

基於 Haar 級聯分類器與多模態影像分析之智慧停車格偵測系統

Final Project Report for YZU 113-2 EEB215A

"Introduction to Computer Vision and Image Processing"

Name: 陳敏逸

Student ID Number: : 1100423

I. 專題目標

本專題旨在開發一套精準且具備高適應性的自動化停車格佔用偵測系統，使用手機相機模擬停車場攝影機，偵測元智大學地下停車場的車位占用情況。傳統的物件偵測方法在面對車輛部分遮蔽、停車角度不一或光線變化等複雜情境時，準確率常會下降。為了解決這些問題，本專題的具體目標如下：

1. **開發高適應性的停車格定義工具：**打造 GUI，使用者可透過滑鼠點擊方式定義不規則多邊形車位並指定唯一編號。
2. **實現多模態偵測引擎：**融合下列三種影像分析技術
 - Haar 級聯分類器：進行基於特徵的車輛結構偵測。
 - 顏色分析 (HSV)：分析停車格區域內的顏色分佈，判斷是否有異於地面的物體。
 - 邊緣分析 (Canny)：分析停車格區域內的紋理複雜度，判斷是否有結構複雜的物體。
3. **提升偵測準確度與可靠性：**將三種方法之結果加權融合，產生綜合信賴分數以決策佔用狀態。
4. **提供視覺化結果與報告：**於影像上標示佔用／空閒車位並輸出佔用率統計。

II. 系統設計

A. 資料集收集與標註

1. 收集資料集

- 正樣本：包含各式車輛之影像，來源包括元智大學地下停車場的實地拍攝，以及來自網路的停車場照片。



- 負樣本：不含車輛之背景影像，來源包括元智大學地下停車場的空景，以及網路上不含車輛的停車場照片



2. 手動標註正樣本

使用 *labelImg* 工具標註車輛位置，產生 XML 描述檔記錄標籤與邊界框座標。



3. 自動化資料預處理與增強

- 提取正樣本與背景圖 (Script 1):
`l_labels_to_pos_neg_imgs.py` 讀取 XML，裁切車輛 ROI (60 × 60 px)，並保存背景作負樣本。
- 資料增強 (Script 3): `3_augmentation.py` 對正樣本執行旋轉、縮放、平移、翻轉等隨機幾何變換。
- 產生描述檔 (Scripts 2 & 4)

- 2_generate-negatives.py 產生 negatives.info
- 4_add_aug_positives_to_list.py 產生 positives.info

4. 產生向量檔

使用 `opencv_createsamples` 讀取 `positives.info` 生成 `positives.vec`。

5. 訓練級聯分類器

透過 `opencv_traincascade` (輸入 `positives.vec` 與 `negatives.info`) 設定特徵類型、階段數與視窗大小，輸出 `cascade.xml`。

B. 主要程式說明

1. class.py (停車格定義工具)

- 滑鼠左鍵逐點定義多邊形車位；右鍵結束並輸入車位 ID
- 快捷鍵：s 儲存、r 重設、u 復原上個車位、c 取消當前標註、q 離開

2. vehicle_detection.py (停車格佔用偵測系統)

- 模式 1：單張影像偵測並可視化
- 模式 2：批次處理所有 pk^* 影像，輸出空閒車位清單並合成網格視窗

III. 核心偵測邏輯

本系統使用三種互補的偵測方法來判斷每個停車格是否被佔用，並透過加權計分方式融合結果，以提高準確度。每種方法分別提供一段信賴分數，最後將這些分數加總以判斷佔用情形。

1. Haar 級聯分類器偵測 (權重 0.2)

- 對輸入影像進行灰階轉換與前處理 (直方圖均衡與高斯模糊) 後，使用 OpenCV 的 Haar 級聯分類器來偵測車輛。
- 若車輛框有任一角點或中心點落入停車格定義區域內，則視為該

格有偵測到車輛，並加上 0.2 分的信賴分數，相較於透過和車位重疊區域判斷，此方式可以更準確的定位車輛是否在停車位中。



2. 顏色分析法（權重 0.4）

- 使用 HSV 色彩空間對車位區域進行遮罩處理與色彩分類。
- 假設地面為低彩度、低飽和度區域，若多邊形區域內非地面像素（顏色明顯異常）佔比超過 30%，則推斷為有車輛存在。
- 成功觸發此條件者，加上 0.4 分。

3. 邊緣密度分析法（權重 0.4）

- 使用 Canny 邊緣偵測器針對灰階影像處理，計算該停車格內的邊緣像素比例。
- 若邊緣密度超過 10%，代表該區內有複雜輪廓（車輛），則視為佔用，並加上 0.4 分。

最終信賴分數為三者加總：

- 總分 $\geq 0.5 \rightarrow$ 判定為「佔用」
- 總分 $< 0.5 \rightarrow$ 判定為「空閒」

此融合邏輯特別適用於處理如下情境：

- 陰影導致 Haar 偵測失敗，但仍可透過輪廓與顏色偵測補足。
- 車輛顏色與地面過於相似導致顏色判斷失效，但 Haar 或邊緣密度可彌補。

透過此三重機制，可顯著提升在實際環境中（如光線變化、角度歪斜、遮蔽物

干擾) 下的系統穩定性與精度。

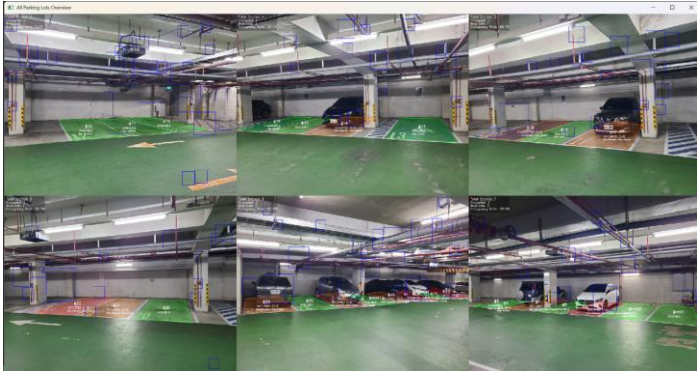
IV. 測試結果與討論

1. 模型訓練結果

使用不同參數來訓練模型，以下是個參數的訓練比較

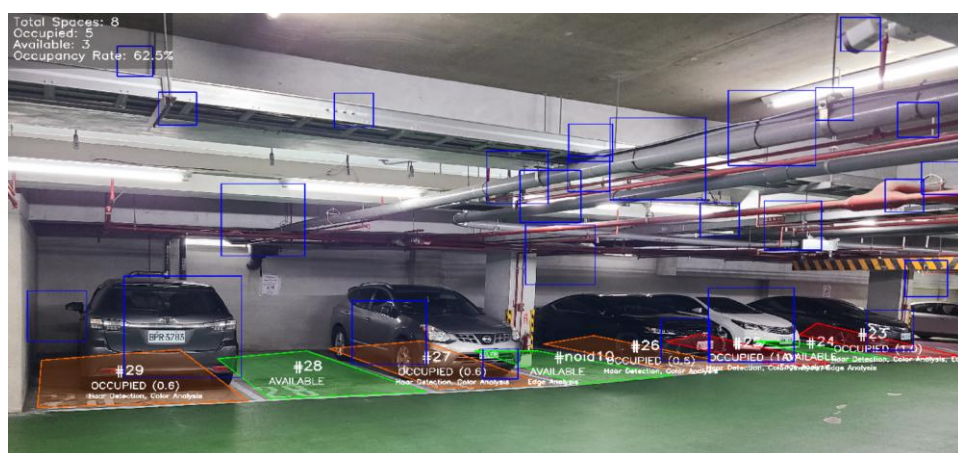
	編號	正樣本描述	負樣本描述	numPos	numNeg	numStages	特徵類型	觀察結果
1	1	完整車輛資料集 + 部分停車場剪裁	空停車場 + 不相關道路	1500	8000	15	LBP	準確率普通
2	2	停車場剪裁部分車輛特徵	空停車場 + 背景圖	700	3000	15	LBP	次佳效果
3	3	略增正負樣本	相同	900	3000	15	LBP	效果不佳，負樣本不足
4	4	同上	增加負樣本數量	900	5000	15	LBP	些微改善，但仍不如 #2
5	5	清除模糊/誤標	維持原負樣本	900	3000	15	LBP	最佳效果
6	6	無數據增強	少量樣本	130	600	15	LBP	樣本太少，效果弱
7	7	同上	增加負樣本	130	2000	15	LBP	些微改善
8	8	同上	同上	130	600	10	LBP	降低 stage 效果有限
9	9	數據增強 + 調整 stage	一般	900	3000	10	LBP	穩定但非最佳
10	11	相同	相同	900	3000	10	HAAR	效果中等

- 模型訓練比較
 - 最佳方案 (#5)：移除模糊與誤標樣本後表現最佳
 - 次佳方案 (#2)：局部特徵配背景對比亦具備良好辨識度
- 主程式執行結果
 - 測試 28 格中偵測到 16 格空位，誤標 3 格 (ID 19、21、22)





由解果可知，模型在判斷車輛特徵的精準度不夠理想，雖然可以框出大部分車輛特徵，但同時存在不小的誤判比例，可能須透過更的資料集合更多不同角度的車輛才能降低誤判比例



雖然停車場的車輛角度不一，且相鄰車位容易出現遮擋，以及模型本身的效能有限，容易出現誤標，但透過其他兩個檢測方式互相彌補，即使在有遮擋的情況下依然能有不錯的效果。

V. 挑戰與解決方法

1. 停車格形狀不規則，傳統矩形框無法適用

解決方法：透過 class.py，使用滑鼠點擊定義任意多邊形車位區域，提升適應性

2. 單一偵測方法準確率不足

解決方法：設計多模態偵測功能，融合 Haar、顏色與邊緣資訊，透過

加權融合抵銷單一模型誤判。

3. 模型訓練速度非常緩慢，使用 HAAR 訓練方式，一層約需要 12 至 24 小時，效率極差

解決方法：先使用 LBP 方式訓練取代 HAAR，等 LBP 訓練效果不錯後再用 haar 訓練(haar 效果沒有比較好，且花更多時間)

4. 訓練樣本太少

解決方法：透過 3_augmentation.py，對正樣本進行隨機的幾何變換，並透過 1_labels_to_pos_neg_imgs.py 將已標註圖片中，去除車輛後剩餘的背景作為額外的負樣本來源。

5. Opencv 4 不再內建 opencv_traincascade 以及 opencv_createsamples

解決方法：上網下載 opencv 3 的安裝包，解壓縮後把
\\opencv\\build\\x64\\vc15\\bin\\ 內的 opencv_traincascade.exe、
opencv_createsamples.exe、opencv_world345.dll、
opencv_world345.pdb 四個檔案放在專題目錄中

6. opencv_traincascade.exe opencv_createsamples 一直抓不到 info 檔

解決方法：將專題目錄放 d 槽，使用絕對路徑，路徑名不能有中文

VI. GitHub 與展示影片連結

- GitHub：<https://github.com/rayyichen310/haar-parking-lot-detection>
- 影片：
https://www.youtube.com/playlist?list=PLiFPyR0pCjc1oTTTQel3Bbor_rg9fKT4N

VII. 心得與反思

雖然現在很多任務都能靠深度學習模型解決，但我發現，傳統影像處理的基本

功依然不可或缺。當未來在設計新的演算法或調整參數時，這些基礎知識常常成為我理解模型行為、找到優化方向的重要依據。有時候模型效果不如預期，其實不是架構的問題，而是像前處理做得不夠好、輸入影像品質有落差，甚至只是某些空間特徵沒被適當捕捉而已。這種情況下，如果只看模型本身很難發現問題，但如果我對影像操作有基本的理解，像是怎麼降噪、怎麼分割區域、怎麼加強邊緣，那麼我就比較能抓到問題點，甚至在一開始設計訓練資料時就能避開一些坑。

在自主學習的過程中，雖然有 AI 可以輔助，讓我們少走一些彎路，但我也發現，AI 給的建議並不總是正確。有時它推薦的參數調整反而讓辨識效果越來越糟，或者在遇到像 `train_cascade` 的路徑設定問題時，它會陷入死循環，一直重複提供無效的修改建議。這時候，還是得靠自己動手查資料、看其他人怎麼解決類似問題，甚至要結合過去的經驗來判斷哪種做法比較合理。這讓我更深刻體會到，工具再怎麼進步，「自己會判斷」才是真正的能力。AI 可以是幫手，但最終還是得靠自己去理解問題的本質、找到最適合的解法。

VIII. 結論

本專題結合 Haar 級聯分類器與多模態影像分析，成功建構出一套具備實用價值的停車格佔用偵測系統。透過 GUI 介面靈活定義不規則車位、整合多種偵測方法並加權融合，有效提升了系統在面對遮蔽、光影變化與角度歪斜等情境下的適應性與準確度。雖然 Haar 的訓練過程繁瑣，且容易受到資料品質與標註精度的限制，但在實作與測試的過程中，透過多種判別方式相互搭配，有效降低了模型失誤的影響。

從開發程式到解決各種大大小小的 bug，不僅加深了我對傳統影像處理技術的理解，也讓我學會如何結合 AI 建議與自身經驗，做出更有判斷力的選擇。未來若要進一步擴展本系統，例如導入深度學習模型、處理即時影像串流，或應用在智慧城市等場景，有了這次的經驗，面對更複雜的應用也會更有方向。

參考資料

- [1] G. Bradski and A. Kaehler, *OpenCV User Guide: Training Cascade Classifier*, OpenCV 2.4 Documentation. [Online]. Available: https://docs.opencv.org/2.4/doc/user_guide/ug_traincascade.html
- [2] A. Sobral, *Vehicle Detection using Haar Cascades*, GitHub Repository. [Online]. Available: https://github.com/andrewssobral/vehicle_detection_haarcascades
- [3] A. Khare, *Automatic Parking Management*, GitHub Project. [Online]. Available: <https://github.com/ankit1khare/Automatic-Parking-Management>
- [4] R. Rouwenhorst, "Train Your Own OpenCV Haar Classifier," *Coding Robin*, Jul. 22, 2013. [Online]. Available: <https://coding-robin.de/2013/07/22/train-your-own-opencv-haar-classifier.html>
- [5] OpenCV Forum, "About traincascade parameters, samples and other," *OpenCV Q&A Forum*, [Online]. Available: <https://answers.opencv.org/question/7141/about-traincascade-paremters-samples-and-other/>
- [6] The Chinese University of Hong Kong, *Comprehensive Cars (CompCars) Dataset*, Multimedia Laboratory. [Online]. Available: https://mmlab.ie.cuhk.edu.hk/datasets/comp_cars/index.html
- [7] Stack Overflow, "Error: Train dataset for temp stage can not be filled while using traincascade," *Stack Overflow*, [Online]. Available: <https://stackoverflow.com/questions/11412655/error-train-dataset-for-temp-stage-can-not-be-filled-while-using-traincascade>
- [8] C. Tseng, "如何訓練一個貓臉偵測器," *程式學習紀錄筆記*, Feb. 7, 2020. [Online]. Available: <https://chtseng.wordpress.com/2020/02/07/如何訓練一個貓臉偵測器/>