1. Hand Detection

import cv2

import mediapipe as mp

import time

import math

cap = cv2.VideoCapture(0)

mpHands = mp.solutions.hands

hands = mpHands.Hands(min\_detection\_confidence=0.7, min\_tracking\_confidence=0.7)

mpDraw = mp.solutions.drawing\_utils

pTime = 0

cTime = 0

while True:

success, img = cap.read()

imgRGB = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

results = hands.process(imgRGB)

if results.multi\_hand\_landmarks:

for handLms in results.multi\_hand\_landmarks:

for id, lm in enumerate(handLms.landmark):

h, w, c = img.shape

cx, cy = int(lm.x\*w), int(lm.y\*h)

cv2.circle(img, (cx, cy), 10, (255, 0, 0), cv2.FILLED)

mpDraw.draw\_landmarks(img, handLms, mpHands.HAND\_CONNECTIONS)

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)

2. Pose Detection

import cv2  
import mediapipe as mp  
import time  
  
mpPose = mp.solutions.pose  
pose = mpPose.Pose()  
mpDraw = mp.solutions.drawing\_utils  
cap = cv2.VideoCapture('PoseVideos/1.mp4')  
pTime = 0  
  
while True:  
 success, img = cap.read()  
 img = cv2.resize(img, (720, 405))  
 imgRGB = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)  
 results = pose.process(imgRGB)  
 print(results.pose\_landmarks)  
 if results.pose\_landmarks:  
 for id, poseLms in enumerate(results.pose\_landmarks.landmark):  
 h, w, c = img.shape  
 cx, cy = int(poseLms.x\*w), int(poseLms.y\*h)  
 print(id, cx, cy)  
 cv2.circle(img, (cx, cy), 10, (255, 0, 0), cv2.FILLED)  
 mpDraw.draw\_landmarks(img, results.pose\_landmarks, mpPose.POSE\_CONNECTIONS)  
  
  
 cTime = time.time()  
 fps = 1 / (cTime - pTime)  
 pTime = cTime  
  
 cv2.putText(img, str(int(fps)), (10, 70), cv2.FONT\_HERSHEY\_SIMPLEX, 3, (255, 0, 255), 3)  
 cv2.imshow("Image", img)  
 cv2.waitKey(1)