**Smart Fingers Tracking System Based on open cv using Mediapipe And Flask**

**Project Team:**

1:Name: Faraz Abbasi Roll No: 2K20/ITE/36

2:Name: Fayaz Panhwar Roll No : 2K20/ITE/39

3: Name: Shoukat Roll No: 2K20/ITE/126

Class: Bs(IT PART-FOUR) Seventh semester

Subject: Computer Vision

Submitted To : Dr Sandar Ali Khowaja

**Project Report**

Introduction:

The goal of this project is to develop a real-time hand gesture recognition system using MediaPipe and Flask. The system uses computer vision techniques to detect and track hand landmarks, and then analyzes the hand gestures to determine the number of fingers held up. The application is implemented using the Python programming language.

Technologies Used:

* Python: The programming language used for the implementation.
* OpenCV (cv2): A popular computer vision library used for image and video processing.
* MediaPipe: A cross-platform framework developed by Google for building multimodal applied machine learning pipelines.
* Flask: A micro web framework used to create the web application for streaming the video feed.

System Overview:

The system consists of the following components:

1. Importing Libraries:

The necessary libraries, including cv2, mediapipe, and Flask, are imported to enable the functionality required for the project.

2. Setting up MediaPipe and Hands Model:

* The MediaPipe Hands module is initialized to enable hand detection and tracking.
* An instance of the Hands model is created using `mpHands.Hands()`.

3. Setting up the Flask Application:

* A Flask application is created using `Flask(\_\_name\_\_)`.
* The root route ('/') is defined to render the index.html template.

4. Implementing Video Streaming:

* `generate\_frames()` function is defined to read frames from the webcam in a continuous loop.
* Each frame is processed using MediaPipe to detect and track hand landmarks.
* The hand landmarks are then used to determine the number of fingers held up.
* The annotated frames are encoded as JPEG images and yielded to the Flask application.

5. Rendering the Video Feed:

* The `video\_feed()` function is defined to return the generated frames as a multipart response.
* The response's mimetype is set to 'multipart/x-mixed-replace' to enable continuous streaming of the video feed.

6. Running the Application:

* The Flask application is started by calling `app.run(debug=True)`.

Conclusion:

This project demonstrates the implementation of a real-time hand gesture recognition system using MediaPipe and Flask. The system detects and tracks hand landmarks in video frames captured from a webcam and analyzes the hand gestures to determine the number of fingers held up. The Flask application provides a web interface for streaming the video feed with the annotated hand landmarks and finger count. This project can be further expanded to recognize and interpret various hand gestures for applications such as sign language recognition or gesture-based control systems.