

# 人工智慧 -- 影像辨識篇

---

## Chapter 1 : OpenCV X 人臉辨識

賴秉樑 debugger

學院創辦人

課程網址 <https://max543.com/debugger>

## 課程目的

以少量的上課時間，快速進入影像辨識的技術應用



## PERSONAL INFO

### 姓名 NAME

賴秉樑 debugger

### 學歷 EDUCATION

台科大資工、電子雙學位  
中興資訊科學與工程碩士

### 經歷 EXPERIENCE

國立大學電子系、資工系講師  
職業訓練、產投、大學業師  
竹科半導體研發工程師  
旗標科技師資培訓講師  
極限翻轉程式學院創辦人

## 個人興趣 INTEREST

# 玩數學、打電腦

## 我的座右銘 MOTTO

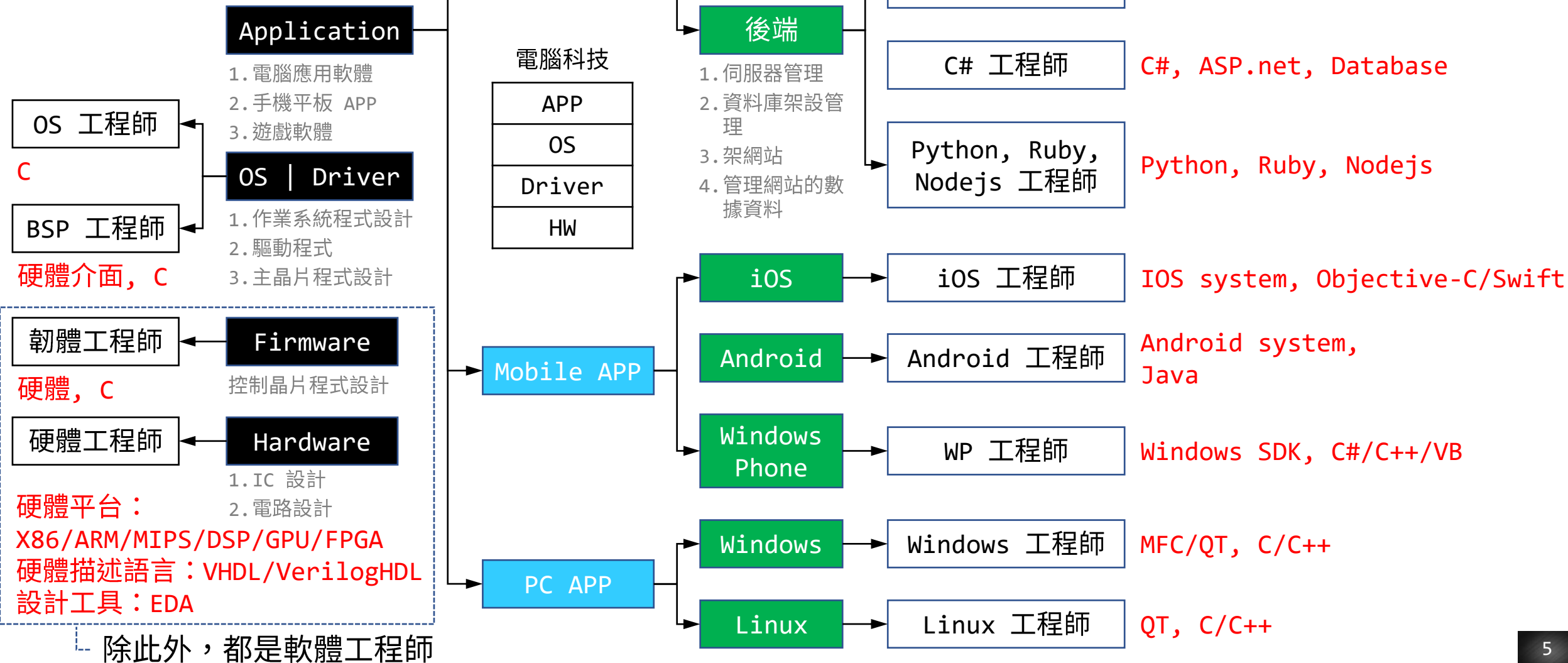
# 動手做、樂趣多

# Coder, Hacker, and Maker

---

- 程式設計師 (coder, 或 programmer)
  - ✓ 主要透過編輯程式，簡稱編程 (coding)，它可以指在程式設計某個專業領域的專業人士，或是從事軟體撰寫，程式開發、維護的專業人員。
- 駭客 (hacker)
  - ✓ 除了精通程式設計、作業系統的人可以被視作駭客，對硬體裝置做創新的工程師通常也被認為是駭客，精通網路入侵的人也被看作是駭客。
- 創客 (maker)
  - ✓ 又稱自造者。是一群酷愛科技、熱衷實踐的人群，他們以分享技術、激發的創造力與交流思想為樂。

# 科技業工程師 主要分類



# 運算思維為何很重要？ (1/2)

- 學會了運算思維，讓我們也能擁有電腦科學家面對問題時，所持有的科學方法。  
各種領域都需要運算思維，例如：
- **科學與工程領域**
  - ✓ 利用運算模擬建築結構，以確認安全性。
  - ✓ 利用運算預測氣象，以增加準確性。
- **金融領域**
  - ✓ 利用運算研究經濟大數據。
  - ✓ 利用運算完成自動交易。

# 運算思維為何很重要？ (2/2)

---

- **人文與社會領域**

- ✓ 利用運算分析，並優化廣告投放策略。
- ✓ 利用運算分析人口老化趨勢，與醫療資源分布。

- **藝術領域**

- ✓ 利用運算建構三維動畫。
- ✓ 利用運算創作數位音樂。

- **工業設計**

- ✓ 利用運算實現工業 4.0。
- ✓ 利用運算實現更多的增值應用，例如：自動化與智慧化。

因為有這些程式  
生活更美、更好





# 科技業的下一個神話

---

電子商務 X 社群網路 X 串流媒體



物聯網 X 人工智慧 X 區塊鏈

Python

Python

Python

Hello, World! 最像英文的 **Python**



## Python

```
# 印出 Hello World! 字串物件  
print("Hello World!")
```

## C

```
/* 印出 Hello World! 字串物件*/  
include <stdio.h>  
  
int main()  
{  
    printf("Hello, World!\n");  
    return 0;  
}
```

## C++

```
//印出 Hello World! 字串物件  
#include <iostream>  
  
using namespace std;  
  
int main()  
{  
    cout << "Hello World" << endl;  
    return 0;  
}
```

## Java

```
//印出 Hello World! 字串物件  
public class HelloWorld{  
  
    public static void main(String []args){  
        System.out.println("Hello World");  
    }  
}
```

# Outline

---

- 1-1 AI 的主流技術：機器學習
- 1-2 在 Windows 上安裝 OpenCV
- 1-3 OpenCV 的基本使用
- 1-4 OpenCV 人臉辨識

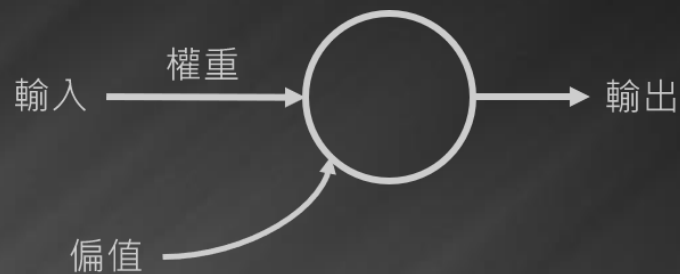
# 1-1 AI 的主流技術：機器學習

---

- 早期的技術：將問題以人工分析後，轉成程式語言的規則法 (rule-based)。
- 現在的技術：機器學習，準備一些問題與對應的答案給電腦後，讓它自行找出其中的規則，並且有能力針對類似的問題給出正確的答案。
  - ✓ 歸功於網路龐大的資訊量，或各式感測器收集的資料。

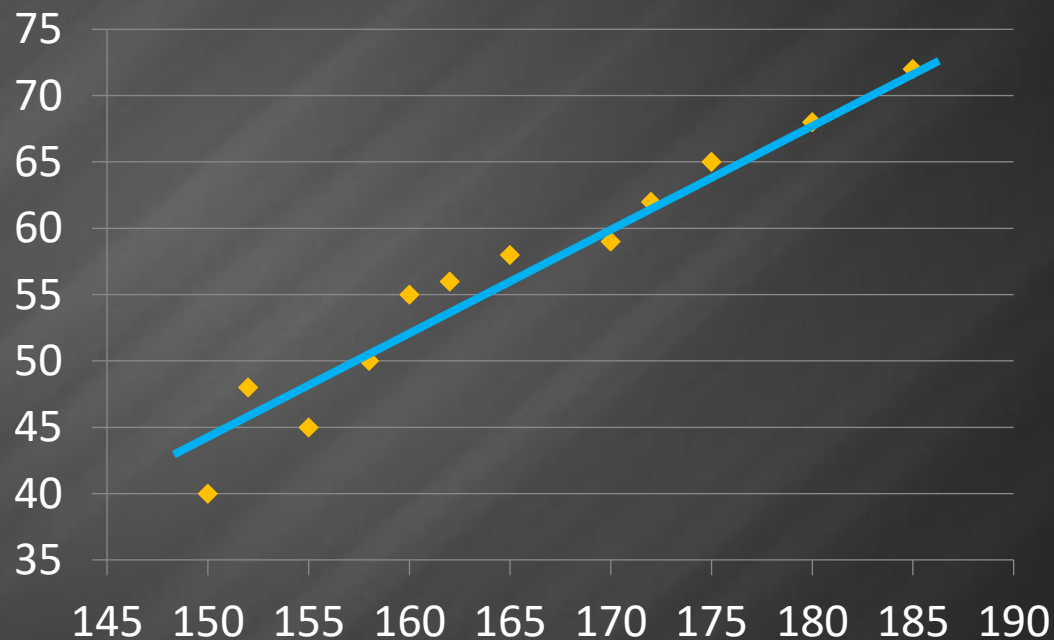
# 機器學習的神經元如何學習迴歸問題

某一班學生的身高和體重是否相關？  
能否用身高來推測某位學生的體重？



$$\text{輸出} = \text{輸入} \times \text{權重} + \text{偏值}$$

身高	體重
150	40
152	48
155	45
158	50
160	55
162	56
165	58
170	59
172	62
175	65
180	68
185	72

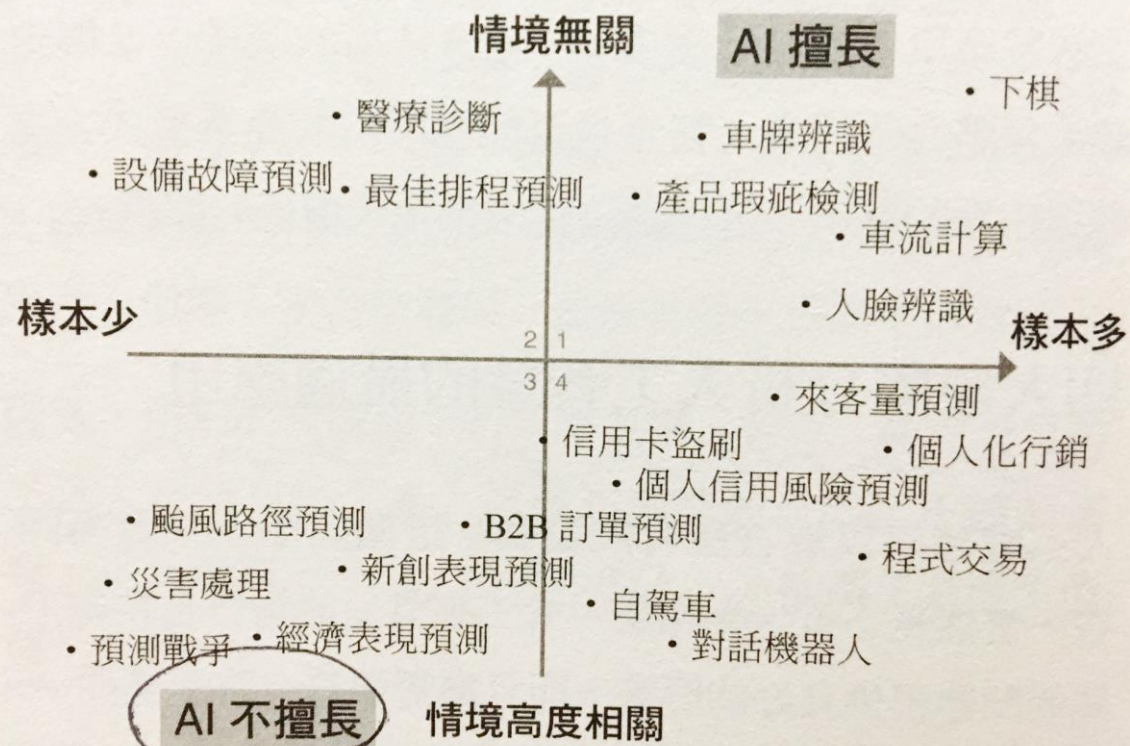


$$y = 0.8337x - 81.331$$

# 現在的人工智慧

- 現在的 AI 只能針對符合特定形式與條件的問題，提供良好的解答。
- 目前以機器學習為基礎的人工智慧，不可能擁有人類的思考及情緒。

圖 3.1 機器學習擅長解決什麼問題？



# 人工智慧發展簡史

## 第一波

1950-1960

### 符號邏輯

把人的**思考邏輯**放進電腦

由領域專家寫下決策邏輯。

人類還沒辦法清楚理解自己的思考過程，如何告訴電腦？

失敗

## 第二波

1980-1990

### 專家系統

把人的**所有知識**放進電腦

由領域專家寫下經驗規則。

太多難題人類無法解答、無法寫成規則、無法以程式碼表示。

失敗

## 第三波

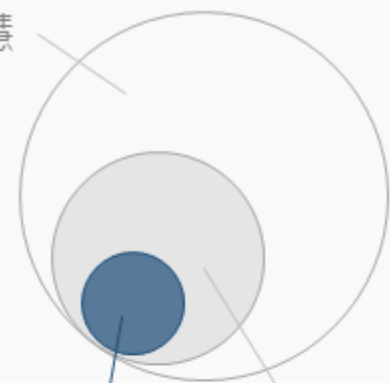
2010-Present

### 機器學習

把人的**所有看見**放進電腦

由領域專家提供歷史資料，讓電腦自己歸納規則。

人工智慧



機器學習  
(第三波人工智慧的代表技術)

深度學習  
(機器學習技術中成長最快、表現最佳)

### 專家系統

專家定義規則

### 傳統機器學習

(與深度學習區隔)

電腦定義規則  
專家定義特徵

### 深度學習

(多層類神經網路)

電腦定義規則 (更準)  
電腦定義特徵



# Outline

---

1-1 AI 的主流技術：機器學習

1-2 在 Windows 上安裝 OpenCV

1-3 OpenCV 的基本使用

1-4 OpenCV 人臉辨識

# 學習 AI 時的重要工具 - Python

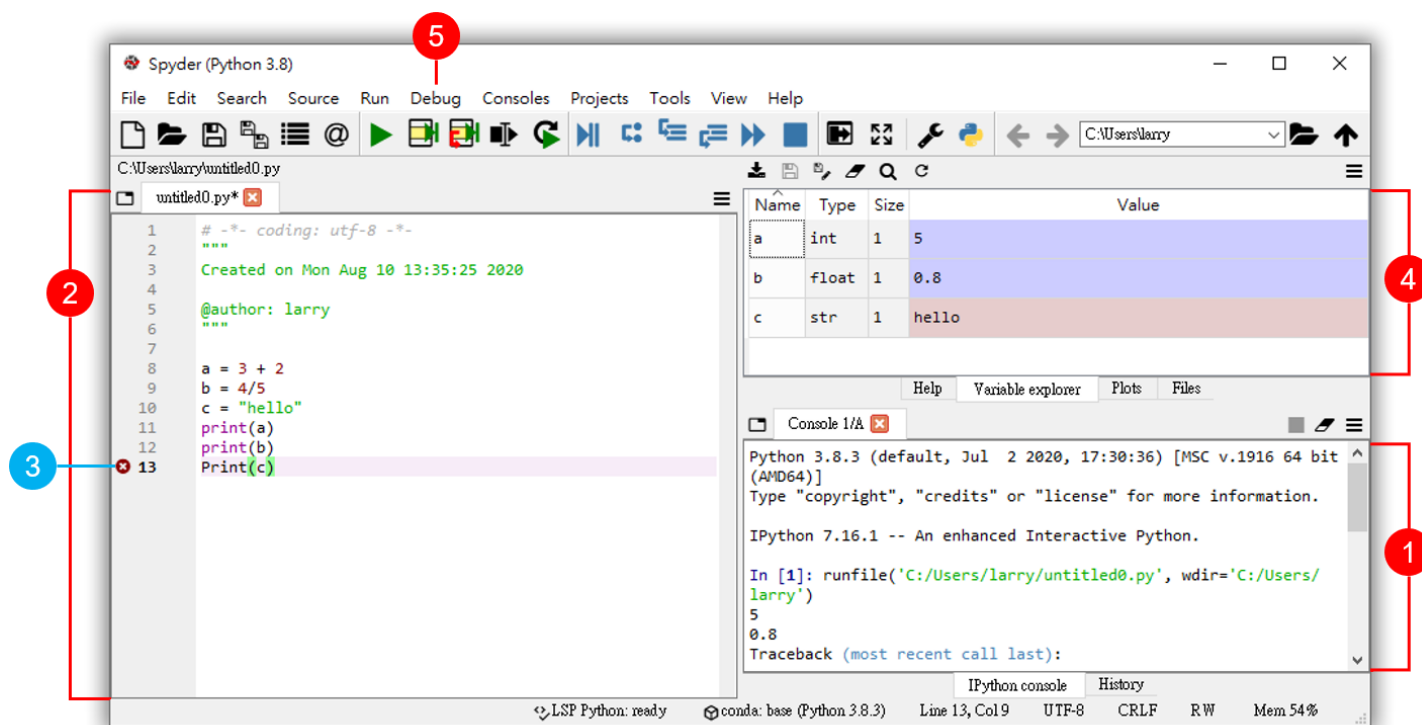
---



# Python 的編譯器：Anaconda

ex1-1

- Anaconda 包含了 Python 直譯器與許多好用的開發工具，而且提供了數百個第三方的函式庫，涵蓋科學、數學、工程、數據分析的 Python 模組。
  - 網址：<https://www.anaconda.com/products/individual>，下載並安裝適當的版本。
  - 安裝後，執行 Anaconda 的整合開發環境 (IDE)：Spyder。



1. IPython 主控台：用來測次單行程式碼，以及程式的輸出資訊。
2. 程式編輯窗格：編寫程式專案的地方。
3. 語法錯誤提示：在執行程式前，可事先發現語法錯誤的程式碼。
4. 變數窗格：可知道目前程式物件的值為何。
5. Debug (除錯選單)：可依序執行程式碼，用來查找錯誤與追蹤變數值。

# Python 的最基礎

---

- 物件
- 物件的資料型別
- 整數與浮點數物件
- 變數
- 內建函式
- 匯入模組

# 物件

- 英文一般寫句子時，會以**名詞 + 動詞**。Python 是以**物件.方法**來描述。

文章寫作	寫 Python 程式	
車子	<code>car</code>	car 物件
車子向前進	<code>car.start()</code>	car 物件的 start 方法

- 方法後面會加上括號()，有些方法需要加入額外的參數。
  - ✓ 例如：`car.go(100)`，車子加速到 100。
- 若方法有多個參數，以逗點分隔。
  - ✓ 例如：`car.left(50, 30)`，以 50 的速度，向左轉 30 度。

# 練習：字串物件

- 在互動模式中，輸入下列敘述：(>>> 指的是在互動模式中，執行單行敘述)

```
>>> "abc".upper() —— 使用字串物件 "abc" 的 upper() 方法，  
                        將字串轉成大寫  
'ABC'  
  
>>> "abc".find('b') —— find() 方法尋找 'b' 的位置 (從 0 開始)  
1  
  
>>> "abc".replace('b', 'z') —— replace() 方法將所有的 'b'  
                                取代成 'z'  
'azc'
```

- 不同的物件會有不同的方法。例如：字串物件與整數物件。

- 除字串物件以雙引號或單引號來表示，寫程式常有整數與浮點數（小數）物件，例如：111 與 11.1。

```
>>> 111 + 111 ————— 整數物件相加
```

```
222
```

```
>>> "111" + "111" ————— 字串物件串聯
```

```
'111111'
```

- 上述 + 的運算，因物件的資料不同而產生不同的結果。物件的種類，程式語言稱之為『物件型態』或『資料型態』(Data Type)。

# 練習：要分清楚資料型別

- 兩個資料型別若不同，可能會導致程式錯誤。

```
>>> 111 + "111" —— 不同型別的資料相加發生錯誤
```

```
Traceback (most recent call last):
```

```
  File "<ipython-input-6-4832c22160be>", line 1, in  
<module>
```

```
    111 + "111"
```

```
TypeError: unsupported operand type(s) for +: 'int'  
and 'str'
```



- 兩個資料型別若要運算，可以使用型別轉換。

```
>>> str(111) + "111"  —— str() 可轉換物件為字串型別
'111111'
>>> 111 + int("111") —— int() 可轉換物件為整數型別
222
```

- 整數與浮點數的數學運算

- ✓ + (加)、- (減)、\* (乘)、/ (取商)、// (取商，整數)、% (取餘數)、\*\* (指數)。
- ✓ Python 允許整數與浮點數直接運算，執行下列程式：

## 加法 (+)

```
>>> 10 + 5.5  
15.5
```

## 減法 (-)

```
>>> 10 - 5.5  
4.5
```

## 乘法 (\*)

```
>>> 10 * 5.5  
55.0
```

## 除法取商 (/)

```
>>> 10 / 5.5  
1.8181818181818181
```

## 除法取整數商 (//)

```
>>> 10 // 5.5  
1.0
```

## 取餘數 (%)

```
>>> 10 % 5.5  
4.5
```

## 指數 (\*\*)

```
>>> 4 ** 0.5, 8 ** (1/3)  
(2.0, 2.0)
```

### TIP

1. 整數與浮點數做運算，結果一定為浮點數。
2. 只有整數與整數做除法取商，結果為浮點數。

- 『變數』(variable) 就像是掛在物件的名牌，幫物件取名之後，讓我們方便識別物件與操作，其語法為：

變數名稱 = 物件

- 例如：

```
>>> n1 = 123456789 —— 將整數物件 123456789 指派給變數 n1
>>> n2 = 987654321 —— 將整數物件 987654321 指派給變數 n2
>>> n1 + n2 —— 實際上是 123456789 + 987654321
1111111110
```

# 常用的變數運算

- 把整數加上特定的值：

```
>>> x = 1
>>> x = x + 1
>>> x
2
```

- 常用的簡式：

簡式	意義
$x += 2$	$x = x + 2$
$x -= 2$	$x = x - 2$
$x *= 2$	$x = x * 2$

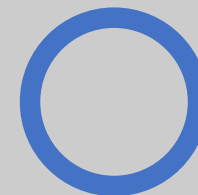
簡式	意義
$x /= 2$	$x = x / 2$
$x //= 2$	$x = x // 2$
$x %= 2$	$x = x \% 2$

- 使用有意義的變數 (variable) 的名稱。

```
a = 3.1  
b = 2.2  
c = a * b * b
```



```
pi = 3.1  
radius = 2.2  
# 使用公式計算圓面積  
circle_area = pi * radius * radius
```



# 寫出可讀性高的程式 (2/2)

---

- 將程式加上註解。
  - ✓ 註解可以幫助其他人了解這個程式。
- 較佳的註解：

# 註解 1：使用公式計算圓面積

- 不好閱讀的註解：

# 註解 2：將圓周率乘半徑乘半徑



# 內建函式

- 『**函式**』 (function) 是一段預先寫好的程式，方便重複使用。而程式先將經常使用到的功能以函式的形式先寫好，稱為『**內建函式**』。
- 例如：`print()` 是最常用的顯示函數：

```
>>> print("abc") —— 顯示字串物件
```

```
abc
```

```
>>> print("abc".upper()) —— 顯示字串物件.方法的執行結果
```

```
ABC
```

```
>>> print(111 + 111) —— 顯示整數物件運算的結果
```

```
222
```

# 匯入模組

- 內建函式不就越越多越好？若內建函式無限制增加，會導致啟動速度越來越慢，執行時佔用的記憶體越來越多。
- 『**模組**』 (module)，就是將同一類的函式打包成模組，預設不會啟用。需要時，再用**匯入** (import) 的方式啟用。預先寫好的稱為『**內建模組**』。

```
>>> import time —— 匯入時間相關的 time 模組
```

```
>>> time.sleep(3) —— 執行 time 模組的 sleep() 函式，暫停 3 秒
```

```
>>> from time import sleep —— 從 time 模組裡匯入 sleep() 函式
```

```
>>> sleep(5) —— 執行 sleep() 函式，暫停 5 秒
```

# 練習：匯入模組

程式

暫停 3 秒後，印出 Hello World! 字串物件。

```
# 暫停 3 秒後，印出 Hello World!  
from time import sleep  
  
sleep(3)  
print("Hello World!")
```

指的是在程式編輯窗格中編輯與執行。

# 觀念整理

各種程式語言的語法邏輯都差不多，就像人類語言的文法。

1. 程式的每一個東西都是**物件**，有些物件有其特定的操作方法。
2. 基本物件有**資料型別**，型別不同，結果不同。甚至有時會錯誤。
3. **變數**是物件的名牌而已，也方便程式設計師操作。
4. 使用有意義的變數名稱與註解。
5. **內建函式**是經常用的函式，預先寫好的。
6. 適當的**匯入模組**，能精簡效能。

先安裝電腦鏡頭

<https://bit.ly/3yEVgF9>



# 安裝第三方軟體，讓手機成為電腦鏡頭

Google

droidcam



[全部](#) [影片](#) [圖片](#) [新聞](#) [購物](#) [更多](#)

工具

約有 690,000 項結果 (搜尋時間：0.31 秒)

<https://www.dev47apps.com> [翻譯這個網頁](#)

**Dev47Apps**


DroidCam turns your phone/tablet into a webcam for your PC. Use it with chat programs like Zoom, MS Teams, and Skype. Main Features: – Chat using “DroidCam ...

<https://www.kocnc.com.tw> [Android App分享介紹](#)



# 安裝電腦端軟體

Dev47Apps

 **DroidCam**  
Use your phone as a webcam


DroidCam turns your phone/tablet into a webcam. Use it with chat programs like Zoom, Skype, etc.


Main Features:

- Chat using "DroidCam Webcam" or "DroidCam HD Mode"
- Connect over WiFi or USB cable.
- Unlimited free usage at standard definition
- Keep using your phone with DroidCam
- Simple, safe, efficient, and trusted




DroidCamX Pro Features:

- Switch to HD Mode for 720p/1080p high definition video.
- Camera controls: toggle light, enable continuous auto focus, zoom in/out
- Rotate, Flip & Mirror the video to better fit your setup.
- Adjust brightness, contrast, add video effects and adjust audio volume (Windows)
- USB-Only mode for extra privacy and security.
- Option to auto-mute phone calls.
- No Ads.


ANDROID APP ON  Google play

Download on the  App Store




The app works with a PC client component that installs the webcam driver on your device. Download the PC client below:

 Download Windows Client	 Download Linux Client	 Help & FAQs
--	--	--

## Windows

 **DroidCam Client**  
v6.4.3 Download

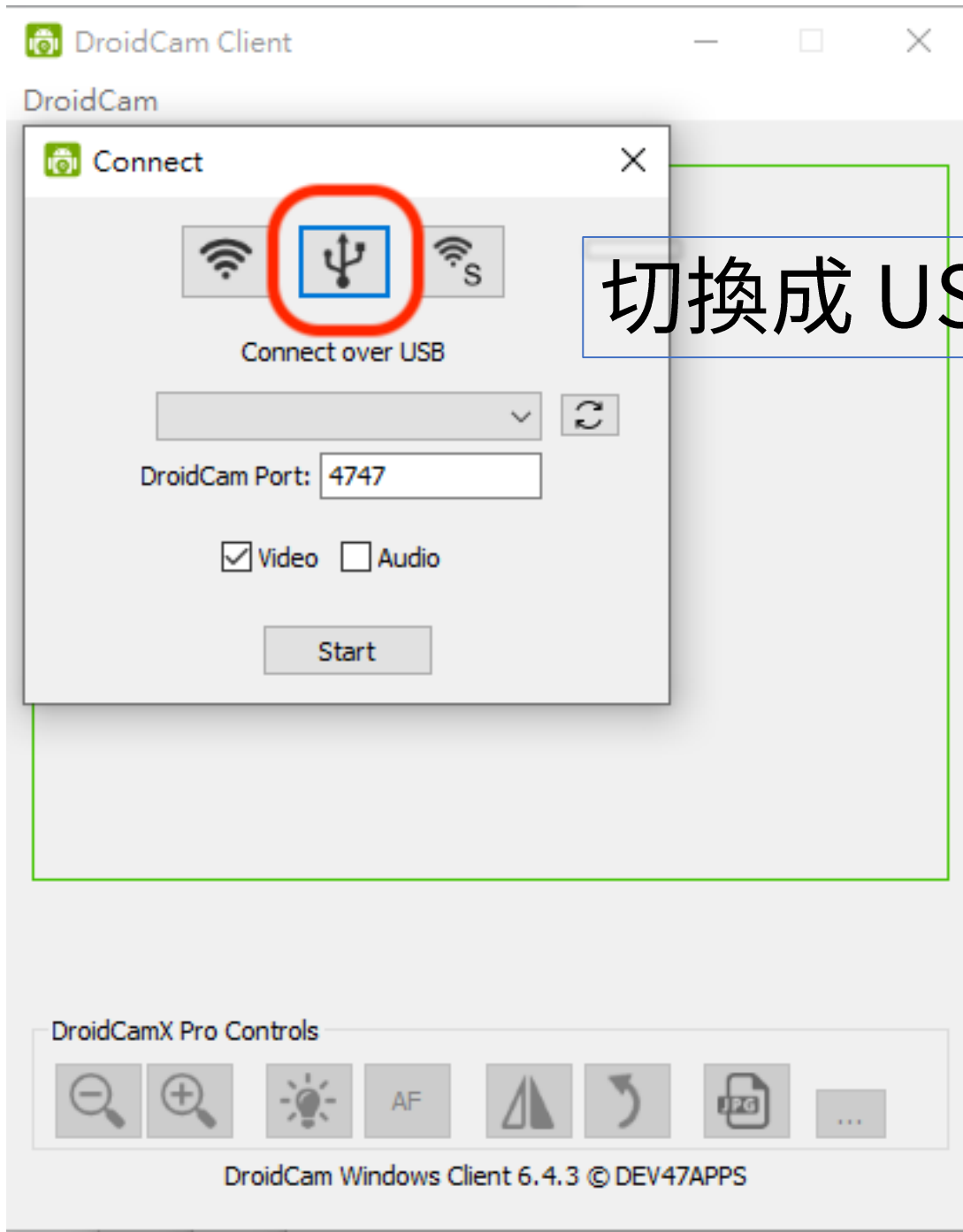
D

-  **DroidCam**  
新增
-  **DroidCam Client**  
新增
-  **HD Mode**  
新增

E

Windows taskbar icons: File Explorer, Chrome, V2, Task View, Settings





切換成 USB 模式

安裝手機端軟體

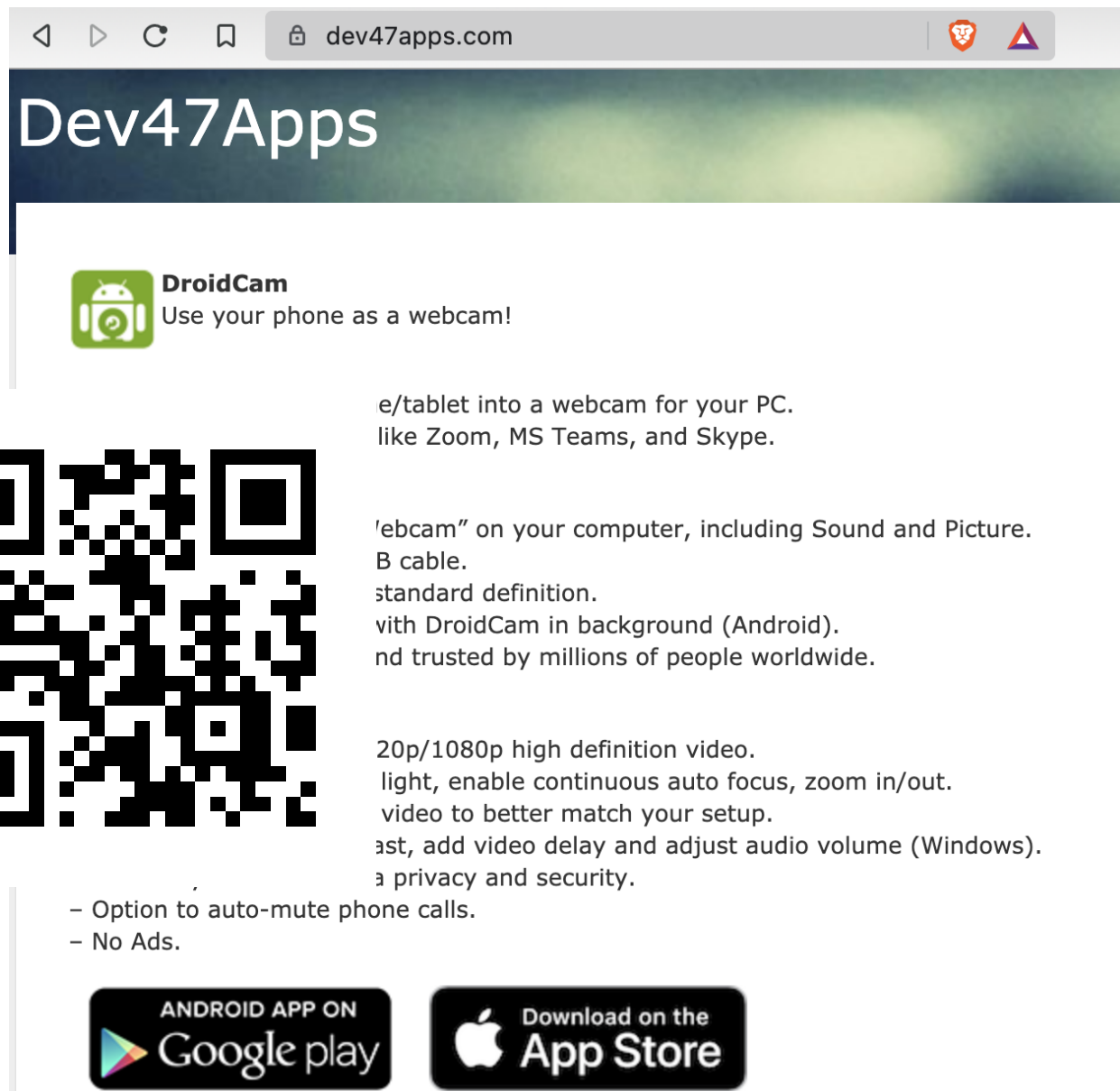
手機連線

<https://www.dev47apps.com/>


或掃描 QR code

選擇自己的手機作業系統

下載並安裝



Dev47Apps


 **DroidCam**  
Use your phone as a webcam!


...e/tablet into a webcam for your PC.  
like Zoom, MS Teams, and Skype.

...ebcam" on your computer, including Sound and Picture.  
B cable.  
standard definition.  
with DroidCam in background (Android).  
nd trusted by millions of people worldwide.

20p/1080p high definition video.  
light, enable continuous auto focus, zoom in/out.  
video to better match your setup.  
ast, add video delay and adjust audio volume (Windows).  
a privacy and security.

- Option to auto-mute phone calls.
- No Ads.

ANDROID APP ON  
 Google play

Download on the  
 App Store

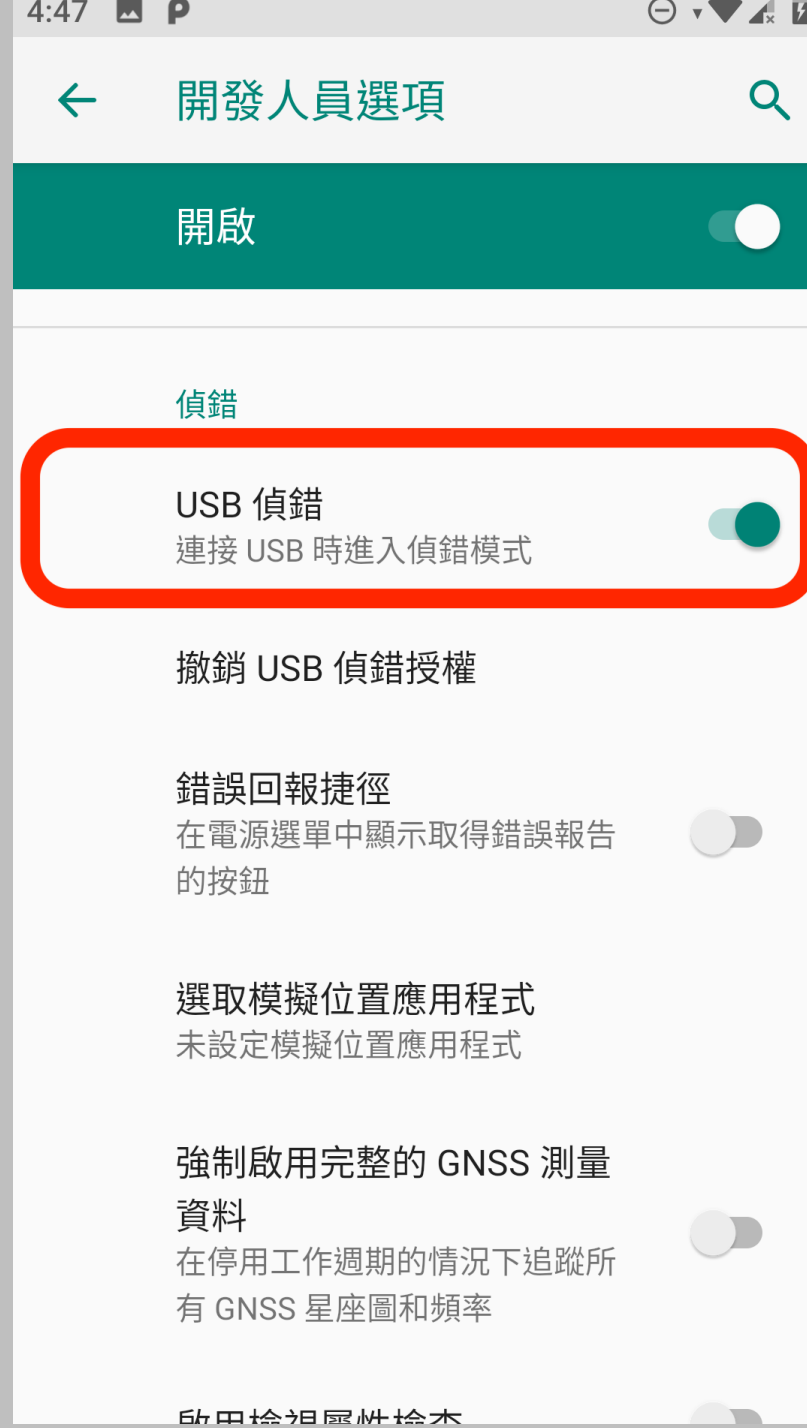
開啟 Android 手機  
開發人員模式

『手機設定/系統/關於手機』



開啟 USB 偵錯

『手機設定 / 系統 / 進階 / 開發人員  
選項』



# iOS 手機連接電腦需安裝 itunes

---

使用 iOS 手機不需額外設定

但電腦需要額外安裝 iTunes 軟體才能連接

Windows 電腦根據作業系統分別連線下載，並依指示安裝：

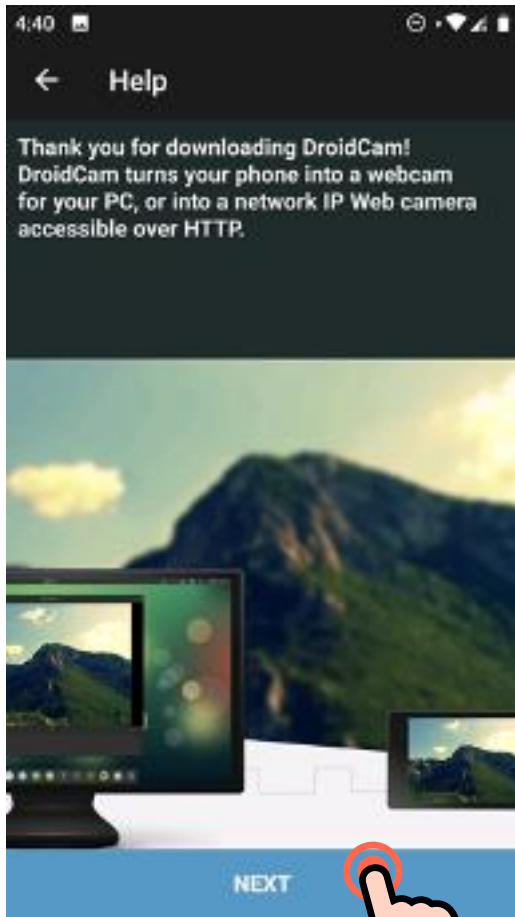
64 位元：

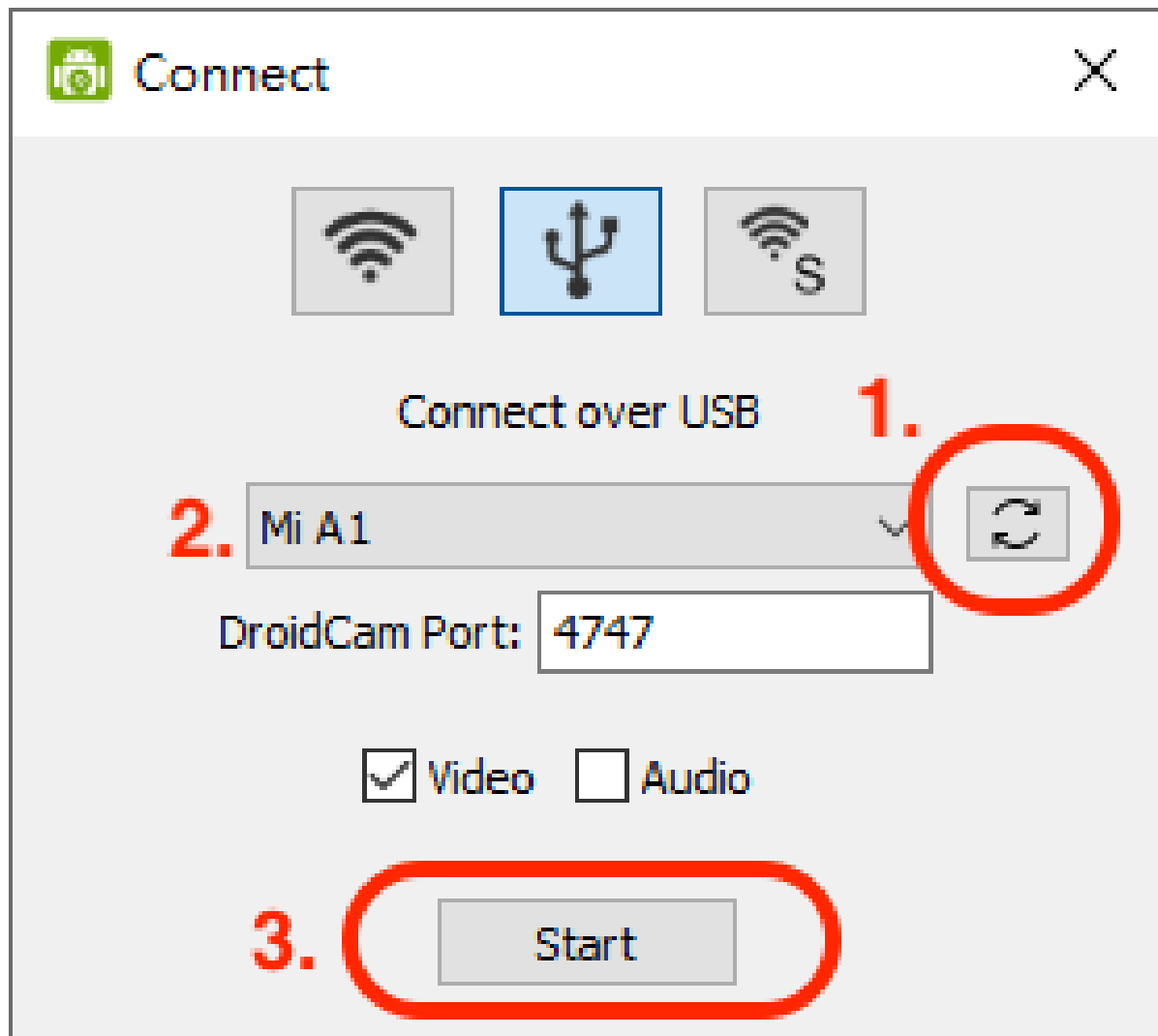
<https://www.apple.com/itunes/download/win64>

32 位元：

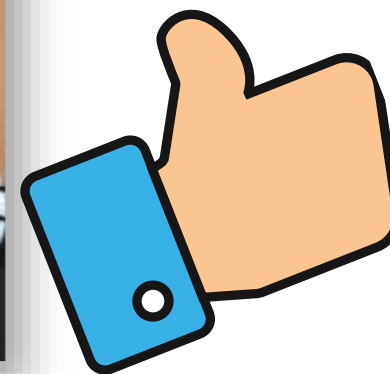
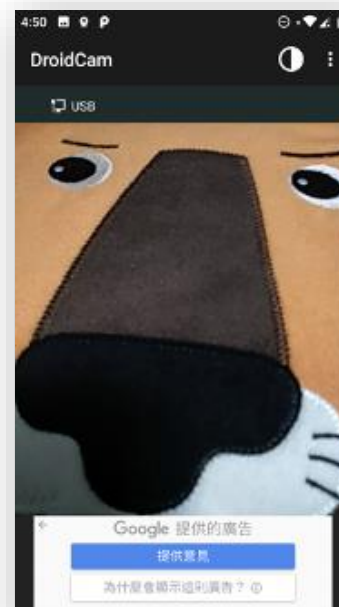
<https://www.apple.com/itunes/download/win32>







1. 按此重新整理
2. 確認裝置列表
3. 按 Start



# 影像處理應用：強大的 OpenCV



# 人臉偵測與追蹤



<https://learn.adafruit.com/diy-wifi-raspberry-pi-touch-cam>

## 1-2 在 Windows 安裝 OpenCV (1/2)

---

- 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`

## 1-2 在 Windows 安裝 OpenCV (2/2)

---

- 步驟一：安裝 OpenCV 的相關支援模組
- 步驟二：安裝 OpenCV
- 步驟三：檢查 OpenCV 是否成功安裝

# 步驟一：安裝 OpenCV 的相關支援模組

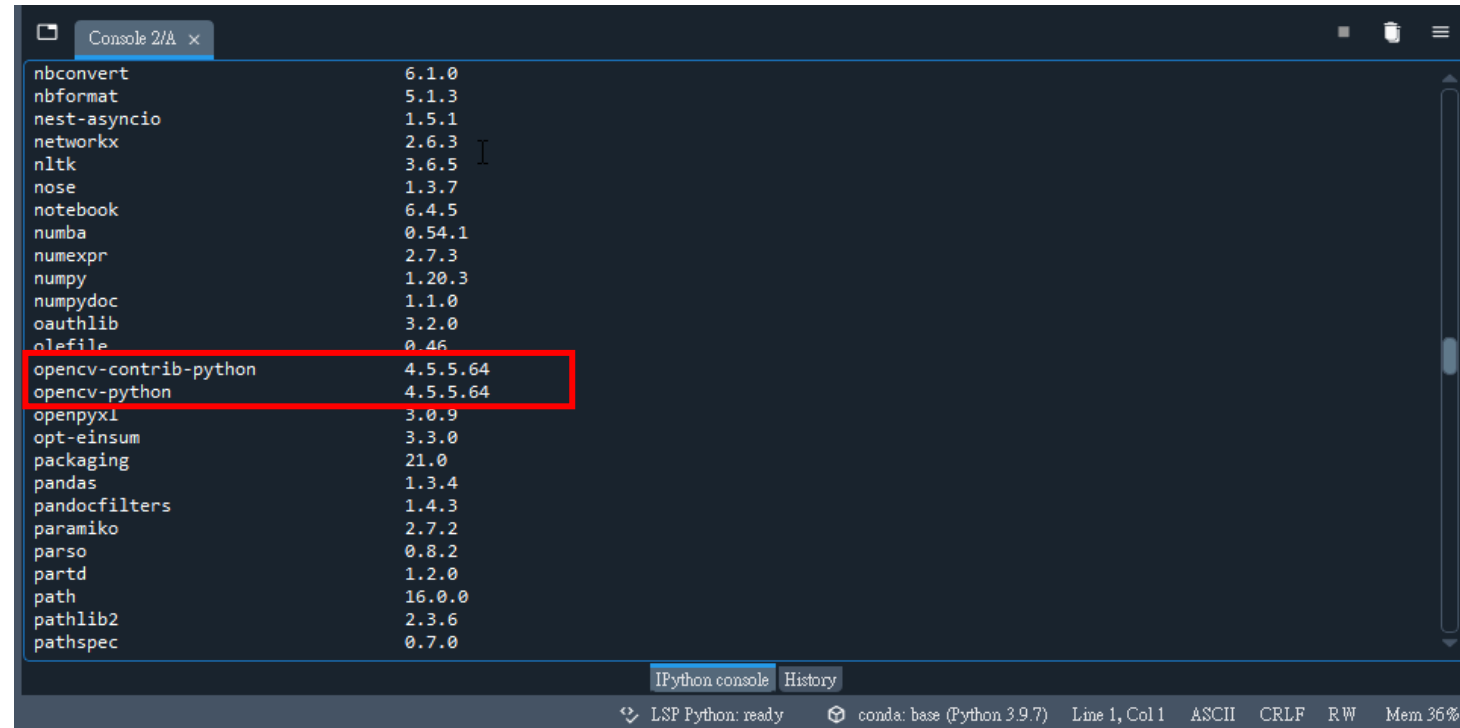
- 以最新版的 python 3.9.7 為例：

- ✓ 安裝支援的 Python 模組，numpy 為陣列容器（資料結構）、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
```

## 步驟二：安裝 OpenCV

- 使用 pip 安裝 OpenCV：  
    >>> `pip install opencv-python`  
    >>> `pip install opencv-contrib-python`
- 輸入指令來檢視虛擬環境安裝的套件清單：  
    >>> `pip list`



```
nbconvert 6.1.0
nbformat 5.1.3
nest-asyncio 1.5.1
networkx 2.6.3
nlTK 3.6.5
nose 1.3.7
notebook 6.4.5
numba 0.54.1
numexpr 2.7.3
numpy 1.20.3
numpydoc 1.1.0
oauthlib 3.2.0
olefile 0.46
opencv-contrib-python 4.5.5.64
opencv-python 4.5.5.64
openpyxl 3.0.9
opt-einsum 3.3.0
packaging 21.0
pandas 1.3.4
pandocfilters 1.4.3
paramiko 2.7.2
parso 0.8.2
partd 1.2.0
path 16.0.0
pathlib2 2.3.6
pathspec 0.7.0
```

IPython console History

LSP Python: ready conda: base (Python 3.9.7) Line 1, Col 1 ASCII CRLF RW Mem 36%

## 步驟三：檢查 OpenCV 是否成功安裝

---

- 檢查 Python 是否安裝好 OpenCV。

```
>>> import cv2
```

```
>>> cv2.__version__
```

```
'4.5.4'
```

```
# 顯示目前 OpenCV 的版本
```

# Outline

---

- 1-1 AI 的主流技術：機器學習
- 1-2 在 Windows 上安裝 OpenCV
- 1-3 OpenCV 的基本使用**
- 1-4 OpenCV 人臉辨識



# OpenCV 基本操作

目的：基本 OpenCV 處理圖片與影像

程式碼下載

<https://max543.com/debugger/class/python02/影像辨識/code/ch01.zip>

# 1-3 OpenCV 的基本使用

---

1-3-1 OpenCV 圖片處理

1-3-2 OpenCV 影片處理

1-3-3 OpenCV 網路攝影機操作

# 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，如下所示：

**1-1**

讀取與顯示圖片。(1-1.py)

```
import cv2
```

```
img = cv2.imread("koala.jpg")  
cv2.imshow("Koala", img)
```

```
gray_img = cv2.imread("koala.jpg", cv2.IMREAD_GRAYSCALE)  
cv2.imshow("Koala:gray", gray_img)
```

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

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。

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，如下所示：

## 1-1a 取得圖片資訊。(1-1a.py)

```
import cv2

img = cv2.imread("koala.jpg")
img2 = cv2.imread("koala.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，如下所示：

## 1-1b 調整圖片尺寸。(1-1b.py)

```
import cv2, imutils

img = cv2.imread("koala.jpg")
print(img.shape)
resized_img = imutils.resize(img, width = 300)
print(resized_img.shape)
cv2.imshow("Koala", img)
cv2.imshow("Koala:resized", resized_img)

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

進行下一小節前，需要先了解

## Python 的資料結構（容器）

目的：了解程式計算數值會選擇適合的容器

# Python 的基本資料結構

- 字串容器：由字元組成。例如：`string = "52python"`。
  - ✓ `string[0]`：字串以索引取值，結果為：`'5'`。
- tuple 容器：由資料物件組成。例如：`tuple = (1, '2', 3)`。
  - ✓ `tuple[0]`：tuple 以索引取值，結果為：`1`。
- 串列容器：由資料物件組成。例如：`list = [1, '2', 3]`。
  - ✓ `list[1]`：串列以索引取值，結果為：`'2'`。
- 集合容器：由資料物件組成。例如：`set = {1, '2', 3}`。
  - ✓ 集合為無序所以沒有索引取值
- 字典容器：由資料物件組成，以鍵：值表示。例如：`dick = {'A':1, 'B':'2', 'C':3}`。
  - ✓ `dick['B']`：字典以鍵取值，結果為：`'2'`。

# Python 的第三方資料結構：numpy.array

- Numpy：Python 的擴充模組，常用於機器學習的資料處理。

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




# 矩陣的切片 (Matrix)


```
import numpy as np
```

```
A = np.array([[1, 4, 2], [3, 2, 0]])
```

```
B = A[1:, 1:] # 2-D array 切片，第一個索引以列切片，第二個索引以行為切片
```

列 (row) 

$$A = \begin{bmatrix} 1 & 4 & 2 \\ 3 & 2 & 0 \end{bmatrix}_{2 \times 3}$$

行 (column) 

$$B = \begin{bmatrix} 2 & 0 \end{bmatrix}_{1 \times 2}$$

# 剪裁圖片

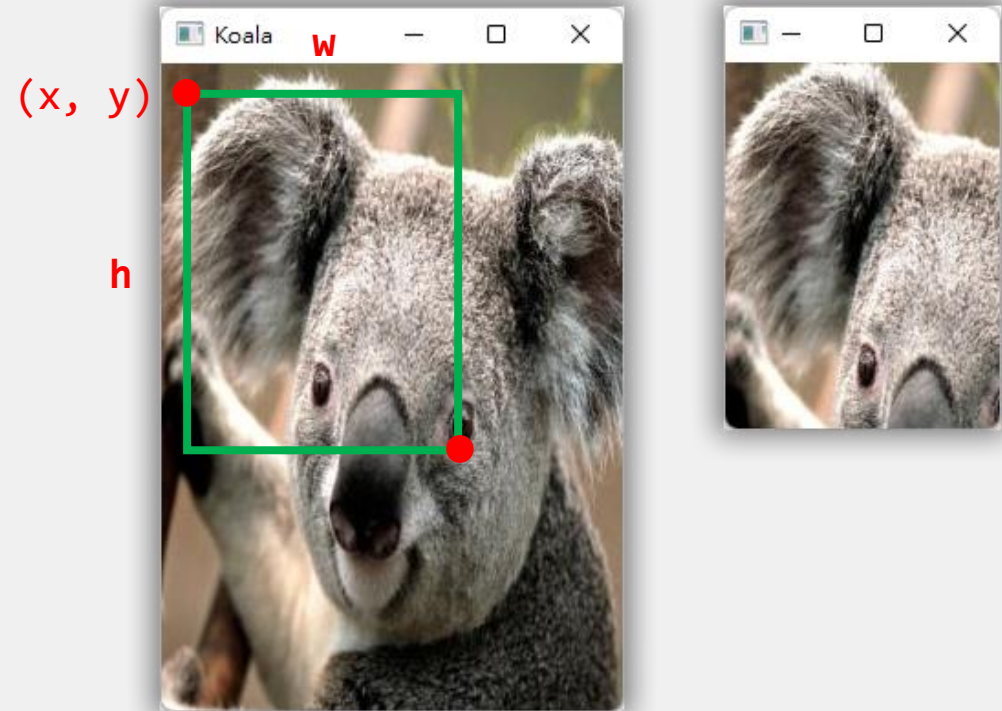
- 在 OpenCV 使用 `imread()` 方法讀取的圖片內容是一個 Numpy 陣列，剪裁圖片就是在切割 Numpy 陣列，如下所示：

## 1-1c 剪裁圖片。(1-1c.py)

```
import cv2

img = cv2.imread("koala.jpg")
print(img.shape)
x = 10; y = 10
w = 150; h= 200
crop_img = img[y:y + h, x:x + w]
cv2.imshow("Koala", img)
cv2.imshow("corp_Koala", crop_img)
print(crop_img.shape)

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



# 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  $0 \sim 255$ 。
  - ✓ 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  $256 \times 256 \times 256 = 16,777,216$ ，這麼多的色彩數。



354

252

$$img = \begin{bmatrix} [65, 91, 108] & \dots \\ \dots & \dots \end{bmatrix}_{354 \times 252}$$

Name	Type	Size	Value
crop_img	Array of uint8	(200, 150, 3)	$\begin{bmatrix} [63, 83, 108] \\ [65, 86, 108] \end{bmatrix}$
h	int	1	200
img	Array of uint8	(354, 252, 3)	$\begin{bmatrix} [65, 91, 108] \\ [65, 89, 106] \end{bmatrix}$
w	int	1	150
x	int	1	10
y	int	1	10

# 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。

## 1-1d 剪裁圖片。(1-1d.py)

```
import cv2, imutils

img = cv2.imread("koala.jpg")
rotated_img = imutils.rotate(img, angle = 90)
fliped_img = cv2.flip(img, -1)
translated_img = imutils.translate(img, 25, -75)
cv2.imshow("Koala:rotated", rotated_img)
cv2.imshow("Koala:fliped", fliped_img)
cv2.imshow("Koala: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）。

**1-1e**

轉換成灰階和 BGR 圖片。(1-1e.py)

```
import cv2

img = cv2.imread("koala.jpg")
gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
bgr_img = cv2.cvtColor(img, cv2.COLOR_RGB2BGR)
cv2.imshow("Koala:gray", gray_img)
cv2.imshow("Koala:bgr", bgr_img)

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

`imread()` 與 `imshow()` 方法都是使用 BGR 格式，所以不需要特殊處理。

# 從 URL 取得圖片

- 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，如下所示：

## 1-1f 從 URL 取得圖片。(1-1f.py)

```
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線、畫長方形、畫圓形、畫橢圓形和加上文字內容，如下所示：

## 1-1g 註記圖片。(1-1g.py)

```
import cv2

img = cv2.imread("koala.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("Koala", 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，如下所示：

## 1-1h 寫入圖片。(1-1h.py)

```
import cv2

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



# 1-3 OpenCV 的基本使用

---

1-3-1 OpenCV 圖片處理

1-3-2 OpenCV 影片處理

1-3-3 OpenCV 網路攝影機操作

# 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圖片所組成，每一個靜態影像稱為『影格』(Frame，或稱幀)，每秒播放的靜態影像圖片數稱為『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。

## 1-2 播放影片檔。(1-2.py)

```
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()
```

# 取得影片資訊

- 在 Python 程式建立 VideoCapture 物件後，我們可以取得影片的尺寸和編碼的影片資訊，如下所示：

## 1-2a 取得影片資訊。(1-2a.py)

```
import cv2

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

def decode_fourcc(v):
    v = int(v)
    return "".join([chr((v >> 8 * i) & 0xFF) for i in range(4)])

width = cap.get(cv2.CAP_PROP_FRAME_WIDTH)
height = cap.get(cv2.CAP_PROP_FRAME_HEIGHT)
print("圖片尺寸:", width, 'x', height)
fourcc = cap.get(cv2.CAP_PROP_FOURCC) →
codec = decode_fourcc(fourcc)
print("Codec編碼:", codec)
```

參數用法：

[https://docs.opencv.org/4.5.3/d4/d15/group\\_\\_videoio\\_\\_flags\\_\\_base.html#baeb8dd9c89c10a5c63c139bf7c4f5704d](https://docs.opencv.org/4.5.3/d4/d15/group__videoio__flags__base.html#baeb8dd9c89c10a5c63c139bf7c4f5704d)。

# 影片處理顯示灰階影片

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

## 1-2b 影片處理顯示灰階影片。(1-2b.py)

```
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()
```

# 取得網路攝影機的影像

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

## 1-3 取得網路攝影機的影像。(1-3.py)

```
import cv2

cap = cv2.VideoCapture(0)

while cap.isOpened():
    ret, frame = cap.read()
    frame = cv2.rotate(frame, rotateCode = 1)
    cv2.imshow("frame", frame)

    if cv2.waitKey(1) == 27:
        break

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

# 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：

## 1-3a 更改影像的解析度。(1-3a.py)

```
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 VideoWriter 物件來寫入影片檔案：

## 1-3b 將影像寫入影片檔案。(1-3b.py)

```
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()
```

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



# Outline

---

- 1-1 AI 的主流技術：機器學習
- 1-2 在 Windows 上安裝 OpenCV
- 1-3 OpenCV 的基本使用
- 1-4 OpenCV 人臉辨識**

## OpenCV 人臉辨識

目的：OpenCV 內建演算法進行人臉辨識

## 1-4 人臉辨識

- OpenCV 內建的是哈爾特徵 (Haar-like feature) 演算法。
  - ✓ 速度快，非類神經網路，適合低階的樹莓派。
  - ✓ 使用哈爾特徵前，先下載人臉偵測的聯集分類器。
- 分類器下載
  - ✓ <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

# 圖片的人臉辨識

## 1-4 圖片的人臉辨識。(1-4.py)

```
import cv2

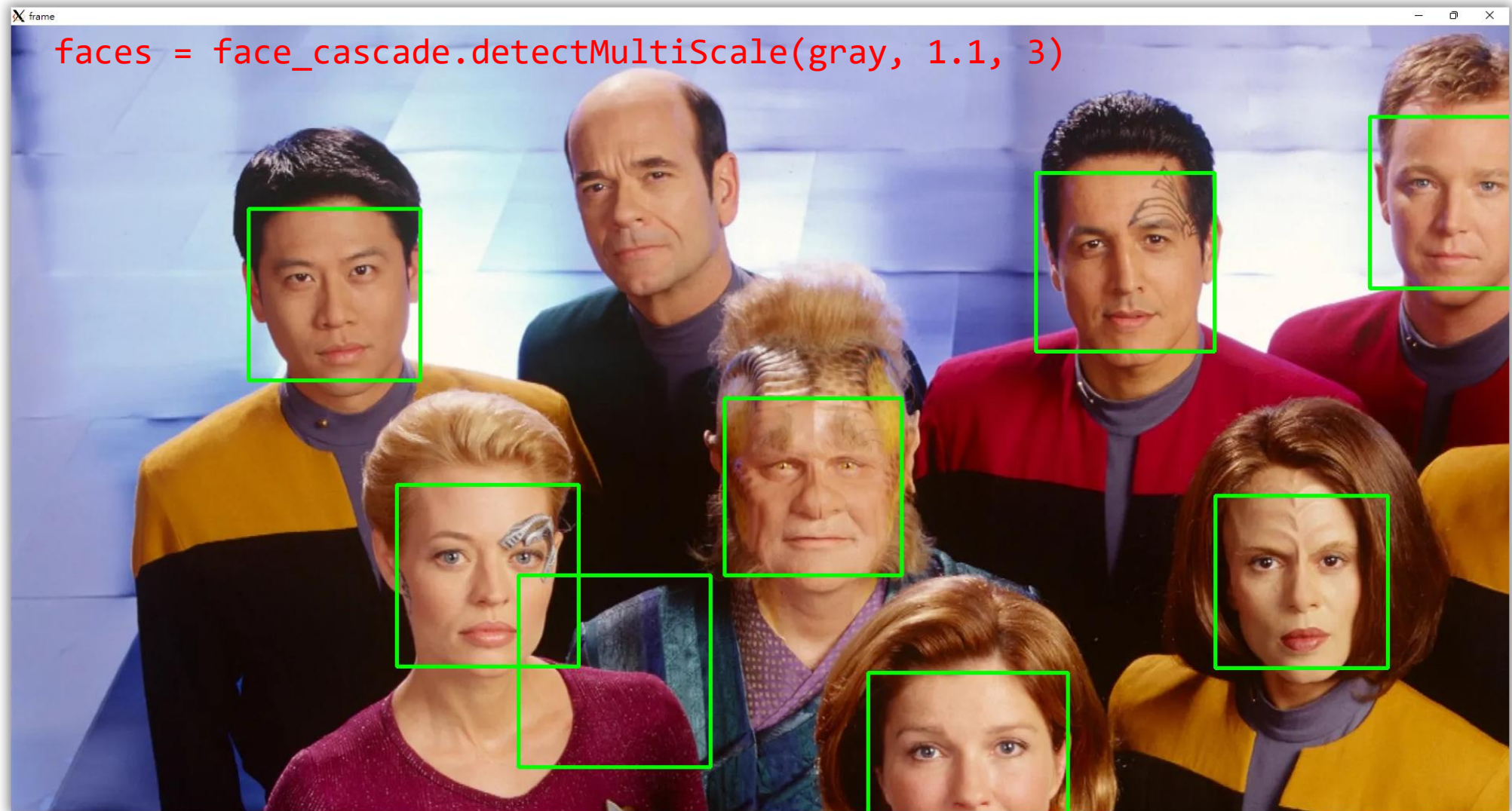
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_alt2.xml')
image = cv2.imread('demo.jpeg')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

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)

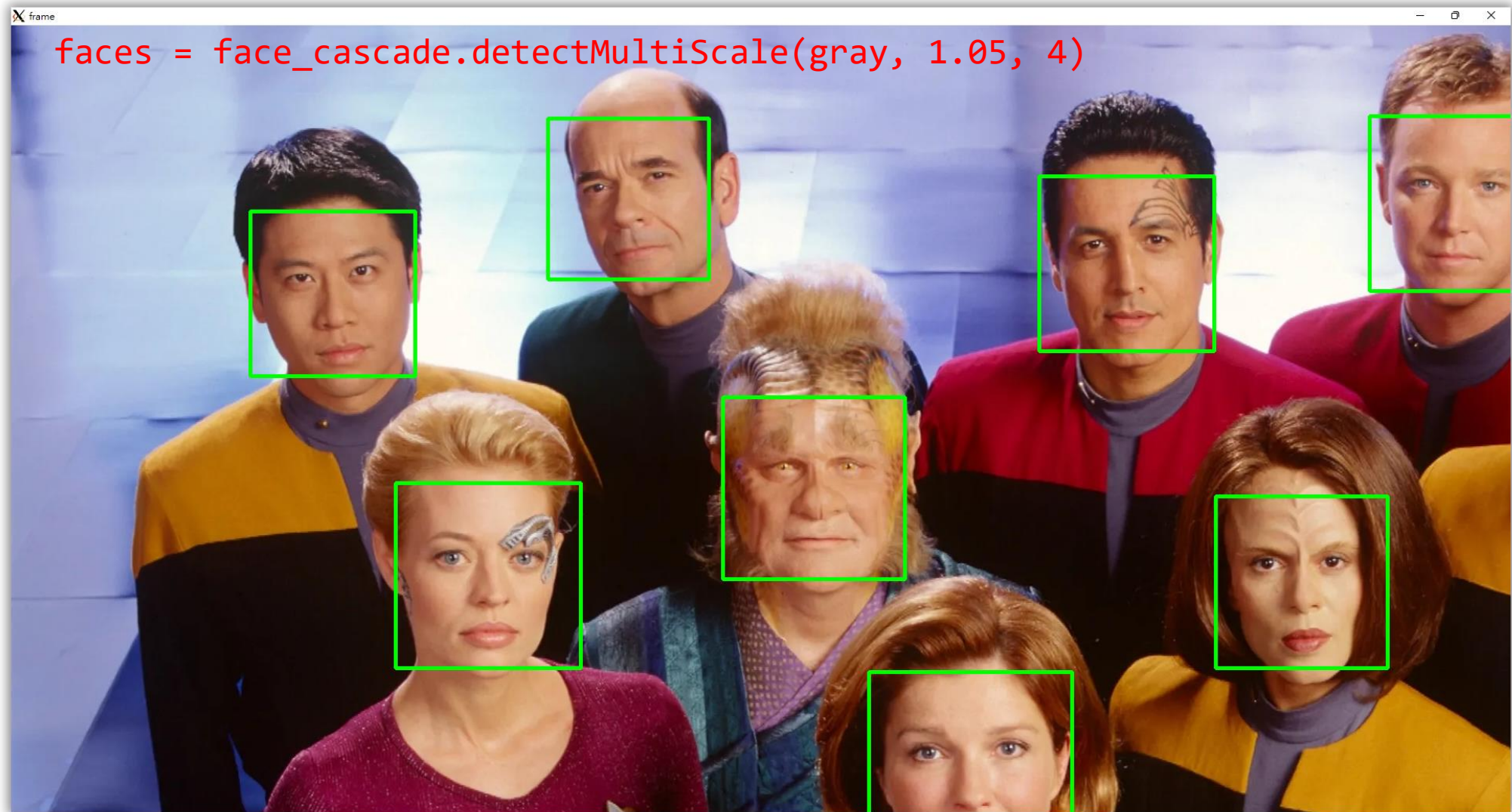
cv2.namedWindow('frame', cv2.WINDOW_NORMAL)
cv2.imshow('frame', frame)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

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

# 當有人臉沒被辨識，需要調整參數 (1-4.py)



# 調整參數後 (1-4a.py)



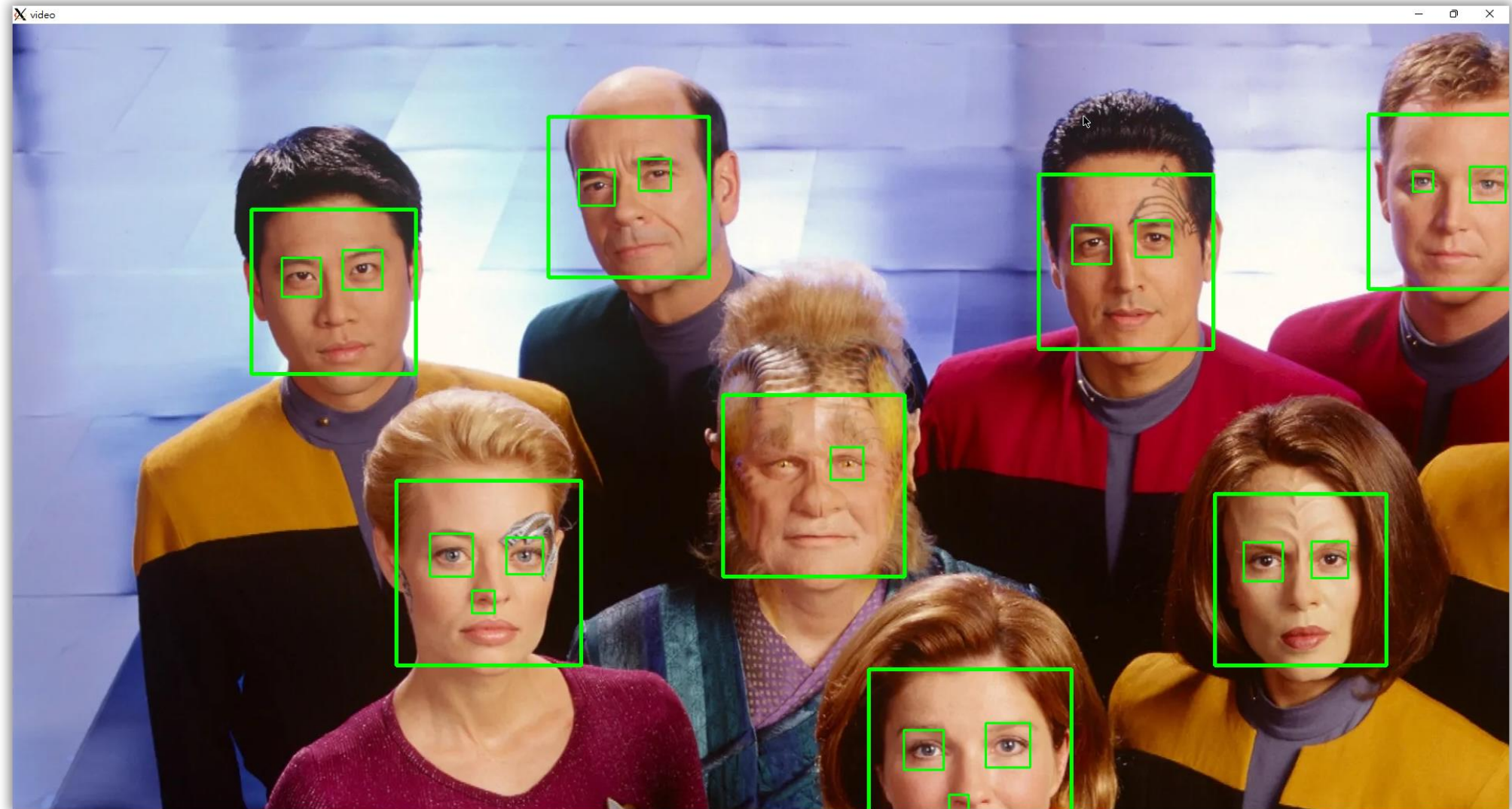
# 圖片的眼睛辨識

## 1-5 圖片人臉與眼睛辨識。(1-5.py)

```
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()
```

# Demo 1-5.py





# OpenCV 即時影像人臉辨識

目的：進行即時人臉辨識

# 即時影像的人臉辨識

1-6

即時影像的人臉辨識。(1-6.py)

```
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

1-7

即時影像的人臉與眼睛辨識。(1-7.py)

```
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()
```

## 實驗 4-3：特定人臉辨識

目的：OpenCV 進行特定人臉辨識

# 特定人臉偵測

---

- 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。
  - ✓ 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。
  - ✓ 訓練：OpenCV 提供三種演算法，Eigen、Fisher、LBPH。
  - ✓ 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。

# Step 1：取樣

---

- 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捷。

# 取樣程式

## 1-8-1 人臉取樣。(1-8-1-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)

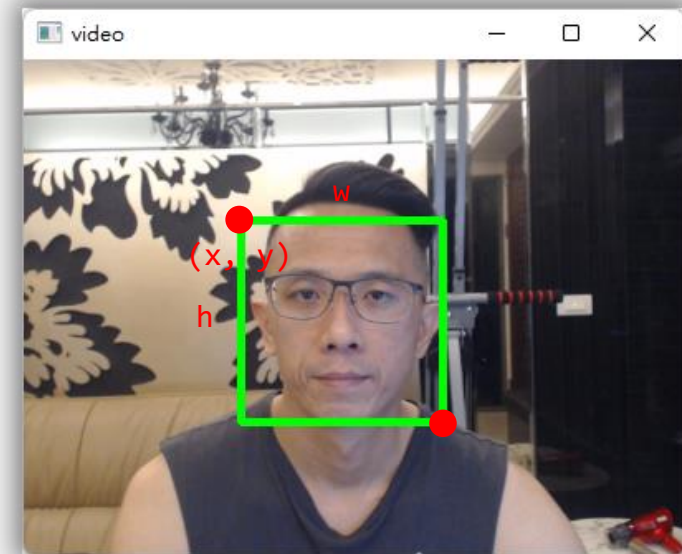
faces = face_cascade.detectMultiScale(gray, 1.1, 3)

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

```





## Step 2：訓練

---

- 訓練階段：將  $h_0$ 、 $h_1$ ... 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。

## 1-8-2 訓練取樣集。(1-8-2-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')
```

# 建立模型，使用 LBPH 演算法。  
# train() 只能接收 numpy 格式的陣列。  
# 儲存訓練好的辨識檔。

## Step 3：辨識

---

- 辨識階段：這一階段是拿訓練好的結果，來驗證辨識效果是否準確，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。

## 1-8-3 人臉辨識。(1-8-3-recognition.py)

```
import cv2

model = cv2.face.LBPHFaceRecognizer_create() # 建立模型，使用 LBPH 演算法。
model.read('faces.data') # 載入訓練好的辨識檔。
print('load training data done')
# 載入聯集分類器（需與取樣時的分類器一致），並開啟攝影機。
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
cap = cv2.VideoCapture(0)
cv2.namedWindow('video', cv2.WINDOW_NORMAL)
# 可識別化名稱
names = ['pilai']
# 讀取攝影機資料，並轉換成灰階影像。
while True:
    ret, frame = cap.read()
    frame = cv2.resize(frame, (600, 400))
    frame = cv2.flip(frame, 1)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

```

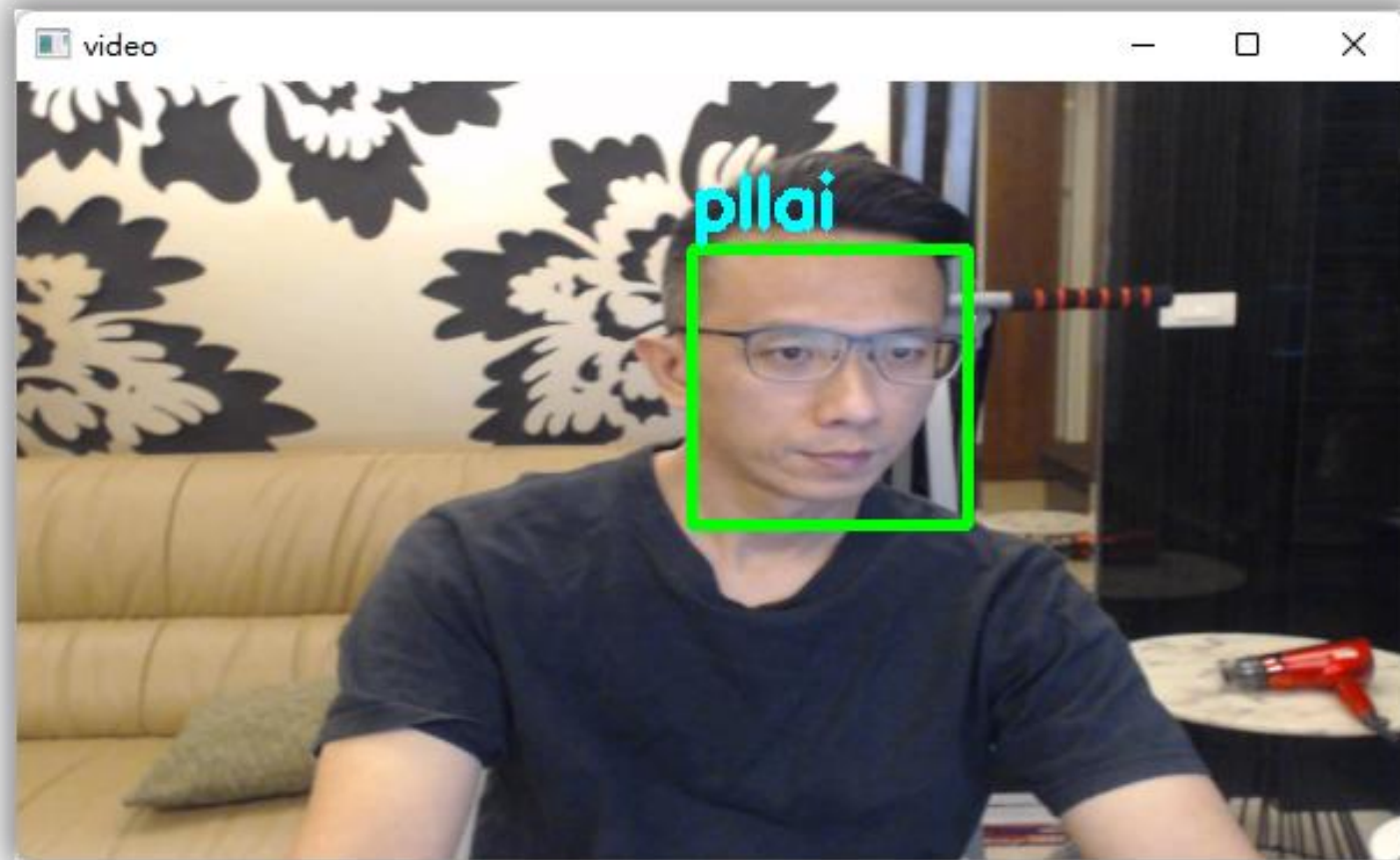
faces = face_cascade.detectMultiScale(gray, 1.1, 3)
for (x, y, w, h) in faces:
    frame = cv2.rectangle(
        frame,
        (x, y), (x + w, y + h),
        (0, 255, 0), 3
    )
    face_img = gray[y: y + h, x: x + w]
    face_img = cv2.resize(face_img, (400, 400))

    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

```

# Demo 特定人臉辨識



整合專案：加入多個特定人臉辨識

Demo

project\_multiFace

請參考資料夾內的程式碼