

摘要

本研究計畫旨在開發一套智慧交通巡檢系統，利用四軸飛行載具（Unmanned Aerial Vehicle, UAV）進行違規停車車輛的偵測。不同於以往先判斷紅線位置再確認車輛是否違規的方法，本研究透過四軸飛行器的 GPS 定位資訊來判斷車輛是否違規，避免車輛停在紅線上或完全遮掩紅線的情況。由於系統僅檢視違規區域，此方法還能加快巡邏速度，提升巡邏效率。

本系計畫採用 YOLOv7 物件偵測模型進行即時車輛辨識，結合飛行器的姿態與相機拍攝角度，利用逆透視變換 (Inverse Perspective Mapping, IPM) 定位，將偵測結果轉換為地面上的絕對位置，這樣能夠快速確定違規停車車輛的位置。透過事先建立的資料庫，無人機將比對飛越指定區域所拍攝的影像，並即時偵測違規車輛，記錄其位置。該系統的優勢在於高效的監控能力，能夠大幅減少設備投資成本，並提升違規車輛檢測的靈活性和即時性。

關鍵字：即時物件偵測、四旋翼無人機、空間對位

Abstract

This research aims to develop an intelligent traffic inspection system utilizing unmanned aerial vehicles (UAVs) to detect illegally parked vehicles. Unlike conventional methods that first identify the location of red lines and then verify whether vehicles are violating parking regulations, this system determines whether a vehicle is in violation by using the UAV's GPS positioning information, preventing vehicles from parking on or completely covering red lines. Since the system only inspects the violation areas, this approach helps speed up patrol operations and enhance overall patrol efficiency.

This research adopts the YOLOv7 object detection model for real-time vehicle identification, combined with the UAV's attitude and camera angle. By using Inverse Perspective Mapping, the system locates and converts the detection results into absolute ground positions, allowing for quick identification of illegally parked vehicles. Through a pre-established database, the UAV compares the images captured while flying over designated areas and immediately detects violation vehicles, recording their locations. The advantage of this system lies in its efficient monitoring capability, significantly reducing equipment investment costs and improving the flexibility and immediacy of illegal parking detection.

Keyword: Real-time Object Detection, Quadcopter, Georeferencing

目錄

摘要	i
Abstract	ii
1 緒論	1
1.1 研究背景	1
1.2 研究動機	1
1.3 研究問題	2
2 文獻探討與回顧	3
2.1 應用 YOLOv7 於車輛物件偵測	3
2.2 應用 VBS-RTK 之 GPS 定位	4
2.3 影像處理與空間對位	5
3 研究方法及流程	5
3.1 飛行路徑規劃	9
3.2 違規車輛偵測	10
3.3 影像處理	10
3.4 空間對位	15
4 研究結果	16
4.1 飛行路徑規劃	16
4.2 IMU 資料獲取	16
4.3 車輛偵測	16
4.4 影像處理	17
4.5 逆透視變換	17
4.6 空間對位	18
5 結論	18
6 參考文獻	19

圖目錄

1	無人機偵測違停車輛示意圖	1
2	本計畫研究問題之示意圖	2
3	YOLO 各版本演進圖 [5]	3
4	YOLOv7 與其它模型比較 [6]	4
5	YOLOv7 架構圖 [7]	4
6	VBS-RTK 運作流程圖 [8]	5
7	共線式幾何示意圖 [10]	5
8	研究流程圖	6
9	本研究所使用之無人機	7
10	本研究使用之單目攝像頭 [13]	7
11	Mission Planner 實際畫面 [15]	8
12	YOLOv8 違規車輛偵測示意圖	10
13	無人機座標系統定義	11
14	非垂直投影示意圖 [10]	12
15	座標轉換流程圖	13
16	QGIS 操作畫面 [21]	15
17	mavlink	16
18	Yolov7 車輛偵測與追蹤	17
19	Yolov7 車輛偵測與追蹤-無人機實拍	17
20	地板測試	18
21	空間對測試	18

表 目 錄

1	C922 Pro Stream 規格	8
2	MSI GP76 Leopard 規格	9
3	MacBook Air M1 規格	9

1 緒論

1.1 研究背景

無人機（Unmanned Aerial Vehicle, UAV）技術近年來快速發展，整合各種附加設備並隨著飛行控制與自動化技術的成熟，在軍事、執法及科技應用領域取得顯著成效。四軸旋翼無人機因其安全性高、成本低的優勢，已廣泛應用於各種場景，其中攝影與錄影功能尤為重要。無人機擁有比傳統攝影設備更廣的視角，且受環境限制較小，使其在大範圍場景捕捉方面表現優異，優於傳統監控方式。

在交通監控領域，無人機的靈活性和效率遠勝於傳統固定式監控設備，如固定式照相桿雖能執行監控任務，但監測範圍有限且設備成本較高。隨著無人機技術的進步，其在交通巡檢方面展現出極大的應用潛力，特別是在違規停車偵測方面，無人機能夠提供更靈活且高效的解決方案。目前，中國部分城市已開始運用無人機進行科技執法 [1]，台北市與台中市警察局亦已成立無人機機隊 [2][3]，用於交通監測與執法，並協助執法人員即時掌握交通狀況，提升執法效率。

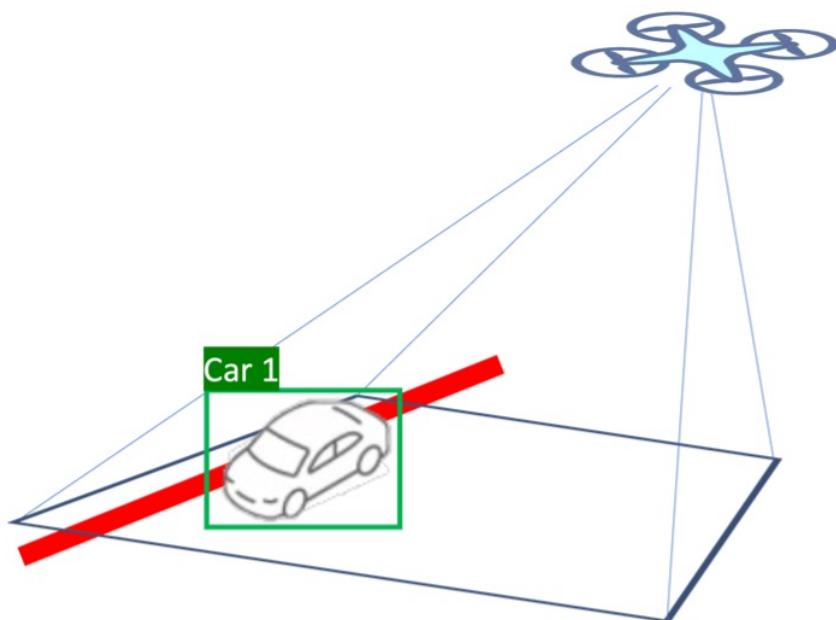


圖 1：無人機偵測違停車輛示意圖

1.2 研究動機

目前的研究和應用大多針對車輛橫跨紅線進行判斷，但對於完全蓋住紅線或停在紅線與白線交界處的情況，現有方法無法準確判斷車輛是否違停。因此，本研究計畫的動機在於利用無人機的 GPS 位置資訊來判斷車輛的違規情形，避免車輛完全遮掩紅線或停在紅線邊界處的情況。如圖 2 所示。並通過建立資料庫並使用無人機拍攝影像，比對車輛所處位置與違規區域，能夠快速偵測違規車輛並進行即時回報，進一步提升交通執法的效率與靈活性。

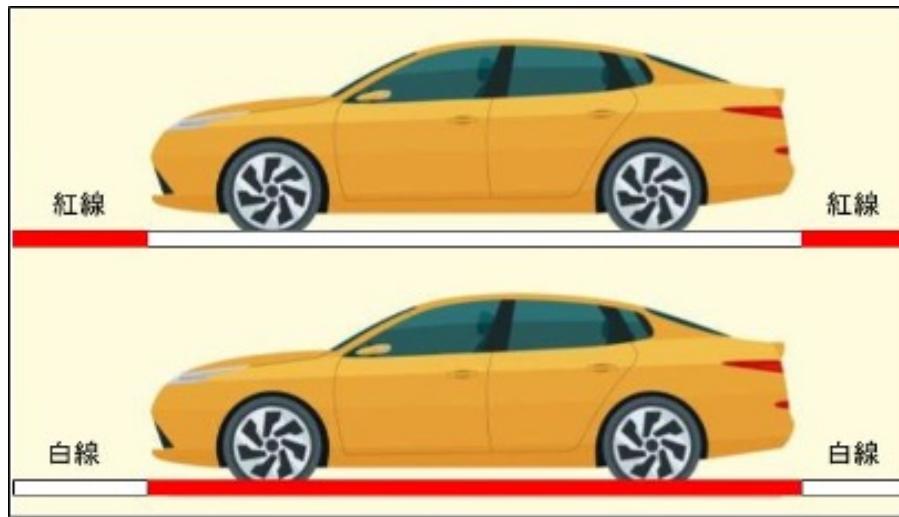


圖 2: 本計畫研究問題之示意圖

1.3 研究問題

利用無人機拍攝出的影像辨識出沿路車輛，利用無人機本身 GPS 位置，換算出影像中車輛之位置，比對車輛所在位置是否可以停車，以判斷車輛否違規，若有違規情形，回傳車輛位置並將其建立成資料庫，以便於人員複查或是違規檢舉。研究問題可以分為三大項，違規車輛偵測、影像處理、空間對位。

1.3.1 違規車輛偵測

為了精確偵測違規停放的車輛，本系統採用先進的影像辨識技術，利用 YOLOv7 進行即時車輛偵測，確保能快速且準確地識別影像中的車輛。透過無人機高空巡檢，可大範圍覆蓋監控區域。此外，系統將結合 GPS 位置資訊，以判斷車輛是否位於可能的違規區域，如紅線停車區、消防通道或人行道上，進一步提高偵測的可靠性與準確性。

1.3.2 影像處理

本計畫將利用鏡頭往斜下的角度拍攝而非垂直向下。以斜下角度拍攝的特點是相較於垂直向下拍攝可以擁有較大的視野範圍。而採用無人機的 GPS 位置資訊判斷，無人機拍攝圖片中每個像素與實際地面之關係格外重要，除了無人機上相機的傾角，還需整合無人機在飛行時的對地高度、翻滾角 (Roll)、俯仰角 (Pitch)、偏航角 (Yaw) 等慣性測量單元 (Inertial measurement unit, IMU)，並依據該照片拍攝時之 IMU 資訊做正射投影，運算每個像素與無人機之間的相對位置，並依照無人機的 GPS 位置準確定位違規車輛之位置。

1.3.3 空間對位

為確保偵測結果的精準性，系統將進行空間對位處理，以克服飛行器姿態變化、鏡頭視角與透視變形等因素所造成的誤差。透過飛行器的 IMU (慣性測量單元) 數據與 GPS 定位資

訊，獲取無人機的姿態與拍攝角度，並利用透視變換技術將影像坐標轉換為地面絕對位置。最後，系統將偵測出的車輛位置與電子地圖上的違規區域進行重合度分析，精確確認車輛是否確實違規停放，並輸出違規證據以供後續處理。

2 文獻探討與回顧

2.1 應用 YOLOv7 於車輛物件偵測

違規停車偵測的部分預計使用物件偵測深度神經網路，找出車輛位置，以判斷車輛是否違規，利用物件偵測深度神經網路，不管是在標註樣本或是進行訓練上都可以節省許多的時間，並且擁有較高的準確度。本計畫採用知名的物件偵測深度神經網 YOLO，其全名為 You Only Look Once，首次出現於 2015 年，由 Joseph Redmon 和 Ali Farhadi 等人提出 [4]。YOLO 能夠在實時情況下執行物體檢測，這意味著它可以快速地識別圖像中的物體，並且適用於需要即時反饋的應用場景，相較於其他物體檢測算法，YOLO 算法在準確性上表現出色，能夠有效地檢測各種大小和形狀的物體，相對於其他複雜的物體檢測算法而言，實現起來相對簡單，這使得它受到廣泛的歡迎並且易於應用於各種領域，有實時檢測、高準確性、簡單易用、可擴展性等優點。

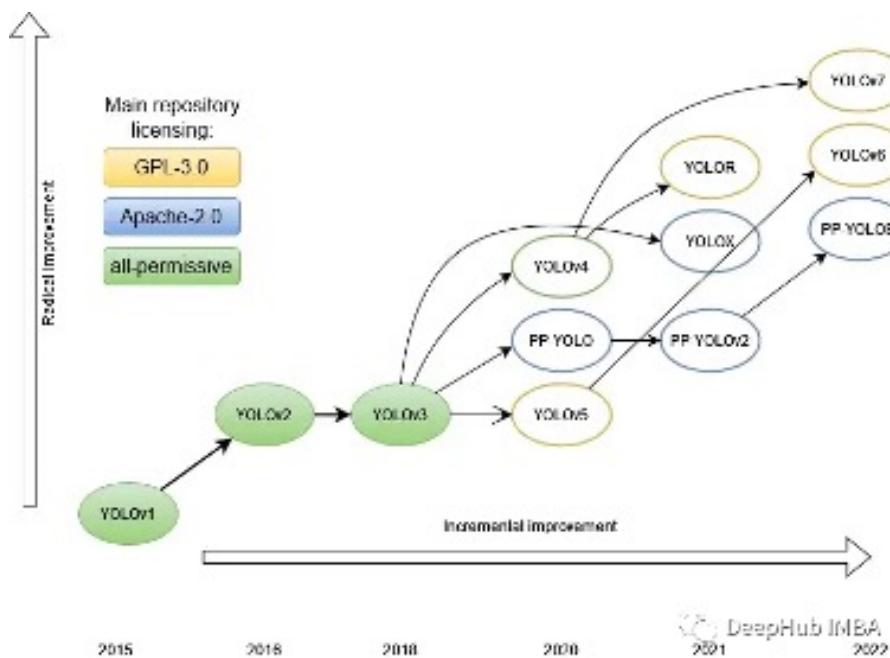


圖 3: YOLO 各版本演進圖 [5]

本研究選用 YOLOv7[6] 作為物件偵測算法，其為 YOLO 系列中的高效版本，具有優異的準確性與速度 (如圖 4 所示)。YOLOv7 屬於 one stage 物件偵測方法，能夠在單次運行中同時完成位置偵測與物件分類，相較於 two stage 方法，其運算速度更快，且準確度仍維持在可接受範圍內。YOLOv7 在架構設計上延續了 YOLOv3 的基本結構，包括主幹網路 (Backbone)、連接層 (Neck) 與檢測頭 (Head)，並透過引入新技術進一步提升效能 (如圖 5 所示)。Backbone 負責特徵提取，輸入圖像經過多層卷積運算，從 B1 至 B5 逐步提取更加豐富

的特徵資訊，特徵圖尺寸逐漸縮小，但通道數量增加；Neck 透過上採樣 (upsampling) 擴增特徵圖，並透過橫向連結融合不同層級的特徵，使模型能夠有效學習多尺度資訊；Head 則負責最終檢測結果的輸出，透過下採樣 (subsampling) 還原金字塔結構，最終產生多種解析度的特徵圖 (P3、H4、H5) 以適應不同大小的物件偵測需求。此外，YOLOv7 透過模型重參數化 (Re-parameterized model) 改進模型結構以提高推理速度，並提出 Extend-ELAN(E-ELAN)，在不破壞梯度傳播的情況下增強網路學習能力。透過這些技術，YOLOv7 在物件偵測任務中展現了更優異的效能，能夠在高準確度與低延遲之間取得良好平衡，適用於多種應用場景。

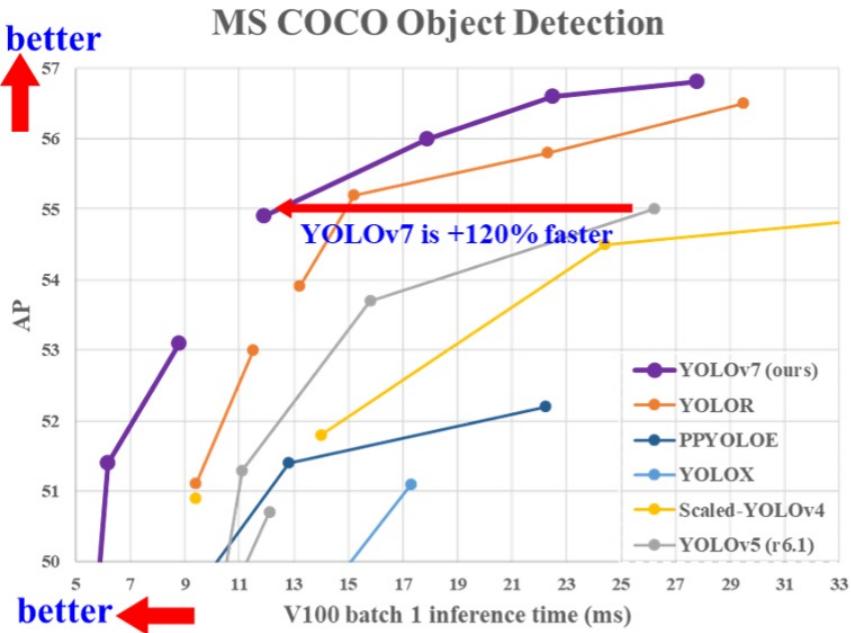


圖 4: YOLOv7 與其它模型比較 [6]

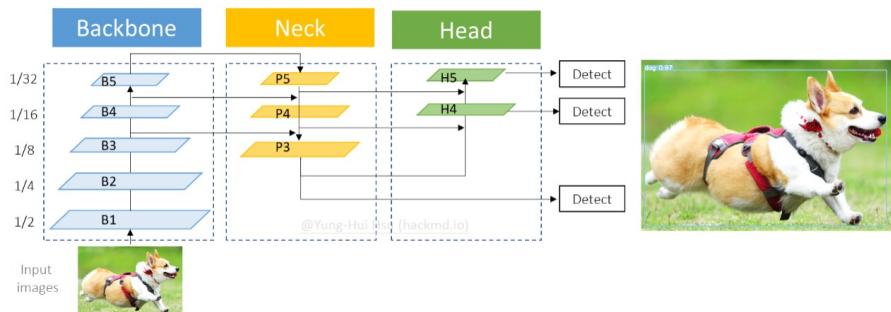


圖 5: YOLOv7 架構圖 [7]

2.2 應用 VBS-RTK 之 GPS 定位

在定位方面採用即時動態定位技術 (Virtual Base Station Real-Time Kinematic, VBS-RTK) 以獲得無人機精確位置，VBS-RTK 是 e-GNSS 即時動態定位系統之核心定位技術。其係採用多個衛星定位基準站所組成的 GNSS 網絡來評估基準站涵蓋地區之定位誤差，再配合最鄰近的實體基準站觀測資料，產製一個虛擬的基準站做為 RTK 主站，所以移動站並不是接收某個實體基準站之實際觀測資料，而是經過誤差修正後的虛擬觀測數據。

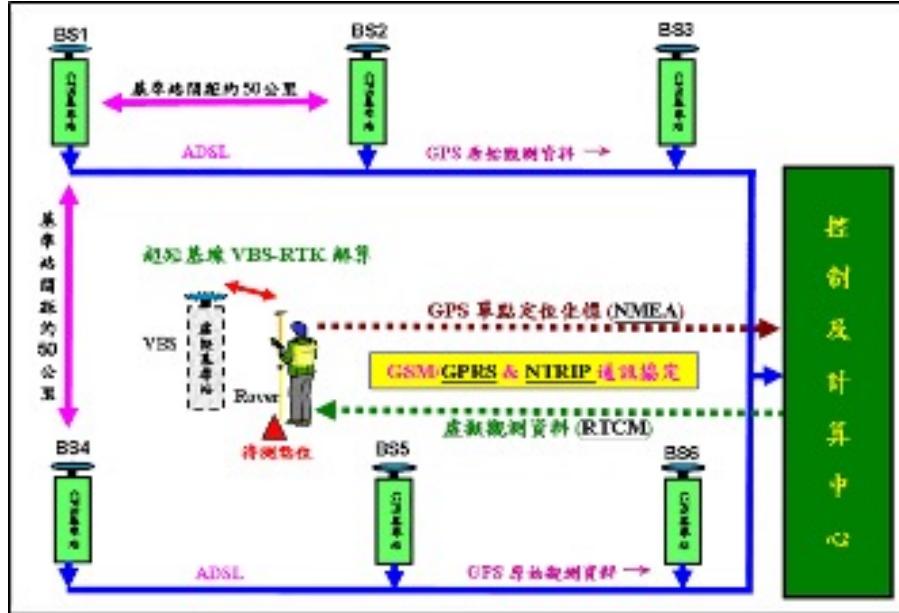


圖 6: VBS-RTK 運作流程圖 [8]

2.3 影像處理與空間對位

本計畫將利用鏡頭往斜下的角度拍攝而非垂直向下。以斜下角度拍攝的特點是相較於垂直向下拍攝可以擁有較大的視野範圍。這部分需應用航空攝影測量原理 [9] 的中共線方程式 (Collinearity equation) 求得地面點在像片上對應之坐標，共線方程式係基於透視中心投影之原則下根據物點、像點、以及攝影時的參數所建立，求得每一地面點在相片上對應之坐標。

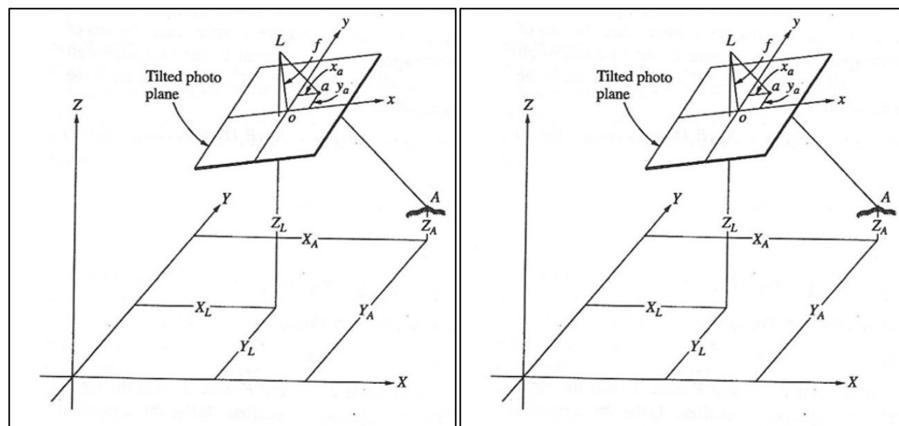


圖 7: 共線式幾何示意圖 [10]

3 研究方法及流程

本計畫要設計的電腦應用程式，能夠標示出地圖中不可停車的區域，而無人機則透過事先建立的資料庫，比對飛越指定區域所拍攝的影像，快速辨識違規車輛並記錄其車牌號碼，運作流程如 8 所示。

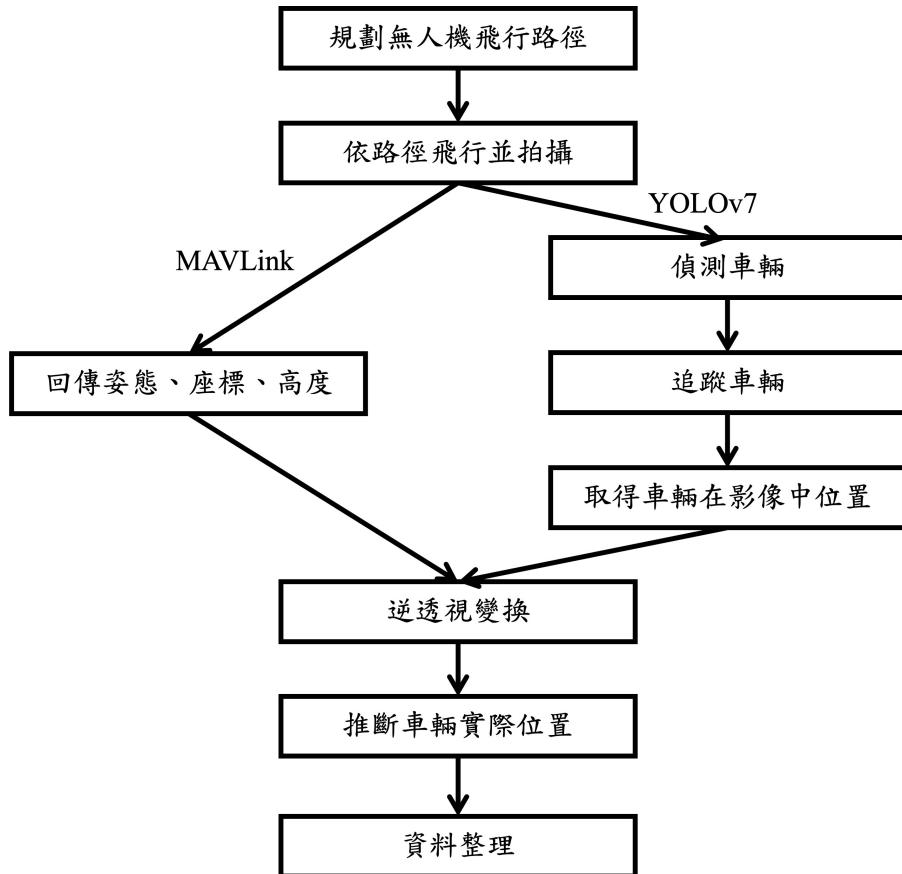


圖 8: 研究流程圖

本研究中所使用的無人機，為委託臺灣希望創股份有限公司 (Taiwan Hope Innovation Corporation) 所特別製作的無人機，實驗中使用的無人機為含有 Pixhawk4 Orange 飛控板、樹梅派、GPS、HUAWEI 4G USB 網卡和 Logitech C922 Pro Stream 單目攝像頭 (具體規格詳見表 1) 的四軸飛行器。錄製到的影像透過無人機上的樹梅派傳送到 RTMP server [11][12]。在由其他電腦從 RTMP server 取得影像並做辨識。為了使無人機飛行更順利，無人機上含有高性能的飛控板、精確的 GPS 定位系統，在兩者相互搭配下，使無人機有更穩定的飛行。



圖 9: 本研究所使用之無人機



圖 10: 本研究使用之單目攝像頭 [13]

表 1: C922 Pro Stream 規格

項目	規格
解析度	1280×720 (HD)
幀率	30fps
水平視角	70.42°
垂直視角	43.3°

無人機控制方面透過 4G 網卡及 VPN 以遠端控制樹莓派，並且在樹莓派中我們透過 MAVProxy(MAVLink Proxy)[14] 指令將飛控板透過 UDP 方式連線至電腦端的 Mission Planner[15]，Mission Planner 提供了一個實時的飛行監控界面(如圖 11所示)，可編輯飛行計劃，並記錄飛行數據。



圖 11: Mission Planner 實際畫面 [15]

本實驗透過 4G 網路連接無人機進行控制，並在電腦端使用 ROS 進行影像擷取與處理。此外，在影像處理的同時，還需透過 Mission Planner 即時控制無人機並藉由 MAVLink 協議獲取相關資訊。因此，筆記型電腦必須具備足夠的性能，以確保系統的穩定運行與即時回應。

本計畫中，主要使用 MSI GP76 Leopard 筆記型電腦(具體規格詳見表 2)。該機型以高效能為主打，搭載 Intel® Core™ i7-10870H 處理器與 NVIDIA® GeForce® RTX 3060 顯示卡，並採用 MSI 最新技術 Cooler Boost 5。其雙散熱模組設計、強力雙風扇與多組熱管，加上四向散熱氣流設計，能有效排除廢熱，確保處理器與顯示卡在高負載運行時維持最佳性能。

表 2: MSI GP76 Leopard 規格

元件	規格
中央處理器	Intel® Core™ i7-10870H
記憶體	DDR4 16GB
硬碟	1TB SSD
顯卡	NVIDIA® GeForce® RTX 3060
作業系統	Ubuntu 18.04
電池	4-Cell 65 Battery (Whr)

此外，本實驗也輔以 Apple MacBook M1 (具體規格詳見表 3) 進行部分運算與開發工作。MacBook M1 採用 Apple M1 晶片，內建 8 核心 CPU 及 7 核心 GPU，並配備 8GB 統一記憶體 (Unified Memory)，提供優異的效能與低功耗表現。在本研究中，MacBook M1 主要用於 ROS 1 相關開發、Python 影像處理、無人機數據分析與程式開發。不過，由於 M1 晶片採用 ARM 架構，與傳統 x86 架構不同，在執行 ROS 及部分開發環境時，需透過 Docker 來確保軟體相容性。

表 3: MacBook Air M1 規格

元件	規格
處理器	Apple M1 (8 核心 CPU)
記憶體	8GB Unified Memory
硬碟	256GB SSD
顯示卡	Apple M1 內建 GPU (7 核心)
作業系統	macOS Sequoia
電池	49.9Wh

綜合而言，MSI GP76 Leopard 主要負責高負載的影像處理與無人機控制，而 MacBook M1 則用於負責高負載的影像處理、程式開發與數據分析，兩者搭配可有效提升本實驗的運算效率與工作流程。

3.1 飛行路徑規劃

3.2 違規車輛偵測

在車輛偵測方面，透過物件辨識技術 YOLO(You Only Look Once)，能夠高效、精確地辨識影像中的車輛。YOLO 是一種基於深度學習的物件偵測模型，能夠在單一的前向傳播過程中，同時進行分類與定位，從而達到快速處理的效果。結合深度學習技術，YOLO 能夠對車輛進行實時的辨識與定位，並在影像中標註出車輛的具體位置。這不僅提高了車輛偵測的準確性，也能在多種複雜環境下有效運行，無論是白天、夜間，還是不同的天氣條件下。藉由這些辨識出的資訊，系統可以進一步進行車輛是否違規的判斷。



圖 12: YOLOv8 違規車輛偵測示意圖

3.3 影像處理

無人機的座標系統定義如圖 13所示。 y 軸的正方向為北方 x 軸與 y 軸垂直且以東方為正。航向角 ψ 是以 yz 平面為基準，沿著 z 軸順時針旋轉的角度，範圍從 0° 到 360° 。俯仰角 θ 是以 xy 平面為基準，沿著 x 軸逆時針旋轉為正的角度，範圍從 90° 到 -90° 。滾轉角 ϕ 是以 xy 平面為基準，繞著 y 軸順時針旋轉的角度，範圍從 -90° 到 90° 。無人機上的攝像頭默認朝向座標系的正 y 軸方向，攝像頭傾斜角度為 θ_c ，範圍從 0° 到 -90° ，攝像頭位置與無人機重心距離忽略不計。

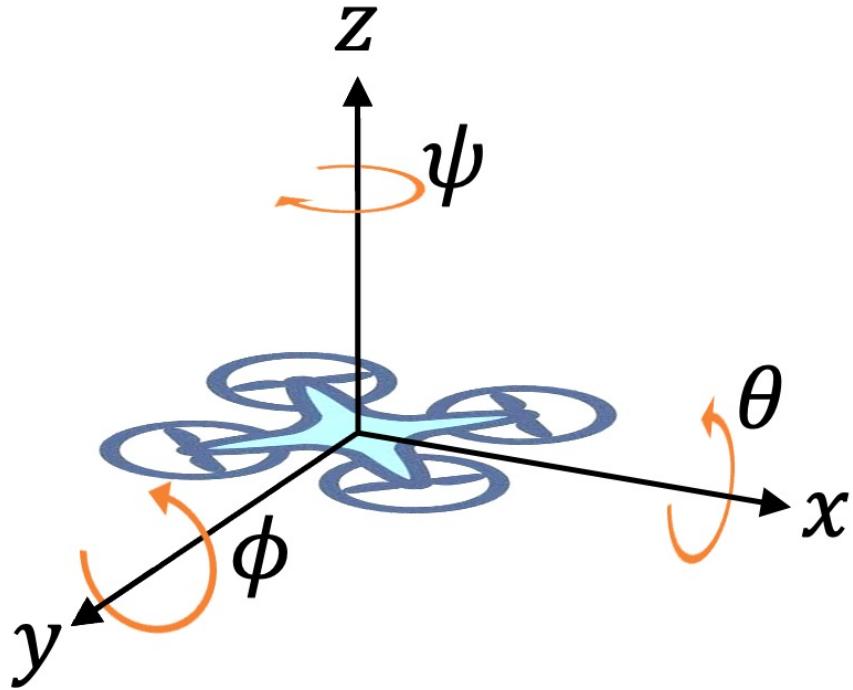


圖 13: 無人機座標系統定義

本計畫將利用鏡頭往斜下的角度拍攝而非垂直向下，首先測試單純考慮無人機相機傾斜之投影，根據航空攝影測量原理 [9]，可推導出以下公式。 $P_r(u_r, v_r)$ 照片上的一個點， $P_v(u_v, v_v)$ 為其在正射投影上的對應點(如圖 14 所示)。以 A 表示為相片中心在鳥瞰圖上的對應點，並定為鳥瞰圖座標系的原點。 f 為焦距，原照片與正射投影的對應關係可由以下算式求得：

$$u_v = f \times \sec(\theta_c - \arctan \frac{v_r}{f}) \times \frac{u_r}{\sqrt{v_r^2 + f^2}} \quad (1)$$

$$v_v = f \times (\tan \theta_c - \tan(\theta_c - \arctan \frac{v_r}{f})) \quad (2)$$

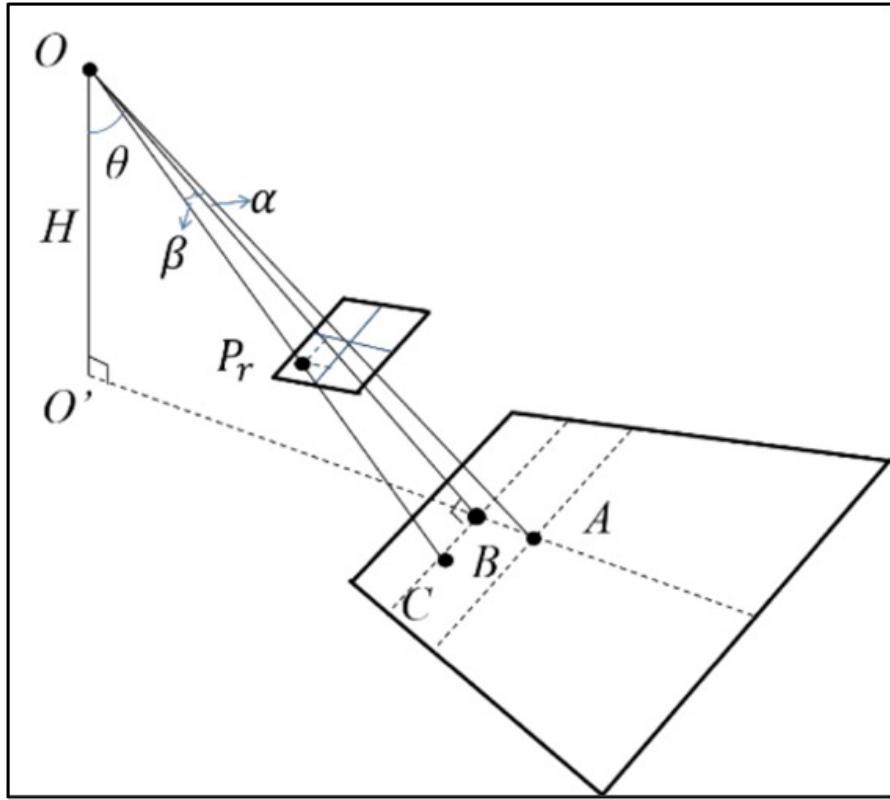


圖 14: 非垂直投影示意圖 [10]

本計畫將利用鏡頭往斜下的角度拍攝而非垂直向下，且飛行時會產生翻滾角 (Roll)、俯仰角 (Pitch)，以上因素皆會影響成像。因此本研究採用逆透視變換 (Inverse Perspective Mapping, IPM)[16][17][18] 技術處理無人機拍攝的影像，以將像素座標轉換為地面座標，最後轉換成 GPS 座標。

首先將影像像素座標系 (u, v) 轉換為圖像座標系 (x, y) 。加入相機的水平視角 $HFOV$ 與相機的垂直視角 $VFOV$ 、相機俯仰角 θ_c ，無人機姿態則藉由 mavlink 從 mission planner 獲取俯仰角 θ_q 、滾轉角 ϕ_q 、航向角 ψ_q ，將圖像座標轉換為相機座標 (x_c, y_c, z_c) 。接著將相機座標系轉換為無人機標系 (x_w, y_w, z_w) ，以取得拍攝影像中內容物與無人機之相對位置。流程圖如下：

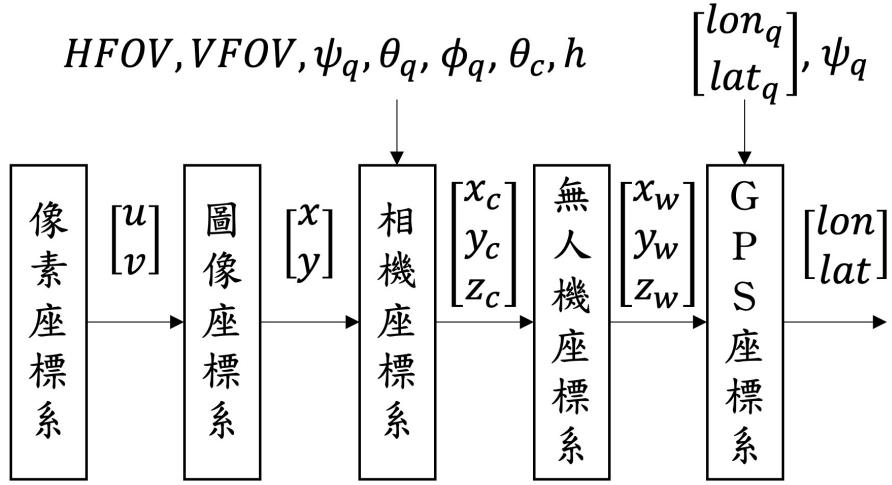


圖 15: 座標轉換流程圖

將像素座標轉換為無人機相對於地平面的世界座標時，過程涉及相機的內參矩陣 K 、旋轉矩陣 R 以及無人機對地高度（相機高度） h 。要算相機的內參矩陣首先需得出影像的水平焦距 f_x 、影像的垂直焦距 f_y 和影像的中心座標 (c_x, c_y) 。

影像的水平焦距 f_x 公式如式 (3)， W 為影像的寬度，HFOV 為相機的水平視角。影像的垂直焦距 f_y 公式如式 (4)， H 為影像的寬度，VFOV 為相機的垂直視角。

$$f_x = \frac{W}{2 \cdot \tan\left(\frac{HFOV}{2}\right)} \quad (3)$$

$$f_y = \frac{H}{2 \cdot \tan\left(\frac{VFOV}{2}\right)} \quad (4)$$

影像的中心座標 (c_x, c_y) 公式如式 (5)、式 (6)：

$$c_x = \frac{W}{2} \quad (5)$$

$$c_y = \frac{H}{2} \quad (6)$$

式 (7) 為相機內參矩陣 K 。

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

旋轉矩陣 R 用來表示相機相對於世界座標系的旋轉。無人機共有三個旋轉角度：俯仰角 θ_q 、滾轉角 ϕ_q 、航向角 ψ_q 。航向角旋轉矩陣 R_{yaw} 、俯仰角旋轉矩陣 R_{pitch} 、滾轉角旋轉矩陣

R_{roll} 的公式分別是式(8)、式(9)、式(10)。

$$R_{yaw} = \begin{bmatrix} \cos \psi_q & -\sin \psi_q & 0 \\ \sin \psi_q & \cos \psi_q & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

$$R_{pitch} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_q + \theta_c) & -\sin(\theta_q + \theta_c) \\ 0 & \sin(\theta_q + \theta_c) & \cos(\theta_q + \theta_c) \end{bmatrix} \quad (9)$$

$$R_{roll} = \begin{bmatrix} \cos \phi_q & 0 & \sin \phi_q \\ 0 & 1 & 0 \\ -\sin \phi_q & 0 & \cos \phi_q \end{bmatrix} \quad (10)$$

其組合的旋轉矩陣 R 如式(11)、式(12)：

$$R = R_{yaw} \cdot R_{pitch} \cdot R_{roll} \quad (11)$$

$$R = \begin{bmatrix} \cos \psi_q & -\sin \psi_q & 0 \\ \sin \psi_q & \cos \psi_q & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_q + \theta_c) & -\sin(\theta_q + \theta_c) \\ 0 & \sin(\theta_q + \theta_c) & \cos(\theta_q + \theta_c) \end{bmatrix} \cdot \begin{bmatrix} \cos \phi_q & 0 & \sin \phi_q \\ 0 & 1 & 0 \\ -\sin \phi_q & 0 & \cos \phi_q \end{bmatrix} \quad (12)$$

將像素座標系 (x, y) 轉換為圖像座標系的公式如式(13)：

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K^{-1} \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (13)$$

將圖像座標系轉換為相機座標系的公式如式(14)：

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = R \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (14)$$

將相機座標系轉換為世界座標系 (x_w, y_w, z_w) 的公式見式(15)、式(16)，此公式即可獲的以無人機(相機)與偵測到的車輛之相對位置。

$$x_w = \frac{x_c}{z_c} \quad (15)$$

$$y_w = \frac{y_c}{z_c} \quad (16)$$

在最後的整合階段，無人機的 GPS 位置作為已知參考點，結合車輛與無人機之間的相對

位置，透過 Pyproj 函式庫 [19] 將這些相對位置轉換為標準的 GPS 坐標。Pyproj 是基於 PROJ 庫的強大 Python 座標轉換與地理投影工具，專為地理資訊系統 (GIS) 設計。它提供靈活且精確的座標系統轉換功能，能夠在全球與區域性坐標系統之間進行準確變換，廣泛應用於地圖服務、導航、遙測影像分析等領域。

3.4 空間對位

透過獲取無人機的絕對位置以及相關的飛行參數，可以應用透視變換技術，將無人機拍攝的影像中的像素座標轉換為相對應的 GPS 座標。這一過程不僅能夠將影像中的物體位置準確地映射到真實的地理座標系統中，還可以達成對影像位置的精確校正，實現空間對位。在此基礎上，空間對位的技術將被應用於 QGIS(Quantum GIS)[20]，這是一款開源的桌面地理信息系統應用程序，廣泛應用於地理數據的分析與處理。

QGIS 支持各種地理信息數據格式，並提供豐富的工具與功能，幫助用戶進行數據的導入、匯出及分析。在這項研究中，我將使用 QGIS 進行空間對位，驗證無人機所拍攝影像中車輛位置的準確性，並與實際地面的位置進行比對。透過無人機的拍攝影像，可以提取影像中的車輛位置，並將其與已知的 GPS 座標進行對比。

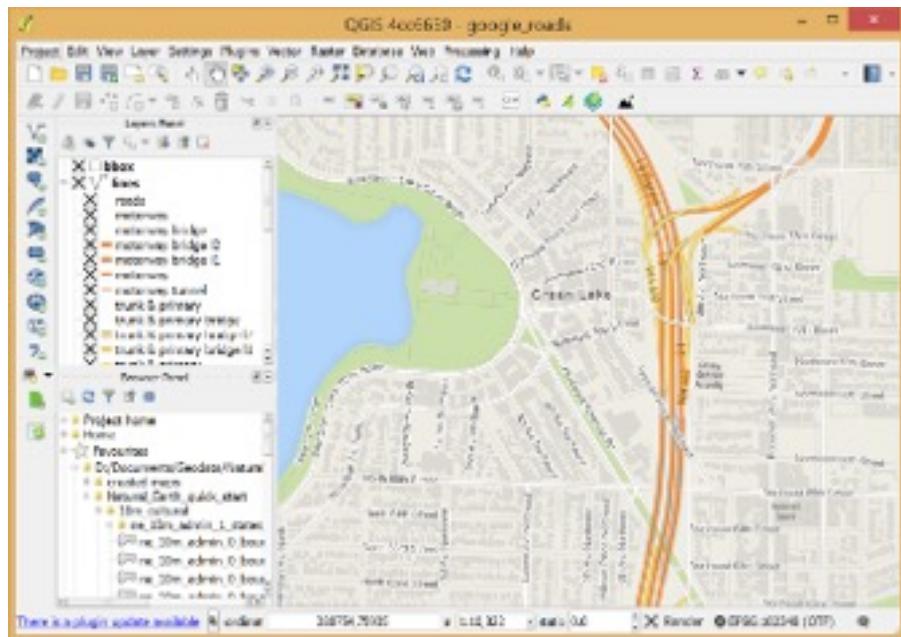


圖 16: QGIS 操作畫面 [21]

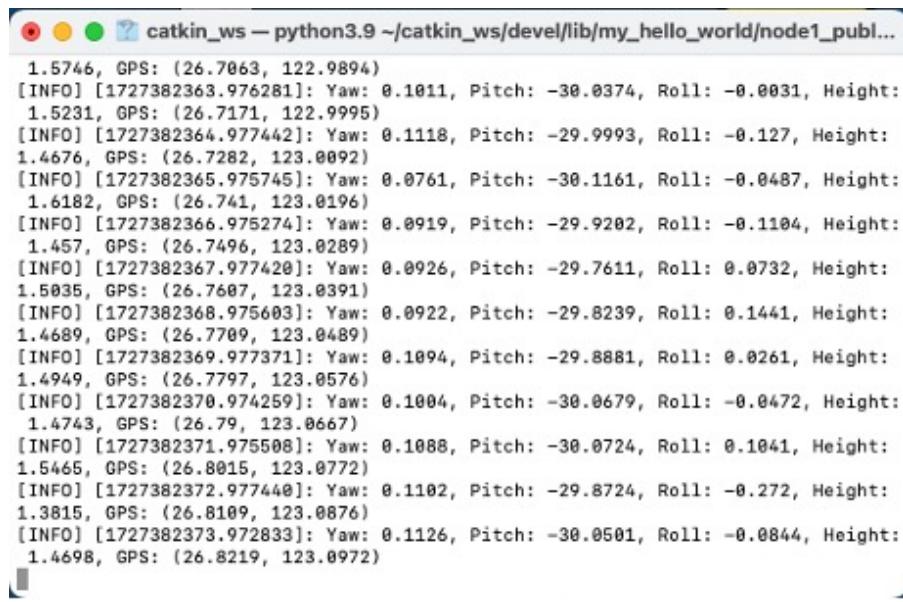
利用透視變換，將影像中的像素座標轉換成地理座標，並使用 QGIS 來執行這些座標的空間對位，將影像中的車輛位置對應到地面實際的位置。接著，將這些透視變換計算得出的 GPS 座標與現場實際測量得到的 GPS 座標進行比對，以檢視兩者之間的誤差，從而評估該方法在實際應用中的精度與可靠性。此方法不僅能夠驗證無人機拍攝影像的地理位置準確性，還能夠提供精確的車輛位置追蹤。

4 研究結果

研究結果主要分成幾部分，包含飛行路徑規劃、IMU 資料獲取

4.1 飛行路徑規劃

4.2 IMU 資料獲取



The screenshot shows a terminal window titled "catkin_ws — python3.9 ~/catkin_ws/devel/lib/my_hello_world/node1_publ...". The window displays a series of [INFO] log messages. Each message contains a timestamp (e.g., 1.5746), GPS coordinates (e.g., (26.7063, 122.9894)), and attitude data (Yaw, Pitch, Roll, Height). The data is repeated approximately 20 times.

```
1.5746, GPS: (26.7063, 122.9894)
[INFO] [1727382363.976281]: Yaw: 0.1011, Pitch: -30.0374, Roll: -0.0031, Height: 1.5231, GPS: (26.7171, 122.9995)
[INFO] [1727382364.977442]: Yaw: 0.1118, Pitch: -29.9993, Roll: -0.127, Height: 1.4676, GPS: (26.7282, 123.0092)
[INFO] [1727382365.975745]: Yaw: 0.0761, Pitch: -30.1161, Roll: -0.0487, Height: 1.6182, GPS: (26.741, 123.0196)
[INFO] [1727382366.975274]: Yaw: 0.0919, Pitch: -29.9202, Roll: -0.1104, Height: 1.457, GPS: (26.7496, 123.0289)
[INFO] [1727382367.977420]: Yaw: 0.0926, Pitch: -29.7611, Roll: 0.0732, Height: 1.5835, GPS: (26.7607, 123.0391)
[INFO] [1727382368.975603]: Yaw: 0.0922, Pitch: -29.8239, Roll: 0.1441, Height: 1.4689, GPS: (26.7709, 123.0489)
[INFO] [1727382369.977371]: Yaw: 0.1094, Pitch: -29.8881, Roll: 0.0261, Height: 1.4949, GPS: (26.7797, 123.0576)
[INFO] [1727382370.974259]: Yaw: 0.1004, Pitch: -30.0679, Roll: -0.0472, Height: 1.4743, GPS: (26.79, 123.0667)
[INFO] [1727382371.975508]: Yaw: 0.1088, Pitch: -30.0724, Roll: 0.1041, Height: 1.5465, GPS: (26.8015, 123.0772)
[INFO] [1727382372.977440]: Yaw: 0.1102, Pitch: -29.8724, Roll: -0.272, Height: 1.3815, GPS: (26.8109, 123.0876)
[INFO] [1727382373.972833]: Yaw: 0.1126, Pitch: -30.0501, Roll: -0.0844, Height: 1.4698, GPS: (26.8219, 123.0972)
```

圖 17: mavlink

4.3 車輛偵測



圖 18: Yolov7 車輛偵測與追蹤



圖 19: Yolov7 車輛偵測與追蹤-無人機實拍

4.4 影像處理

4.5 逆透視變換



圖 20: 地板測試

4.6 空間對位

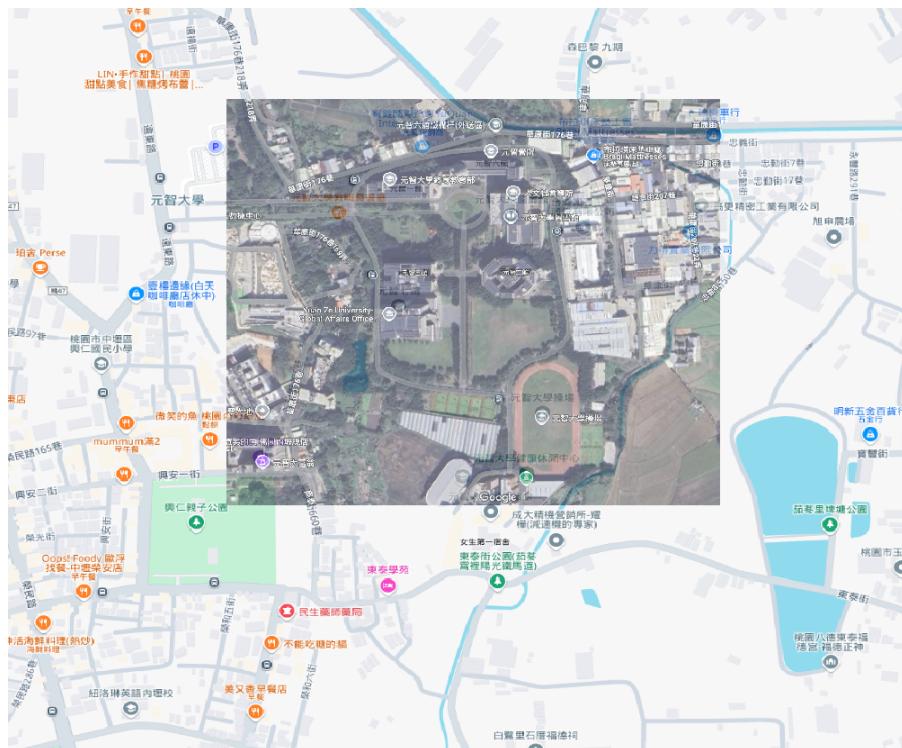


圖 21: 空間對測試

5 結論

6 參考文獻

- [1] 中時新聞網, 2019. <https://www.chinatimes.com/newspapers/20190304000187-260309?chdtv>.
- [2] 中央通訊社, 2022. <https://www.cna.com.tw/news/asoc/202210030248.aspx>.
- [3] 臺中市政府全球資訊網, 2021. <https://www.taichung.gov.tw/1783403/post>.
- [4] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *arXiv:1506.01497 [cs.CV]*, 2015.
- [5] deephub, 2023. <https://cloud.tencent.com/developer/article/2212232?areaSource=104001.4&traceId=bTXJdiCqPM48qF63XTtFY>.
- [6] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 7464–7475.
- [7] YungHuiHsu, 2019. <https://hackmd.io/xhLeIsoSToW0jL61QRWDcQ>.
- [8] 內政部國土測繪中心, *E-gnss 即時動態定位系統入口網站*, <https://egnss.nlsc.gov.tw>, 2020.
- [9] 趙健哲, 鄭傑文, 陳詹閔, 王匯智, 林義乾, 莊子毅, 鄭傑中, and 彭念豪, *航空攝影測量學講義*. 中華民國地籍測量學會, Sep. 2013.
- [10] P. Tsao, “以保距空拍影像拼接提供地面物件定位,” 交通大學網路工程研究所學位論文, 2018.
- [11] 陳嘉祥, “基於雲端運算之即時影音串流轉碼服務平台之研製,” 交通大學網路工程研究所學位論文, 2011.
- [12] 3.-1. R. 傳輸協議, 2018. <https://ithelp.ithome.com.tw/articles/10206368>.
- [13] logitech, *Logitech c922 pro stream 直播攝影機*. <https://store.logitech.tw/products/c922-pro-stream-%E7%B6%B2%E8%B7%AF%E6%94%9D%E5%BD%B1%E6%A9%9F>.
- [14] *Mavproxy - ardupilot*, <https://ardupilot.org/mavproxy/>, 2023.
- [15] Pixhawk, *Pixhawk 2.1 cube mission planner voltage and current problem*, <https://discuss.px4.io/t/pixhawk-2-1-cube-mission-planner-voltage-and-current-problem/8178>, 2023.
- [16] S.-F. Lin, J.-Y. Chen, and H.-X. Chao, “Estimation of number of people in crowded scenes using perspective transformation,” *IEEE*, vol. 31, no. 6, 2001.
- [17] M. Bertozzi, A. Broggi, and A. Fascioli, “Stereo inverse perspective mapping: Theory and applications,” *Image and vision computing*, vol. 16, no. 8, pp. 585–590, 1998.
- [18] H. A. Mallot, H. H. Bülthoff, J. J. Little, and S. Bohrer, “Inverse perspective mapping simplifies optical flow computation and obstacle detection,” *Biological cybernetics*, vol. 64, no. 3, pp. 177–185, 1991.
- [19] pyproj, *Pyproj documentation*. <https://pyproj4.github.io/pyproj/stable/>.
- [20] QGIS Development Team, *Qgis geographic information system*, 2021. <https://www.qgis.org>.

[21] *Qgis official website*, 2020. <https://qgis.org/en/site/about/index.html>.