Real-Time Communication SystemPowered by AI for Specially Abled

NALAIYA THIRAN PROJECT BASED LEARNING ON PROFESSIONAL READLINESS FOR INNOVATION, EMPLOYNMENT AND ENTERPRENEURSHIP

A PROJECT REPORT

TEAM ID – PNT2022TMID27674

MOHAMED SHATHIR S – 311419104057

SAI PRAVEEN M -311419104067

MAHESHRAJ V – 311419104053

MANICK RAYBON S - 311419104054

BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING

MEENAKSHI COLLEGE OF ENGINEERING

CHENNAI – 600078

1.INTRODUCTION

1.1 Overview

People get to know one another by sharing their ideas, thoughts, and experiences with those around them. There are numerous ways to accomplish this, the best of which is the gift of "Speech." Everyone can very convincingly transfer their thoughts and understand each other through speech. It will be unjust if we overlook those who are denied this priceless gift: the deaf and dumb. In such cases, the human hand has remained the preferred method of communication.

1.2 Purpose

The project's purpose is to create a system that translates sign language into a humanunderstandable language so that ordinary people may understand it.

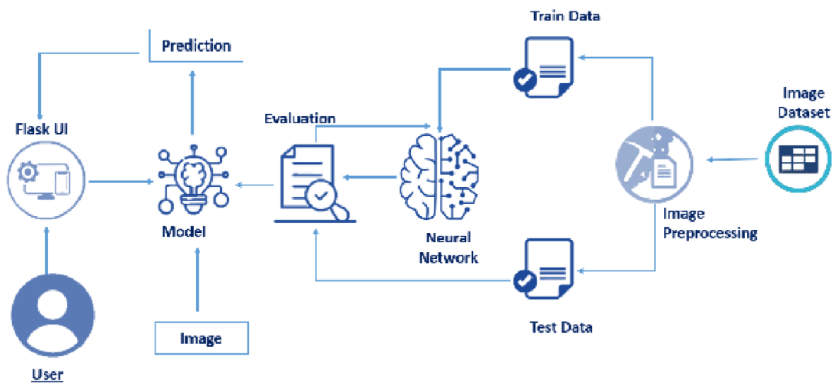
2. LITERATURE SURVEY

S.No	Title	Author	Year	Abstract	Methodology
1	Converting Indian Sign Language to English text	Dhivyasr	2021	The system was able to convert sign language to the English language for a very small dataset.	The system uses various convolution neural network techniques
2	Converting Arabic Text to Arabic Sign Language	Jamil	2020	The system successfully converted Arabic text to Arabic sign language with 87% efficiency and then show the corresponding sign language animatedly.	The system uses various text parsing and word processing techniques.
3	A System to translate English to ISL	Patel	2020	The System takes input in English and with a 77% accuracy it converts the input to SIGML animation, but the system is only shown to convert words or letters.	Authors have used various NLP and Google APIs.

4	AAWAAZ: A Communication System for Deaf & Dumb.	Anchal Sood, Anju Mishra	2016	The paper proposes a framework for recognizing hand gesture which would serve not only as a way of communication between deaf and dumb and mute people, but also, as an instructor. Deaf and dumb individuals lack in proper communication with normal people and find it difficult to properly express themselves. Thus, they are subjected to face many issues in this regard.	From the input RGB image, the hand is separated and morphological operations are performed to identify the region of interest. The features of the gesture are then extracted and compared to a database of features of standard gestures. Finally, based on the comparison the output is generated.
5	A two-way communication system between deaf and normal people	Ahire	2015	The system only shows ISL signs for those words which are stored in the database and skip any other word.	The authors have utilized various machine learning algorithms and NLP techniques.
6	Full Duplex Communication System for Deaf & Dumb People	Shraddha R. Ghorpade, Surendra K. Waghamare	2015	One of the important problems that our society faces is that people with disabilities are finding it hard to cope-up with the fastgrowing technology. The access to communication technologies has become essential for the handicapped people.	The methodology used is similar to except that, instead of bare hands, the system requires the user to wear gloves to extract hand gesture.

3. THEORITICAL ANALYSIS

Block diagram



Hardware / Software designing

Hardware Requirements:

Operating system	Windows, Mac, Linux
CPU(For Training)	Multi Core Processors (i3 or above/equivalent)
GPU(For Training)	NVIDIA AI Capable / Google's TPU
WEB CAM	Integrated or External with FullHD Support

Software Requirements:

Python	v3.9.0 or Above
Python Packages	flask, tensorflow, opencv-python, keras, numpy, pandas, virtualenv, pillow
Web Browser	Mozilla Firefox, Google Chrome or any modern web browser
IBM Cloud (for training)	Watson Studio - Model Training & Deployment as Machine Learning Instance

4. EXPERIMENTAL INVESTIGATIONS

Training and Testing using Dataset Provided

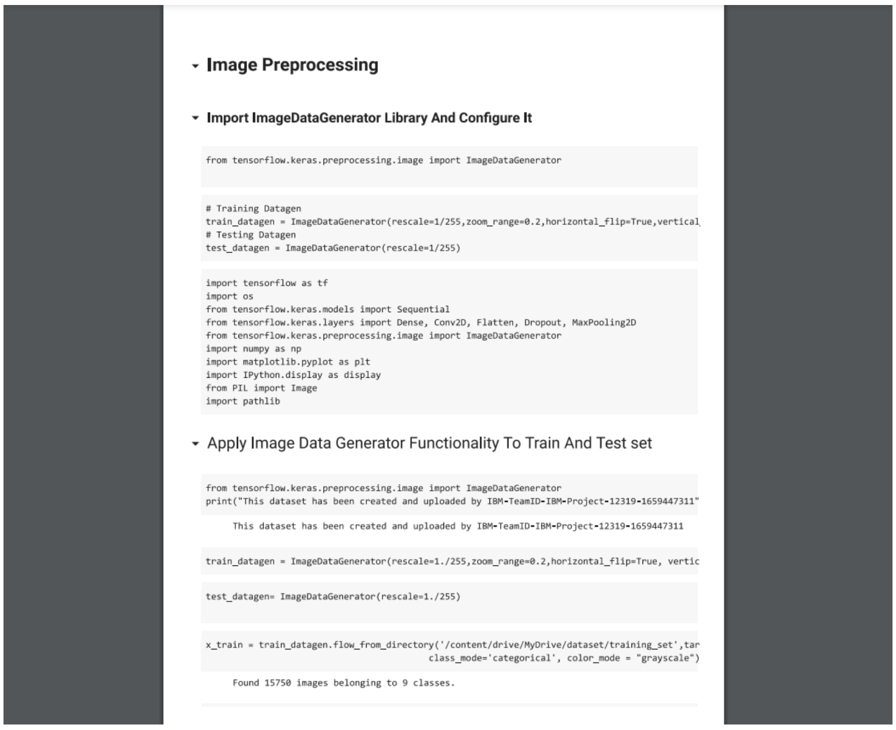


Image Preprocessing

Import ImageDataGenerator Library And Configure It

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

```
import tensorflow as tf
import os
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, Flatten, Dropout, MaxPooling2D
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt
import IPython.display as display
from PIL import Image
import pathlib
```

Apply Image Data Generator Functionality To Train And Test set

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
print("This dataset has been created and uploaded by IBM-TeamID-IBM-Project-12319-1659447311"
```

```
    This dataset has been created and uploaded by IBM-TeamID-IBM-Project-12319-1659447311
```

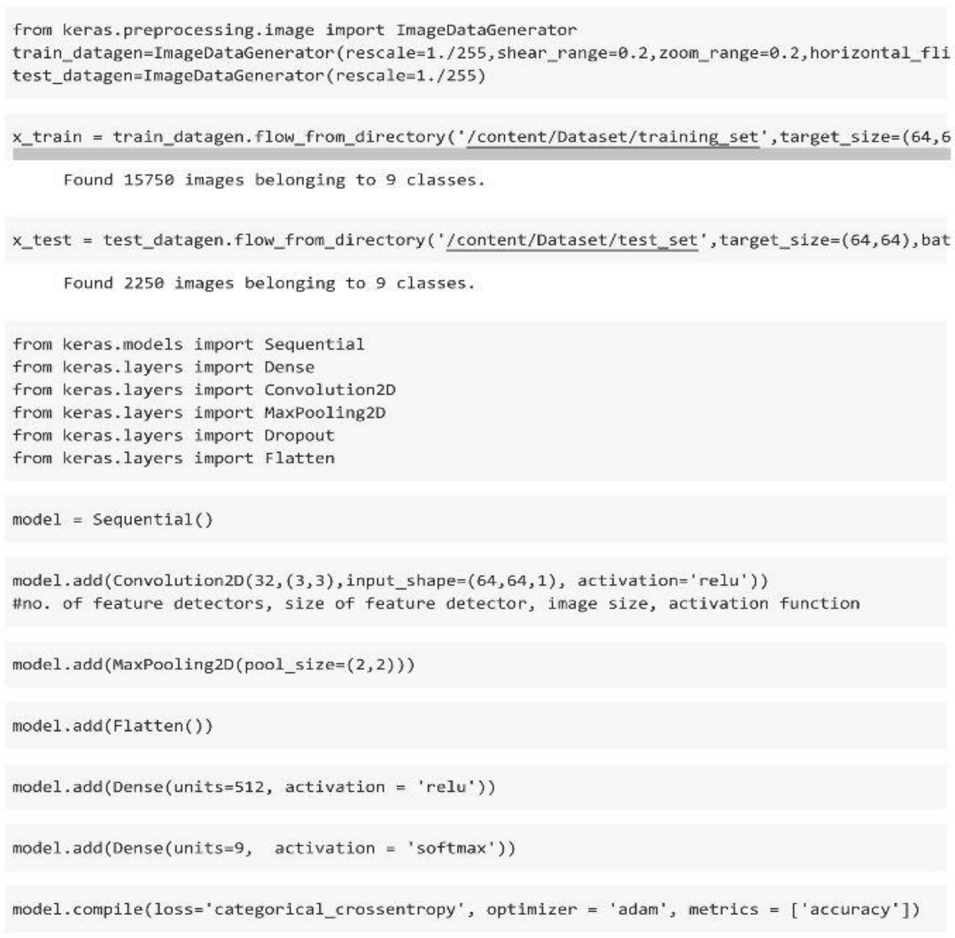
```
train_datagen = ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True, vertic
```

```
test_datagen= ImageDataGenerator(rescale=1./255)
```

```
x_train = train_datagen.flow_from_directory('/content/drive/MyDrive/dataset/training_set',tar
                                             class_mode='categorical', color_mode = "grayscale")
```

```
    Found 15750 images belonging to 9 classes.
```

Model Creation



```
from keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_fli
test_datagen=ImageDataGenerator(rescale=1./255)
```

```
x_train = train_datagen.flow_from_directory('/content/Dataset/training_set',target_size=(64,6
```

```
    Found 15750 images belonging to 9 classes.
```

```
x_test = test_datagen.flow_from_directory('/content/Dataset/test_set',target_size=(64,64),bat
```

```
    Found 2250 images belonging to 9 classes.
```

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Dropout
from keras.layers import Flatten
```

```
model = Sequential()
```

```
model.add(Convolution2D(32,(3,3),input_shape=(64,64,1), activation='relu'))
#no. of feature detectors, size of feature detector, image size, activation function
```

```
model.add(MaxPooling2D(pool_size=(2,2)))
```

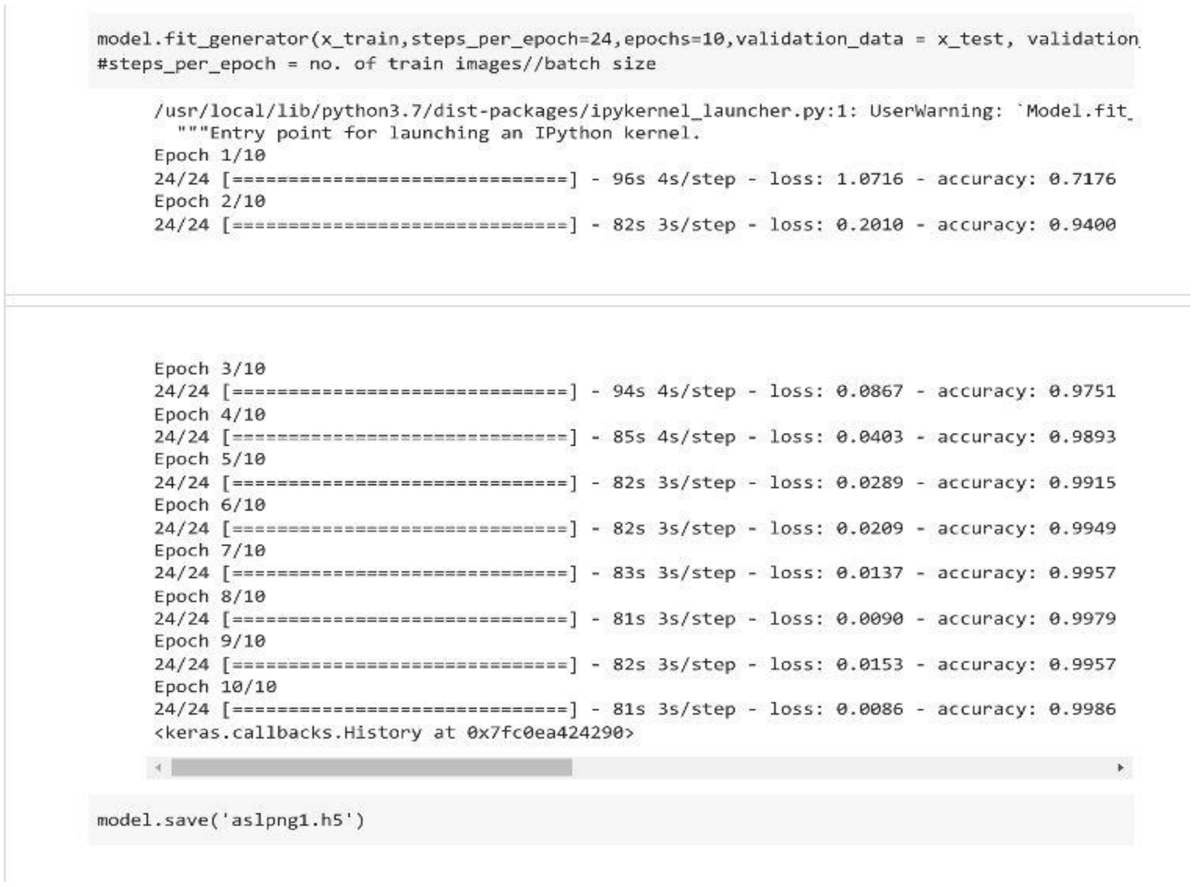
```
model.add(Flatten())
```

```
model.add(Dense(units=512, activation = 'relu'))
```

```
model.add(Dense(units=9,  activation = 'softmax'))
```

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

SAVING THE MODEL



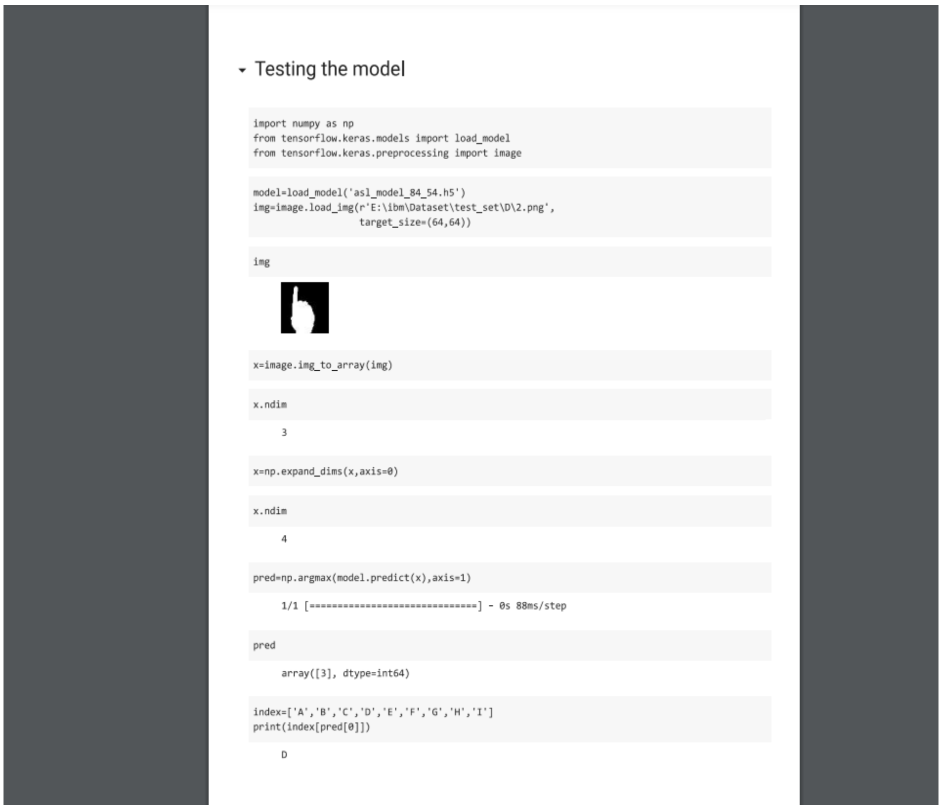
```
model.fit_generator(x_train,steps_per_epoch=24,epochs=10,validation_data = x_test, validation
#steps_per_epoch = no. of train images//batch size
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_
  """Entry point for launching an IPython kernel.
Epoch 1/10
24/24 [==============================] - 96s 4s/step - loss: 1.0716 - accuracy: 0.7176
Epoch 2/10
24/24 [==============================] - 82s 3s/step - loss: 0.2010 - accuracy: 0.9400
```

```
Epoch 3/10
24/24 [==============================] - 94s 4s/step - loss: 0.0867 - accuracy: 0.9751
Epoch 4/10
24/24 [==============================] - 85s 4s/step - loss: 0.0403 - accuracy: 0.9893
Epoch 5/10
24/24 [==============================] - 82s 3s/step - loss: 0.0289 - accuracy: 0.9915
Epoch 6/10
24/24 [==============================] - 82s 3s/step - loss: 0.0209 - accuracy: 0.9949
Epoch 7/10
24/24 [==============================] - 83s 3s/step - loss: 0.0137 - accuracy: 0.9957
Epoch 8/10
24/24 [==============================] - 81s 3s/step - loss: 0.0090 - accuracy: 0.9979
Epoch 9/10
24/24 [==============================] - 82s 3s/step - loss: 0.0153 - accuracy: 0.9957
Epoch 10/10
24/24 [==============================] - 81s 3s/step - loss: 0.0086 - accuracy: 0.9986
<keras.callbacks.History at 0x7fc0ea424290>
```

```
model.save('aslpng1.h5')
```

TESTING THE MODEL



▾ Testing the model

```
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```

```
model=load_model('asl_model_84_54.h5')
img=image.load_img(r'E:\ibm\Dataset\test_set\D\2.png',
                   target_size=(64,64))
```

```
img
```

```
x=image.img_to_array(img)
```

```
x.ndim
```

```
3
```

```
x=np.expand_dims(x,axis=0)
```

```
x.ndim
```

```
4
```

```
pred=np.argmax(model.predict(x),axis=1)
```

```
1/1 [==============================] - 0s 88ms/step
```

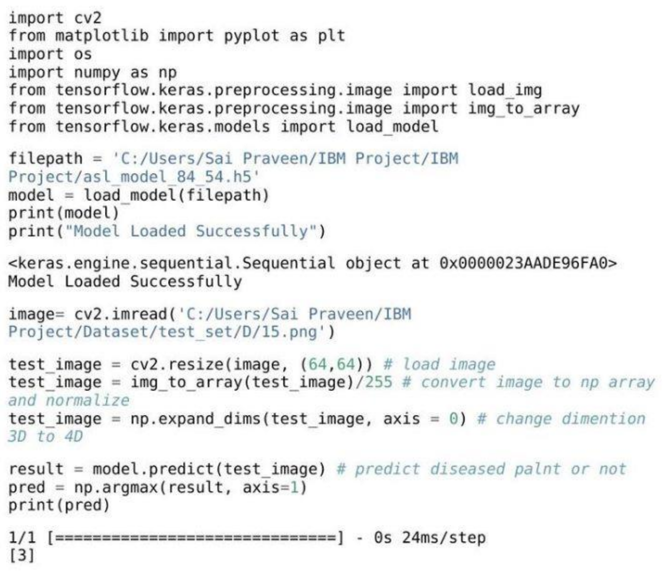
```
pred
```

```
array([3], dtype=int64)
```

```
index=['A','B','C','D','E','F','G','H','I']
print(index[pred[0]])
```

```
D
```

PREDICTING THE MODEL



```
import cv2
from matplotlib import pyplot as plt
import os
import numpy as np
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
```

```
filepath = 'C:/Users/Sai Praveen/IBM Project/IBM
Project/asl_model_84_54.h5'
model = load_model(filepath)
print(model)
print("Model Loaded Successfully")
```

```
<keras.engine.sequential.Sequential object at 0x0000023AADE96FA0>
Model Loaded Successfully
```

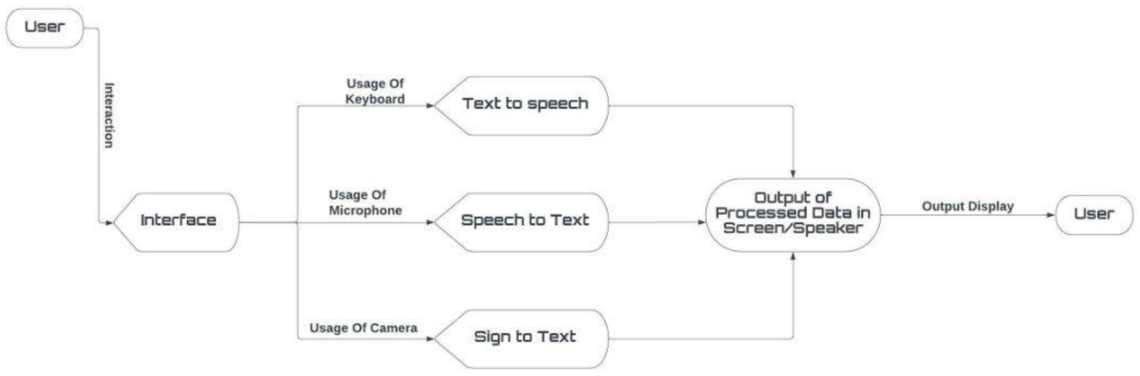
```
image= cv2.imread('C:/Users/Sai Praveen/IBM
Project/Dataset/test_set/D/15.png')
```

```
test_image = cv2.resize(image, (64,64)) # load image
test_image = img_to_array(test_image)/255 # convert image to np array
and normalize
test_image = np.expand_dims(test_image, axis = 0) # change dimention
3D to 4D
```

```
result = model.predict(test_image) # predict diseased palnt or not
pred = np.argmax(result, axis=1)
print(pred)
```

```
1/1 [==============================] - 0s 24ms/step
[3]
```

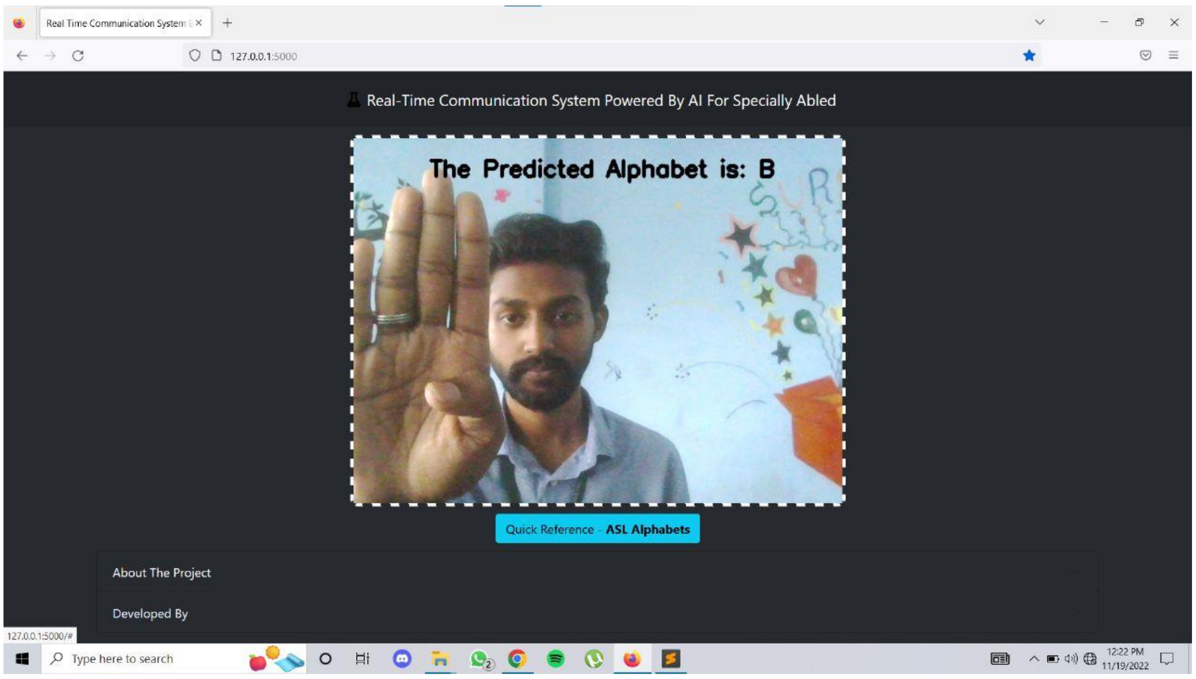
5.FLOWCHART

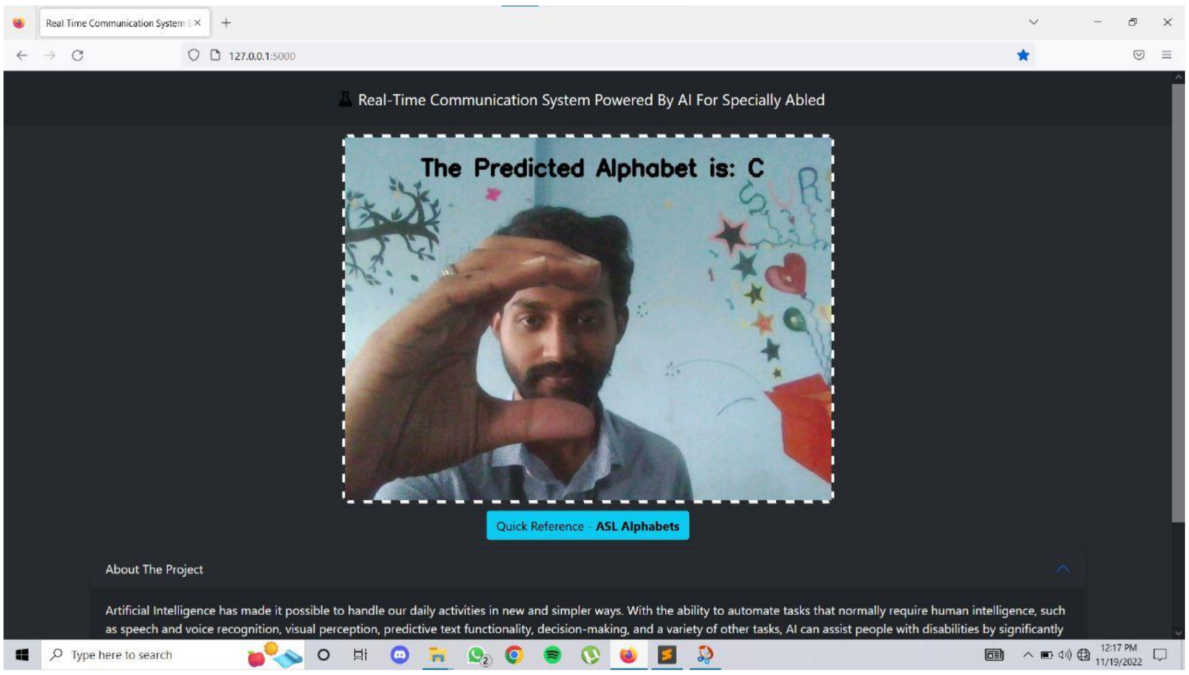


6. RESULT

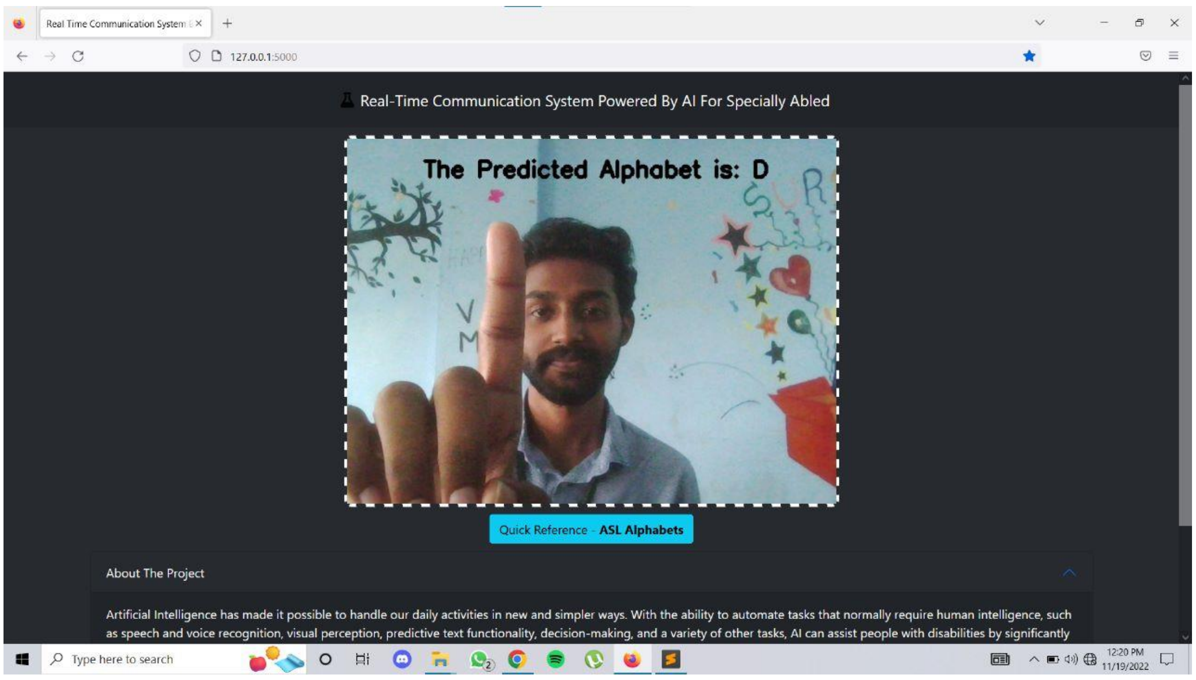
The proposed procedure was implemented and tested with set of images. The set of 15750 images of Alphabets from “A” to “I” are used for training database and a set of 2250 images of Alphabets from “A” to “I” are used for testing database. Once the gesture is recognise the equivalent Alphabet is shown on the screen

Some sample images of the output are provided below:


Real Time Communication System
127.0.0.1:5000
Real-Time Communication System Powered By AI For Specially Abled
The Predicted Alphabet is: B
Quick Reference - ASL Alphabets
About The Project
Developed By
Type here to search
12:22 PM
11/19/2022


Real Time Communication System
127.0.0.1:5000
Real-Time Communication System Powered By AI For Specially Abled
The Predicted Alphabet is: C
Quick Reference - ASL Alphabets
About The Project
Type here to search
12:17 PM
11/19/2022

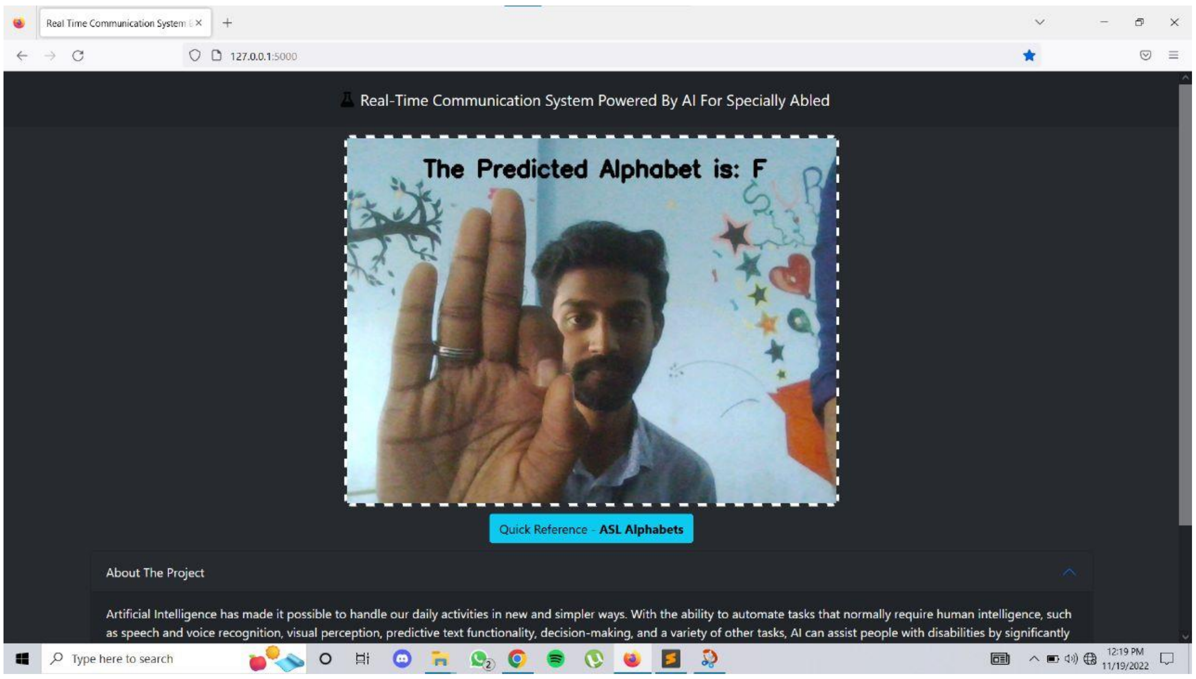
Artificial Intelligence has made it possible to handle our daily activities in new and simpler ways. With the ability to automate tasks that normally require human intelligence, such as speech and voice recognition, visual perception, predictive text functionality, decision-making, and a variety of other tasks, AI can assist people with disabilities by significantly


Real-Time Communication System Powered By AI For Specially Abled

Quick Reference - **ASL Alphabets**

About The Project

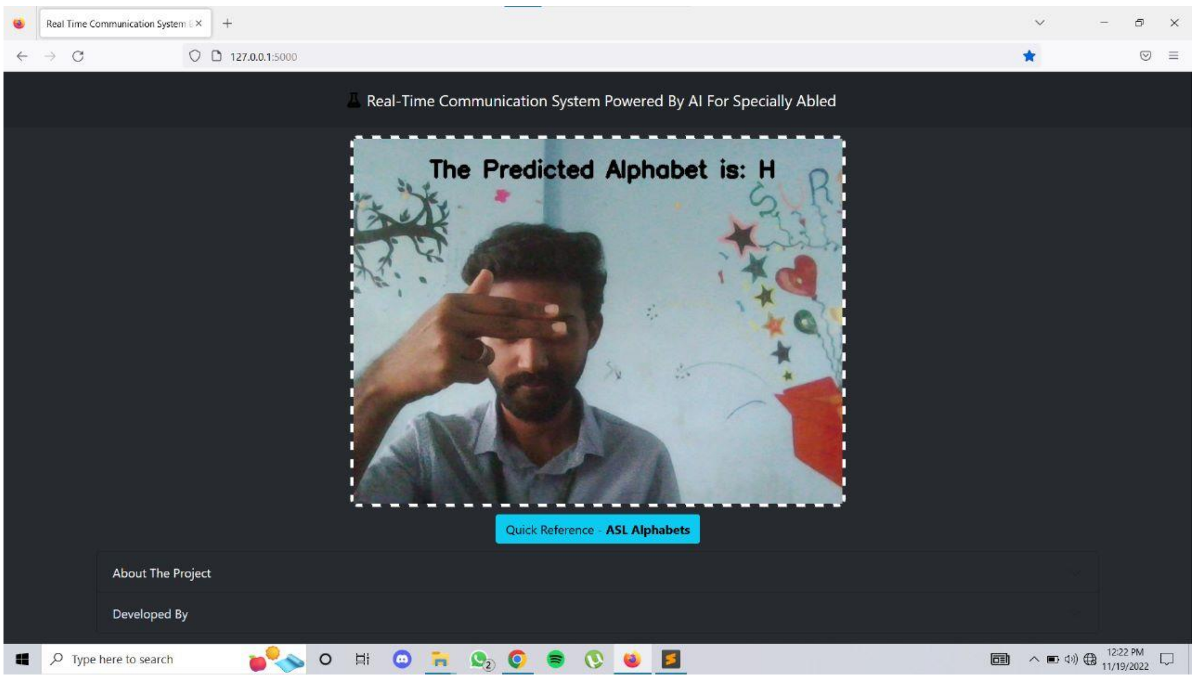
Artificial Intelligence has made it possible to handle our daily activities in new and simpler ways. With the ability to automate tasks that normally require human intelligence, such as speech and voice recognition, visual perception, predictive text functionality, decision-making, and a variety of other tasks, AI can assist people with disabilities by significantly

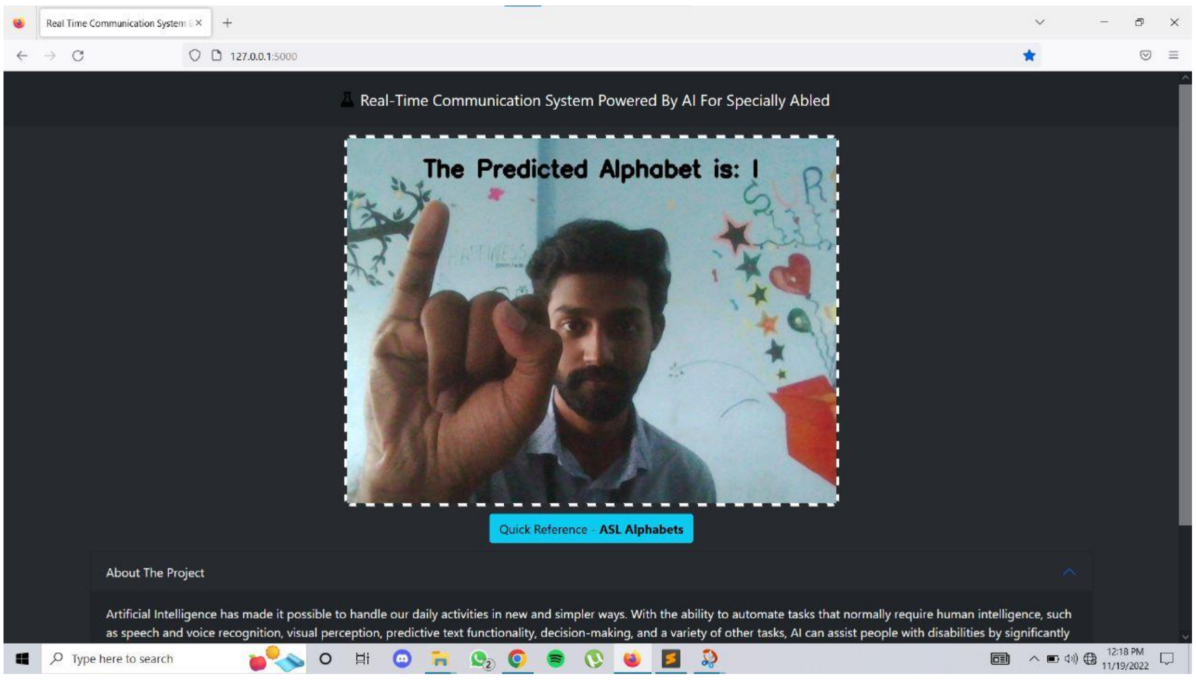

Real-Time Communication System Powered By AI For Specially Abled

Quick Reference - **ASL Alphabets**

About The Project

Artificial Intelligence has made it possible to handle our daily activities in new and simpler ways. With the ability to automate tasks that normally require human intelligence, such as speech and voice recognition, visual perception, predictive text functionality, decision-making, and a variety of other tasks, AI can assist people with disabilities by significantly


Real Time Communication System
127.0.0.1:5000
Real-Time Communication System Powered By AI For Specially Abled
The Predicted Alphabet is: H
Quick Reference - ASL Alphabets
About The Project
Developed By
Type here to search
12:22 PM
11/19/2022


Real Time Communication System
127.0.0.1:5000
Real-Time Communication System Powered By AI For Specially Abled
The Predicted Alphabet is: I
Quick Reference - ASL Alphabets
About The Project
Type here to search
12:18 PM
11/19/2022

Artificial Intelligence has made it possible to handle our daily activities in new and simpler ways. With the ability to automate tasks that normally require human intelligence, such as speech and voice recognition, visual perception, predictive text functionality, decision-making, and a variety of other tasks, AI can assist people with disabilities by significantly

7. ADVANTAGES & DISADVANTAGES

Advantages:

1. It is possible to create a mobile application to bridge the communication gap between deaf and dumb persons and the general public.

2. As different sign language standards exist, their dataset can be added, and the user can choose which sign language to read.

Disadvantages:

1. The current model only works from alphabets A to I.
2. In absence of gesture recognition, alphabets from J cannot be identified as they require some kind of gesture input from the user.
3. As the quantity/quality of images in the dataset is low, the accuracy is not great, but that can easily be improved by change in dataset.

8. APPLICATIONS

1. It will contribute to the development of improved communication for the deafened. The majority of people are unable to communicate via sign language, which creates a barrier to communication
2. As a result, others will be able to learn and comprehend sign language and communicate with the deaf and dumb via the web app.
3. According to scientific research, learning sign language improves cognitive abilities, attention span, and creativity.

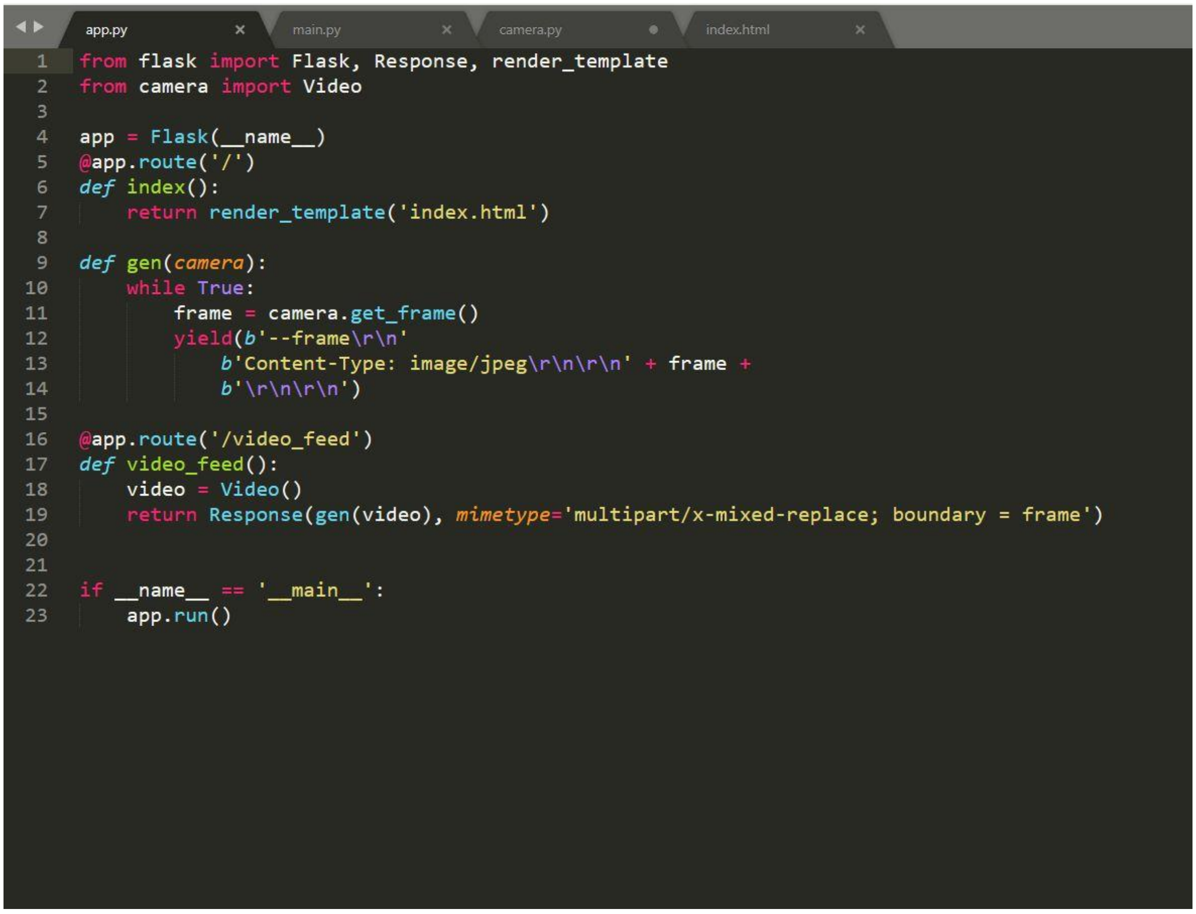
9. CONCLUSION

Sign language is a useful tool for facilitating communication between deaf and hearing people. Because it allows for two-way communication, the system aims to bridge the communication gapbetween deaf people and the rest of society. The proposed methodology translates language into English alphabets that are understandable to humans.

This system sends hand gestures to the model, who recognises them and displays the equivalentAlphabet on the screen. Deaf-mute people can use their hands to perform sign language, which will then be converted into alphabets, thanks to this project.

10. APPENDIX

Web app code



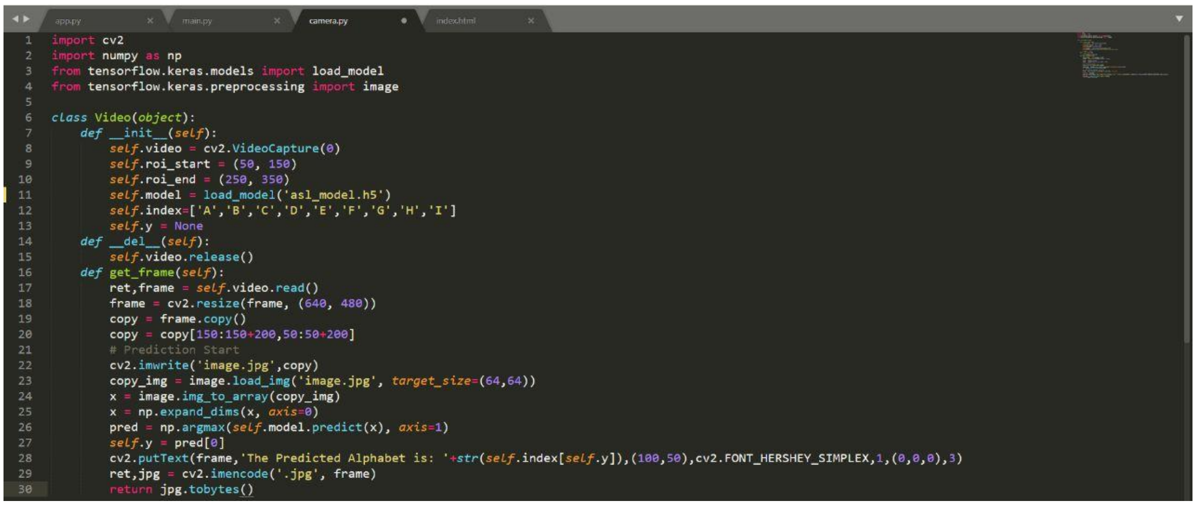
```python
from flask import Flask, Response, render_template
from camera import Video

app = Flask(__name__)
@app.route('/')
def index():
    return render_template('index.html')

def gen(camera):
    while True:
        frame = camera.get_frame()
        yield(b'--frame\r\n'
            b'Content-Type: image/jpeg\r\n\r\n' + frame +
            b'\r\n\r\n')

@app.route('/video_feed')
def video_feed():
    video = Video()
    return Response(gen(video), mimetype='multipart/x-mixed-replace; boundary = frame')


if __name__ == '__main__':
    app.run()
```



```python
import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

class Video(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)
        self.roi_start = (50, 150)
        self.roi_end = (250, 350)
        self.model = load_model('asl_model.h5')
        self.index=['A','B','C','D','E','F','G','H','I']
        self.y = None
    def __del__(self):
        self.video.release()
    def get_frame(self):
        ret,frame = self.video.read()
        frame = cv2.resize(frame, (640, 480))
        copy = frame.copy()
        copy = copy[150:150+200,50:50+200]
        # Prediction Start
        cv2.imwrite('image.jpg',copy)
        copy_img = image.load_img('image.jpg', target_size=(64,64))
        x = image.img_to_array(copy_img)
        x = np.expand_dims(x, axis=0)
        pred = np.argmax(self.model.predict(x), axis=1)
        self.y = pred[0]
        cv2.putText(frame,'The Predicted Alphabet is: '+str(self.index[self.y]),(100,50),cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0),3)
        ret,jpg = cv2.imencode('.jpg', frame)
        return jpg.tobytes()
```

American Sign Language Standard Reference: