HAND GESTURE RECOGNITION

PROJECT SYNOPSIS

OF MAJOR PROJECT

BACHELOR OF TECHNOLOGY

COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

YASH ARYAN (06816403220)

FEBRUARY 2024

UNDER GUIDANCE OF

Dr. PRIYANKA BHUTANI

University School of Information, Communication and Technology

Guru Gobind Singh Indraprastha University, Delhi

TABLE OF CONTENT

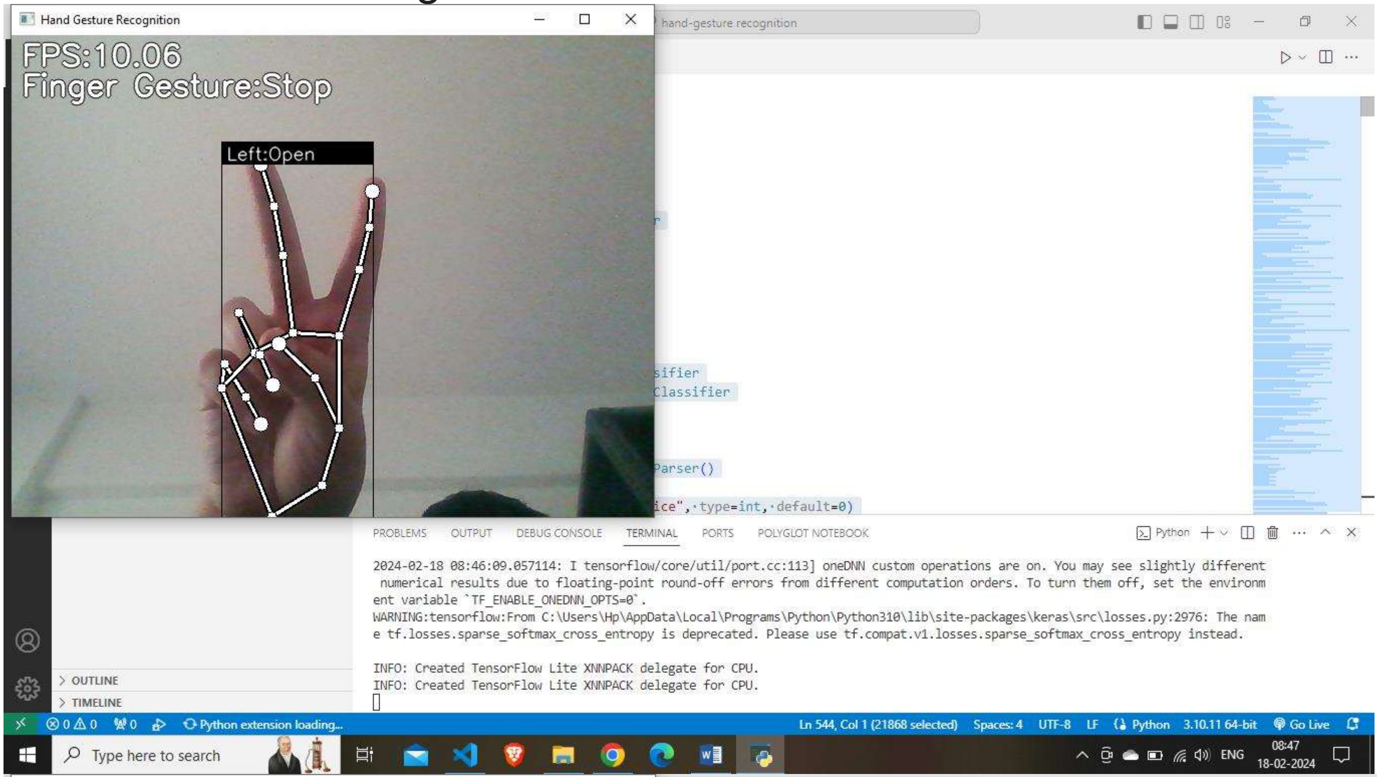
OBJECTIVE

The objectives of making a hand gesture recognition project using Mediapipe and OpenCV are:

1. Real-time Gesture Recognition: Developing a system capable of accurately recognizing hand gestures in real-time from camera feed.
2. Gesture Classification: Building a model that can classify different hand gestures into predefined categories or commands.
3. Human-Computer Interaction: Enabling natural and intuitive interactions between humans and computers or devices through hand gestures.
4. Accessibility: Creating interfaces that allow users with disabilities or limitations to interact with technology more easily through gestures, without relying solely on traditional input methods like keyboards or mice.

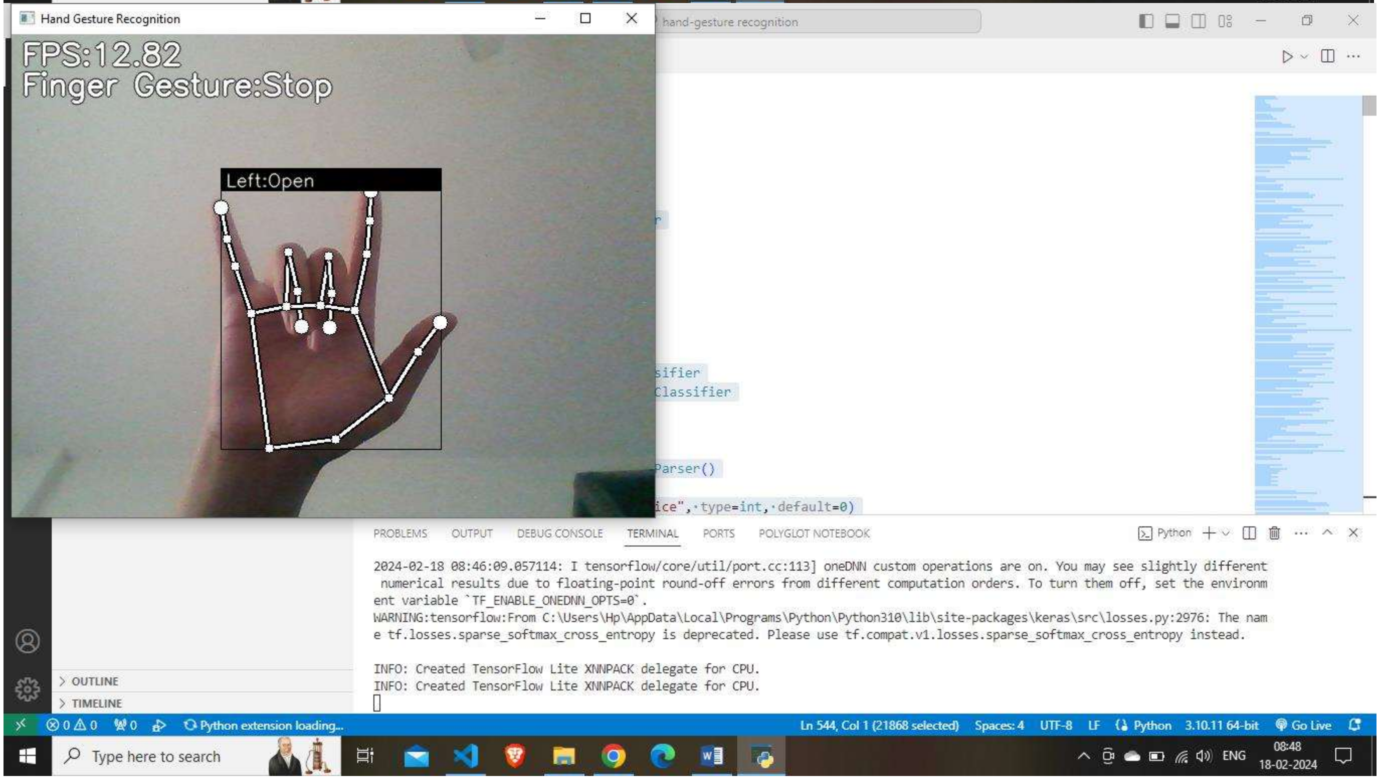
OVERVIEW

This project uses Mediapipe, OpenCV and Tensorflow for recognizing the hand gestures. This project was made using Python. When a user makes a gesture, it detects the hand gesture, recognizes it and displays the frames per second along with the detected gesture to the user.



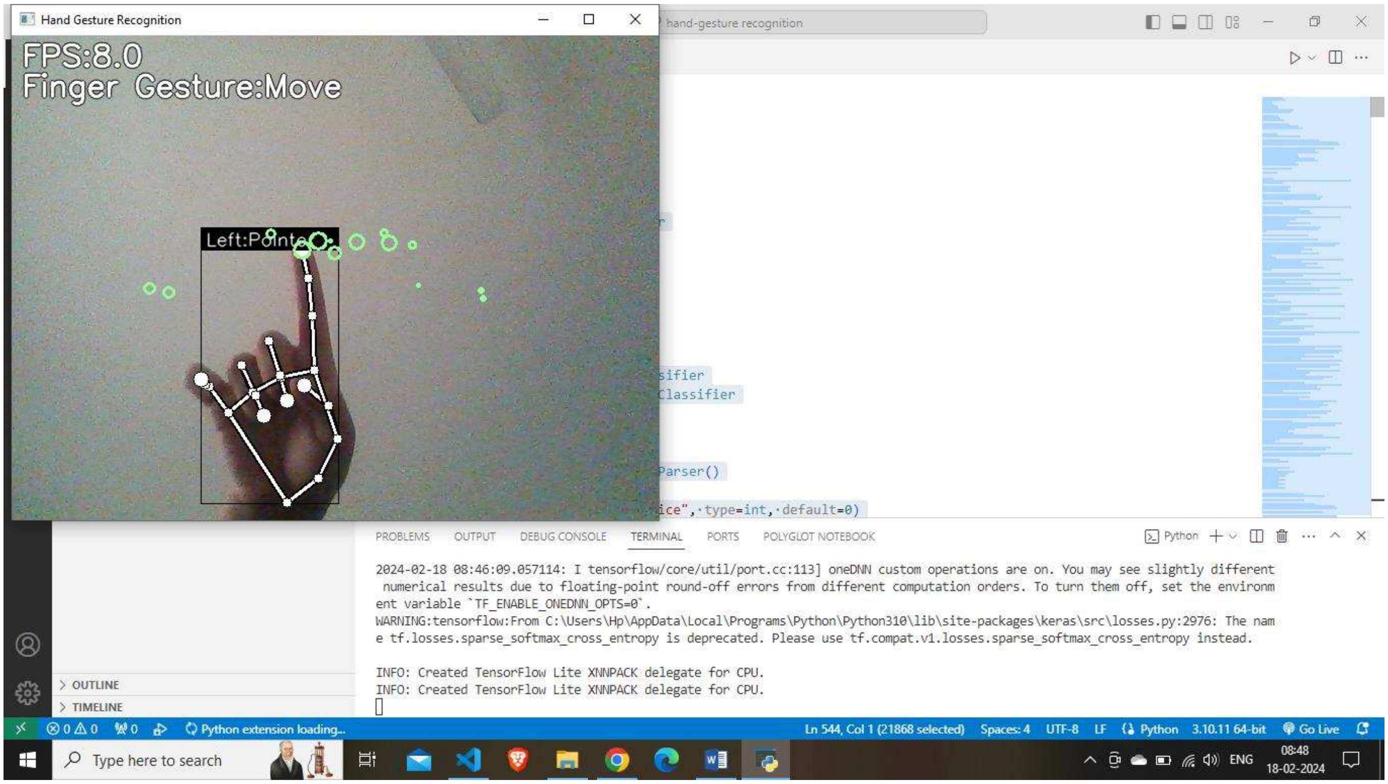
```
FPS:10.06
Finger Gesture:Stop
Left:Open
```

```
2024-02-18 08:46:09.057114: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different
 numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environm
ent variable `TF_ENABLE_ONEDNN_OPTS=0`.
WARNING:tensorflow:From C:\Users\Hp\AppData\Local\Programs\Python\Python310\lib\site-packages\keras\src\losses.py:2976: The nam
e tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.

INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
```

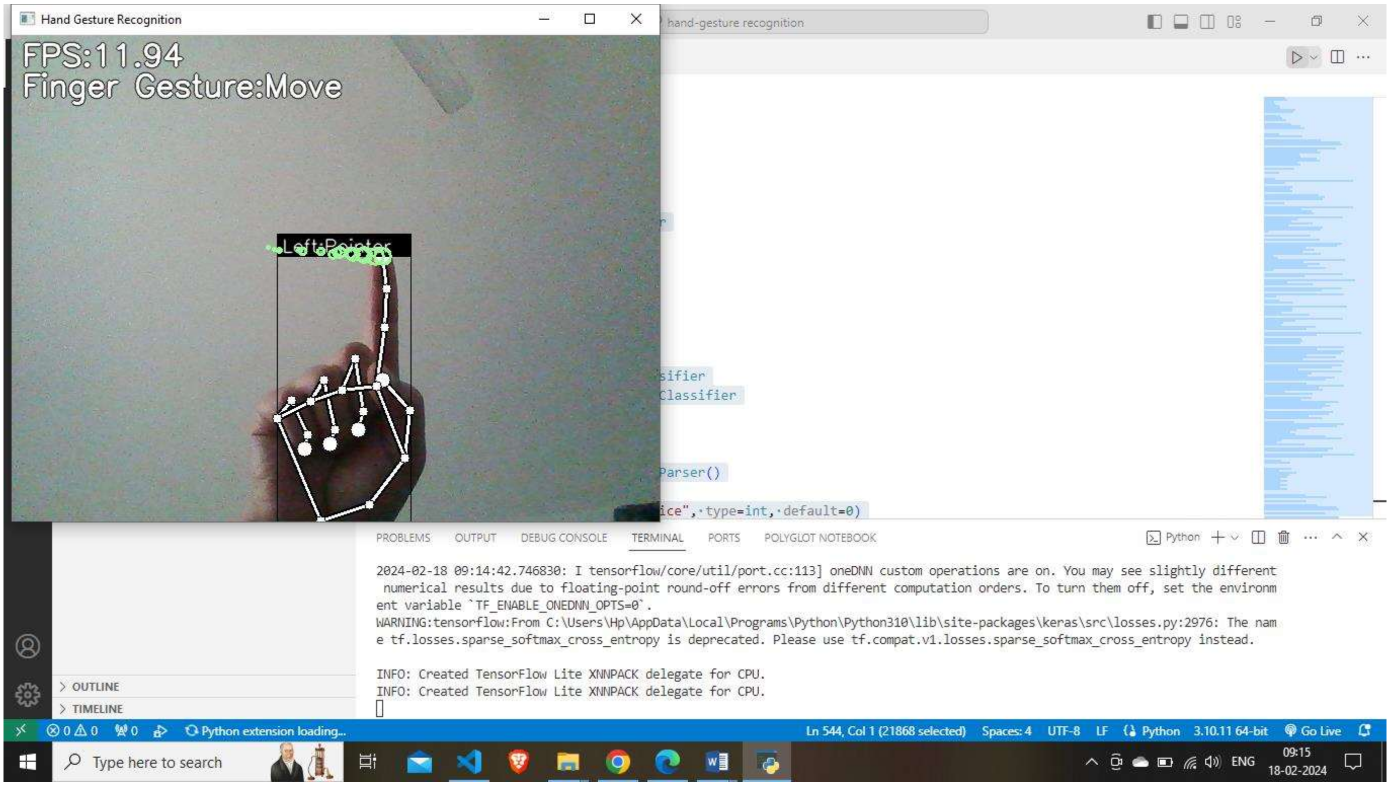


```
FPS:12.82
Finger Gesture:Stop
Left:Open
```

```
2024-02-18 08:46:09.057114: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different
 numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environm
ent variable `TF_ENABLE_ONEDNN_OPTS=0`.
WARNING:tensorflow:From C:\Users\Hp\AppData\Local\Programs\Python\Python310\lib\site-packages\keras\src\losses.py:2976: The nam
e tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.

INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
```



```
2024-02-18 08:46:09.057114: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different
 numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environm
ent variable `TF_ENABLE_ONEDNN_OPTS=0`.
WARNING:tensorflow:From C:\Users\Hp\AppData\Local\Programs\Python\Python310\lib\site-packages\keras\src\losses.py:2976: The nam
e tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.

INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
```



```
2024-02-18 09:14:42.746830: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different
 numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environm
ent variable `TF_ENABLE_ONEDNN_OPTS=0`.
WARNING:tensorflow:From C:\Users\Hp\AppData\Local\Programs\Python\Python310\lib\site-packages\keras\src\losses.py:2976: The nam
e tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.

INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
```

SCOPE

This project has a broad scope and can be used for a variety of tasks, including:

1. Gesture Recognition Tasks: Define the specific hand gestures to be recognized within the project. These could include simple gestures like thumbs-up, pointing, waving, or more complex gestures for controlling devices, interacting with virtual environments, or conveying specific commands.

2. Data Preprocessing: Preprocess the collected data to enhance its quality and prepare it for training machine learning models. This may involve tasks such as resizing images, normalization, noise reduction, and augmentation to increase dataset diversity.

3. Feature Extraction: Extract relevant features from the preprocessed data that capture important information about hand gestures. This could include hand keypoints, motion trajectories, spatial-temporal features, or any other relevant descriptors.

4. Model Selection and Training: Choose an appropriate machine learning or deep learning model architecture for hand gesture recognition. Train the selected model using the preprocessed data, optimizing model parameters to minimize a chosen loss function while maximizing accuracy on a validation dataset.

5. Real-time Implementation: Develop a real-time hand gesture recognition system capable of processing video or camera feed inputs and recognizing gestures in real-time. Implement the necessary logic for interpreting recognized gestures and triggering appropriate actions in the target application or system.

RESOURCES

HARDWARE RESOURCES:

- A computer with a modern processor like i7-10700 and graphics card.
- A minimum of 8GB of RAM.
- A hard drive or solid-state drive with at least 10GB of free space.

SOFTWARE RESOURCES:

- mediapipe 0.8.1
- OpenCV 3.4.2 or Later
- Tensorflow 2.3.0 or Later
- tf-nightly 2.5.0.dev or later (Optional)
- scikit-learn 0.23.2 or Later (Optional)
- matplotlib 3.3.2 or Later (Optional)

CONCLUSION

This project uses Mediapipe, OpenCV and Tensorflow for recognizing the hand gestures. This project was made using Python. When a user makes a gesture, it detects the hand gesture, recognizes it and displays the frames per second along with the detected gesture to the user.

REFERENCES

[1] Paulo Trigueiros, “Computer Vision and Machine Learning based Hand Gesture Recognition”, 2015

[2] Geron Aurelien, “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow”, 2022

[3] Nishant Shukla, “Machine Learning with TensorFlow”, 2018

[4] Adrian Kaehler, “Learning OpenCV”, 2008

[5] Andreas Muller, “Introduction to Machine Learning with Python”, 2016