import cv2

import time

import math

import numpy as np

import mediapipe as mp

mpDraw = mp.solutions.drawing\_utils #呼叫繪圖工具

mpHands = mp.solutions.hands #呼叫手部工具

#呼叫手部工具內的手部辨識器

hands = mpHands.Hands(

static\_image\_mode=False, #單張或串流(True單張模式，False串流模式)

model\_complexity=0,#0->精簡模型(快)，1->完整模型(慢)，rpi記得註解

max\_num\_hands=1, #辨識最多手

min\_detection\_confidence=0.7, #辨識信任度

min\_tracking\_confidence=0.5 #追蹤信任度

)

AngleTH=130 #判斷張開角度

def findAngleF(a,b,c):

ang = math.degrees(math.atan2(c[2]-b[2], c[1]-b[1]) - math.atan2(a[2]-b[2], a[1]-b[1]))

if ang<0 :

ang=ang+360

if ang >= 360- ang:

ang=360-ang

return round(ang,2)

#打碼函數

def do\_mosaic(frame,x,y,w,h,neighbor=10):

# 來源：https://www.796t.com/article.php?id=10626

fh,fw = frame.shape[0],frame.shape[1]

x,y,w,h=round(x\*fw),round(y\*fh),round(w\*fw),round(h\*fh)

if (y + h > fh) or (x + w > fw):

return

for i in range(0,h - neighbor,neighbor): # 關鍵點0 減去neightbour 防止溢位

for j in range(0,w - neighbor,neighbor):

rect = [j + x,i + y,neighbor,neighbor]

color = frame[i + y][j + x].tolist() # 關鍵點1 tolist

left\_up = (rect[0],rect[1])

right\_down = (rect[0] + neighbor - 1,rect[1] + neighbor - 1) # 關鍵點2 減去一個畫素

cv2.rectangle(frame,left\_up,right\_down,color,-1)

from picamera2 import Picamera2

picam2 = Picamera2()

preview\_config = picam2.create\_preview\_configuration(main={"size": (800, 600)})

picam2.configure(preview\_config)

picam2.start()

while True:

stime=time.time()

frame = picam2.capture\_array()

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

h, w, c = frame.shape #取得螢幕長寬色彩

frame=cv2.flip(frame,1) #翻轉：-1上下、0上下左右、1左右

results = hands.process(frame) #手部辨識

if results.multi\_hand\_landmarks: #如果有找到手部

for i in range(len(results.multi\_handedness)): #所有的手

thisHandType=results.multi\_handedness[i].classification[0].label #手的屬性

thisHand=results.multi\_hand\_landmarks[i] #取得這隻手

mpDraw.draw\_landmarks(frame, thisHand, mpHands.HAND\_CONNECTIONS) #利用工具畫

#學習自己畫關節(了解關節座標位置)

thisHandLMList = []

for id, lm in enumerate(thisHand.landmark): #id=編號,lm=座標

thisHandLMList.append([id, lm.x, lm.y,lm.z])

hx, hy = int(lm.x \* w), int(lm.y \* h) #計算座標

cv2.circle(frame, (hx, hy), 5, (255, 0, 0), cv2.FILLED) #在關節點上標藍色圓形

cv2.putText(frame,str(id),(hx,hy), cv2.FONT\_HERSHEY\_PLAIN, 1, (255, 0, 255), 1)

if id==0:

cv2.putText(frame,thisHandType,(hx,hy-30), cv2.FONT\_HERSHEY\_PLAIN, 2, (0, 255, 0), 2)

finger=[0,0,0,0,0]

if (findAngleF(thisHandLMList[0],thisHandLMList[3],thisHandLMList[4])>AngleTH):

finger[0]=1

if (findAngleF(thisHandLMList[0],thisHandLMList[6],thisHandLMList[8])>AngleTH):

finger[1]=1

if (findAngleF(thisHandLMList[0],thisHandLMList[10],thisHandLMList[12])>AngleTH):

finger[2]=1

if (findAngleF(thisHandLMList[0],thisHandLMList[14],thisHandLMList[16])>AngleTH):

finger[3]=1

if (findAngleF(thisHandLMList[0],thisHandLMList[18],thisHandLMList[20])>AngleTH):

finger[4]=1

print(finger)

#-----------------判斷手勢------------------------

text=""# 姆,食,中,無,小

if (finger==[1,0,0,0,0]):

text="Good"

if (finger==[0,0,1,1,1]):

text="OK"

if (finger==[1,0,0,0,1]):

text="666"

if (finger==[0,1,1,0,0]):

text="Ya"

if (finger==[0,0,0,0,0]):

text="Zero"

if (finger==[1,1,1,1,1]):

text="Hi"

if (finger==[0,0,1,0,0]):

text="E04"

# 用opencv劃一個矩形，範圍是整隻手

lmArray=np.array(thisHandLMList) #List轉陣列

min\_x,max\_x=np.amin(lmArray[:,1]),np.amax(lmArray[:,1]) #取第一欄(x)最小值及最大值

min\_y,max\_y=np.amin(lmArray[:,2]),np.amax(lmArray[:,2]) #取第二欄(y)最小值及最大值

do\_mosaic(frame,min\_x,min\_y,(max\_x-min\_x),(max\_y-min\_y))

# 影像 文字 位置 字形 大小 顏色 粗細

cv2.putText(frame, text, (0, 200), cv2.FONT\_HERSHEY\_PLAIN, 5 , (255, 0, 0), 5) #在畫面中寫字（英文）

etime=time.time()

fps=round(1/(etime-stime),2)

cv2.putText(frame,"FPS:" + str(fps),(10,50), cv2.FONT\_HERSHEY\_PLAIN, 3, (0, 0, 255), 3)

cv2.imshow('Webcam',frame) #顯示畫面內容

key=cv2.waitKey(1) #等候使用者按鍵盤指令

if key==ord('a'): #a拍照

cv2.imwrite('webcam.jpg',frame) #拍照

if key==ord('q'): #q退出

break