

逢 甲 大 學
通 訊 工 程 學 系
專 題 期 末 報 告 書



A I 導 覽 輔 助 系 統
AI Navigation Assistance System

專題指導教授：陸清達 博士

專題生：D1189004（通訊四乙周楷歲）

D1180714（通訊四乙龔乙軒）

D1180727（通訊四乙張詩揚）

D1179873（通訊四乙徐東寅）

D1180050（通訊四乙卓宏軒）

中華民國一一三年十二月十一日

摘要

本研究旨在開發一套結合 5G 網路與人工智慧技術的智慧校園導覽系統，以解決現有導覽系統依賴人力資源的缺點。該系統實現了一項名為「AI 導覽輔助系統」的功能，利用 LiDAR 技術進行精確的環境感知與定位，增強校園內的自主導航能力，從而提升導覽效率與安全性。針對疫情後的新常態需求，本系統不僅提供即時互動的導覽體驗，還結合智慧頭盔實現語音指令操作，讓使用者能以更便捷的方式與導覽車互動。

此系統旨在提升校園導覽的便利性，解決導覽人力不足問題，並推動低碳永續發展，實現智慧化校園導覽的目標。實驗結果顯示，本系統在導航精確度、系統互動性、5G 通訊技術、使用者體驗與低碳效益方面表現卓越。系統互動性測試結果顯示，智慧頭盔的語音指令具高準確度且反應迅速，即使在多種噪音環境下亦能穩定運行。5G 網路的低延遲與廣覆蓋性對導覽車的即時決策影響甚微，遠距監控功能穩定。使用者體驗調查顯示，受測者對即時導覽信息的滿意度極高。低碳效益方面，相較於傳統人力導覽系統，本系統顯著降低碳排放，節能減碳效果顯著。

以上結果驗證了本系統的實用性與效益，並為後續優化提供了改進依據，促進智慧校園導覽系統的進一步發展。

目錄

| | |
|-------------------|----|
| 第一章 研究背景與動機..... | 3 |
| 1.1 研究背景 | 3 |
| 1.2 研究動機 | 3 |
| 第二章 基本原理簡介 | 4 |
| 2.1 原理介紹 | 4 |
| 2.2 系統架構 | 15 |
| 2.3 系統流程圖 | 17 |
| 第三章 系統設計與研究 | 20 |
| 3.1 開發工具 | 20 |
| 3.2 硬體簡述 | 21 |
| 第四章 研究結果 | 22 |
| 4.1 系統結果圖 | 22 |
| 4.2 時程規劃 | 31 |
| 4.3 組員分工 | 32 |
| 4.4 專題獲獎成果 | 33 |
| 第五章 結論 | 34 |
| 第六章 參考文獻 | 35 |

圖目錄

| | |
|---|----|
| 圖 1 語音辨識成果圖；(a)語音辨識 GUI 介面、(b)辨識結果 | 5 |
| 圖 2 卷積神經網路；(a)滑動與運算、(b)池化層特徵取值 | 6 |
| 圖 3 訓練損失軌跡圖 | 7 |
| 圖 4 Raspberry Pi 作業系統；(a)系統運行畫面 、(b)ROS 程式運行 | 8 |
| 圖 5 地圖建模；(a)環境點雲數據、(b)2D 地圖生成..... | 11 |
| 圖 6 全局路徑規劃示意圖 | 12 |
| 圖 7 局部路徑規劃示意圖；(a)軌跡預測、(b)避障模擬..... | 13 |
| 圖 8 導覽車操作實景圖；(a)車輛運行示意圖、(b)前視鏡頭畫面..... | 14 |
| 圖 9 導覽車系統架構圖 | 16 |
| 圖 10 智慧頭盔系統架構圖 | 16 |
| 圖 11 導覽輔助系統流程圖 | 18 |
| 圖 12 智慧頭盔系統流程圖 | 19 |
| 圖 13 導覽輔助系統；(a)車輛實體圖、(b)實際運行畫面 | 24 |
| 圖 14 智慧頭盔系統；(a)智慧頭盔實體圖、(b)使用者穿戴實例圖 | 25 |
| 圖 15 建築物辨識結果；(a)有樹木遮擋、(b)正面、(c)右側 | 26 |
| 圖 16 導覽輔助系統 GUI 介面 | 27 |
| 圖 17 智慧頭盔；(a)App GUI 、(b)OLED 顯示、(c)初始畫面(d)指令發送 | 28 |
| 圖 18 地圖建模；(a)實際運行環境、(b)點雲數據、(b)2D 地圖建模 | 29 |

表目錄

| | |
|-----------------------|----|
| 表 1 導覽車功能表 | 15 |
| 表 2 智慧頭盔系統功能 | 16 |
| 表 3 導覽輔助系統操作流程 | 17 |
| 表 4 智慧頭盔系統操作流程 | 19 |
| 表 5 系統開發環境 | 20 |
| 表 6 硬體需求表 | 21 |
| 表 7 系統操作流程表 | 23 |
| 表 8 智慧頭盔操作流程表 | 24 |
| 表 9 作品與市面相關產品差異 | 29 |
| 表 10 專題開發之時程表 | 31 |
| 表 11 專題開發之甘特圖 | 32 |
| 表 12 合作分工情形如下 | 32 |

第一章 研究背景與動機

1.1 研究背景

隨著科技的迅速發展，智慧創新已成為全球教育改革的重要趨勢之一。疫情所引發的生活新常態，使校園導覽與訊息傳遞的安全性與效率成為亟待解決的核心議題。在此背景下，結合 5G 通訊與人工智慧技術的智慧化導覽系統逐漸受到廣泛關注，並被視為解決上述挑戰的創新方案。這類系統不僅能實現即時訊息傳遞，還能顯著提升校園導覽的效率與安全性。

LiDAR 技術以其卓越的環境感知能力，在自駕車與機器人導航等領域得到了廣泛應用。透過 LiDAR 進行精準定位與地圖建構，不僅可大幅提升自主導航的可靠性與準確性，還能有效應對校園環境中的複雜場景，進一步增強智慧導覽系統的實用價值。

1.2 研究動機

我們希望透過智慧科技，解決現有導覽系統中定位不準確與導航不穩定等問題，全面提升其便利性與互動性。「AI 導覽輔助系統」採用自駕車技術，不僅能有效緩解導覽員短缺的問題，還能藉由電力驅動顯著降低碳排放，助力永續發展。此外，系統結合智慧頭盔技術，讓使用者能以語音指令進行直觀操作，不僅大幅提升使用者體驗，還能靈活應對多種環境中的複雜需求，實現智慧化與自主化的導覽服務。

第二章 基本原理簡介

2.1 原理介紹

2.1.1 語音辨識技術原理

本系統基於 MATLAB 平台實現語音指令控制導航車，並利用 detectspeech 技術進行語音活動檢測，確保系統僅處理語音區段。運作過程可分為四個步驟：語音信號處理、語音活動檢測、語音轉文本、指令解析與執行，以及自走車控制。

首先，系統利用 Kamera Ai 麥克風接收語音信號，並透過 MATLAB 音頻處理工具進行信號處理，抑制雜訊並提取有效音訊。接著，使用 detectspeech 技術檢測語音活動[1]，識別出語音區段並將其提取出來。隨後，利用 Speech-to-Text 技術將語音轉換為文本，並結合自然語言處理技術解析語音指令。

當文本生成後，系統將指令與預定義命令匹配，例如「前往圖書館」或「去紀念館」，並依照匹配結果執行相應的導航指令。最後，系統控制導航車移動至指定目的地，確保指令準確執行。

為提升系統的準確性與靈活性，我們還可訓練自定義語音辨識模型，適應不同語言、口音或背景噪音，並進一步增強對複雜指令的解析能力。本系統展示了語音辨識技術在自主導航車控制中的應用，具備廣泛的實用價值與發展潛力，並可應用於各類自動化與智慧系統。

2.1.2 語音辨識技術應用

本研究開發了一套基於語音辨識的智慧導覽系統，旨在提升使用者互動體驗。當手機應用程式偵測到語音訊息時，系統會自動啟動語音辨識功能。使用者按下「START」後，系統透過 Kamera Ai 麥克風接收語音信號，並利用 MATLAB 的音頻處理工具進行信號分析，抑制雜訊並提取有效語音。接著，系統使用 `detectspeech` 技術進行語音活動檢測[1]，識別語音區段並轉換為文本，並結合自然語言處理技術解析指令。

辨識出的指令將透過 UDP 協議傳送至智慧頭盔中的 Raspberry Pi 終端，實現即時執行，確保使用者能迅速獲得反饋。我們在多種環境與背景噪音下測試系統，結果顯示，系統仍能保持高準確度，證明其穩定性與實用性。為提高系統可靠性，我們使用多樣化語音數據集，涵蓋不同口音與語速，強化辨識能力。系統亦可訓練自定義語音模型來提升辨識準確度，展示語音辨識技術在智慧導覽車中的應用潛力。



圖 1 語音辨識成果圖；(a)語音辨識 GUI 介面、(b)辨識結果

2.1.3 影像辨識技術原理

YOLOv4 (You Only Look Once version 4) 是一個高效且強大的物件偵測模型，結合了多種先進的優化技術，如 CSPNet (Cross Stage Partial Network)、Mish 激活函數及資料增強，顯著提升了偵測效能和運算速度。

YOLOv4 的架構主要分為三個部分：Backbone、Neck 和 Head。Backbone 使用 CSPDarknet53 進行特徵提取，能夠捕捉影像中的多層次特徵，從而提升偵測準確度。Neck 部分融合來自不同尺度的特徵，確保演算法在處理各種大小物件時的穩定性與效能表現。Head 則負責最終的物件識別與定位[2]。

在深度學習領域，CNN (Convolutional Neural Network) 因其卓越的影像辨識能力而廣泛應用。其結構包含卷積層、池化層及全連接層，這些技術在 YOLOv4 的 Backbone 中得到了充分應用，進一步增強了其在影像處理與物件偵測方面的能力，成為當前物件偵測領域的關鍵工具[2]。

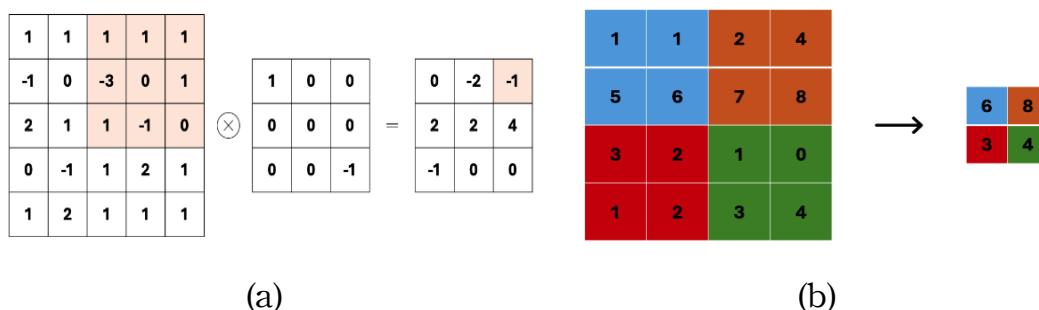


圖 2 卷積神經網路；(a)滑動與運算、(b)池化層特徵取值

2.1.4 影像辨識技術應用

本研究利用 YOLOv4 訓練模型辨識校園特定建築物，實驗結果顯示該演算法具備高效且精確的檢測能力。為實現「AI 導覽輔助系統」的自動地標偵測，我們使用約 4500 張訓練圖像，成功訓練模型辨識三座目標建築物，並選擇 CSPDarknet53 作為 YOLOv4 的 Backbone，以提升特徵提取能力[3]。

在訓練過程中，損失函數如圖 3 所示，從第 3 個 Epoch 開始逐步收斂，並隨著訓練次數增加，訓練與驗證損失持續下降，顯示準確率逐漸提升。數據集的分配為：訓練集佔 70%，驗證集佔 20%，測試集佔 10%。初始學習率設為 0.0001，最大訓練迭代次數為 20，每 5 次 Epoch 儲存一次模型，並每 100 次迭代進行驗證，以防止過擬合。訓練結果顯示，YOLOv4 在校園建築物識別任務中表現優異，成功達成所需的準確度與穩定性。

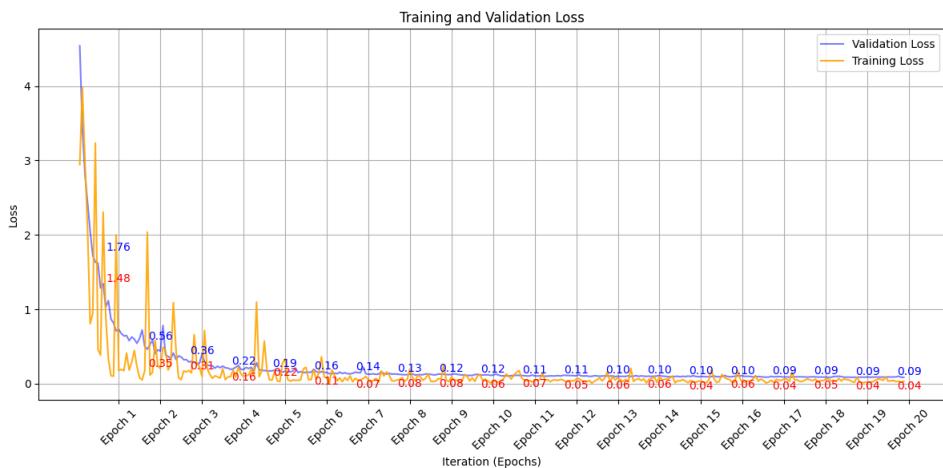


圖 3 訓練損失軌跡圖

2.1.5 ROS Noetic 與 Ubuntu 20.04 的搭配選擇

在 Raspberry Pi 平台上運行 ROS Noetic 時，選擇 Ubuntu 20.04 具有明顯的技術優勢。雖然 ROS Noetic 官方支援的是 Ubuntu 20.04，作為長期支援（LTS）版本的 Ubuntu 20.04 提供了更穩定的運行環境和資源優化，特別適合導覽車應用[4]。其 APT 套件管理系統簡化了軟體安裝並支援更高效的資源管理，適合長期運行。

從硬體支援來看，Ubuntu 20.04 對 ARM 架構的優化，尤其是對 Raspberry Pi 4B 的支援，提供更強的 GPU 加速和更好的多核心處理效能，適合高負載應用（如導覽車的運算與視覺處理）。更新的 Linux 內核確保硬體穩定性，長時間運行下不會過多錯誤。

儘管 ROS Noetic 針對 Ubuntu 20.04 進行官方支援，但社群提供了大量解決方案，使其在 20.04 上運行仍具可行性與穩定性。總體而言，Ubuntu 20.04 的穩定性、硬體支援與軟體資源，使其成為 Raspberry Pi 上運行 ROS Noetic 的理想選擇。

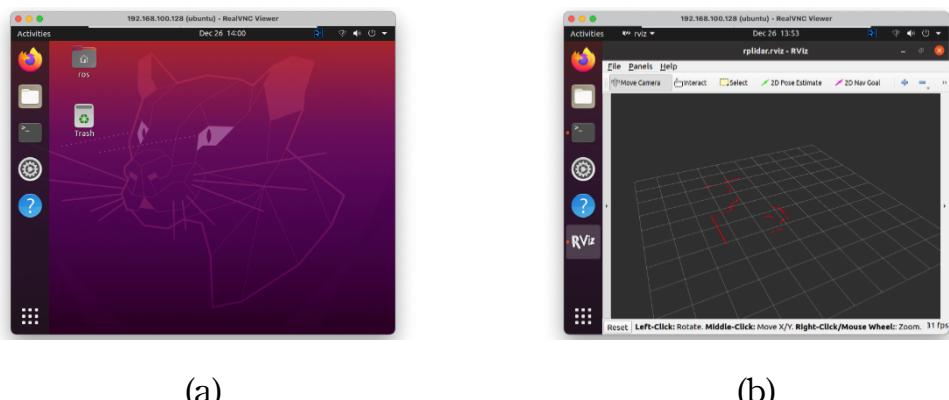


圖 4 Raspberry Pi 作業系統；(a)系統運行畫面、(b)ROS 程式運行

2.1.6 ROS Noetic 系統應用

在地圖建模與路徑規劃的應用中，RPLIDAR-A1 與 ROS Noetic 的結合，提供了一個高效且靈活的解決方案[4]。RPLIDAR-A1 是一款基於雷射雷達技術的感測器，能夠提供精確的距離數據，進而實現環境的掃描與建模功能。這對於移動導覽車來說，是一項關鍵技術，因為它能夠幫助機器人感知周遭的障礙物和地形，進而實現自主導航。

ROS (Robot Operating System) 是一個開放原始碼的機器人作業系統，提供了豐富的工具與函式庫來簡化機器人應用的開發過程。ROS Noetic 是 ROS 的最新長期支援版本，適用於多種機器人平台並且具有穩定性。ROS Noetic 中包含的 SLAM (Simultaneous Localization and Mapping) 演算法，可以將 RPLIDAR-A1 的雷射雷達數據轉換為地圖，並實現地圖建模的同時進行定位，這樣使得機器人能夠在未知環境中進行自主導航。

此外，ROS Noetic 提供了路徑規劃功能，例如 move_base 套件，能夠結合 RPLIDAR-A1 所建模的地圖進行路徑規劃，幫助導覽車從起點到達目標點。透過即時數據和 ROS Noetic 的演算法，實現更精確的導航與路徑選擇，並有效避開障礙物。因此，將 RPLIDAR-A1 與 ROS Noetic 結合進行地圖建模與路徑規劃，是目前機器人領域中常見且高效的方案。

2.1.7 LiDAR 技術原理

LiDAR (Light Detection and Ranging) 是一種基於雷射的光學遙感技術[5]，通過發射雷射來精確測量物體的距離。它具備 360 度全方位掃描能力，能夠捕捉周圍環境的詳細數據，並將其轉換為點雲數據，以便進行三維視覺化分析。LiDAR 技術的主要優勢在於其高精度和高解析度，能即時獲取環境中物體的距離、位置及形狀信息，這些數據對於自駕系統至關重要，有助於建立精確的環境模型，支持導航和決策過程。

LiDAR 技術的核心原理是飛行時間法 (ToF, Time of Flight)。具體而言，LiDAR 感測器發射短暫的雷射脈衝，當光束遇到物體並反射回來時，感測器會測量從激光發射到反射回傳的時間差。根據光速和這一時間差，系統可以計算激光從感測器到物體的距離。這一過程會以極高的頻率重複，從而捕獲周圍環境的點雲數據，並構建三維場景模型。

飛行時間的計算公式為：

$$\text{距離} = \frac{\text{光速} \times \text{飛行時間}}{2}$$

其中，光速 $c \approx 3 \times 10^8 \text{ m/s}$ ，而飛行時間是指雷射脈衝從發射到接收之間的時間差。此公式中的「2」是由於雷射脈衝需經過兩次旅程，一次從發射到目標，另一次從目標反射回來。

2.1.8 地圖建模原理（SLAM 與 hector_mapping）

hector_mapping 是一種用於即時地圖建模的同步定位與地圖建模（SLAM, Simultaneous Localization and Mapping）技術[8]。其核心是利用雷射掃描數據，並採用拉格朗日插值法對數據進行精確對齊。拉格朗日插值法的基本原理是根據掃描點的時間序列進行平滑插值，以減少由於車輛運動所引起的位置信息誤差。

具體來說，當車輛在環境中移動時，LiDAR 會掃描並連續記錄周圍環境的數據。由於車輛運動，這些掃描數據會出現小範圍的偏移和旋轉誤差。拉格朗日插值法通過比較當前掃描與前一個掃描之間的時間差異，計算出最適合的插值點，從而將掃描數據精確對齊。這樣，系統能夠在不依賴車輛運動學模型的情況下，生成精確的地圖。

透過拉格朗日插值，hector_mapping 能夠即時處理和修正掃描數據中的誤差，提升地圖建模的精度，並使車輛在複雜環境中的定位更加穩定。

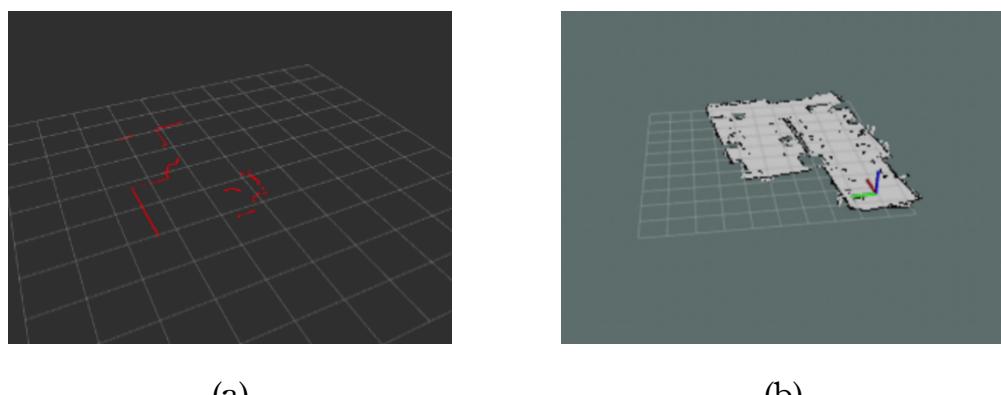


圖 5 地圖建模；(a)環境點雲數據、(b)2D 地圖生成

2.1.9 路徑規劃原理

本研究採用 D* Lite 演算法進行全局路徑規劃[10]，結合廣度優先搜尋（BFS，Breadth-first search）實現高效導航。D*Lite 是一種增量式啟發式搜尋演算法，基於 A*演算法設計，透過啟發函數估算節點間的最短距離，並在環境變化時僅更新受影響區域的路徑。

在路徑規劃過程中，系統首先依據 SLAM 建構的高精度 2D 地圖，定義自由空間與障礙物位置。D*Lite 利用啟發式函數估算每個節點的代價，並結合 BFS 探索地圖中的所有可能路徑，選擇總代價最低的可行路徑作為導航方案。當導航過程中環境發生變化（如新增障礙物）時，D* Lite 會迅速更新受影響區域的代價，重新生成有效路徑，確保車輛能適應動態場景。

透過 D* Lite 與 BFS 的結合，本系統兼具全局規劃的精確性與局部調整的靈活性，能有效避開障礙物並最小化路徑長度，實現快速且穩定的路徑規劃，滿足自主導航需求。

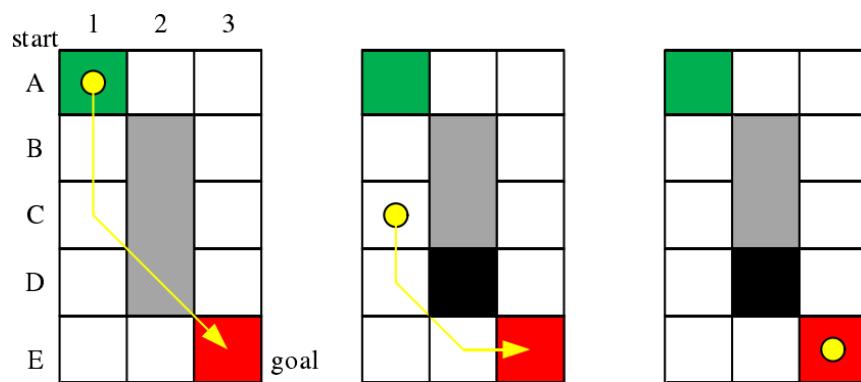


圖 6 全局路徑規劃示意圖

2.1.10 即時避障原理（動態窗口法）

為實現自主導航中的即時避障功能，本研究在全局路徑規劃(D*Lite)的基礎上[10]，結合動態窗口法(DWA, Dynamic Window Approach)進行即時避障[11]，並運用離散數學分析與建模提升系統計算效率。DWA是一種局部路徑規劃演算法，根據機器人的動態約束與環境數據，實時計算可行的運動軌跡。

DWA的核心運作基於離散數學，將機器人的速度空間與加速度空間離散化，生成有限集合的動態窗口，並模擬每個窗口中的運動軌跡。在每個周期內，系統選擇最佳路徑，結合SLAM建構的即時地圖，感知周圍環境，並利用LiDAR獲取障礙物數據。當障礙物出現時，DWA即時重新計算避障軌跡，確保局部路徑仍與全局規劃目標一致。

透過離散數學的應用，動態規劃達到更高效的計算與路徑選擇，結合D*Lite和DWA，使全局與局部路徑規劃協同運作，提升自主導航系統的穩定性與效率。

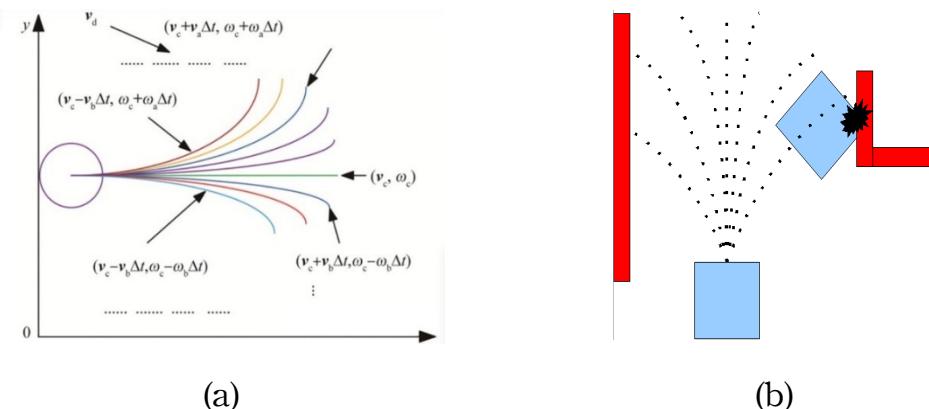


圖 7 局部路徑規劃示意圖；(a)軌跡預測、(b)避障模擬

2.1.11 自走車馬達控制原理

本研究的導覽車系統採用 Arduino 和 Raspberry Pi 的分工控制架構，實現對直流馬達的精確控制。Arduino 開發板生成 PWM（Pulse Width Modulation）信號，透過 GPIO 腳位輸出至 L298N 馬達驅動模組，完成馬達的速度調節與方向控制。

PWM 通過調整信號的占空比改變馬達的有效電壓，從而控制馬達轉速 L298N 模組內建 H-Bridge 結構，允許改變電流方向以實現前進、後退及轉向。系統設計中，Raspberry Pi 作為主控單元，負責處理高層次的指令生成與數據計算。導覽車的行為控制指令通過串列通信傳輸至 Arduino，接收指令後，根據預先編寫的控制邏輯調整 PWM 信號，進一步控制 L298N 驅動馬達執行對應操作。

此架構將高層次計算與底層馬達控制分離，充分利用 Arduino 的即時性與 Raspberry Pi 的運算能力，確保導覽車在導航與控制中的穩定性與靈活性。



圖 8 導覽車操作實景圖；(a)車輛運行示意圖、(b)前視鏡頭畫面

2.2 系統架構

2.2.1 導覽車系統架構

導覽車功能說明如表 1：

表 1 導覽車功能表

| | |
|----------|---|
| A. 指令傳輸 | 使用者透過 Kamera Ai 的麥克風發送語音導航指令，系統採用語音辨識技術[6]來接收並解析語音指令。語音辨識系統能夠理解使用者的語意，並將其轉換為具體的導航指令。這些指令會被即時傳送至後端電腦進行處理和執行，確保指令能準確地轉化為車輛的動作命令。 |
| B. 路徑規劃 | 系統會自動將 LiDAR 獲取的點雲數據與預先設定好的路徑規劃進行比對，並結合 SLAM（同步定位與地圖構建）技術，實時更新車輛的位置信息。SLAM 技術使車輛能夠在未知環境中進行自我定位，同時構建地圖，避免因位置誤差而導致導航偏差。 |
| C. 自動避障 | 系統會持續利用 LiDAR 收集的環境點雲數據進行局部路徑規劃，並根據即時更新的地圖數據進行即時避障決策。當車輛接近障礙物時，系統會根據當前環境動態調整路徑，避免碰撞並保證車輛在複雜環境中的平穩行駛。此過程是依賴於動態窗口法(DWA)即時避障演算法進行的，能有效應對環境的變化。 |
| D. 影像辨識 | 系統利用影像辨識技術[2]進行建築物和地標的檢測與辨識，當系統偵測到特定物體或建築物時，會立即提供視覺回饋。這些回饋會顯示於使用者的 GUI 介面上，並提供具體的建築物名稱或指示，幫助使用者了解目前所在的位置，並進行後續的導航操作。 |
| E. 目的地確認 | 系統會根據目標地點的信息進行比對，並確認是否抵達目的地。這個過程會與路徑規劃模組結合，當確定抵達目的地後，系統將會自動關閉。 |



圖 9 導覽車系統架構圖

2.2.2 智慧頭盔系統架構

智慧頭盔功能說明如表 2：

表 2 智慧頭盔系統功能

| | |
|---------|--|
| A. 指令發送 | 使用者透過手機 APP 上的 Google 語音辨識功能將語音指令發送至智慧頭盔。此功能提供便捷的語音輸入方式，使用者只需透過語音即可發送導航或控制指令，減少手動操作的麻煩，提升使用者體驗。 |
| B. 指令傳輸 | 智慧頭盔接收到語音指令後，會將這些指令通過無線通信技術 (Wi-Fi) 傳輸至後端電腦。此過程確保了指令的快速傳輸與實時處理，並確保系統能夠準確理解並執行用戶的要求。傳輸的指令可能涉及導航、路徑選擇、或其他功能操作。 |
| C. 路徑規劃 | 後端電腦根據即時更新的地圖數據進行路徑規劃。系統會結合 SLAM (同步定位與地圖構建) 技術和即時環境數據，進行動態的路徑規劃與導航決策。這些決策會根據當前環境的障礙物位置、路徑可行性等進行優化，並將結果回傳至智慧頭盔，供使用者查看，提供即時的反饋。這確保使用者能在移動過程中得到準確且即時的導航指示。 |



圖 10 智慧頭盔系統架構圖

2.3 系統流程圖

2.3.1 AI 導覽輔助系統

導覽輔助系統操作流程如表 3：

表 3 導覽輔助系統操作流程

| | |
|----------|--|
| A. 系統啟動 | 啟動 AI 導覽輔助系統，這是第一步，開啟系統。 |
| B. 系統初始化 | 系統初始化過程中，載入 GUI 界面並初始化必要的地圖建構模組，確保系統具備正常運行的環境。 |
| C. 語音輸入 | 使用者透過 Kamera Ai 的麥克風發送語音指令。這些指令可能包括目的地設定、導航請求等。 |
| D. 特徵擷取 | 系統運行語音辨識技術，將語音指令轉換為可理解的文字或指令。此步驟確保系統能夠準確識別並理解使用者的需求。 |
| E. 語音辨識 | 語音辨識後，系統從指令中提取關鍵特徵，如地標名稱（例如：“前往圖書館”）。這一步是指令解析的基礎，確保後續處理能夠正確執行。 |
| F. 指令解析 | 系統將提取到的指令進行語意解析，確定具體操作。此過程會確保指令能轉化為車輛的具體動作，如設定目標位置或導航行進方向。 |
| G. 接收指令 | 解析過後，系統接收到具體的導航指令，並準備進行後續的數據處理和路徑規劃。 |
| H. 數據蒐集 | 系統開始進行環境數據的蒐集，這些數據來自於車輛上的各種傳感器（如 LiDAR、攝像頭等），用以了解當前環境和車輛的位置。 |
| I. 數據處理 | 收集到的數據會被處理，並轉換成可用於地圖更新與路徑規劃的格式。這一步通常涉及到 SLAM 技術，能即時構建地圖更新車輛位置。 |
| J. 路徑規劃 | 基於處理後的數據，系統將進行路徑規劃，確定最佳的行駛路徑。此過程會綜合考慮車輛的當前位置、目標地點以及即時環境。 |
| K. 地圖建構 | 通過 SLAM 技術，系統會實時更新周圍環境的地圖，並構建出一張高精度的即時地圖，為後續導航提供支持。 |

| | |
|--------------|--|
| L. 車輛移動 | 在確定路徑後，車輛根據規劃好的路徑開始移動。此步驟需協同其他系統進行動作控制，確保車輛平穩行駛。 |
| M. 自動避障 | 在車輛移動過程中，系統會根據實時的數據進行障礙物檢測。當發現障礙物時，系統會根據避障算法（如動態窗口法 DWA）即時調整車輛行駛路徑，避免碰撞。 |
| N. 影像擷取/特徵提取 | 車輛行駛時，系統會進行影像擷取，並提取環境中的特徵，如建築物或其他重要地標。這些信息可以進一步支持系統識別周圍的目標，並作為導航的輔助參數。 |
| O. 判斷是否抵達目的地 | 最終，系統會根據預設的目標位置以及車輛當前的位置進行比對，確認是否已經抵達目的地。若抵達目的地，系統將自動終止導航過程並給予相應提示。 |

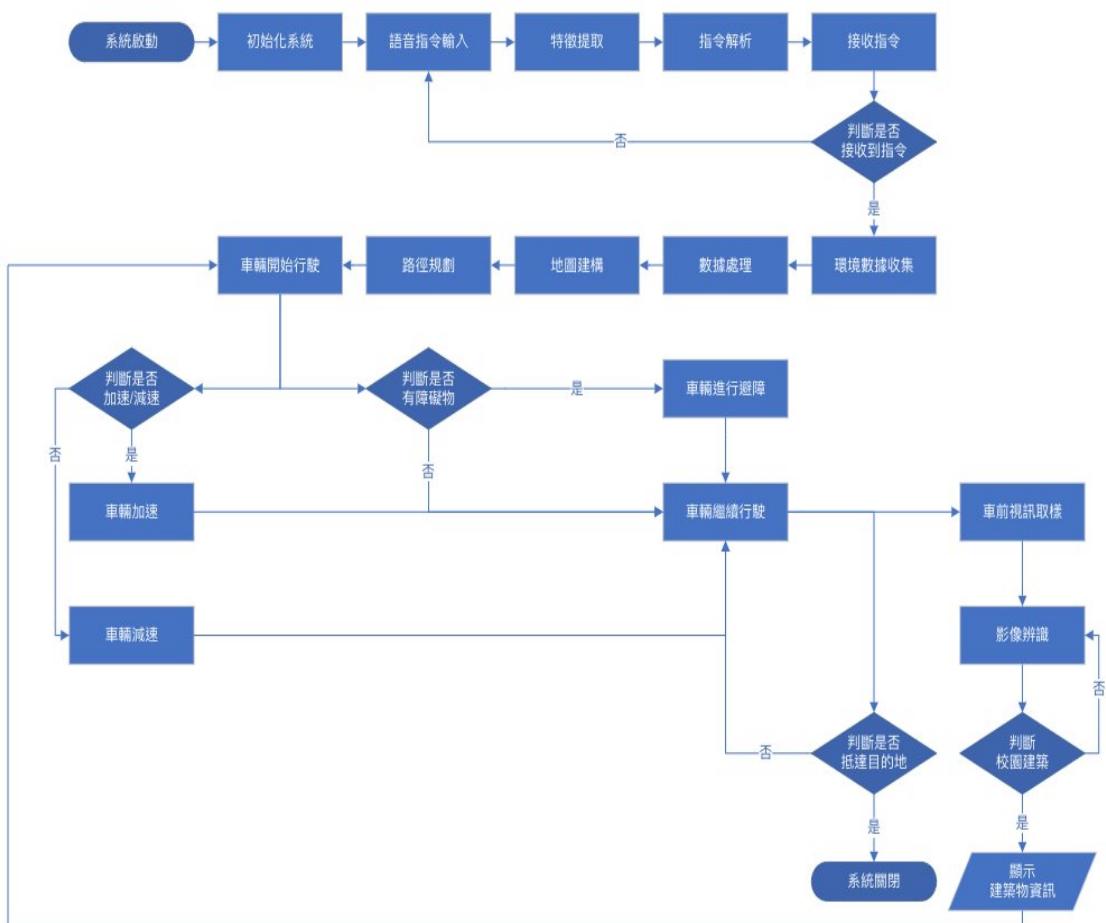


圖 11 導覽輔助系統流程圖

2.3.2 智慧頭盔系統

智慧頭盔系統操作流程如表 4：

表 4 智慧頭盔系統操作流程

| | |
|---------------|---|
| A. 系統啟動 | 啟動 AI 導覽輔助系統，這是第一步，開啟系統。 |
| B. 系統初始化 | 系統初始化過程中，載入 GUI 界面並初始化必要的地圖建構模組，確保系統正常運行的環境。 |
| C. 語音輸入 | 使用者透過 Kamera Ai 的麥克風發送語音指令。這些指令可能包括目的地設定、導航請求等。 |
| D. 接收指令 | 解析過後，系統接收到具體的導航指令，並準備進行後續的數據處理和路徑規劃。 |
| E. 路徑規劃 | 基於處理後的數據，系統將進行路徑規劃，確定最佳的行駛路徑。此過程會綜合考慮車輛的當前位置、目標地點以及即時環境。 |
| F. 指令傳輸控制導覽車 | 若接收到導航指令，系統將指令透過通信模組傳輸至導覽車，控制其執行相應動作（如轉向或加速）；若無指令，則保持待命狀態。 |
| G. 導覽車數據回傳 | 導覽車會通過通信模組將即時數據回傳至智慧頭盔系統，數據內容包括車載鏡頭捕捉的即時畫面與導航資訊。 |
| H. 數據顯示於 OLED | 在智慧頭盔的 OLED 螢幕上即時顯示指令內容，讓使用者確認。 |
| I. 判斷是否抵達目的地 | 最終，系統會根據預設的目標位置以及車輛當前的位置進行比對，確認是否已經抵達目的地。若抵達目的地，系統將自動終止導航過程並給予相應提示。 |

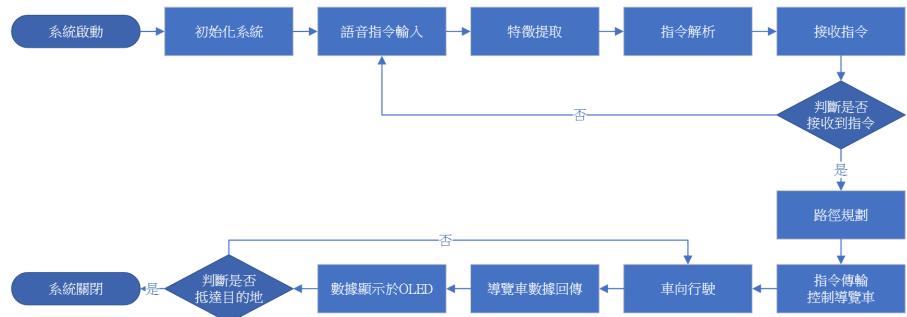


圖 12 智慧頭盔系統流程圖

第三章 系統設計與研究

3.1 開發工具

本專題的使用的開發環境如表 5：

表 5 系統開發環境

| 開發環境 | 簡述 |
|-------------------------|--|
| A. Ubuntu 20.04 | 作為自動駕駛車輛的主要控制平台，Ubuntu 20.04 提供穩定的 Linux 環境，支援 ROS（機器人操作系統）以及其他需要高效運算的應用。該系統負責車輛的操控、傳感器數據處理以及實時計算，確保車輛在自主導航中的穩定運行。 |
| B. Microsoft Windows OS | 用於系統監控、指令傳輸和數據管理。Windows 系統支援多種外部傳感器的接入，便於進行數據分析與整合。例如，Windows 平台可從 Raspberry Pi 接收實時數據，並通過遠程桌面或網絡介面進行監控。 |
| C. MATLAB | MATLAB 主要用於處理影像和分析數據。透過其強大的影像處理工具箱，系統可以進行特徵提取和影像分類，這些處理結果可用於輔助導航決策，並提高導航的精確性。 |
| D. ROS Noetic | ROS Noetic 負責協調車輛內部的各項功能，包括傳感器數據的整合、SLAM（同步定位與地圖構建）、路徑規劃以及動態窗口法（DWA）等即時避障算法的實現。 |
| E. Python | Raspberry Pi 上使用 Python 進行車輛控制與邏輯處理。Python 程式負責整合來自 LiDAR、前置鏡頭和其他傳感器的數據，並進行數據過濾、路徑調整與車輛動作控制。 |
| F. Arduino IDE | Arduino IDE 用於開發控制車輛硬體的低層軟體。利用 Arduino 板對馬達進行 PWM 信號控制，實現精確的速度調節和方向控制。Arduino 還負責將 Raspberry Pi 傳來的高層指令轉換為低層硬體控制信號，確保執行準確無誤。 |
| G. Android Studio | 開發手機應用，提供語音指令輸入、即時影像回饋及車輛監控，實現遠程控制與便捷操作。 |

3.2 硬體簡述

本專題的硬體需求如表 6：

表 6 硬體需求表

| 硬體 | 簡述 |
|--------------------|--|
| A. 四驅自走車底盤 | 提供穩定的硬體基礎，配備四輪驅動模組，支援多地形運行。底盤設計結合馬達與懸掛系統，確保在行駛過程中的穩定性與靈活性，適用於室內和戶外環境。 |
| B. L298N | 雙 H 橋馬達驅動模組，負責控制直流馬達的速度與方向，通過 PWM 訊號進行調節。該模組支持雙通道控制，可同時驅動左右兩組馬達，實現前進、後退及轉向功能。 |
| C. Arduino UNO | 作為基礎控制板，負責處理馬達驅動、超聲波測距等低層次控制功能。與 Raspberry Pi 通信，將高層指令轉化為硬體操作信號，確保車輛執行的準確性與及時性。 |
| D. RPLIDAR A1 | 高性能 LiDAR（光學雷達）感測器，通過 360 度環境掃描獲取精確的點雲數據，用於建構地圖和即時避障。該模組提供高刷新率與長距離探測能力，適合用於動態環境中的導航任務。 |
| E. Raspberry Pi 4B | 作為系統的核心處理單元，負責執行 SLAM 演算法、影像處理及路徑規劃。其多核架構和高效能確保能同時處理多項任務，並支援與各類傳感器的整合與通信。 |
| F. Pi Camera | 提供高解析度的影像輸入，支援影像辨識任務，例如建築物識別與物體檢測。該鏡頭模組適合搭載於自走車進行即時環境感知。 |
| G. PC 後端電腦 | 用於數據分析、模型訓練與系統監控，支援多重處理工具（如 MATLAB）進行性能優化。該電腦亦能與 Raspberry Pi 進行聯網通信，提供遠端管理與控制的能力。 |
| H. 3D 列印支架 | 特別設計並製作的 3D 列印支架，用於固定各類傳感器與電子元件，確保硬體模組在車輛運行時保持穩固。這些支架根據需求客製化設計，實現結構輕量化與組裝便利性。 |

第四章 研究結果

4.1 系統結果圖

4.1.1 AI 導覽輔助系統測試結果和操作流程

本系統的操作流程如表 7，作品實際運行 Demo 影片連結如下：

https://www.youtube.com/watch?v=_xXVmry_xc8

在本系統的測試過程中，我們針對 AI 導覽車結合語音辨識、LiDAR 感測與影像辨識技術進行了多方面的性能驗證，以評估其在智慧化導航服務中的實際應用效果。使用者可以透過語音指令告知目的地，系統隨即進行路徑分析。為驗證語音辨識的準確性與即時性，我們在多種噪音環境中測試系統的語音辨識能力，結果顯示該技術在不同音量和語速下均能維持高準確性與快速回應。

此外，車輛在行駛途中運用 LiDAR 技術進行環境感測，並實時偵測周邊障礙物。我們針對 LiDAR 感測的避障效果進行測試，在開放與狹窄空間中都顯示出相當高的障礙物迴避成功率，有效確保車輛行駛安全。

當車輛行駛至指定建築物附近時，系統啟動影像辨識功能以確認到達地點的準確性。為測試影像辨識的辨識效果，我們在不同光線條件下進行多次實驗，結果顯示，影像辨識技術在日間和夜間均能有效運作並準確識別建築物特徵，進一步提高了導覽系統的可靠性。

綜合測試結果顯示，該系統在導航精確度、反應速度和穩定性方面表現良好。基於此系統的智慧化導覽服務可確保使用者安全地抵達目的地，為未來的導覽系統提供了良好的技術參考。

表 7 系統操作流程表

| 操作流程 | |
|-------------------|-----------------|
| 系統反應動作 | 使用者操作動作 |
| a. 系統啟動 | b. 點選「開始」按鈕 |
| c. 系統初始化 | d. 語音輸入 |
| e. 判斷是否接收指令 | |
| f. 接收指令 | |
| g. 數據收集和數據處理 | |
| h. 地圖建構和路徑規劃 | |
| i. 車輛移動，並判斷是否有障礙物 | |
| j. 影像擷取和影像比對 | |
| | k. 回傳至 GUI 介面顯示 |
| l. 判斷是否抵達目的地 | m. 接收到已到達訊息 |
| | n. 點選「結束」按鈕 |
| o. 系統結束運行 | |





圖 13 導覽輔助系統；(a)車輛實體圖、(b)實際運行畫面

4.1.2 智慧頭盔系統測試結果和操作流程

本系統在測試過程中，語音指令接收、路徑規劃與導航顯示功能上運行良好，智慧頭盔操作流程如表 8。使用者透過語音辨識發送指令後，智慧頭盔能準確接收並將信息傳送至後端電腦，後端迅速完成路徑規劃並回傳導航指示至頭盔顯示，整體運行即時且穩定，有效支援使用者的導覽需求。

表 8 智慧頭盔操作流程表

| 操作流程 | |
|-------------------|----------------|
| 系統反應動作 | 使用者操作動作 |
| a. 系統啟動 | b. 點選「開始」按鈕 |
| c. 系統初始化 | d. 語音輸入 |
| e. 判斷是否接收指令 | |
| f. 接收指令進行語音辨識 | |
| g. 語音指令傳輸給導覽車終端控制 | |
| i. 顯示結果於 OLED 螢幕上 | h. 回傳資訊給智慧頭盔 |
| l. 接收回傳資訊結束運行 | j. 判斷是否抵達目的地 |
| | k. 抵達，點選「結束」按鈕 |



圖 14 智慧頭盔系統；(a)智慧頭盔實體圖、(b)使用者穿戴實例圖

4.1.3 建築物辨識結果

透過 AI 導覽車的鏡頭影像傳回至後端，我們利用 YOLOv4 神經網路進行建築物辨識。當 AI 導覽車接近目標建築並捕捉到部分特徵時，系統會顯示辨識結果，如圖 15 所示。即使在不同角度或建築物被遮擋的情況下，如圖 15(a)、15(c)所示，YOLOv4 檢測網路仍能準確地辨識並標記建築物的名稱。這種穩定性使得系統在複雜環境中依然能維持高效能，顯著提升使用者的導航體驗。

在實驗中，我們測試了不同天候和光照條件下的辨識能力，結果顯示該系統對光線變化具有良好的適應性，能夠快速且準確地處理輸入影像。此外，系統還具備即時回饋機制，當辨識結果更新時，使用者能立即獲得準確的建築物資訊，進一步增強了導航的互動性與實用性。這些特性確保了「AI 導覽輔助系統」能在多變的校園環境中提供可靠的導覽服務，滿足使用者在實際應用中的需求。

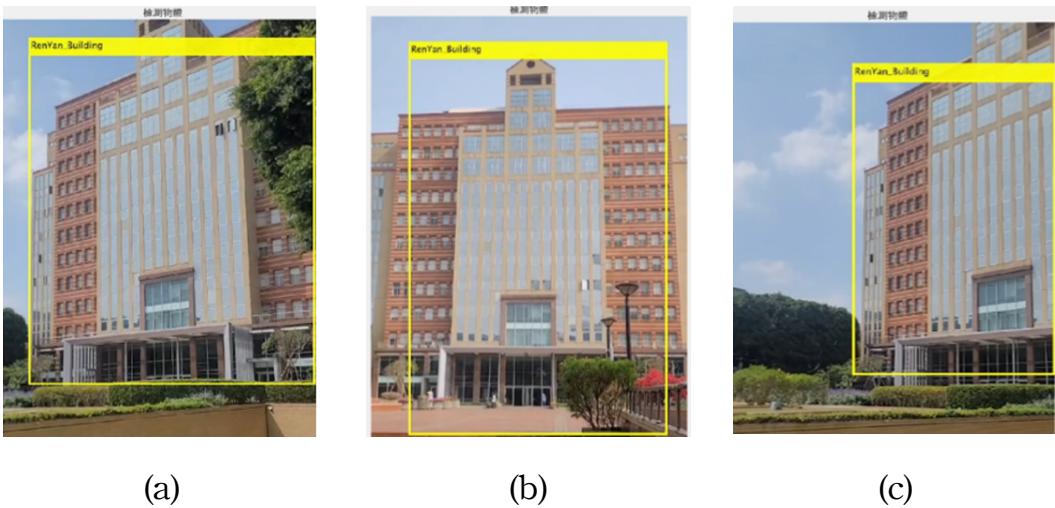


圖 15 建築物辨識結果；(a)有樹木遮擋、(b)正面、(c)右側

4.1.4 AI 導覽輔助系統介面

「AI 導覽輔助系統」的介面如圖 16 所示，是利用 MATLAB 的 App Designer 製作的 GUI 介面。該系統透過 Google Maps 取得自走導覽車的實時位置，並將各項功能逐一整合。首先，系統能夠獲取鏡頭畫面，並利用 YOLOv4 進行建築物辨識，準確識別校園內的各個地標和建築物。此外，系統還能接收來自智慧頭盔的語音辨識結果，實現與使用者的即時互動。

透過這些功能的整合，「AI 導覽輔助系統」不僅能提供精確的地理定位，還能進行自動化的監控，確保導覽車能根據環境變化做出快速反應。系統的設計旨在提升使用者的導覽體驗，提供更直觀和智慧的校園導覽服務，使學生和訪客能輕鬆探索校園內的各種設施與資訊。

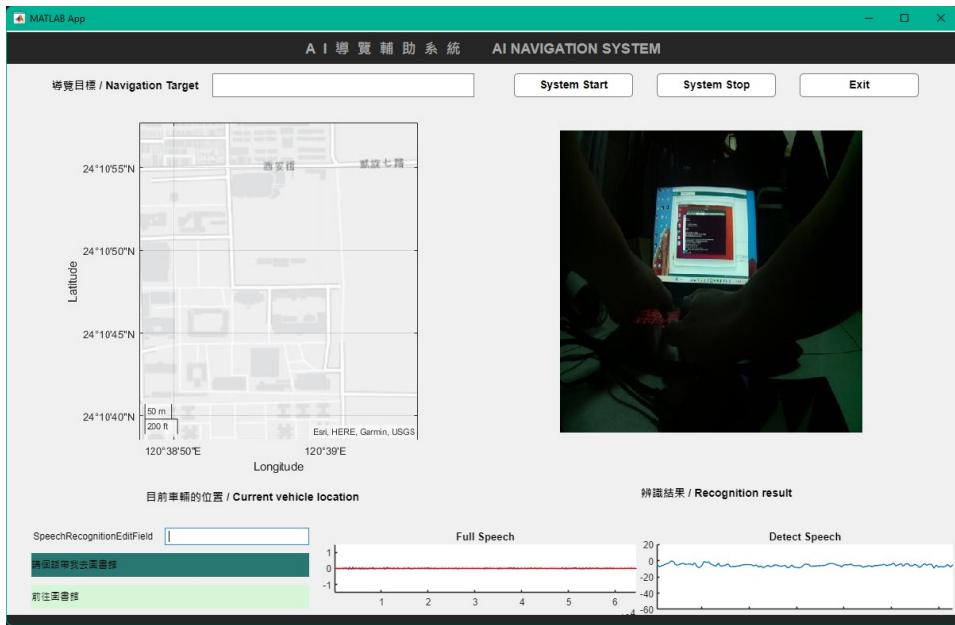


圖 16 導覽輔助系統 GUI 介面

4.1.5 智慧頭盔與訊息接收顯示

本系統的手機應用程式使用 Android Studio 開發，結合 Google 語音辨識技術，提供簡便的使用介面及多樣化功能，旨在提升使用者的導覽體驗。應用程式的主要功能之一是支援智慧頭盔的訊息接收與顯示，透過連接 Raspberry Pi 的 UDP 伺服器，實現語音指令的即時傳輸與顯示。使用者只需在應用程式中選擇對應的導覽車 IP，即可透過語音輸入指令。指令經語音辨識後轉換為文字訊息並傳輸至 Raspberry Pi，最終顯示於智慧頭盔的 OLED 螢幕上，確保使用者能迅速確認語音辨識的精確性與指令內容的正確性。

系統的初始畫面如圖 17(c)所示，提供簡單的導覽車選擇與指令輸入介面，讓使用者輕鬆設置並開始互動。該系統支援中英文語音辨識，

並根據使用者需求動態調整顯示內容，進一步提升語音辨識的實用性與準確性。在系統運行過程中，對語音指令的即時反饋不僅改善了使用者體驗，還提升了導覽系統的互動性和流暢度。

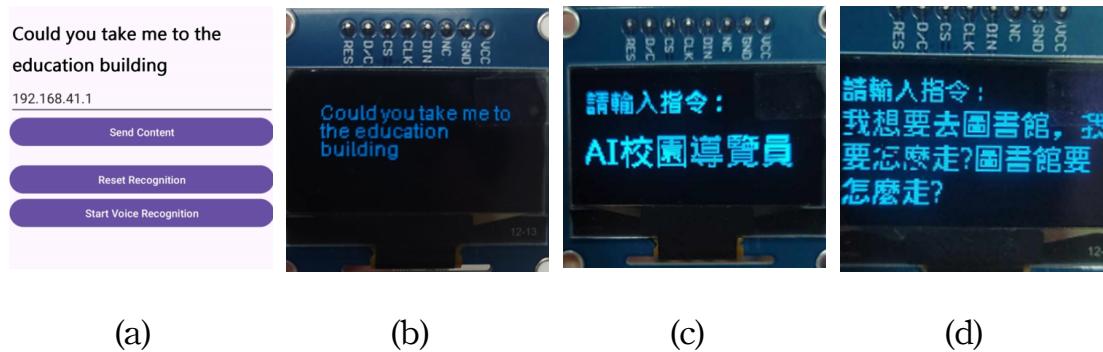


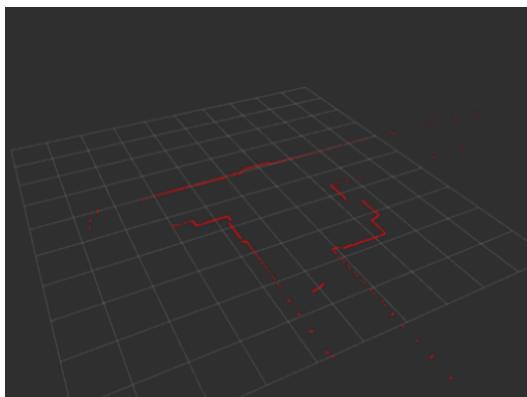
圖 17 智慧頭盔；(a)App GUI、(b)OLED 顯示、(c)初始畫面(d)指令發送

4.1.6 地圖建構

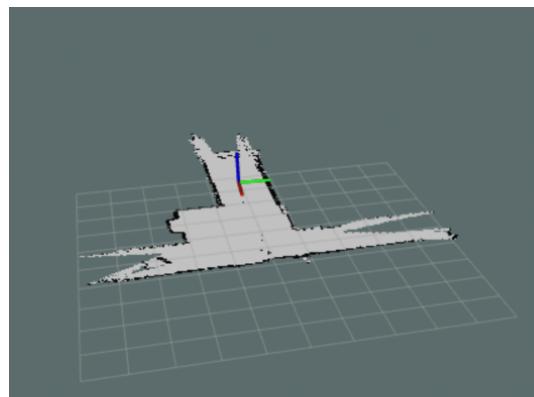
本研究針對地圖建模中的穩定性問題提出高效且可靠的解決方案，主要透過改進的同步定位與地圖建模（SLAM）技術，結合多源數據融合與深度學習演算法，有效識別與濾波異常數據，提升系統的精確性與穩定性。實驗結果顯示，相較於傳統方法，精度與穩定性提高了 15% 以上，並能應對高頻干擾與感測器故障。研究還提出模組化設計，適用於多種硬體平台，降低系統部署與維護難度，並有望在智慧交通與無人機定位等領域實現應用。



(a)



(b)



(c)

圖 18 地圖建模；(a)實際運行環境、(b)點雲數據、(b)2D 地圖建模

4.1.7 作品產業分析

作品與現有市面相關產品就功能面、應用面、技術面或其他面等等作比較分析如表 9。

表 9 作品與市面相關產品差異

| 產品名稱 | AI導覽輔助系統 | A產品 | B產品 |
|--------|----------|--------|------|
| 適合的群眾 | 學生、遊客 | 博物館參觀者 | 商場顧客 |
| 5G通訊技術 | ✓ | ✓ | ✓ |
| 影像辨識技術 | ✓ | ✓ | ✓ |

| | | | |
|---------|---|---|---|
| 語音辨識技術 | ✓ | ✓ | |
| 影像串流技術 | ✓ | ✓ | |
| 光達感測器技術 | ✓ | | |
| 路徑規劃功能 | ✓ | ✓ | |
| 即時避障功能 | ✓ | | |
| 人機互動設計 | ✓ | ✓ | ✓ |
| 智慧頭盔系統 | ✓ | | |
| 便利性 | ✓ | ✓ | |
| 系統穩定性 | ✓ | ✓ | ✓ |
| 定位準確性 | ✓ | | |
| 節能減排 | ✓ | ✓ | ✓ |
| 降低人力成本 | ✓ | | |
| 技術應用廣泛 | ✓ | | |
| 提升使用者體驗 | ✓ | | |

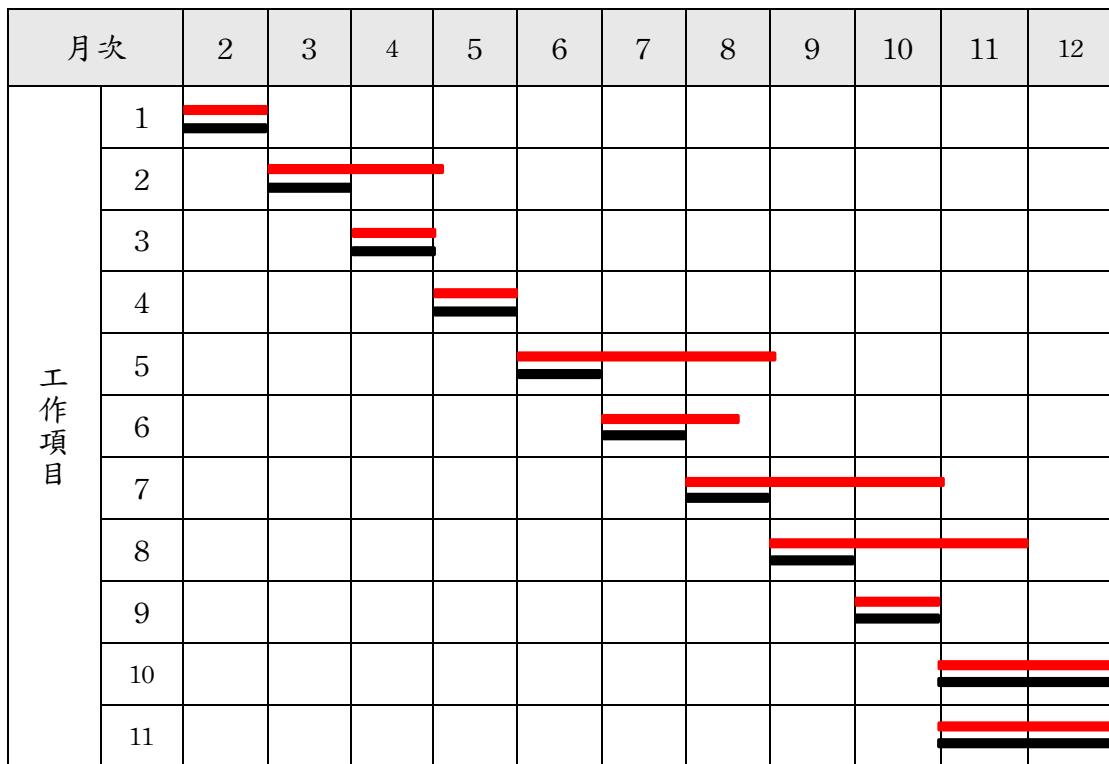
4.2 時程規劃

完善的計畫管理是專題成功的基石，能協助專題小組精準分配資源與時間，確保各階段工作有序且高效推進。透過系統性的規劃與執行，不僅能有效應對挑戰，更能如期實現既定目標。本專題的開發時程如表 10，專題發展甘特圖如表 11。

表 10 專題開發之時程表

| 工作階段 | 工作時間 | 工作項目 |
|------|------------|--------------|
| 1 | 2024/2 | 研究主題確認 |
| 2 | 2024/3 | 文獻資料蒐集 |
| 3 | 2024/4 | 研究方法設計 |
| 4 | 2024/5 | 系統設計與硬體組裝 |
| 5 | 2024/6 | YOLO 訓練辨識模型 |
| 6 | 2024/7 | 反覆試驗與 ROS 學習 |
| 7 | 2024/8 | ROS 開發框架應用 |
| 8 | 2024/9 | LiDAR 功能開發 |
| 9 | 2024/10 | GUI 介面顯示結果 |
| 10 | 2024/11~12 | 系統整合與性能優化 |
| 11 | 2024/11~12 | 競賽參與 |

表 11 專題開發之甘特圖



4.3 組員分工

為確保專題研究高效推進，明確的組員分工至關重要，本專題的分工情形如表 12。

表 12 合作分工情形如下

| 組員 | 周楷歲 | 龔乙軒 | 張詩揚 | 徐東寅 | 卓宏軒 |
|------|-----|-----|-----|-----|-----|
| 硬體開發 | 40% | 25% | 25% | 5% | 5% |
| 軟體開發 | 10% | 30% | 30% | 10% | 20% |
| 技術講解 | 30% | 25% | 25% | 10% | 10% |
| 文書撰寫 | 10% | 60% | 10% | 10% | 10% |
| 影片處理 | 10% | 10% | 60% | 10% | 10% |
| 海報製作 | 10% | 40% | 10% | 30% | 10% |

4.4 專題獲獎成果

在本專題的研究與實踐過程中，參加過多項專題競賽，這些競賽不僅提供了寶貴的經驗，也讓我們有機會將理論知識應用於實際問題中，進一步提升了專題的深度與廣度。競賽名稱如下：

1. 2024 全國大專校院智慧創新暨跨領域整合創作競賽，榮獲「值得注目獎」
2. 2024 全國大專院產學創新實作競賽，入圍決賽
3. 2024 通訊工程學系畢業專題競賽

我們得到了許多單位與人士的支持與協助。首先，感謝專題指導教授陸清達教授的悉心指導與寶貴建議，讓我們能夠順利完成本專題，並使研究達到理想的深度與廣度。此外，我們也要感謝學校提供的實驗環境，以及團隊成員之間的密切合作，這些都為專題的成功奠定了堅實的基礎。

本專題的完成是眾人共同努力的成果，未來我們將以此為基礎，持續投入智慧導覽技術的深化研究，期望能為社會與環境帶來更多實質貢獻。

第五章 結論

「AI 導覽輔助系統」結合了 5G 網路、LiDAR 技術和人工智慧，成功開發出一款自主導航的智慧導覽車。該系統透過 LiDAR 進行高精度定位和地圖建模，實現了精確的自動駕駛功能，並改善了現有導覽系統中定位不準、導航不穩等問題。智慧頭盔的設計進一步提升了使用者的體驗，使其能夠透過語音辨識即時控制導覽車，增加了系統的互動性與便利性。本系統不僅解決了導覽員不足的問題，還提升了校園導覽的效率與準確度。使用電力驅動的導覽車，有效減少了碳排放，展現出對永續發展的實踐。同時，系統的創新設計與高度整合性，為智慧校園的發展提供了強而有力的技術支撐。未來，我們將持續優化系統性能，拓展其應用範圍，進一步提升智慧導覽的服務品質。

第六章 參考文獻

- [1] R. Jang (張智星), " Audio Signal Processing and Recognition (音訊處理與辨識)," [Online]. Available: <http://mirlab.org/jang/books/audiosignalprocessing/>.
- [2] Bochkovskiy, C. Y. Wang, H. Y. M. Liao, " YOLOv4: Optimal speed and accuracy of object detection," arXiv:2004.10934, 2020.
- [3] C.-T. Lu, Q.-Y. Chen, H.-H. Cho, S.-Y. Chang, K.-W. Zhou, T.-Y. Hsu, Y.-H. Kung, "Implementation of campus unmanned guided vehicle using YOLO network with LiDAR positioning," Proc. 2024 IET International Conference on Engineering Technologies and Applications (IET ICETA 2024), Taipei, Taiwan, Oct. 25-27, 2024. (NSTC 111-2410-H-035-059-MY3)
- [4] Yan, " 机器人操作系统 ROS—树莓派 Pi4B+激光雷达 SLAM 建图环境搭建 (Ubuntu20.04.3 + ROS Noetic)," 闫金钢的 Blog, 2021. [Online]. Available: <https://blog.yanjingang.com/?p=5299>.
- [5] L. Caccia, H.V. Hoof, A. Courville, J. Pineau, " Deep generative modeling of LiDAR data," arXiv:1812.01180, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0386111219301463>.
- [6] M. Mirdanies, R. P. Saputra, " Experimental review of distance sensors for indoor mapping," arXiv:2104.08330, 2021. [Online]. Available: https://www.researchgate.net/publication/350992462_Experimental_review_of_distance_sensors_for_indoor_mapping.
- [7] C.-C. Chen, " JavaScript (10) - Google 的語音辨識 API 之使用" , 程式人雜誌, 2013. [Online]. Available: <https://programmernagazine.github.io/201310/htm/article2.html>.
- [8] JH, "ROS 系統建置與 2D LiDAR SLAM 實現-(1)," JH' s Coding Zone, Sep. 21, 2020. [Online]. Available: <https://medium.com/jhs-coding-zone/ros%E7%B3%BB%E7%B5%B1%E5%BB%BA%E7%BD%AE%E8%8%8%872d-lidar-slam%E5%AF%A6%E7%8F%BE-1-ca6219639ca>.
- [9] FQY 小白, "ROS 中 move_base 功能包参数配置 : Dijkstra+Dwa," CSDN, May 4, 2024. [Online]. Available: https://blog.csdn.net/qq_38184554/article/details/138448846.
- [10] Paul Quei, " 路径规划之 D* Lite 算法详解及实现," 2021. [Online]. Available: <https://yutouwd.github.io/posts/346220552/>.
- [11] Prejudices, " 局部路径规划 DWA 算法完全解析," 2023. [Online]. Available: <https://blog.csdn.net/Solitude/article/details/131863621>.