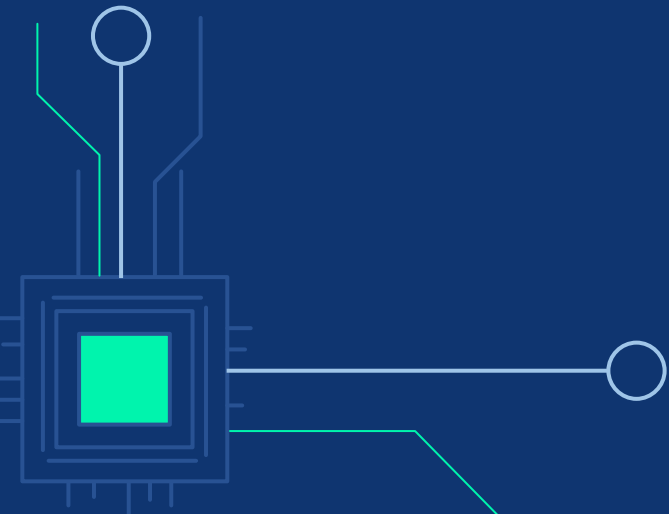# Hand Gesture Recognition
Team name: NAND

Team Members
Navateja Cherukuri
Durga Vara Prasad Reddy Maryada
Navya Lakshmi Kusam
Sai Venkata Anudeep Sanchula

Objective

➢ The main objective of this project is to recognize the hand gestures.

➢ The image is captured from the camera and the hand gesture captured in the image is predicted.

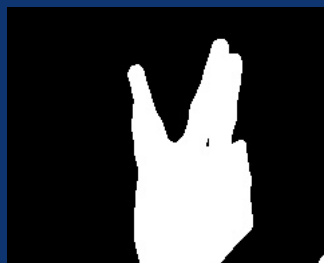➢ In this project we are going to train the data using deep learning modules
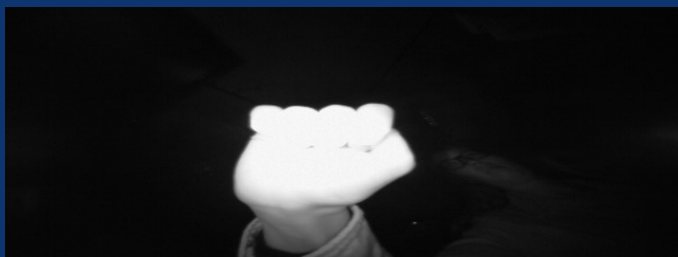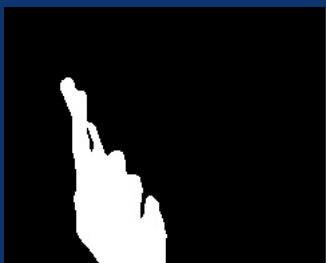
Dataset

Source : Kaggle

Datasets
1. Hand Gesture Classification
2. Sign Language for Numbers
3. Sign Language Gesture Images Dataset

Hand Gestures

Hand Gesture Categories

```python
category = {
    '0': 0,
    '1': 1,
    '2': 2,
    '3': 3,
    '4': 4,
    '5': 5,
    '6': 6,
    '7': 7,
    '8': 8,
    '9': 9,
    'C Shape': 10,
    'Call me': 11,
    'Down': 12,
    'fingers_crossed': 13,
    'Fist': 14,
    'Index': 15,
    'L': 16,
    'Moved Fist': 17,
    'Moved Palm': 18,
    'Ok Super': 19,
    'Palm': 20,
    'Paper Palm': 21,
    'Peace': 22,
    'Rock Fist': 23,
    'scissor': 24,
    'Thumb': 25,
    'unknown_numbers': 26,
    'up': 27,
    'YO-YO': 28
}
```
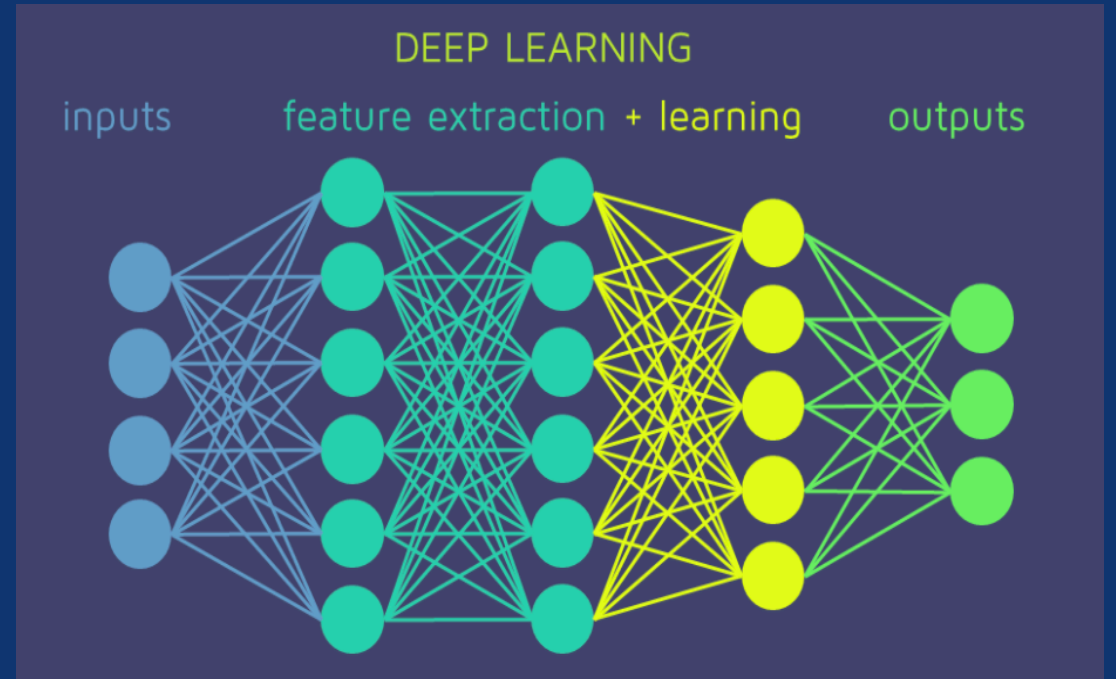
Technology and Libraries

Technology
- Deep Learning

Libraries
- NumPy and Pandas for preprocessing the data
- Open-CV to read the image
- Keras for training the data and visualizing the model

DEEP LEARNING

inputs   feature extraction + learning   outputs

Importing The data

There are total 28 categories.

150 Images are loaded from each category for training the data.

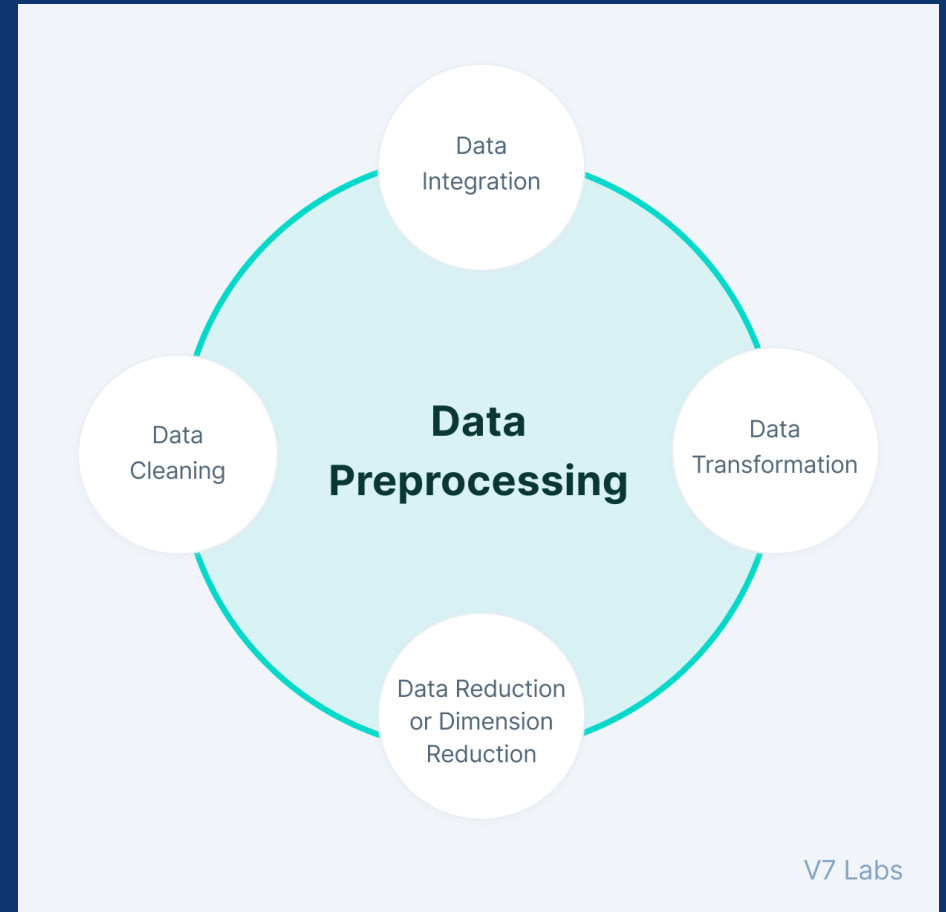50 images are loaded from each category for testing the data.

Data pre-processing

Converting data types

Reshaping

Scaling the values between 0 and 1

Transforming target data to binary class matrix

Model Summary

```
Model: "sequential"
_____
 Layer (type)                 Output Shape              Param #
===============================================================
 conv2d (Conv2D)              (None, 58, 58, 32)        832

 max_pooling2d (MaxPooling2D  (None, 29, 29, 32)        0
 )

 conv2d_1 (Conv2D)            (None, 27, 27, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 13, 13, 64)        0
 2D)

 conv2d_2 (Conv2D)            (None, 11, 11, 64)        36928

 max_pooling2d_2 (MaxPooling  (None, 5, 5, 64)          0
 2D)

 flatten (Flatten)           (None, 1600)               0

 dense (Dense)               (None, 128)                204928

 dense_1 (Dense)             (None, 29)                 3741


===============================================================
```

```
===========================
Total params: 264,925
Trainable params: 264,925
Non-trainable params: 0
```

Model Flow Diagram and Layered View
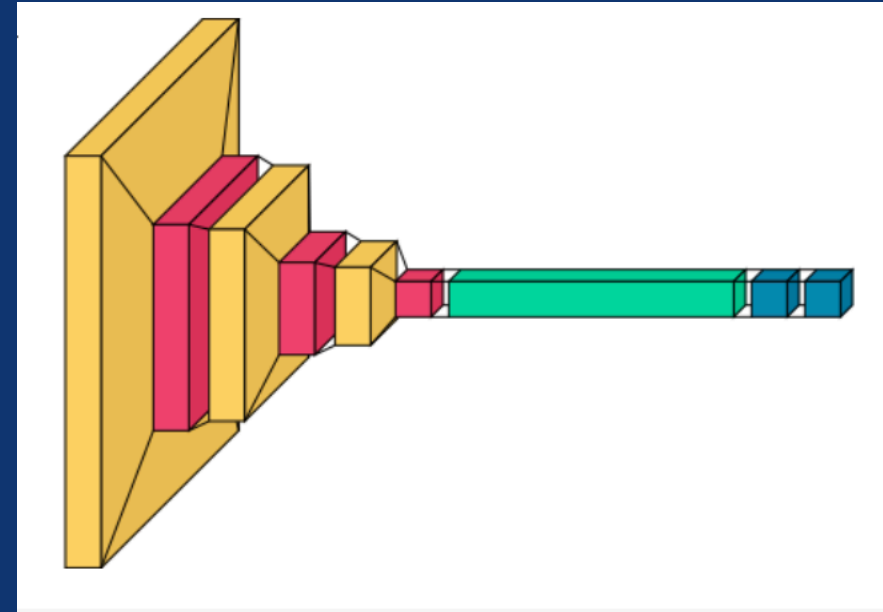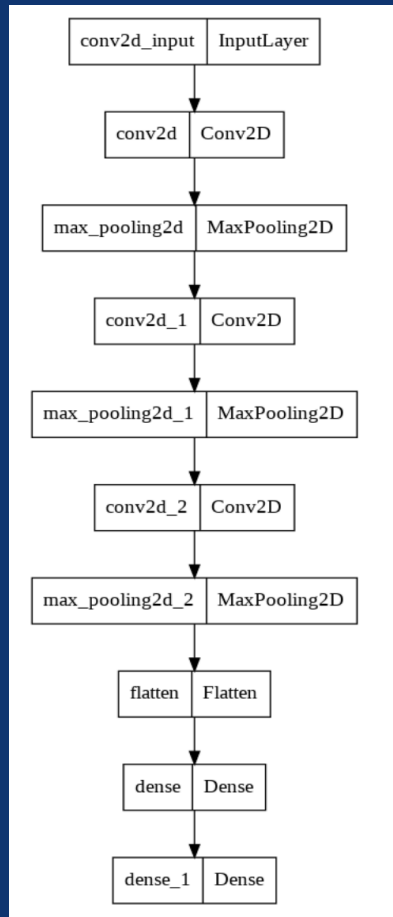
Training the Data

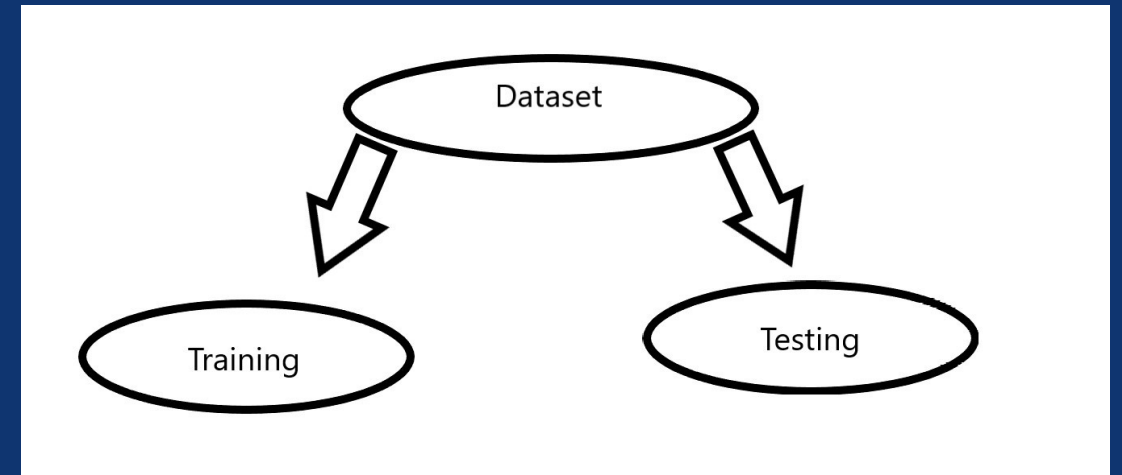4650 samples for train data

1550 samples for test data

Number of epochs is 10

Model Performance

```
Epoch 1/10
37/37 [==============================] - 23s 610ms/step - loss: 2.4259 - accuracy: 0.3013 - val_loss: 1.5321 - val_accuracy: 0.5471
Epoch 2/10
37/37 [==============================] - 22s 603ms/step - loss: 1.0763 - accuracy: 0.6581 - val_loss: 1.1496 - val_accuracy: 0.6406
Epoch 3/10
37/37 [==============================] - 22s 597ms/step - loss: 0.6784 - accuracy: 0.7824 - val_loss: 0.9037 - val_accuracy: 0.7232
Epoch 4/10
37/37 [==============================] - 22s 590ms/step - loss: 0.4807 - accuracy: 0.8482 - val_loss: 0.8520 - val_accuracy: 0.7432
Epoch 5/10
37/37 [==============================] - 22s 591ms/step - loss: 0.3534 - accuracy: 0.8860 - val_loss: 0.9408 - val_accuracy: 0.7381
Epoch 6/10
37/37 [==============================] - 22s 584ms/step - loss: 0.3096 - accuracy: 0.8985 - val_loss: 0.8269 - val_accuracy: 0.7884
Epoch 7/10
37/37 [==============================] - 22s 588ms/step - loss: 0.2202 - accuracy: 0.9357 - val_loss: 0.9110 - val_accuracy: 0.7858
Epoch 8/10
37/37 [==============================] - 22s 589ms/step - loss: 0.1762 - accuracy: 0.9499 - val_loss: 0.8007 - val_accuracy: 0.8103
Epoch 9/10
37/37 [==============================] - 22s 599ms/step - loss: 0.1289 - accuracy: 0.9589 - val_loss: 0.9279 - val_accuracy: 0.8232
Epoch 10/10
37/37 [==============================] - 22s 593ms/step - loss: 0.1003 - accuracy: 0.9731 - val_loss: 0.9779 - val_accuracy: 0.8148
```
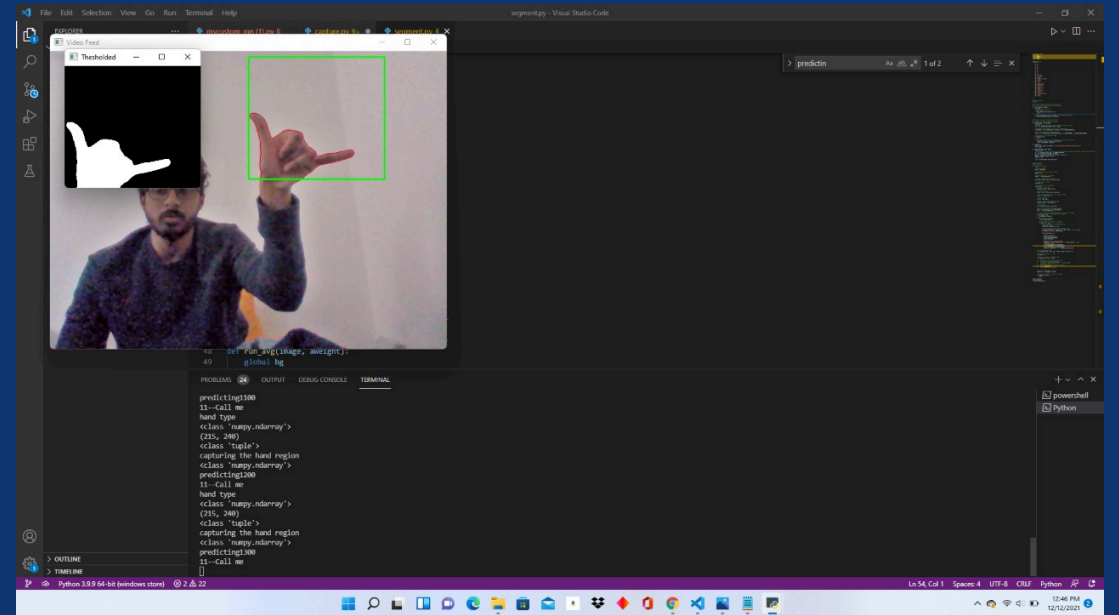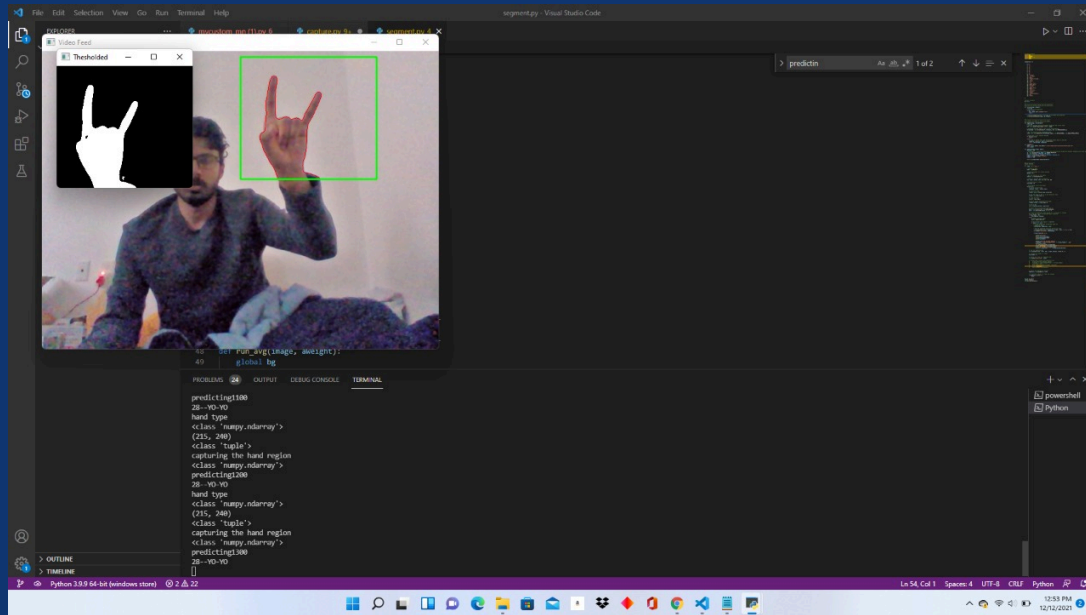
```python
# The accuracy of the model is 76%. We have tried by adding some layers but this has given us the best accuracy
scores = model.evaluate(X_test, y_test, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))

Accuracy: 76.90%
```
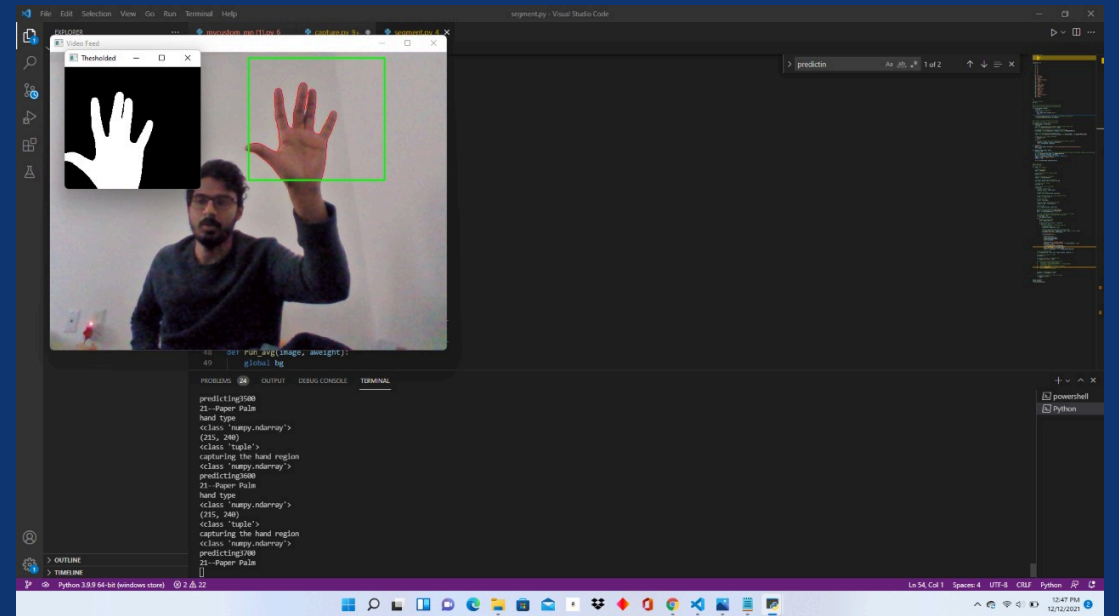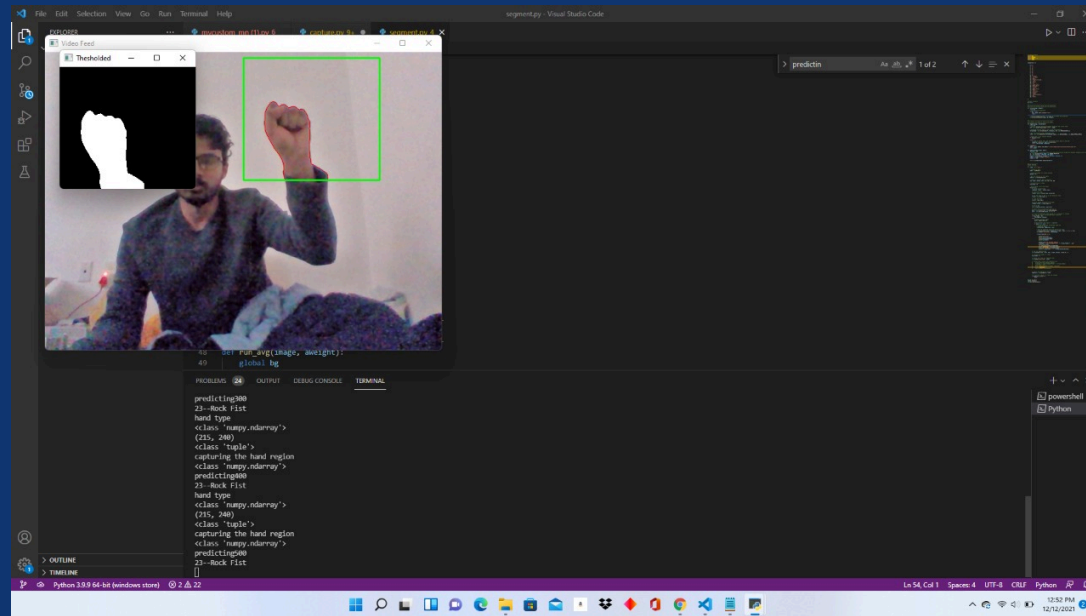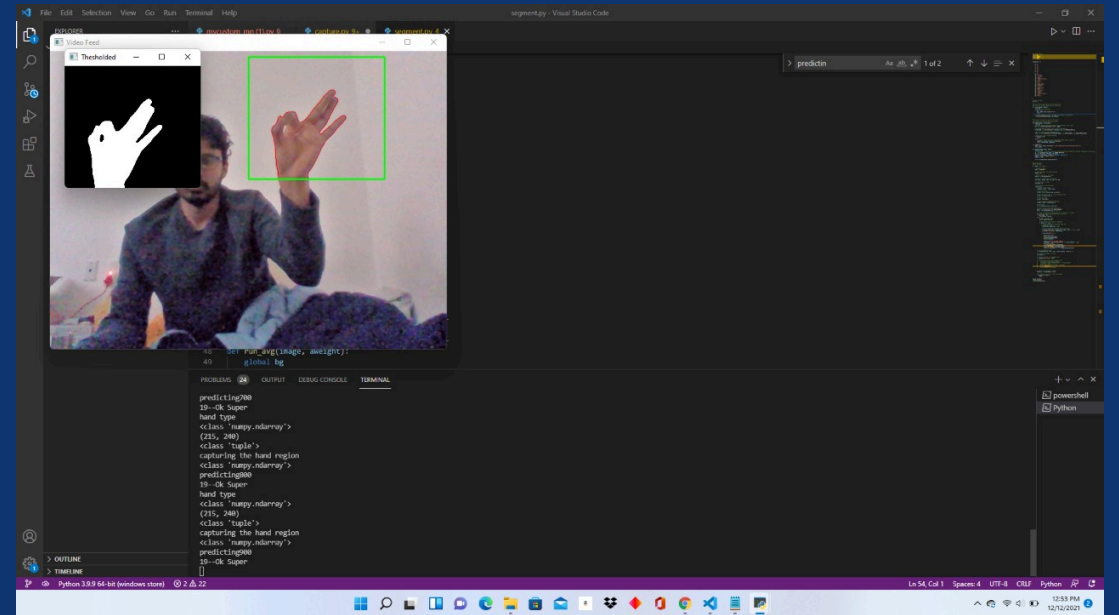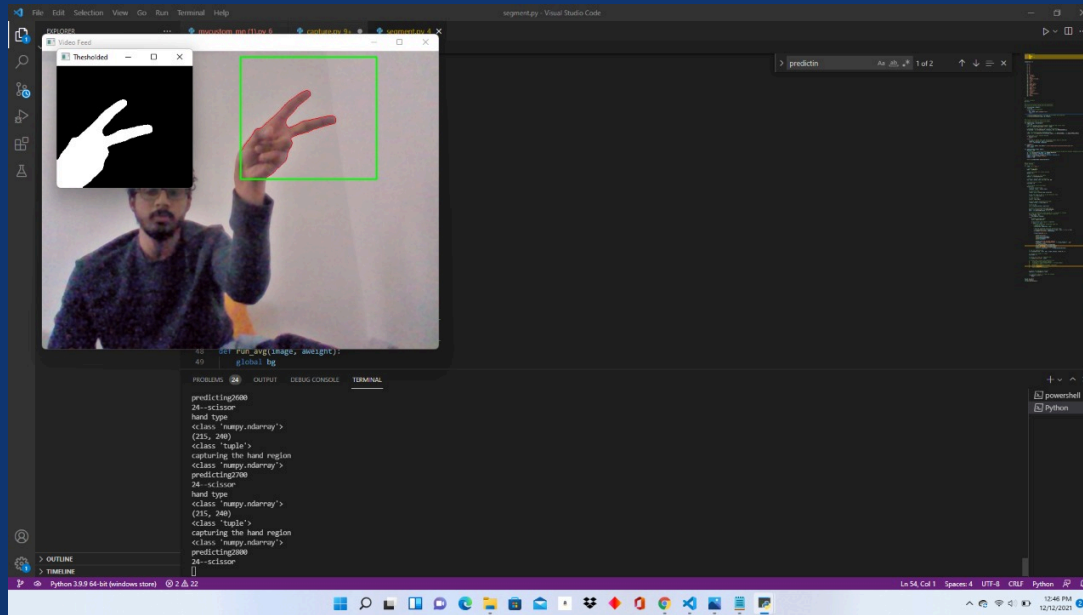
User Interface
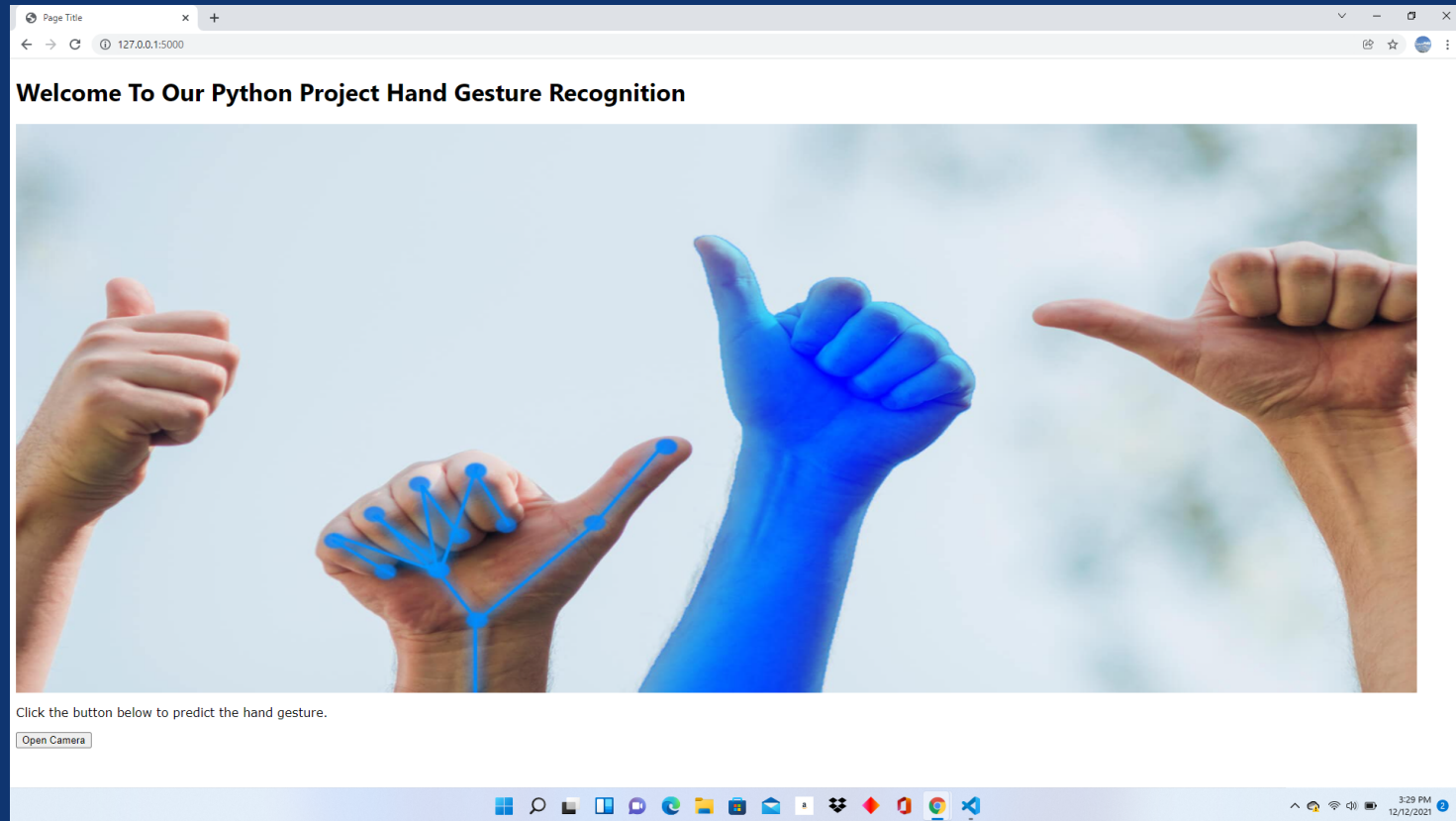
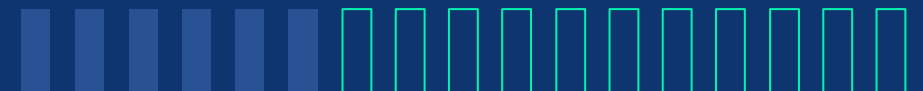# Predictions from the Model
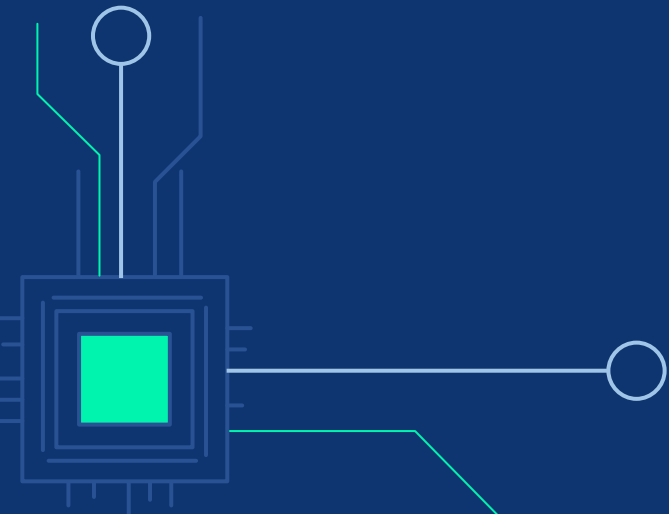
Predictions from the Model

Predictions from the Model

Future Scope

This project can be further developed to aid handicapped people for day-to-day activities from their self defined gesture configurations.

Any Questions?

Thank You