

題目:手部追蹤模型

學號:409411401

日期:2022/6/22

## 製作動機

在看到影像處理專題時，我想到老師有提到人臉辨識，因此我想到除了能辨識臉之外，也可以用來追蹤手部的姿勢和動作，進而應用在生活中，像是拍照或是各種指令

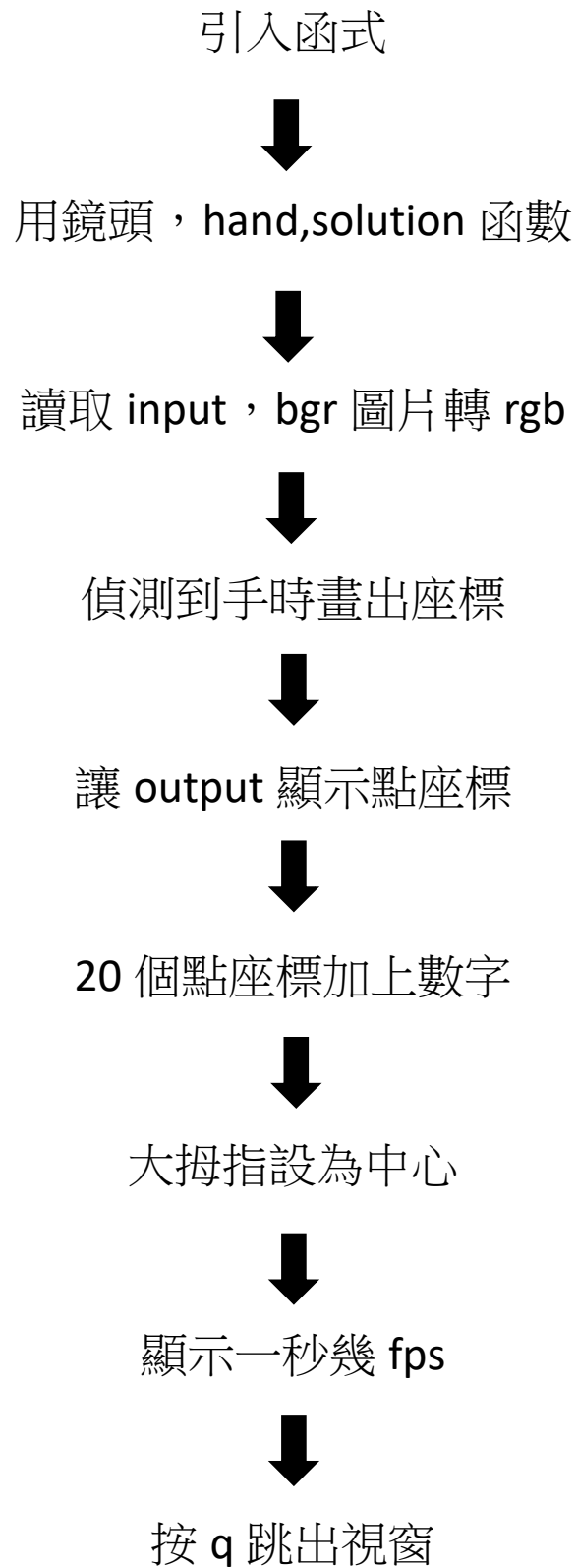
## 功能介紹

1. 會追蹤手部，並由 20 個點組成，點跟點之間會連線，並標上這是第幾個點
2. 在 `output` 中會顯示每一個點的座標
3. 在螢幕中會顯示更新頻率
4. 大拇指會著重追蹤放大，方便辨識

## 工作原理

1. 使用 `cv2` 的鏡頭功能，捕捉手部
2. 用 `mediapipe` 的手部模型，運用 `hand` 函數
3. 用 `mediapipe, cv2` 畫點
4. 用 `enumerate` 標記座標
5. `putText` 顯示每一個點的數字
6. 手部偵測模型需要 `rgb` 的圖片

## 程式流程



# 程式碼

```
#引入函式
import cv2
import mediapipe as mp
import time

# 用鏡頭, hand, solution 函數
cap = cv2.VideoCapture(0)
mpHands = mp.solutions.hands
hands = mpHands.Hands(min_detection_confidence=0.5, min_tracking_confidence=0.5)
mpDraw = mp.solutions.drawing_utils
handLmsStyle = mpDraw.DrawingSpec(color=(0, 0, 255), thickness=3)
handConStyle = mpDraw.DrawingSpec(color=(0, 255, 0), thickness=5)
pTime = 0
cTime = 0

while True:
    # 讀取input
    ret, img = cap.read()
    if ret:
        # bgr 圖片轉rgb
        imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        result = hands.process(imgRGB)

        imgHeight = img.shape[0]
        imgWidth = img.shape[1]

        if result.multi_hand_landmarks:
            # 偵測到手時畫出座標
            for handLms in result.multi_hand_landmarks:
                mpDraw.draw_landmarks(img, handLms, mpHands.HAND_CONNECTIONS, handLmsStyle, handConStyle)

            if result.multi_hand_landmarks:
                # 偵測到手時畫出座標
                for handLms in result.multi_hand_landmarks:
                    mpDraw.draw_landmarks(img, handLms, mpHands.HAND_CONNECTIONS, handLmsStyle, handConStyle)

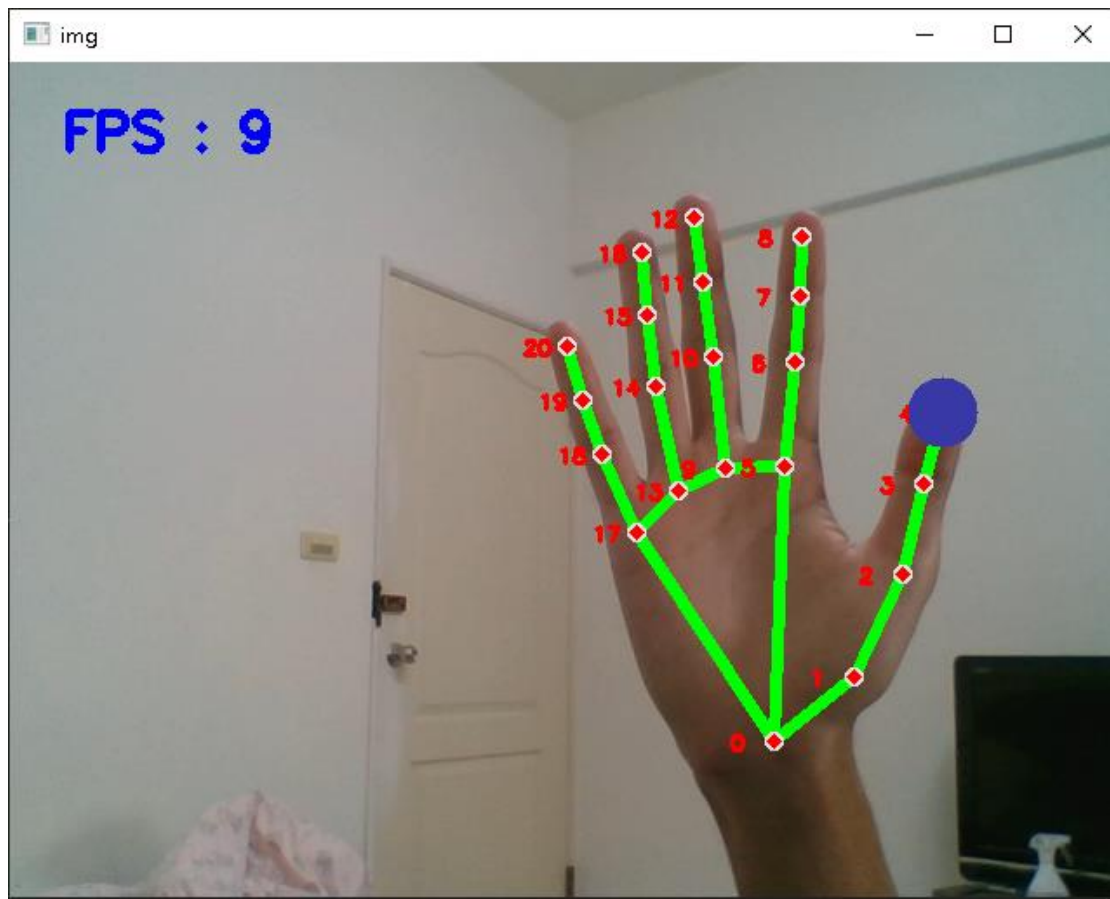
                    # 讓output顯示點座標
                    for i, lm in enumerate(handLms.landmark):
                        xPos = int(lm.x * imgWidth)
                        yPos = int(lm.y * imgHeight)

                        # 20個點座標加上數字
                        cv2.putText(img, str(i), (xPos-25, yPos+5), cv2.FONT_HERSHEY_SIMPLEX, 0.4, (0, 0, 255), 2)

                        # 大拇指設為中心
                        if i == 4:
                            cv2.circle(img, (xPos, yPos), 20, (166, 56, 56), cv2.FILLED)
                            print(i, xPos, yPos)

            # 顯示一秒幾fps
            cTime = time.time()
            fps = 1/(cTime-pTime)
            pTime = cTime
            cv2.putText(img, f"FPS : {int(fps)}", (30, 50), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 3)
            cv2.imshow('img', img)

# 按q跳出視窗
if cv2.waitKey(1) == ord('q'):
    break
```



```
0 460 429
1 506 383
2 532 317
3 545 258
4 556 212
5 462 252
6 467 185
7 471 142
8 473 103
9 424 254
10 412 180
11 407 133
12 401 92
13 394 269
14 377 203
15 369 158
16 362 117
17 368 297
18 344 249
19 330 214
20 319 181
```

## 問題討論與心得

這一次的期末報告，雖然看起來程式碼並不多，但因為都是沒什麼接觸過的，所以也花費不少時間研究，尤其是了解新的 **function** 以及內建參數和模型時，也下了一番功夫，但在這次專題中，我學到了很多，也發現只要會使用 **python** 的模組，許多看似困難的題目，都能迎刃而解，希望未來自己能在影像處理的方面更加精進，能完成更難的專案

## 成果影片上傳

<https://youtu.be/Hg4kTVbZDK8>