國立彰化師範大學資訊工程學系專題期末報告
自駕車智慧停車系統

丁德榮(指導教授)/賈承翔/林成龍/華柏凱/高瑞廷/謝金錦/楊育誠

國立彰化師範大學資訊工程學系

500彰化市進德路一號

Email: deron@cc.ncue.edu.tw(丁德榮)

摘要

隨著近年來自駕車產業與車聯網技術的蓬勃發展與逐漸普及，未來都市的道路設備系統也必將經歷一場革新。傳統的停車場系統，也將結合智慧化的設備與技術，帶給駕駛人許多使用上便利。因此本專題嘗試透過模擬未來都市的藍圖，結合人工智慧與時下流行的影像辨識技術及移動式的應用程式，再運用雲端資料庫儲存各項停車資訊並串聯各項設備，來建置一個更加智慧方便的停車場系統。主要功能為使用者可在停車場出入口，利用應用程式與自駕車進行連線，來通知車輛自動前往系統所分配的目標車位，提供自駕車更自動化的停車服務。我們希望能藉由此系統來解決現今現代人因停車而引發的眾多不便與糾紛，使得停車不再是一大夢靨。

Abstract

With the vigorous development and gradual popularization of the self-driving car industry and Internet of Vehicles technology in recent years, the road equipment system in the future city will also undergo an innovation. The traditional parking lot system will also combine intelligent equipment and technology to bring many conveniences to drivers. Therefore, this topic attempts to build a more comprehensive solution by simulating the blueprint of the future city, combining artificial intelligence with the popular image recognition technology and mobile applications, and then using the cloud database to store various parking information and connect various devices. Smart and convenient parking system. The main function is that the user can connect with the self-driving car at the entrance and exit of the parking lot to notify the vehicle to automatically go to the target parking space allocated by the system, and to provide a more automated parking service for the self-driving car. We hope that this system can solve the many inconveniences and disputes caused by modern people due to parking, so that parking is no longer a big dream.

一、簡介

1.1 研究動機與目的

近年來，全球汽車產業內最火熱的話題，就是自動駕駛汽車，眾多車廠投入大量資源，期盼能搶先卡位，取得重大商機。而隨著自駕車的蓬勃發展與普及，也將重重影響著未來都市的規劃與面貌，其中，停車場的建置與規劃，也勢必要進行重大的改變與革新。傳統的停車場，不但要靠駕駛人親自尋找車位，取票、付款，一連串複雜的程序讓人心浮氣躁，也浪費許多寶貴的時間。雖然近年來，隨著在停車場內裝設許多連網的設備與感測器，也讓停車場逐步升級，但那也只限用於傳統的汽車，而非更為智慧的自駕車。

而隨著行動網路與智慧型手機的普及，帶動了手機應用程式的使用率提升。因此若能讓駕駛人透過直接控制手機應用程式來進行停車場的各項服務，必定可提升使用者的便利性並進而增加停車效率。

1.2 研究問題

專題主要目的在設計一個自駕車智慧停車系統，主要功能如以下幾點，以提升自駕車停車的便利性與效率：

(1)在停車場周圍架設攝影機拍攝停車場，並利用人工智慧視覺辨識停車位的使用情況，再儲存至雲端資料庫。

(2)在停車場入口處架設攝影機辨識車牌，以避免傳統停車場複雜的入場程序，並同時能管控車輛的進入及會員確認、記錄進場時間等。

(3)乘客可在入口處下車，駕駛可利用 APP 及藍芽連線控制自駕車自行進入停車場內停車，減少駕駛的等待時間。

(4)乘客要取車離開時，可透過操作 APP 進行取車及線上付款。

(5)自駕車接獲取車指令後可自行離開停車場，並前往取車暫停車位，等待駕駛上車後再駛離。

二、系統軟硬體運用與技術探討

2.1 軟硬體的運用與規劃

在模擬自駕車的方面，我們選擇搭載 Jetson Nano 開發板的 SparkFun JetBot 人工智慧小車，利用小車上的鏡頭拍照並訓練後，則可實現小車在規劃的地圖上智慧行走。而在停車場的設備方面，首先我們使用一顆 webcam，在進口處辨識自駕車的車牌，控管自駕車的進入，接著在停車場周圍安裝數顆 webcam，結合 YOLOv4物件辨識對整個停車場進行即時的車位監控，並提示自駕車往目標車位前進。再來我們使用 Android Studio 開發的一個智慧化的停車 APP，讓使用者可直接透過操控 APP，來進行停車與取車的服務，並可查詢停車場的即時車位情況與使用者的會員信息、停車歷史記錄等等。最後我們使用 Firebase 雲端資料庫來儲存各種停車資訊，並藉以串聯各項設備，軟硬體完整運用規劃如表1。

表1、軟硬體完整運用規畫表

硬體	主功能	開發環境	開發語言	其他演算法/軟體之運用	資料庫
SparkFun JetBot (搭配 Jetson Nano 開發板、CSI camera)	道路循跡	JupyterLab	Python	PyTorch、TensorRT	Firebase Realtime Database
	物件辨識			YOLOv4	
	行駛車行方向			OpenCV	
	避障			PyTorch、TensorRT	
Logitech Webcam C270	車牌辨識	JupyterLab	Python	Tesseract 、OpenCV、YOLOv4	
	停車場辨識、車位使用情況	Spyder	Python	YOLOv4、OpenCV	
手機	停車與取車、車輛控制、停車會員管理	Android Studio	Java	無	
Arduino Uno、七段顯示器、紅綠燈	外部顯示功能	Arduino IDE	C	無	

2.2 軟硬體技術探討與文獻參考

2.2.1 Android Studio [1][2]

Android Studio 是一個為 Android 平台開發程式的整合式開發環境。2013年在 Google I/O 上發布，可供開發者免費使用。

主要功能如下:

(1)支援多種行動載具的 App 開發環境

Android Studio 支援各種行動載具的開發環境，包括 Android 手機、平板電腦、穿戴裝置、Google 眼鏡、Android TV、Android Auto 智慧車載系統...等可運行 Android 系統的裝置。同時 Android Studio 也支援 APK 讓開發者可對不同裝置，發布 APK 檔案。

(2)智慧程式碼編輯提供語法提示

開發者編寫程式碼時，會自動提供相關的語法提示，協助開發者重組、完整化與分析程式碼。此外，Android Studio 也提供程式碼範本，來協助開發者建置應用程式的基本功能、並支援豐富的版面編輯工具。

(3)效能分析工具監控 App 記憶體使用量

Android Studio 提供視覺化的監控工具 Memory Monitor，為開發者追蹤連結裝置的記憶體使用量，並監控 App 的記憶體使用情形。當開發者要在裝置上運行或模擬 App 時，可點選右下方的 Memory Monitor 來啟動這項記憶體監控工具。

2.2.2 Firebase [3][4]

Firebase 是一個同時支援 Android、iOS 及網頁的 App 雲端開發平台，協助 App 開發者在雲端快速建置後端服務，提供即時資料庫，有效縮短 app 開發時間，並幫助開發者更專注在前端的優化。

Realtime Database 是許多 APP 應用開發者使用 Firebase 的重要原因，它本身與大多數開發者熟練的關聯式資料庫 SQL 有很大的不同。即時資料庫提供的不只是資料的存取，它的 SDK 功能也包括了監聽等功能， 因此可以在開啟應用的同時，當資料庫資料有更新時，很容易的即時 同步到本機中，而不需要特別在後台設定一些監聽封包程式。這在需要即時監聽的應用特別好用，如討論區、即時通訊等，而且大多數的同步都是以毫秒計的時間完成同步的，不但支援 APP，也支援 JavaScript 同步到網頁或其他應用平台上。

2.2.3 YOLOv4 [5][6][7]

YOLO (You Only Look Once) 是一個 one-stage 的 object detection 演算法，將整個影像輸入只需要一個 DNN 就可以一次性的預測多個目標物位置及類別，這種 end-to-end 的算法可以提升辨識速度，能夠實現 real-time 偵測並維持高準確度。YOLO 可以一次性預測多個 Box 位置和類別的卷積神經網路，能夠實現端到端的目標檢測和識別，其最大的優勢就是速度快。

YOLO 的實現方式則如圖1 ：

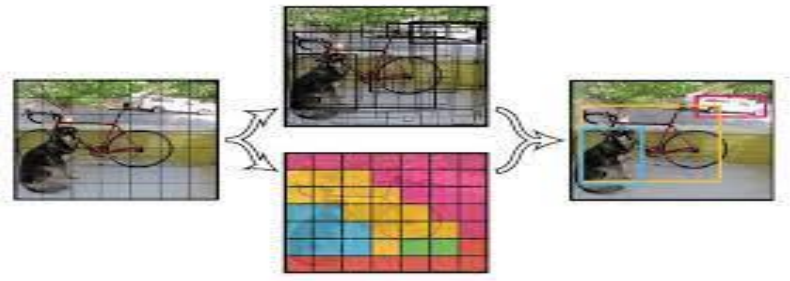

圖1、YOLO 實現範例圖[5]

(1)首先要預 B 個 bounding box，而此 bonding box 除了回歸自身位置， 也要附加自身的 confidence。confidence 代表所預測的 box 中有此物件的可信度以及 box 預測準確度。以下是其計算式： (Pr(Object)*IOU(pred|truth))

(2)每個 bounding box 要預測(x, y, w, h)和 confidence 共5個值，每個網格還要預測一個類別信息，記為 C 類。將原圖劃分為 S*S 的網格，每個網格要預測 B 個 bounding box 還要預測 C 個 categories。輸出就是 S*S*(5*B+C)的一個 tensor。

(3)類神經網路架構(如圖2)：

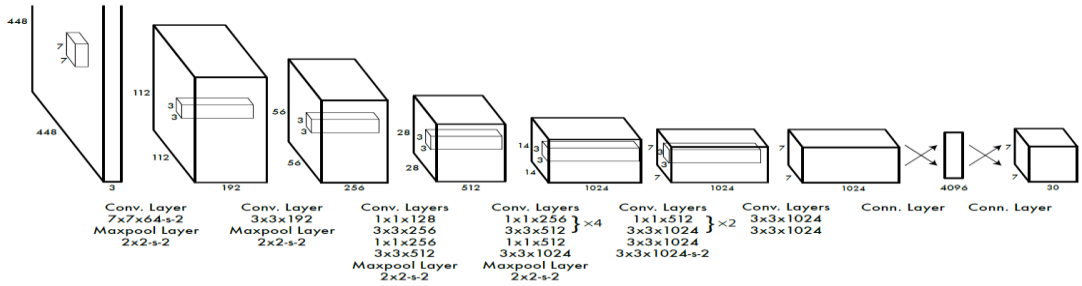

圖2、類神經網路架構圖[5]

而在測試的時候，每個網格預測的類別訊息和 bounding box 預測的 confidence 相乘，就得到 class-specific confidence score：

$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

YOLOv4主要改善如下：

(1)建立了一個高效強大的目標檢測模型，並且使用 1080Ti 或 2080Ti 的 GPU 就可以進行訓練。

(2)驗證了 SOTA (State of the Art) 的 Bag-of-Freebies 和 Bag-of-Specials 目標檢測法在檢測器訓練過程中的影響。

(3)改進了一些 tricks 、SOTA 的方法，包括 CBN、PAN、SAM 等，使之更加高效，並能夠在單 GPU 上訓練。

2.2.4 PyTorch [8][9]

PyTorch 是一個開源的 Python 機器學習庫，基於 Torch，底層由 C++實現，應用於人工智慧領域，如自然語言處理。它最初由 Facebook 的人工智慧研究團隊開發，並且被用於 Uber 的概率編程軟體 Pyro。類似於 NumPy，它具備 GPU 附加功能。與此同時，它也是一個深度學習框架，為實現和構建深層神經網絡體系結構提供了最大程度的靈活性和速度。

最近發佈的 PyTorch 1.0幫助研究人員應對以下四大挑戰：

(1)大面積的返工

(2)耗時的訓練

(3)Python 語言缺乏靈活性

(4)慢速擴展

從本質上講，PyTorch 與其他深度學習框架有兩個不同點：

(1) 命令式編程：PyTorch 在遍歷每一行代碼的同時執行計算，這與 Python 程式的執行方式非常類似，這概念稱為命令式編程，優點是可以動態地除錯代碼和編程邏輯。

(2) 動態計算圖：PyTorch 被稱為“由運行定義的”框架，這意味著計算圖結構(神經網絡體系結構)是在運行時生成的。該屬性的主要優點是：它提供了一個靈活的編程運行時接口，通過連接操作來方便系統的構建和修改。在 PyTorch 中，每個前向通路處定義一個新的計算圖，這與使用靜態圖的 TensorFlow 形成了鮮明的對比。

2.2.5 TensorRT [10]

TensorRT 是 Nvidia 提出的深度學習推論平台，能夠在 GPU 上實現低延遲、高吞吐量的部屬。可將訓練好的神經網路輸入 TensorRT 中，產出最佳化後的推理引擎。使用 TensorRT 的應用程式，比起 CPU 平臺的執行速度還要快40倍。TensorRT 建構於平行可程式化模型 CUDA 之上，提供精度 INT8和 FP16的最佳化外，也支援 TensorFlow、Pytorch、Mxnet、等深度學習框架，包括嵌入式、自動駕駛以及 GPU 計算平臺，讓用戶可將神經網路部署到資料中心，同時執行多個模型。低精度推理能夠大幅地減少應用程式延遲，符合更多即時服務、自動化與嵌入式應用程式的需求。

TensorRT 主要的優化方法為以下兩種：

(1) 降低資料精準度

大規模深度學習框架在訓練神經網絡訓練時網絡中的張量（Tensor）是32位浮點數的精確，完成網絡完成，在推理的過程中由於沒有洋裝傳播，完全可以適當降低數據精度，例如降為 FP16或 INT8的精度。

(2) 對模型進行重構與優化

A. 先刪除沒有使用到的輸出層。
B. 垂直優化：由於CUDA 會在每一層輸入的時候啟動，而這個垂直優化就是透過將常使用的層作合併，像圖中將 Conv、bias、relu 合併為 CBR，減少 CUDA 啟動關閉的次數來達到時間上的縮減。
C. 水平優化：如果同層其他分支也在運作一樣的架構則直接合併，要注意的地方是合併成同一層之後輸出必須要做分割，在輸出到不同層去。

2.2.6 OpenCV [11]

Opencv是由英特爾公司1999年發起並參與開發的跨平台電腦視覺庫，能使使用者在進行相關程式開發時變得更易於上手，降低入門門檻高度，一般用於圖像辨識、處理和電腦視覺等相關領域的應用開發。

Opencv 的優點如下:

(1)用途廣

Opencv 是現今最主流的函式庫之一，提供了大量的函式支援，並且包含了許多方面，像是擴增實境、物體辨識、人機互動和圖像分割等...，這些方面都有一定得重疊又不完全相同，Opencv 可以很好的將它們結合起來。

(2)多人使用

使用同個基礎函式庫，不重造輪子，並且在這個前提下，程式碼更容易被理解、閱讀和轉讓，促進相關領域的發展，並且提供多種開發語言以供使用。

(3)跨平台、裝置

在主流平台如 Windows、Linux、Android、ios 等平台上皆可運行，不必太多考慮作業系統的相容性，尤其支援許多使用嵌入式 Linux 系統的開發板，如 Rassberry PI、Jeston nano 等。

2.2.7 Tesseract [12][13]

Tesseract 是一個光學字元辨識(OCR)引擎，可以讓圖片中的文字經由 Tesseract 被轉化為文字字串，Tesseract 最早由 HP 實驗室於1985年開始研發，現在作為開放原始碼項目放在 Google Project 上，並且支持超過一百種語言，常分為以下三步驟:

(1)前期處理

主要針對圖片本身，分成三步驟:

A. 二值化:將彩色圖片中不需要的色彩資訊排除，以更好的進行辨識，用簡單的話描述便是黑白化。
B. 圖像降噪:不同的圖像，噪點定義也不同，根據噪點(一種妨礙辨識的干擾)的特徵進行去噪的過程。
C. 傾斜校正:將圖片水平、垂直對齊。

(2)中期處理

A. 版面分析:將文章分段、分行，不過以當前技術無法做的很完美。
B. 字元切割:將不同字元分隔開，提高效能、準確率。
C. 字元辨識:提取特徵以做辨識，多種因素皆影響辨識結果、品質。
D. 版面還原:將辨識得到的字串復位。

(3)後期處理

根據上下文，對結果進行校正。

2.2.8 Logitech Webcam C270 [14]

一款由羅技生產的 usb webcam，具有720p 且30fps 的畫面，以及自動光線校正功能。體積小巧靈活且定位穩固，適用於許多的作業系統。

2.2.9 Jetson Nano 開發板 [15]

NVIDIA Jetson 是 NVIDIA 為嵌入式系統設計的人工智慧平台，使用 NVIDIA 開發的 Tengra 處理器，包含有圖形處理器、音效處理器、南北橋晶片和記憶體控制器等功能。

Jetson nano 的特色如下 ：

(1)人工智慧的全新境界

尺寸僅70x45mm，是全球最小的 Jetson 裝置。這種可量產的模組系統(SOM)能將人工智慧部署至各個產業的終端裝置，包括智慧城市和機器人。

(2)強大運算效能

具備472 GFLOP 的運算能力，可快速執行現代人工智慧演算法，並可同時執行多個神經網路，以及同步處理數個高解析度感應器，非常適合應用在入門級網路錄影機(NVR)、家用機器人，以及具備完整分析功能的智慧閘道器。

(3)低電量需求

Jetson Nano 釋放終端裝置創新潛能。只需耗費 5 到 10 瓦特，就能運用功能強大又有效率的人工智慧、電腦視覺和高效能運算。

2.2.10 SparkFun JetBot [16]

SparkFun JetBot AI Kit Jetson Nano 人工智慧小車開發套件，為開源機器人，並使用 NVIDIA Jetson Nano 做為運算核心。它易於設置和使用，並與許多流行的配件兼容，包括可使用無線搖桿控制器或網頁控制 JetBot 小車、透過攝影機拍照收集資料等，最後可在 Jetson Nano 裡面訓練、實現避免碰撞、物件跟隨、道路跟隨等效果。

SparkFun JetBot 開發套件的特色 ：

(1)快速連結感應器、LCD、馬達。

(2)不需額外設定即可使用的 AWS 雲端。

(3)適用於 I2C 通訊的 SparkFunQwiic 生態系統。

(4)可以使用 GPIO 接頭上的4個 Qwiic 連接器擴展生態系統。

(5)示例代碼：基本運動，遙距操作，避免碰撞和對象跟隨

(6)緊湊的外形尺寸可優化 NVIDIA 現有的神經網絡

(7)136°FOV 攝像機可實現機器視覺

(8)機箱組裝提供可擴展的架構

2.2.11 Arduino [17]

Arduino可以讓你的計算機能夠擁有感應、控制真實世界的能力，而不僅局限於鍵盤、滑鼠、等單一的標準 I/O 設備。它同時也能作為獨立的核心，作為機器人、智能車等電子設備的控制器，應用非常簡單。可用於開發交互式對象，採取各種開關或傳感器輸入，控制各種燈，電機和其他物理輸出。Arduino 的項目，可以獨立，或者與計算機上運行的軟體通信。Arduino 包括硬體平台(Arduino Board)，和開發工具(Arduino IDE)。兩者都是開放的，既可以獲得 Arduino 開發板的電路圖，也可以獲得 ArduinoIDE 的原始碼。除了購買 Arduino 電路板外，不需要支付額外的費用。Arduino Board 基於簡單的微控制器，如 ATmega328，提供了基本的接口和 USB 轉串口模塊。使用者只需要用一個 USB 線就可以連接電腦和 Arduino Board，完成編程和調試，而不需要專門的下載器。Arduino 使用一種簡單的專用程式語言，使用者不必掌握彙編語言和 C 語言等複雜技術就可以進行開發。IDE 可免費下載，並開放原始碼，跨平台，極為便利。

三、系統功能設計及架構

系統功能架構圖如圖3

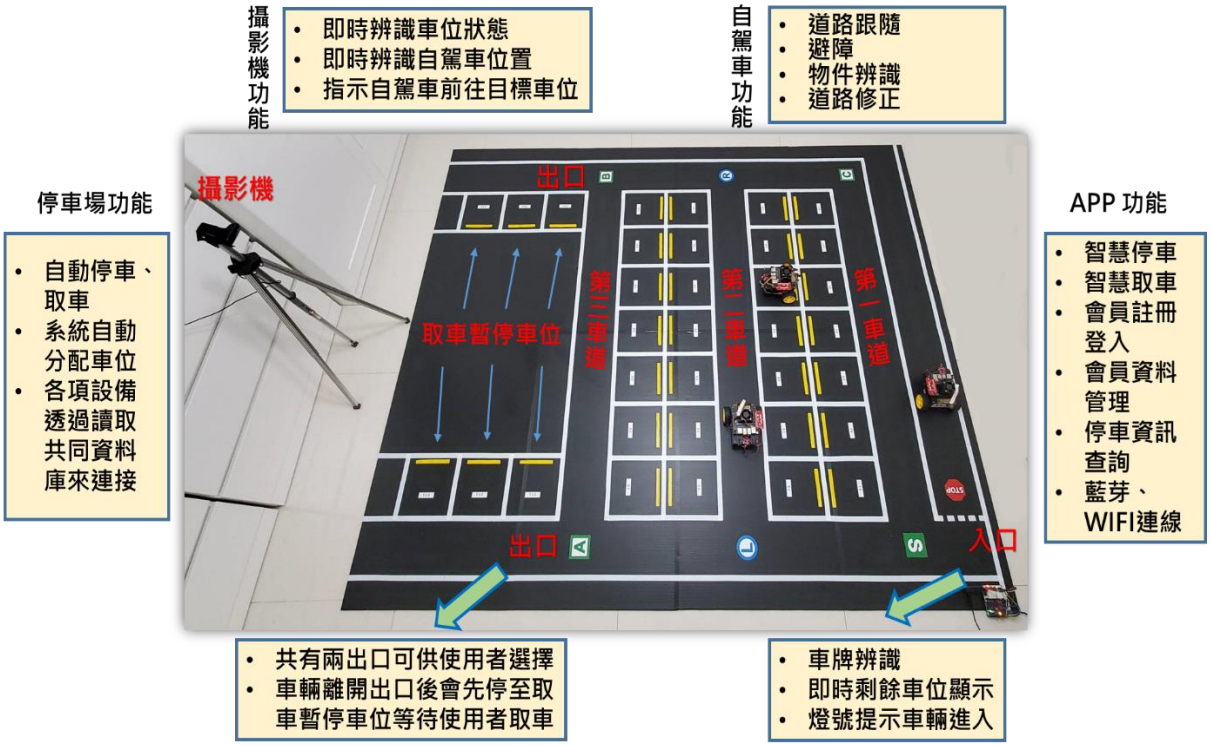

圖3、系統功能架構圖

3.1系統分項功能及實作說明

3.1.1 APP 實作功能說明

App 功能流程如圖4：

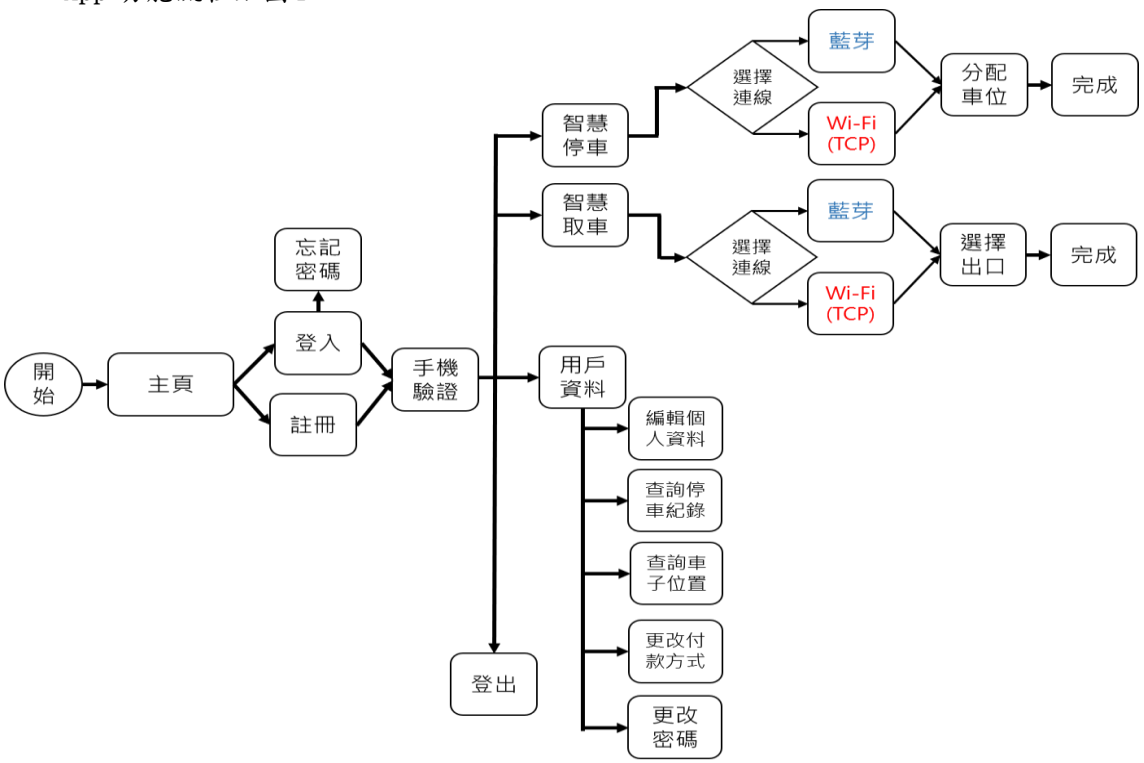

圖4、App 功能流程圖

藍芽連線功能流程如圖5：

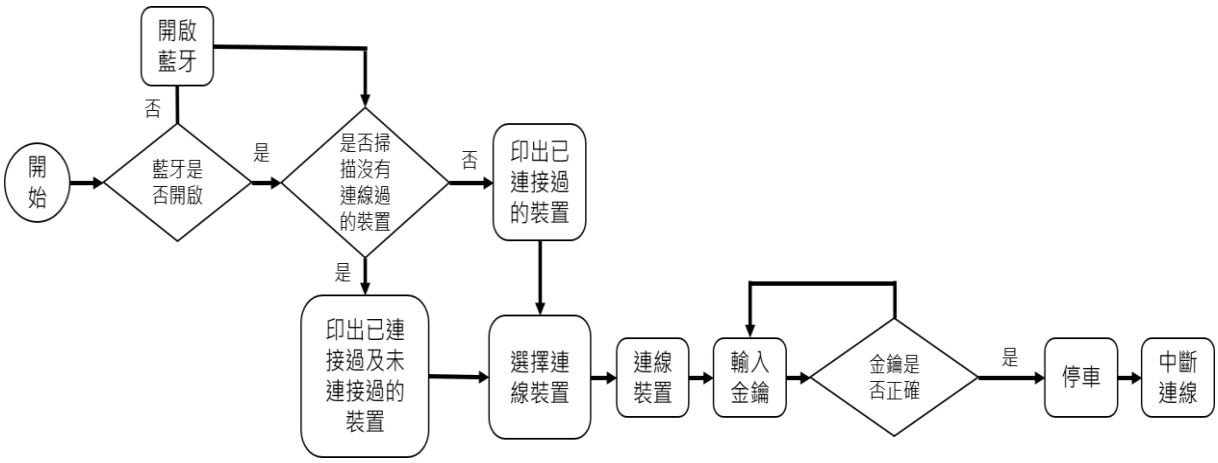

圖5、藍芽連線功能流程圖

TCP 連線功能流程如圖6：

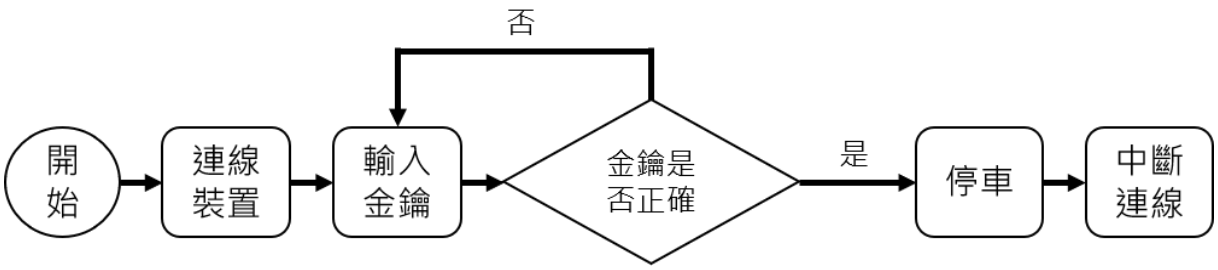

圖6、TCP 連線功能流程圖

(1) APP 簡介功能

圖7、APP 載入介面

圖8、APP 簡介1

圖9、APP 簡介2

圖10、APP 簡介3

圖11、APP 簡介4

圖12、APP 簡介5

使用者進入 APP 後，會先進入載入介面(如圖7)，這時系統會檢測使用者是否為第一次使用 APP，如果是則會自動跳入 APP 簡介介面(如圖8~12)，讓使用者了解 APP 的基本功能；反之則會跳入「未/已登入主畫面」頁面(如圖13、14)。

(2) 未登入會員介面

圖13、未登入會員介面

如果使用者沒有登入會員，就會進入未登入會員介面(如圖13)，此介面可讓使用者選擇「登入」、「註冊」、查看「簡介」、查看「關於 Smart P」等功能。

(3) 已登入會員介面

圖14、已登入會員介面

如果使用者已經登入會員就會進入已登入會員介面(如圖14)。此介面可讓使用者選擇「智慧停車」、「智慧取車」、「尋找車子」、查看「用戶資料」以及「登出」等功能。

(4) 註冊功能

圖15、車牌輸入介面

圖16、註冊資料輸入介面

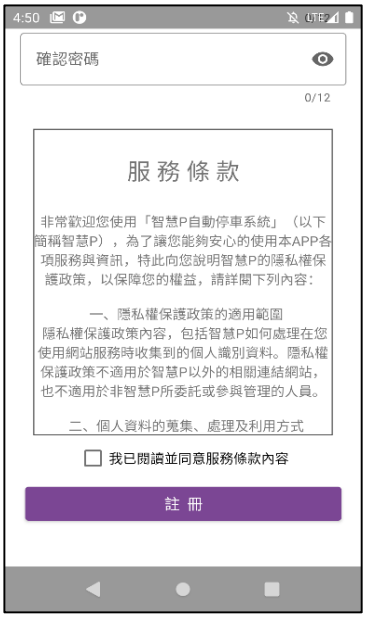

圖17、服務條款介面

圖18、手機簡訊認證介面

圖19、註冊資料確認介面

註冊功能流程如下:

I. 使用者要註冊會員時，需要先輸入使用者的車牌號碼，然後系統會先檢測此車牌號碼是否曾經被註冊過。(如圖15)

II. 如果車牌未被註冊過，則可開始填入會員基本資料(如圖16)，並在閱讀完服務條款和勾選「我已閱讀並同意服務條款內容」後(如圖17)，即可按下註冊並送出資料。其中如果註冊資料未完整填入，則會顯示該欄位的提示訊息。

III. 接下來需要做手機簡訊認證(圖18)，認證流程會在後面詳細說明。

IV. 手機簡訊認證成功，並確認完會員資料後，即可完成註冊。(如圖19)

(5) 登入功能

圖20、登入資料輸入介面

圖21、手機簡訊認證介面

登入功能流程如下:

I. 使用者登入 APP 時，需要填入車牌號碼及密碼。若使用者未曾註冊過會員或是忘記密碼，也能在下方選擇註冊或是忘記密碼的功能(如圖20)。

II. 接下來需要做手機簡訊認證(圖21)，認證流程會在後面詳細說明。

III. 手機簡訊認證成功後，即可成功登入。

(6) 忘記密碼功能

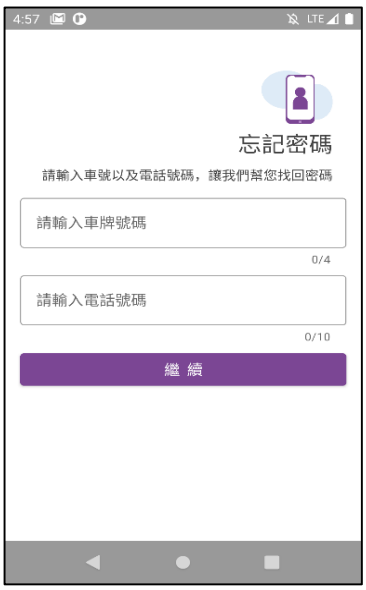

圖22、資料輸入介面

圖23、手機簡訊認證介面

圖24、成功找回密碼介面

忘記密碼功能流程如下:

I. 使用者如果忘記會員密碼時，可以利用此功能找回密碼。首先需要使用者輸入「車牌號碼」以及「電話號碼」，送出資料後，系統將會搜尋資料庫做比對。(如圖21)

II. 確認使用者身分後，再來需要做手機簡訊認證(圖23)，認證流程會在後面詳細說明。

III. 手機簡訊認證成功後，即可成功在畫面看到會員密碼。 (如圖23)

(7) 手機簡訊認證功能

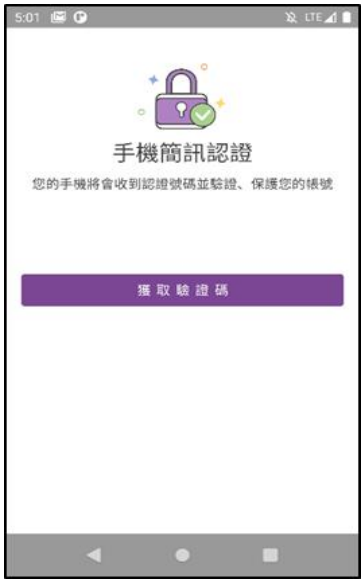

圖25、獲取驗證碼介面

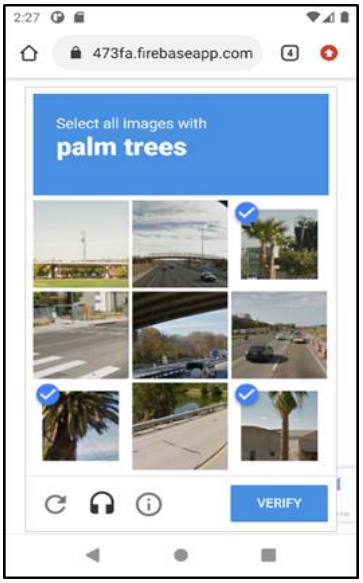

圖26、非機器人認證

圖27、輸入認證碼介面

使用者在進行諸多系統功能時，都會需要做手機簡訊認證，比如上述的登入、註冊、忘記密碼功能等。手機簡訊認證流程如下:

I. 首先，使用者需要點擊「獲取認證碼」按鈕，系統即會發送認證碼簡訊至使用者註冊時填寫的手機號碼。(如圖24)

II. 接著需要進行「非機器人驗證」。(如圖25)

III. 最後只要輸入正確的驗證碼，即可完成驗證。如果未收到簡訊也可在60秒後按下「重新獲取驗證碼」按鈕。(如圖27)

(8) 登出功能

圖28、登出提醒介面

使用者決定登出 APP 時，會跳出視窗確認是否真的要退出或是不小心按到，確認後即會返回未登入的主畫面。(如圖28)

(9) 客服功能

圖29、客服介面

使用者可以在此介面獲得此應用程式的FB、IG連結，也可以利用 E-mail 寄信至開發公司，或查看服務條款、版本號碼等。(如圖29)

(10) 智慧停車功能

圖30、狀況檢查介面

圖31、選擇連線方式介面

圖32、等待分配車位介面

圖33、成功分配車位介面

智慧停車流程如下:

I. 首先系統會檢測使用者駕駛之車輛是否已經通過車牌辨識。(如圖30)

II. 接著系統提供兩種與車子連線的方式可供使用者做選擇，分別為藍芽連線、TCP 連線(如圖31)。兩種連線流程，會在後面詳細說明。

III. 連線成功後，系統將會開始分配停車位給使用者。(如圖32)

IV. 分配車位完成後，畫面即會顯示使用者的車牌、連線方式與目標車位(如圖33)，並在使用者按下「停車」按鈕後，系統即會發送訊息通知自駕車，之後自駕車即會開始前往目標車位。

(11) 智慧取車

圖34、狀況檢查介面

圖35、選擇連線方式介面

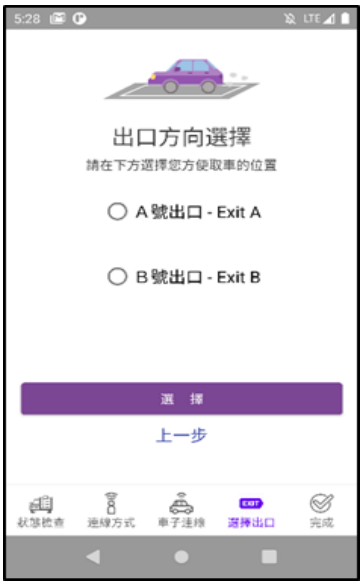

圖36、出口選擇介面

圖37、等待分配車位介面

圖38、成功分配車位介面

圖39、取車完成確認介面

智慧取車流程如下:

I. 首先系統會檢測使用者駕駛之車輛是否已經通過車牌辨識。(如圖34)

II. 接著系統提供兩種與車子連線的方式可供使用者做選擇，分別為藍芽連線、TCP 連線(如圖35)。兩種連線流程，會在後面詳細說明。

III. 連線成功後，使用者有兩個停車場出口可以選擇車輛前往取車。(如圖36)

IV. 選擇完成後，系統將會開始分配取車暫停車位給使用者。(如圖37)

V. 分配車位完成後，畫面即會顯示使用者的車牌、連線方式與目標車位(如圖38)，並在使用者按下「取車」按鈕後，系統即會發送訊息通知自駕車，之後自駕車即會開始前往目標車位。

VI. 最後使用者在取車暫停車位取得車輛後，需按下尋找車輛內的「候車區取車完成」按鈕，即可完成取車。(如圖39)

(12) 藍芽連線功能

圖40、藍芽連線功能介面

圖41、搜尋附近藍芽裝置介面

圖42、選擇裝置介面

圖43、輸入 PIN 碼介面

圖44、成功連線介面

當手機和車子在同一 RFCOMM channel 的 Bluetooth Socket 連接時，則為連線成功。手機端進行連線時，系統會執行 SDP 查找 UUID 匹配的設備，如果成功找到且設備接受連線，則共享 RFCOMM 且在連接時使用。

智慧取車流程如下:

I. 進入藍芽連線介面後(如圖40)，點選「搜尋可連線的藍芽裝置」按鈕，手機即開始搜尋附近可連線的藍芽裝置，並條列顯示至畫面上(如圖41)。

II. 點選要連接的裝置後，會彈出與是否連線的畫面(如圖42)。

III. 點選「連線」按鈕後，會跳出一個小視窗，這時需要輸入車子端的連線 PIN 碼（如圖43)，如果輸入正確，則可以開始進行連線，否則會提示「PIN 碼不正確」並關閉視窗。

IV. 連線成功後並顯示至畫面後(如圖44)，則可繼續往下執行停車及取車的服務。

(13) TCP 連線功能

圖45、TCP 連線介面

圖46、輸入 PIN 碼介面

手機和車子在同一個區域網路中，手機端輸入車子的 IP 位址和 Port 號，透過 TCP 協定進行連接(如圖45)。此時會跳出一個小視窗，需要輸入車子端的連線 PIN 碼 (如圖46)，如果輸入正確，則可以進行連線，否則會提示「PIN 碼不正確」並關閉視窗。

(14) 更改會員資料功能

圖47、更改姓名介面

圖48、更改密碼介面

圖49、更改付款方式介面

I. 使用者可隨時更改名稱。(如圖47)

II. 使用者如果要更改密碼，則需要輸入舊的密碼以及想要更改的新密碼，送出後即可更改完成，並在下次登入時使用新密碼登入。(如圖48)

III. 系統提供的付款方式共有三種：第一種為信用卡付款，使用者須輸入16碼卡號、日期以及 CVV 後即可設定成功；第二種為手機小額付款，此付款方式必須通過手機認證才能設定成功；第三種為帳單付款，輸入地址後即可設定成功。(如圖49)

(15)停車資訊查詢功能

圖50、查看車輛位置介面

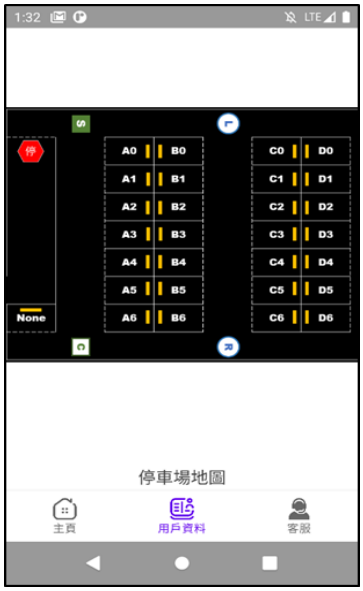

圖51、停車場地圖

圖52、停車紀錄查詢介面

I. 使用者可透過「尋找停子」來查看現在車子現在的停放位置(如圖50)。

II. 如果對於停車場位置不太熟悉的使用者，可按下「停車場介面圖」查看整個停車場的地圖(如圖51)，快速了了解車子停放位置以及方向、出口、入口…等。如果使用者在沒有停車的情況下進行查看，則會顯示使用者目前尚未停車。

III. 使用者可查詢每筆停車紀錄，內容包括：進場時間、出場時間、停車位置、停車價格、付款方式。(如圖52)

3.1.2 自駕車實作方法及功能說明

Jetbot 功能流程圖如圖53:

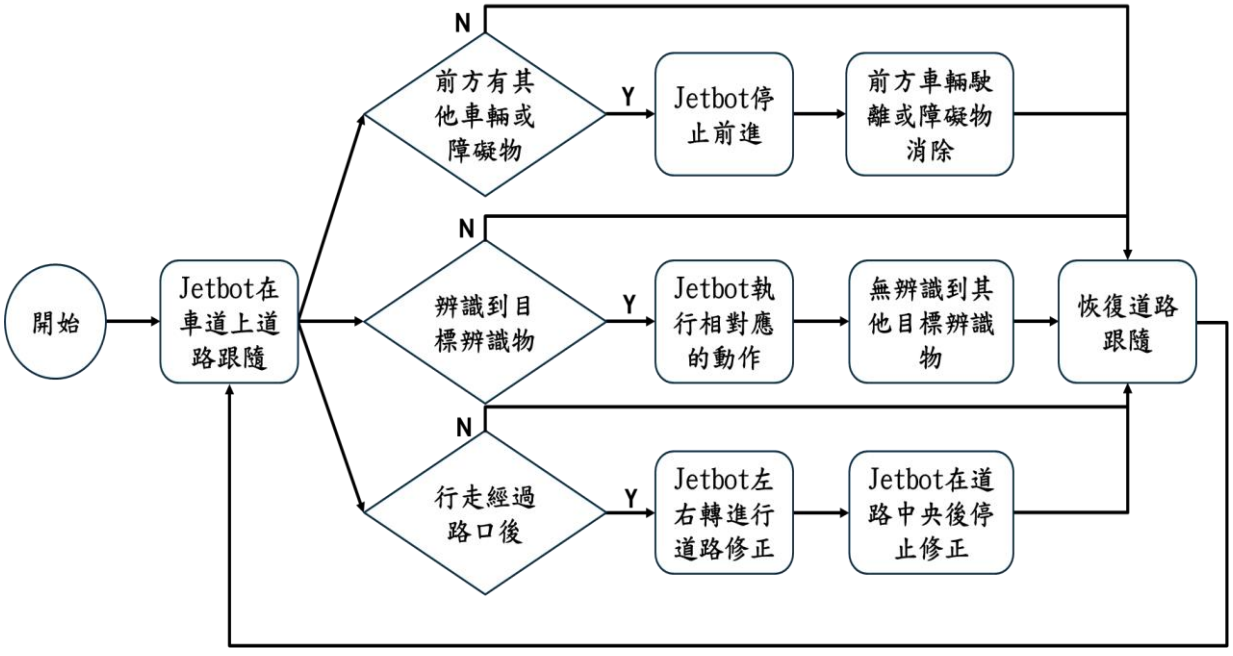

圖53、Jetbot 功能流程圖

(1)道路跟隨

功能說明:

道路跟隨是模擬車輛在道路上自動駕駛的情況，讓車輛可以遇到轉彎處即會轉彎，直線時即會直行並且不會偏離道路。

實作方法:

A. 資料搜集:

利用 JetBot 前方的攝影機拍攝影像，並即時顯示在畫面上。畫面上顯示一個前端有綠色圓點的線(如圖54)，綠色圓點所代表的是畫面的 X 軸跟 Y 軸的值，是用來之後判斷 JetBot 該朝甚麼方向行駛的資料。綠色的圓點朝向道路的中心點指引，要搜集各個方向的資料(不同的位置及角度)，直線前進時綠色圓點對準其道路的中心點，轉彎時 Y 軸數值可以調低，降低轉彎時的速度。數據變化越多是關鍵，當搜集的資料不同的角度方向越多執行的效果越好。

B. 訓練模型:

先匯入 PyTorch, NumPy, Python Imaging Library 等函式庫，並將資料集分成訓練集跟驗證集。之後定義神經網絡模型，使用 ResNet-18 model 檔來訓練資料，可自行調整訓練的回合數，訓練完成會產生一個訓練模型檔。

C. 優化訓練模型檔:

首先將模型檔轉換為 onnx 中間件，並使用 onnxsim.simplify 對轉換後的 onnx 進行簡化，再來解析 onnx 文件構建 trt 推理引擎並加載引擎執行推理、為引擎輸入、輸出、模型分配空間，最後將待推理的數據賦值給 inputs，執行 Y 推理，拿到輸出的模型檔。

D. 實際運用:

首先宣告使用 PyTorch 函式庫設定模型參數並導入模型檔及設定馬達參數。成功執行程式後，攝影機看到的畫面會根據之前 X 軸跟 Y 軸所指向的方向來指示 JetBot 做出相對應的前進方向。

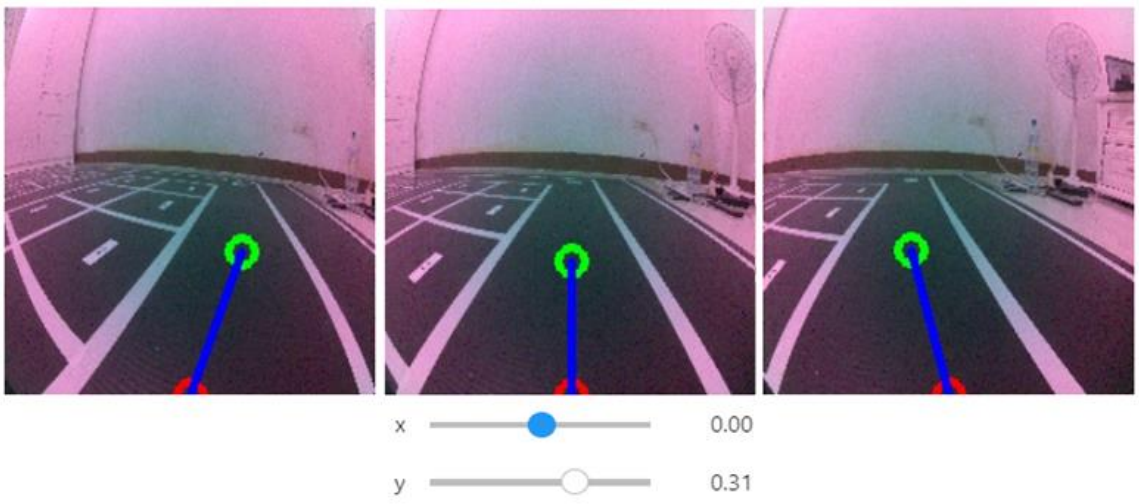

圖54、Jetbot 道路跟隨資料搜集範例圖

(2)避障

功能說明:

車輛在道路上行進時，若前方有其他車輛或障礙物時，須停止前進直到前方車輛駛離或障礙物消除後，車子才會繼續前進，以避免碰撞。

實作方法:

A. 資料搜集:

利用 JetBot 前方的攝影機拍攝影像，並即時顯示在畫面上。再來建立一個 dataset 資料夾裡面有 blocked 跟 free 兩個資料夾，用來儲存等等要拍照分類的照片。之後建立一個增加 free 類別跟增加 blocked 類別的按鈕，並顯示該類別照片的數量。最後我們將決定甚麼是障礙物甚麼是可自由行走的分類，並將 JetBot 放置在違反其“安全行走”的場景中，將這些場景拍照並添加分類為 blocked;放置在可以安全向前行走的場景中，將這些場景拍照並添加分類為 free(如圖55)。

B. 訓練模型:

首先使用深度學習庫 PyTorch 及多種 PyTorch 函式庫來訓練模型，再來將數據集拆分為訓練集和驗證集，之後設定完訓練集和驗證集的參數及訓練的回合數後，就可以開始訓練，訓練完成會產生一個訓練模型檔。

C. 實際運用:

首先宣告使用 PyTorch 函式庫設定模型參數並導入模型檔及設定馬達參數。成功執行程式後，攝影機看到的畫面會根據之前拍攝 free 或是 blocked 的模型檔來判斷 blocked 的機率。規範 blocked 之後的動作，就可讓車輛達到避障的效果，並且可以利用調整 blocked 的機率來指定車輛停止於障礙物前的距離。

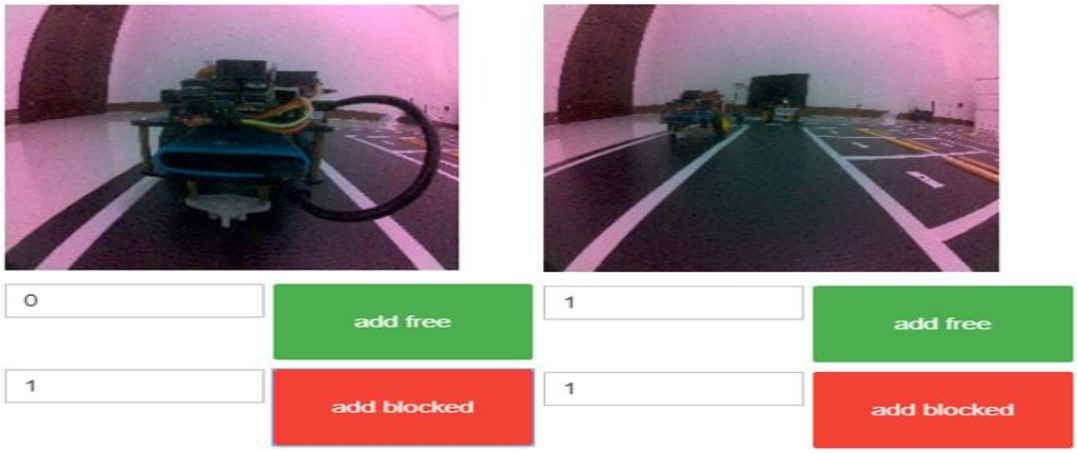

圖55、Jetbot 避障資料搜集範例圖

(3)物件辨識

功能說明:

車輛在真實的道路上行走時，必須要遵守道路上的各式標誌，來維持行車順利。利用物件辨識的技術，可讓 Jetbot 利用鏡頭辨識地圖上各種不同的物件，並且可以編寫程式來指定 Jetbot 辨識出不同物件後，執行各種不同的動作，即可呈現相似於真實世界的效果。

實作方法(YOLOv4物件辨識):

A. 決定好目標辨識物件:

本專題共有九種目標辨識物:六個路口辨識標誌、入口停車標誌、停車格編號標誌、長條型停車格檔。

B. 資料搜集:

利用 JetBot 前方的攝影機拍攝包含目標辨識物件的影像，要搜集各個方向拍攝的資料(不同的位置及角度)，以利於訓練及提高成功辨識率。

C. 人工影像標記:
除了看出圖片中的物體，電腦需要知道該物體在圖片中的位置，因此運用影像標記軟體將圖片中的物體 label(標示)起來讓電腦去學習。

D. 訓練模型:
使用 YOLOv4 tiny model 來做訓練，並將數據集拆分為訓練集和驗證集，訓練完成後將得到一個訓練效果最好的權重檔。

E. 實際運用:
使用時，需載入權重檔、cfg 檔和辨識物的類別檔，之後即可利用 Jetbot 鏡頭來辨識物件，若有辨識到目標物件，則會在畫面上顯示 Bounding box。(如圖 56)

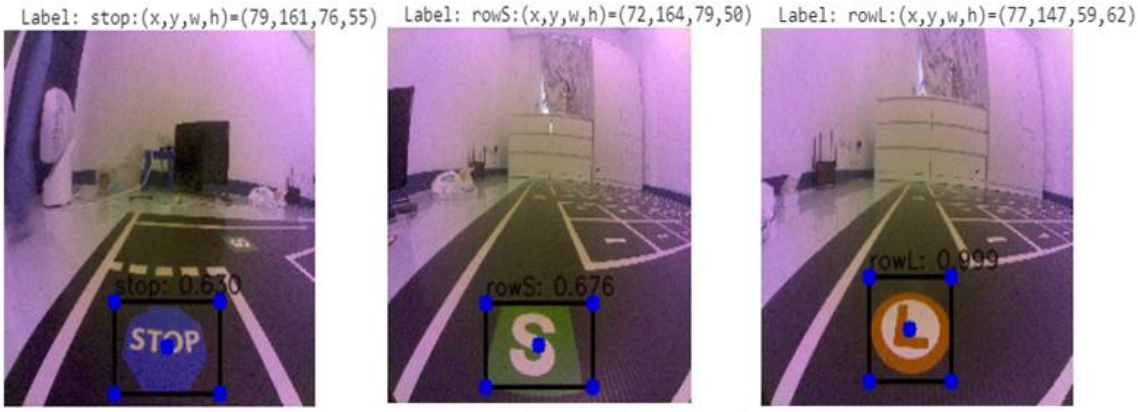

圖56、YOLOv4物件辨識範例圖

(4)行經路口後道路修正

功能說明:

Jetbot 在行經路口並轉向其他道路後，因為摩擦力和電量的不同，而造成每次都會呈現不同角度的轉向，因此在轉向後，必須讓 Jetbot 根據轉向後的道路，做道路修正，讓 Jetbot 維持在車道的中心，並且車頭平行於道路。

實作方法(不斷捕捉影像並重複以下步驟直到修正完成):

A. 影像前處理(灰階、高斯濾波、邊緣檢測)。

B. 將影像分割成左右圖片，並算出左右輪廓數。(如圖57)

C. 找出左圖最右輪廓的 X 座標和右圖最左輪廓的 X 座標，然後得出 bias (相加除以二)。

D. 依據 bias 來決定直走或往右/左修正。

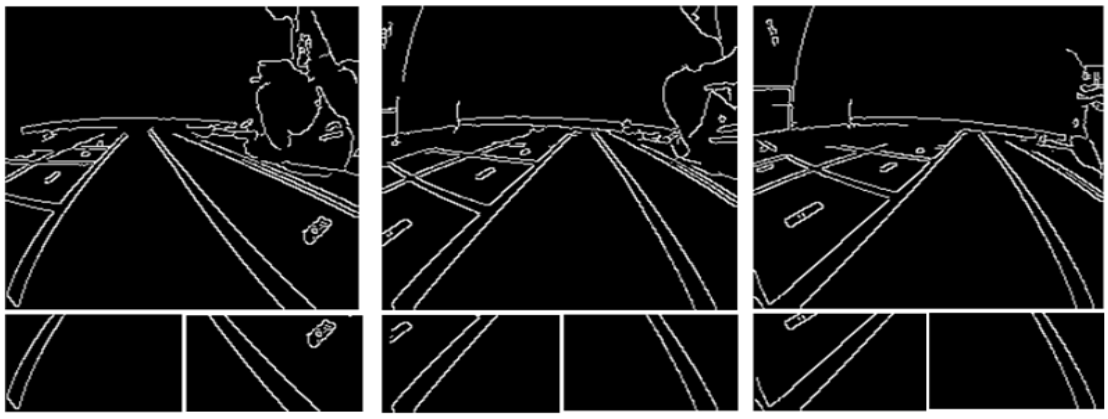

圖57、道路修正前處理範例圖

3.1.3 停車場的規劃與辨識說明

(1)停車場規劃(如圖58)

圖58、停車場規劃圖

我們停車場的規劃有幾個部分，首先是圖58左下角的入口處，要停車的車主只要將車子開到入口處停車，車子就能在通過車牌辨識系統後自動進入停車場，前往由系統分配給它的停車位；我們做的這個地圖總共有二十八個停車位，以及出口一和出口二的各三個取車暫停車位，這六個取車暫停車位的用途是假設車主想要取車時，車子會自動開往車主所指定的出口附近的取車暫停車位來等候車主取車離開。

(2)鏡頭辨識

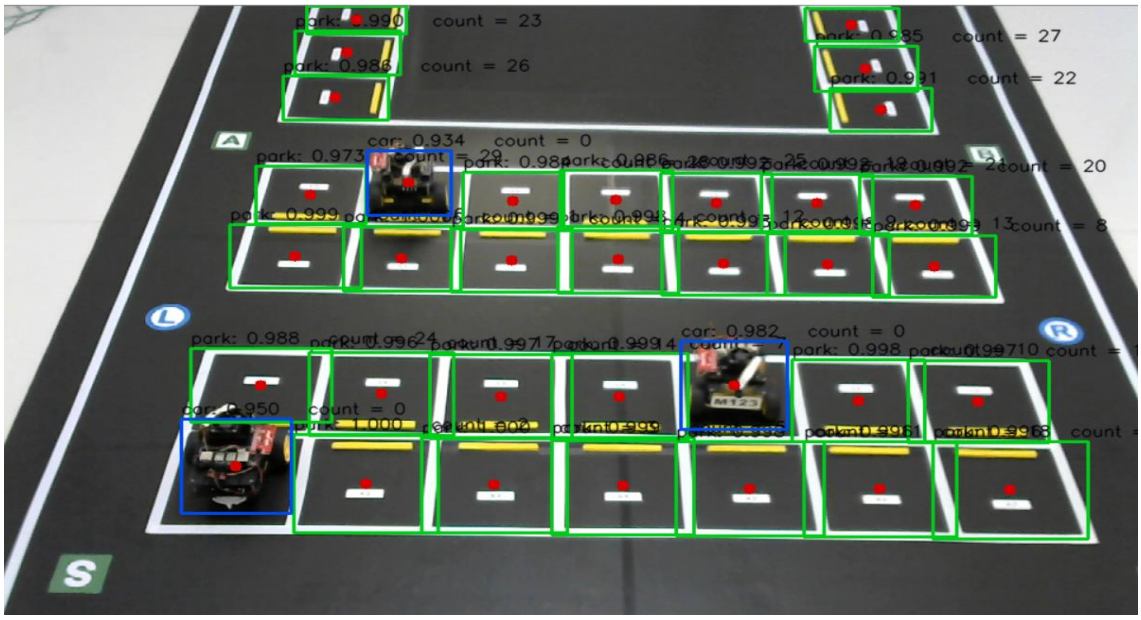

圖59、鏡頭辨識實作圖

利用事先蒐集好由鏡頭端拍攝的照片，讓電腦訓練並學習辨識特定的物件(YOLOv4物件辨識)，最後即可成功辨識並框出車子以及停車位(如圖59)，之後再對車位進行排列，幫每個停車位給予自己的編號，並將座標與狀態儲存至雲端資料庫。

(3)系統導航

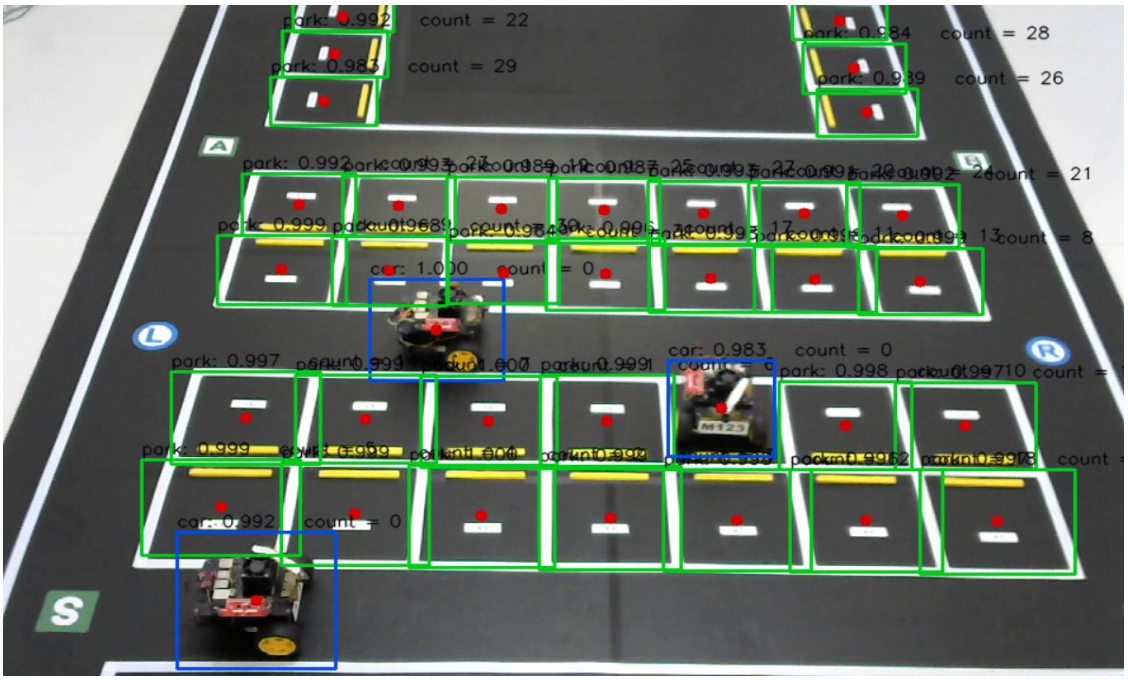

圖60、系統導航辨識實作圖

我們的系統導航，主要將停車跟取車分為三個區塊，分別是 A 車道，B 車道，以及 C 車道，三個區塊都會各自獨立執行停車以及取車，鏡頭端透過分析畫面中指定車位以及車子的 XY 中心座標(如圖60)，將要進入停車場的車子導向至系統分配的車位，並將要離開停車場車的車子導向至駕駛所選出口的取車暫停車位。

3.1.4 車牌辨識實做功能說明

車牌辨識流程如圖 61:

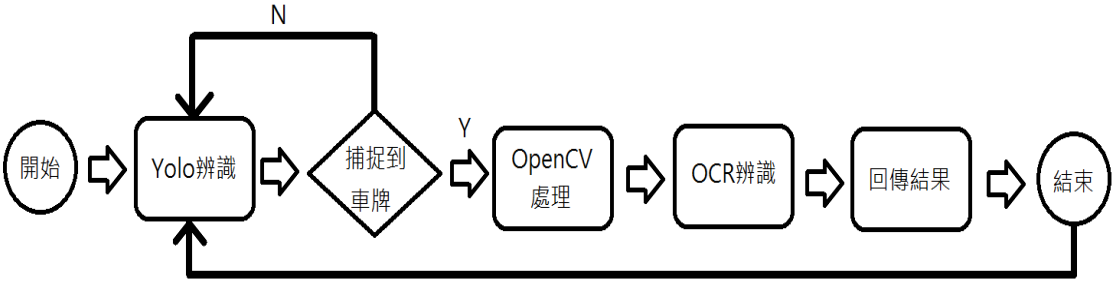

圖 61、車牌辨識流程圖

(1)YOLOv4 辨識

功能說明:

使用 YOLOv4 辨識車牌以及截圖，並對截圖大小標準化(每張長寬皆一致，以減小干擾)(如圖 62)。

實作方法:

A. 訓練模型:

針對性訓練辨識專用權重檔，以期讓辨識結果更好。

B. 實時辨識:

在迴圈中完成整個流程，讓程式不間斷的偵測入口處有無來車，避免產生漏，在辨識成功時截圖以做接下來的處理。

(2)OpenCV 處理

功能說明:

使用 OpenCV 函式庫相關功能對圖片進行處理，把與辨識無關或影響辨識的雜訊清除，使辨識結果更準確。

實作方法:

A. 灰階 : 把圖片的色彩去除，只留下需要的資訊。(如圖 63)

B. 高斯模糊 : 降低圖片的雜訊和細節層次。(如圖 64)

C. 二值化 : 將文字和背景之間的對比拉大以更容易被辨識到。(如圖 65)

圖 62、YOLOv4 辨識

圖 63、灰階處理

圖 64、高斯模糊處理

圖 65、二值化處理

(3)OCR 光學字元辨識

功能說明:

使用 Tesseract 對處理好的圖片進行辨識，並在回傳結果前先判斷是否為正確車牌格式。(如圖 66、67)

實作方法:

A. 標準訓練檔:

使用 Google 提供的英文辨識訓練檔，便可辨識所有車牌號碼相關資訊。

B. Tesseract 函式庫:

使用相關函式功能將辨識結果以字元字串的格式顯示出來。

C. 回傳結果:

將結果傳回 firebase，以做後續相關判斷。

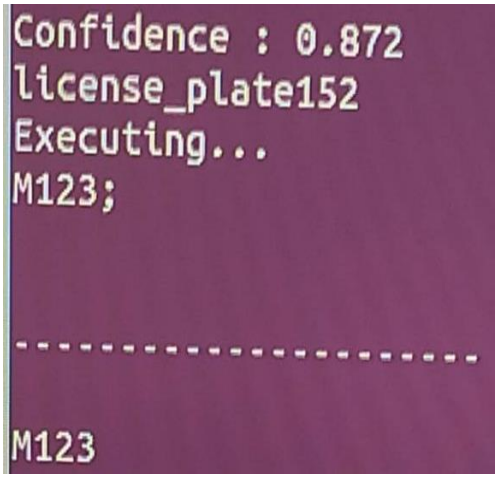

圖 66、原始辨識結果

圖 67、程式調整後結果

3.1.5 其他週邊顯示功能說明

七段顯示器、紅綠燈、ESP8266、Arduino Uno 流程如圖68:

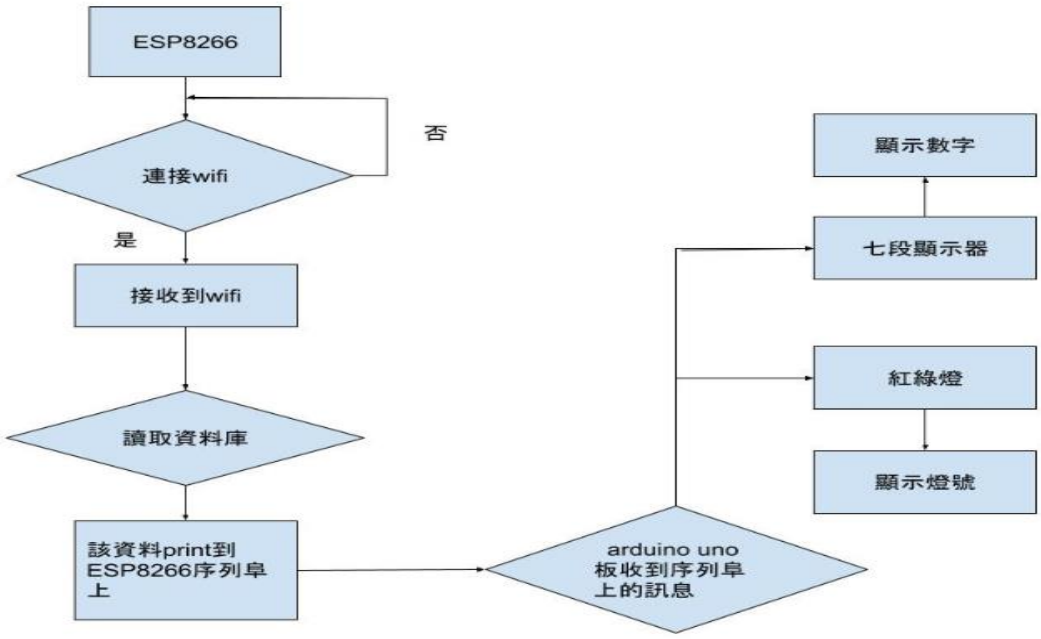

圖68、週邊顯示功能流程圖

(1) 七段顯示器:顯示剩餘車位。

(2) 紅綠燈:用來代替停車場入口柵欄，紅燈指示禁止進入，綠燈指示可以進入。

(3) ESP8266:接在 Arduino Uno 板上，用來連接 wifi。

(4) Arduino Uno:用來連接以上三者配件，可燒入程式進去。

3.1.6 資料庫實做儲存說明

本系統運用之資料庫總共分為三大項(如圖69)：

(1) 停車場現況資料（如圖70）

I. 車道(1-4)

A. 此車道的佔用情況

B. 車輛即將前往的目標停車位

C. 車輛是否已到達目標停車位

II. 紅綠燈顯示(顯示有車將進入停車場)

III. 總剩餘車位

(2) 會員資料（以車牌作為母節點）(如圖71)

I. 密碼

II. 手機

III. 帳號(車牌)

IV. 付款方式

V. 此車輛是否在停車場內

VI. 停車紀錄(入場時間、出場時間、停車位置、停車費用)

(3) 停車位資料（以車位編號作為母節點）(如圖72)

I. 此車位的 X 座標

II. 此車位的 Y 座標

III. 此車位的現況(是否有車停入)

IV. 此車位現在停入車輛的車牌號碼

資料庫實際命名與運用：

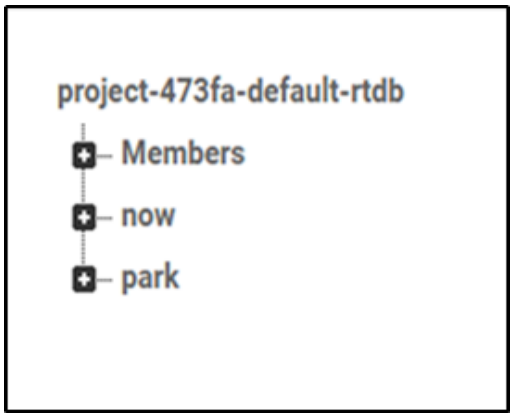

圖69、系統資料表

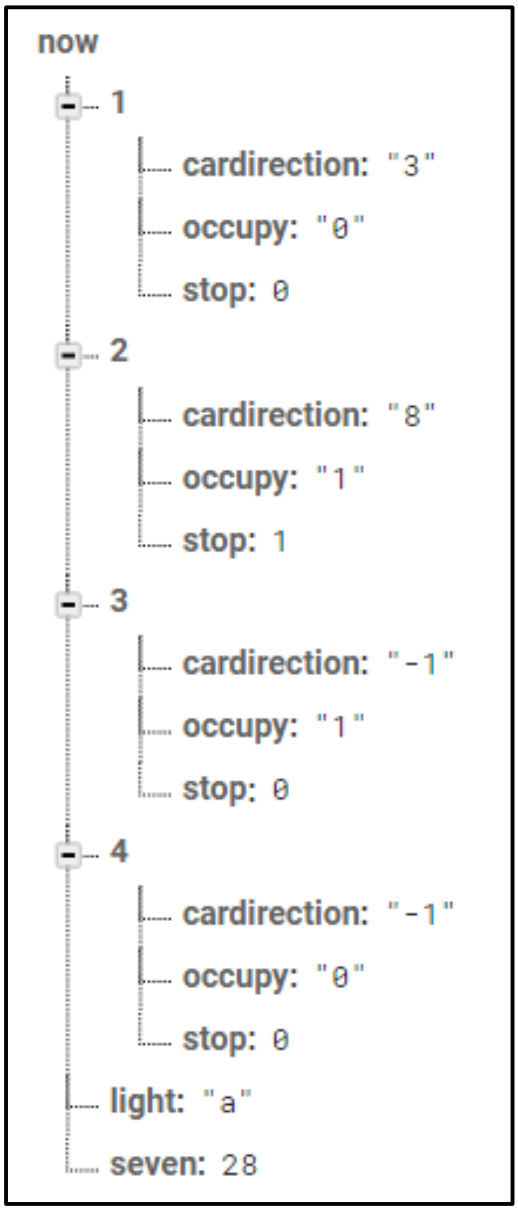

圖70、停車場現況資料

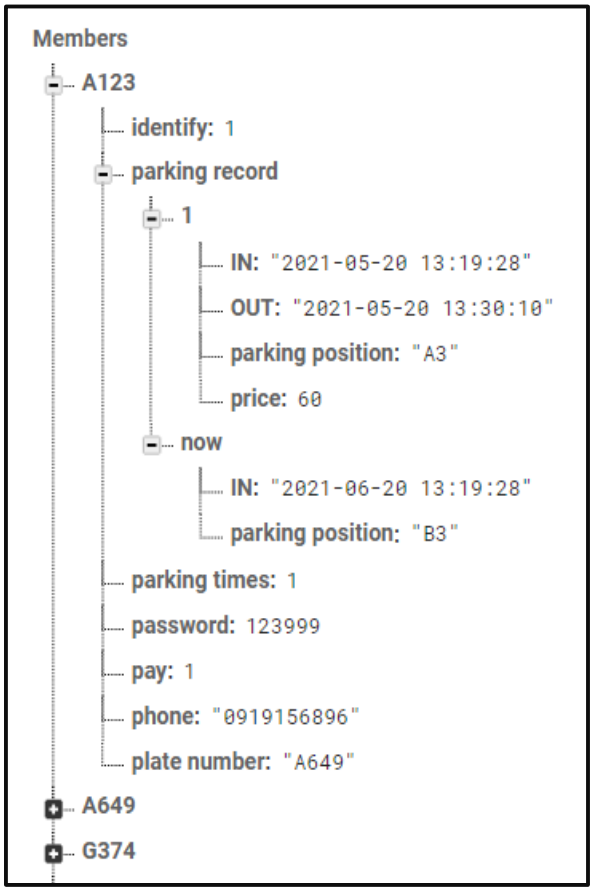

圖71、會員資料

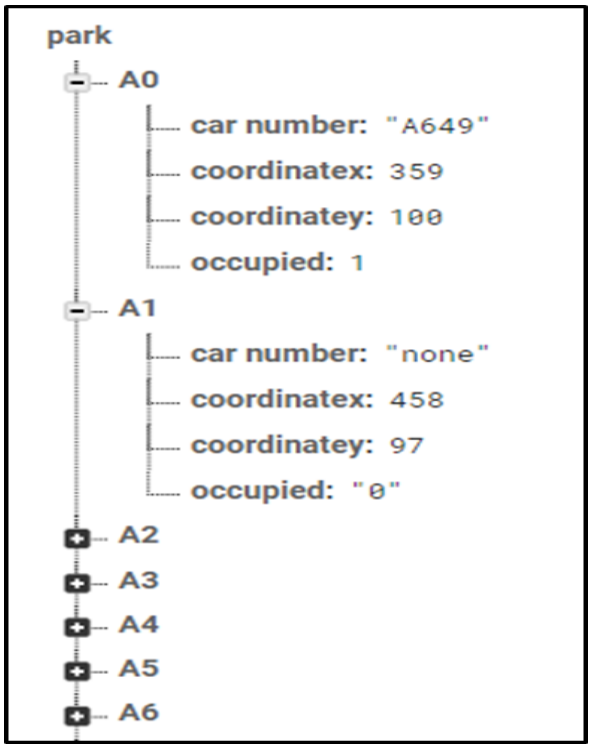

圖72、停車位資料

3.2 完整停車與取車流程

3.2.1 停車系統流程圖

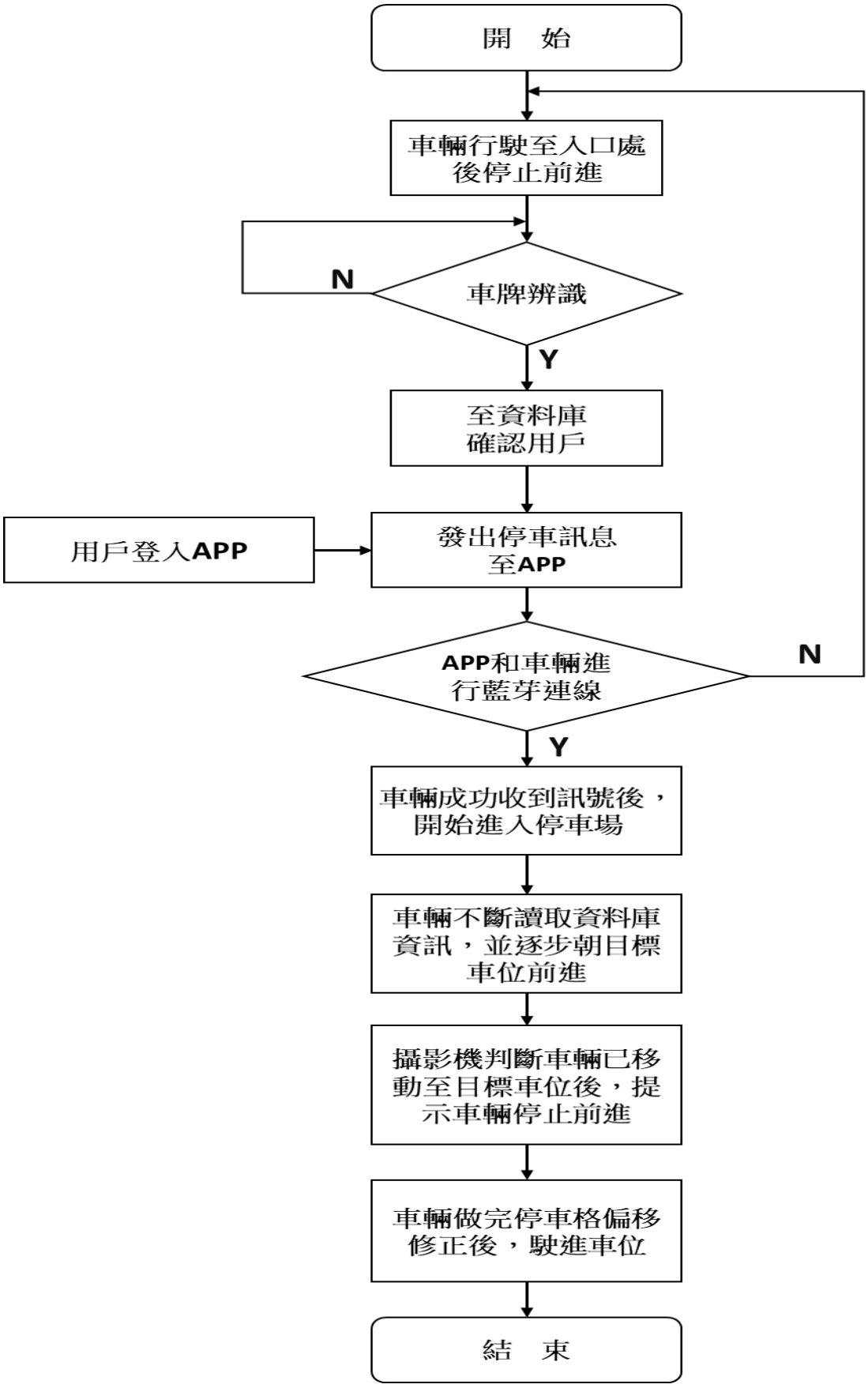

圖73、停車系統流程圖

3.2.2 取車系統流程圖

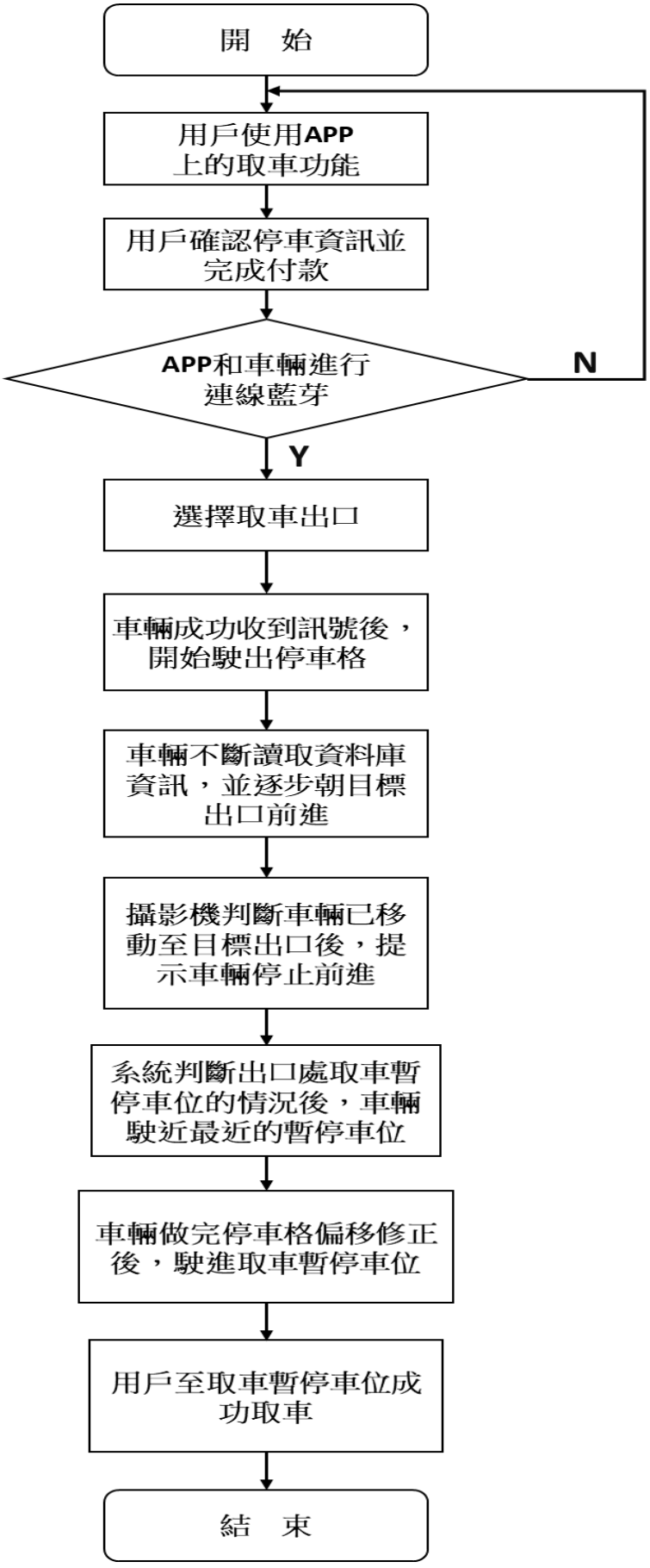

圖74、取車系統流程圖

四、結論

4.1 系統未來展望

本專題的製作是利用小型的人工智慧小車，並在我們規劃好的地圖上行走，因此與現實中的自駕車，還是有許多的差距和細節我們無法實現。首先，對於APP 和自駕車的連線，我們還需克服距離與網路連線上的困難。再來，對於外部攝影機辨識車位及車輛時，偶爾會有辨識死角，因此我們還需再架設多部攝影機，交叉辨識來獲得更加穩定的結果。最後，我們還需提升能夠同時進行停車及取車的數量及效率，來避免車輛阻塞的情況發生。

4.2 專題製作心得

透過這次的專題製作過程，讓我們能夠實際運用課堂上學過的技術與知識，並且當我們遇到未曾學過的難題時，也能試著透過尋找網路上的資源，並與組員們合作思考，來共同解決困難。除此之外，在一次次的專題討論過程，組員們的表達能力也有明顯的提升，雖然偶爾對於專題的實作方法會有意見上的衝突，但也因此，在爭論的過程中，我們往往能注意到各自看法上的缺陷與不足，並在彙整了大家的想法後，使我們的專題更加完整。

五、誌謝

在此特別感謝專題指導老師丁德榮教授，在專題製作的這段時間裡，不斷的提供我們許多專題相關的各種資訊，每當我們遇到困難時，也會給予我們意見和想法，並且常常督促我們應該將專題的各種大小細節都考慮清楚，實作成果才會更加完整。有了教授的幫助，才使能我們在剛接觸專題時，更快進入狀況，並在最後能順利的完成專題。另外，我們也想感謝系辦助理曾雪寶小姐，在各種設備、教室的借用上，都給予了我們非常多的幫助，使我們能夠沒有後顧之憂的專注於專題製作上。

六、參考文獻

[1] Android Studio

Android App 程式設計教本之無痛起步:使用 Android Studio 開發境(ISBN 978-986-312-259-3)

[2] Android Studio 开发教程
https://www.youtube.com/channel/UCR0wEta3XtmzoLDwUH4Eo4g

[3] Firebase 介紹與建立專案
http://dog0416.blogspot.com/2018/06/firebase-firebase.html

[4] 使用電話號碼在 Android 上使用 Firebase 進行身份驗證
https://firebase.google.com/docs/auth/android/phone-auth

[5] YOLO 背景介紹
https://www.itread01.com/content/1544112970.html

[6] YOLOv4詳細介紹

https://medium.com/ching-i/yolo%E6%BC%94%E9%80%B2-3-yolov4%E8%A9%B3%E7%B4%B0%E4%BB%8B%E7%B4%B9-5ab2490754ef

[7] YOLOv4 訓練教學
https://medium.com/ching-i/yolo-c49f70241aa7

[8]Pytorch 簡介

https://ithelp.ithome.com.tw/articles/10244106

[9]手把手教你用 PyTorch 快速準確地建立神經網絡(附4個學習用例)
https://www.ipshop.xyz/1250.html

[10]Nvidia 開源高效能推理平臺 TensorRT 函式庫元件
https://www.ithome.com.tw/news/131710

[11]OpenCV 維基百科
https://zh.wikipedia.org/wiki/OpenCV

[12] Tesseract-OCR 實用心得
https://b98606021.medium.com/%E5%AF%A6%E7%94%A8%E5%BF%83%E5%BE%97-tesseract-ocr-eef4fcd425f0

[13]光學字元辨識
https://zh.wikipedia.org/wiki/%E5%85%89%E5%AD%A6%E5%AD%97%E7%AC%A6%E8%AF%86%E5%88%AB

[14] Logitech Webcam C270
https://www.logitech.com/zh-tw/products/webcams/c270-hd-webcam.960-000626.html

[15] Jetson Nano 開發套件

https://robotkingdom.com.tw/product/nvidia-jetson-nano-developer-kit-b01/

[16] SparkFun JetBot

https://www.taiwansensor.com.tw/product/sparkfun-jetbot-ai-kit-jetson-nano-%E4%BA%BA%E5%B7%A5%E6%99%BA%E6%85%A7%E5%B0%8F%E8%BB%8A%E9%96%8B%E7%99%BC%E5%A5%97%E4%BB%B6-nvidia-%E6%8E%A8%E8%96%A6/

[17]Arduino/簡介
https://zh.m.wikibooks.org/zh-tw/Arduino/%E7%AE%80%E4%BB%8B