

# 行人偵測系統 (Raspberry Pi)

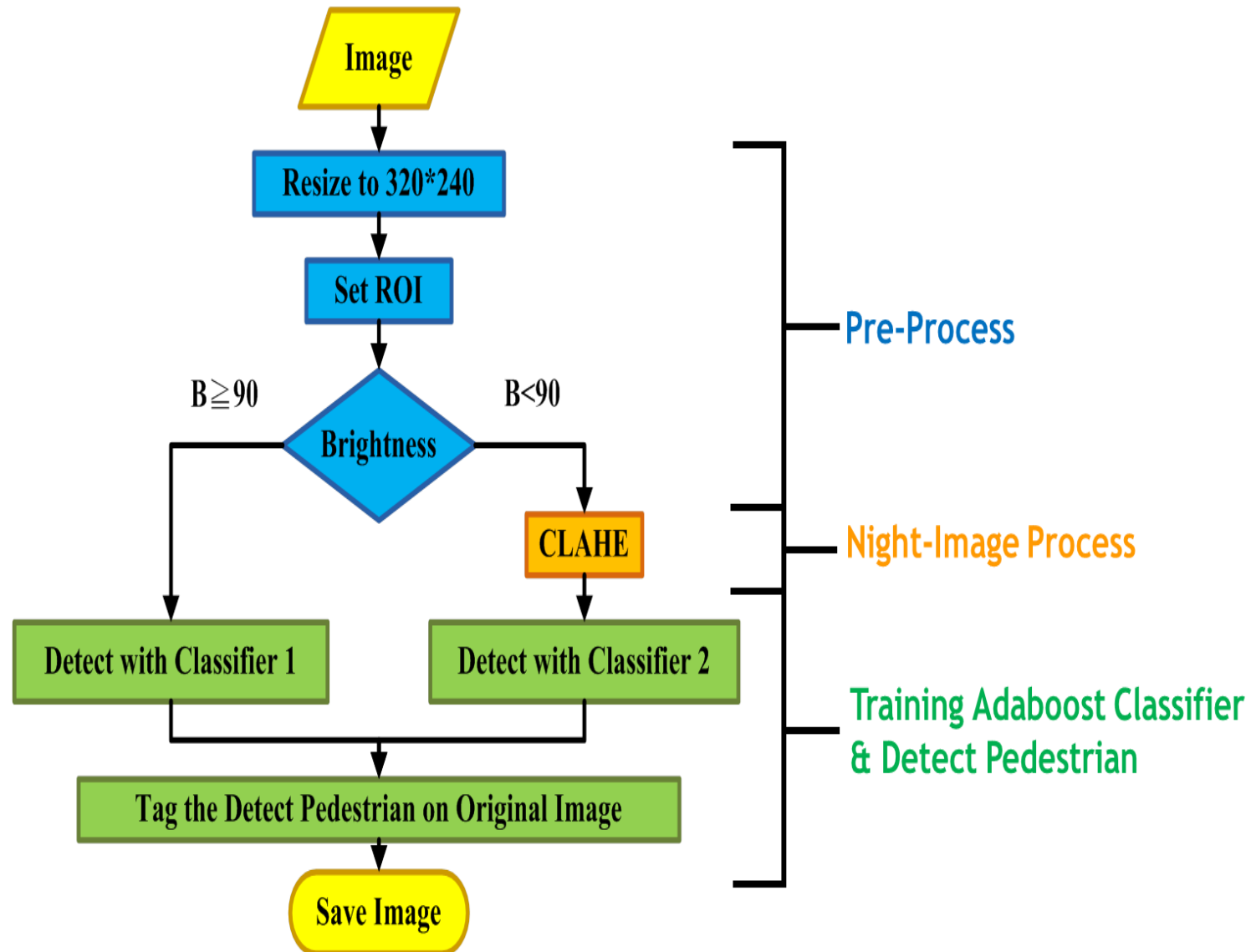
# OUTLINE

- 實驗目標
- OpenCV介紹
- 軟硬體需求
- 開發環境準備
- 圖片處理
- 影像前置處理
- 夜間影像處理
- 人形偵測系統
- 參考網站

# 實驗目標

- 實驗說明：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 → 圖像前置處理 → 影像前置處理  
→ 夜間影像之處理(CLAHE) → 行人偵測
- 實驗目的：
  1. 學習電腦視覺處理技巧
  2. 了解影像基本操作
  3. 如何將OpenCV運用在Raspberry pi上

# SYSTEM ARCHITECTURE



# OPENCV介紹



- **Open Source Computer Vision Library**
- Intel公司發起開發，BSD license
- 可以製做圖片、視訊、矩陣運算、統計、圖論、資料儲存的相關C語言函式庫
- 可以整合不同圖檔格式的矩陣運算
- 應用在靜態圖片(BMP,JPG,TIF,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

# 軟硬體需求

- 硬體準備：

1. Raspberry PI 3 (樹莓派)
2. Micro SD 卡 - 至少 2GB 以上因為要安裝作業系統
3. HDMI 線
4. Micro USB 線 跟 5V / 2A 的電源供應器
5. USB 滑鼠 與 鍵盤
6. Webcam

- 軟體準備：

1. 作業系統-從 [Raspberry PI](https://www.raspberrypi.org/downloads/noobs/) 官網 下載 NOOBS 壓縮檔。
2. OpenCV

# 開發環境建置

## · 階段一：安裝套件

- ◆ 安裝好作業系統後，進行系統更新及升級，並重新開機
- ◆ 安裝一些需要的編譯工具和影像 I/O 套件包含 JPEG, PNG, TIFF 等所需套件
- ◆ 安裝 video I/O 所需套件，使用 OpenCV 載入 video 檔案
- ◆ 安裝 GTK 開發 library，這個 library 用在建立使用者介面 (Graphical User Interfaces, GUIs)，並可以編譯 OpenCV 的 highgui 子模組，才能顯示影像在畫面上。

## · 階段二：使用 USB Camera

- ◆ 安裝 USB Camera

# 圖片處理 (1)

- Step1: 開啟影像
- 用到的functions:
  - cvLoadImage(...):載入圖片
  - cvReleaseImage(...):清除IplImage圖形資料結構記憶體
- 完成出現影像:





## 圖片處理 (2)

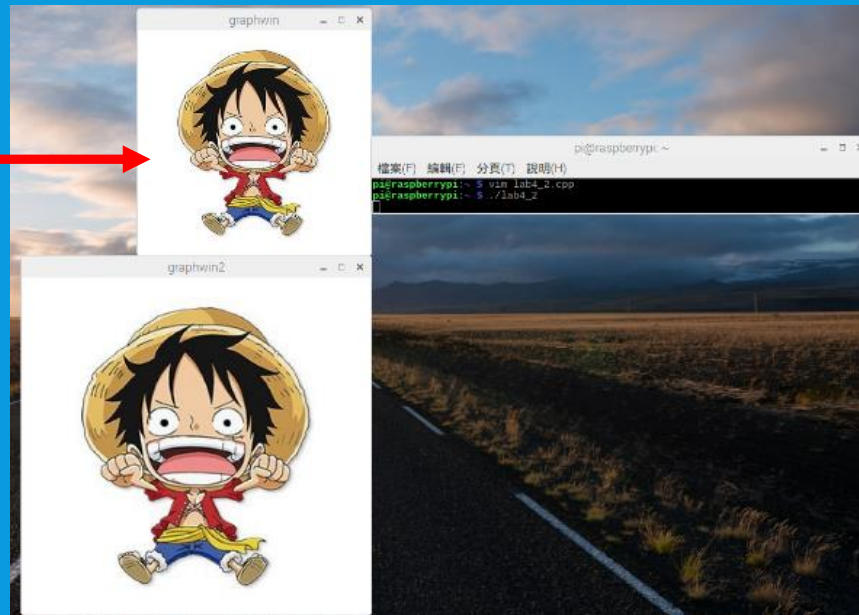
- Step2: 縮放尺寸
- 用到的functions:

`cvCreateImage( CvSize size, int depth, int channels );`

`cvResize(pImage, small, CV_INTER_AREA );`

- 完成出現影像:

原始大小



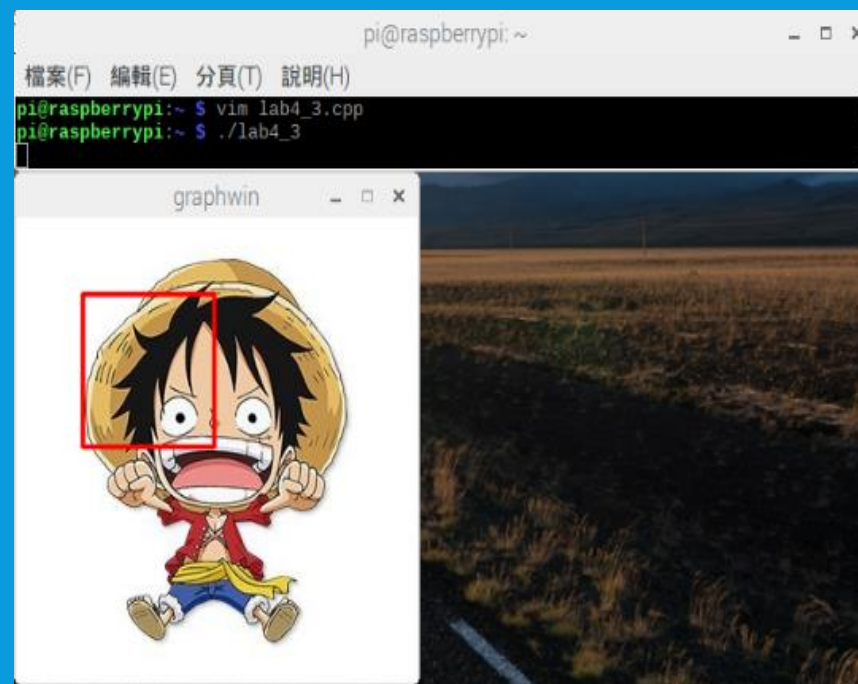
# 圖片處理 (3)

- **Step 3:** 圈選感興趣的區域 (ROI, Region of Interest)

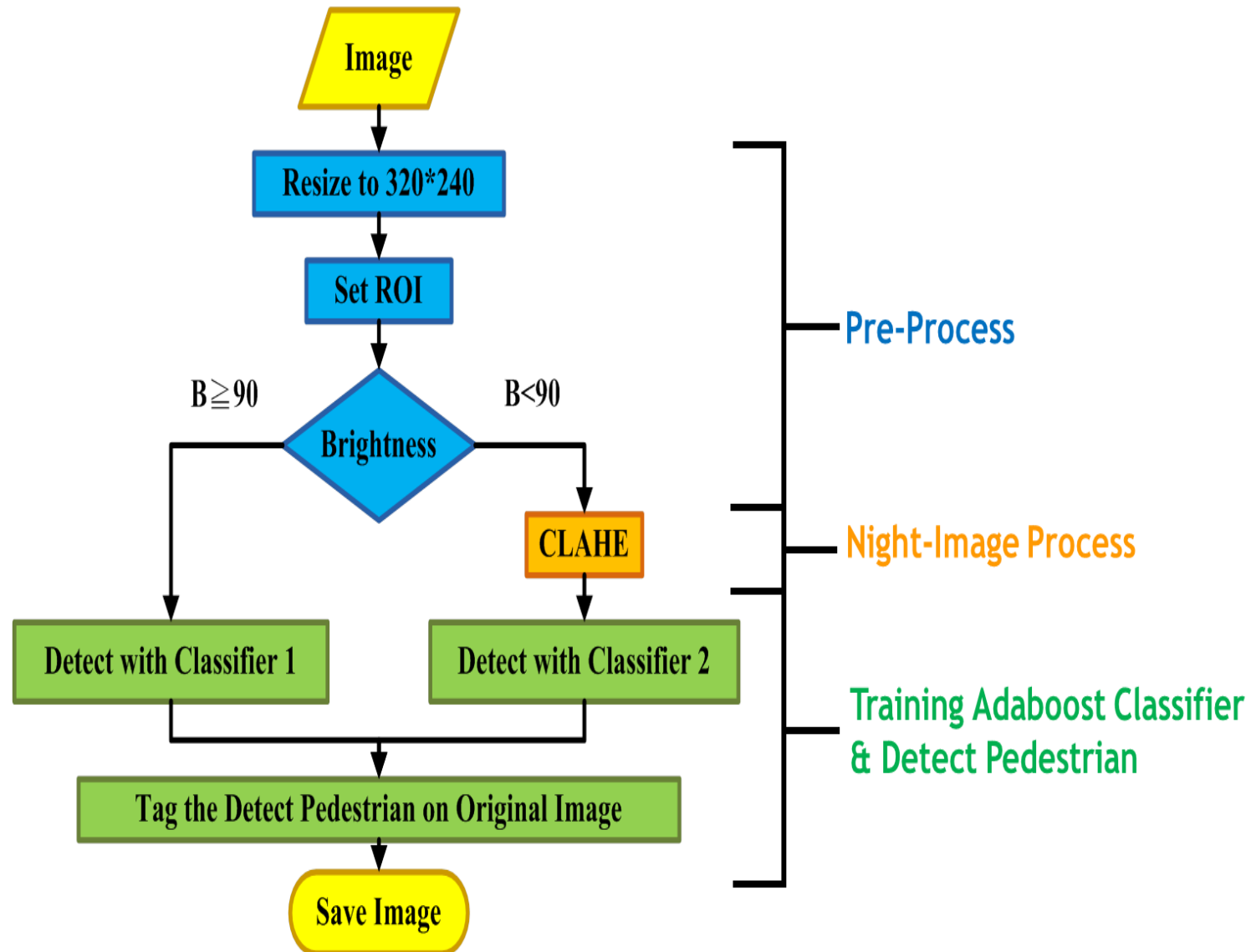
- 用到的functions:

`cvRectangle(CvArr* img, CvPoint pt1, CvPoint pt2, CvScalar color, int thickness=1, int line_type=8, int shift=0)`

- 完成出現影像:



# SYSTEM ARCHITECTURE



# 影像前置處理(PRE-PROCESS) (1)

- **Step 1:** 取得影像
- 用到的functions:
  1. 宣告`cvCapture* pCapture;`
  2. `pCapture = cvCaptureFromFile(filename);`
  3. `IplImage*plImage = cvQueryFrame( pCapture );`
  4. `cvGetCaptureProperty`
- 完成出現影像:



要幫你準備一個大臉盆

# 影像前置處理(PRE-PROCESS) (2)

- **Step 2: 縮小尺寸 (Resize)**

- 用到的functions:

`cvCreateImage( CvSize size, int depth, int channels );`

`cvResize(pImage, small, CV_INTER_AREA );`

- 完成出現影像:



# 影像前置處理(PRE-PROCESS) (3)

- **Step 3:** 圈選感興趣區域 (ROI)

- 用到的functions:

`cvRectangle(CvArr* img, CvPoint pt1, CvPoint pt2, CvScalar color, int thickness=1,  
int line_type=8,`

`int shift=0 )`

- 完成出現影像:





# 夜間影像處理 (NIGHT-IMAGE PROCESS)

圖片：

處理前



處理後



# 夜間影像處理 (NIGHT-IMAGE PROCESS)

影片：

處理前



處理後

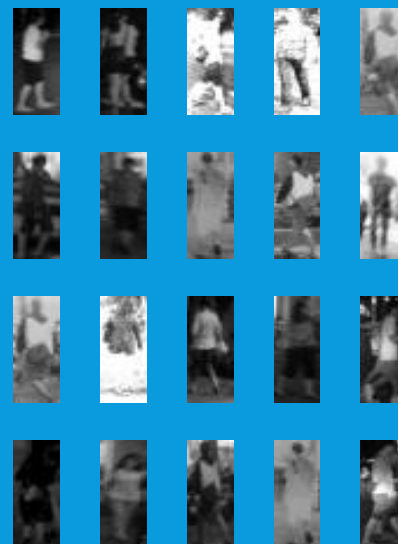




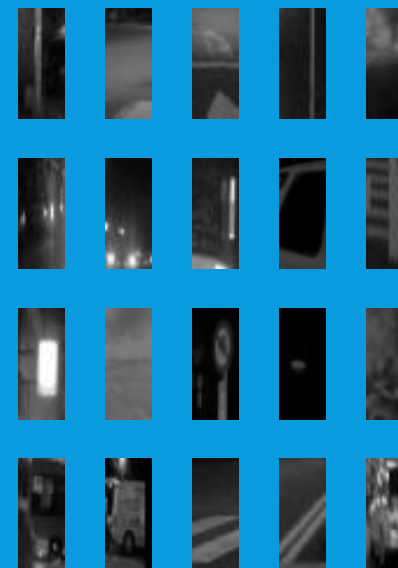
# 行人偵測: 利用ADABOOST演算法

## 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

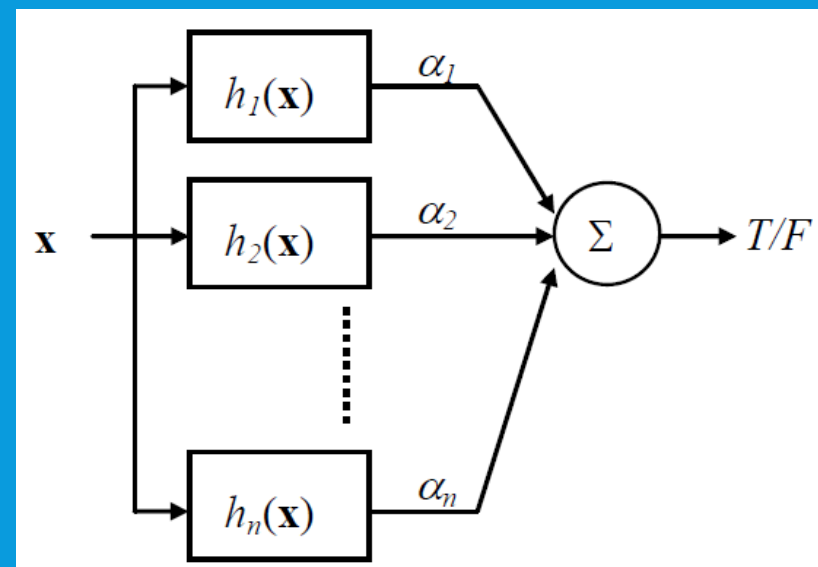
- 每一個樣本特徵形成一個弱分類器(weak classifier)規則
- 樣本經由一種特徵值做分類，在正特徵值範圍內的被判定為目標物，反之則否。在負樣本特徵值範圍內的判定為非目標物。再經由**Adaboost**做樣本權重調整

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”

# 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， $f(\mathbf{x})$ 為強分類器， $h_t(\mathbf{x})$ 為弱分類器 $\alpha_t$ 為每回合權重
- 強分類器示意圖

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



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”

# 人形偵測系統 (1)

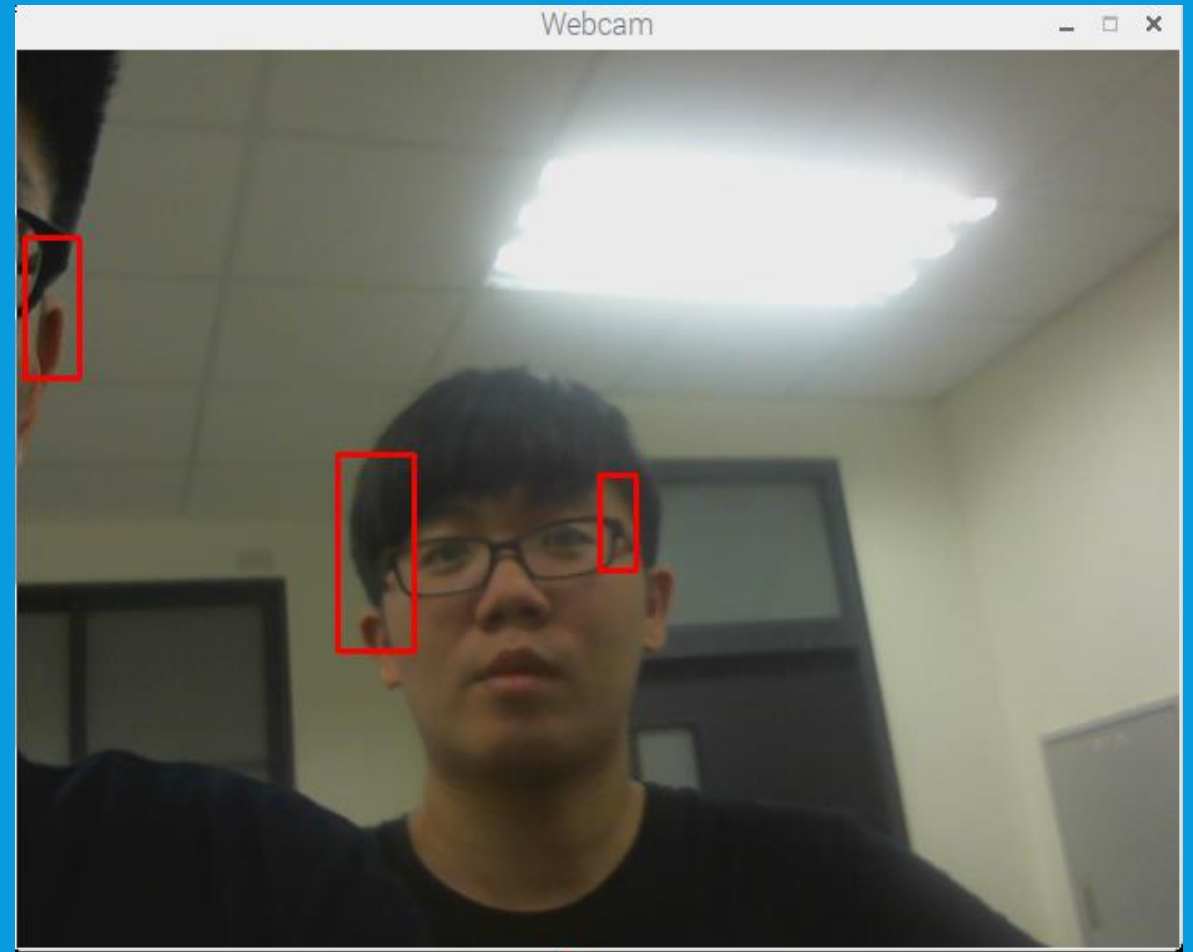
## ▪ Step 1: 抓影片人形

1. 使用取樣文件: 998\_763\_14\_35.xml (已經訓練好的分類器)
2. 使用draw()來框出分析的人形
3. 使用取樣文件來辨識人形並放置到彩色影片
4. 最後利用cvCreateVideoWriter記錄下分析的影片



# 人形偵測系統 (2)

- **Step 2:**使用webcam抓取人形
  1. 使用webcam來擷取成影像
  2. 使用取樣文件:998\_763\_14\_35.xml
  3. 使用draw()來框出分析的人形



# 參考網站

- <http://leonxlin.pixnet.net/blog/post/329986608-%E7%AC%AC%E4%B8%80%E6%AC%A1%E7%8E%A9%E6%A8%B9%E8%8E%93%E6%B4%BE%E5%B0%B1%E4%B8%8A%E6%89%B---raspberry-pi-3-model-b-%E6%96%B0>
- <http://atceiling.blogspot.tw/2017/02/raspberry-pi-opencv.html>
- <http://nickinwork.blogspot.tw/2015/06/rpi-usb-camera.html>

# 實驗DEMO

- 圖片處理: 1.開啟圖片 2. 圖片縮放尺寸 3. 圈選圖片區域
- 影像處理: 1.開啟影像 2. 影像縮放尺寸 3.圈選影像區域
- 夜間處理: 1.圖片夜間處理 2.影像夜間處理
- 人形偵測系統: 1. 影像抓取人形 2. 用Webcam抓取人形