IBM-Project-5609-1658811745

Team ID: PNT2022TMID00980

Team Leader-Dhivagar T
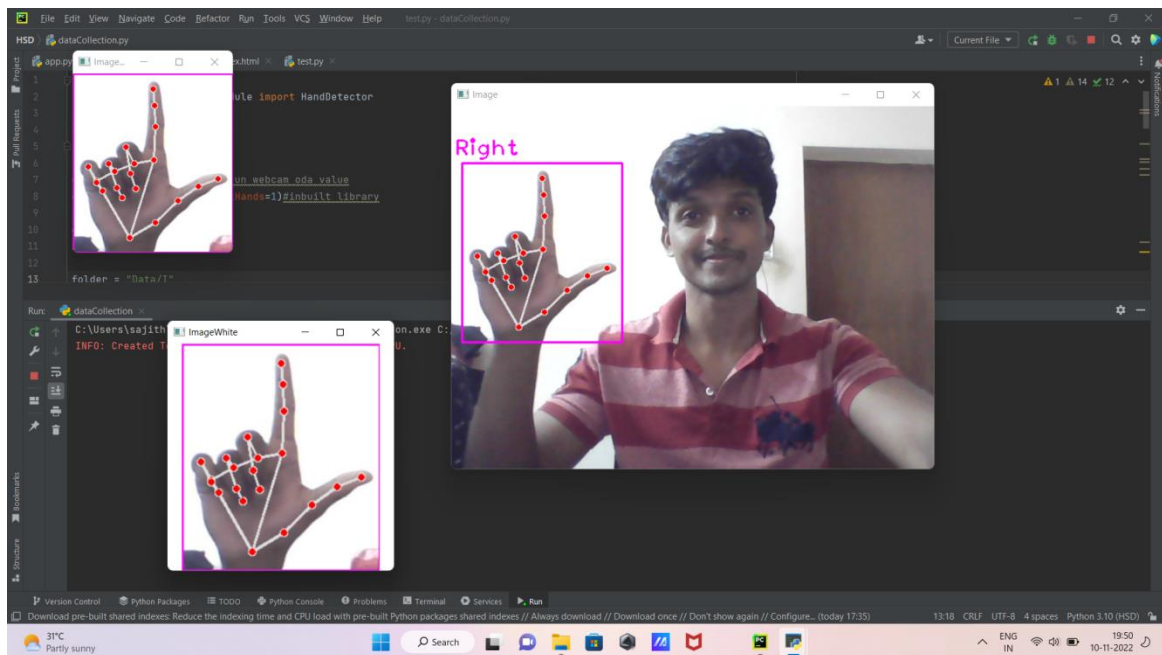
Team Member 1 – Balaji G

Team Member 2 – Dinesh kumar K

Team Member 3 –Dharanidharan

Team Member 4- Abubakkar Siddq M

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

```python
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
```

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

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

Applying Convolution ,Dense Layers:

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

```python
model = Sequential()
```

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

```python
model.add(MaxPooling2D(pool_size = (2,2))) #Pooled matrix
```

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

```python
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:

```python
img
```



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

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

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

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

```python
y
```

```
array([1])
```

```python
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']
```
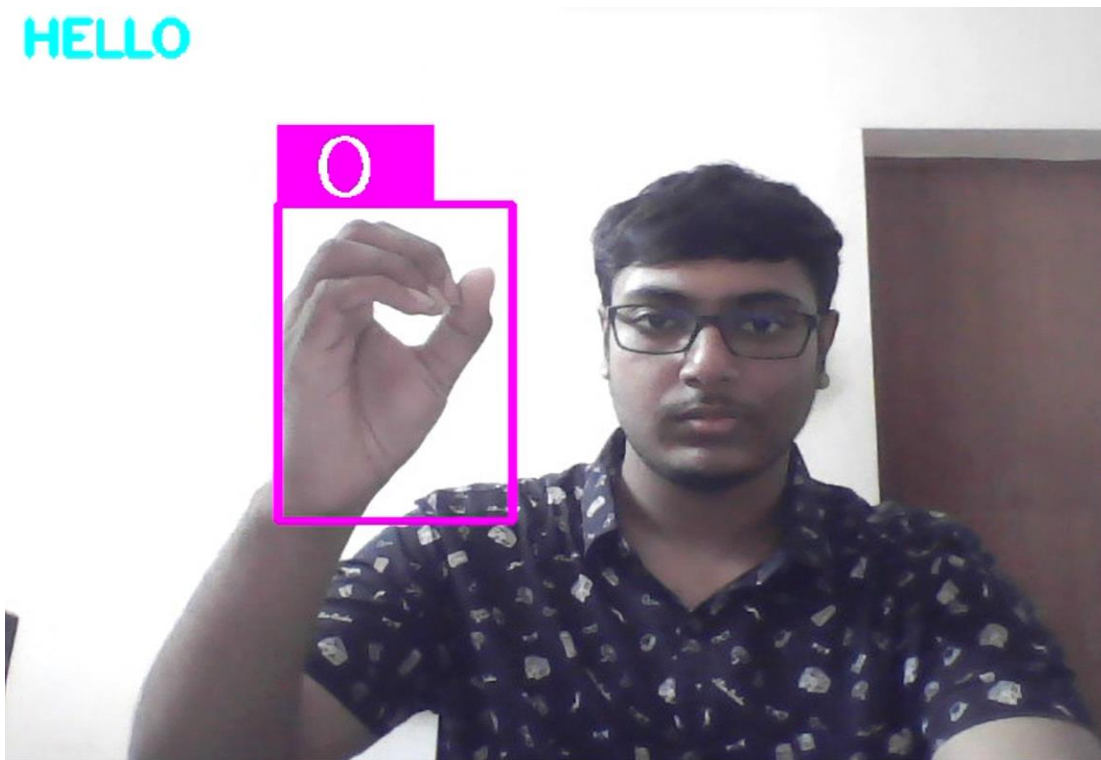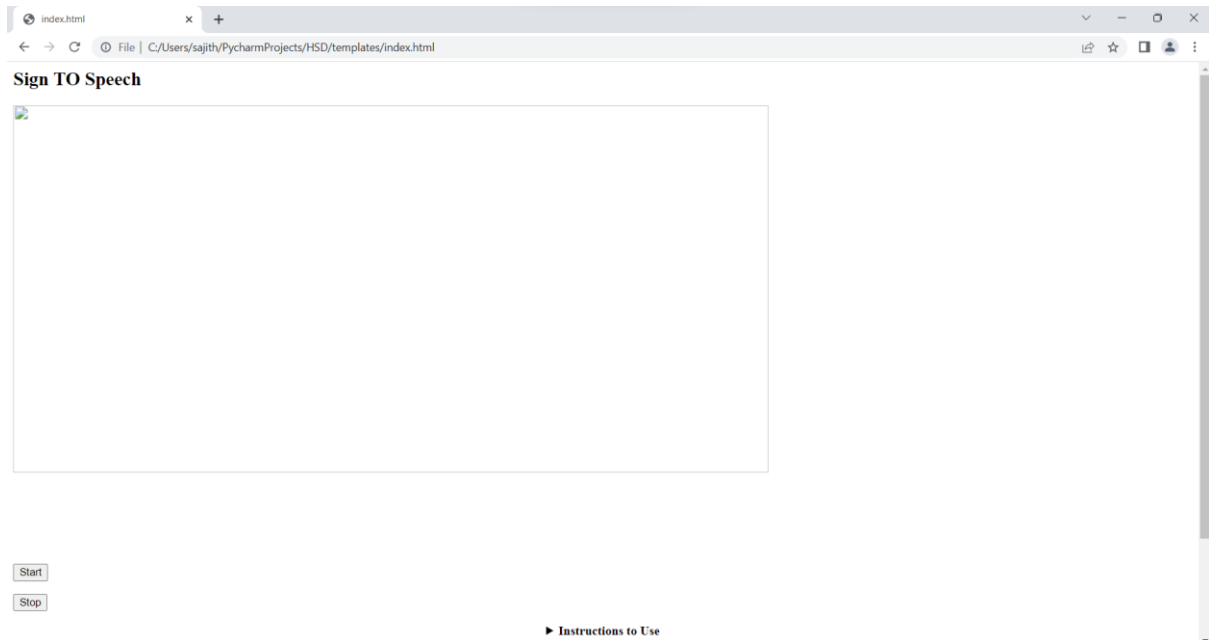
```python
index[y[0]]
```

```
'B'
```

Testing:

Saving text:
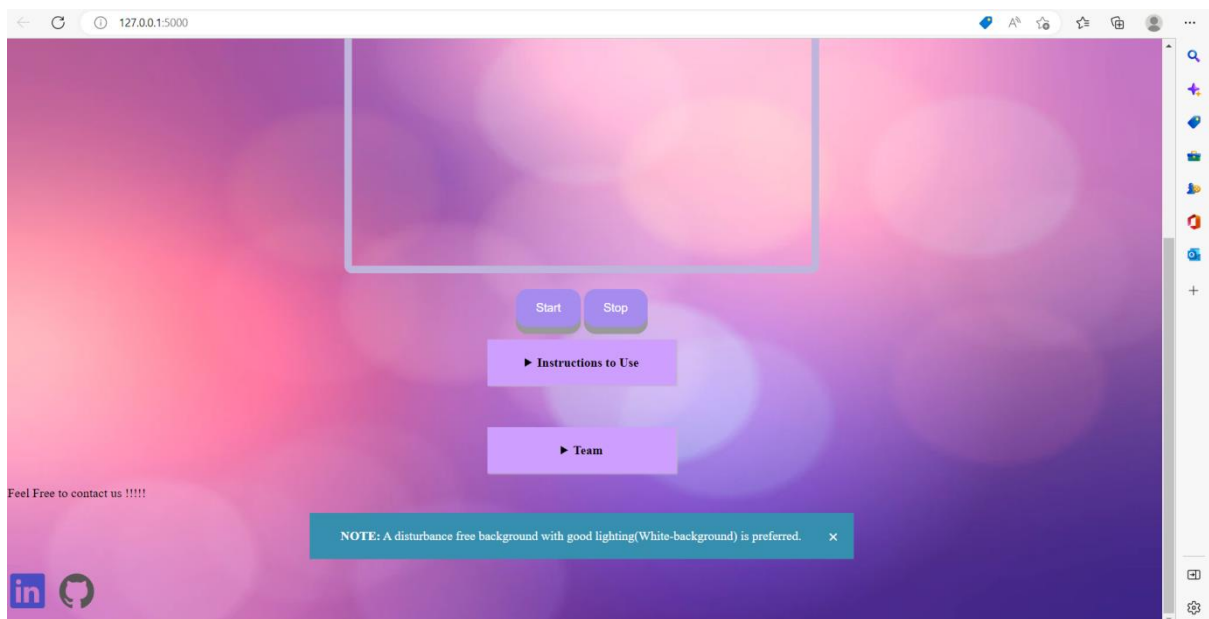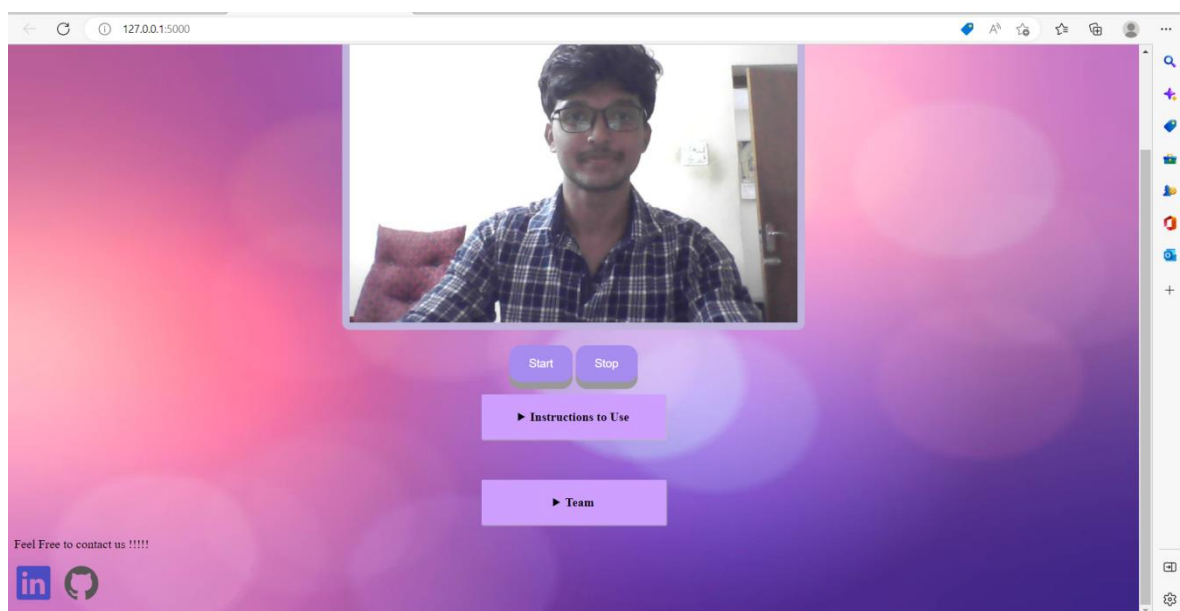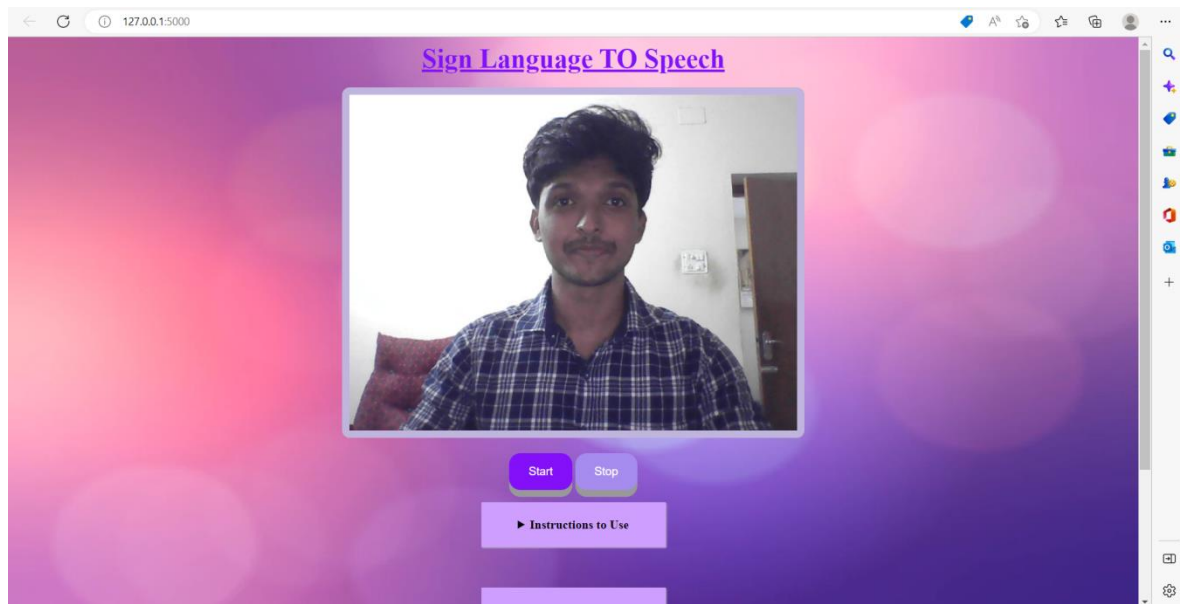
Building Flask Application:

Without Css:

With Css:

Flask Output:

Final output Converted to voice: