from signal import Signals

from sre\_constants import CATEGORY\_UNI\_DIGIT

import cv2 as cv

import mediapipe as mp

import numpy as np

hands = mp.solutions.hands.Hands(False,2,1,0.5,0.5)

#create a mp draw to draw

mpDraw = mp.solutions.drawing\_utils

#to start the webcam

capture = cv.VideoCapture(0)

capture.set(3,1440)

while True:

isTrue, frame = capture.read()

frame = cv.flip(frame,1)

#height = 1280,width = 720

width = int(frame.shape[0])

height = int(frame.shape[1])

frameRGB = cv.cvtColor(frame, cv.COLOR\_BGR2RGB)

#show results

results = hands.process(frameRGB)

# if any hand is detected

if results.multi\_hand\_landmarks != None :

count = {"RIGHT":0, "LEFT":0}

for hand\_index,hand\_info in enumerate(results.multi\_handedness):

hand\_label= hand\_info.classification[0].label

handlandmarks = results.multi\_hand\_landmarks[hand\_index]

#multi handedness contains only label

myHandx = []

myHandy = []

mpDraw.draw\_landmarks(frame,handlandmarks,mp.solutions.hands.HAND\_CONNECTIONS)

for Landmark in handlandmarks.landmark:

myHandx.append(Landmark.x)

myHandy.append(Landmark.y)

if myHandy[8]<myHandy[6] and myHandy[12]>myHandy[10] and myHandy[16]>myHandy[14] and myHandy[20]<myHandy[18]:

cv.putText(frame,f"{hand\_label.upper()}:SPIDERMAN SIGN",(50,100),cv.FONT\_HERSHEY\_SIMPLEX,1,(0,0,255),2)

if myHandy[8] < myHandy[6] and myHandy[12] < myHandy[10] and myHandy[16] > myHandy[14] and myHandy[20] > myHandy[18]:

ret,screen = capture.read()

if ret:

while(True):

cv.imshow("SCREENSHOT",screen)

if cv.waitKey(1) & 0xFF == ord('q'):

break

cv.imshow("Video",frame)

if cv.waitKey(20) & 0xFF ==ord('d'):

break

capture.release()

cv.destroyAllWindows()