



ID 984093591409



ID 546543867893



ID 325465645710



ID 870293477729



點名系統

指導老師：張顯盛老師

組員：林孝陽、莊子弘、許由、許嘉安

2022/1/11

目錄

- ▶ 前言
- ▶ 理論探討
- ▶ 專題設計
- ▶ 遭遇困難、解決方法
- ▶ 成果
- ▶ 未來展望
- ▶ 參考資料
- ▶ Q&A

前言

- 專題背景：在學校上課遲到時，可能會發生人為的疏忽，或是老師剛好不在，而登記錯誤。製作一台可以正確傳達訊息的點名系統機器就能避免這種情況發生。
- 目的：製作出班級點名系統，辨識出是否有同學遲到或是曠課的情形。

理論探討

➤ 硬體

➤ 軟體

硬體

- 伺服馬達
- Arduino Uno版
- 羅技C922網路攝像頭

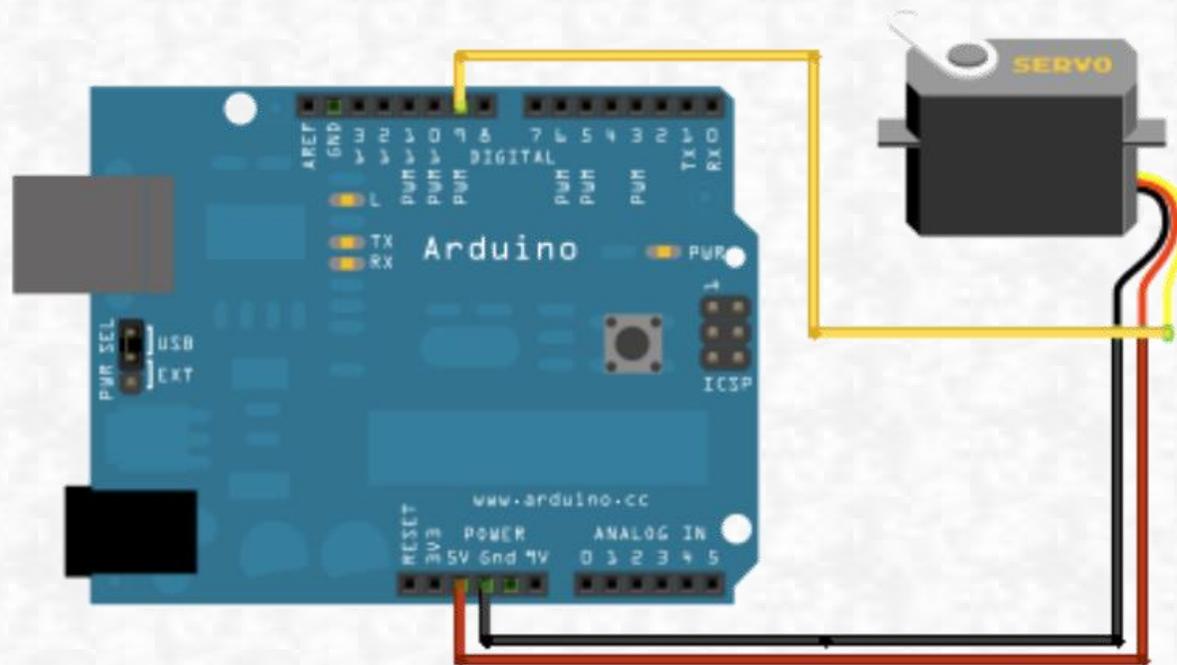
伺服馬達

➤ 用途：帶動雲台轉動



ARDUINO UNO版

- 用途：控制馬達轉動的角度



攝像頭

▶ 用途：拍攝人臉

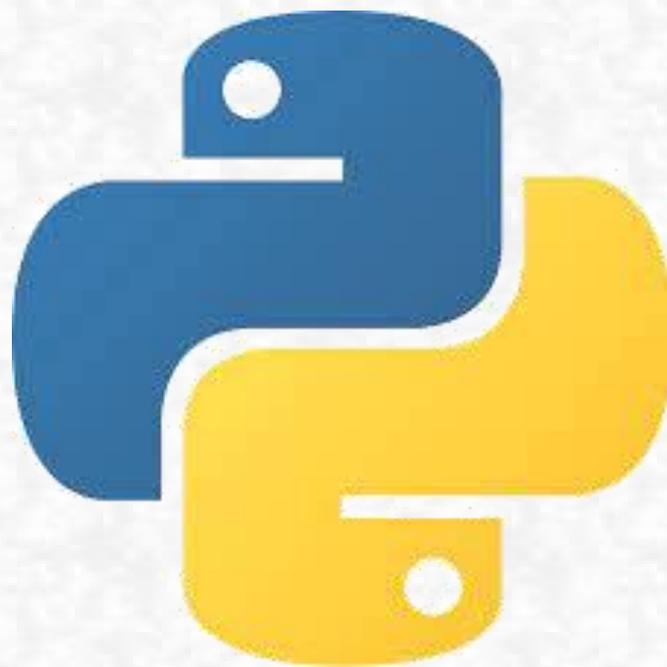


軟體應用程式

- Python
- Arduino
- Tinkercad
- Tensorflow
- Open cv
- Keras

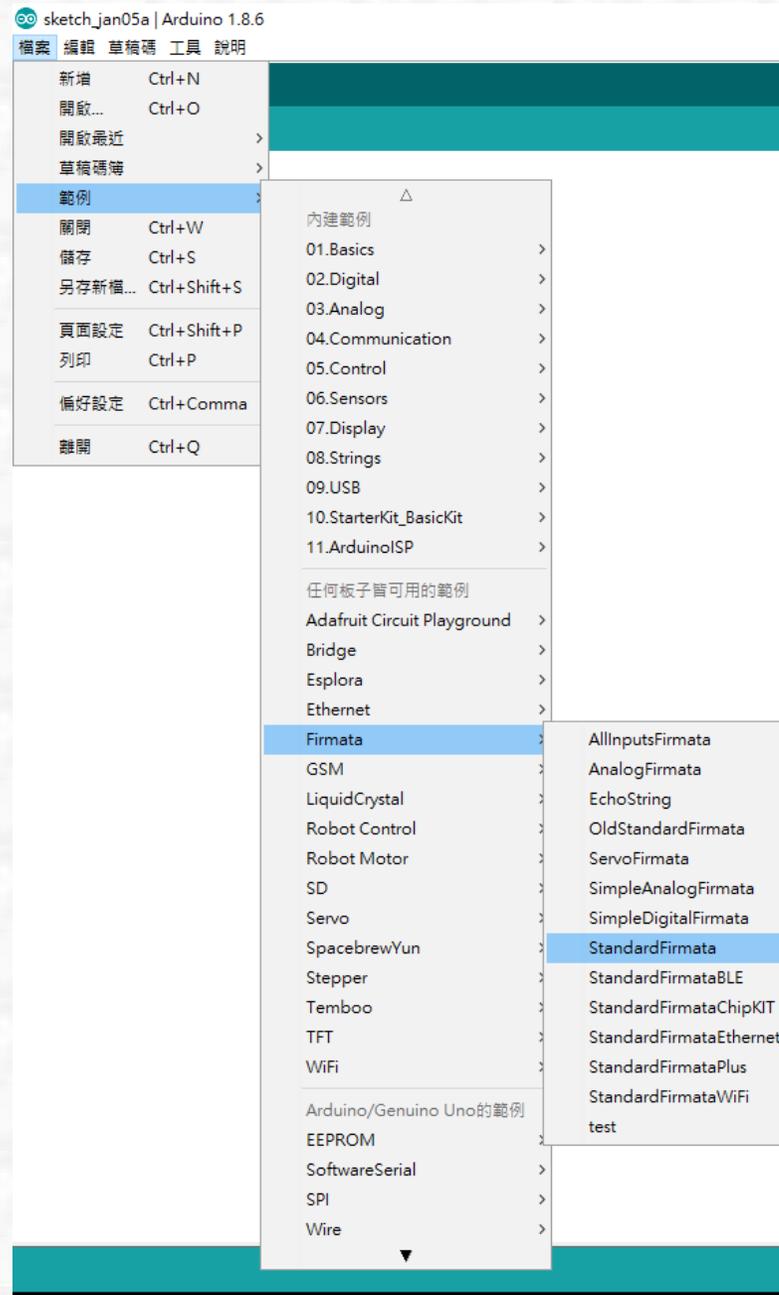
PYTHON

- ▶ 用途：人臉辨識與馬達操作
- ▶ 原因：跨平台



ARDUINO

➤用途：載入與Python結合的函式庫



TINKERCAD

➤ 用途：3D建模



OPENCV

OpenCV是一個跨平台的電腦視覺庫，可用於開發即時的圖像處理、電腦視覺以及圖形識別程式。

應用：

- 臉部辨識
- 運動辨識
- 手勢辨識
- 物體辨識
- 擴增實境

TENSORFLOW

TensorFlow是一個機器學習框架，提供一系列的工作流程，可使用 Python 或 JavaScript 來開發及訓練模型。



Python 開發

CPU GPU TPU

1

載入及預先處理資料

TENSORFLOW

建立 TensorFlow 輸入管線

[探索](#)



2

建構、訓練及重複使用模型

TENSORFLOW

使用 Keras 建構及訓練模型

[探索](#)



3

部署

TENSORFLOW

使用 Python 部署



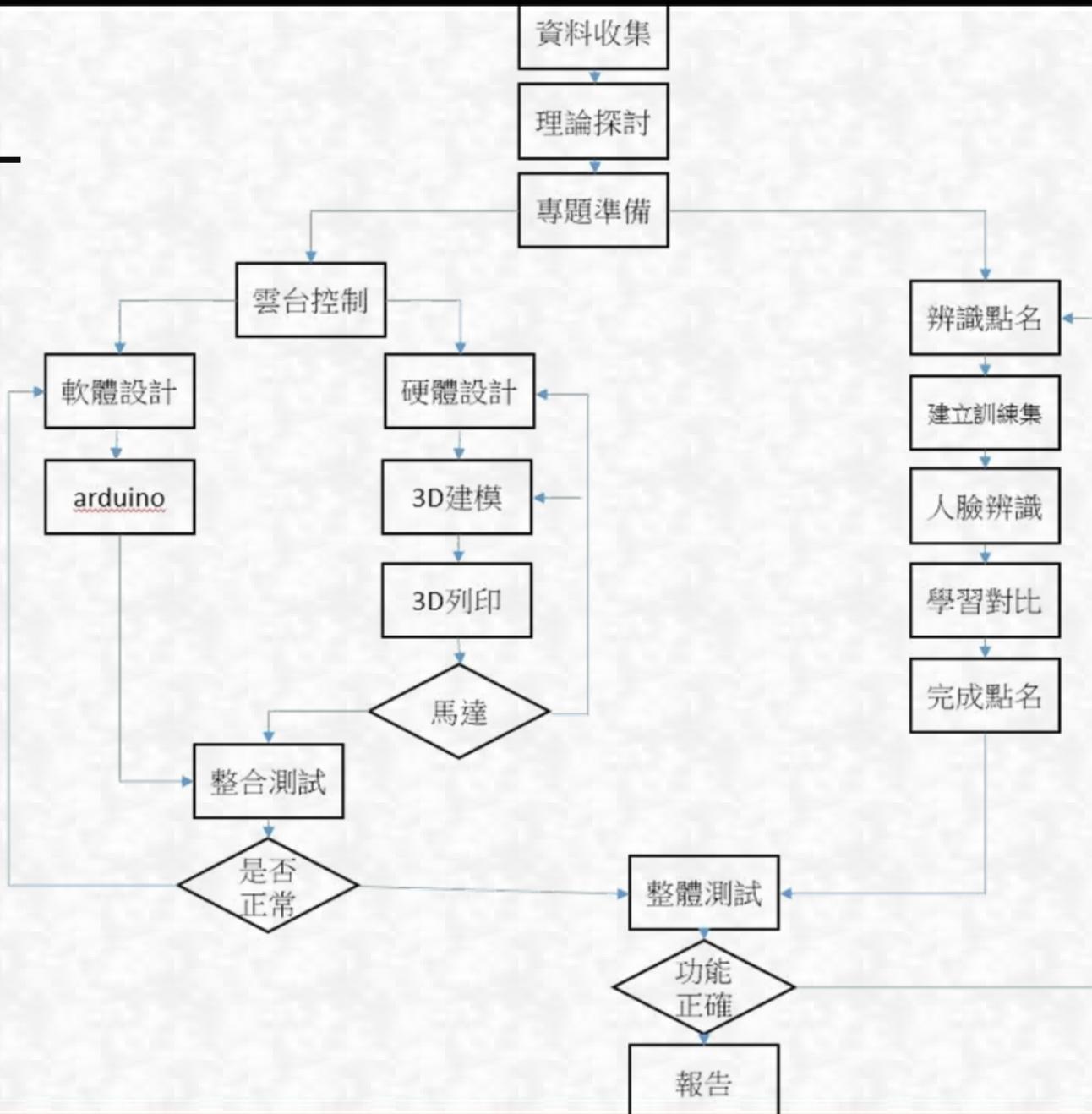
KERAS

Keras是個高度模組化的函式庫，它可以讓用戶快速的去建造神經網路並訓練模型。

TensorFlow有提供tf.keras可以使用，tf.keras是基於Keras標準並且與TensorFlow整合。

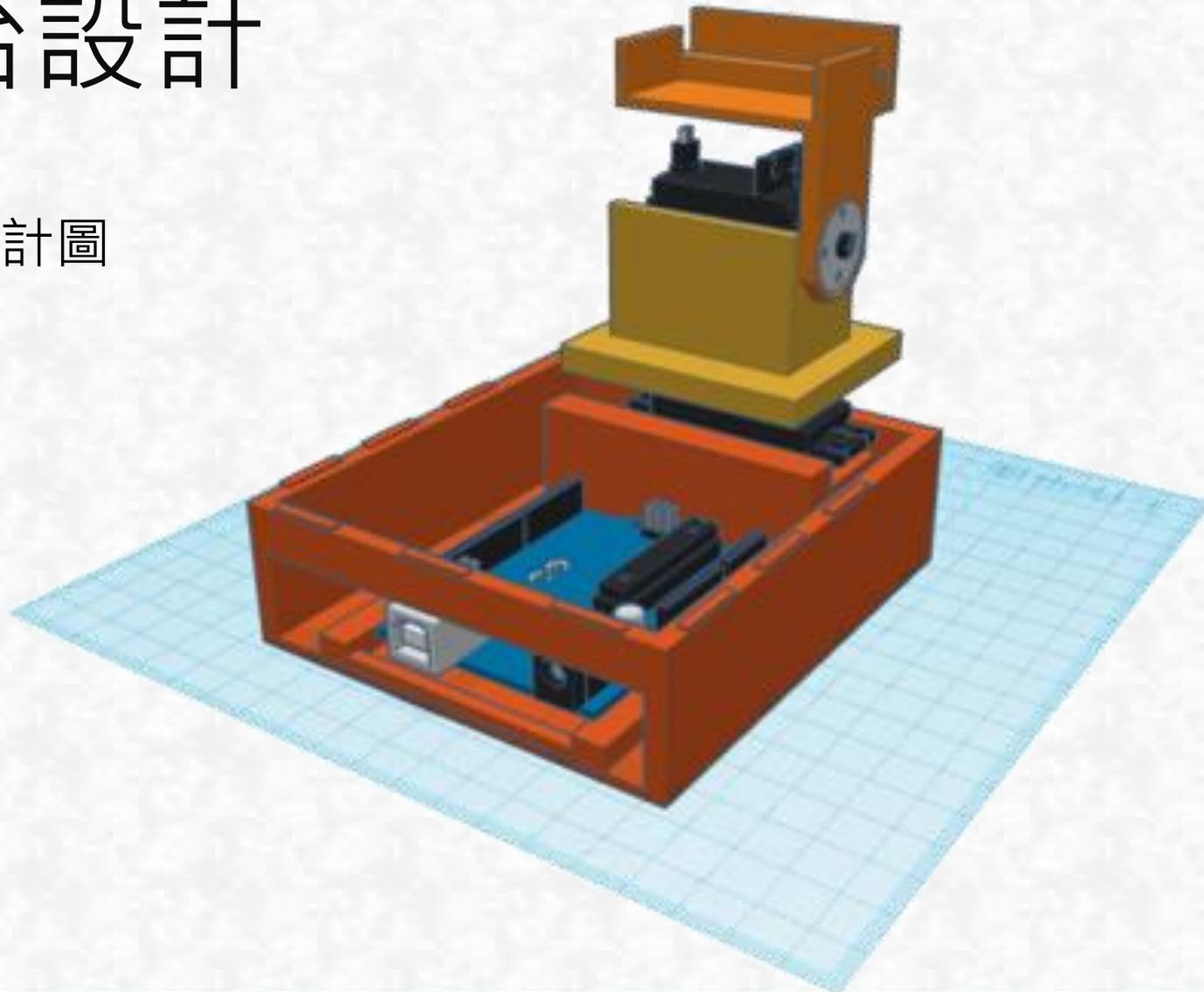
專題設計

- 雲台設計
- 人臉辨識
- 整體控制
- 點名
- 通知訊息



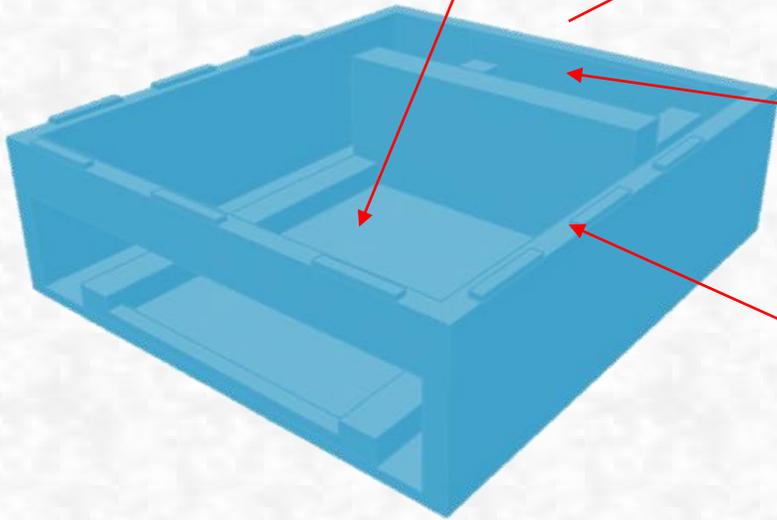
雲台設計

▶ 整體設計圖

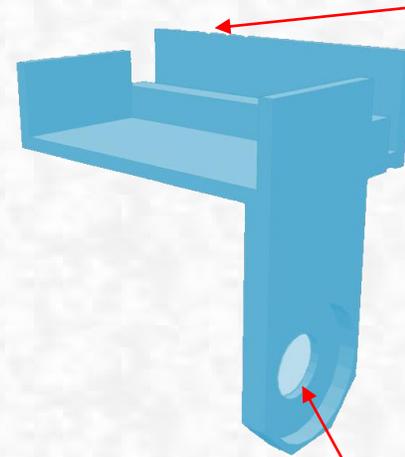
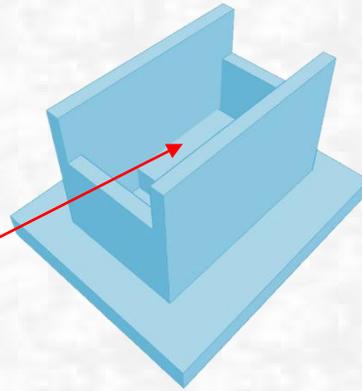


機構

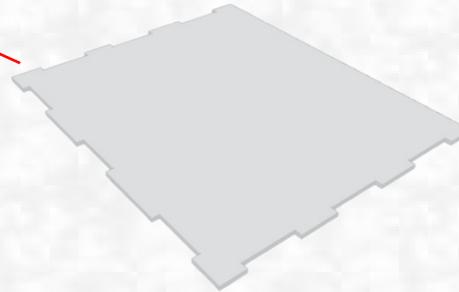
底座



機身

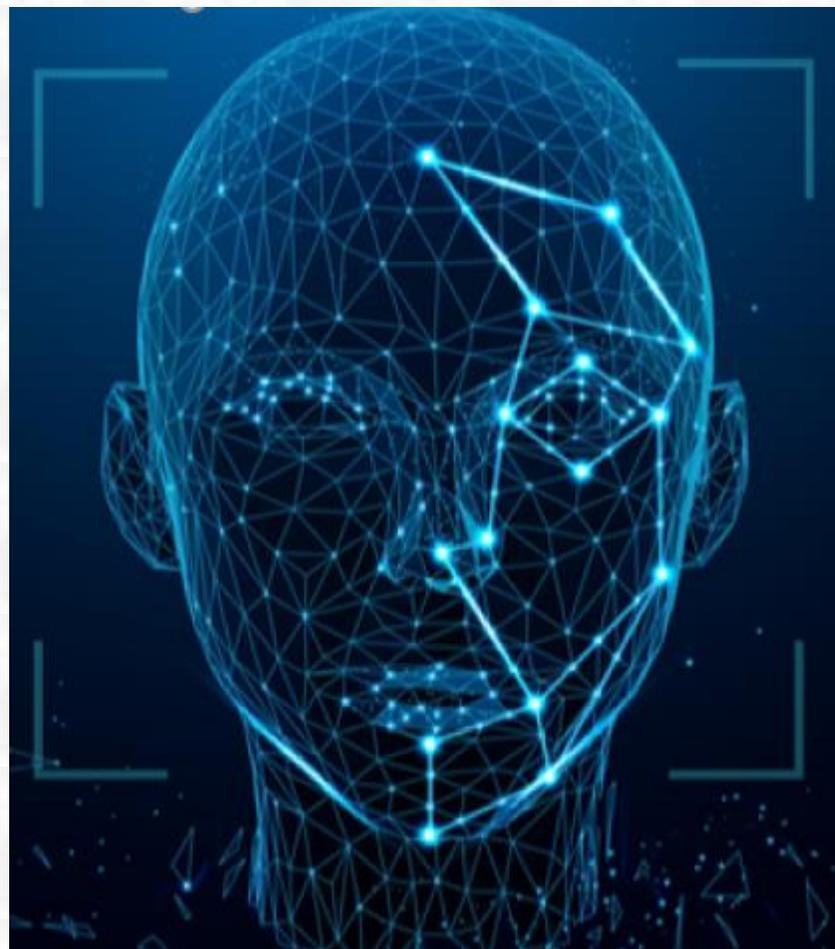


蓋子



人臉辨識

- 捕捉人臉
- 建立模型
- 訓練模型
- 人臉辨識



捕捉人臉

利用opencv2裡的cascade.detectMultiScal函式做臉部的辨識。

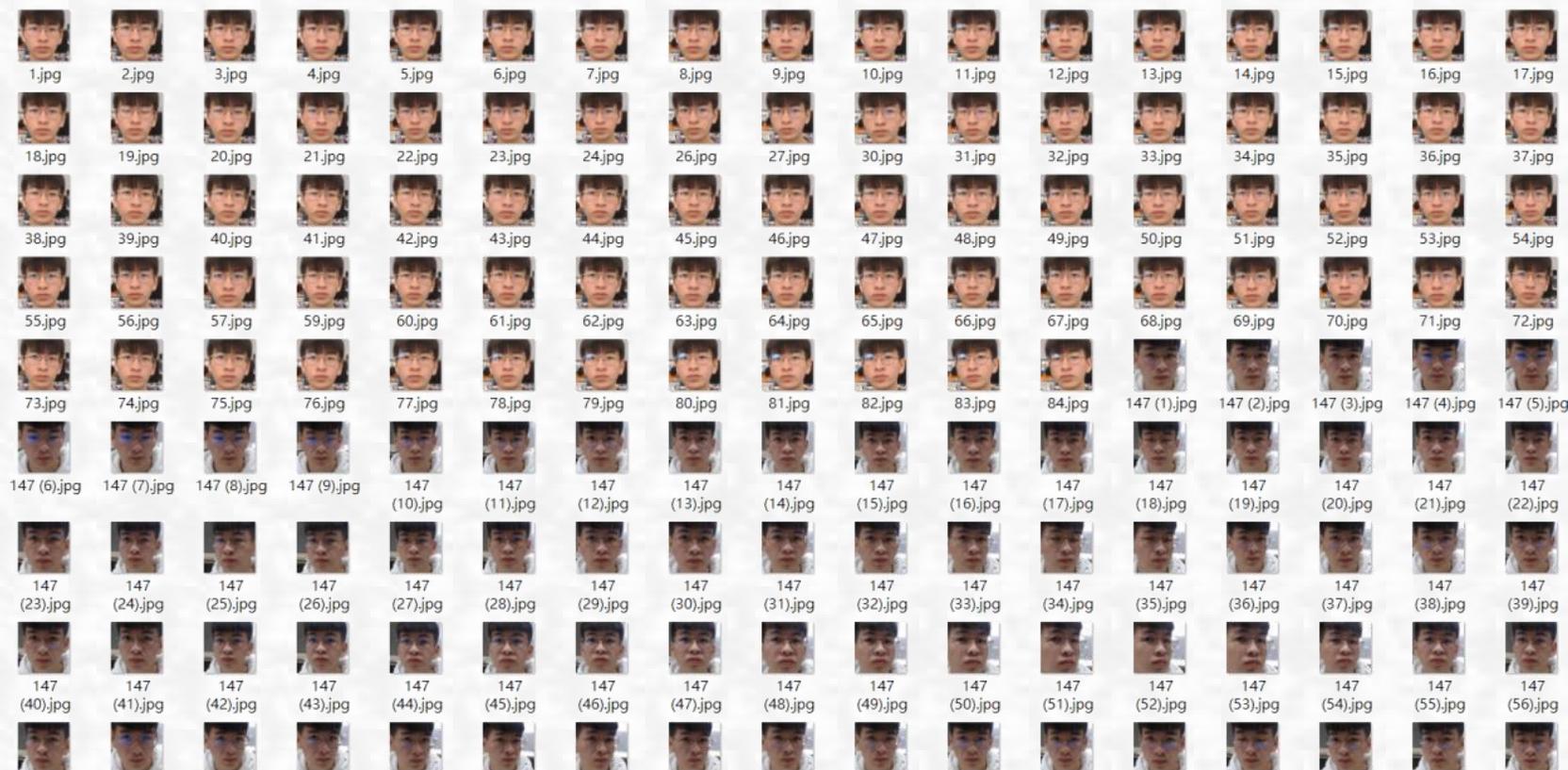
- gray：每個點的RGB數據變成了一維的灰度，可以降低計算強度。
- ScaleFactor：每次搜尋方塊減少的比例。

- minNeighbors：每個目標至少檢測到幾次以上，才可被認定是真數據。
- minSize：設定數據搜尋的最小尺寸。

```
facerect = cascade.detectMultiScale(frame_gray,  
                                     scaleFactor=1.2,  
                                     minNeighbors=5,  
                                     minSize=(12, 12))
```

訓練集

4個環境，每個環境取80張，共320張

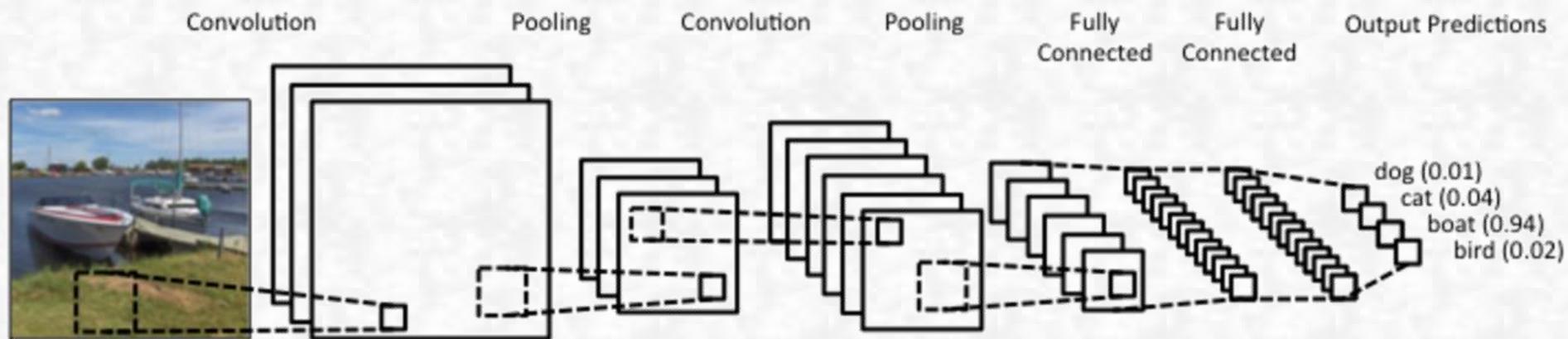


建立模型

利用tensorflow加keras建立一個CNN卷積神經網路

CNN卷積神經網路：

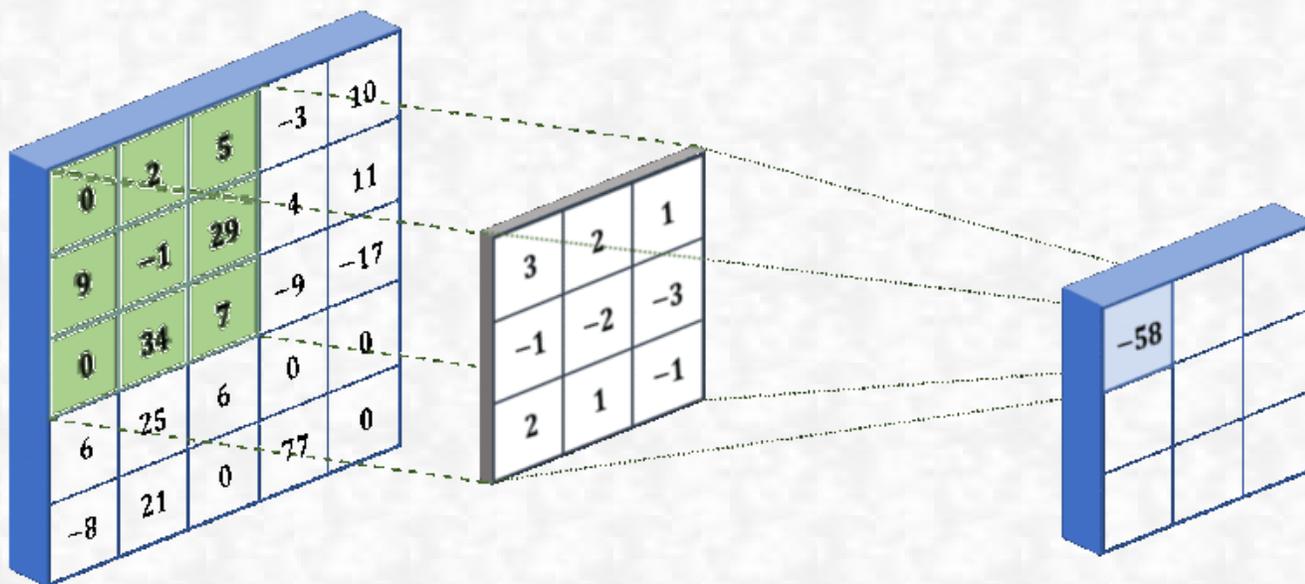
- 卷積層
- 池化層
- 全連接層



卷積層

卷積神經網路中是負責對輸入的影像進行特徵擷取，在經過卷積計算後得到輸出的特徵圖。

- 卷積核
- 填補
- 步幅



池化層

池化層是用來將輸入的影像進行壓縮，並且強調影像的主要特徵。

池化層的運算過程與卷積層相似，一樣是用固定大小的區塊在輸入的影像上滑動。

沒有參數。

Feature Map

1	3	2	5
0	8	7	0
6	3	1	9
2	3	0	7

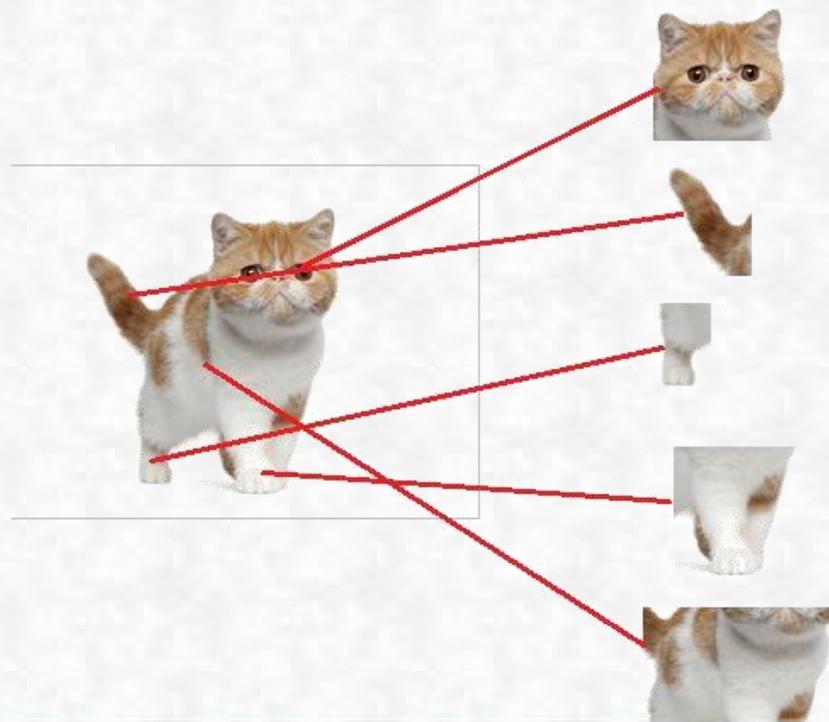
Max Pooling

Average Pooling

Max pooling and Average pooling of a 4x4 input map with a 2x2 pooling window and stride = 2
thedatabus.io

全連接層

全連結層就是最後的分類器，將前面所擷取出來的特徵，經過權重的計算之後，來辨識出這個所輸入的圖像到底屬於哪一個分類。



訓練模型

- ▶ 找到模型中參數調整的方向與大小
- ▶ 對參數進行調整

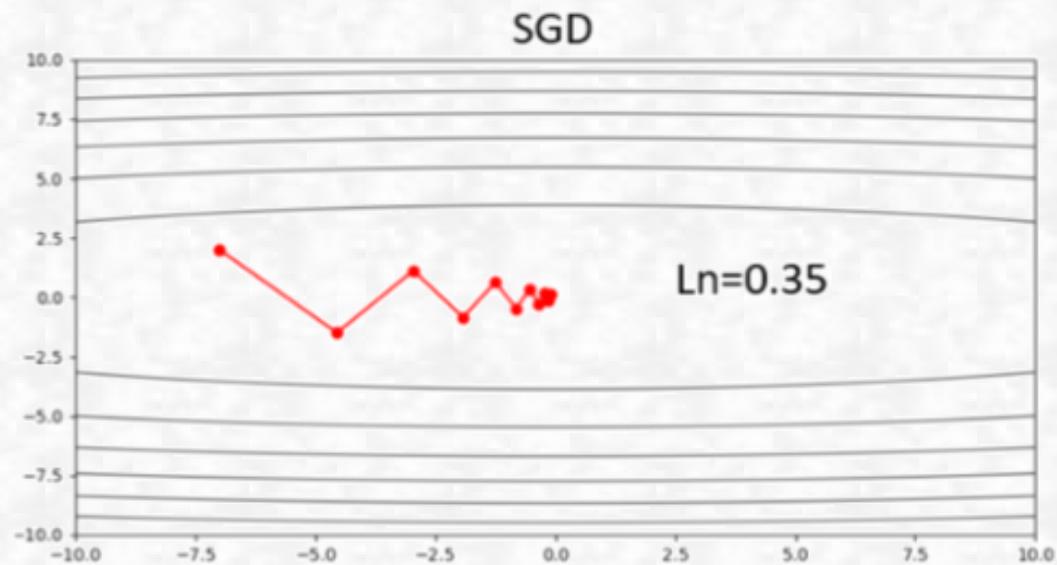
訓練模型的過程就是不斷的重複這兩個行為，直到模型的輸出達到我們滿意的水準。

利用SGD+Momentum優化器

SGD

➤ SGD(隨機梯度下降法)

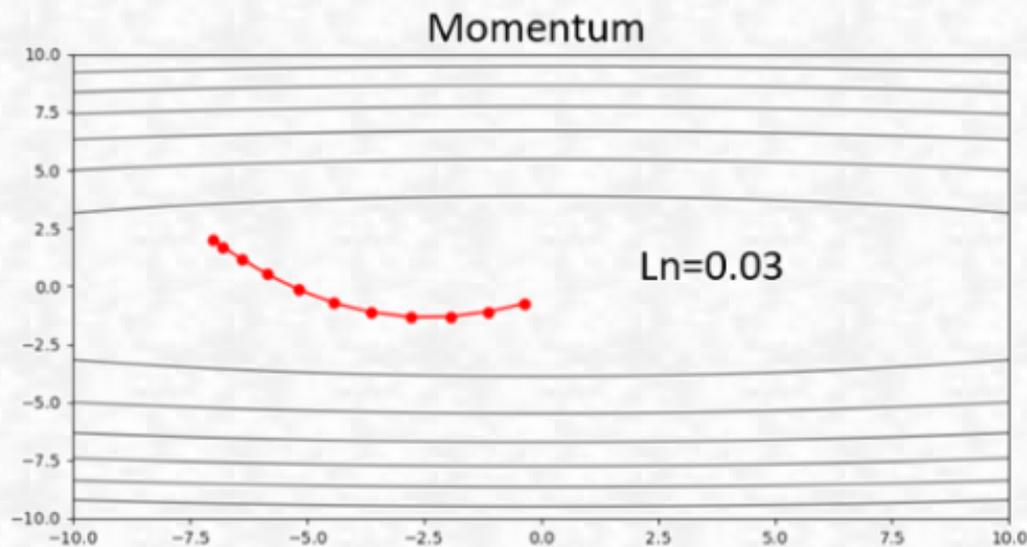
找出參數的梯度(微分)，往梯度的方向更新參數。



MOMENTUM

➤ Momentum (運動量)

為模擬物理動量的概念，在同方向的維度上學習速度會變快，方向改變的時候學習速度會變慢。



人臉辨識

- 載入模型
- 識別人臉
- 辨識人臉

載入模型

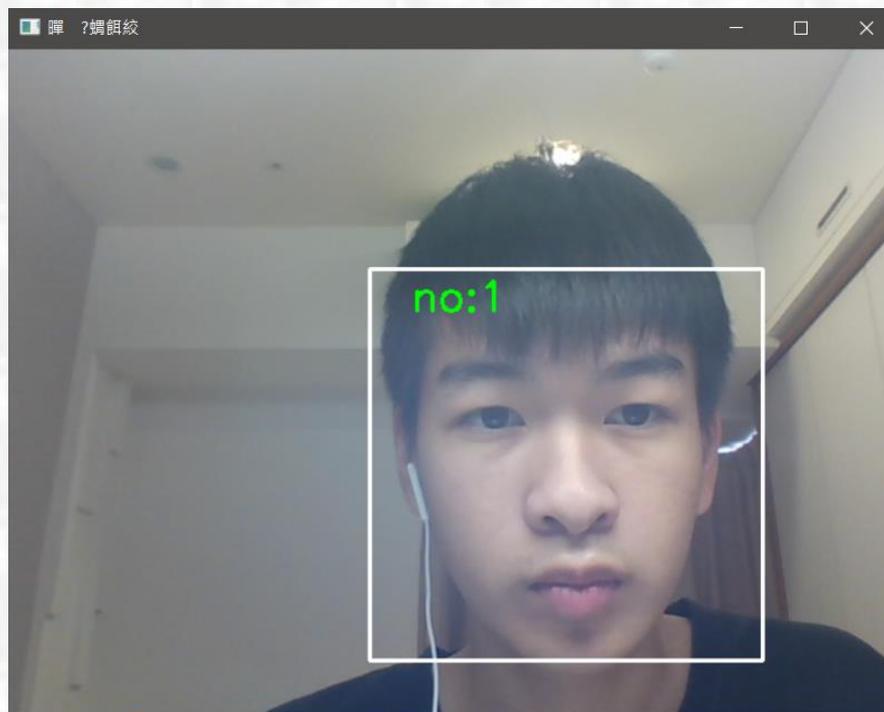
載入訓練好的模型

 model_15.h5	2021/11/10 下午 06:59	H5 檔案	924 KB
 model_16.h5	2021/11/13 下午 05:54	H5 檔案	924 KB
 model_17.h5	2021/11/15 下午 07:25	H5 檔案	924 KB
 model_18.h5	2021/12/22 下午 07:28	H5 檔案	924 KB
 model_19.h5	2021/12/22 下午 09:32	H5 檔案	924 KB
 model_20.h5	2022/1/4 下午 01:38	H5 檔案	888 KB
 model_21.h5	2022/1/4 下午 02:11	H5 檔案	924 KB

```
model = Model()  
model.load_model('./store/model_21.h5')
```

識別人臉

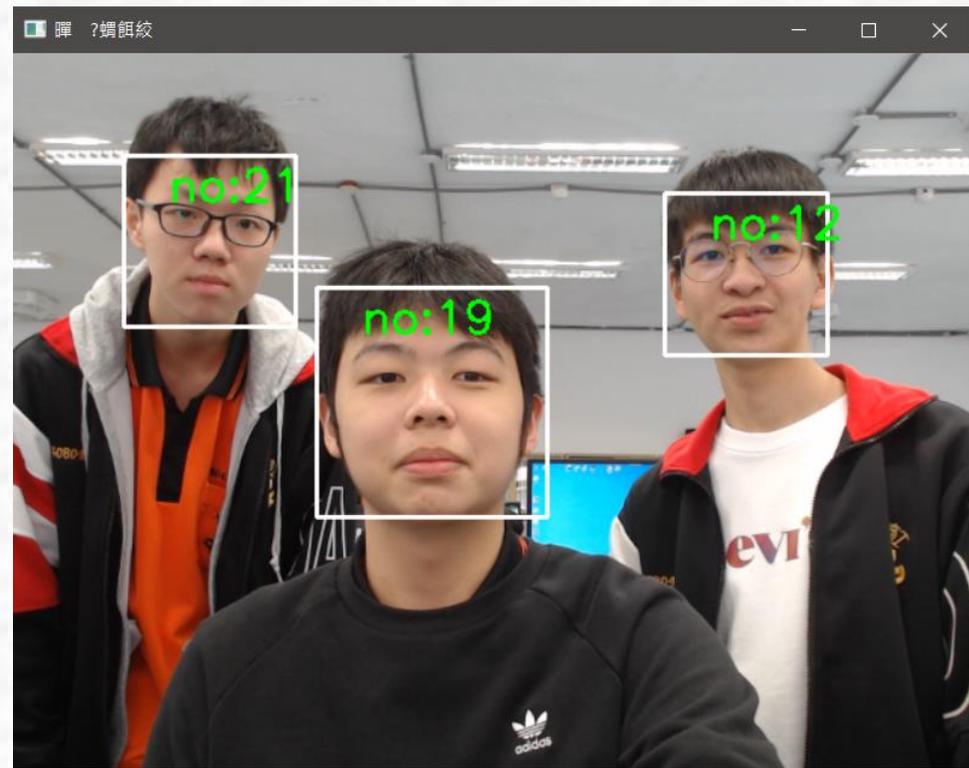
利用前面捕捉人臉提到的分類器(opencv2裡的cascade.detectMultiScal函式) 辨識哪個區域為人臉。



辨識人臉

利用opencv2辨識人臉並顯示辨識到的人臉號碼。

```
cv2.putText(frame, 'no:%s' % (result.argmax()),  
            (x + 30, y + 30),  
            font, 1, (0, 255, 0), 2)
```



整體控制

- ▶ 雲台水平轉到20、40、60、80度垂直往上轉動30度後回來
- ▶ 每轉完停下來進行人臉辨識五秒

```
def GoFace(pin, endang):  
    board.digital[pin].write(endang)  
    print(endang)  
    face()
```

```
while True:  
    GoFace(pin2, 0)  
    if face() == 0:  
        break  
    GoFace(pin1, 20)  
    if face() == 0:  
        break  
    GoFace(pin1, 40)  
    if face() == 0:  
        break  
    GoFace(pin1, 60)  
    if face() == 0:  
        break  
    GoFace(pin1, 80)  
    if face() == 0:  
        break  
    GoFace(pin2, 30)  
    if face() == 0:  
        break  
    GoFace(pin1, 60)  
    if face() == 0:  
        break  
    GoFace(pin1, 40)  
    if face() == 0:  
        break  
    GoFace(pin1, 20)  
    if face() == 0:  
        break  
    GoFace(pin1, 0)  
    if face() == 0:  
        break
```

點名

把辨識到的號碼數量超過設定值來判斷實到與未到

```
num=[]
renum=[]
if scores.count(12)<100:
    print('12未到')
    num.append(12)
elif scores.count(12)>100:
    print('12到')
    renum.append(12)
if scores.count(18)<100:
    print('18未到')
    num.append(18)
elif scores.count(18)>100:
    print('18到')
    renum.append(18)
if scores.count(19)<100:
    print('19未到')
    num.append(19)
elif scores.count(19)>100:
    print('19到')
    renum.append(19)
if scores.count(21)<100:
    print('21未到')
    num.append(21)
elif scores.count(21)>100:
    print('21到')
    renum.append(21)
print(num)
print(renum)
```

通知訊息

利用IFTTT傳送LINE訊息通知實到及未到

```
evt = 'rc' # 事件名稱
key = 'i23Fung-0jL46S1T8PWVHXsXIcG6kQHfF_p19W0UG7z'
url = ('https://maker.ifttt.com/trigger/{evt}' +
       '/with/key/{key}?value1={val1}').format(
    evt=evt,
    key=key,
    val1= num,

)

evt = 'rc2' # 事件名稱
key = 'i23Fung-0jL46S1T8PWVHXsXIcG6kQHfF_p19W0UG7z'
url2 = ('https://maker.ifttt.com/trigger/{evt}' +
        '/with/key/{key}?value1={val1}').format(
    evt=evt,
    key=key,
    val1= renum,

)

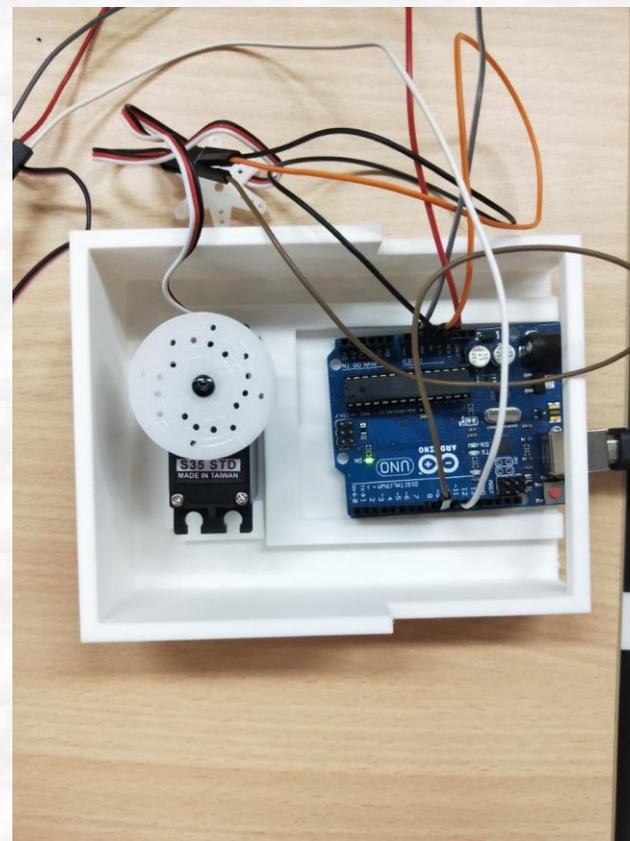
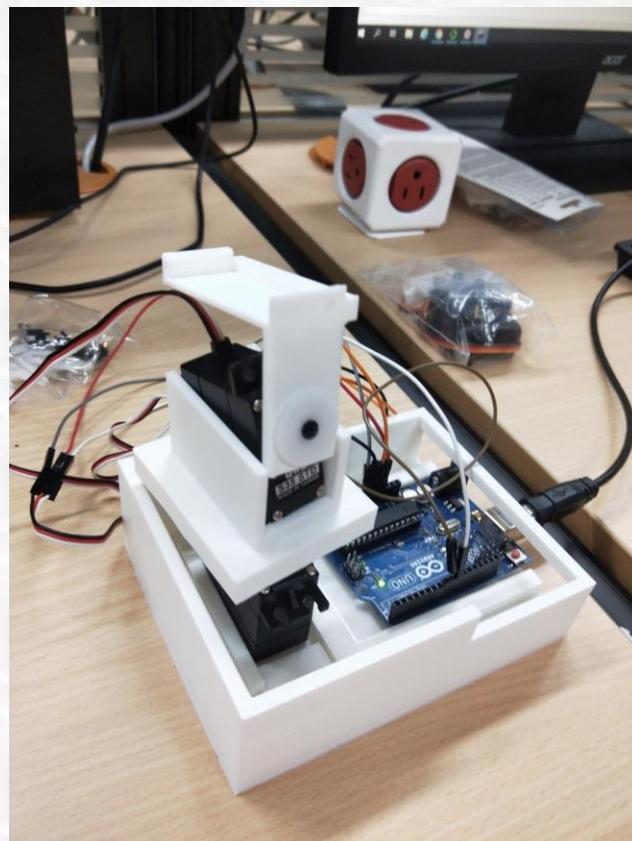
r = req.get(url)
r = req.get(url2)
r.text
```

遭遇困難、解決方法

- 雲台
- 雲台控制
- 人臉辨識

雲台

- ▶ 前一代：
 - 馬達會左右搖晃
 - 上方的馬達是斜的
 - 沒有蓋子



雲台

- ▶ 新一代：
加蓋子
馬達不會搖晃



雲台控制

- 之前用到360度的伺服馬達
- 無法控制角度

伺服馬達類型	180度伺服馬達	360度伺服馬達
控制方式	PWM控制旋轉角度	PWM控制旋轉速度和旋轉方向
是否可控制角度	可	不可
是否可連續旋轉	不可	可
主要應用	需控制角度時，如機械臂、雲台控制	需連續旋轉時，如驅動輪子

人臉辨識

➤ 結合

➤ 辨識率

結合

➤問題：

原以平行的方式結合馬達程式與人臉辨識程式，但程式複雜執行慢，且相機一直在轉動辨識率低。

➤解決：

後改為每轉20度才辨識一次，辨識5秒後才使馬達轉動。

辨識率

➤問題：

受到環境的影響太大，例如:光線亮度、背景、角度。

➤解決：

更多的人臉數，以及在不同環境捕捉人臉，辨識率就穩定了許多。

成果圖



成果影片

<https://youtu.be/BHHpiMh-RhM>



12未到
18到
19未到
21到
[12, 19]
[18, 21]

未來展望

希望在這之後可以把點名系統的規模擴大，延伸到整個班級，還可以偵測到每位同學的狀況，例如睡覺、玩手機等等，讓整個專題表現得更加人性化。



參考資料

- AI初見 從零開始的TensorFlow生活:

<https://ithelp.ithome.com.tw/users/20112114/ironman/3536>

- 資料分析&機器學習] 第5.1講: 卷積神經網絡介紹:

<https://medium.com/jameslearningnote/%E8%B3%87%E6%96%99%E5%88%86%E6%9E%90-%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-%E7%AC%AC5-1%E8%AC%9B-%E5%8D%B7%E7%A9%8D%E7%A5%9E%E7%B6%93%E7%B6%B2%E7%B5%A1%E4%BB%8B%E7%B4%B9-convolutional-neural-network-4f8249d65d4f>

➤ 機器學習ML NOTE]SGD, Momentum, AdaGrad, Adam Optimizer

<https://medium.com/%E9%9B%9E%E9%9B%9E%E8%88%87%E5%85%94%E5%85%94%E7%9A%84%E5%B7%A5%E7%A8%8B%E4%B8%96%E7%95%8C/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92ml-note-sgd-momentum-adagrad-adam-optimizer-f20568c968db>

➤ 利用python、tensorflow、opencv實現人臉識別

<https://www.itread01.com/content/1546931718.html>

➤ Python與arduino溝通橋樑

<https://medium.com/jeasee%E9%9A%A8%E7%AD%86/python-%E5%8F%8A-firmata-c8d104c1cf00>

Q&A



先不要

謝謝聆聽 喵



@meme-33