**POSE ESTIMATION**

**import cv2**

**import mediapipe as mp**

**# Initialize MediaPipe Pose and drawing utilities**

**mp\_pose = mp.solutions.pose**

**mp\_drawing = mp.solutions.drawing\_utils**

**# Capture video from webcam**

**cap = cv2.VideoCapture(0)**

**# Initialize Pose model**

**with mp\_pose.Pose(min\_detection\_confidence=0.5, min\_tracking\_confidence=0.5) as pose:**

**while cap.isOpened():**

**ret, frame = cap.read()**

**if not ret:**

**break**

**# Convert the frame to RGB**

**image = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)**

**image.flags.writeable = False**

**# Process the image and detect poses**

**results = pose.process(image)**

**# Convert the image back to BGR for rendering**

**image.flags.writeable = True**

**image = cv2.cvtColor(image, cv2.COLOR\_RGB2BGR)**

**# Draw the pose landmarks**

**if results.pose\_landmarks:**

**mp\_drawing.draw\_landmarks(**

**image,**

**results.pose\_landmarks,**

**mp\_pose.POSE\_CONNECTIONS,**

**mp\_drawing.DrawingSpec(color=(0, 255, 0), thickness=2, circle\_radius=2),**

**mp\_drawing.DrawingSpec(color=(0, 0, 255), thickness=2, circle\_radius=2)**

**)**

**# Show the frame**

**cv2.imshow('Pose Estimation', image)**

**# Break on pressing 'q'**

**if cv2.waitKey(10) & 0xFF == ord('q'):**

**break**

**# Release the webcam and close windows**

**cap.release()**

**cv2.destroyAllWindows()**