import cv2 - **For webcam video capture and display**

import mediapipe as mp - **For detecting hand and finger landmarks**

import serial -  **For communicating with ESP32 via USB**

import time -  **For adding delays between gesture detection**

try:

**Try to connect to ESP32 on COM8 at 115200 baud rate**

ser = serial.Serial("COM8", 115200, timeout=1) -  **Open COM8 with 115200 speed**

print("[INFO] Serial connected to COM8")

except serial.SerialException as e:

print(f"[ERROR] Could not open COM8: {e}") - **If connection fails, print error and disable serial**

ser = None

mp\_hands = mp.solutions.hands - **Access hand detection solution**

hands = mp\_hands.Hands(static\_image\_mode=False, max\_num\_hands=1) - **Detect 1 hand in live mode**

**Function to detect thumb gesture**

def detect\_gesture(hand\_landmarks):

- **Get positions of thumb tip, thumb IP joint (middle joint), and wrist**

thumb\_tip = hand\_landmarks.landmark[mp\_hands.HandLandmark.THUMB\_TIP]

thumb\_ip = hand\_landmarks.landmark[mp\_hands.HandLandmark.THUMB\_IP]

wrist = hand\_landmarks.landmark[mp\_hands.HandLandmark.WRIST]

**Thumb is above wrist (y is smaller) and thumb is straight (x difference is very small)**

if thumb\_tip.y < wrist.y and abs(thumb\_tip.x - thumb\_ip.x) < 0.05: return "A" - **Gesture "A" means Thumbs Up**

elif thumb\_tip.y > wrist.y and abs(thumb\_tip.x - thumb\_ip.x) < 0.05: return "B" - **Gesture "B" means Thumbs Down**

else:

return None -  **If neither up nor down, return nothing**

cap = cv2.VideoCapture(0) - **Start the webcam (camera index 0)**

print("[INFO] Starting continuous thumb gesture detection and serial sending...")

try:

while cap.isOpened(): - **Continue until webcam is working**

ret, frame = cap.read() - **Capture frame from webcam**

if not ret:

print("[ERROR] Failed to read from webcam")

break - **Exit loop if frame not read**

frame = cv2.flip(frame, 1) - **Mirror the image for natural view**

rgb\_frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB) - **Convert BGR to RGB**

results = hands.process(rgb\_frame) -  **Detect hand landmarks**

gesture = None - **Start with no gesture**

if results.multi\_hand\_landmarks: - **If at least one hand detected**

for hand\_landmarks in results.multi\_hand\_landmarks:

gesture = detect\_gesture(hand\_landmarks) - **Detect gesture from landmarks**

if gesture:

break **- If gesture is found, stop checking more hands**

if gesture:

gesture\_name = "Thumbs Up" if gesture == "A" else "Thumbs Down"

print(f"[DETECTED] {gesture\_name} ({gesture})") - **Print gesture is detected**

if ser:

try:

- **Send "A" or "B" to ESP32**

ser.write(gesture.encode())

print(f"[SENT] Sent '{gesture}' successfully")

except Exception as e:

print(f"[ERROR] Failed to send '{gesture}': {e}")

else:

print("[WARN] Serial not connected; cannot send data")

time.sleep(0.5) - **Delay to avoid sending too fast**

else:

time.sleep(0.1) - **Short delay if no gesture is found**

cv2.imshow("Thumb Gesture Detection", frame) - **Show video frame**

if cv2.waitKey(1) & 0xFF == ord("q"): - **Press 'q' to quit**

break

except KeyboardInterrupt:

print("[INFO] Exiting on user interrupt")

finally:

cap.release() - **Turn off webcam**

if ser:

ser.close() - **Close serial connection**

print("[INFO] Program terminated")