;margin-right:400px;}
```

```
input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  display: inline-block;
  margin-bottom: 18px;
  border: 1px solid #ccc;
  box-sizing: border-box;
}
```

```
button {
  background-color: #091425;
  color: white;
  padding: 14px 20px;
  margin-bottom: 10px;
  border: none;
  cursor: pointer;
  width: 17%;
  border-radius: 4px;
```

```
font-family:Montserrat;  
}
```

```
button:hover {  
  opacity: 0.8;  
}
```

```
.cancelbtn {  
  width: auto;  
  padding: 10px 18px;  
  background-color: #f44336;  
}
```

```
.imgcontainer {  
  text-align: center;  
  margin: 24px 0 12px 0;  
}
```

```
img.avatar {  
  width: 30%;  
  border-radius: 50%;  
}
```

```
.container {  
  padding: 16px;  
}
```

```
span.psw {  
    float: right;  
    padding-top: 16px;  
}
```

```
/* Change styles for span and cancel button on extra small screens */
```

```
@media screen and (max-width: 300px) {  
    span.psw {  
        display: block;  
        float: none;  
    }  
    .cancelbtn {  
        width: 100%;  
    }  
}
```

```
.home{  
    margin:80px;  
  
    width: 84%;  
    height: 500px;  
    padding-top:10px;  
    padding-left: 30px;  
  
}
```

```
.login{  
    margin:80px;
```

```
        box-sizing: content-box;
width: 84%;
height: 420px;
padding: 30px;
border: 10px solid blue;
}
```

```
.left,.right{
    box-sizing: content-box;
    height: 400px;
    margin:20px;
    border: 10px solid blue;
}
```

```
.mySlides {display: none;}
img {vertical-align: middle;}
```

```
/* Slideshow container */
.slideshow-container {
    max-width: 1000px;
    position: relative;
    margin: auto;
}
```

```
/* Caption text */
.text {
    color: #f2f2f2;
    font-size: 15px;
```

```
padding: 8px 12px;
position: absolute;
bottom: 8px;
width: 100%;
text-align: center;
}
/* The dots/bullets/indicators */
.dot {
height: 15px;
width: 15px;
margin: 0 2px;
background-color: #bbb;
border-radius: 50%;
display: inline-block;
transition: background-color 0.6s ease;
}

.active {
background-color: #FCAD98;
}

/* Fading animation */
.fade {
-webkit-animation-name: fade;
-webkit-animation-duration: 1.5s;
animation-name: fade;
animation-duration: 1.5s;
```

```
}
```

```
@-webkit-keyframes fade {  
  from {opacity: .4}  
  to {opacity: 1}  
}
```

```
@keyframes fade {  
  from {opacity: .4}  
  to {opacity: 1}  
}
```

```
/* On smaller screens, decrease text size */  
@media only screen and (max-width: 300px) {  
  .text {font-size: 11px}  
}
```

```
.bar  
{  
margin: 0px;
```

```

padding:20px;
background-color:white;
opacity:0.6;
color:black;
font-family:'Roboto',sans-serif;
font-style: italic;
border-radius:20px;
font-size:25px;
}
a
{
color:grey;
float:right;
text-decoration:none;
font-style:normal;
padding-right:20px;
}
a:hover{
background-color:black;
color:white;
border-radius:15px;0
font-size:30px;
padding-left:10px;
}

p
{

```



```
color:black;
font-style:italic;
font-size:30px;
}
</style>
</head>
```

```
<body style="background-
image:url({ {url_for('static',filename='images/bck3.png')}});background-position:
center;background-repeat: no-repeat;
        background-size: cover;">
```

```
<div class="header">
<div style="width:50%;float:left;font-size:2vw;text-align:left;color:black;
padding-top:1%;padding-left:5%;">Real Time Communication System for Deaf &
Dumb</div>
```

```
<div class="topnav-right"style="padding-top:0.5%;">
```

```
<a href="/home">Home</a>
```

```
<a class="active" href="/upload">Open Web Cam</a>
```

```
</div>
```

```
</div>
```

```
</body>
```

Main.css

```
.img-preview {  
  width: 256px;  
  height: 256px;  
  position: relative;  
  border: 5px solid #F8F8F8;  
  box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);  
  margin-top: 1em;  
  margin-bottom: 1em;  
}
```

```
.img-preview>div {  
  width: 100%;  
  height: 100%;  
  background-size: 256px 256px;  
  background-repeat: no-repeat;  
  background-position: center;  
}
```

```
input[type="file"] {  
  display: none;  
}
```

```
.upload-label{  
  display: inline-block;  
  padding: 12px 30px;  
  background: #39D2B4;  
  color: #fff;
```

```
font-size: 1em;
transition: all .4s;
cursor: pointer;
}
```

```
.upload-label:hover{
  background: #34495E;
  color: #39D2B4;
}
```

```
.loader {
  border: 8px solid #f3f3f3; /* Light grey */
  border-top: 8px solid #3498db; /* Blue */
  border-radius: 50%;
  width: 50px;
  height: 50px;
  animation: spin 1s linear infinite;
}
```

```
@keyframes spin {
  0% { transform: rotate(0deg); }
  100% { transform: rotate(360deg); }
}
```

Video out.txt

<html>

```

<head>
  <title>Real Time Communication System</title>

  <link rel="stylesheet" href="../static/style.css">
  <link rel="stylesheet" href="../static/index.css">
  <link
href="https://fonts.googleapis.com/css2?family=Cuprum&family=Merriweather:ital,wght@1,300&family=Montserrat&family=Open+Sans&family=Roboto&family=Sacramento&display=swap" rel="stylesheet">
  <link
href="https://fonts.googleapis.com/css2?family=Carattere&display=swap"
rel="stylesheet">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Noto+Sans&display=swap"
rel="stylesheet">
  <script src="https://kit.fontawesome.com/872673ab28.js"
crossorigin="anonymous"></script>
  <link rel="icon" href="../static/img/favicon_1.ico">
</head>

<body>

  <div class="nav_head">
    <div class="nav_title"><h1 style="margin-top:15px ;">Realtime

```

Communication System</h1></div>

<div class="icons">

<ul class="header-list">

<li class="header-list-elements">Profile

<li class="header-list-elements">About

<li class="header-list-elements">Home

<li class="header-list-elements"><i class="fa-brands fa-github"></i>

</div>

</div>

<script>

setInterval(function(){

fetch('/label').then(

response =>response.json()

).then(

data => update_label(data)

)

},500

```

);

function update_label(data)
{
    document.getElementById("txt").innerHTML=data;
}

</script>
<div class="main">
    <div class="cont">
        <div class="flex">
            <div class="output_1">
                <div class="cent">
                    <h1>Video Output</h1>
                </div>
                <div class="cent">
                    <div class="video_stream">
                        
                        <!--  -->
                    </div>
                </div>
            </div>

            <div class="output_2">
                <div class="cent">

```

```

        <h1>Predicted Text</h1>
    </div>
    <div class="cent">
        <div class="P_txt">
            <h1 class="txt"> <span id="txt">A</span> </h1>
        </div>
    </div>

</div>

</div>
<p>The above program predict the sign language gesture with the help
trained CNN Model</p>
<div class="btn">
    <a href="../index.html" class="an">Stop Translate</a>
</div>

</div>
</div>
<footer>
    <p id="copyrights" style="color: #474E68; font-size:15px;"></p>
</footer>
<script>
    const d = new Date();
    document.getElementById("copyrights").innerHTML ="Copyright © "+
d.getFullYear();
</script>

```

</html>

Index.txt

<html>

<script>

</script>

<style>

```
.header { position: relative;
top:0;
margin:0px;
z-index: 1;
left: 0px;
right: 0px;
position: fixed;
background-color: #FCAD98 ;
color: white;
box-shadow: 0px 8px 2px grey;
overflow: hidden;
padding-left:20px;
font-family: 'Times New Roman';
font-size: 2vw;
width: 100%;
height:8%;
text-align: center;
```



```
}  
.topnav {  
  overflow: hidden;  
  background-color: #FCAD98;  
}  
.topnav-right a {  
  float: left;  
  color: black;  
  text-align: center;  
  padding: 14px 16px;  
  text-decoration: none;  
  font-size: 18px;  
}  
.topnav-right a:hover {  
  background-color: #FCAD98;  
  color: black;  
}  
.topnav-right a.active {  
  background-color: #FCAD98;  
  color: white;  
}  
.topnav-right {  
  float: right;  
  padding-right: 100px;  
}  
body {  
  background-image: -webkit-linear-gradient(90deg, skyblue 0%, steelblue 100%);
```

```

background-image: url("");
background-size: cover;
background-attachment: fixed;
background-size: 100% 100%;
background-color: ;
background-repeat: no-repeat;
background-size:cover;
background-position: 0px 0px;
}
.button {
background-color: #091425;
border: none;
color: white;
padding: 15px 32px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 12px;
border-radius: 16px;
}
.button:hover {
box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
}
form {border: 3px solid #f1f1f1; margin-left:400px;margin-right:400px;}
input[type=text], input[type=password] {
width: 100%;
padding: 12px 20px;

```

```
display: inline-block;
margin-bottom: 18px;
border: 1px solid #ccc;
box-sizing: border-box;
}
button {
background-color: #091425;
color: white;
padding: 14px 20px;
margin-bottom: 10px;
border: none;
cursor: pointer;
width: 17%;
border-radius: 4px;
font-family: Montserrat;
}
button:hover {
opacity: 0.8;
}
.cancelbtn {
width: auto;
padding: 10px 18px;
background-color: #f44336;
}
.imgcontainer {
text-align: center;
margin: 24px 0 12px 0;
```

```

}
img.avatar {
width: 30%;
border-radius: 50%;
}
.container {
padding: 16px;
}
span.psw {
float: right;
padding-top: 16px;
}
/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
span.psw {
display: block;
float: none;
}
.cancelbtn {
width: 100%;
}
}
.home{
margin:80px;
width: 84%;
height: 500px;
padding-top:10px;

```

```

padding-left: 30px;
}
.login{
margin:80px;
box-sizing: content-box;
width: 84%;
height: 420px;
padding: 30px;
border: 10px solid blue;
}
.left,.right{
box-sizing: content-box;
height: 400px;
margin:20px;
border: 10px solid blue;
}
.mySlides {display: none;}
img {vertical-align: middle;}
/* Slideshow container */
.slideshow-container {
max-width: 1000px;
position: relative;
margin: auto;
}
/* Caption text */
.text {
color: #f2f2f2;

```

```

font-size: 15px;
padding: 8px 12px;
position: absolute;
bottom: 8px;
width: 100%;
text-align: center;
}

/* The dots/bullets/indicators */
.dot {
height: 15px;
width: 15px;
margin: 0 2px;
background-color: #bbb;
border-radius: 50%;
display: inline-block;
transition: background-color 0.6s ease;
}

.active {
background-color: #FCAD98;
}

/* Fading animation */
.fade {
-webkit-animation-name: fade;
-webkit-animation-duration: 1.5s;
animation-name: fade;
animation-duration: 1.5s;
}

```

```

@-webkit-keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}
@keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}
/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
  .text {font-size: 11px}
}
@import url('https://fonts.googleapis.com/css2?family=Poppins&display=swap');
* {
  box-sizing: border-box;
}
body {
  min-height: 100vh;
  margin: 0;
  color: #fff;
  font-family: 'Poppins',sans-serif;
  display: flex;
  align-items: center;
  justify-content: center;
  background-color: #f5f5f5;
}
.container {

```

```
max-width: 1376px;
margin: auto;
padding: 2rem 1.5rem;
}
.cards {
display: flex;
flex-wrap: wrap;
align-items: center;
justify-content: center;
}
.card {
cursor: pointer;
background-color: transparent;
height: 300px;
perspective: 1000px;
margin: 1rem;
align-items: center;
justify-content: center;
}
.card h3 {
border-bottom: 1px #fff solid;
padding-bottom: 10px;
margin-bottom: 10px;
text-align: center;
font-size: 1.6rem;
word-spacing: 3px;
}
```



```
.card p{
opacity: 0.75;
font-size: 0.8rem;
line-height: 1.4;
}
.card img {
width: 360px;
height: 300px;
object-fit: cover;
border-radius: 3px;
}
.card-inner {
position: relative;
width: 360px;
height: 100%;
transition: transform 0.9s;
transform-style: preserve-3d;
}
.card:hover .card-inner {
transform: rotateY(180deg);
}
.card-front,
.card-back {
position: absolute;
width: 360px;
height: 100%;
-webkit-backface-visibility: hidden;
```

```

backface-visibility: hidden;
}
.card-back {
background-color: #222;
color: #fff;
padding: 1.5rem;
transform: rotateY(180deg);
}
.text-block {
position: absolute;
bottom: 20px;
right: 20px;
background-color: black;
color: white;
padding-left: 20px;
padding-right: 20px;
}
p
{
color:black;
font-style:Times New Roman;
font-size:30px;
}
</style>
<body style="background-
image:url({ {url_for('static',filename='images/bck3.png')}});background-position:
center;background-repeat: no-repeat;

```

```

background-size: cover;">
<div class="header">
<div style="width:50%;float:left;font-size:2vw;text-align:left;color:black;
padding-top:1%;padding-left:5%;">Real Time Communication System for Deaf &
Dumb</div>
<div class="topnav-right"style="padding-top:0.5%;">
<a class="active" href="/home">Home</a>
<a href="/upload">Open Web Cam</a>
</div>
</div>
<div class="container">
<p>In our society, we have people with disabilities. The technology is developing
day by day but no significant developments are undertaken for the betterment of
these people. Communication between deaf-mute and a normal person has always
been a challenging task. It is very difficult for mute people to convey their
message to normal people. Since normal people are not trained on hand sign
language. In emergency times conveying their message is very difficult. The
human hand has remained a popular choice to convey information in situations
where other forms like speech cannot be used. Voice Conversion System with
Hand Gesture Recognition and translation will be very useful to have a proper
conversation between a normal person and an impaired person in any
language.</p>
</div>
</body>
<html>

```

About us.txt

```

!DOCTYPE html>
<html>
<head>
<title>About Us</title>
<style>
body{
    background-image: url("paper.gif");
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-size: cover;
}
h1,p{
    font-family:Times New Roman;
}
p3{
    font-family:Times New Roman;
    font-size:30px;
}
.column {
    float: left;
    width: 13.3%;
    padding: 40px;
}
</style>
<body>
<h1><center><u>About Us</u></center></h1>

```

<hr>

<p> The Project i.e. <p1 style="color:Green">Real Time Communication Powered By AI</p1> was made by the students of SVHEC in the view of <p2 style="color:Green"> Specially Disabled people </p2> who find extreme difficult , involving in the day to day communication and accepting negative comments from the outsiders.

<p>Hope this project will satisfy the requirements of the end users and motivate them in all walks of life</p>

<hr>

<p>To know about the Project in detail,Click here</p>

<pre>

<p3><u>Contributed by,</u></p3>

<div class="row">

<div class="column">

<p>SREENITHI S</p>

</div>

<div class="column">

<p>PARIMALA B</p>

</div>

<div class="column">

```
<p>SINDHUJA A</p>
</div>
<div class="column">
  <p>SRIDHARAN B</p>
</div>
</div>
</body>
</head>
</html>
```

PROJECT DEMO LINK

<https://drive.google.com/file/d/1Sr2TXbKUJ7O5Q2yyuBQW6m2JckldPmXc/view?usp=sharing>

GITHUB

<https://github.com/IBM-EPBL/IBM-Project-42782-1660709093>