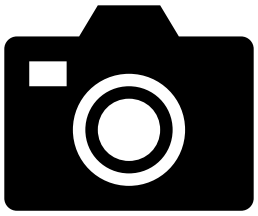
Chapter 7：實務 x 人臉辨識

本章重點

本章介紹：由淺入深、一層一層的堆疊技巧，了解後並完成簡單的人臉辨識與物聯網應用。技術是為商業活動服務的，沒有人用的技術是沒有多大幫助的；當你在學習基礎技術時，看似無用，等到養成一定的技術水平後，才有機會了解更多有趣的應用。

準備材料

可上網的電腦



電腦鏡頭

學習目標

1. 安裝 OpenCV 的模組。
2. OpenCV 的基本操作
3. OpenCV 的人臉辨識。

7-1. 安裝 OpenCV

7-1-1. 在 Windows 安裝 OpenCV

OpenCV (Open Source Computer Vision Library) 是跨平台 BSD 授權的一套著名的電腦視覺函式庫，這是 Intel 發起並參與開發，幫助開發圖片處理、影片處理、電腦視覺的人臉辨識和物體辨識等人工智慧的相關應用。

- OpenCV 可以在 Android、iOS 上開發，支援的程式語言也有 C/C++、Java、Python，重點是免費的。

基本上，安裝 OpenCV 有兩種方式，如下所示：

- 使用 pip 指令：使用 Python 套件管理 pip 安裝 OpenCV，安裝過程比較簡單，但是不會優化 OpenCV 的執行效能。(本次採此種安裝方式)
- 自行編譯 OpenCV 程式碼進行安裝：如果有特殊 OpenCV 版本和優化需求，我們需要自行編譯 OpenCV 程式碼。

以 pip 安裝 OpenCV 套件

找到你應用程式區，開啟你的 Anacodna Prompt 視窗：

```
>>> pip install 模組名稱
```

以最新版的 python 3.9.7 為例：

Step 1 安裝 OpenCV 的相關支援模組

安裝支援的 Python 模組，numpy 為陣列容器、matplotlib 為 Python 及其數值計算庫 numpy 的繪圖庫、imutils 可以讓我們更容易使用 OpenCV 圖片處理：

```
>>> pip install numpy     # 陣列容器，用於影像數值計算
>>> pip install matplotlib     # 畫圖用
>>> pip install imutils     # 常用的圖片處理函式
```

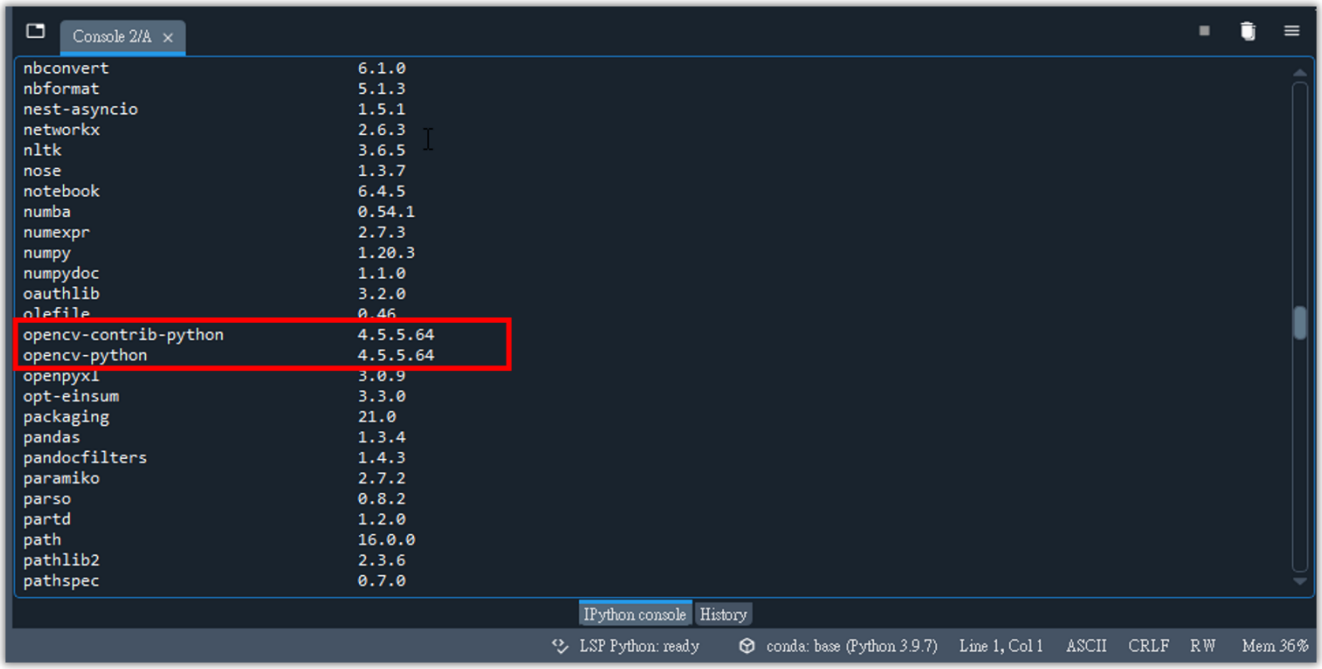
Step 2　安裝 OpenCV

使用 pip 安裝 OpenCV：

```
>>> pip install opencv-python
>>> pip install opencv-contrib-python
```

輸入指令來檢視目前 Python 安裝的套件清單：

```
>>> pip list
```



Step 3　檢查 OpenCV 是否成功安裝

檢查 Python 是否安裝好 OpenCV。

```
>>> import cv2
>>> cv2.__version__    # 顯示目前 OpenCV 的版本
'4.5.4'
```

7-2. OpenCV 基本操作

7-2-1. OpenCV 圖片處理

讀取與顯示圖片

Python 程式是呼叫 OpenCV 的 imread() 方法讀取圖片，和 imshow() 方法顯示圖片。例如：

7-1 讀取與顯示圖片。(7-1.py)

```
import cv2

img = cv2.imread('capybara.jpg')
cv2.imshow('Capybara', img)
gray_img = cv2.imread('capybara.jpg', cv2.IMREAD_GRAYSCALE)
cv2.imshow('Capybara:gray', gray_img)

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

cv2.IMREAD_GRAYSCALE 為物件.屬性。不是方法，當宣告一個物件實體後，會有相關的物件靜態或特徵描述，稱為屬性。這樣可以簡化物件的使用，物件的詳細說明請參考物件導向程式的類別設計。更多的屬性設定：

- cv2.IMREAD_COLOR：彩色，預設值。
- cv2.IMREAD_UNCHANGED：沒有改變，包含透明度的圖片內容。
- cv2.IMREAD_GRAYSCALE：灰階。

取得圖片資訊

我們可以使用 shape 屬性取得圖片資訊的尺寸和色彩數。例如：分別讀取成彩色和灰階圖片後，顯示圖片資訊。例如：

7-1a 取得圖片資訊。

```
import cv2

img = cv2.imread('capybara.jpg')
img2 = cv2.imread('capybara.jpg', cv2.IMREAD_GRAYSCALE)
print(img.shape)
print(img2.shape)
h, w, c = img.shape
print('圖片高: ', h)
print('圖片寬: ', w)
```

調整圖片尺寸

OpenCV 的 cv2.resize() 在調整圖片尺寸時，會改變圖片的長寬比例，我們準備改用 imutils 模組的方法來調整圖片尺寸。然後呼叫 imutils.resize() 方法調整圖片尺寸。例如：

7-1b 調整圖片尺寸。

```
import cv2, imutils

img = cv2.imread('capybara.jpg')
print(img.shape)
resized_img = imutils.resize(img, width = 300)
print(resized_img.shape)
cv2.imshow('Capybara', img)
cv2.imshow('Capybara:resized', resized_img)

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

儲存影像的容器：numpy.array

Numpy：Python 的擴充模組，常用於機器學習的多維度資料處理。

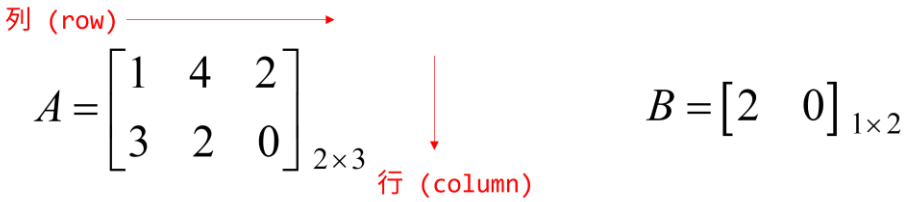
例如：建立一維陣列與索引。

```
>>> import numpy as np    # 匯入 numpy
>>> a = np.array([10, 2, 45, 32, 24])    # 建立五個元素的一維陣列
>>> len(a)    # 計算元素個數
>>> a[2:4]    # 取出第 2、3 個元素
>>> a[:4]     # 取出第 1 ~ 3 個元素
```

例如：建立二維陣列與索引。

```
>>> import numpy as np
>>> A = np.array([[1, 4, 2], [3, 2, 0]])    # 建立二維矩陣
>>> B = A[1:, 1:3]     # 二維切片，第一個索引以列切片，第二個索引以行為切片
```

示意圖如下：



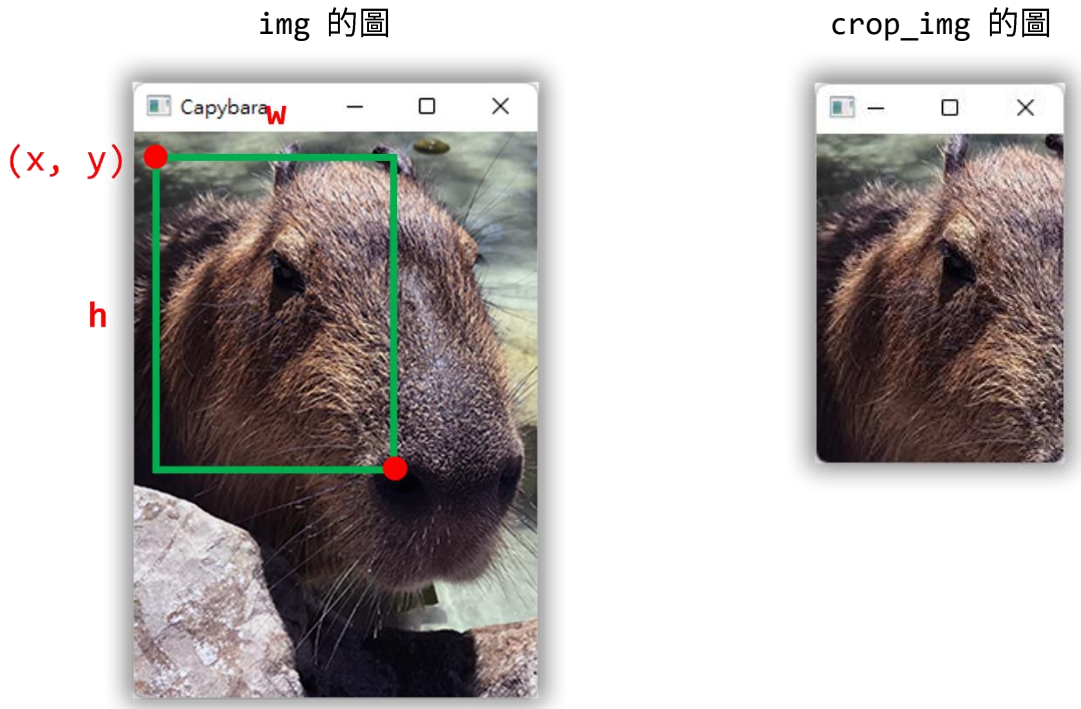
剪裁圖片

當了解多維矩陣的概念後，在 OpenCV 使用 imread() 方法讀取的圖片內容是一個 Numpy 容器，剪裁圖片就是在切割 Numpy 矩陣。例如：

7-1c 剪裁圖片。

```
import cv2

img = cv2.imread('capybara.jpg')
print(img.shape)
x = 10; y = 10
w = 150; h= 200
crop_img = img[y:y + h, x:x + w]
cv2.imshow('capybara', img)
cv2.imshow('corp_Capybara', crop_img)
print(crop_img.shape)

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

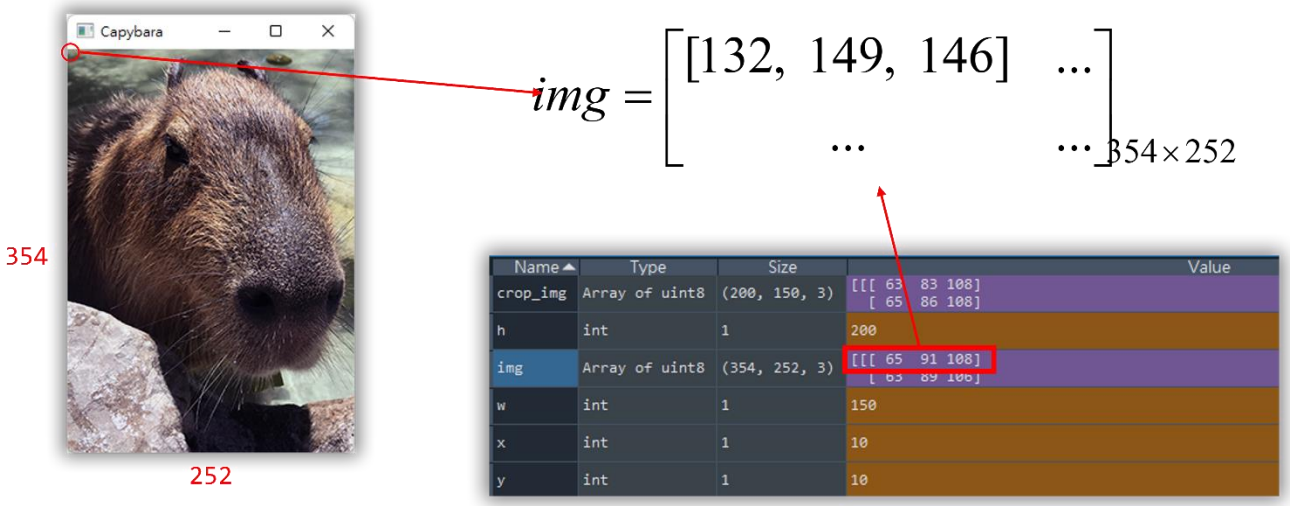
示意圖如下：



深入了解儲存全彩影像的矩陣

電腦以 RGB (紅綠藍) 三原色作為色彩顯示，每個原色可以設定 0 ~ 255。
例如：(R, G, B) = (0, 0, 0) 代表黑色；(R, G, B) = (255, 255, 255) 代表白色。
每一個像素可以用 RGB 表示來 256 x 256 x 256 = 16,777,216，這麼多的色彩數。

以三維矩陣表示一張 RGB 全彩圖片，示意圖如下：



旋轉、翻轉和位移圖片

OpenCV 沒有旋轉和位移圖片的方法 (只有 cv2.flip() 方法來翻轉圖片)，需自行運算來旋轉和位移圖片，在 imutils 提供有相關方法來旋轉和位移圖片。例如：

7-1d 旋轉、翻轉和位移圖片。

```
import cv2, imutils

img = cv2.imread('capybara.jpg')
rotated_img = imutils.rotate(img, angle = 90)
fliped_img = cv2.flip(img, 1)
translated_img = imutils.translate(img, 25, -75)
cv2.imshow('Capybara', img)
cv2.imshow('Capybara:rotated', rotated_img)
cv2.imshow('Capybara:fliped', fliped_img)
cv2.imshow('Capybara:translated', translated_img)

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

轉換成灰階和 BGR 圖片

OpenCV 讀取圖片後，可以呼叫 cvtColor() 方法轉換彩色圖片成為灰階圖片，方法的第 2 個參數是 cv2.COLOR_BGR2GRAY。

- 圖片是 RGB 格式，可以使用參數 cv2.COLOR_RGB2BGR，將 RGB 轉換成 BGR (同理，如果需要 RGB 格式，可以使用參數 cv2.COLOR_BGR2RGB 將 BGR 轉換成 RGB)。例如：

7-1e 旋轉、翻轉和位移圖片。

```
import cv2

img = cv2.imread('capybara.jpg')
gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
bgr_img = cv2.cvtColor(img, cv2.COLOR_RGB2BGR)
cv2.imshow('Capybara:gray', gray_img)
cv2.imshow('Capybara:bgr', bgr_img)

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

注意：imread() 與 imshow() 方法都是使用 BGR 格式，只需要 cv2.COLOR_BGR2GRAY 轉成灰階。

轉換成灰階和 BGR 圖片

在 imutils 提供 url_to_image() 方法，可以讓我們直接從網路讀取圖檔內容。例如：

7-1f 從 URL 取得圖片。

```
import cv2, imutils

url = 'https://fchart.github.io/img/koala.png'
img = imutils.url_to_image(url)
cv2.imshow('Koala', img)

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

註記圖片

註記圖片就是在圖片上繪圖，我們可以在圖片上畫直線、畫長方形、畫圓形、畫橢圓形和加上文字內容。例如：

7-1g 註記圖片。

```
import cv2

img = cv2.imread('capybara.jpg')
cv2.line(img, (0, 0), (200, 200), (0, 0, 255), 5)
cv2.rectangle(img, (20, 70), (120, 160), (0, 255, 0), 2)
cv2.rectangle(img, (40, 80), (100, 140), (255, 0, 0), -1)
cv2.circle(img, (90, 210), 30, (0, 255, 255), 3)
cv2.circle(img, (140, 170), 15, (255, 0, 0), -1)
cv2.putText(img, 'OpenCV', (10, 40), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,
              255, 255), 5, cv2.LINE_AA)
cv2.imshow('Capybara', img)

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

寫入圖片

OpenCV 可以呼叫 imwrite() 方法將圖片內容寫入圖檔。例如：

7-1h 寫入圖片。

```
import cv2

img = cv2.imread('capybara.jpg')
gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imwrite('result_gray.jpg', gray_img)
cv2.imwrite('result.png', img)
```

7-2-2. OpenCV 影片處理

影片事實上就是一種動態影像，這是一連串連續的靜態影像所組成，每一個靜態影像稱為『影格』(Frame，或稱幀)，每秒播放的靜態影像數稱為『影格率』(Frame per Second，或稱幀率)。

OpenCV 支援讀取和播放影片檔案，我們可以取得影片資訊和播放出灰階黑白內容的影片。

播放影片檔

Python 程式是建立 OpenCV 的 VideoCapture 物件來播放影片檔。例如：

7-2 播放影片檔。

```
import cv2

cap = cv2.VideoCapture('YouTube.mp4')

while cap.isOpened():
    ret, frame = cap.read()
    if ret:
        cv2.imshow('frame', frame)
    if cv2.waitKey(1) == 27:
        break

cap.release()
cv2.destroyAllWindows()
```

注意：有些方法會回傳多個參數，必須指派相同數量的變數來接收。例如：cap.read()。

寫程式來顯示灰階影片

我們只需使用 8-1e.py 的方法來處理圖片，就可以播放出灰階的黑白影片。例如：

7-2a 寫程式來顯示灰階影片。

```
import cv2

cap = cv2.VideoCapture('YouTube.mp4')

while cap.isOpened():
    ret, frame = cap.read()
    if ret:
        gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        cv2.imshow('frame', gray_frame)
    if cv2.waitKey(1) == 27:
        break

cap.release()
cv2.destroyAllWindows()
```

7-2-3. OpenCV 電腦攝影機操作

取得電腦攝影機的影像

在 OpenCV 的 VideoCapture 物件除了開啟影片檔案，也可以開啟攝影機。例如：

7-3 取得網路攝影機的影像。

```
import cv2

cap = cv2.VideoCapture(0)

while cap.isOpened():
    ret, frame = cap.read()
    cv2.imshow('frame', frame)
    if cv2.waitKey(1) == 27:
        break

cap.release()
cv2.destroyAllWindows()
```

注意：cv2.VideoCapture(0) 的 0 代表第一台攝影機，1 代表第二台，依此類推。

更改影像的解析度

在 Python 程式建立 VideoCapture 物件後，可以呼叫 set 方法更改影片的寬、高和影格率的屬性。例如：

7-3a 更改影像的解析度。

```
import cv2

cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 320)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 180)
cap.set(cv2.CAP_PROP_FPS, 25)

while cap.isOpened():
    ret, frame = cap.read()
    cv2.imshow('frame', frame)
    if cv2.waitKey(1) == 27:
        break

cap.release()
cv2.destroyAllWindows()
```

將影像寫入影片檔案

OpenCV 可以建立 VideoWrite 物件來寫入影片檔案。例如：

7-3b 將影像寫入影片檔案。

```
import cv2

cap = cv2.VideoCapture(0)
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('output.avi', fourcc, 20, (640, 480))

while cap.isOpened():
    ret, frame = cap.read()
    if ret == True:
        out.write(frame)
        cv2.imshow('frame', frame)
        if cv2.waitKey(1) == 27:
            break
    else:
        break

cap.release()
out.release()
cv2.destroyAllWindows()
```

7-3b 的 cv2.VideoWriter_fourcc() 為設定影片的編碼格式，參數為編碼字串。例如：cv2.VideoWriter_fourcc(*'XVID') 代表編碼成 MPEG-4 (副檔名 .avi)。

編碼名稱	編碼字串	影片副檔名
YUV	*'I420'	.avi
MPEG-I	*'PIMT'	.avi
MPEG-4	*'XVID'	.avi
MP4	*'MP4V'	.mp4
Ogg Vorbis	*'THEO'	.ogv

7-3. OpenCV 人臉辨識

OpenCV 內建的是哈爾特徵 (Haar-like feature) 演算法。

- 使用哈爾特徵前，先下載人臉偵測的聯集分類器。
- https://github.com/opencv/opencv/tree/master/data/haarcascades

以下是跟人臉有關的：

分類器	說明
haarcascade_frontalface_default.xml	人臉正面與側面
haarcascade_frontalface_alt2.xml	人臉正面效果比較好
haarcascade_profileface.xml	人臉側面效果比較好
haarcascade_eye.xml	偵測眼睛

7-3-1. 人臉辨識

圖片的人臉辨識

先對一個圖片進行測試與調整相關參數。例如：

7-4 圖片的人臉辨識。

```
import cv2

face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_alt2.xml')
image = cv2.imread('demo.jpeg')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
faces = face_cascade.detectMultiScale(gray, 1.1, 3)
# faces = face_cascade.detectMultiScale(gray, 1.05, 4)

for (x, y, w, h) in faces:
    frame = cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 3)

cv2.namedWindow('frame', cv2.WINDOW_NORMAL)
cv2.imshow('frame', frame)

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

其中，更多有關視窗的屬性設定：

- cv2.WINDOW_NORMAL：視窗大小可改變。
- cv2.WINDOW_AUTOSIZE：視窗大小不可改變。
- cv2.WINDOW_FREERATIO：自適應比例。
- cv2.WINDOW_KEEPRATIO：保持比例。

以下請參考程式結果：

- 當有人臉沒被辨識，需要調整參數（Demo 7-4.py）：
- 調整參數後（Demo 7-4a.py）：

圖片的眼睛辨識

辨識出人臉後，再進行眼睛辨識。使用巢狀 for 迴圈，第一層走訪人臉，第二層走訪該人臉的眼睛。例如：

7-5 圖片人臉與眼睛辨識。

```
import cv2

face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_alt2.xml')
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')
frame = cv2.imread('demo.jpeg')
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
# faces = face_cascade.detectMultiScale(gray, 1.1, 3)
faces = face_cascade.detectMultiScale(gray, 1.05, 4)

for (x, y, w, h) in faces:
    frame = cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 3)
    face_rect = gray[y:y+h, x:x+w]
    eyes = eye_cascade.detectMultiScale(face_rect, 1.15, 8)
    for (ex, ey, ew, eh) in eyes:
        frame = cv2.rectangle(frame, (x + ex, y + ey), (x + ex + ew, y +
                              ey + eh), (0, 255, 0), 2)

cv2.namedWindow('frame', cv2.WINDOW_NORMAL)
cv2.imshow('frame', frame)

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

即時影像的人臉辨識

7-6 即時影像的人臉辨識。

```
import cv2

faceCascade=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

while True:
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(gray, scaleFactor = 1.1,
                                minNeighbors = 5, minSize = (30, 30))
    print('人臉數: ', len(faces))
    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
    cv2.imshow('preview', frame)
    if cv2.waitKey(1) == 27:
        break

cap.release()
cv2.destroyAllWindows()
```

即時影像的人臉與眼睛辨識

7-7 即時影像的人臉與眼睛辨識。

```
import cv2

faceCascade=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

while True:
    ret, frame = cap.read()
    frame = cv2.rotate(frame, rotateCode = 1)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(gray, scaleFactor = 1.1,
                                minNeighbors = 5, minSize = (30, 30))
    print('人臉數: ', len(faces))
    for (x, y, w, h) in faces:
        frame = cv2.rectangle(frame, (x, y), (x + w, y + h),(0, 255, 0),3)
        face_rect = gray[y:y + h, x:x + w]
        eyes = eye_cascade.detectMultiScale(face_rect, 1.3, 8)
        for (ex, ey, ew, eh) in eyes:
            cv2.rectangle(frame, (x + ex, y + ey), (x + ex + ew, y + ey +
                            eh), (0, 255, 0), 2)
    cv2.imshow('preview', frame)
    if cv2.waitKey(1) == 27:
        break

cap.release()
cv2.destroyAllWindows()
```

7-3-2. 特定的人臉辨識

單一個的人臉辨識

藉由專門分析人臉特徵的演算法找出的特徵值，然後跟已經儲存的特徵值比對，判定是誰的臉。

分為三階段：取樣、訓練、辨識。

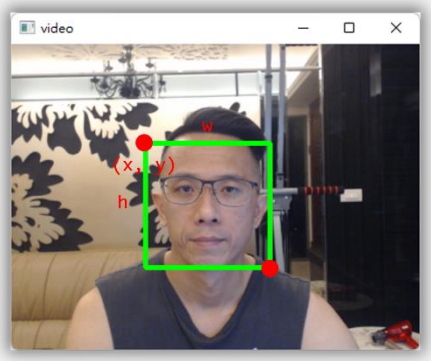
1. 取樣：收集訓練用的人臉圖片，建議每人有 100 張才有比較好的準確率。
2. 訓練：OpenCV 提供三種演算法，Eigen、Fisher、LBPH。
3. 辨識：依照訓練完的資料，判斷目前攝影機看到的人臉屬於哪一位。

取樣階段 我們透過程式碼來自動抓取 100 張人臉圖片，這樣比用相機拍攝 100 張圖片要來的方便快速。

7-8-1 人臉取樣。(7-8-1-capture.py)

```
import cv2

ESC = 27
n = 1
index = 0
total = 100

def saveImage(face_image, index):
    filename = 'images/h0/{:03d}.pgm'.format(index)
    cv2.imwrite(filename, face_image)
    print(filename)

face_cascade=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
cap = cv2.VideoCapture(0)
ratio = cap.get(cv2.CAP_PROP_FRAME_WIDTH) /
cap.get(cv2.CAP_PROP_FRAME_HEIGHT)
WIDTH = 400
HEIGHT = int(WIDTH / ratio)
cv2.namedWindow('video', cv2.WINDOW_NORMAL)

while n > 0:
    ret, frame = cap.read()
    frame = cv2.resize(frame, (WIDTH, HEIGHT))
     frame = cv2.flip(frame, 1)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

```
    faces = face_cascade.detectMultiScale(gray, 1.1, 3)

    for (x, y, w, h) in faces:
        frame = cv2.rectangle(frame,(x, y),(x + w, y + h),(0, 255, 0),3)
        if n % 5 == 0:
            face_img = gray[y: y + h, x: x + w]
            face_img = cv2.resize(face_img, (400, 400))
            saveImage(face_img, index)
            index += 1
            if index >= total:
                print('get training data done')
                n = -1
                break
        n += 1

    cv2.imshow('video', frame)
    if cv2.waitKey(1) == 27:
        cv2.destroyAllWindows()
        break
```



訓練階段　將 h0、h1… 資料夾中的圖片取出、標籤化後送進人臉特徵演算法中計算特徵值，並將結果存檔後以供後續使用。

7-8-2　訓練取樣集。(7-8-2-train.py)

```
import cv2
import numpy as np

images = []
labels = []

for index in range(100):
    filename = 'images/h0/{:03d}.pgm'.format(index)
    print('read ' + filename)
    img = cv2.imread(filename, cv2.COLOR_BGR2GRAY)
    images.append(img)
    labels.append(0)    # 第一張人臉的標籤為 0

print('training...')
model = cv2.face.LBPHFaceRecognizer_create()
model.train(np.asarray(images), np.asarray(labels))
model.save('faces.data')
print('training done')
```

辦識階段　這一階段是拿訓練好的結果，來驗證辦識效果是否準確，所以我們開啟攝影機，將攝影機拍到的人臉來跟訓練結果比對，如果發現有『認識』的人臉，就在畫面上顯示這個人臉的名字。

7-8-3　人臉辨識。(7-8-3-recognition.py)

```
import cv2

model = cv2.face.LBPHFaceRecognizer_create()
model.read('faces.data')
print('load training data done')
face_cascade=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

cap = cv2.VideoCapture(0)
cv2.namedWindow('video', cv2.WINDOW_NORMAL)
names = ['pllai']

while True:
    ret, frame = cap.read()
    frame = cv2.resize(frame, (600, 400))
    frame = cv2.flip(frame, 1)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.1, 3)

    for (x, y, w, h) in faces:
        frame=cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 3)
        face_img = gray[y: y + h, x: x + w]
        face_img = cv2.resize(face_img, (400, 400))
        val = model.predict(face_img)
        print('label:{}, conf:{:.1f}'.format(val[0], val[1]))
        if val[1] < 50:
            cv2.putText(frame, names[val[0]], (x, y - 10),
                        cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 0), 3)

    cv2.imshow('video', frame)
    if cv2.waitKey(1) == 27:
        cv2.destroyAllWindows()
        break
```

練習 7-3

1. 7-3-2 小節的程式只能辨識單一個人臉，請修改成可以辨識多個人臉。