

```
# 18BCS6201-CV Practical-5 (B) (Gauri Prabhakar) (AI-ML-2)(A)
# Aim: To implement handtracking using mediapipe in python and OpenCV.

# Importing necessary modules.
import cv2
import mediapipe as mp

# Creating a variable to store the video using the '.VideoCapture()' function.
cap=cv2.VideoCapture(r"C:\Users\gauri\Desktop\OpenCV Media\hands.mp4")

# Importing 'mp.solutions.hands' to a variable 'mpsh'.
mpsh=mp.solutions.hands
# Declaring the object 'hands' from 'mp.solutions'.
hands=mpsh.Hands()
# Importing 'mp.solutions.hands' to a variable 'mpsdu'.
# We will use 'drawing_utils' to draw the key points.
mpsdu=mp.solutions.drawing_utils

while True:
    # Capturing the video frame by frame using the '.read()' method.
    success,frame=cap.read()
    # Reading the frames and converting them to RGB.
    frame_RGB=cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

    # Detecting hands in the frame using the function 'hands.process()'.
    results=hands.process(frame_RGB)
    # Printing the results.
    #print(results.multi_hand_landmarks)

    # If hands are detected that is 'results.multi_hand_landmarks' returns true:
    if results.multi_hand_landmarks:
        # For 'handLms' variable in 'results.multi_hand_landmarks':
        for handLms in results.multi_hand_landmarks:
            # Looping through the 'handLms' variable to find out the coordinates of each key point.
            for id, lm in enumerate(handLms.landmark):
                # Printing the landmarks and 'ids'.
                print(id, lm)

                # Declaring the variables 'h', 'w' and 'c'.
                h,w,c=frame.shape

                # Declaring the variables of the 'x' and 'y' coordinates.
                cx,cy=int(lm.x*w), int(lm.y*h)

                # Numbering the key points.
                cv2.putText(frame, str(int(id)), (cx + 10, cy + 10), cv2.FONT_HERSHEY_COMPLEX_SMALL,
                            1, (254, 31, 79), 2)

                # Printing the 'id' and the coordinates of the key points.
                print(id, cx, cy)

                # Highlighting the key points.
                cv2.circle(frame, (cx,cy), 10, (31,79,254), cv2.FILLED)

    # Connecting the key points using the function 'mpsdu.draw_landmarks()'.
    mpsdu.draw_landmarks(frame, handLms, mpsd.HAND_CONNECTIONS)

    # Rendering the video with effective Handtracking to the console by using the function '.imshow()'.
    cv2.imshow("Hand Tracking", frame)

    # Setting up '.waitkey()' to wait for a specific time until any key is pressed and break the loop.
    cv2.waitKey(1)
```