**Hand Gesture Recognition with OpenCV and MediaPipe**

This project utilizes **OpenCV** and **MediaPipe** to create a real-time hand gesture recognition system. The system captures video input from the webcam, detects hands, and recognizes specific gestures such as **Thumbs Up**, **Victory (Peace Sign)**, and **Palm**. The program also counts the number of hands visible in the frame and labels the recognized gestures.

**Libraries Used:**

* **OpenCV (cv2)**: OpenCV is a library used for computer vision tasks. It captures video from the webcam and displays it on the screen. It also processes each frame to identify hands and gestures.
* **MediaPipe**: MediaPipe is a framework developed by Google that provides pre-built machine learning models for various tasks, including hand tracking. It detects the positions of key hand landmarks and helps identify hand gestures.

**Explanation of Code**

Below is a breakdown of the code used to implement the hand gesture recognition system:

**Import Libraries**



* **cv2 (OpenCV)**: Used for capturing video and manipulating frames.
* **mediapipe**: Provides the functionality for hand detection and landmark recognition.

**Initialize MediaPipe Hands**

**A screenshot of a computer

Description automatically generated**

* **mp\_hands.Hands()**: Initializes the hand tracking model.
* **mp\_draw.draw\_landmarks()**: Used to draw the landmarks and hand connections on the frame.

A close up of a text

Description automatically generated

* **cv2.VideoCapture(0)**: Captures video from the default webcam. The 0 argument refers to the first camera available.

**Gesture Detection Function**

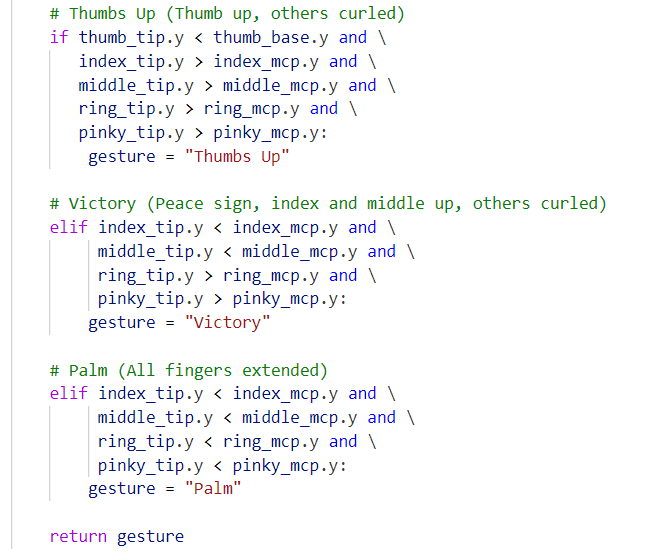
A screenshot of a computer code

Description automatically generated

**Landmark Extraction**: The landmarks parameter represents the detected hand landmarks. We extract the relevant landmark positions (e.g., **thumb\_tip**, **index\_tip**, **middle\_tip**, etc.) for gesture recognition.

The landmarks parameter contains a collection of 3D coordinates representing key points on a detected hand, such as fingertips, knuckles, and the palm base, provided by the MediaPipe Hand Tracking module. These landmarks are essential for analyzing the hand's posture and movements. Each point is identified by predefined constants like THUMB\_TIP or INDEX\_FINGER\_TIP and includes x, y, and z values, where x and y denote the point's position relative to the frame dimensions, and z indicates its depth relative to the camera. By extracting specific landmark positions (e.g., thumb\_tip or index\_tip), we can determine the spatial relationship between different parts of the hand, enabling gesture detection such as recognizing if fingers are curled or extended.

**Gesture Recognition Conditions**



* **Thumbs Up**: If the thumb is pointing upwards and all other fingers are curled downwards, the gesture is identified as "Thumbs Up".
* **Victory**: If the index and middle fingers are extended, and the rest are curled, the gesture is identified as "Victory" (Peace Sign).
* **Palm**: If all fingers are extended (open palm), the gesture is identified as "Palm".

A close-up of a white background

Description automatically generated

* **cap.read()**: Reads each frame from the webcam. If the frame is not captured correctly, it breaks the loop.

A close up of a text

Description automatically generated

* **cv2.flip(frame, 1)**: Flips the captured frame horizontally, making the video feel like a mirror image (which is natural for hand gestures).

A close up of a text

Description automatically generated

* **cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)**: Converts the captured frame from BGR (used by OpenCV) to RGB (required by MediaPipe).

A close up of words

Description automatically generated

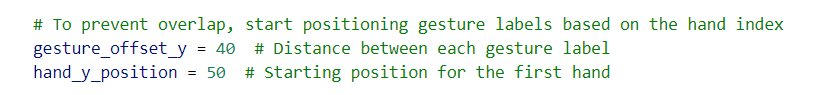
* **hands.process(rgb\_frame)**: Processes the frame using the MediaPipe hand tracking model to detect hand landmarks.

**Hand Detection and Gesture Recognition**

A computer screen shot of a computer code

Description automatically generated

* **results.multi\_hand\_landmarks**: This contains the detected hand landmarks (if any). We count the number of hands detected.



* **gesture\_offset\_y**: Defines the vertical spacing between the labels of each detected hand gesture.
* **hand\_y\_position**: Defines the starting position for the first hand's gesture label.

A computer screen with text

Description automatically generated

* **Drawing Landmarks**: For each detected hand, the landmarks and their connections are drawn on the frame.
* **Gesture Detection**: For each hand, the gesture is detected using the detect\_gesture() function.
* **Displaying Gesture**: The recognized gesture label is displayed on the screen along with the hand index.



* **Display Hand Count**: The number of detected hands is displayed on the top left of the screen.

A white background with colorful text

Description automatically generated

* **Display Frame**: The processed frame is shown in a window titled "Hand Gesture Recognition".
* **Exit on 'q'**: The program will break the loop and exit if the user presses the 'q' key.

**Clean-Up**

A close up of a white background

Description automatically generated

* **cap.release()**: Releases the webcam when done.
* **cv2.destroyAllWindows()**: Closes the OpenCV window showing the video feed.

**Conclusion**

* This code provides a real-time hand gesture recognition system that tracks hands using MediaPipe and detects simple gestures such as Thumbs Up, Victory, and Palm. It uses OpenCV for video capture and drawing landmarks. The recognized gestures are displayed on the screen, and the number of hands detected is updated in real-time.

**License**  
This document and the accompanying code are licensed under the **MIT License**. You are free to use, modify, and distribute the code, provided proper attribution is given.

**Acknowledgments**

* This project uses **MediaPipe**, which is licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0).
* The implementation leverages **OpenCV**, which is licensed under the BSD License.

**Contact and Links**

* **GitHub**: Explore the project repository at [Hand Gesture Recognition on GitHub](https://github.com/ramanareddy1097/Hand-Gesture-Recognition/tree/main).
* **X (formerly Twitter)**: Follow updates and share your thoughts at [@ramanareddy1097](https://x.com/ramanareddy1097).
* **LinkedIn**: Connect with me on LinkedIn at  [Ramana Reddy](https://www.linkedin.com/in/ramana10/).
* **Quora**: Visit my Quora profile for insights and discussions at [Venkat Valryon](https://venkatvalryon.quora.com/?invite_code=n5pIjajs67SSrdmxWDgu).

Bottom of Form