

Day 64

OPENCV

# Face Detection by OPENCV簡介




# 本日知識點目標

---



目標  
知識點

學習在OpenCV 使用face cascade



獲得  
知識點

完成今日課程後你應該可以了解

- OPENCV的 python lib: cv2
- 學習使用使用face cascade 進行人臉辨識

# OPENCV 初始

---

- OpenCV全名是Open Source Computer Vision Library，是一個影像處理函式庫，由Intel發起並參與開發，以BSD授權條款發行，可在商業和研究領域中免費使用，目前是非營利的基金組織[OpenCV.org](http://OpenCV.org)在維護
- 讀取一副圖像，根據圖元的行和列的座標獲取它的圖元值，對於RGB圖像而言，返回RGB的值，對於灰度圖則返回灰度值

```
import cv2
import numpy
img = cv2.imread('home.jpg')
px=img[100,100]
print(px)
```

# 說明

---

- 使用工具Python
- 使用庫numpy;opencv
- 從圖片的導入到另存
- 1.cv2.imread(文件名,標記)讀入圖像
  - cv2.IMREAD\_COLOR(): 讀入彩色圖像
  - cv2.IMREAD\_GRAYSCALE(): 以灰度模式讀入圖像
- 2.cv2.imshow()顯示圖像
  - cv2.waitKey()等待鍵盤輸入，為毫秒級
  - cv2.destroyAllWindows()可以輕易刪除任何我們建立的窗口，括號內輸入想刪除的窗口名



# 範例

---

- 獲取圖像屬性

圖像屬性包括：行，列，通道，  
圖像數據類型，像素數目等

- `img.shape`可以獲得圖像的形狀，  
返回值是一個包含行數，列數，  
通道數的元組

```
import cv2  
  
import numpy  
  
img = cv2.imread('home.jpg')  
print(img.shape)
```

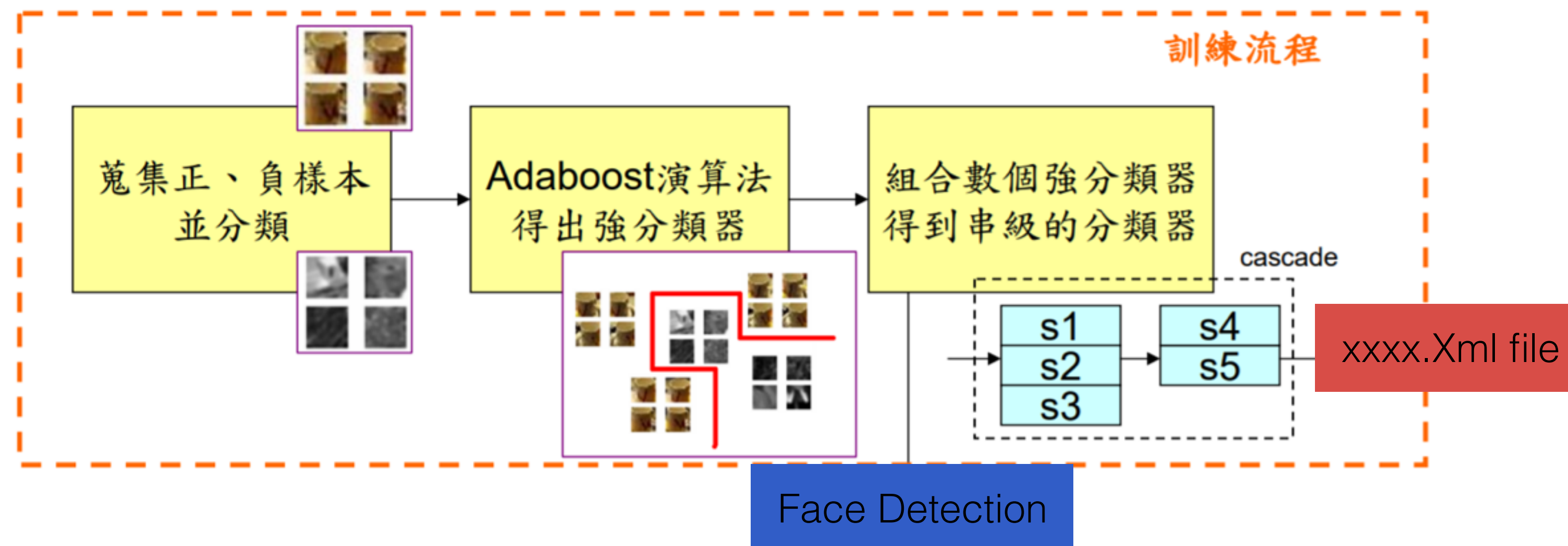
# Face Detection

---

- `detectMultiScale (image , scaleFactor , minNeighbors)` :
- 這是一個檢測對象的通用函數，在這種情況下，它會檢測面，因為我們調用了face cascade。如果找到一個面，它將以“Rect (x , y , w , h)”的形式返回所述面的位置列表，如果沒有，則返回「None」。
  - (1)image：第一個輸入是灰度圖像。因此，請確保圖像為灰度。
  - (2)scaleFactor：這個函數補償了當一個面看起來比另一個面更大時發生的錯誤感覺，因為它更接近相機。
  - (3)minNeighbors：這是一種使用移動窗口來檢測對象的檢測算法，它通過在聲明找到的面部之前定義在當前對象附近找到多少個對象來實現。

# Face Detection - face cascade

- 採用Haar分類器
- Haar分類器實際上是Boosting演算法的一個應用，Haar分類器用到了Boosting演算法中的AdaBoost演算法，只是把AdaBoost演算法訓練出的強分類器進行了級聯，並且在底層的特徵提取中採用了高效率的矩形特徵和積分圖方法



# 前述流程 / python程式 對照

---

- python 程式 (請參閱今日範例)

```
# cv2.cvtColor is an OpenCV function to convert images to different color spaces  
def convertToRGB(img):  
    return cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

```
def detect_faces(f_cascade, colored_img, scaleFactor = 1.1):  
    #just making a copy of image passed, so that passed image is not changed  
    img_copy = colored_img.copy()  
  
    #convert the test image to gray image as opencv face detector expects gray images  
    gray = cv2.cvtColor(img_copy, cv2.COLOR_BGR2GRAY)  
  
    #let's detect multiscale (some images may be closer to camera than others) images  
    faces = f_cascade.detectMultiScale(gray, scaleFactor=scaleFactor, minNeighbors=5);  
  
    #go over list of faces and draw them as rectangles on original colored img  
    for (x, y, w, h) in faces:  
        cv2.rectangle(img_copy, (x, y), (x+w, y+h), (0, 255, 0), 2)  
  
    return img_copy
```



# 前述流程 / python程式 對照

---

```
#load cascade classifier training file for haarcascade
haar_face_cascade = cv2.CascadeClassifier('data/haarcascade_frontalface_alt.xml')

faces = haar_face_cascade.detectMultiScale(gray_img, scaleFactor=1.1, minNeighbors=5);

#go over list of faces and draw them as rectangles on original colored
for (x, y, w, h) in faces:
    cv2.rectangle(test1, (x, y), (x+w, y+h), (0, 255, 0), 2)
    #let's detect multiscale (some images may be closer to camera than others) images
#print the number of faces found
print('Faces found: ', len(faces))
```

Faces found: 2

# 重要知識點複習

---

- 使用工具Python，  
使用庫numpy；opencv，  
從圖片的導入到另存，
- `cv2.imread(文件名，標記)`讀入圖像，
- `cv2.IMREAD_COLOR ()`：讀入彩色圖像
- `cv2.IMREAD_GRAYSCALE ()`：以灰度模式讀入圖像
- `cv2.imshow()`顯示圖像，
  - \* `cv2.waitKey()`等待鍵盤輸入，為毫秒級
  - \* `cv2.destroyAllWindows()`可以輕易刪除任何我們建立的窗口，括號內輸入想刪除的窗口名
- 導入使用的模組
- `import cv2`
- `import numpy`

# 解題時間 It's Your Turn

請跳出PDF至官網Sample Code & 作業  
開始解題

