Pedestrian Sign Detection

Team: S Sankara Subramanian, Muhammed Kaif,Sanjeev Kumar Upadhyay, Arun

Problem Statement

- Develop a solution for recognizing the hand signals provided by pedestrians and take appropriate actions on the Autonomous Vehicle.

Solution

- We propose to create a program to understand normal message signals and make the Autonomous Vehicle take appropriate measures.

How it works

- The video is captured with the help of OpenCV. Then keypoints from various elements are extracted using mediapipe.
- These keypoints and landmarks are used to train the new model.
- By using this model, the signal that the pedestrian is showing is interpreted.

Extraction of keypoints using mediapipe

- First, the video is captured with the help of OpenCV.
- The keypoints are extracted from the face, pose, left and right hand with the help of mediapipe.
- Each of these keypoints are connected with the help of mediapipe drawing utilities and the Holistic model is created.

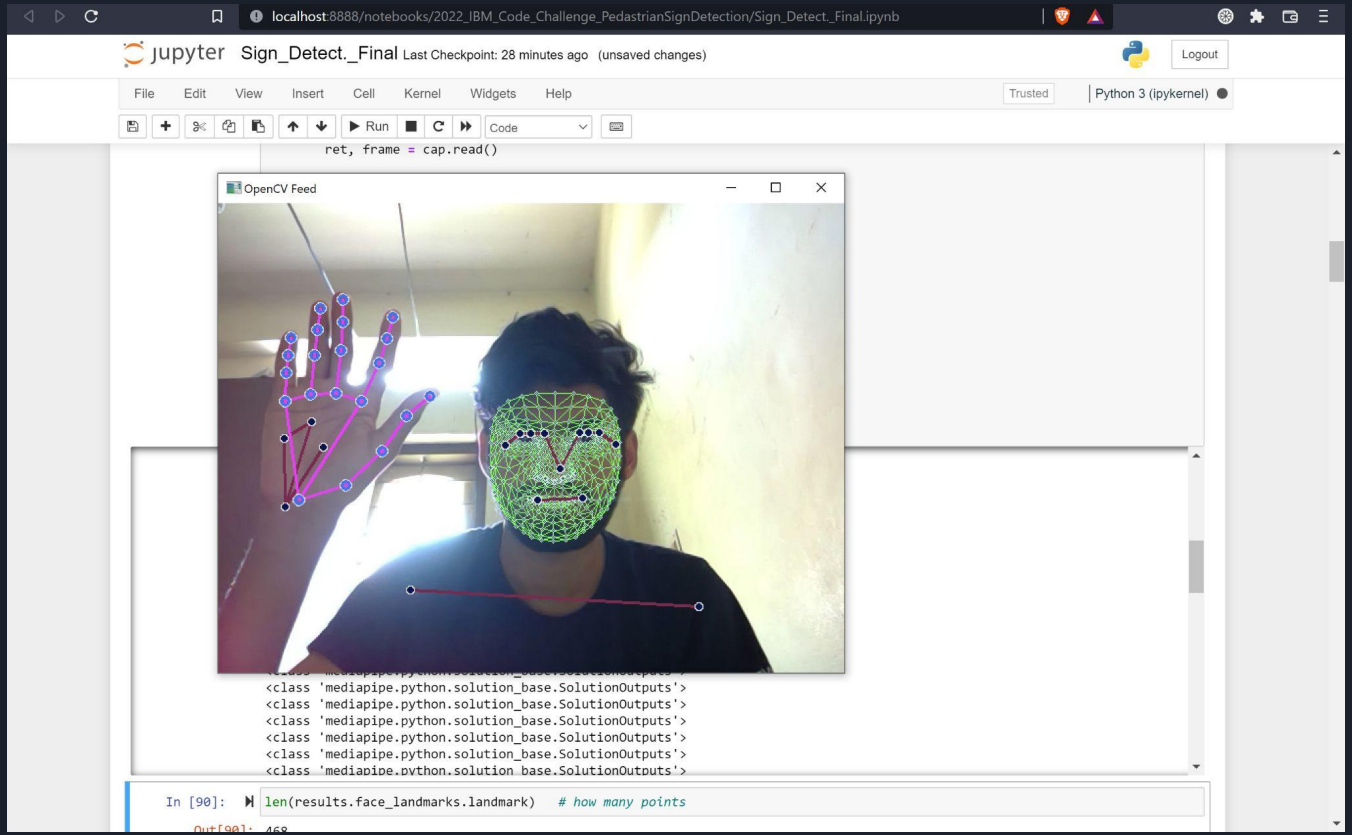

Fig 1. Extracting Keypoints using Mediapipe.

Build and Train LSTM Neural Network

- Build the LSTM model and used for training.

```
Model: "sequential_22"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_64 (LSTM)               (None, 30, 64)            442112
_________________________________________________________________
lstm_65 (LSTM)               (None, 128)               98816
_________________________________________________________________
dense_63 (Dense)             (None, 64)                8256
_________________________________________________________________
dense_64 (Dense)             (None, 32)                2080
_________________________________________________________________
dense_65 (Dense)             (None, 2)                 66
=================================================================
Total params: 551,330
Trainable params: 551,330
Non-trainable params: 0
_________________________________________________________________
```

Fig 2. Model Parameters.

Evaluation using Confusion Matrix and Accuracy

- Model is evaluated using Confusion Matrix and accuracy is analyzed.

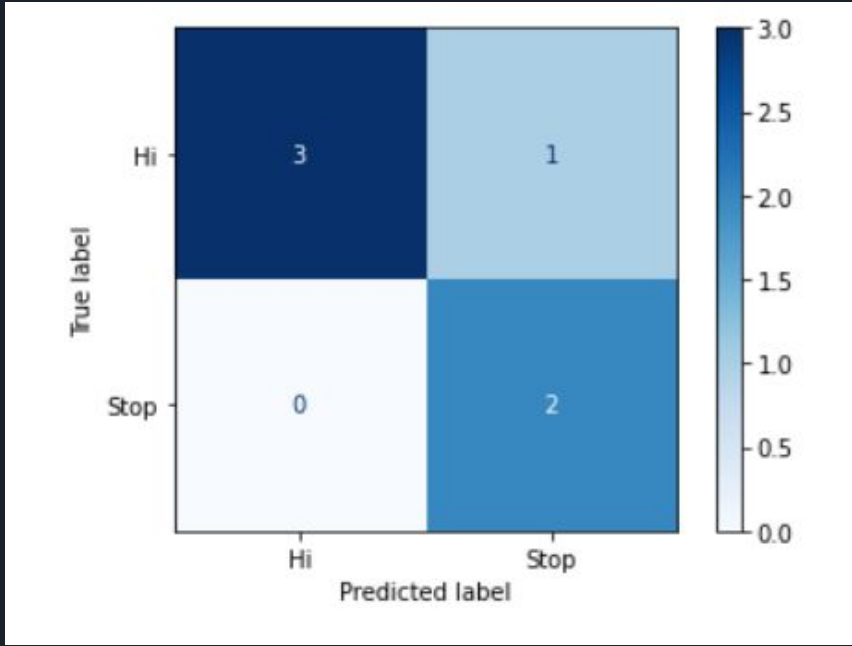

Fig 3. The Confusion Matrix.

Test in Real Time

- The results are tested in Real Time.

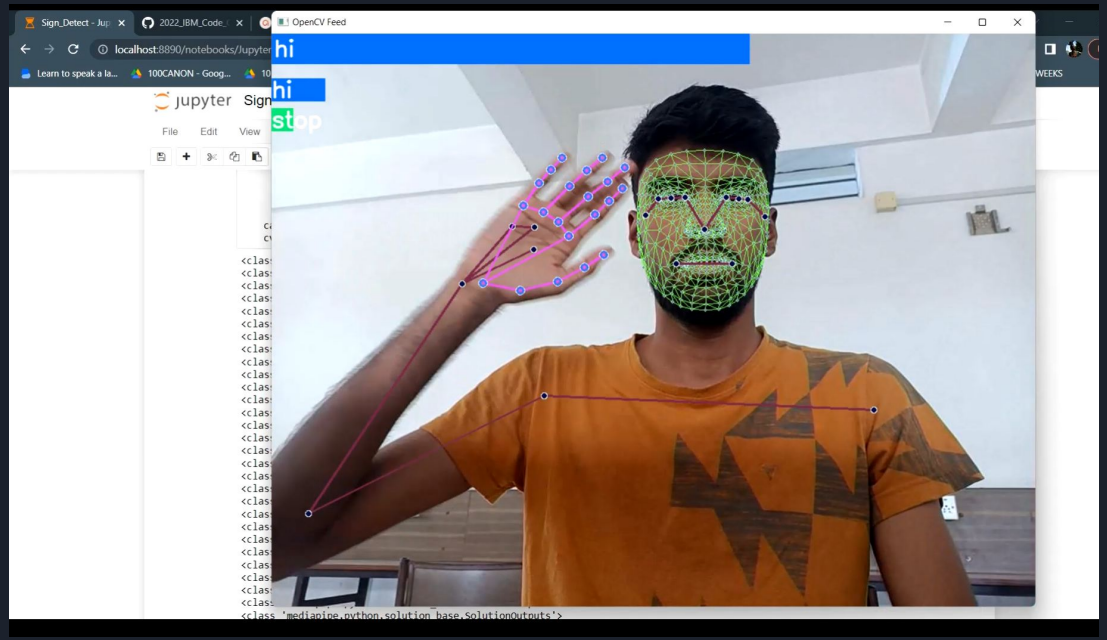

Fig 4. Testing Of ‘HI’ signal.

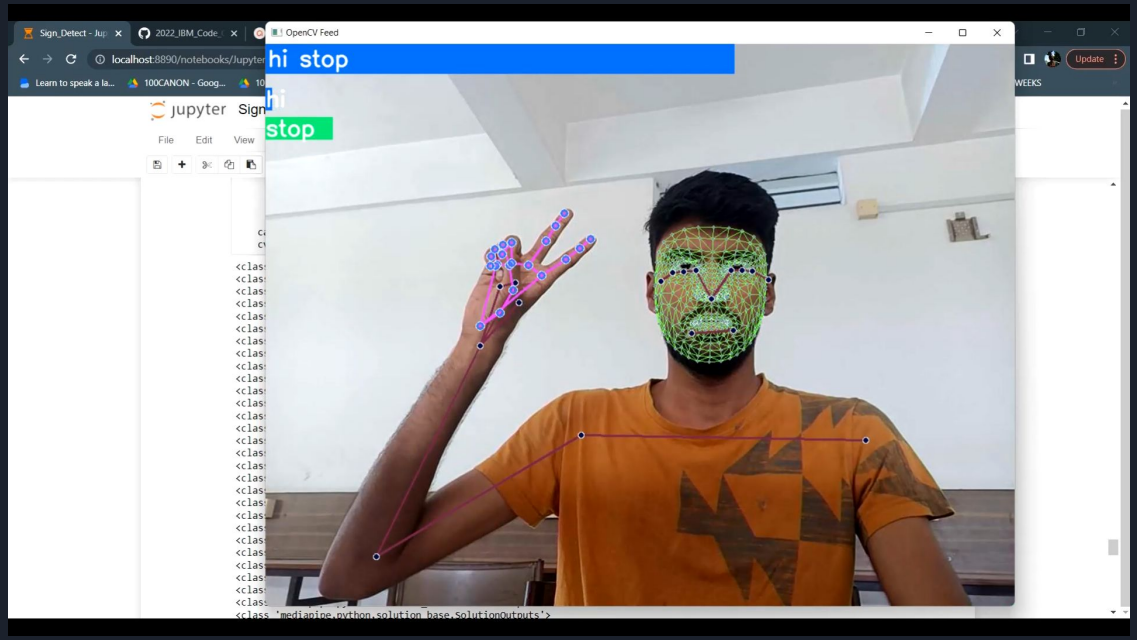

Fig 5. Testing of ‘STOP’ signal.

Tools Used

Jupyter Notebook

Python

Tensorflow

Opencv

Keras

Scikit learn

Links

Github :
https://github.com/S-Sankara/2022_IBM_Code_Challenge_PedastrianSignDetection.git

Readme:
https://github.com/S-Sankara/2022_IBM_Code_Challenge_PedastrianSignDetection/blob/main/README.md

Contributions

Sanjeev - Collection of Dataset.

Muhammed Kaif - Extraction of keypoints using mediapipe.

S Sankara Subramanian - Model building and training.

Arun - Verification and modification.

Thank you!