



PROJECT 1

(COMPUTER VISION)

Submitted To: Dr. Muhammad Jawad Khan

Submitted By: Usman Zaheer, 327700



Problem Statement

This is a mini project. You will have to use your previous concepts to complete the project. Once the video processing part will be discussed you will have to submit real-time results of your codes. The objective of this project is to make you used to with video processing and object detection. You will have to write a code to skeletonize the persons present within a video.

Choice Opted:

- Recognition of hand gesture using skeletonization.
- In case of multiple people, all hand gestures needs to be identified.

Solution to the problem

The below mentioned code contains all the necessary steps to detect the persons in the video. User can use it on webcam or a video as per the requirement. I have written all the necessary functions in the python file named “gesturedetector.py”. This file is imported in the main file (jupyter notebook given) where the recognition of hand gestures for single and multiple persons takes place.

- It can be used on live video webcam and recorded video as well.

PYTHON CODE

The code given below contains all the necessary functions which are called in the handgestureskeletonization jupyter notebook.

“gesturedetector.py”

```
import cv2
import mediapipe as mp
import time

class handDetector():
    def __init__(self, mode=False, maxHands=10, detectionCon=0.5, trackCon=0.5):
        self.mode = mode
        self.maxHands = maxHands
        self.detectionCon = detectionCon
        self.trackCon = trackCon

        self.mpHands = mp.solutions.hands
        self.hands = self.mpHands.Hands(self.mode, self.maxHands,
                                         self.detectionCon, self.trackCon)
        self.mpDraw = mp.solutions.drawing_utils

    def findHands(self, img, draw=True):
        imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        self.results = self.hands.process(imgRGB)
```

```

# print(results.multi_hand_landmarks)
if self.results.multi_hand_landmarks:
    for handLms in self.results.multi_hand_landmarks:
        if draw:
            self.mpDraw.draw_landmarks(img, handLms,
                                        self.mpHands.HAND_CONNECTIONS)

    return img

def findPosition(self, img, handNo=0, draw=True):

    lmList = []
    if self.results.multi_hand_landmarks:
        myHand = self.results.multi_hand_landmarks[handNo]
        for id, lm in enumerate(myHand.landmark):
            # print(id, lm)
            h, w, c = img.shape
            cx, cy = int(lm.x * w), int(lm.y * h)
            # print(id, cx, cy)
            lmList.append([id, cx, cy])
            if draw:
                cv2.circle(img, (cx, cy), 15, (255, 0, 255), cv2.FILLED)

    return lmList

def main():
    pTime = 0
    cTime = 0
    cap = cv2.VideoCapture(1)
    detector = handDetector()
    while True:
        success, img = cap.read()
        img = detector.findHands(img)
        lmList = detector.findPosition(img)
        if len(lmList) != 0:
            print(lmList[4])

        cTime = time.time()
        fps = 1 / (cTime - pTime)
        pTime = cTime

        cv2.putText(img, str(int(fps)), (10, 70), cv2.FONT_HERSHEY_PLAIN, 3,
                    (255, 0, 255), 3)
        cv2.imshow("Image", img)
        cv2.waitKey(1)

if __name__ == "__main__":
    main()

```

“Hand Gesture Skeletonization.ipynb”

```
import cv2 as cv
import time
import gesturedetector as gd

wCam, hCam = 640, 480
cap = cv.VideoCapture(0)
cap.set(3, wCam)
cap.set(4, hCam)

pTime = 0

detector = gd.handDetector(detectionCon=0.75)

tipIds = [4, 8, 12, 16, 20]

while True:
    success, img = cap.read()
    img = detector.findHands(img)
    lmList = detector.findPosition(img, draw=False)

    if len(lmList) != 0:
        fingers = []

        if lmList[tipIds[0]][1] > lmList[tipIds[0] - 1][1]:
            fingers.append(1)
        else:
            fingers.append(0)

        # 4 Fingers
        for id in range(1, 5):
            if lmList[tipIds[id]][2] < lmList[tipIds[id] - 2][2]:
                fingers.append(1)
            else:
                fingers.append(0)

        totalFingers = fingers.count(1)

        cTime = time.time()
        fps = 1 / (cTime - pTime)
        pTime = cTime

    cv.imshow("Hand Gesture Detector", img)
    if cv.waitKey(1) == 27:
        break
```

Results on video for single person:

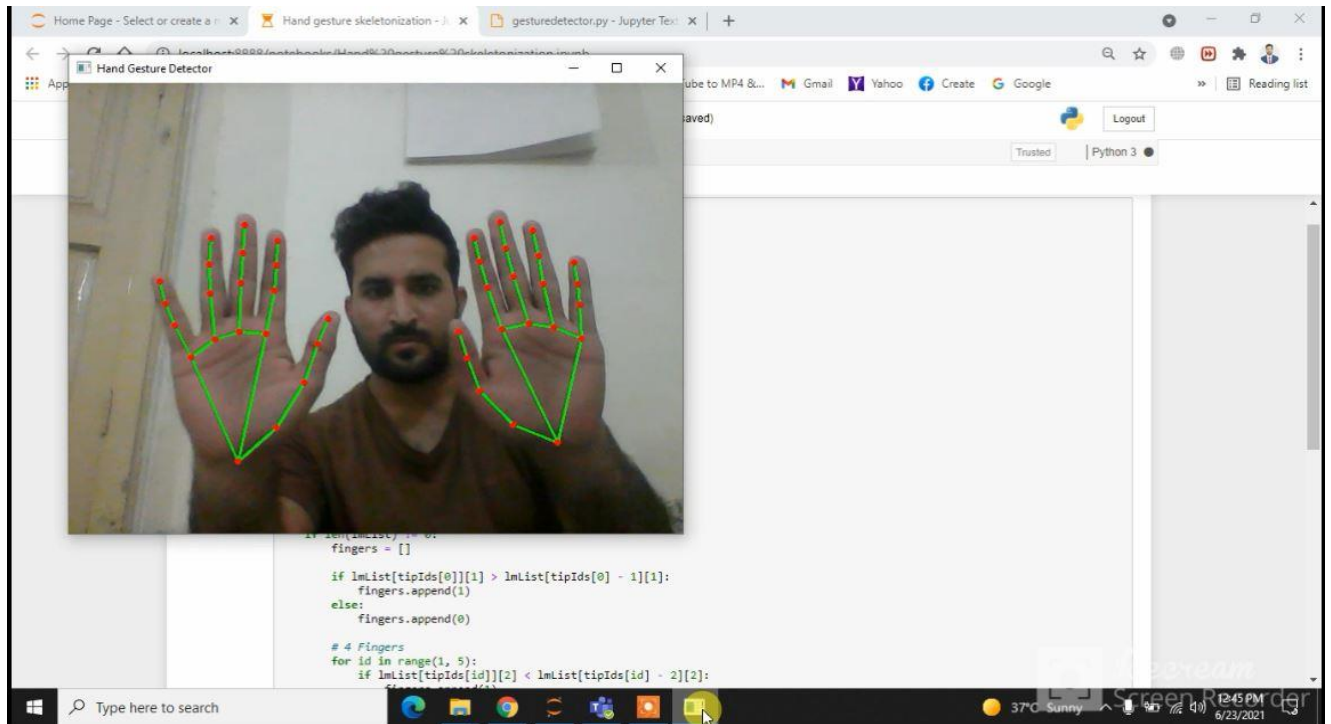


Figure 1: Result on live video for single person

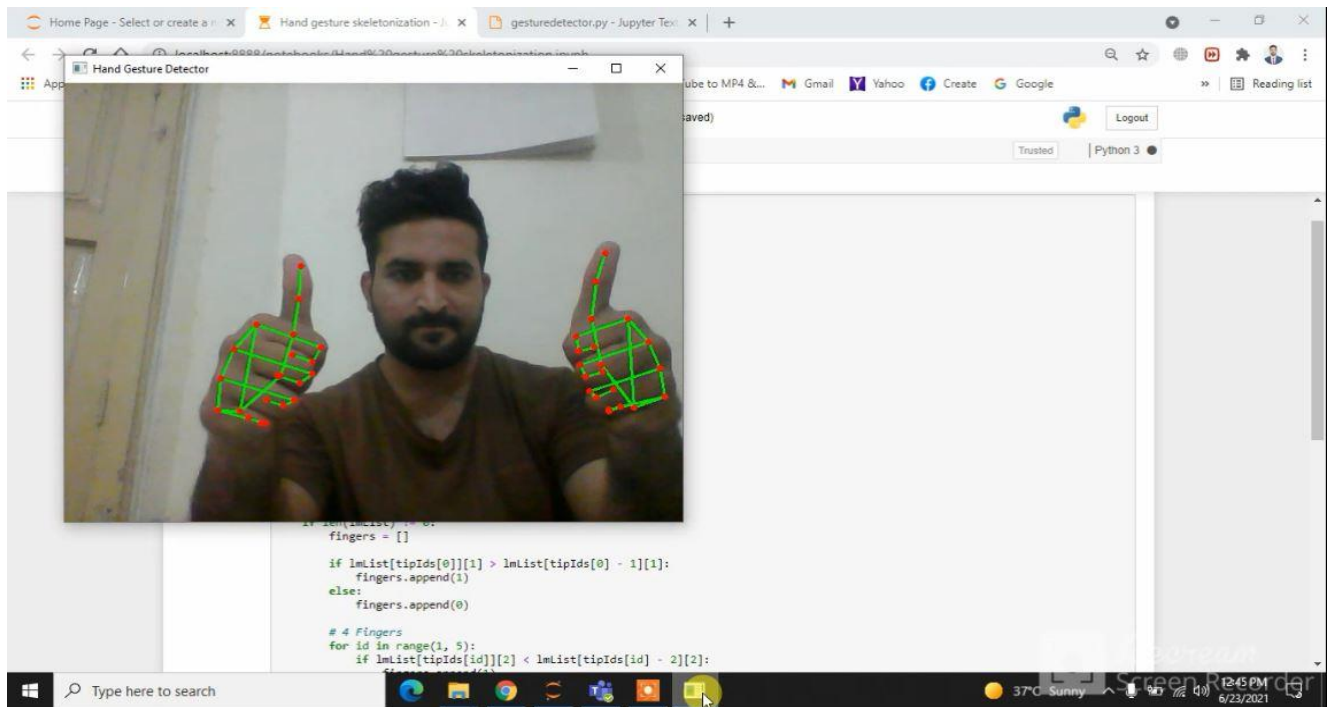


Figure 2: Result on live video for single person

Results on live video for multiple persons:

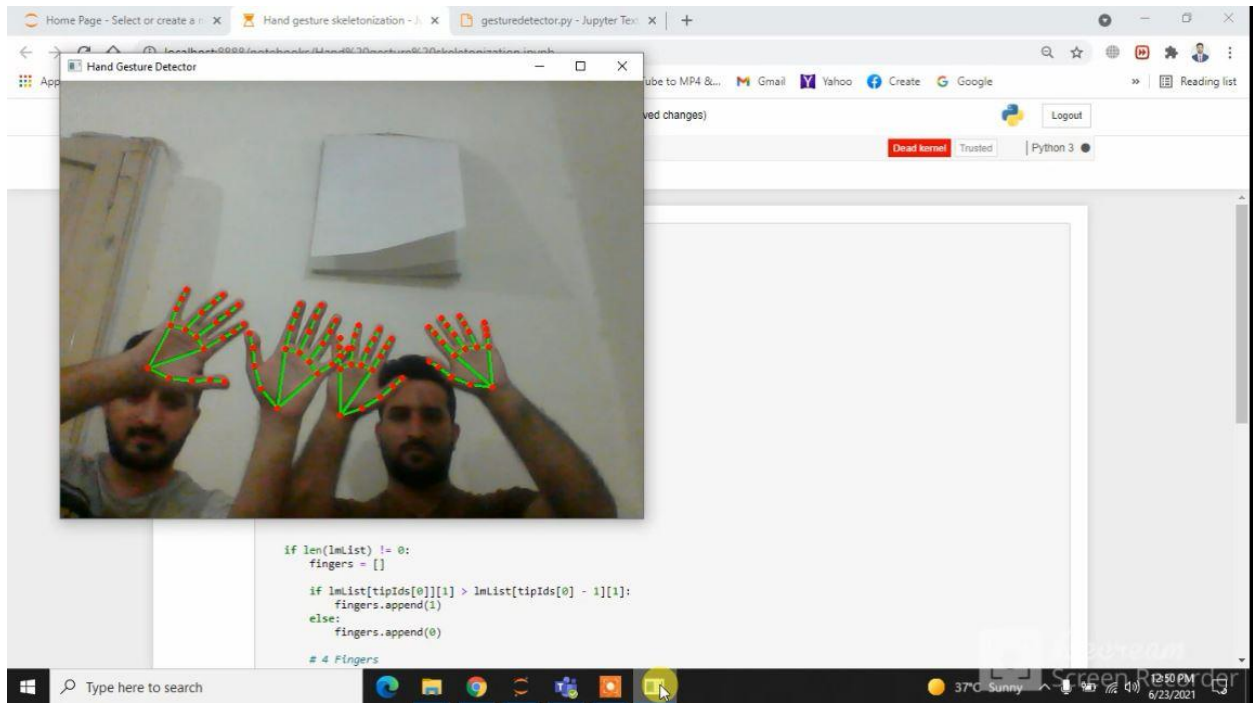


Figure 3: Result on live video for multiple person's

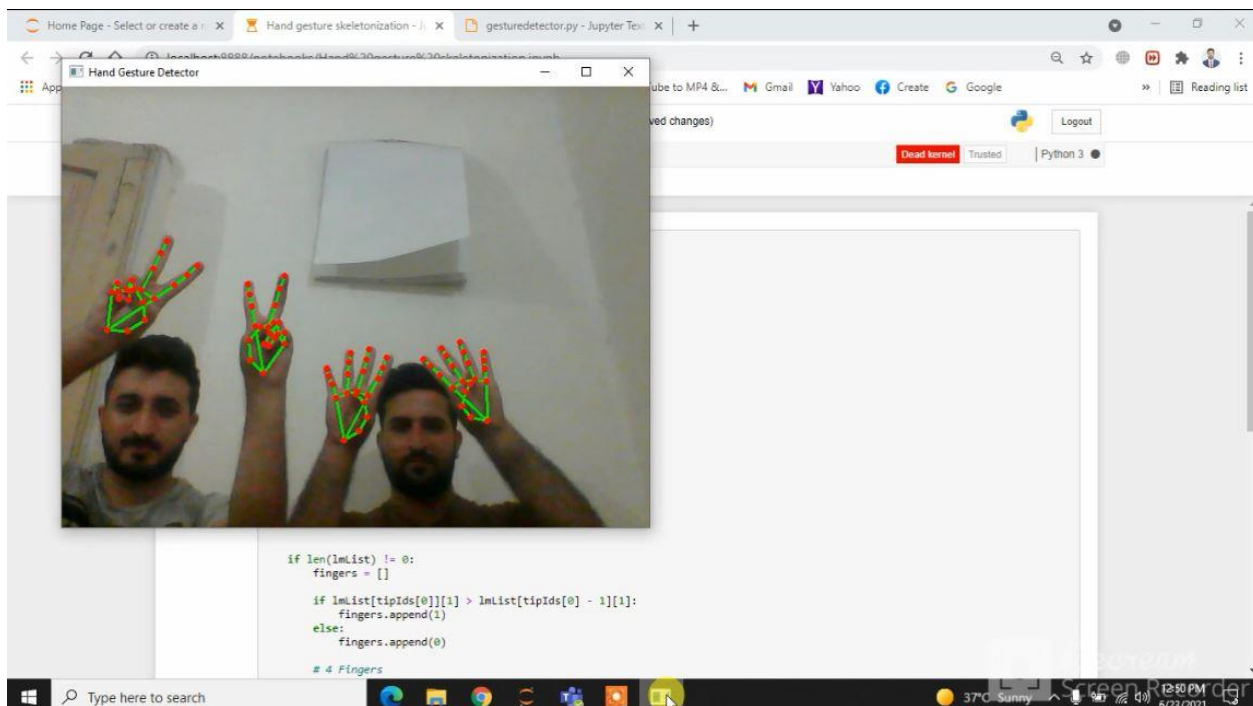


Figure 4: Result on live video for multiple person's



Instructions on running the code:

Just load the “gesturedetector.py” module in the home folder directory and run the jupyter notebook provided (Handdestureskeletonization.ipynb).



References:

1. YouTube. (2021). Retrieved 20 June 2021, from <https://www.youtube.com/watch?v=ZmkfWNLzQ-c>
2. Chen, Z., Kim, J., Liang, J., Zhang, J., & Yuan, Y. (2014). Real-Time Hand Gesture Recognition Using Finger Segmentation. *The Scientific World Journal*, 2014, 1-9. doi: 10.1155/2014/267872