



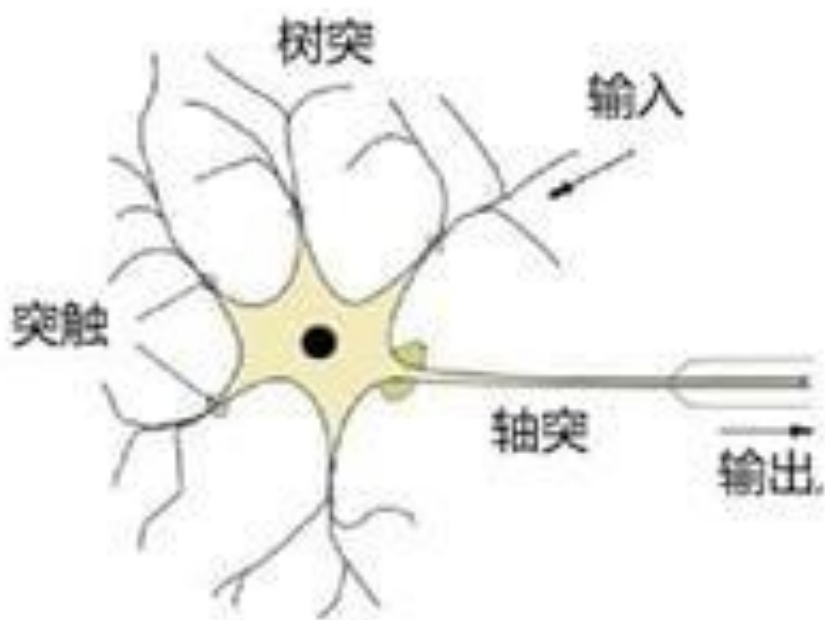
ST LECTUREs
小 孫 學 堂

Python

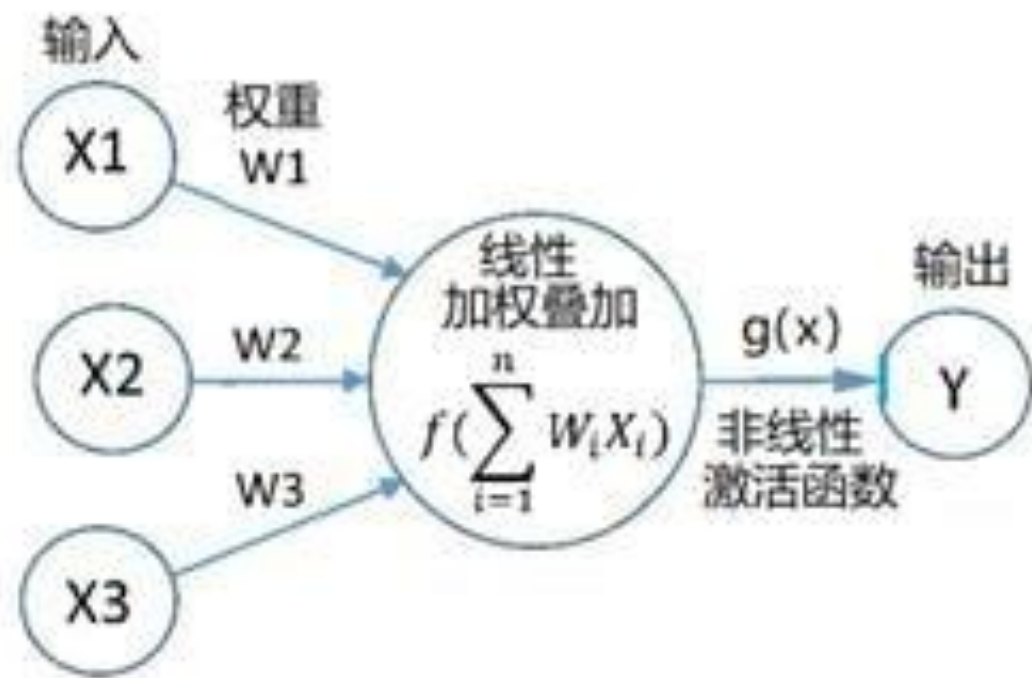
人臉偵測與人臉辨識

Created by 孫善堂 【小孫學堂】

類神經網路

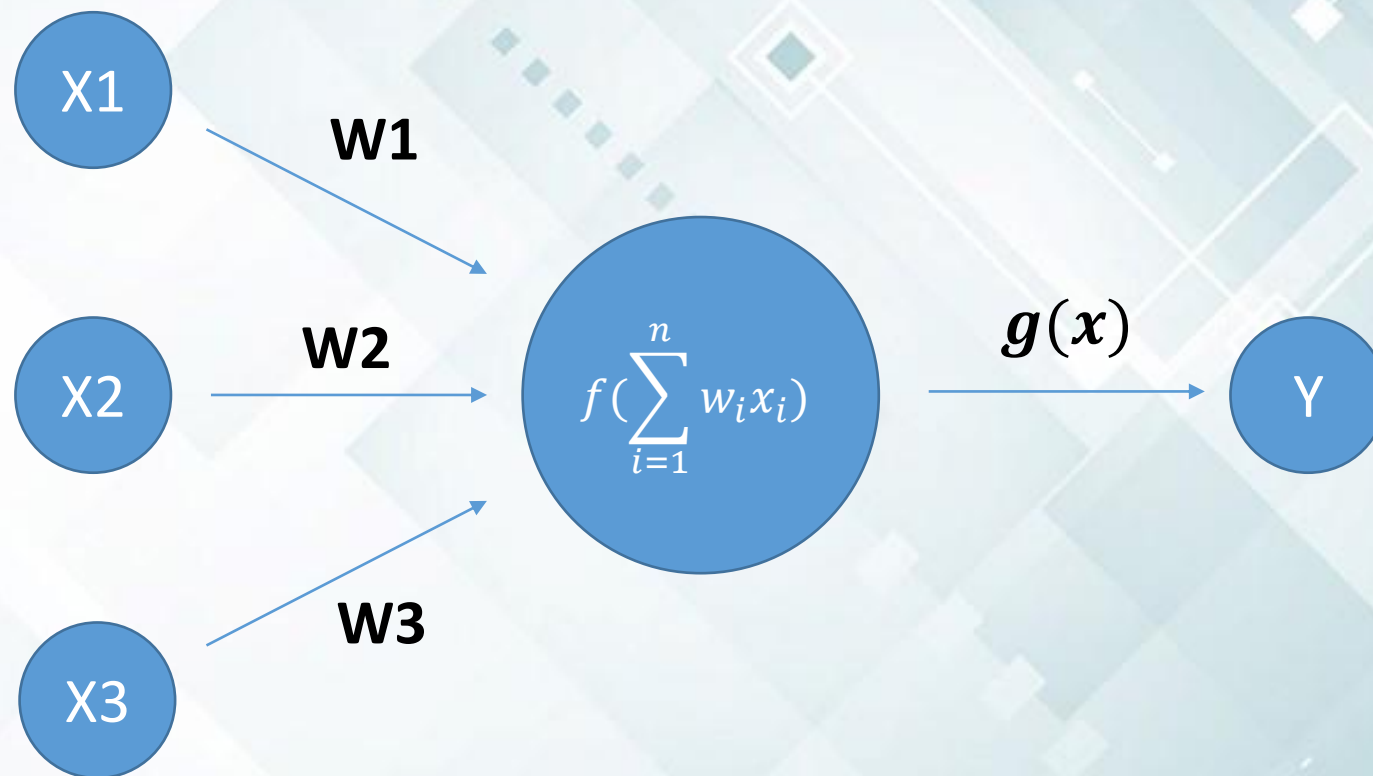


(a) 人脑中的神经元



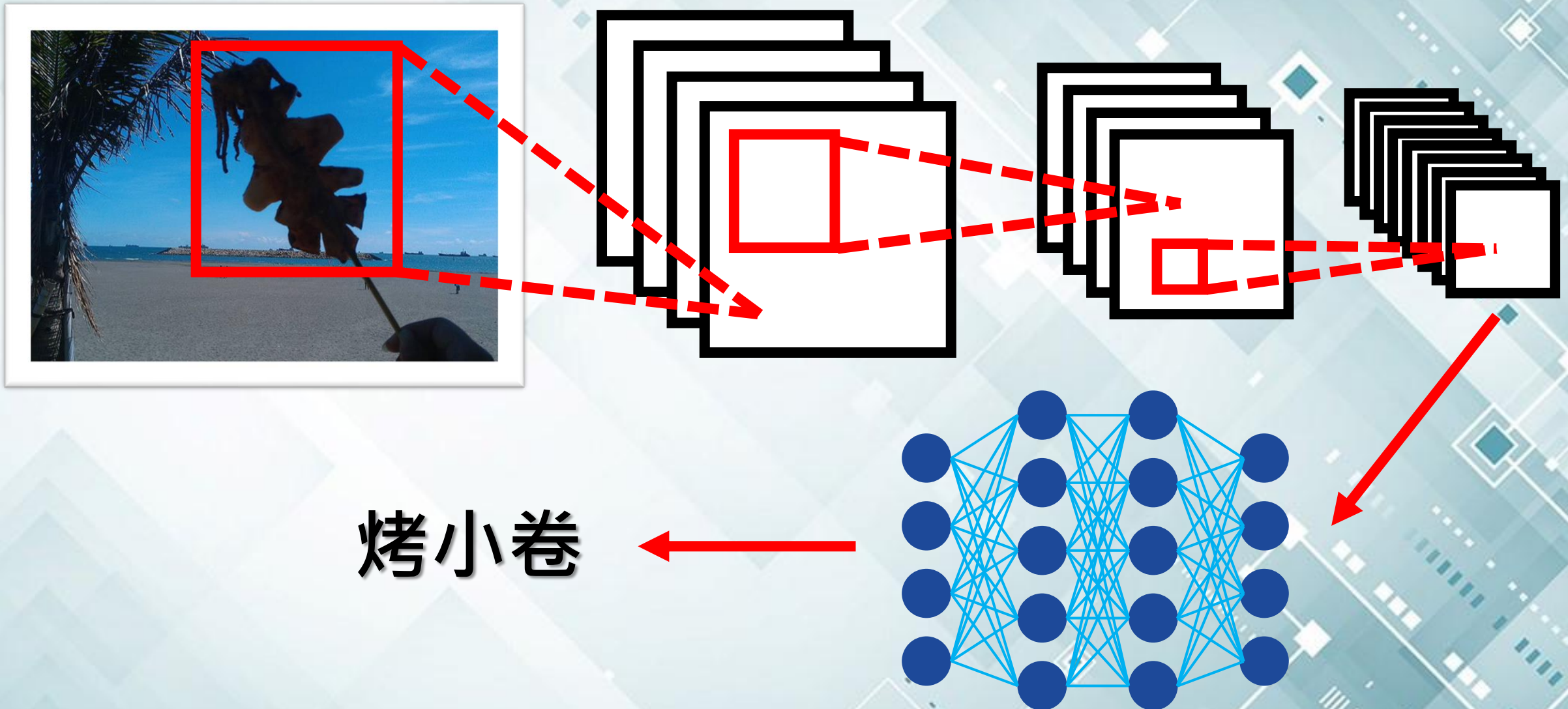
(b) 人工神经网络中的神经元

類神經網路



捲積神經網路

Convolutional neural network, CNN



人臉偵測步驟流程

人臉特徵
辨識模型

大量人臉
辨識訓練

提取人臉
共同特徵

利用特徵
偵測人臉

偵測結果
視覺呈現



運用openCV進行 人臉偵測實作

Created by 孫善堂 【小孫學堂】

基礎指令

安裝openCV:

Pip install numpy

Pip install opencv_python

樹梅派(linux)安裝openCV :

Sudo apt-get install python3-opencv

基礎指令

引用openCV:

```
import cv2
```

讀取圖片:

```
存取物件 = cv2.imread('檔名 or 路徑')
```

開啟預覽視窗及圖片物件:

```
cv2.imshow('視窗標題', 圖片存取物件)
```


基礎指令

等待按鍵(回傳按鍵ASCII 無則回傳-1):

`cv2.waitKey(等待時間)`

關閉指定視窗:

`cv2.destroyWindow(視窗名稱)`

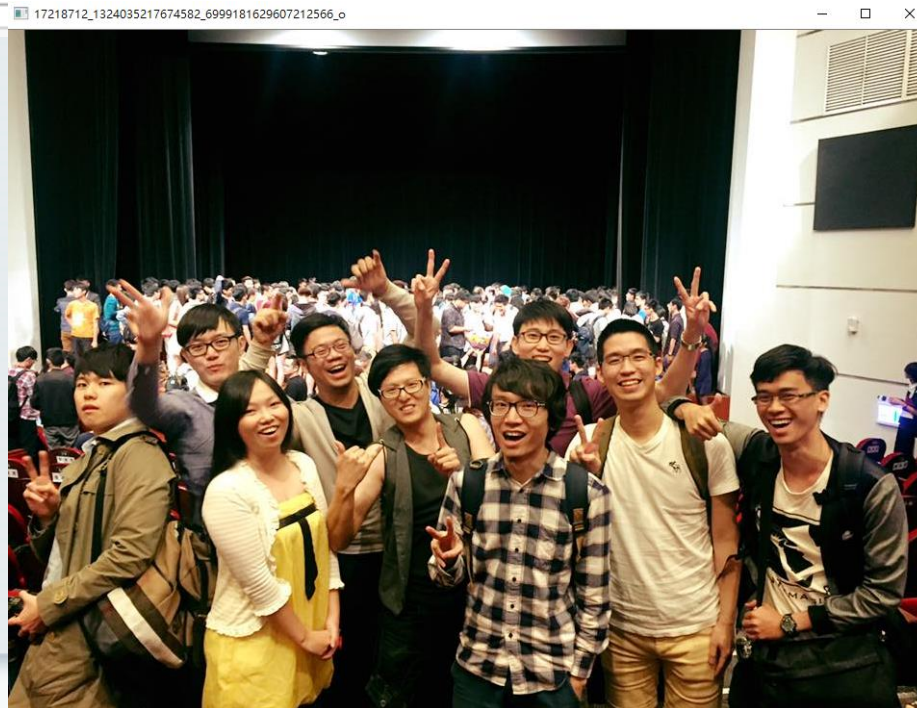
關閉所有視窗:

`cv2.destroyAllWindows()`

基礎指令

In [1]: `import cv2`

```
img = cv2.imread('13240769_1314796055202314_6119621336882432159_n.jpg')  
cv2.imshow('17218712_1324035217674582_6999181629607212566_o',img)
```



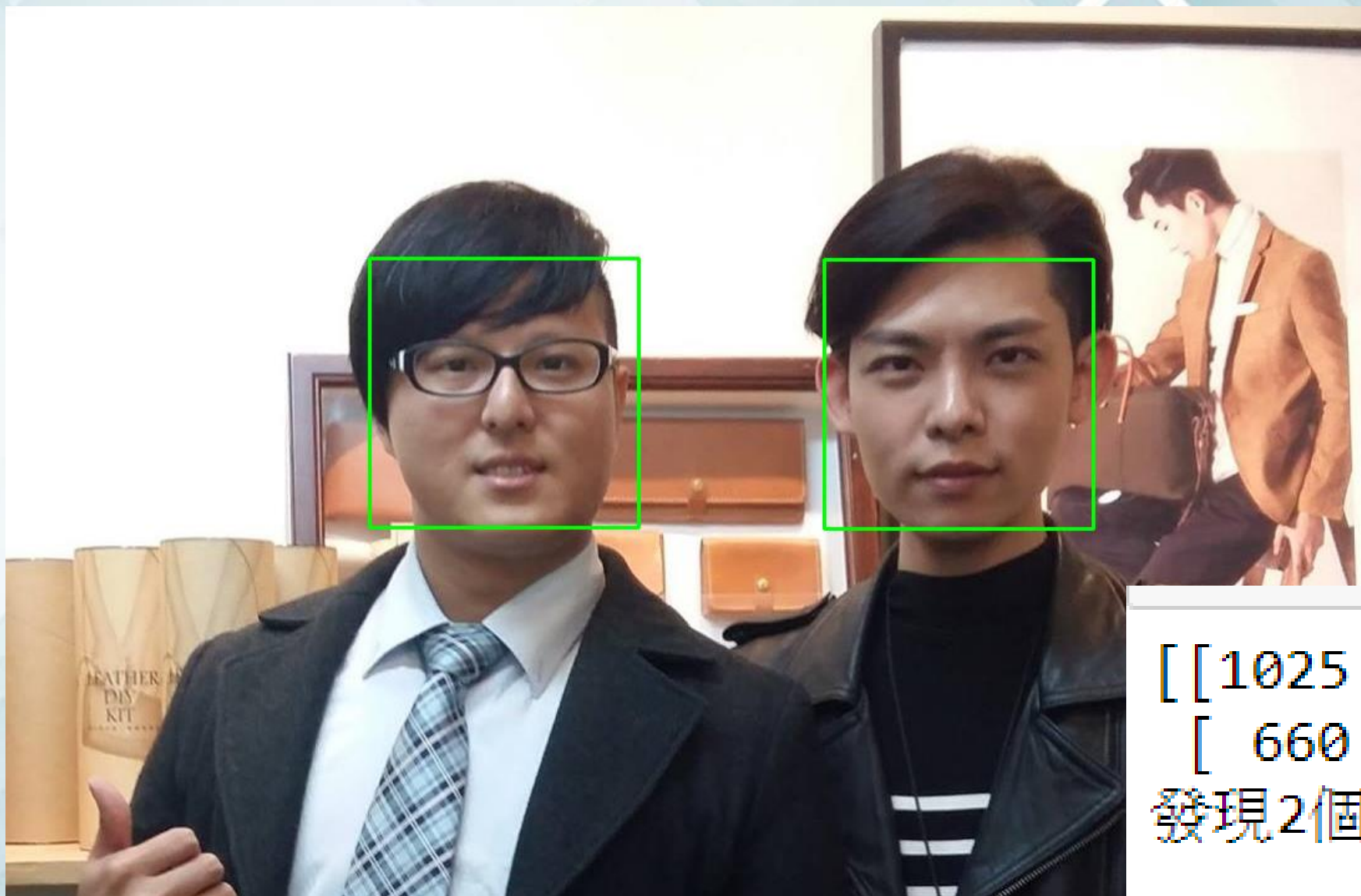
基礎指令

In [*]: `import cv2`

```
img = cv2.imread('13240769_1314796055202314_6119621336882432159_n.jpg')  
cv2.imshow('17218712_1324035217674582_6999181629607212566_o',img)
```

```
cv2.waitKey(0)  
cv2.destroyAllWindows()
```

人臉偵測



$\begin{bmatrix} 1025 & 222 & 216 & 216 \\ 660 & 221 & 216 & 216 \end{bmatrix}$

發現2個人臉!

基礎指令

將圖片物件轉換為灰階:

灰階物件 = `cv2.cvtColor(圖片物件,cv2.COLOR_BGR2GRAY)`

基礎指令

```
In [*]: import cv2
import numpy as np
img = cv2.imread('13240769_1314796055202314_6119621336882432159_n.jpg')
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

cv2.imshow('13240769_1314796055202314_6119621336882432159_n',gray)

cv2.waitKey(0)
cv2.destroyAllWindows()
```



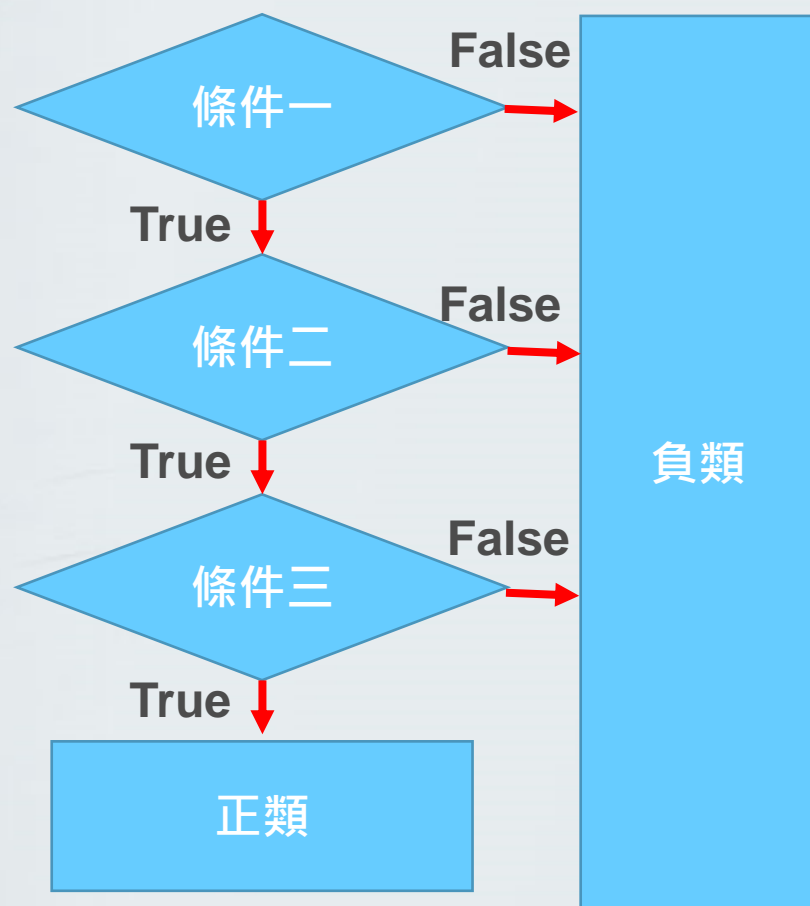
人臉偵測指令

設定串聯分類器種類:

分類器物件 = `cv2.CascadeClassifier(分類器參考文件)`

人臉正面分類參考：`haarcascade_frontalface_default.xml`

串聯分類器原理



分類器對多個影像特徵進行識別，當所有影像特徵皆符合條件則歸為正類，反之則歸為負類。

本次使用分類器將人臉正面歸為正類。

OpenCV附帶之三聯串分類器

1.Haar串聯分類器

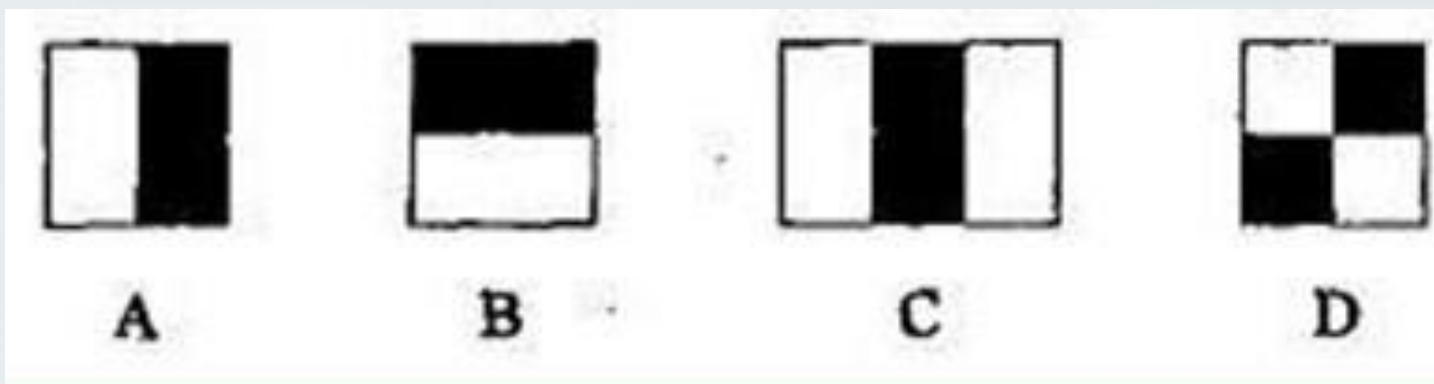
2.HOG串聯分類器

3.LBP串聯分類器

OpenCV附帶之三聯串分類器

1. Haar串聯分類器 <http://blog.csdn.net/zouxy09/article/details/7929570>

Haar特征分為三類：邊緣特征、線性特征、中心特征和對角線特征，組合成特征模板。矩形特征只對一些簡單的圖形結構，如邊緣、線段較敏感，所以只能描述特定走向（水平、垂直、對角）的結構。



OpenCV附帶之三聯串分類器

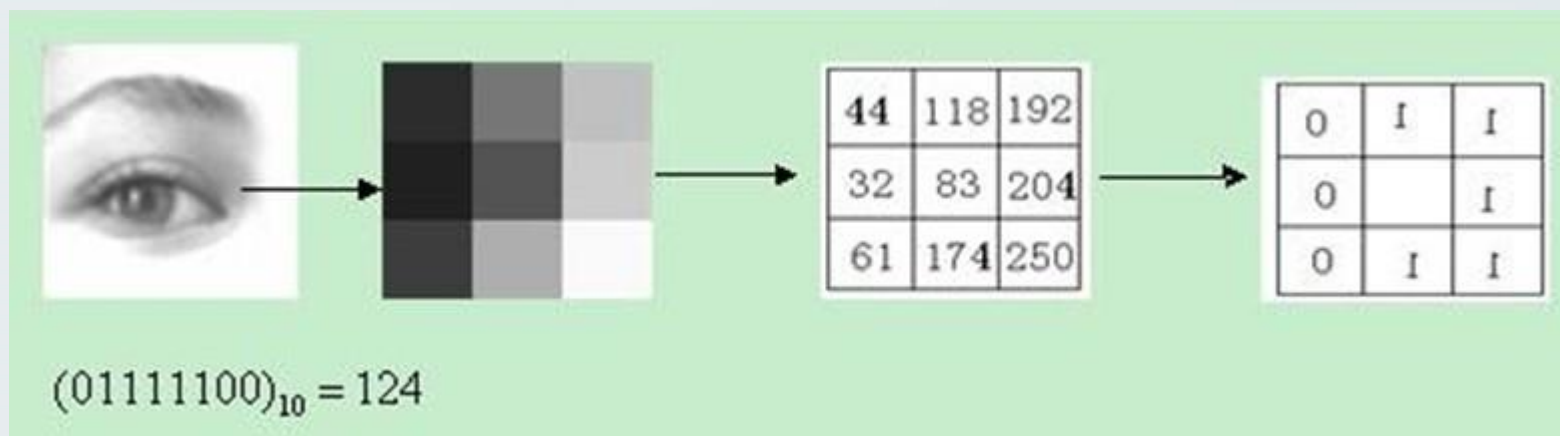
2.HOG串聯分類器 <http://blog.csdn.net/zouxy09/article/details/7929348>

方向梯度直方圖（Histogram of Oriented Gradient, HOG）特征是一種在計算機視覺和圖像處理中用來進行物體檢測的特征描述子。它通過計算和統計圖像局部區域的梯度方向直方圖來構成特征。Hog特征結合SVM分類器已經被廣泛應用於圖像識別中，尤其在行人檢測中獲得了極大的成功。

OpenCV附帶之三聯串分類器

3.LBP串聯分類器 <http://blog.csdn.net/zouxy09/article/details/7929531>

LBP (Local Binary Pattern , 局部二值模式) 是一种用来描述图像局部纹理特征的算子；它具有旋转不变性和灰度不变性等显著的优点。



常有Bug

找不到串聯分類器參考文件:

```
error: OpenCV(4.1.2) C:\projects\opencv-python\opencv\modules\objdetect\src\cas  
cadedetect.cpp:1689: error: (-215:Assertion failed) !empty() in function 'cv::C  
ascadeClassifier::detectMultiScale'
```

‘haarcascade_frontalface_default.xml’

改成

r'C:\Users\Json\Anaconda3\Lib\site-
packages\cv2\data\haarcascade_frontalface_default.xml'

人臉偵測指令

設定人臉偵測參數:

```
faces = faceCascade.detectMultiScale(  
    gray,  
    scaleFactor=1.12,  
    minNeighbors=3,  
    minSize=(5,5)  
)
```

人臉偵測指令

參數1：**image**--待檢測圖片，一般為灰度圖像加快檢測速度

參數2：**objects**--被檢測物體的矩形框向量組

參數3：**scaleFactor**--表示在前後兩次相繼的掃描中，搜索窗口的比例系數。默認為1.1即每次搜索窗口依次擴大10%

參數4：**minNeighbors**--表示構成檢測目標的相鄰矩形的最小個數(默認為3個)。

人臉偵測指令

畫製人臉方框:

`cv2.rectangle`(影像, 頂點座標, 對向頂點座標, 顏色, 線條寬度)

```
for(x,y,w,h) in faces:  
    cv2.rectangle(img,(x,y),(x+w,y+w),(0,255,0),2)
```

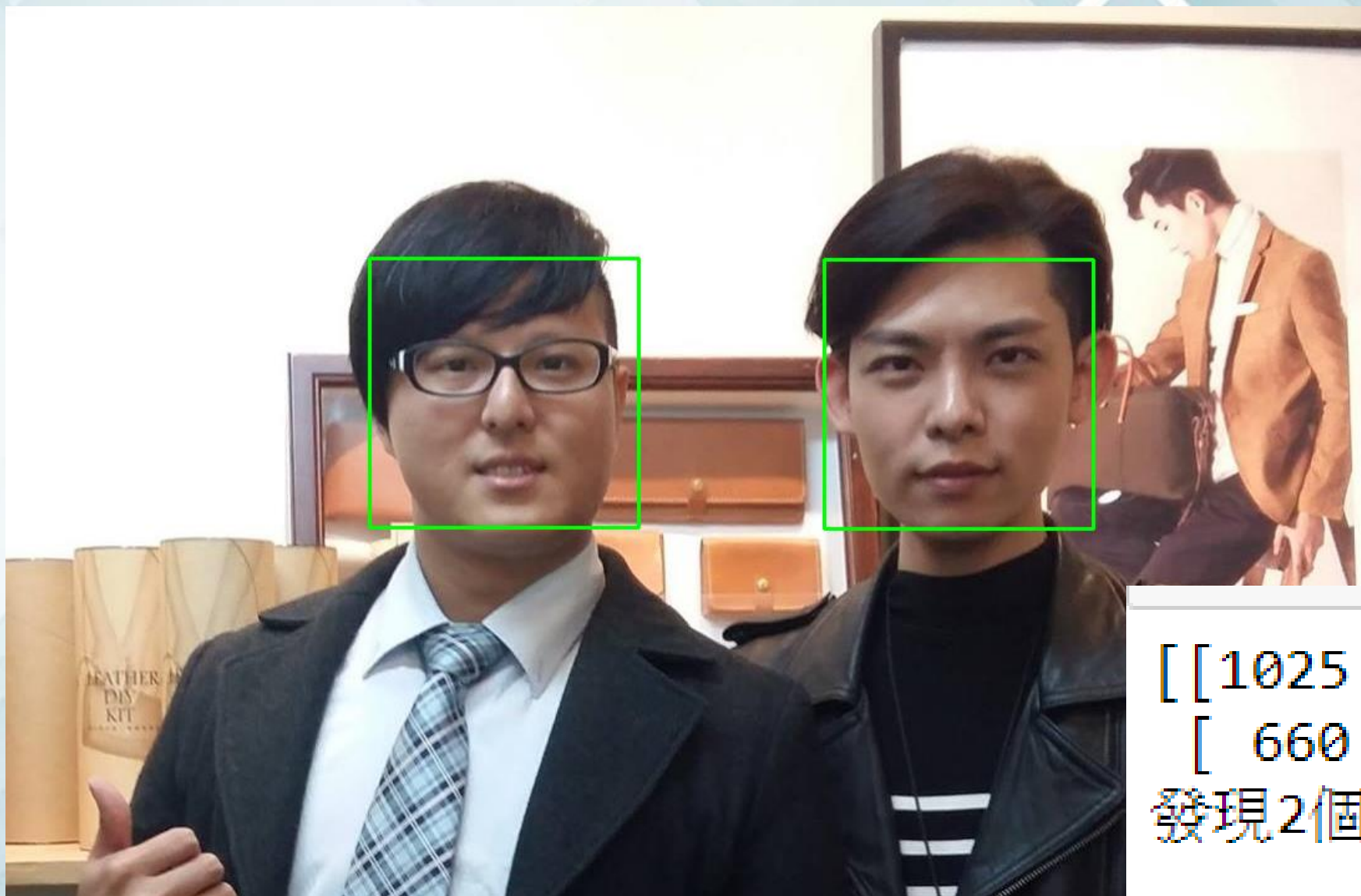

人臉偵測

```
In [2]: import cv2
import numpy as np
img = cv2.imread('17218712_1324035217674582_6999181629607212566_o.jpg')
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
faceCascade =cv2.CascadeClassifier(r'C:\Users\Json\Anaconda3\Lib\site-packages\cv2\data\haarcascade_frontalface_default.xml')

faces = faceCascade.detectMultiScale(
    gray,
    scaleFactor=1.12,
    minNeighbors=3,
    minSize=(5,5)
)
print(faces)
print('發現',len(faces),'個人臉!')
for(x,y,w,h) in faces:
    cv2.rectangle(img,(x,y),(x+w,y+w),(0,255,0),2)
cv2.imshow('13240769_1314796055202314_6119621336882432159_n',img)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

人臉偵測



$\begin{bmatrix} 1025 & 222 & 216 & 216 \\ 660 & 221 & 216 & 216 \end{bmatrix}$

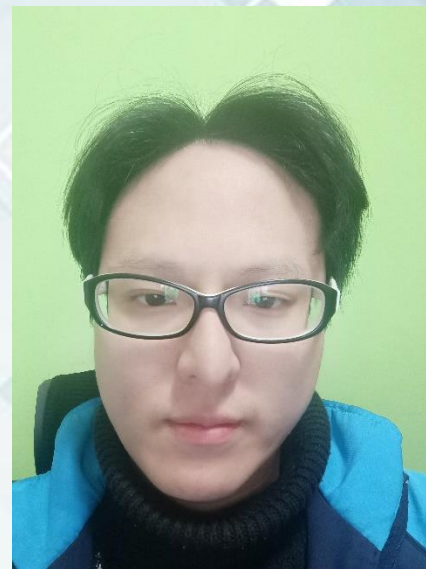
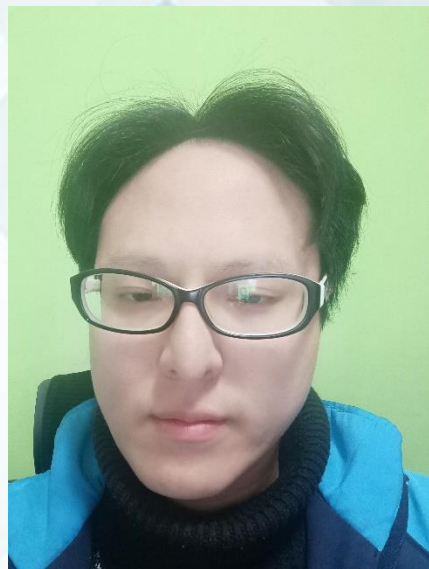
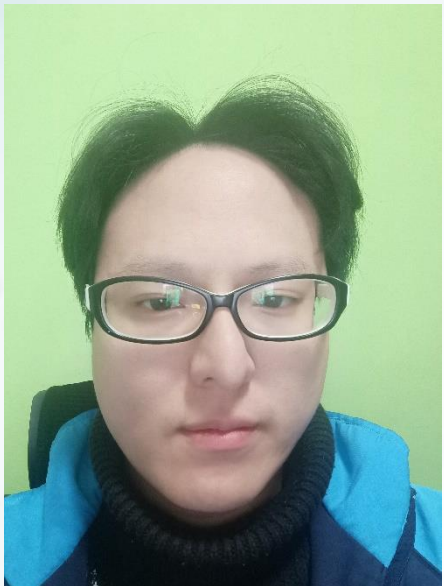
發現2個人臉！



運用openCV進行 人臉識別實作

Created by 孫善堂 【小孫學堂】

人臉辨識



label= 0
confidence= 1.8099379919169658

人臉辨識步驟流程

建立人臉
資料庫

建立訓練
辨識模型

針對照片
進行預測

預測結果
產出

人臉辨識指令

建置人臉識別器:

識別器物件 = cv2.face.LBPHFaceRecognizer_create()

訓練人臉識別器：

識別器物件.train(訓練材料，識別標籤)

```
recongner = cv2.face.LBPHFaceRecognizer_create()  
recongner.train(img,np.array(labels))
```

Numpy array vs list

In [46]:

```
1 a=[2,3,4]
2 b=np.array(a)
3
4 print(type(a))
5 print(type(b))
6 print()
7 print(a)
8 print(b)
9 print()
10 print(a*3)
11 print(b*3)
```

```
<class 'list'>
<class 'numpy.ndarray'>
```

```
[2, 3, 4]
[2 3 4]
```

```
[2, 3, 4, 2, 3, 4, 2, 3, 4]
[ 6  9 12]
```

人臉辨識指令

設定識別回傳值:

```
label,confidence=recongner.predict(src)
```

Label : 傳回的識別結果標籤

Confidence : 傳回的可靠度評分(0表完全符合)

Src : 需要識別的人臉影像

人臉辨識

```
In [6]: import cv2
import numpy as np
img = []
img.append(cv2.imread('c1.jpg',cv2.IMREAD_GRAYSCALE))
img.append(cv2.imread('c2.jpg',cv2.IMREAD_GRAYSCALE))
img.append(cv2.imread('c3.jpg',cv2.IMREAD_GRAYSCALE))
img.append(cv2.imread('d1.jpg',cv2.IMREAD_GRAYSCALE))
img.append(cv2.imread('d2.jpg',cv2.IMREAD_GRAYSCALE))
img.append(cv2.imread('d3.jpg',cv2.IMREAD_GRAYSCALE))

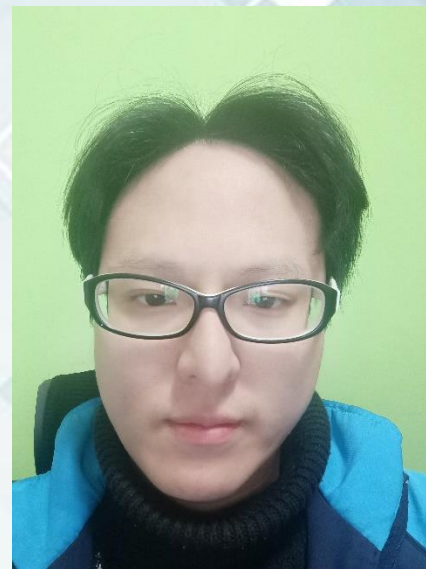
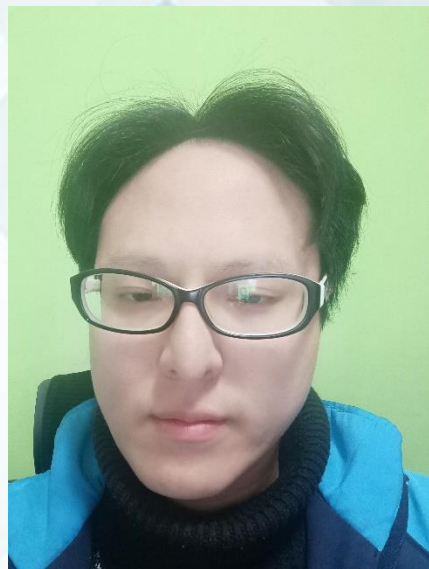
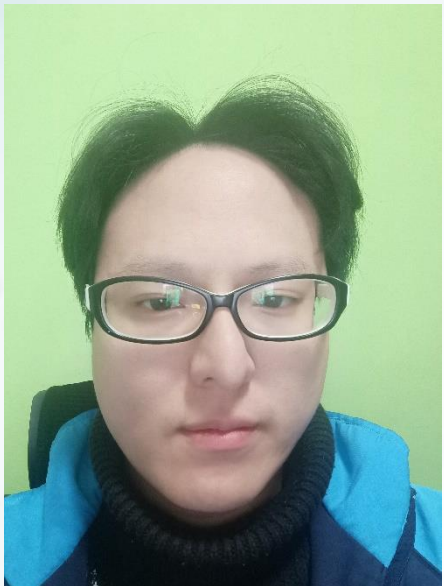
labels=[0,0,0,1,1,1]

recongizer = cv2.face.LBPHFaceRecognizer_create()
recongizer.train(img,np.array(labels))

predict_image=cv2.imread('a1.jpg',cv2.IMREAD_GRAYSCALE)

label,confidence=recongizer.predict(predict_image)
print('label=',label)
print('confidence=',confidence)
```


人臉辨識



label= 0
confidence= 1.8099379919169658


```
1 import cv2
2 import numpy as np
3
4 img=[]
5 labels=[]
6
7 num=0
8 while num<9:
9     num+=1
10    img.append(cv2.imread('L'+str(num)+'.jpg',cv2.IMREAD_GRAYSCALE))
11    labels.append(0)
12
13 num=0
14 while num<9:
15     num+=1
16    img.append(cv2.imread('CH'+str(num)+'.jpg',cv2.IMREAD_GRAYSCALE))
17    labels.append(1)
18
19 print(len(img))
20 print(labels)
```

18

[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1]

```
1 recon=cv2.face.LBPHFaceRecognizer_create()  
2 recon.train(img,np.array(labels))
```

```
1 predict_img=cv2.imread('CH10.jpg',cv2.IMREAD_GRAYSCALE)  
2  
3 label,confidence=recon.predict(predict_img)  
4 print('label=',label)  
5 print('confidence=',confidence)
```

label= 1
confidence= 70.43505983228134

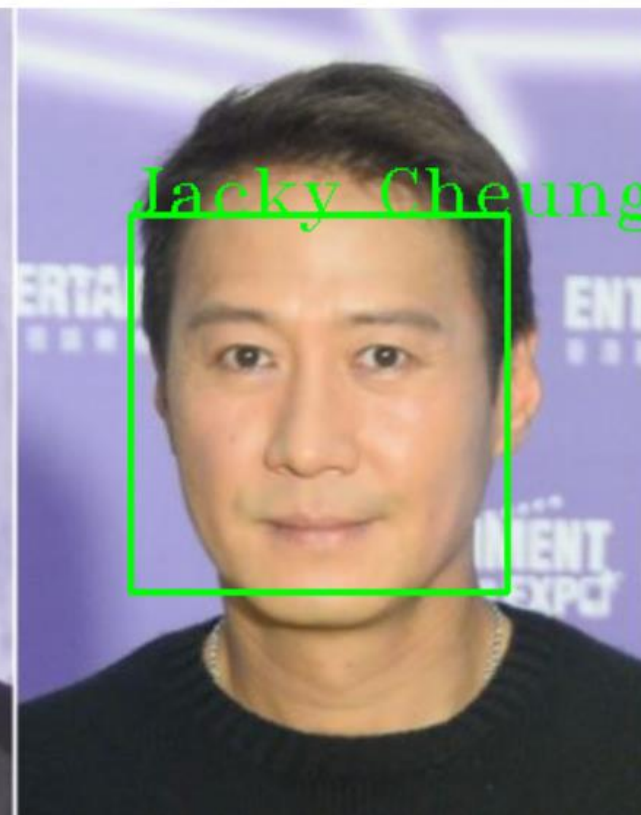
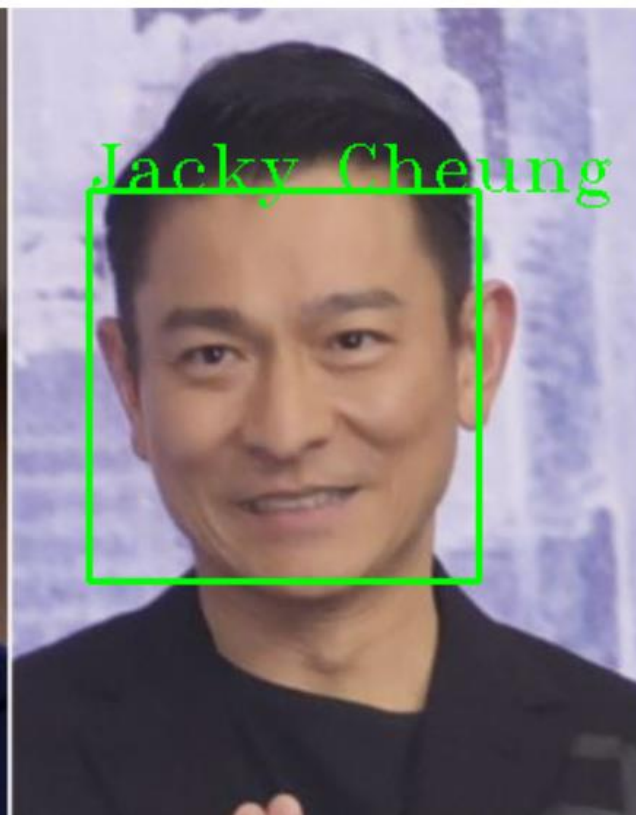
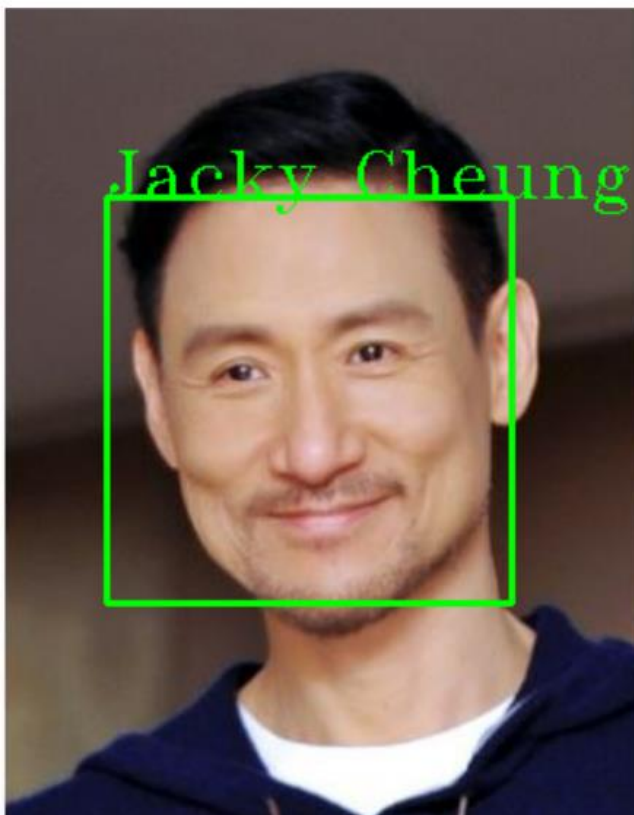
```
1 name_list=['劉德華','周杰倫']  
2  
3 print('此人為',name_list[label])  
4 print('confidence=',confidence)
```

此人為 周杰倫
confidence= 70.43505983228134



運用openCV進行 合照人臉識別

Created by 孫善堂 【小孫學堂】



圖片切割與儲存

```
# 裁切區域的 x 與 y 座標 ( 左上角 )
```

```
x = 100
```

```
y = 100
```

```
# 裁切區域的長度與寬度
```

```
w = 250
```

```
h = 150
```

```
# 裁切圖片
```

```
crop_img = img[y:y+h, x:x+w]
```

```
cv2.imwrite('crop.jpg', crop_img)
```


圖片置入文字

`cv2.putText(影像, 文字, 座標, 字型, 大小, 顏色, 線條寬度, 線條種類)`

這裡的座標是指文字的左下角座標點，而字型可使用 OpenCV 內建的幾種字型（請看以下範例），大小則是代表字型的縮放比例（例如 1.5 就代表放大 1.5 倍）。

最後一項線條種類參數可以指定如何繪製線條，若想要有反鋸齒（anti-aliasing）的效果，可設定為 `cv2.LINE_AA`。這項參數也可以省略不寫，若省略的話預設值為 `cv2.LINE_8`。（可用字體：`cv2.FONT_HERSHEY_TRIPLEX`）

合照多人人臉辨識步驟流程

建立人臉
資料庫

建立訓練
辨識模型

合照偵測
圖片切割

針對圖片
個別預測

預測結果
原圖呈現

建立人臉識別資料庫與模型

```
1 import cv2
2 import numpy as np
3
4 kings=['Andy Lau','Aaron Kwok','Leon Lai','Jacky Cheung']
5 kings_en=['L','Q','M','Z']
6 img_train=[]
7 labels=[]
8
9 king_num=0
10 for king in kings_en:
11
12     num=0
13     while num<10:
14         num+=1
15         img_train.append(cv2.imread(king+str(num)+'.JPG',cv2.IMREAD_GRAYSCALE))
16         labels.append(king_num)
17     king_num+=1
18
```

```
1 rec=cv2.face.LBPHFaceRecognizer_create()
2 rec.train(img_train,np.array(labels))
```

人臉偵測並切割成辨識用圖片

```
1 img = cv2.imread('4kings.jpg')
2 gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
3 faceCascade =cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
4
5 faces = faceCascade.detectMultiScale(
6     gray,
7     scaleFactor=1.12,
8     minNeighbors=3,
9     minSize=(50,50)
10 )
11 print(faces)
12 print('發現',len(faces),'個人臉!')
13
14 cut_list=[]
15 face_num=0
16 for(x,y,w,h) in faces:
17     face_num+=1
18     cut_img=img[y:y+h,x:x+w]
19     cv2.imwrite('cut'+str(face_num)+'.jpg',cut_img)
20     print('已儲存','cut'+str(face_num)+'.jpg')
21     cut_list.append('cut'+str(face_num)+'.jpg')
22 print(cut_list)
```


將圖片個別辨識後結果印在原圖上

```
1 cut_who=[]
2 for cut in cut_list:
3     predict_img=cv2.imread(cut,cv2.IMREAD_GRAYSCALE)
4
5     label,confidence=rec.predict(predict_img)
6     cut_who.append(kings[label])
7
8     print('此人是',kings[label],'confidence=',confidence)
```

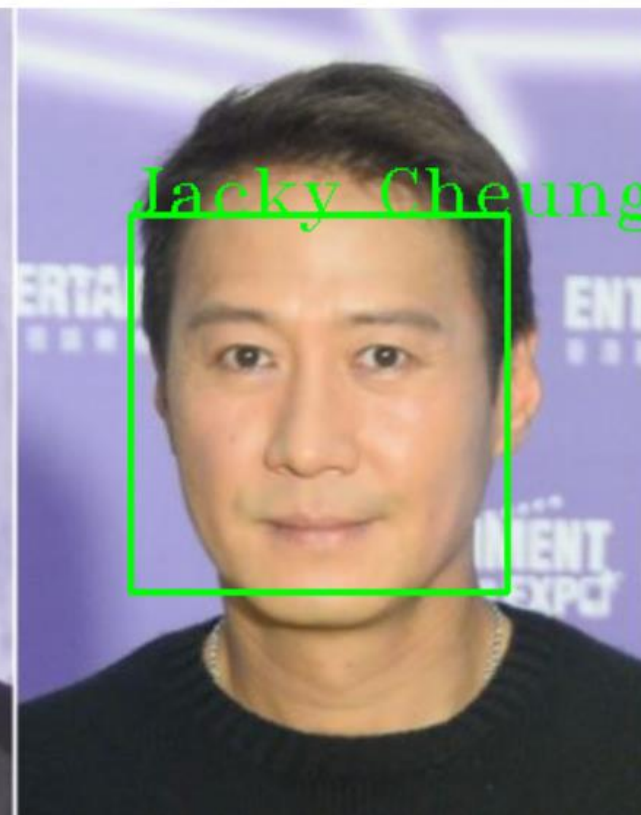
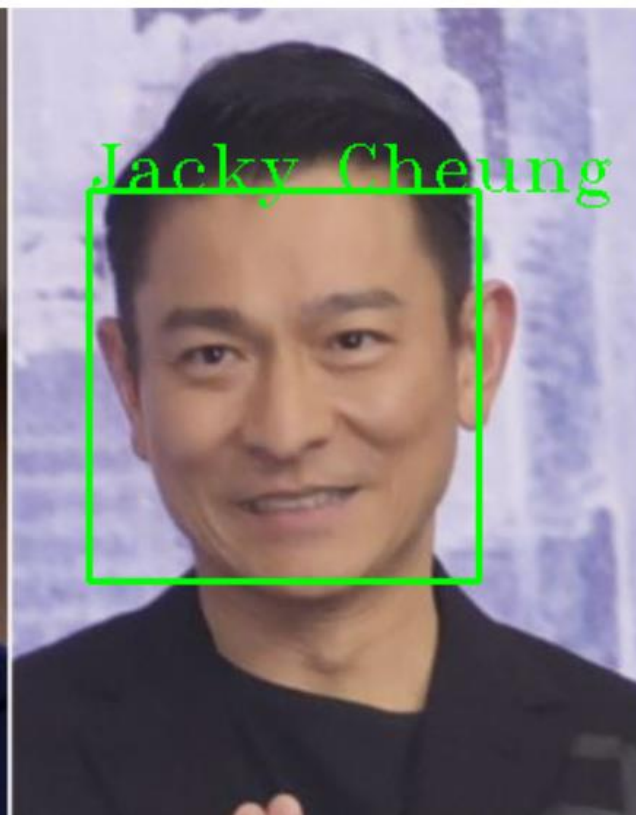
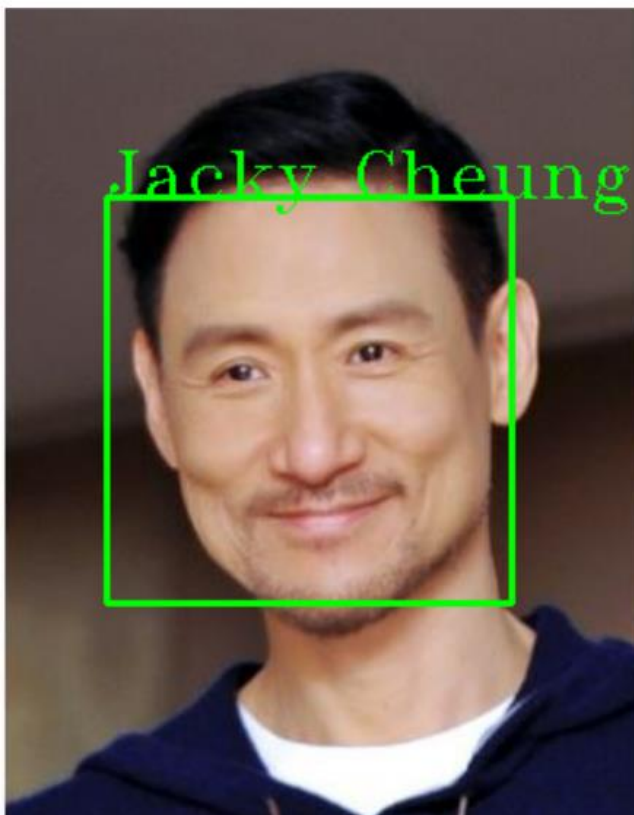
此人是 Jacky Cheung confidence= 61.999293296288634

此人是 Jacky Cheung confidence= 65.64424283368207

此人是 Jacky Cheung confidence= 70.4447070565961

此人是 Aaron Kwok confidence= 62.119314443013266

```
1 who_num=0
2 for(x,y,w,h) in faces:
3     cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)
4     cv2.putText(img,cut_who[who_num],(x,y),cv2.FONT_HERSHEY_TRIPLEX,1,(0,255,0),1)
5     who_num+=1
6
7 cv2.imshow('findface',img)
8 cv2.waitKey()
```

Thank you for your attention

