國 立 虎 尾 科 技 大 學

電 子 工 程 系

專題製作報告

車牌辨識系統

(License plate recognition system)

指導教授 :蔡振凱

班 級 :夜四電子三甲

製作學生 :曾品鈞、陳岳鋒、楊岡穎、蔡佳展

學號 :40540411、40540415、40540419、40540425

中華民國 108 年 6 月 18 日

[摘要 2](#_Toc11783718)

[**動機** 2](#_Toc11783719)

[**預期成果** 3](#_Toc11783720)

[所需工具 4](#_Toc11783721)

[**使用軟體介紹** 5](#_Toc11783722)

[**OpenCV** 5](#_Toc11783723)

[**應用領域** 5](#_Toc11783724)

[Python 6](#_Toc11783725)

[**簡介** 6](#_Toc11783726)

[**歷史** 6](#_Toc11783727)

[參考資料 7](#_Toc11783728)

# **摘要**

在現今生活中，汽機車已經是每個人不可或缺的交通工具，車輛的使用越來越頻繁，但也衍生出許多治安與控制問題，例如:交通違規、車輛遺失等情形。在事件發生之後，藉由車牌辨別車主已經是常態，並記錄進入車牌號碼及車型，可依過往進出紀錄判斷是否為附近居民，現今警政單位依然是依靠人工辨別，大幅增加員警工作量與精神消耗，為有效減少警力負擔，此車牌辨識可應用於辨識使用車輛停車場或管制區。

藉由arduino控制攝像頭傳輸至電腦做紀錄並判別是否為可進入車輛，程式使用Python搭配OpenCV作影像辨識，並加入最新的機器學習(Machine Learning)辨別車型種類，且建立一資料檔並做紀錄。卻認為可進入之車輛就給指令給arduino控制閘門打開。若為危險車輛並控制arduino通知管理員避免意外發生。

## **動機**

每當我經過學校門口，時常看到有機車等待被警衛放行，他們並非不是校園教職員工，而單純只是校園車牌辨識系統僅能辨別小客車車牌，以致機車需要下車刷磁釦才能通過障礙，若能增進校園門口辨識系統想必須相當值得榮幸的事，並與專題組員討論過後，決定以此為題目作為大學專題內容。

現在最新型的停車場都是用車牌辨識系統來處理車輛，顯得更為重要，傳統用人工進行管理的方法不符合經濟成本於是才有車牌辨識系統的產生。還能記錄進出校門車輛紀錄，達到校園安全掌控。

## **預期成果**

使用YOLO物件辨識(Object Detection)，學習如何在一張圖片中找出車牌位子，並將它讀取出來，傳統的車牌辨識尋找車牌位子使用圖片輪廓化，再將其作二值化，透過膨脹與收縮輪廓，可以找到線條區域密度較高的車牌位子，找出車牌後利用投影法將兩張圖片做矩陣相減，相減值最小的圖片，即是該圖片代表的數字或英文。

傳統車牌辨識的做法有個缺點，就是在車牌位子有過多的裝飾容易造成機器判讀失敗，又或者未停留在規定區域及造成讀取失敗，且在判別汽機車上困難，不便於記錄，車牌讀取時，有特殊字母容易造成辨識錯誤，如:8和B、5和S、I和1、D和0等等，且運算量龐大，費時較長。

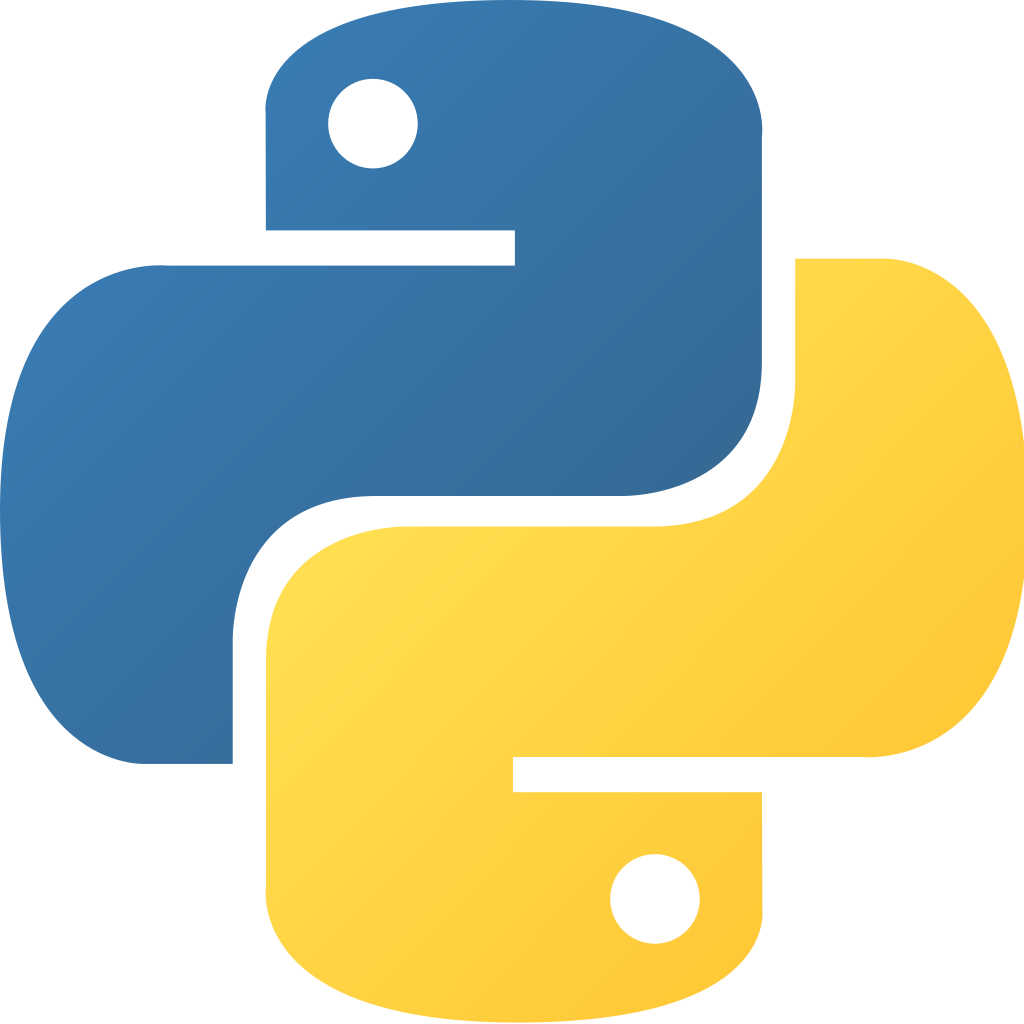
經以上分析我們決定採機器學習中的物件辨識(Object Detection)，並使用YOLO這套辨識方式，不僅可以辨識車種便於紀錄與日後的查詢，且可以降低字元辨識錯誤率。

YOLO將object detection視為一個single regression problem，YOLO在從影像輸入到輸出預測結果僅靠一個CNN來實現，利用CNN來同時預測多個bounding-box並且針對每一個box來計算物體的機率，而在訓練的時候也是直接拿整張圖丟到NN中來訓練，這樣end-to-end的算法可以避免傳統object detection的必須分開訓練的缺點，並且大幅加快運算速度，一般的YOLO版本可以在single Titan X GPU可以有45FPS，而輕量化版本甚至達到150FPS。

不僅可利用GPU大幅增加辨識速度，若模型訓練得當還可降低錯誤機率，且程式若能稍做修改，就可以使用在停車場各各角落達安全落實，將行經車輛的車牌做紀錄，若遇到車子被竊取就可以清楚知道從哪些出口離開，加快辦案速度，增加尋回機率。

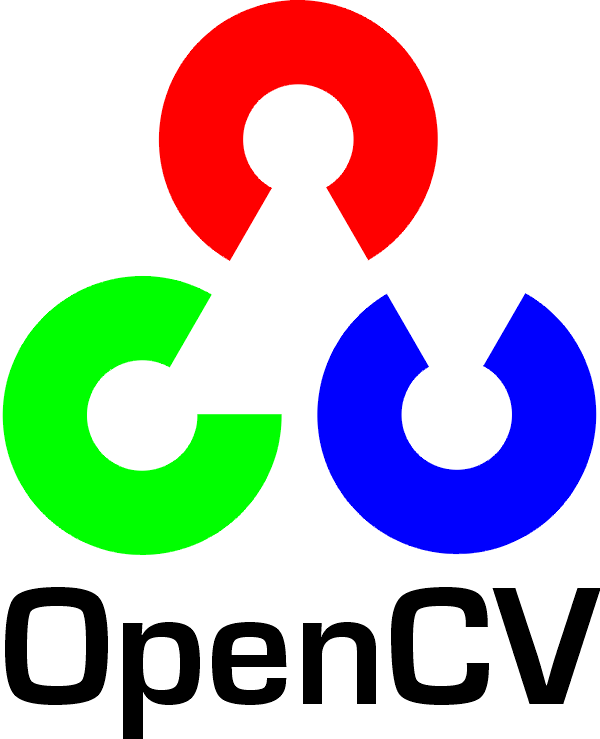
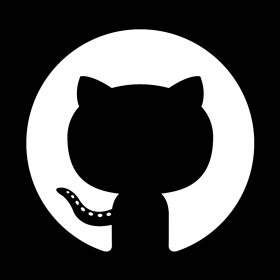
**所需工具**

* 使用工具 : 羅技 C920r HD Pro 視訊攝影機、台達DC小型馬達、3D列印
* 電腦硬體 : AMD\_R7\_1700X、RAM\_16G、HDD\_500G、GPU\_GTX970(總機)
* 開發程式 : 主要程式語言、主要開發套件OpenCV、主要程式編輯器Visual Studio Code、版本控制軟件Git、3D列印軟件SolidWork2013。

****

**Python**

**羅技 C920r HD Pro 視訊攝影機**



**GitHub**

**OpenCV**

## **使用軟體介紹**

**OpenCV**

　　OpenCV全名是Open Source Computer Vision Library，是一個影像處理函式庫，由Intel發起並參與開發，以BSD授權條款發行，可在商業和研究領域中免費使用，目前是非營利的基金組織OpenCV.org在維護。

　　OpenCV在影像處理方面應用廣泛，可以讀取儲存圖片、視訊、矩陣運算、統計、影像處理等，可用在物體追蹤、人臉辨識、傅立葉轉換、紋理分析、動態視訊的影像處理等。

　　OpenCV提供的函式方便我們推演更進階的影像處理演算法，就好像MATLAB的功用，但是執行速度比MATLAB快上許多，通常也比我們自己用C/C++寫的函式還快，而除了C/C++之外，OpenCV也提供其他語言的支援，像Java或Python等。

**應用領域**

　　影象資料的操作 ( 分配、釋放、複製、設定和轉換)。 影象是視訊的輸入輸出I/O （檔案與攝像頭的輸入、影象和視訊檔案輸出）。   
矩陣和向量的操作以及線性代數的演算法程式（矩陣積、解方程、特徵值以及奇異值等）。   
各種動態資料結構（列表、佇列、集合、樹、圖等）。   
基本的數字影象處理（濾波、邊緣檢測、角點檢測、取樣與差值、色彩轉換、形態操作、直方圖、影象金字塔等）。   
結構分析（連線部件、輪廓處理、距離變換、各自距計算、模板匹配、Hough變換、多邊形逼近、直線擬合、橢圓擬合、Delaunay 三角劃分等）。   
攝像頭定標（發現與跟蹤定標模式、定標、基本矩陣估計、齊次矩陣估計、立體對應）。   
運動分析（光流、運動分割、跟蹤）。   
目標識別（特徵法、隱馬爾可夫模型：HMM）。   
基本的GUI （影象與視訊顯示、鍵盤和滑鼠事件處理、滾動條）。H

**Python**

**簡介**

　　Python 是物件導向程式級高階程式語言，也是直譯式程式語言。Python 以強調對程式語言的語句易讀、易懂、易學(簡潔和清晰的語法特點)及加快程式開發的時效，方便使用，可以完成各種難度的應用，並可在大多數的系統中運行，以減少開發及維護成本的觀念進行發展。

　　Python 同時支援 modules 和 packages ，另外 Python 為跨平台程式語言也支援 unicode 字元。功能強大而完善的通用型語言，可以用於很多種軟體開發動態程式，使得 Python 非常有吸引力，發展至今已有十多年的歷史，成熟且穩定。

通常，有許多程式設計者接觸 Python 後都會被她的優點所吸引，並提供了許多自行開發的  library(函式庫)以提供其他 Python 程式設計者下載使用，所以具有豐富和龐大的類別函式庫，可以負荷支持平常大部份的應用，使得現今 Python 發展迅速及使用廣泛，相關資源也相當豐富。

**歷史**

　　Python 的創始人為 Guido van Rossum。在1989年聖誕節期間的阿姆斯特丹，Guido 為了打發聖誕節的無趣，決心開發一個新的腳本解釋程序，作為 ABC 語言的一種繼承。之所以選中 Python（大蟒蛇的意思）作為程序的名字，是因為他是一個 Monty Python 的飛行馬戲團的愛好者。可以說 Python 是從 ABC 發展起來，主要受到了Modula-3（另一種相當優美且強大的語言，為小型團體所設計的）的影響。並且結合了 Unix shell 和C的習慣。

　　早期 Python 是以 C 語言為基底撰寫而成，目前則有以 Java 語言為基底的 Jython 及架構在  .NET 的 IronPython 版本，甚至有以 Python 語言為基底的 PyPy 實作品，它可以將 Python 程式碼編譯成其他語言的執行檔，像是 C、CLI (.Net)、JavaScript 等。

　　Python 具有腳本語言中最豐富和强大的類庫，足以支持绝大多數日常應用。它的名字來源於一個喜劇，也許最初設計Python這種語言的人並没有想到今天 Python會在工業和科研上獲得如此廣泛的使用。

**參考資料**

程式前言

<https://codertw.com/%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80/541606/>

阿州的程式教學

<http://monkeycoding.com/?p=514>

宅學習

<https://sls.weco.net/CollectiveNote20/Python/Intro>

Pchome

<https://shopping.pchome.com.tw/>

git

<https://github.com/>

<https://backlog.com/git-tutorial/tw/>

yolo

<https://mc.ai/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-ml-noteyolo%E5%88%A9%E7%94%A8%E5%BD%B1%E5%83%8F%E8%BE%A8%E8%AD%98%E5%81%9A%E7%89%A9%E4%BB%B6%E5%81%B5%E6%B8%ACobject-detection%E7%9A%84%E6%8A%80%E8%A1%93/>

<https://medium.com/@chenchoulo/yolo-%E4%BB%8B%E7%B4%B9-4307e79524fe>

<https://www.itread01.com/content/1544112970.html>

python

<https://www.python.org/downloads/>