

# 嵌入式系統軟體設計與實作 實驗報告十一

資工三 許耘熙 411410054

## 實驗名稱：

行人偵測專題

## 實驗目的：

利用 OpenCV 取得影像資訊再到圖像處理來完成行人偵測，學習電腦的視覺處理技巧並了解影像基本操作。

## 實驗步驟：

1. 使用 ssh 進行遠端連線

```
ssh -X yunhsihsu@172.16.1.53
```

2. 使用 sftp 將檔案傳到樹梅派

```
sftp yunhsihsu@172.16.1.53  
put -r ./pedestrian_detection
```

3. 設置相關環境以及安裝套件

```
sudo apt install python3  
which python3  
python -m venv env  
source env/bin/activate  
which python  
pip install opencv-python imutils  
python test_opencv.py
```

4. 觀察 processing.py 並執行

```
python processing.py
```

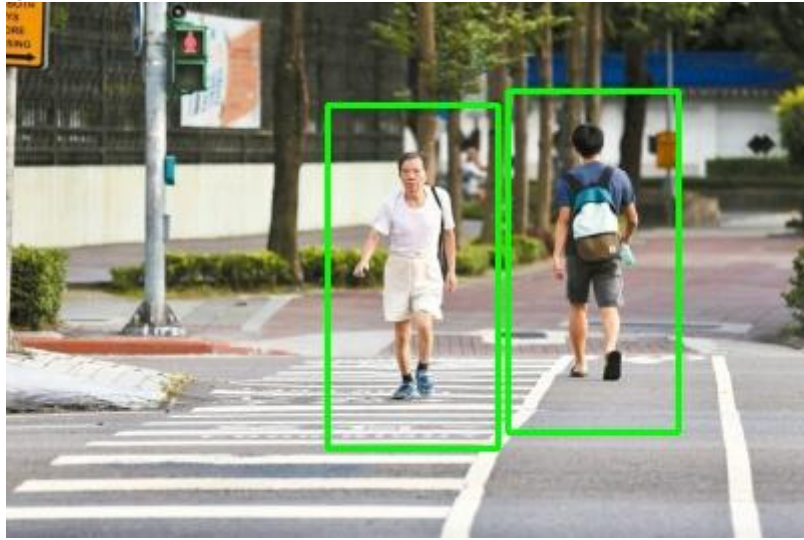
5. 觀察 video\_processing.py 並執行

```
python video_processing.py
```

6. 安裝 fswebcam 並拍照

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

## 7. 完成 sample.py 並執行產生結果圖



### 問題與討論：

#### 1. (指定問題)簡單說明 HOG algorithm 的原理。

##### (1) 梯度計算：

對輸入灰階圖像的每一個像素，計算水平方向與垂直方向的梯度。並得到梯度值以及梯度方向。

##### (2) Cell 切分與方向分箱：

將圖片分成小區塊(cell)，每個 cell 內根據梯度方向進行分箱，累加每個像素的梯度值到對應方向分箱中，得到該 cell 的方向直方圖

##### (3) 區塊正規化：

將相鄰的多個 cell 內的直方圖串聯成一個向量  $v$ ，並進行梯度的正規化，以減少光線與對比影響。

##### (4) 特徵向量生成：

將所有區塊正規化後的向量依序拼接形成整張圖或滑動視窗內的最終 HOG 特徵向量。

##### (5) 分類器訓練與判斷：

將 HOG 特徵向量輸入到線性 SVM 等分類器中，事先用正(行人)/負(非行人)樣本訓練出分類模型。偵測時對不同尺度的圖像金字塔(image pyramid)與滑動視窗分別計算 HOG、分類，最後用非極大值抑制(NMS)合併重疊的偵測結果。

### 2. 實驗心得：

在本次實驗中，我學習到了如何使用 OpenCV 套件實作行人偵測與影片分析，不僅強化了我對電腦視覺技術的理解，也了解了如何處理影片與即時影像資料。在延伸應用的部分，我選擇對於影片中的行人進行分析，並且統計每幀偵測到的行人數量。整個實驗下來，我認為最困難的是我想不到延伸應用可以做什麼，因此耽誤了好多時間才完成實驗。

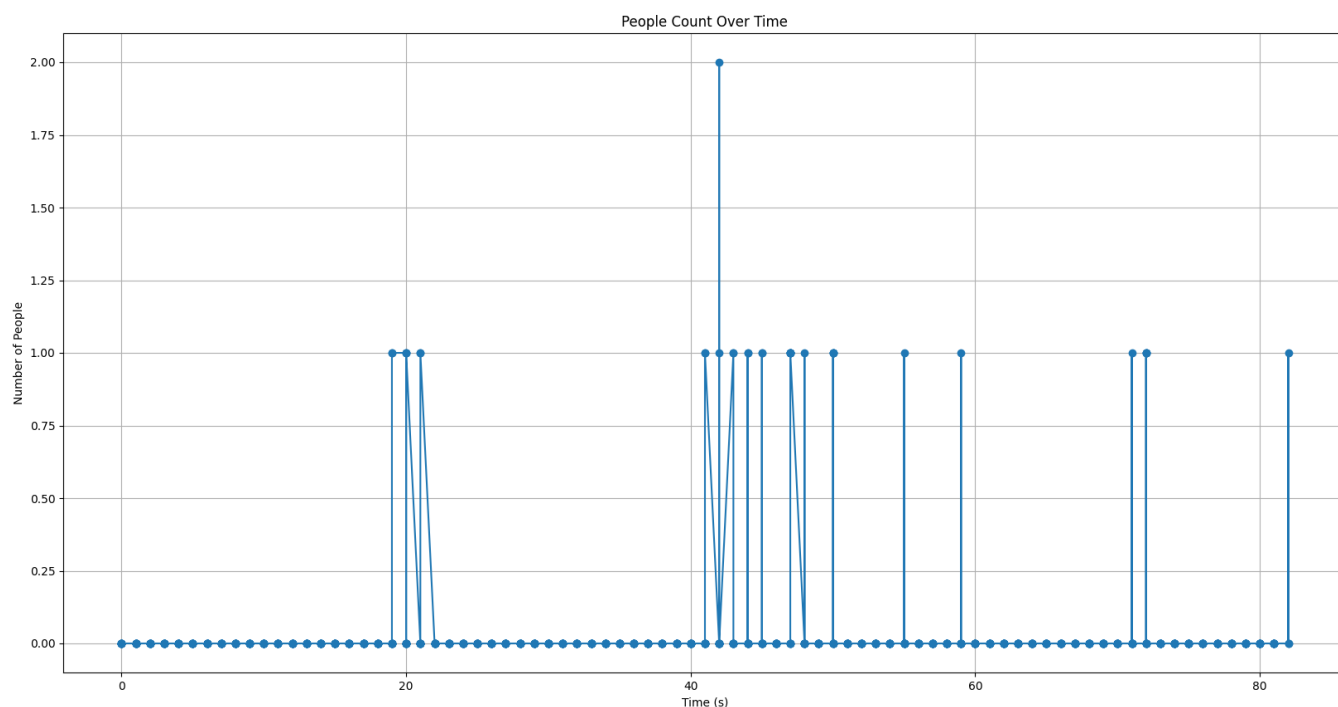
### 3. 延伸應用：

使用圖表統計在影片的那些時刻出現了幾個人

使用 Python 的 matplotlib 模組來進行圖表繪製

修改 video\_processing.py 程式碼並統計每一幀的行人出現次數

以下為結果圖：



### 4. 樹梅派操作畫面：

