

[IBM-Project-2475-1658472446](#)

Team ID: PNT2022TMID01172

Team Leader-Stanley Deivanayagam

Team Member 1 – Sajith

Team Member 2 – Sachin Prakash Raj

Team Member 3 –Harish Kumar

Project Description

The project deals on building an application which helps the specially challenged people to communicate between them and the common people. Communication between a person with hearing/speech impairment and a normal person has always been a challenging task. This application tries to reduce the barrier of communication by developing an assistive application for specially challenged people.

Data Collection:

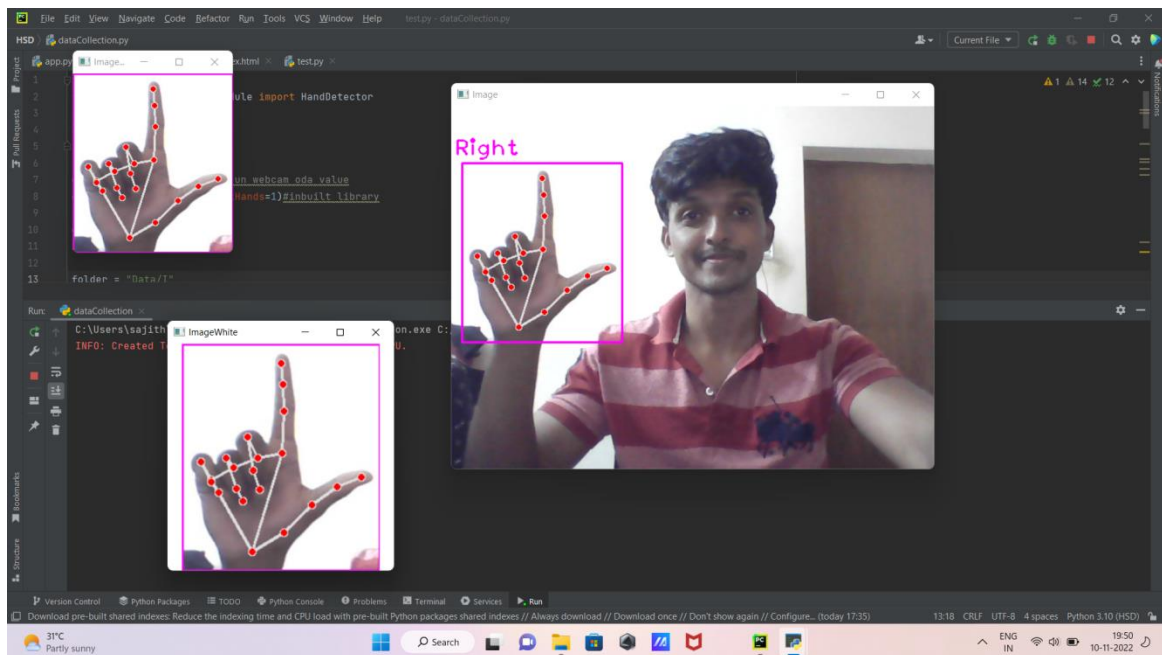


Image Augmentation:

Image Augmentation

```
[ ] from tensorflow.keras.preprocessing.image import ImageDataGenerator
    print("This dataset has been created and uploaded by IBM-TeamID-IBM-Project-2475-1658472446")

    This dataset has been created and uploaded by IBM-TeamID-IBM-Project-2475-1658472446

[ ] train_datagen = ImageDataGenerator(rescale=1./255, zoom_range=0.2, horizontal_flip=True, vertical_flip=False)

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

Applying Convolution ,Dense Layers:

```
[ ] from tensorflow.keras.layers import Dense, Convolution2D, MaxPooling2D, Flatten

[ ] model = Sequential()

[ ] model.add(Convolution2D(32, (3,3), input_shape=(100,100,3), activation = 'relu')) #Feature map

[ ] model.add(MaxPooling2D(pool_size = (2,2))) #Pooled matrix

[ ] model.add(Flatten())

[ ] model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 98, 98, 32)	896
max_pooling2d (MaxPooling2D)	(None, 49, 49, 32)	0
flatten (Flatten)	(None, 76832)	0

Tested Model:

```
[ ] img



[ ] x=image.img_to_array(img)

[ ] x=np.expand_dims(x,axis=0)

[ ] y=np.argmax(model.predict(x),axis=1)

1/1 [=====] - 0s 56ms/step

[ ] y

array([1])

[ ] index=[ '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' ]

[ ] index[y[0]]

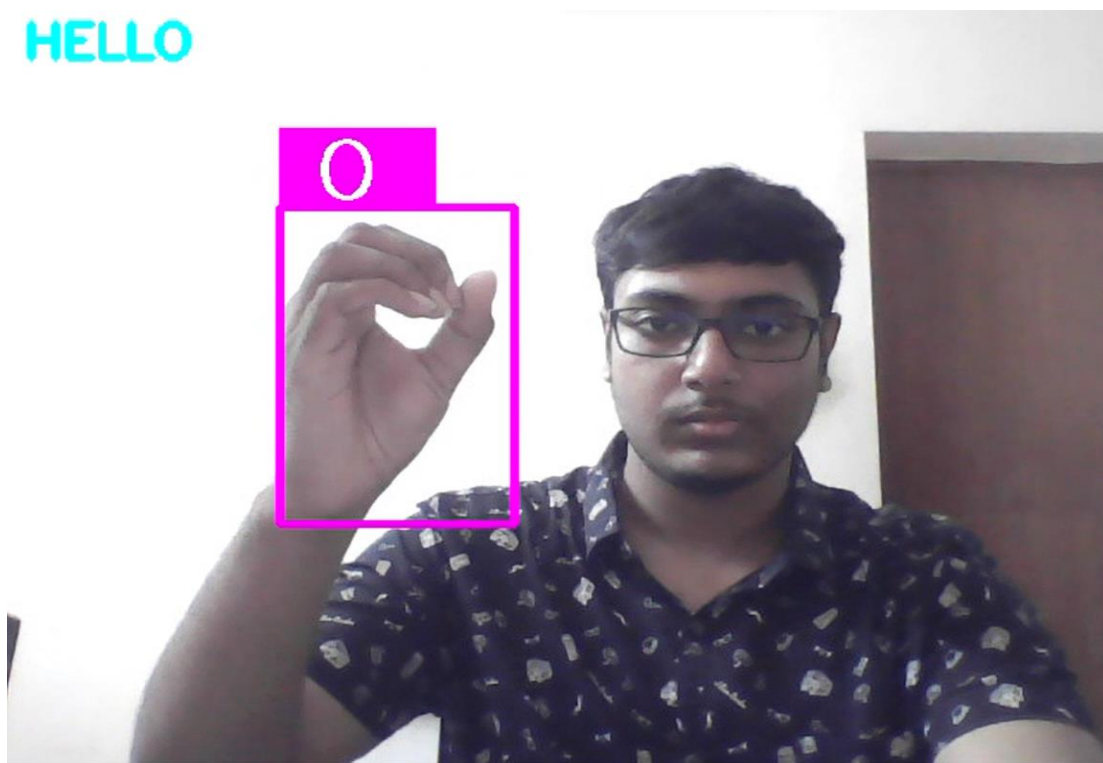
'B'
```

Testing:



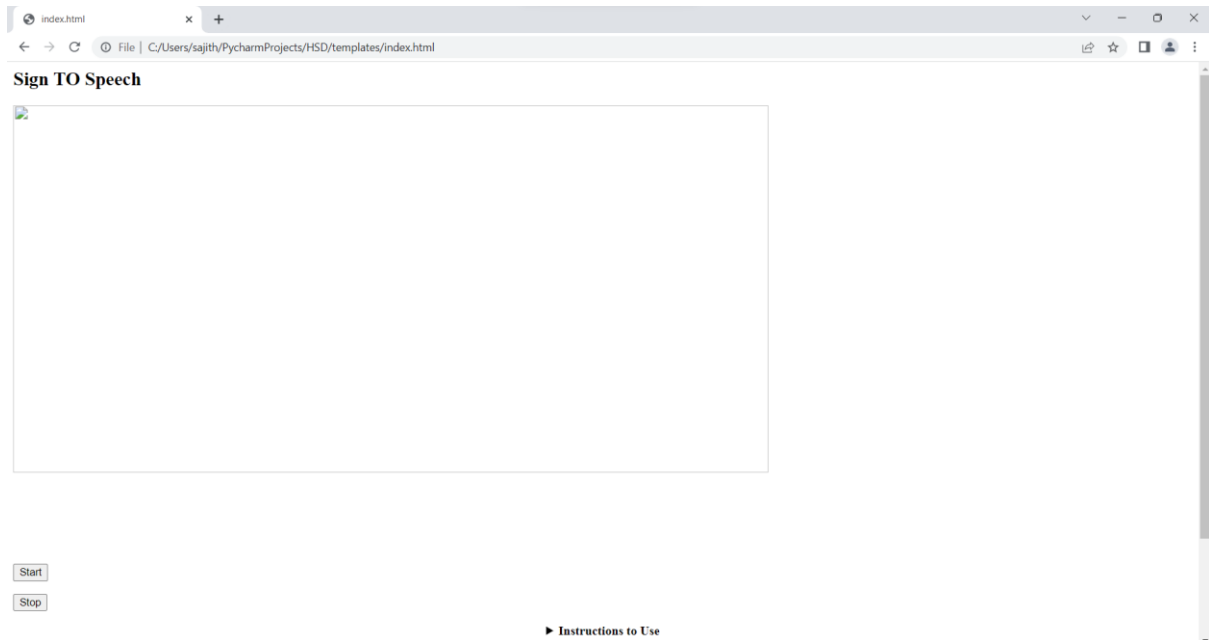
Saving text:

HELLO

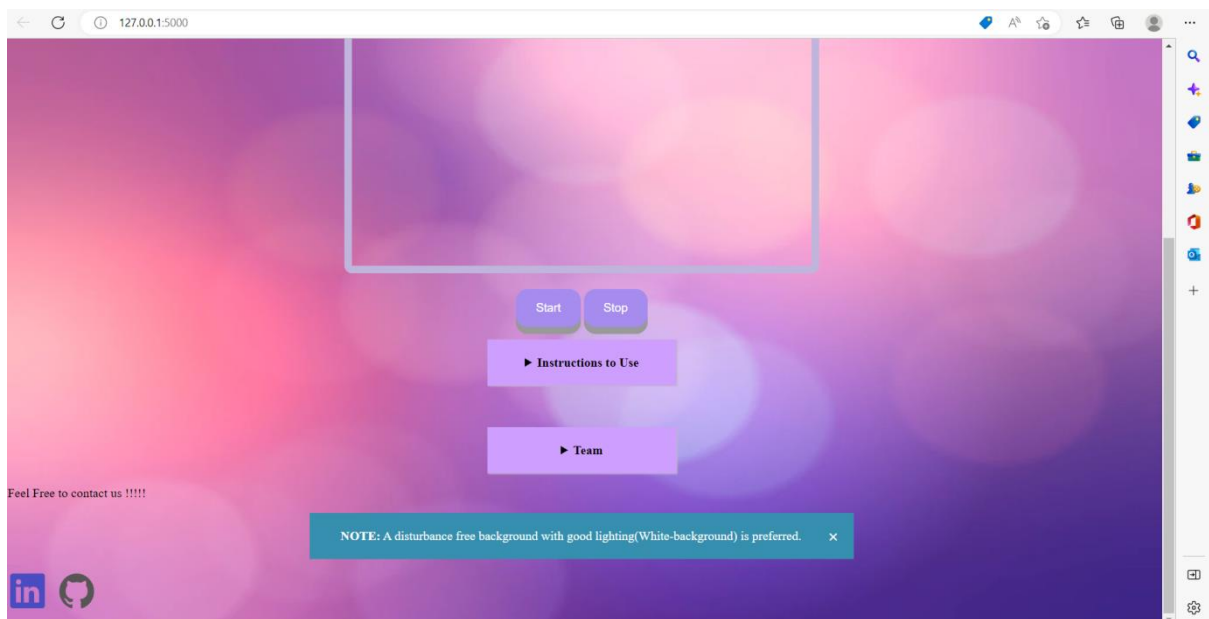


Building Flask Application:

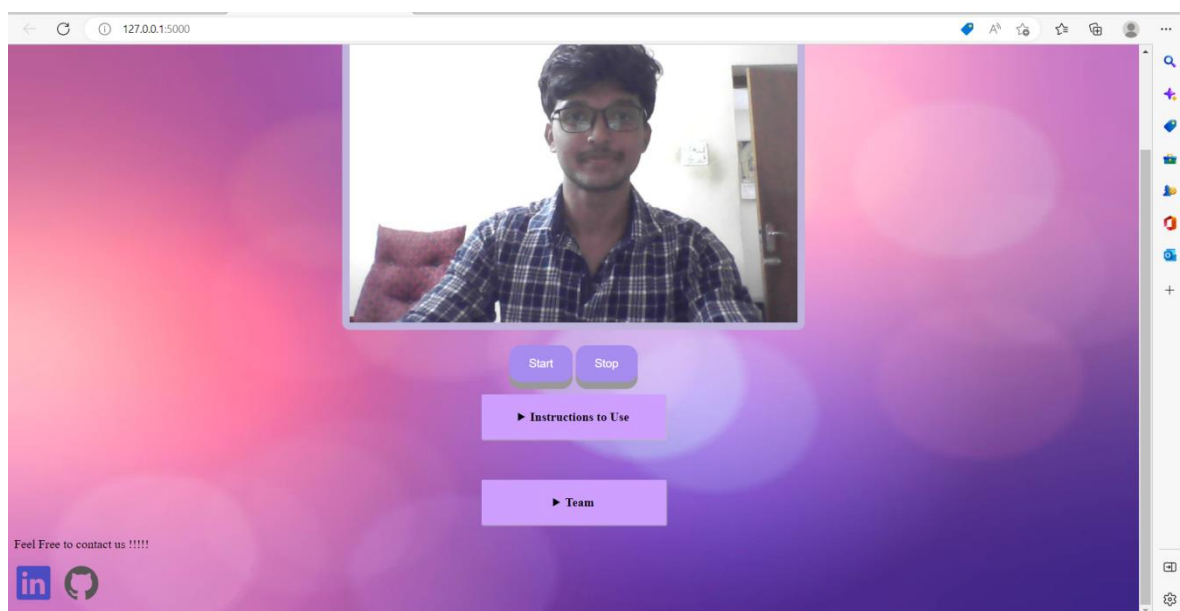
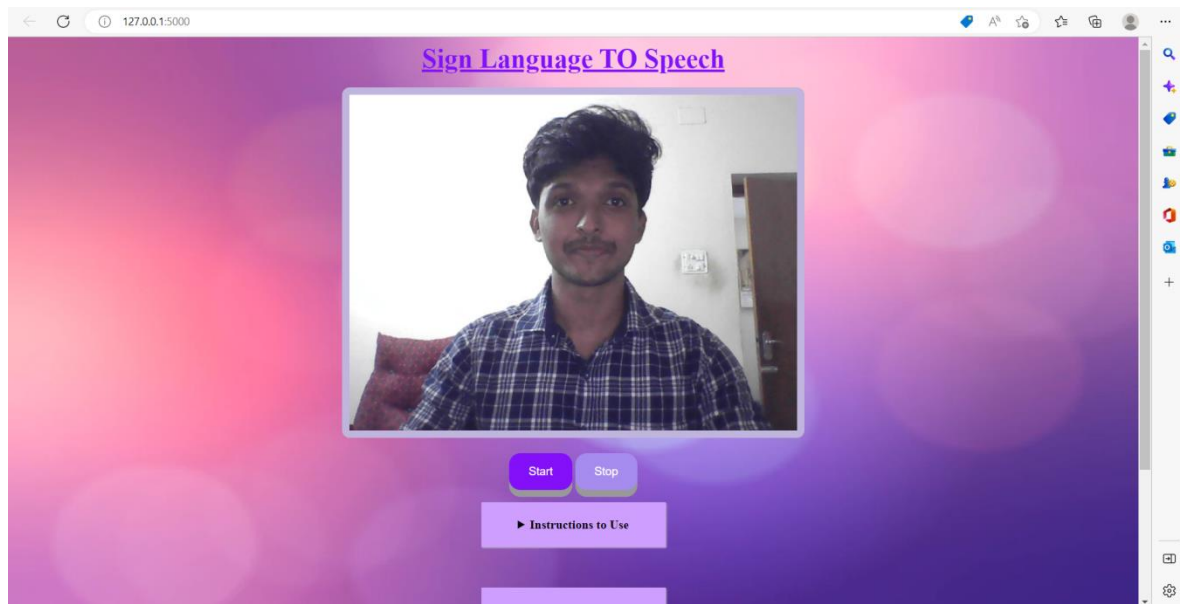
Without Css:



With Css:



Flask Output:



Final output Converted to voice:

