

新竹市第 41 屆中小學科學展覽會

作品說明書

科 別：生活與應用科技（一）

組 別：國小組

作品名稱：畫中有話

關鍵詞：圖像辨識、手勢辨識、語音朗讀

編 號：

摘要

我們利用 Anaconda3 的 Jupyter Notebook，使用 Python 程式達到辨識效果。一開始我們想做手語的「手勢辨識」，後來加入了語音功能，這樣就可以達到手語加口語的效果，方便一般人與不方便說話的人溝通。一開始，我們先設計可用在手機上的「手語手勢辨識 APP」，但因圖片資料庫建立遇到阻礙，加上辨識正確率太低，試了很多辦法都沒有改善，只好換另外的辨識系統研究。Python 程式能成功辨識靜態手勢，且辨識系統能夠加入語音朗讀。目前已經能成功辨識一些手勢並發出語音，不過判讀時畫面會稍有延遲。雖然我們還有很多技術層面的問題要克服，但也為手勢辨識系統建立了一個不錯的開端，希望未來能有專業的設計師能做出我們理想中的 APP。

壹、研究動機

有一次在一家咖啡廳內，看到兩個人在用手語溝通，她們比著我完全看不懂的手語，加上十二年國教推動手語成為母語之一，因此藉由此研究，希望能研發出一個能辨識手語手勢的 APP，讓看不懂手語手勢的人能即時翻譯，了解其中的意思。

有了這個想法後，我們開始上網搜尋相關資料，最後在老師的鼓勵下，開啟了這段手語手勢辨識系統的研究之路。

貳、研究目的

- 一、寫出可辨識靜態手勢並語音讀出的手機 APP。
- 二、寫出可辨識靜態手勢並語音讀出的 python 程式。

參、研究設備及器材

一、logitech C922 Pro 網路攝影機



二、電腦

三、滑鼠

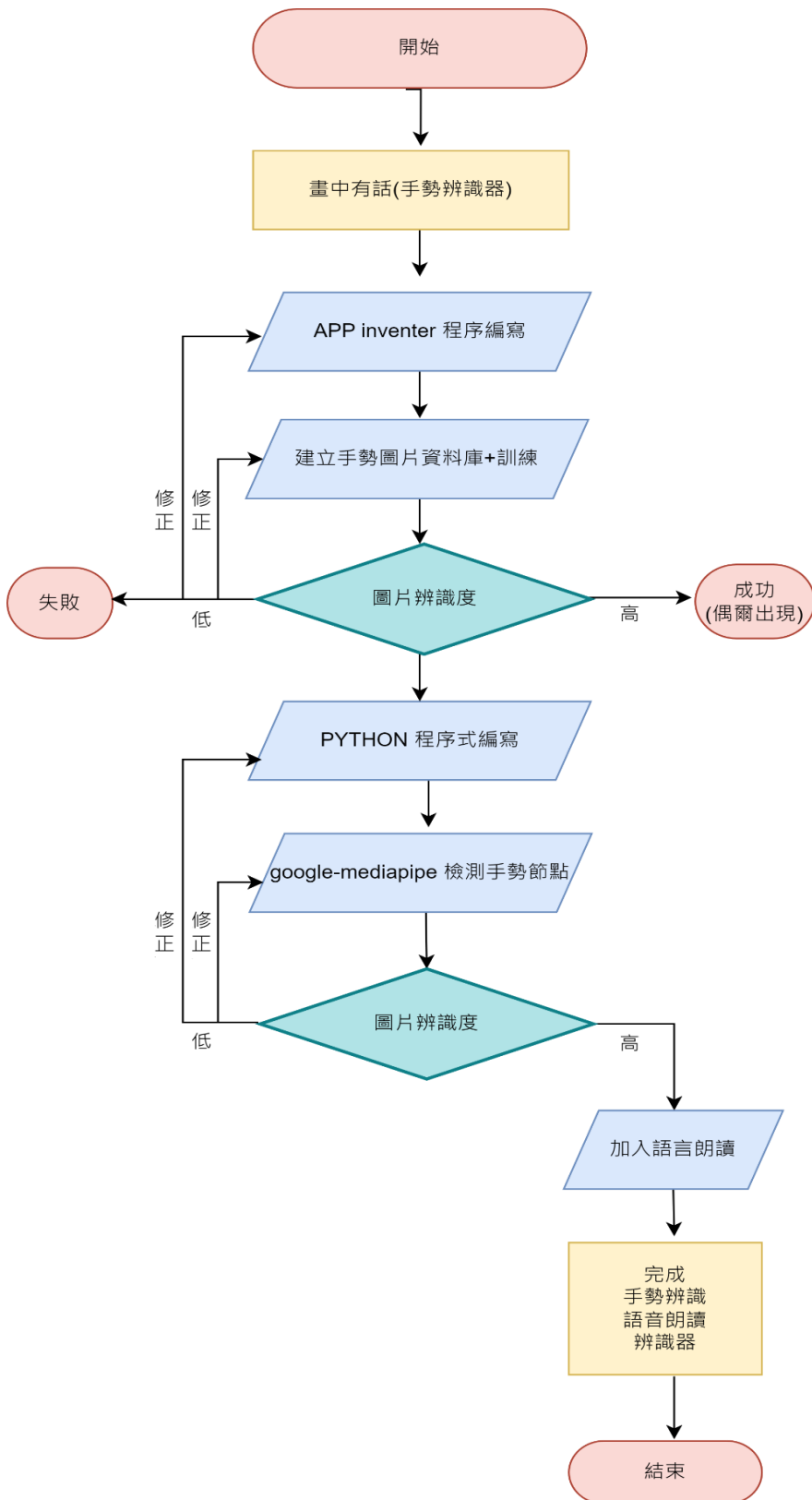
四、鍵盤

五、喇叭

六、螢幕

肆、研究過程或方法

一、研究架構



二、研讀文獻

(一) 研讀 MIT App Inventor 程式編寫教學

訓練完的手型檔案匯入 MIT App Inventor，並寫程式，下載至手機中。下載後會形成一個 app，接著利用手機 APP 來辨識。辨識後，會顯示出相似度，但因 Personal Image Classifier 模型訓練會黑屏的緣故，因此改用其他程式。

(二) 研讀 Personal Image Classifier 訓練模組教學

用相機拍完手勢照片後會使用「Personal Image Classifier」免費網站來訓練。訓練完畢後，會把訓練好的照片下載成一個檔案，放進「MIT App Inventor」裡面，並寫程式。但倘若使用太多照片檔畫面會黑屏，因而改用 Python 程式語言，希望未來能將手勢推廣成大眾都明白的一項母語。

27 8月

APPINVENTOR - 圖形化ANDROID

【教學】App Inventor的人工智慧應用：Personal Image Classifier (PIC) Tools 自訂神經網路視覺辨識

POSTED ON 2019-08-27 BY CAVEDU 阿吉·雜工

Personal Image Classifier

1 Add Training Data 2 Select Model 3 Add Testing Data 4 View Results

Label Correctness

Labelled Correctly

Labelled Incorrectly

RIGHT: 0.39468
LEFT: 0.33059
STOP: 0.27473

Label Correctness
Confidence Graph

STOP

RIGHT

STOP: 0.96078
RIGHT: 0.03912
LEFT: 0.00010

RIGHT: 0.97627
STOP: 0.01631
LEFT: 0.00742

Search...

近期文章

Wio Terminal 小技巧：中斷與 LCD 折線圖

工業電腦 PE1000N-開箱使用教學·支援4G LTE！

YOLO v5, v7 與 v8 物件偵測模型效能比較·執行於 Jetson AGX Orin 與 RTX 4070 Ti

安裝Discord社群平台伺服器-從0到加入CAVEDU學習頻道

NVIDIA Jetson專案 製作樂高輸送帶 執行自動光學檢測 AOI(Automated Optical Inspection)

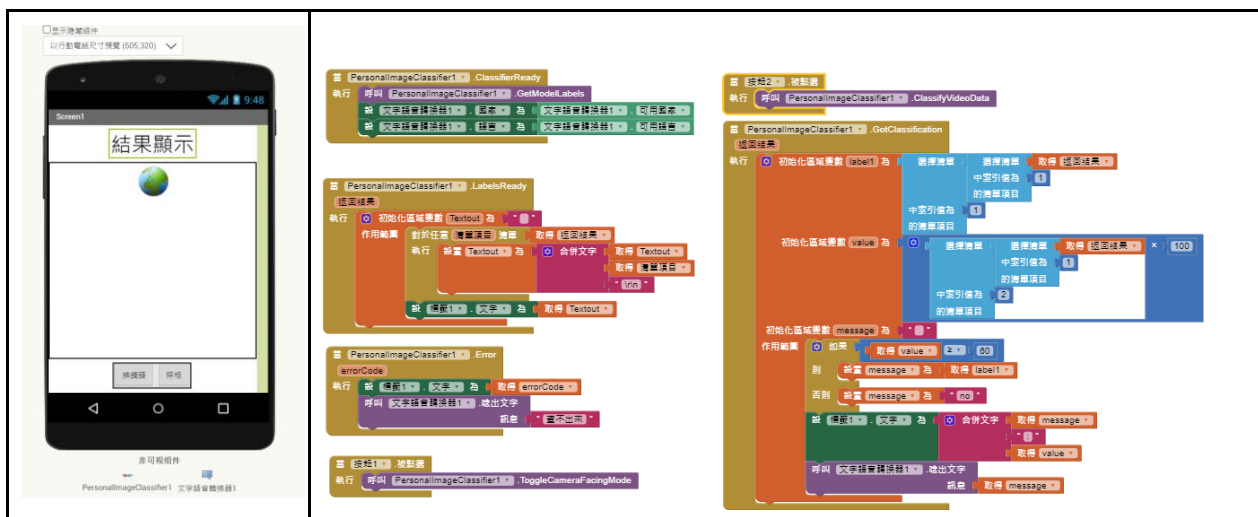
(三) 研讀 Python 程式編寫教學

(四) 研讀 Google Mediapipe 操作教學

(五) 研讀語音合成程式編寫教學

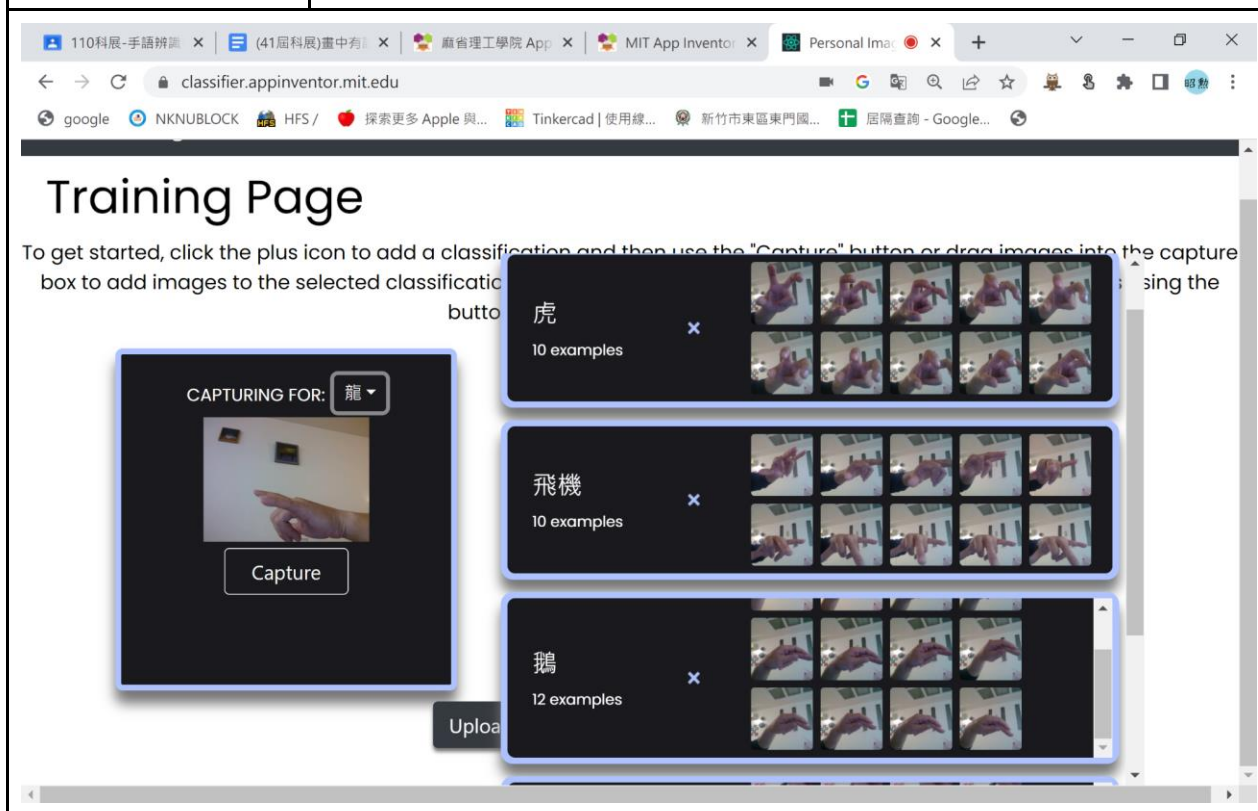
(六) 研讀教育部線上常用手語辭典

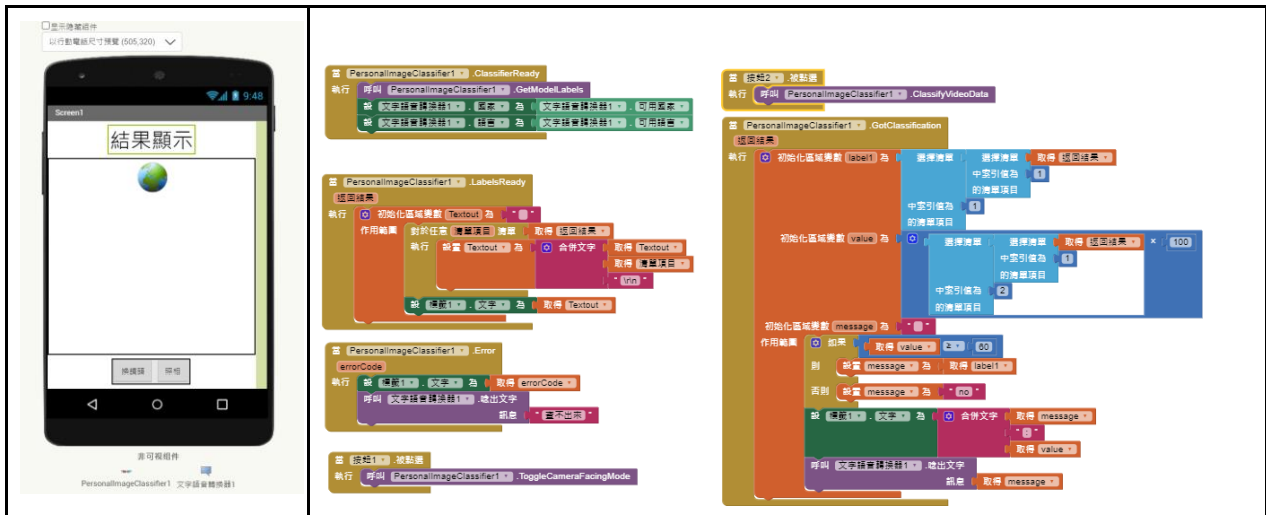
二、利用 Personal Image Classifier 建立模組，搭配 MIT App Inventor 編寫手語辨識 APP 供手機下載使用，並加以測試手機辨識能力。



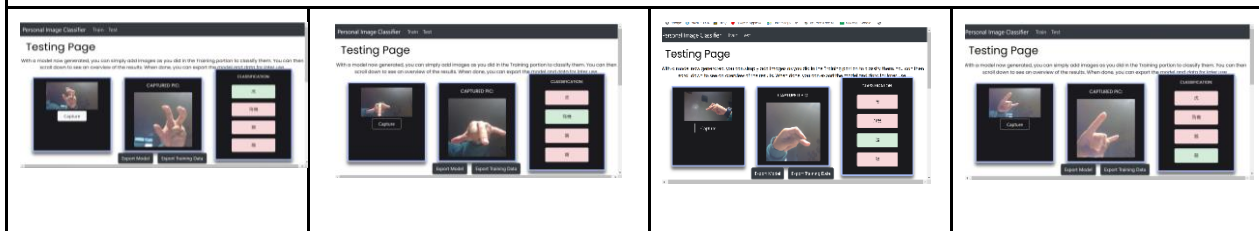
手機 APP 的畫面編排

手機 APP 的程式設計





拍攝手勢圖檔，建立資料庫



Personal Image Classifier 訓練模組情形

三、照相存檔，建立可供對照的手勢圖庫，搭配 Python 程式。

四、修改程式語法

(一) 例 1：起初將「`aspeak -t "秀" -l zh-CN`」直接寫在程式內，但發現不合乎程式語法，所以將程式修改成下列函式語法：

```
import os
```

```
os.popen('aspeak -t "秀" -l zh-CN')
```

(二) 例 2：起初語音是使用另一個程式執行，程式編寫如下：

```
import os
```

```
f = open(path, 'w') ←選擇要編輯的 TXT 檔案
```

```
print(text, file=f) ←將變數 text 寫入 TXT 檔案內
```

f.close() ←結束編輯

os.system("start TEST.bat ") ←開啟 BAT 檔案，讓 BAT 檔案朗讀文字

→因中間是透過一個檔案傳輸文字，但發現會延遲，所以將程式修改如下：

```
import os
```

```
if text == "秀" :
```

```
    os.popen('aspeak -t "秀" -l zh-CN')
```

伍、研究結果

一、完成手機手語辨識 APP 及 Python 程式編寫。

二、手機 APP 辨識後能成功進行語音說明，但辨識正確度差。



三、Python 程式辨識成功，且可語音合成，手勢後轉語音時畫面會延遲(lag)。目前資料庫有建立 0、1、2、3、4、5、6、7、8、9、秀、姊、兄等字，均可辨識並進行語音說明。

程式碼：

```
#import 需要的函示庫
```



```

#!/bin/python3
from easydl import clear_output
import os
import cv2
import mediapipe as mp
import math

mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles
mp_hands = mp.solutions.hands

# 根據兩點的座標，計算角度
def vector_2d_angle(v1, v2):
    v1_x = v1[0]
    v1_y = v1[1]
    v2_x = v2[0]
    v2_y = v2[1]
    try:
        angle_ =
            math.degrees(math.acos((v1_x*v2_x+v1_y*v2_y)/(((v1_x**2+v1_y**2)**0.5)*((v
                2_x**2+v2_y**2)**0.5))))
    except:
        angle_ = 180
    return angle_

# 根據傳入的 21 個節點座標，得到該手指的角度
def hand_angle(hand_):
    angle_list = []

    angle_ = vector_2d_angle(

```

```

        ((int(hand_[0][0])- int(hand_[2][0])),(int(hand_[0][1])-int(hand_[2][1]])),
        ((int(hand_[3][0])- int(hand_[4][0])),(int(hand_[3][1])- int(hand_[4][1]]))
    )
angle_list.append(angle_)

angle_ = vector_2d_angle(
    ((int(hand_[0][0])-int(hand_[6][0])),(int(hand_[0][1])- int(hand_[6][1]])),
    ((int(hand_[7][0])- int(hand_[8][0])),(int(hand_[7][1])- int(hand_[8][1]]))
)
angle_list.append(angle_)

angle_ = vector_2d_angle(
    ((int(hand_[0][0])- int(hand_[10][0])),(int(hand_[0][1])- int(hand_[10][1]])),
    ((int(hand_[11][0])- int(hand_[12][0])),(int(hand_[11][1])- int(hand_[12][1]]))
)
angle_list.append(angle_)

angle_ = vector_2d_angle(
    ((int(hand_[0][0])- int(hand_[14][0])),(int(hand_[0][1])- int(hand_[14][1]])),
    ((int(hand_[15][0])- int(hand_[16][0])),(int(hand_[15][1])- int(hand_[16][1]]))
)
angle_list.append(angle_)

angle_ = vector_2d_angle(
    ((int(hand_[0][0])- int(hand_[18][0])),(int(hand_[0][1])- int(hand_[18][1]])),
    ((int(hand_[19][0])- int(hand_[20][0])),(int(hand_[19][1])- int(hand_[20][1]]))
)
angle_list.append(angle_)
return angle_list

```

根據手指角度的串列內容，返回對應的手勢名稱

```
def hand_pos(finger_angle):
```

```
    f1 = finger_angle[0] # 大拇指角度
```

```
    f2 = finger_angle[1] # 食指角度
```

```
    f3 = finger_angle[2] # 中指角度
```

```
    f4 = finger_angle[3] # 無名指角度
```

```
    f5 = finger_angle[4] # 小拇指角度
```

小於 50 表示手指伸直，大於等於 50 表示手指捲縮

```
if f1>=50 and f2>=50 and f3<50 and f4>=50 and f5>=50:
```

```
    return '兄'
```

```
elif f1>=50 and f2>=50 and f3>=50 and f4>=50 and f5>=50:
```

```
    return '0'
```

```
elif f1>=50 and f2<50 and f3>=50 and f4>=50 and f5>=50:
```

```
    return '1'
```

```
elif f1>=50 and f2<50 and f3<50 and f4>=50 and f5>=50:
```

```
    return '2'
```

```
elif f1>=50 and f2<50 and f3<50 and f4<50 and f5>50:
```

```
    return '3'
```

```
elif f1>=50 and f2<50 and f3<50 and f4<50 and f5<50:
```

```
    return '4'
```

```
elif f1<50 and f2<50 and f3<50 and f4<50 and f5<50:
```

```
    return '5'
```

```
elif f1<50 and f2<50 and f3>=50 and f4>=50 and f5>=50:
```

```
    return '6'
```

```
elif f1<50 and f2<50 and f3<50 and f4>=50 and f5>=50:
```

```
    return '7'
```

```
elif f1<50 and f2<50 and f3<50 and f4<50 and f5>=50:
```

```

    return '8'
elif f1<50 and f2<50 and f3<50 and f4<50 and f5<=50:
    return '9'
elif f1<50 and f2<50 and f3>50 and f4>50 and f5>50:
    return '10'
elif f1>50 and f2<50 and f3>50 and f4>50 and f5<50:
    return '秀'
elif f1>50 and f2>50 and f3>50 and f4<50 and f5>50:
    return '姊'
else:
    return "

```

```

cap = cv2.VideoCapture(0)# 讀取攝影機
fontFace = cv2.FONT_HERSHEY_SIMPLEX # 印出文字的字型
lineType = cv2.LINE_AA # 印出文字的邊框

```

```

# mediapipe 啟用偵測手掌

```

```

with mp_hands.Hands(
    model_complexity=0,
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5) as hands:

```

```

if not cap.isOpened():
    print("Cannot open camera")
    exit()

```

```

w, h = 540, 310

```

```

while True:
    ret, img = cap.read()
    if not ret:

```

```

print("Cannot receive frame")
break
img2 = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) # 轉換成 RGB 色彩
results = hands.process(img2)# 偵測手勢
if results.multi_hand_landmarks:
    for hand_landmarks in results.multi_hand_landmarks:
        finger_points = []# 記錄手指節點座標的串列
        for i in hand_landmarks.landmark:
            # 將 21 個節點換算成座標，記錄到 finger_points
            x = i.x*w
            y = i.y*h
            finger_points.append((x,y))
        if finger_points:

            finger_angle = hand_angle(finger_points) # 計算手指角度，回傳長度為 5
            的串列
            #print(finger#_angle)# 印出角度 (有需要就開啟註解)
            text = hand_pos(finger_angle)# 取得手勢所回傳的內容
            clear_output() # 清除輸出
            print (text)

            path = 'test.txt'
            #語音朗讀
            if text == "兄" :
                os.popen('aspeak -t "兄" -l zh-CN')
            elif text == "1" :
                os.popen('aspeak -t "1" -l zh-CN')
            elif text == "0" :
                os.popen('aspeak -t "0" -l zh-CN')
            elif text == "2" :

```

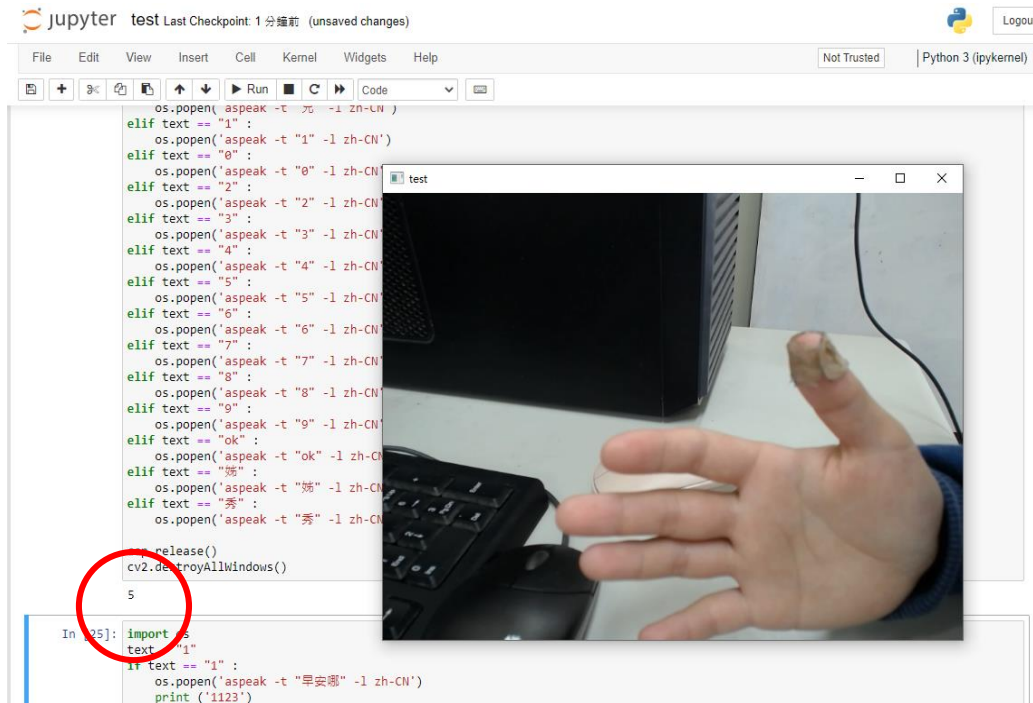
```

        os.popen('aspeak -t "2" -l zh-CN')
elif text == "3" :
        os.popen('aspeak -t "3" -l zh-CN')
elif text == "4" :
        os.popen('aspeak -t "4" -l zh-CN')
elif text == "5" :
        os.popen('aspeak -t "5" -l zh-CN')
elif text == "6" :
        os.popen('aspeak -t "6" -l zh-CN')
elif text == "7" :
        os.popen('aspeak -t "7" -l zh-CN')
elif text == "8" :
        os.popen('aspeak -t "8" -l zh-CN')
elif text == "9" :
        os.popen('aspeak -t "9" -l zh-CN')
elif text == "ok" :
        os.popen('aspeak -t "ok" -l zh-CN')
elif text == "姊" :
        os.popen('aspeak -t "姊" -l zh-CN')
elif text == "秀" :
        os.popen('aspeak -t "秀" -l zh-CN')
#f = open(path, 'w')
#print(text, file=f)
#f.close()
#subprocess.Popen("TEST.bat")
#os.system("TEST.bat ")
#subprocess.Popen('py sound.py')
cv2.imshow('test', img)
if cv2.waitKey(5) == ord('q'):
    break

```

cap.release()

cv2.destroyAllWindows()



↑ 對著鏡頭比出手勢「5」，系統能正確辨識出「5」：

陸、討論

一、手機 APP 和 Python 程式那一個更適合發展手勢辨識功能？

二、為何手勢辨識後轉語音畫面會延遲？

(一) 是否為網速的問題？

(二) 是否為程式平台被多數人使用佔用資源？

(三) 是否為程式語法的問題，造成執行等待或中斷？

三、能否辨識更多中文手勢或英文字母手勢？

因時間有限，目前只有挑選相似度不高的文字實驗，所以有以下問題：

(一) 若資料庫增加較為相似的字是否會辨識錯誤或延遲更久？

(二) 若資料庫增加英文字母手勢，是否也能成功辨識？

四、是否可以做出動態或雙手的手勢辨識？

目前選用的都是教育部線上常用手語辭典六十基本手勢中的文字，大部分是靜態的手勢辨識，所以有以下問題：

(一) 雙手的手勢是否可以辨識？

(二) 動態的手勢是否可以辨識？

柒、結論

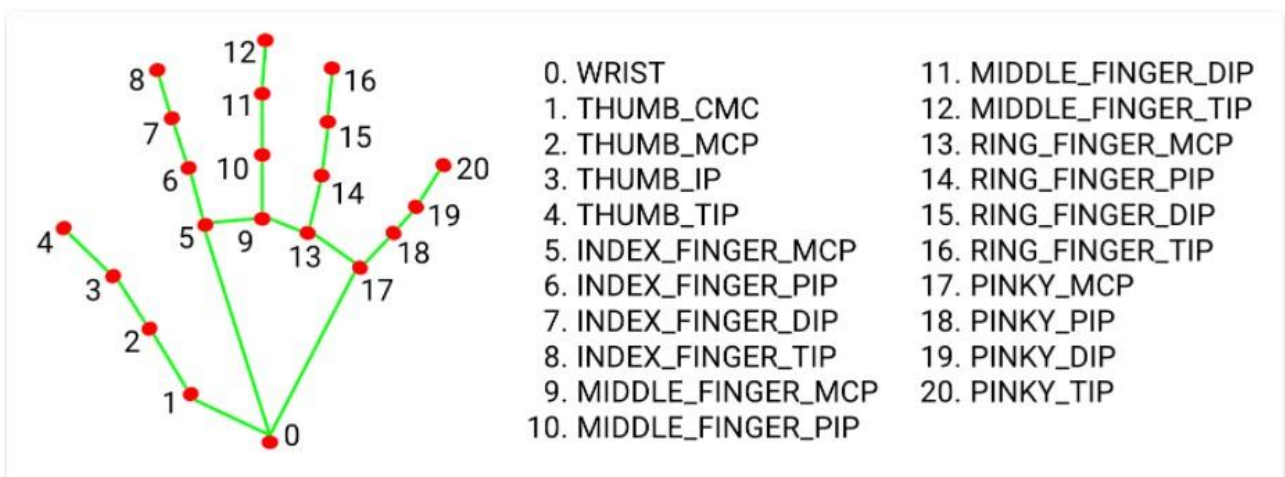
一、MIT App Inventor 搭配 Personal Image Classifier 是純圖像記憶式的辨識，對於 5 隻手指的組合變化辨識訓練困難，且常當機，不適合大量字彙的訓練。而 Python 程式搭配 Google Mediape 平台是以手指關節的節點作為判斷的依據，較為精準，辨識能力較強。

二、目前檢視 Python 編寫的程式，大致沒有問題，猜測是平台系統資源、學校網路被大量占用等問題，未來有機會會去作證實。

三、英文字母手勢可以辨識，一開始在做手機 APP 時也有加入資料庫，但有些手勢動作需要移動手(如下圖中的 J、Z)和有些很像的(如 H、R、U、V)電腦可能會辨識錯誤，手勢相似的部分，在拍攝相片建立資料庫時是否可以改善，仍在積極嘗試中。



四、目前使用的原理是利用手指的節點作為辨識的依據（如下圖），雙手的節點過多，Google Mediapipe 尚無法作正確的辨識。另外，還沒找到支援動態偵測的平台，所以對於動態的手勢暫時不能辨識。



五、目前因辨識平台因素，無法做到雙手偵測及動態偵測，在更進一步的開發前，建議大家還是可以多學習手語。

捌、參考文獻資料

一、Python

(一) 基本介紹

使用 Anaconda - Python 教學。STEAM 教育學習網。取自：

<https://steam.oxxostudio.tw/category/python/info/anaconda.html>

(二) Mediapipe

1.使用 MediaPipe - AI 影像辨識教學 (Python)。STEAM 教育學習網。取自：

<https://steam.oxxostudio.tw/category/python/ai/ai-mediapipe.html>

2. Mediapipe 手掌偵測 (hands) - AI 影像辨識教學 (Python)。STEAM 教育學習網。取自：

<https://steam.oxxostudio.tw/category/python/ai/ai-mediapipe-hand.html>

3. Mediapipe 手勢辨識 - AI 影像辨識教學 (Python)。STEAM 教育學習網。取自：

<https://steam.oxxostudio.tw/category/python/ai/ai-mediapipe-gesture.html>

(三) OpenCV

OpenCV 函式庫 - OpenCV 教學 (Python)。STEAM 教育學習網。取自：

<https://steam.oxxostudio.tw/category/python/ai/ai-mediapipe-gesture.html>

二、語音合成

媲美真人配音！微软语音合成助手 - 免费文字转换音频的 TTS 工具。異次元/微軟世界網站。取自：<https://www.iplaysoft.com/azure-tts.html>

三、OS 函式庫

pythonOS：系統指令和應用。IT 邦幫忙網站。取自：

<https://ithelp.ithome.com.tw/article`s/10226650>

四、手語辭典

常用手語辭典。教育部。取自：<https://special.moe.gov.tw/signlanguage/>

五、辨識平台

mediapipe。取自：<https://google.github.io/mediapipe/>