

# LAB：行人偵測專題

---

實驗模組作者：沈思鎧、高效能計算實驗室

實驗模組改寫：洪祐鈞、高效能計算實驗室

若是做了有問題，可以email聯絡課程助教。

## 實驗目標：

---

- 本實驗透過逐步概念的解釋，讓同學實作出基本的行人偵測系統：利用OpenCV取得影像資訊 → 圖像處理 → 行人偵測
- 學習電腦視覺處理技巧、了解影像基本操作。
- 如何於Raspberry Pi上使用OpenCV。

## 術語：

---

本機：你自己的電腦，會和樹莓派遠端連線

## 需求：

---

- 請先閱讀本文檔，了解要做的事情是什麼，有可能有些步驟你不需要也可以達到目標。
- 本機是Linux系統，強烈建議直接安裝Linux系統 (不透過虛擬化的軟硬體技術)。
- 使用Windows或是WSL的理論上可以行得通，但必須理解整個實驗模組在做的事情，並且自行尋找解決方法。

## 需要材料：

---

- 樹莓派
- SD卡
- SD卡讀卡機
- WiFi 環境
- 電腦螢幕、HDMI線

## 設置：

### 安裝樹莓派映像檔

1. 先安裝Raspberry Pi Imager進本機：

下載連結：[\(https://www.raspberrypi.com/software/\)](https://www.raspberrypi.com/software/)

2. 下載完後，將SD卡插進本機。

Operating System點選erase

Storage點選你的SD卡，將SD卡內的資料格式化

(注意！所有SD卡裡面的資料將會抹除，若是之前有未完成的實驗模組請先完成或備份！！)



3. 格式化完成之後，請為你的Raspberry Pi做設定：

Set Hostname: pi

勾選enable ssh -> Use password authentication

Username: pi

Password: raspberry

填入WiFi帳號密碼，儲存。

4. 之後安裝OS

Operating System點選Raspberry PI OS (64-bit)

Storage點選你的SD卡

5. 完成之後就可以把樹莓派接到HDMI去測試是否成功。

## SSH遠端連線

### 連接WiFi

儘管設置了WiFi密碼，可能還是會有設置WiFi地區的錯誤，這個時候你需要接上電腦螢幕，點選桌面右上的網路去設置區域，並且重開機重新輸入WiFi帳密，這樣應該就可以成功連接WiFi了。

### 取得WiFi ip

在樹莓派終端機上面可以打 `ifconfig` 以得到樹莓派ip位址

會輸出：

```
...
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.182.209  netmask 255.255.255.0  broadcast 192.168.182.255
        inet6 2401:e180:8863:8e3:e0d0:7632:cb2d:81fd  prefixlen 64  scopeid
0x0<global>
...
```

樹莓派的ip位址會在 `inet 192.168.xxx.xxx`

---

如果在本機上你需要和你的樹莓派共處一個網路

在本機的終端機上面打上

```
sudo nmap -sn <本機ip位址>/24
```

也可以看到樹莓派ip

### 連接SSH

在本機上，以SSH連接樹莓派指令，其中 `-X` 代表X11 forwarding將樹莓派上運行的圖形界面顯示到本機上：

```
ssh -X pi@192.168.xxx.xxx
```

輸入之前設過的密碼：`raspberrypi`

就可以本機連上樹莓派了，而這代表的意義是接下來的指令都可以在本機上ssh的終端機對樹莓派做操作！

## sftp

若是可以使用ssh遠端登入，那就可以在本機"另"開終端機，以sftp互傳檔案。

請自行學習sftp指令，以找到相對應的目錄並將 `pedestrian_detection` 之檔案傳到樹莓派。

指令大約是：`cd`, `lcd`, `ls`, `lls`, `put`, `get`

以下只是重點提示：

1. sftp連線到樹莓派

```
sftp pi@192.168.xxx.xxx
```

2. 使用put指令將本機的檔案傳到樹莓派。

```
put -r ./pedestrian_detection
```

## 設置環境及安裝套件

在樹莓派上安裝：

1. 設置虛擬環境

樹莓派有預設安裝Python，現在可以藉由虛擬環境來隔離環境，對於佈署Python環境很有幫助，使用 `which python` 可以看輸出的python在哪個環境執行。

```
$ sudo apt install python3      # install Python3
$ which python3
$ python -m venv env            # 創建一個名為env的虛擬環境
$ source env/bin/activate      # 啟用虛擬環境
$ which python
```

2. 安裝必要套件

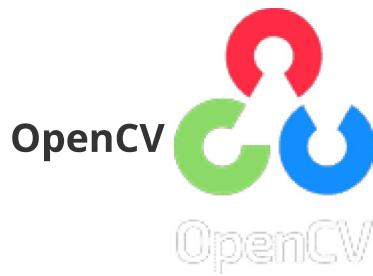
```
pip install opencv-python imutils
```

```
(xxx) pschen@LAPTOP-5808QBSI:~$ pip install opencv-python imutils
Collecting opencv-python
  Downloading opencv_python-4.9.0.80-cp37-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (62.2 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 9.2/62.2 MB 1.9 MB/s eta 0:00:29
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.2/62.2 MB 2.1 MB/s eta 0:00:00
Collecting imutils
  Downloading imutils-0.5.4.tar.gz (17 kB)
  Preparing metadata (setup.py) ... done
Collecting numpy>=1.21.2
  Downloading numpy-1.26.4-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (18.3 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 10.3/18.3 MB 1.9 MB/s eta 0:00:05
```

3. 測試

```
python test_opencv.py
```

如果終端機出現文字圖片就代表你應該安裝完成了。



- **Open Source Computer Vision Library** (since 1999)
- Intel公司發起開發，Apache 2 License (It is free for commercial use)
- OpenCV is the world's biggest computer vision library. It's open source, contains over 2500 algorithms and is operated by the non-profit Open Source Vision Foundation.
- 可以製做圖片、視訊、矩陣運算、統計、圖論、資料儲存的函式庫，支援C/C++、Python、Java等程式語言。
- 可以整合不同圖檔格式的矩陣運算，應用在靜態圖片(例如: BMP, JPG, TIFF, PNG)，動態Webcam的影像處理。
- 相關的領域為: 影像處理、電腦視覺、圖形識別、電腦圖學、資訊檢索或遊戲設計。
- 比較有名的應用: 物體追蹤、人臉辨識、傅立葉轉換、紋理分析。

## 圖像基本操作

請參照 `processing.py`

### 1. 讀取圖像

從path這個路徑讀取圖片到image

```
image = cv2.imread("path")
```

### 2. 展示圖像

跳出視窗，標題為GUI title, 顯示image物件的圖片於視窗。

```
cv2.imshow("GUI title", image)
```

程式最後記得 `cv2.waitKey(0)`，以等待鍵盤輸入暫停程式，要不然視窗會跳掉。

若是遠端連線 `ssh -x` 還是沒有東西在本機出現，可能x11出了問題，建議使用下方的 `cv2.imwrite()` 儲存圖像後，搭配 `sftp` 把所儲存的圖像結果傳回來檢視。

### 3. 儲存圖像

儲存img圖片到路徑path

```
...  
cv2.imwrite('path', img)  
...
```

#### 4. 縮放圖像

將圖片縮放為 400 \* 400

```
resized_image = cv2.resize(image, (400, 400))
```

#### 5. 感興趣區域 (ROI, Region Of Interest)

將座標點 x1, x2, y1, y2 區域的圖片取出

```
roi = resized[y1:y2, x1:x2]  
cv2.imshow('ROI', roi)
```

#### 6. 畫長方形

畫一個長方形於img圖片上，其座標為(x1, y1), (x2, y2)長方形，其線寬為2，顏色為綠色(b, g, r)

```
cv2.rectangle(img, (x1, y1), (x2, y2), (0, 255, 0), 2)
```

#### 7. 圖像轉灰階

將image圖片轉為gray灰階圖片。

```
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

#### 8. CLAHE

CLAHE (Contrast Limited Adaptive Histogram Equalization) 為限制對比度自適應直方圖均衡，藉由重新分配直方圖來作到加強對比的效果。(使用單通道照片為輸入)

```
clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))  
clahe_img = clahe.apply(gray)
```

## 影片操作

請參照 `video_processing.py`

#### 1. 讀取圖片並抓取每一個frame

首先使用 `VideoCapture` 創建cap物件，抓取路徑path的影片。

接著使用 `cap.read` 來取得frame

按下q會跳出視窗。

```
cap = cv2.VideoCapture(path)
while(cap.isOpened()):
    ret, frame = cap.read()
    if ret == True:
        cv2.imshow(frame)
        if cv2.waitKey(16) & 0xFF == ord("q"):
            break
    else:
        break
```

## 2. 儲存圖片：

使用 `Videowriter` 來設定寫影片的物件 `out`。設定使用 `xvid` codec，寫入尺寸為 `640 * 360`，其 `fps` 為 `10.0`，寫入至 `path` 路徑。

並且使用 `out.write` 來將抓到的 `frame` 寫入指定位置。

```
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('path', fourcc, 10.0, (640, 360))

...

out.write(frame)
```

## 視訊鏡頭相關

### 測試鏡頭

安裝 `fswebcam`，然後拍一張 `1280*720`，檔名為 `image.jpg` 的圖片。

```
$ sudo apt install fswebcam
$ fswebcam -r 1280x720 --no-banner ./image.jpg
```

將會產生 `image.jpg` 圖片。

## OpenCV

參照 `webcam.py`

直接給 `0` 就會抓取預設第 `0` 個裝置：

```
cap = cv2.VideoCapture(0)
```

或者使用 `v4l2-ctl --list-devices`

其出現的 `/dev/video0` 等等的路徑。

然後再 `cap = cv2.VideoCapture('/dev/video0')`

就可以和影片一樣抓取每個frame的資料了

## 行人偵測

video\_processing.py和webcam.py都是使用級聯分類器 (Cascade Classifier)，讀取預先訓練的xml檔案來取得匡選的區域：

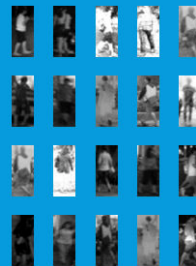
```
classifier = cv2.CascadeClassifier('xml_file_path')
objs = classifier.detectMultiScale(gray, 1.3, 5)
```

之後遍歷回傳的objs就可以取得偵測結果。

## 行人偵測: 利用ADABOOST演算法

### 訓練分類器

- 蒐集正負樣本
  - 樣本目的為找出行人特徵值的範圍
  - 正樣本為行人圖像
  - 負樣本為任何不包含行人特徵的圖像
  - 尺寸必須統一



正樣本



負樣本



## WEAK CLASSIFIER

- 每一個樣本特徵形成一個弱分類器(weak classifier)規則
- 樣本經由一種特徵值做分類，在正特徵值範圍內的被判定為目標物，反之則否。在負樣本特徵值範圍內的判定為非目標物。再經由**Adaboost**做樣本權重調整

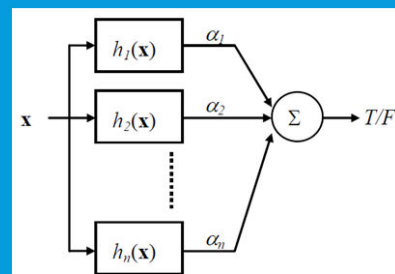
Ref. 鄭文昌, 鄭永義, and 廖建倫,  
“即時行人偵測與追蹤嵌入式系統實作”

18

## STRONG CLASSIFIER

- 強分類器(Strong Classifier)是由數個弱分類器(Weak Classifier)所組成的，可分線性組成或樹狀組成。
- 每個分類器都可視為一個函式， $f(x)$ 為強分類器， $h_t(x)$ 為弱分類器 $\alpha_t$ 為每回合權重
- 強分類器示意圖

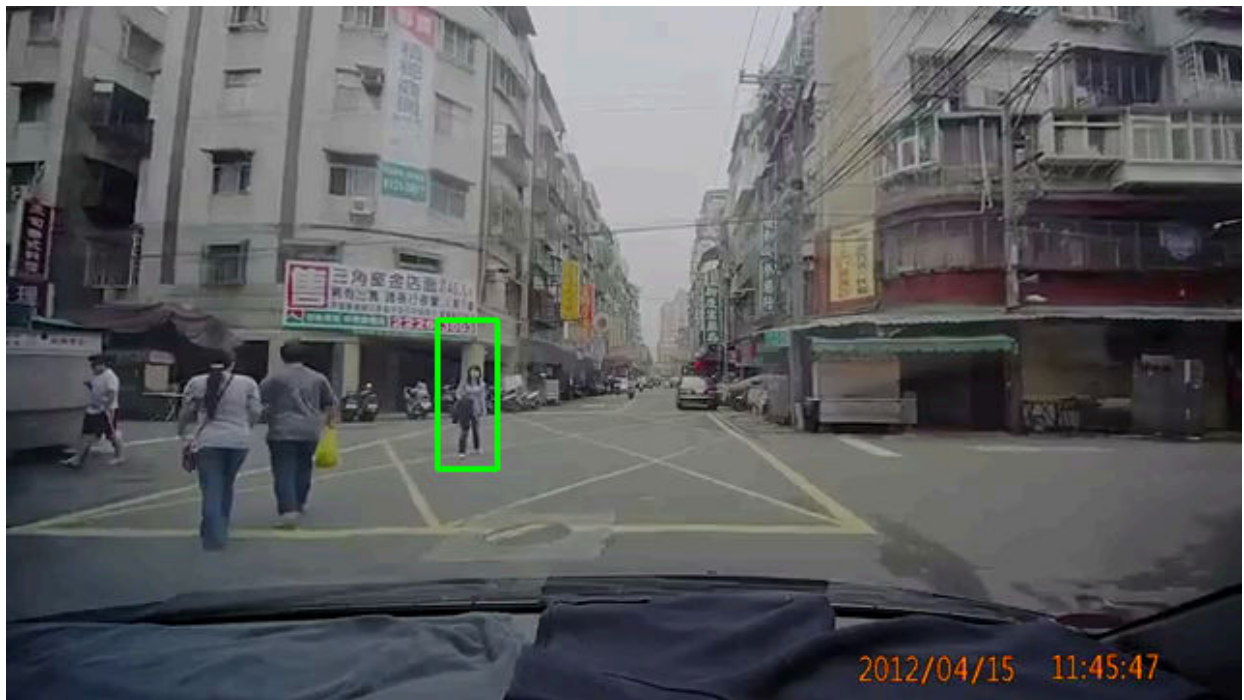
$$f(x) = \sum_{t=1}^T \alpha_t h_t(x)$$



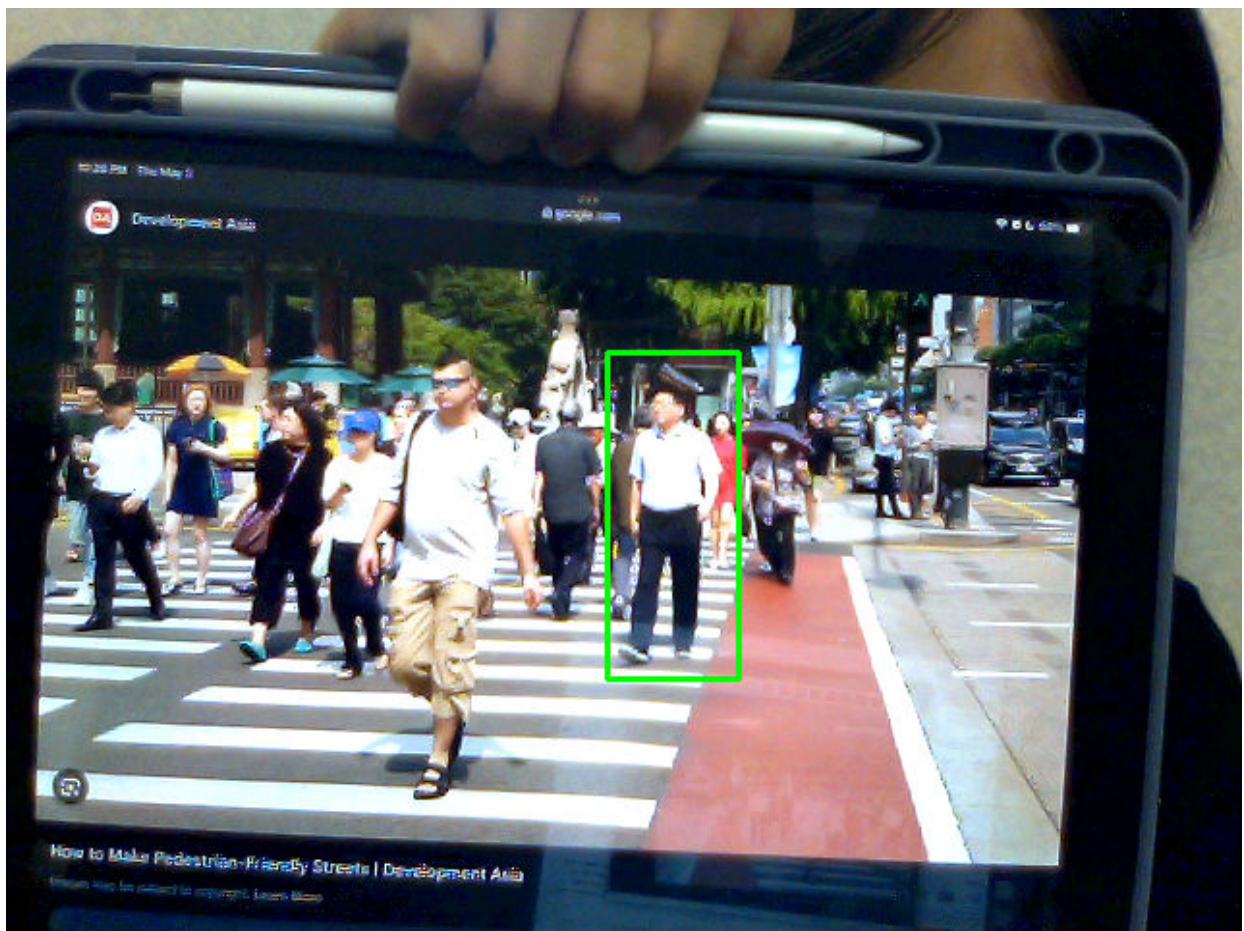
Ref. 鄭文昌, 鄭永義, and 廖建倫,  
“即時行人偵測與追蹤嵌入式系統實作”

19

video\_processing.py:



webcam.py:



## 實驗DEMO

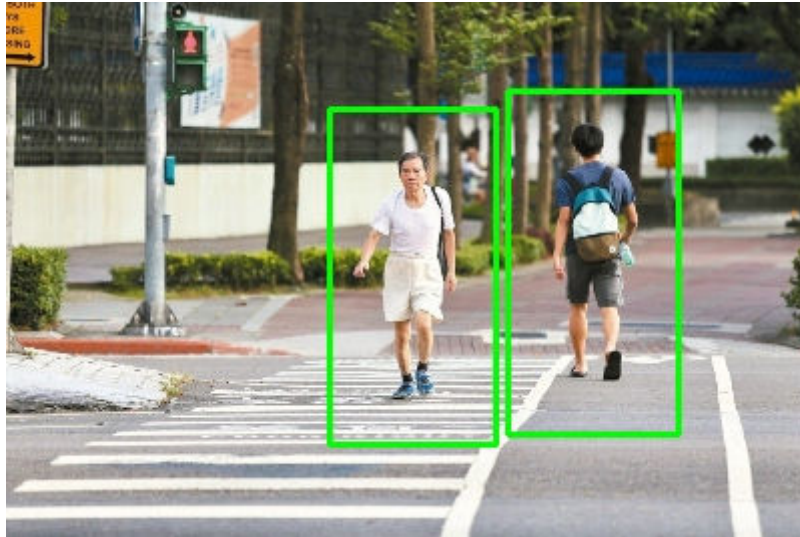
繳交程式碼，和輸入指令的終端機需要有raspberry的帳號名稱截圖。(必須在樹莓派上面跑！)

### 1. 完成sample.py的填空

1. 將圖片縮放為寬400，等比例縮放的圖片 (30%)

2. 於該照片上框出所有的偵測目標 (30%)

其結果應該如：



2. sample.py 使用HOG algorithm來偵測行人，請簡單說明HOG algorithm的偵測原理 (10%)

3. 根據實驗模組的教學做延伸應用 (20%)

4. 心得 (10%)