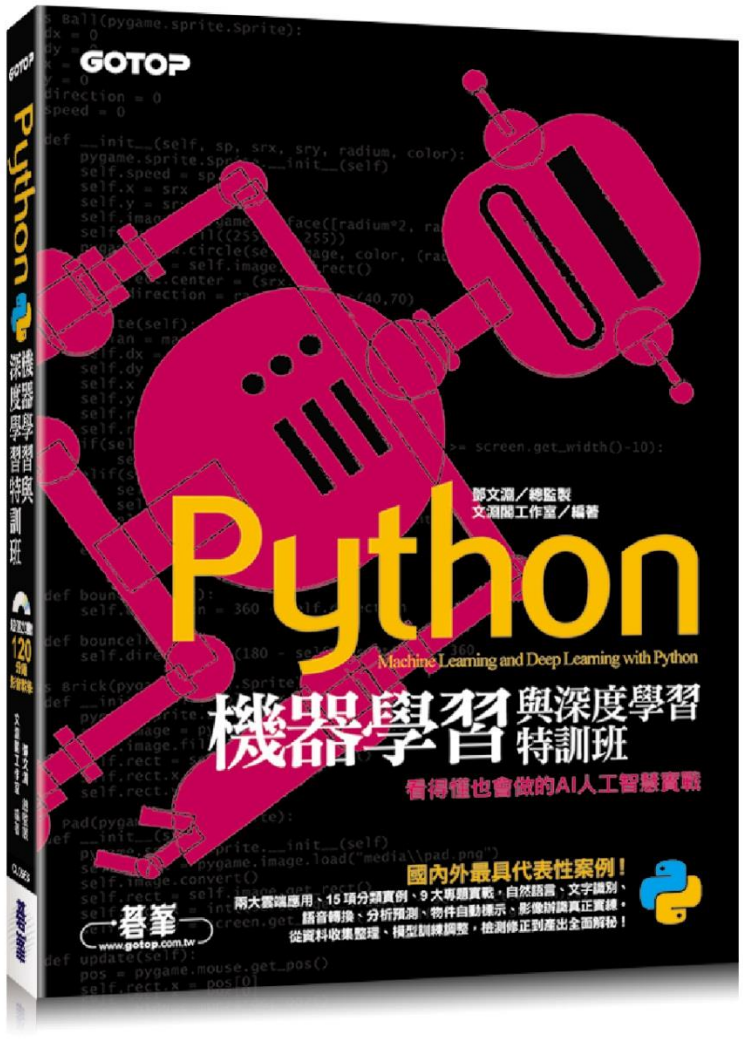


Python

Machine Learning and Deep Learning with Python

機器學習與深度學習特訓班

看得懂也會做的AI人工智慧實戰

11
自動標示物件：用HAAR特徵分類器擷取車牌

11.1 專題：車牌號碼偵測 (不是辨識)

- 首先蒐集「**正樣本圖片**」，就是包含要偵測物件的圖片，本專題就是含有車牌號碼的圖片，正樣本圖片格式必須為 BMP，所以將圖片格式轉換為 BMP，同時將圖形大小轉換為 300x225 像素。
- 再蒐集「**負樣本圖片**」，負樣本圖片就是不包含要偵測物件的圖片，轉換負樣本圖片格式為灰階，圖形大小為 500x375 像素。
- 正樣本圖片中必須將要偵測的物件框選出來，系統才知道要偵測的物件是什麼，本專題要偵測的是車牌號碼，所以必須框選所有正樣本圖片中的車牌號碼。使用 **Haar特徵分類器**模型進行物件偵測時，會根據**訓練**時的寬高比來框選物件。新式車牌的寬約為高的 3.8 倍，撰寫程式 將框選區域寬高比小於 3.8 的圖片，調整寬高比為 3.8。
- 正樣本圖片的數量越多，建立的 Haar 特徵分類器模型效果越好，撰寫程式增加正樣本圖片，本專題使用的方法是移除邊緣長寬各 10% 圖形來產生新圖形，如此每張圖片可新增 4 張新圖片。
- 最後進行訓練建立模型。
- **完成 Haar 特徵分類器模型**後，可使用 Opencv 讀入模型，對未知的車牌進行偵測，觀察偵測車牌號碼的正確率如何。

回首頁

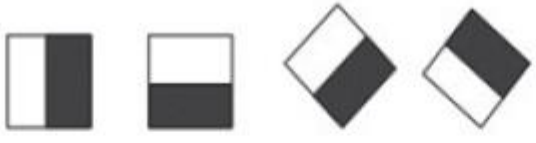
11.2 準備訓練 Haar 特徵分類器資料

- Opencv 最為人稱道的就是「人臉偵測」。
- 使用 Opencv 提供的 Haar 特徵分類器人臉模型，即可輕鬆偵測人臉位置。是否也可以偵測其他物件呢？
- 若要偵測指定物件，就要建立該物件的Haar 特徵分類器模型，再使用自行建立的 Haar 特徵分類器模型偵測物件。

回首頁

11.2.1 認識Haar 特徵分類器

- **Haar 特徵**是用來描繪一張圖片。Haar 特徵是一個矩形區域，此矩形可進行旋轉、平移、縮放等，共計有 15 個類型：



▲ 邊緣特徵

▲ 中心特徵

▲ 線性特徵

▲ 對角線特徵

- 使用 Haar 特徵分類器模型時，系統會在要偵測的圖片左上角產生一個**檢測矩形**，檢查此矩形內的圖形是否符合 Haar 特徵分類器模型特徵。
- 接著將此矩形向右移動檢測，到最右方時移到左側下方檢測，直到圖片右下角為止，這樣就可以檢測整張圖片。

11.2.2 處理正樣本及實測圖片

訓練 Haar 特徵分類器模型需要正樣本圖片及負樣本圖片，訓練完成後會產生模型，可用實測圖片 (非訓練圖片) 來**測試**模型的偵測效果。

「正樣本圖片」及「實測圖片」都是包含要偵測圖形的圖片，以本章要建立的偵測車牌號碼 Haar 特徵分類器模型為例，正樣本及實測圖片就是含有車牌號碼的圖片。

轉換圖片尺寸

- 手機拍攝圖片的解析度非常大，不適合做為訓練圖片，必須轉換為較小圖片才能進行訓練。**resize.py**會將圖片轉換為 300x225 像素大小，因為有 2 個資料夾檔案要轉換 (正樣本及實測圖片)，所以將轉換程式寫成函式。

正樣本圖片轉換為 bmp 格式

- 建立 Haar 特徵分類器模型時，正樣本圖片必須使用 bmp 格式。**changeBmp.py**會將<carPlate> 資料夾中所有 jpg 圖片皆轉換為 bmp 格式。

11.2.3 處理負樣本圖片

訓練 Haar 特徵分類器模型除了正樣本圖片外，還需要負樣本圖片，所謂「 負樣本」圖片就是沒有包含要偵測物件的圖片。以本章要建立的偵測車牌號碼 Haar 特徵分類器模型為例，負樣本圖片就是不包含車牌號碼的圖片，如果能與要偵測的物件有點關係的圖片更好，所以此處負樣本圖片使用各種道路圖片，<carNegative_sr> 資料夾包含 293 個負樣本圖片檔案。(原始檔<carNegative_sr>資料夾僅有290 個負樣本圖片)

- 訓練 Haar 特徵分類器時，負樣本圖片的尺寸500x375需略大於正樣本圖片尺寸，且負樣本圖片格式必須為灰階。(**resize_gray.py**)

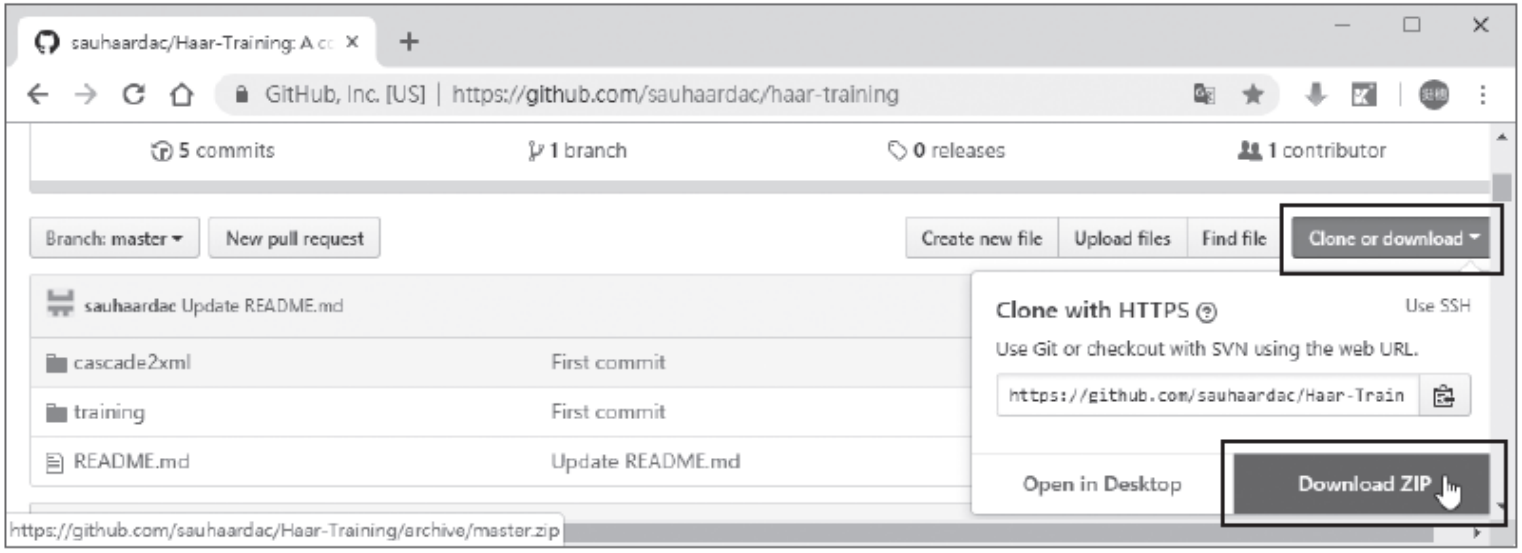
11.3 車牌號碼 Haar 特徵分類器模型訓練&建立

11.3.1 加入正、負樣本

Opencv 官網有提供自行建立 Haar 特徵分類器模型的方法，但過程非常繁複。

「https://github.com/sauhaardac/haar-training」整理建立了訓練 Haar 特徵分類器模型的相關程式，並為許多操作撰寫批次檔，大幅簡化建立 Haar 特徵分類器模型的程序。(本書範例已下載之，包含於範例中)

開啟https://github.com/sauhaardac/haar-training網頁，點選 **Clone or Download** 鈕，可自行再按 **Download ZIP** 下載壓縮檔。**(可略)**



回首頁

首先**加入正樣本圖片檔案**：正樣本圖片需置於 <Haar-Training_carPlate\training\positive\rawdata> 資料夾中。

- 先移除 <rawdata> 中所有檔案
- 將前一節建立的 73個正樣本圖片檔案 (位於 <carPlate> 資料夾，bmp 格式) 複製到 <rawdata> 資料夾中。

接著**加入負樣本圖片檔案**：負樣本圖片需置於 <Haar-Training_carPlate\training\negative> 資料夾中。

- 先移除 <negative> 中所有圖片檔案及 <bg.txt>，只留下 <create_list.bat> 檔。
- <bg.txt> 文字檔記錄所有負樣本圖片檔案名稱。
- 將前一節建立的 290 個負樣本圖片檔案 (位於 <carNegative> 資料夾，灰階圖片)複製到 <negative> 資料夾中。

1.3.2 正樣本標記資料

正樣本圖片中必須將要偵測的物件框選出來，系統才知道要偵測的物件是什麼，這個過程稱為「**標記**」。

- <Haar-Training_carPlate\training\positive**objectmarker.exe**> 為標記程式，以滑鼠按兩下執行，程式會自動載入第一張正樣本圖片。
- 在框選區域左上角按一下滑鼠左鍵，拖曳滑鼠到框選區域右下角放開滑鼠完成框選，再按 空白 鍵就可將框選資料記錄下來。如果圖片還有其他車牌號碼，可繼續框選，本章正樣本圖片都只有一個車牌號碼，按 **Enter** 鍵完成這張圖片標記，程式會自動載入下一張圖片讓使用者框選。重複框選圖片，直到所有圖片都完成為止。

11.3.3 顯示及修改框選區域

標記的正確性對 Haar 特徵分類器模型的影響極大，標記完成後最好檢視全部圖片的框選區域，如果發現框選區域偏差較大的圖片，可以進行修正。

- Opencv 並未提供檢視標記的功能，所以自行撰寫程式**picMark.py**，用以在所有正樣本圖片上產生框選區域。
- **picMark.py**程式會將結果全部改存放至<picMark>資料夾當中，共73個正樣本檔案。

修改圖片框選區域

使用者可以檢視所有繪製框選區域的圖片，將需要修改框選區域的圖片編號記錄下來。啟動 Windows 內建的 **小畫家**，開啟要修改的圖片檔案，將滑鼠移到框選區域左上角，記錄**小畫家**左下角的座標值；拖曳滑鼠到框選區域右下角，記錄此時的座標值。由起點及終點座標值計算框選區域的寬度及高度，然後手動修改 <info.txt> 檔案資料。

11.3.4 調整框選區域寬高比

新式車牌有七碼，較舊式車牌多一碼，因此新式車牌較為扁平，即寬高比數值較大。經實際測量，新式車牌的寬約為高的 3.8 倍，而舊式車牌寬約為高的 3 倍。使用 Haar 特徵分類器模型進行物件偵測時，會根據訓練時的寬高比來框選物件。如果正樣本的標記很多寬高比小於 3.8 時，使用模型進行物件偵測時新式車牌常會只擷取到部分車牌號碼。

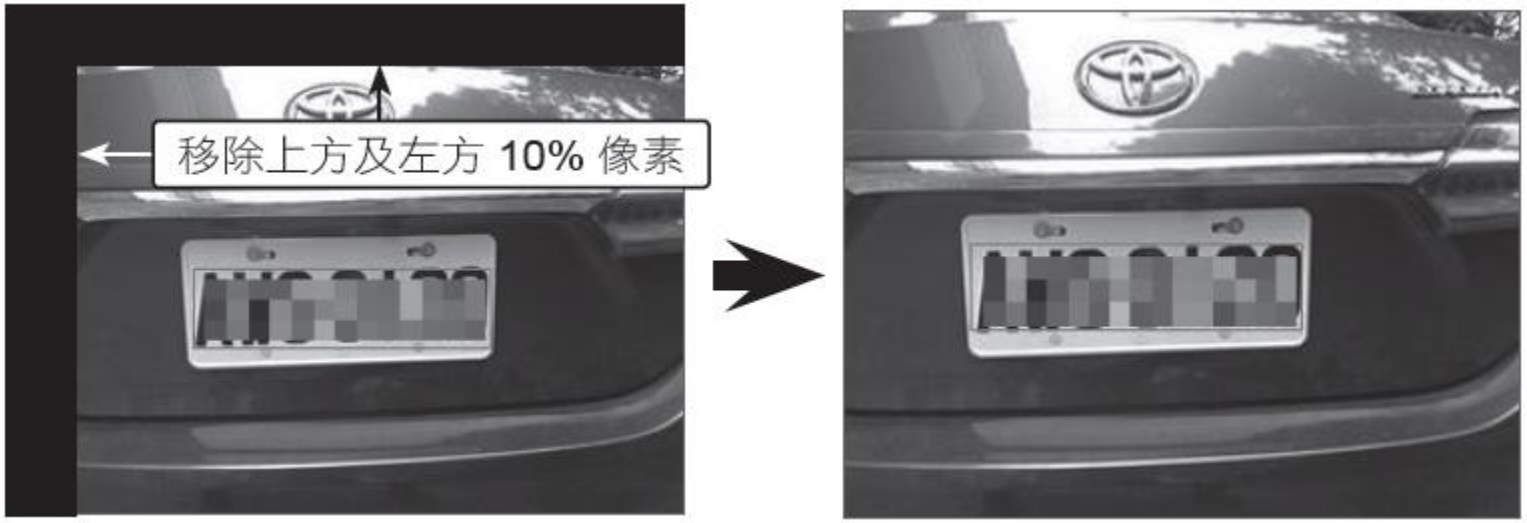
- modMark.py會將框選區域寬高比小於 3.8 的圖片，將框選區域的寬高比調整為 3.8，同時自動修改 <info.txt> 檔。
- picMark.py程式需重新執行一次，結果全部重新存放至<picMark>資料夾當中，共73個正樣本檔案。

11.3.5 增加車牌數量

正樣本圖片的數量越多，建立的 Haar 特徵分類器模型效果越好，但是要在街頭拍攝大量車牌圖片是件繁重的工作，還要提防某些人以為你是「抓扒仔」正義魔人，質問拍攝動機。

- make4Pic.py由現有圖片製造出更多圖片是常用增加正樣本圖片數量的方法，其方式很多，本章使用的方法是移除邊緣長寬各 10% 圖形來產生新圖形，如此每張圖片可新增 4 張新圖片 (若車牌號碼太靠近邊緣，新圖片車牌號碼可能部分被移除時，則不產生新圖片)。以移除左上角圖形為例：移除上方 30 像素及左方 22 像素，再將移除後的圖片尺寸放大為 300x225 像素成為新圖片。此時要注意框選區域也要依比例放大及調整框選位置。最後<rawdata> 資料夾中由73個正樣本檔案增加為329個。
- picMark.py程式需重新執行一次，結果全部重新存放至<picMark>資料夾當中，共329個正樣本檔案。

右上角、左下角、右下角可使用相同方法產生新圖片。



11.3.6 訓練 Haar 特徵分類器

打包向量檔 (*.vec)

- 正樣本圖片資料必須打包為向量檔才能訓練。
- 將正樣本圖片資料打包為向量檔的批次檔為 <training\samples_creation.bat>，請按下面內容修改後執行：

```
createsamples.exe -info positive/info.txt -vec
  vector/facevector.vec -num 329 -w 76 -h 20
```

進行訓練

- 訓練 Haar 特徵分類器的批次檔為 <training\haarTraining.bat>，請按照下面文字修改其內容後執行：

```
haartraining.exe -data cascades -vec vector/facevector.vec
  -bg negative/bg.txt -npos 329 -nneg 293 -nstages 15
  -mem 512 -mode ALL -w 76 -h 20 -nonsym
```

建立 Haar 特徵分類器模型

◆ 利用訓練結果建立 Haar 特徵分類器模型的批次檔為 <cascade2xml\convert.bat>，請按照下面文字修改其內容：

11.4 使用 Haar 特徵分類器模型

11.4.1 Haar 特徵分類器模型語法

- 使用 Haar 特徵分類器模型必須先安裝 Opencv 模組，如果尚未安裝。首先要匯入 Opencv 模組：

```
import cv2
```

- 接著以 CascadeClassifier 方法載入 Haar 特徵分類器模型，語法為：

```
模型變數 = cv2.CascadeClassifier(模型路徑)
```

- 然後用模型變數的 detectMultiScale 方法偵測物件，語法為：

```
偵測變數 = 模型變數.detectMultiScale(圖片, minSize=(寬, 高),
  scaleFactor=放大比例, minNeighbors=最小相鄰數)
```

- detectMultiScale 方法的傳回值是二維串列，第一維是偵測物件的個數，第二維是偵測物件資訊，偵測物件資訊包含 4 個元素，依序為偵測物件的左上角 X 座標、左上角 Y 座標、寬度、高度。例如偵測到 2 個物件的傳回值範例：

```
signs = [[67,112,142,37], [102,56,167,43]]
```

回首頁

11.4.2 車牌號碼偵測

- regCarPlate.py就是要以前一節所建立的模型，對於實測圖片中的車牌號碼進行偵測，並將車牌號碼框選起來。

批次框選車牌號碼

- <realPlate> 資料夾有 8 張實測圖片，regCarPlate_all.py可連續偵測所有圖片車牌號碼。