

國立臺灣大學工學院土木工程學系

碩士論文

Department of Civil Engineering

College of Engineering

National Taiwan University

Master's Thesis



以無人機建立即時自動化橋梁裂縫影像辨識系統

Developing a Real-time Automated Bridge Crack
Detection System Using Unmanned Aerial Vehicles
(UAVs)

蔡宜真

Yi-Jinn Tsai

指導教授: 曾惠斌 博士

Advisor: Hui Ping Tserng, Ph.D.

中華民國 113 年 6 月

June, 2024





誌謝

感謝老師，感謝學長的鞭策跟點子以及數不盡的開會飲料，感謝研究室的同學跟我一起討論模型的做法，感謝中興工程的學長提供橋檢報告，感謝 hplab 的同學載我去測橋，感謝所有提供精神支持的朋友們，感謝爸媽供我在台北讀了兩年的書，感謝在最後關頭有買到新的 Tello。



摘要

在建築物和橋梁的生命週期中，使用階段佔了絕大部分的時間和資源，這包括了後續的維護和定期檢修，因此要如何更有效地管理和監測它們成為一個極為重要的議題。傳統的檢測方法主要依賴目視檢測，這需要花費大量的時間和人力成本。尤其在橋梁檢測方面，橋檢人員為了監測特定區域，必須承受高安全風險。而在建築物檢測方面，通常需要監測有倒塌風險的建築物，這使得監測人員面臨極高的生命風險，因此，如何改善這些監測方法成為一個迫切需要解決的問題。

近年來，隨著 AI (Artificial Intelligence) 相關技術的發展，尤其是深度學習演算法的興起，目前已經許多研究學者將無人機 (Unmanned Aerial Vehicle, UAV) 和影像辨識結合，利用無人機的高機動性以解決因位置不便而橋檢人員難以到達的問題，或是無人機的大面積偵查能力提高工作效率。然而，過去的研究大多僅將無人機用於現場拍攝，再將所拍攝的影像做後續的影像處理和辨識，這樣需花費額外人力和時間成本，而且導致缺乏即時性，使用者無法即時獲取影像資訊。

隨著 ROS (Robot Operating System) 系統的蓬勃發展，這一問題迎刃而解，ROS 系統為軟硬體整合提供了一個便利的平台，藉此，無人機影像辨識就能夠實現即時化，從而進一步提升了監測效率和即時性。本研究旨在結合無人機和影像辨識模型，透過 ROS 系統建立一個即時的裂縫辨識系統。



該系統首先將無人機捕獲的影像進行一系列的影像處理步驟，最終將處理完的影像，匯入已訓練好的影像辨識模型進行裂縫辨識。辨識結果將被儲存，同時向使用者發出警示通知，實現即時的裂縫檢測和通報功能。這樣的整合機制不僅提高了工作效率，也確保了即時性和準確性，有望未來能在建築物及橋梁的監測和維護中發揮重要作用。

除此之外，該系統還可應用於災後搜救和災害評估，提高搜救工作的效率和準確性。本研究旨在建構一個即時的裂縫影像辨識系統，以提高建築物和橋梁監測作業的效率和安全性，以應對日益嚴峻的社會需求。

關鍵字：深度學習、電腦視覺、ROS



Abstract

In the life cycle of buildings and bridges, the operational phase occupies the majority of time and resources, including subsequent maintenance and periodic inspections. Effective management and monitoring of these structures have become crucial issues, given the substantial investment required. Traditional inspection methods primarily rely on visual assessments, which demand significant time and human resources. Particularly in bridge inspections, personnel face high safety risks when monitoring specific areas. Similarly, building inspections often require monitoring structures at risk of collapse, posing severe life-threatening risks to the inspectors. Thus, improving these monitoring methods is an urgent problem that needs to be addressed.

In recent years, with the development of Artificial Intelligence (AI) technologies, especially the rise of deep learning algorithms, many researchers have begun integrating Unmanned Aerial Vehicles (UAVs) with image recognition to leverage UAVs' high mobility for addressing the difficulties inspectors face in accessing certain areas, or to en-

hance work efficiency through UAVs' extensive reconnaissance capabilities. However, past studies mostly utilized UAVs for onsite shooting, followed by subsequent image processing and recognition, which incurs additional labor and time costs and lacks real-time capabilities, preventing users from obtaining immediate image information.

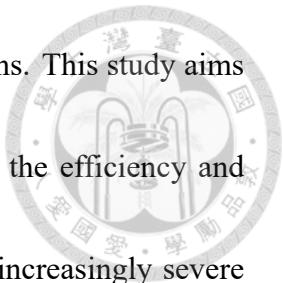
With the rapid advancement of the Robot Operating System (ROS), this issue can now be effectively resolved. The ROS provides a convenient platform for hardware and software integration, enabling real-time image recognition through UAVs, thereby further enhancing monitoring efficiency and immediacy. This study aims to develop a real-time crack recognition system by integrating UAVs and image recognition models through the ROS.

The system first processes images captured by the UAV through a series of image processing steps. The processed images are then input into a pre-trained image recognition model for crack detection. The recognition results are stored and simultaneously send alert notifications to users, achieving real-time crack detection and reporting. This integrated mechanism not only improves work efficiency but also ensures immediacy and accuracy, promising significant contributions to the monitoring and maintenance of buildings and bridges in the future.

Additionally, the system can be applied in post-disaster search and rescue and disaster

assessment, enhancing the efficiency and accuracy of rescue operations. This study aims to construct a real-time crack image recognition system to improve the efficiency and safety of building and bridge monitoring operations, addressing the increasingly severe societal demands.

Keywords: Machine Learning, Computer Vision, Robot Operating System





目次

	Page
口試委員審定書	i
誌謝	iii
摘要	iv
Abstract	vi
目次	ix
圖次	xiii
表次	xv
第一章 緒論	1
1.1 研究背景與動機	1
1.2 研究目的	3
1.3 論文架構	4
第二章 文獻回顧	5
2.1 影像辨識於建物裂縫之發展	5
2.2 無人機影像辨識	8
2.3 小物件影像辨識	11
2.4 小結	12

第三章 研究方法



3.1 硬體介紹	14
3.1.1 NVIDIA Jetson Orin Nano	15
3.1.2 無人機載具	16
3.1.2.1 DJI Tello EDU	16
3.1.2.2 DJI Inspire 2	17
3.2 研究流程圖	18
3.3 影像蒐集及處理	19
3.3.1 裂縫資料集蒐集與建立	20
3.3.2 裂縫標註	21
3.3.3 影像處理	22
3.3.4 資料集分割	23
3.4 模型建置及訓練	24
3.4.1 模型選擇	24
3.4.1.1 損失函數選擇	26
3.4.2 模型驗證指標	30
3.4.2.1 精確度 (Precision)	30
3.4.2.2 召回率 (Recall)	30
3.4.2.3 mAP (mean Average Precision)	31
3.4.3 調整模型架構	32
3.4.3.1 YOLOv7	32
3.4.3.2 YOLOv7-UAV	33
3.4.4 調整模型參數	34
3.5 無人機即時影像辨識	37
3.5.1 無人機影像取得	38

3.5.1.1	DJI Tello EDU	38
3.5.1.2	DJI Inspire 2	39
3.5.2	影像處理	40
3.5.2.1	DJI Tello EDU	40
3.5.2.2	DJI Inspire 2	41
3.5.3	YOLOv7 模型	42
3.6	裂縫警示	42
3.7	流程圖	43
3.7.1	DJI Tello EDU	43
3.7.2	DJI Inspire 2	44
3.8	節點圖	45
3.8.1	DJI Tello EDU	45
3.8.2	DJI Inspire 2	46
3.9	小結	46
第四章	研究結果	47
4.1	裂縫辨識模型	47
4.1.1	設備規格	47
4.1.2	模型架構	48
4.1.3	YOLOv7 初步參數微調	49
4.1.3.1	Batch Size	49
4.1.3.2	Loss Function	51
4.1.3.3	Learning Rate	53
4.1.3.4	Epoch、凍結層	55
4.1.3.5	Evolution	58
4.1.4	YOLOv7-UAV 初步參數微調	59

4.1.4.1	Batch Size	59
4.1.4.2	Loss Function	61
4.1.4.3	Learning Rate	63
4.1.4.4	Epoch、凍結層	65
4.1.4.5	Evolution	68
4.1.5	實際裂縫影像測試	69
4.1.6	實際裂縫影片測試	70
4.2	無人機影像辨識系統實測	71
4.2.1	DJI Tello EDU	71
4.2.1.1	建物裂縫辨識	72
4.2.1.2	一壽橋	73
4.2.2	DJI Inspire 2	74
4.2.2.1	一壽橋	75
4.2.2.2	後龍觀海大橋	76
4.3	實驗結論與探討	77
第五章	結論與未來發展	78
5.1	結論	78
5.2	限制與討論	79
5.3	未來發展	81
參考文獻		82



圖次

2.1 物件偵測里程碑圖	5
3.1 無人機圖	17
3.2 研究流程圖	18
3.3 影像蒐集及處理流程圖	19
3.4 公開資料集影像示意圖	20
3.5 標註示意圖	21
3.6 影像處理手法示意圖	22
3.7 資料集分割比例圖	23
3.8 IoU 示意圖	27
3.9 GIoU 示意圖	27
3.10 CIoU 示意圖	28
3.11 SIoU 示意圖	29
3.12 YOLOv7 架構圖	32
3.13 YOLOv7-UAV 架構圖	33
3.14 基因演算法流程圖	36
3.15 DJI Tello EDU 影像蒐集流程圖	38
3.16 DJI Inspire 2 影像蒐集流程圖	39
3.17 DJI Tello EDU 影像格式轉換流程圖	40
3.18 DJI Inspire 2 影像格式轉換流程圖	41
3.19 裂縫警示流程圖	42
3.20 DJI Tello EDU 流程圖	43
3.21 DJI Inspire 2 流程圖	44
3.22 DJI Tello EDU 節點圖	45



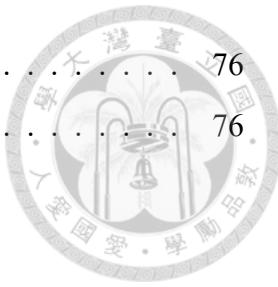
3.23 DJI Inspire 2 節點圖	46
4.1 YOLOv7 Batch 參數調整 mAP 圖	50
4.2 YOLOv7 Batch 參數調整 Loss 圖	50
4.3 YOLOv7 Loss 參數調整 mAP 圖	52
4.4 YOLOv7 Loss 參數調整 Loss 圖	52
4.5 YOLOv7 LR 參數調整 mAP 圖	54
4.6 YOLOv7 LR 參數調整 Loss 圖	54
4.7 YOLOv7 Epoch、凍結層參數調整 mAP 圖	57
4.8 YOLOv7 Epoch、凍結層參數調整 Loss 圖	57
4.9 YOLOv7-UAV Batch 參數調整 mAP 圖	60
4.10 YOLOv7-UAV Batch 參數調整 Loss 圖	60
4.11 YOLOv7-UAV Loss 參數調整 mAP 圖	62
4.12 YOLOv7-UAV Loss 參數調整 Loss 圖	62
4.13 YOLOv7-UAV LR 參數調整 mAP 圖	64
4.14 YOLOv7-UAV LR 參數調整 Loss 圖	64
4.15 YOLOv7-UAV Epoch、凍結層參數調整 mAP 圖	67
4.16 YOLOv7-UAV Epoch、凍結層參數調整 Loss 圖	67
4.17 實際裂縫影片測試圖	70
4.18 DJI Tello EDU 硬體架構圖	71
4.19 建物裂縫辨識結果圖	72
4.20 建物裂縫辨識警示圖	72
4.21 DJI Tello EDU 一壽橋現地實測示意圖	73
4.22 DJI Inspire 2 硬體架構圖	74
4.23 DJI Inspire 2 一壽橋現地實測示意圖	75
4.24 DJI Inspire 2 後龍觀海大橋現地實測示意圖	76



表次

3.1	NVIDIA Jetson Orin Nano 環境版本表	15
3.2	DJI Tello EDU 及 Inspire 2 規格比較表	17
3.3	公開裂縫資料集影像資料表	20
3.4	資料集影像分配表	23
4.1	桌機軟硬體規格	47
4.2	YOLOv7 架構模型 Batch 參數表	49
4.3	YOLOv7 架構模型 Loss 參數表	51
4.4	YOLOv7 架構模型 Learning Rate 參數表	53
4.5	YOLOv7 架構模型 Epoch、凍結層參數表	56
4.6	YOLOv7 超參數進化模型分數表	58
4.7	YOLOv7-UAV 架構模型 Batch 參數表	59
4.8	YOLOv7-UAV 架構模型 Loss 參數表	61
4.9	YOLOv7-UAV 架構模型 Learning Rate 參數表	63
4.10	YOLOv7-UAV 架構模型 Epoch、凍結層參數表	66
4.11	YOLOv7-UAV 超參數進化模型分數表	68
4.12	實際影像資料集測試模型分數表	69
4.13	超參數進化後架構模型分數表	69
4.14	影片拍攝資訊表	70
4.15	影像測試結果表	70
4.16	DJI Tello EDU 一壽橋拍攝資訊表	73
4.17	DJI Tello EDU 一壽橋測試結果表	73
4.18	DJI Inspire 2 一壽橋拍攝資訊表	75
4.19	DJI Inspire 2 一壽橋測試結果表	75

4.20 DJI Inspire 2 後龍觀海大橋拍攝資訊表	76
4.21 DJI Inspire 2 後龍觀海大橋測試結果表	76





第一章 緒論

1.1 研究背景與動機

由於臺灣地形多山，且河川交錯，橋梁的建設成為不可或缺的連接方式。根據截至 2024 年 4 月的統計數據，臺灣目前使用的橋梁總數達到 26,149 座 [1]。由於在建築物的生命週期 (Life Cycle Assessment, LCA) 中，其營運階段佔大部分的時間，佔了絕大部分的資源 [2]，因此橋梁的定期維護至關重要，疏於管理可能導致嚴重的安全隱患。例如，2019 年 10 月 1 日，宜蘭的南方澳橋因轄區問題長期疏於檢測，最終導致橋梁倒塌 [3]。這個事件彰顯了定期檢測橋梁的重要性。除了橋梁的定期檢測之外，建物的維護也是不容忽視的，尤其臺灣位於環太平洋地震帶，地震頻繁，本就屬於需要加強維護橋梁的地區。

傳統的橋梁檢測主要依賴目視檢查，並輔以相關儀器 [4]。這種方式確實可以掌握一定程度的橋梁狀況，但同時也存在一些不足之處，由於進行橋梁檢測需要花費大量的人力和財力成本，橋檢人員往往需要以橡皮艇或吊掛的方式前往橋梁較為難以到達的區域，這不僅增加了檢測的時間成本，也使得檢測人員暴露在一定的風險之中。此外，僅依賴目視和簡單儀器的檢測方式，也難以全面檢測橋梁的實際狀況，尤其是一些隱藏的結構性問題或隱形損傷。因此，如何採用更加先進和全面的橋梁檢測手段，成為了橋梁管理部門面臨的重要課題。

近年來，隨著運算硬體的快速進步，AI 技術也快速興起和發展。其中，深度學習 (Deep Learning, DL) 和電腦視覺 (Computer Vision, CV) 成為了熱門的研究領域。那是因為透過深度學習、電腦視覺等先進的 AI 技術，物件辨識 (Object Detection) 的可取得性和準確性都得到了大幅提升。在過去，進行物件辨識需要大量的手工標註和複雜的程式設計，但隨著深度學習算法的成熟，結合大量的訓練數據，AI 系統能夠自行學習提取影像的關鍵特徵，從而實現更加智能和精準的影像辨識。這使得影像辨識技術在各個領域都得到了廣泛的應用，如自動駕駛、智慧醫療、智慧城市等。

基於 AI 技術的迅速發展，已有許多相關文獻提及應將 AI 技術應用於橋梁檢測上 [5]，也已經有許多文獻證明物件辨識此結合無人機可以進行高速大面積搜索的特性，證實無人機可以滿足進行搜索以及救援行動的需求 [6]，或是可以應用於建物損害的評估中 [7]，AI 技術的應用不僅可以提高檢測的效率和準確性，還能最大限度地降低工作人員在過程中的安全隱患。

本研究將利用 ROS 系統，即時將無人機採集的影像傳輸至開發板，並利用基於深度學習的物體檢測技術，對影像進行即時分析，為檢測人員提供橋梁即時的狀況警示，這不僅可以針對目前目視檢測的現狀進行效率上的改善，還能最大程度降低資源的使用，也能為未來智能化橋梁檢測奠定基礎。



1.2 研究目的

近年來針對於建築物和橋梁的維護工作方法主要依賴目視檢測，這需要大量的時間和人力成本，而且通常還會涉及到高風險的工作環境。特別在橋梁檢測方面，橋檢人員為了監測特定區域，必須承受高生命風險；在建築物檢測的方面，則通常需要監測有倒塌風險的建築物，這使得監測人員暴露在高生命風險中。隨著近年來人工智慧相關技術的飛速發展，尤其是深度學習算法的興起，許多研究開始將無人機 (Unmanned Aerial Vehicles, UAV) 和影像辨識結合。然而，營建領域過往的研究大多僅將無人機用於現場拍攝，而後續的影像處理和辨識則需花費額外人力和時間成本，導致無法立即獲取影像資訊，缺乏即時性。因此，該如何去結合目前的技術去達成即時的影像辨識，是本研究的重點，以下為本研究的目的。

1. 利用無人機高靈活度以及大面積搜索的特性，搭配影像辨識技術，以方便抵達橋檢人員難以檢測的區域做影像紀錄，並同時解決解決橋梁檢測低效率的問題。另外結合開發板搭建影像辨識系統，除了開發板低功率以及便攜性，方便在外地部署系統，另外可以即時辨識橋梁裂縫狀態以監測橋梁安全性。

研究問題：應該要使用何種影像辨識模型或是架構進行裂縫的辨識？需要何種類型的影像建立裂縫資料集？影像處理及影像辨識的自動化該如何去實現？

2. 透過利用開發板將即時取得的無人機影像進行即時的處理以及辨識，並把結果提供給使用者，以達到立即警示的目的。

研究問題：該用何種通訊方式去取得即時的無人機影像？應該如何達成即時辨識的目的？即時的程度是多即時？

3. 透過 ROS 系統進行系統軟硬體整合，將無人機影像即時傳輸至開發板上，

進行影像處理以及影像辨識，除了可以滿足裂縫辨識的即時且自動化的需求，也降低後續處理的時間以及人力成本。

研究問題：使用 ROS 系統有什麼優點？



當今無人機已普及至各個領域，利用無人機進行各種應用已成為近年來的熱門話題，然而，現階段的軟硬體整合還是有許多的進步空間。本研究將利用無人機跟低功率開發板的便攜性及性能，透過 ROS 系統將其串連起來，使無人機可以透過 ROS 系統將影像資訊傳輸傳輸至開發板上，再利用 ROS 上的相關套件進行後續的影像處理及影像辨識、提供警示，建立即時的裂縫辨識系統。此系統後續可以添加更多種類的物件辨識來提昇其通用性，以真正的減輕橋檢人員之負擔，提高工作效率並減少人力成本。因此本研究希望能以運用在實際橋梁例子為目標，以此研究來奠定相關的基礎。

1.3 論文架構

本研究論文將依照以下架構撰寫：第二章將進行文獻回顧，介紹目前裂縫辨識的方法、無人機影像辨識的演算法，以及針對小物件影像辨識的技術。第三章將詳細介紹本研究所採用的研究方法和相關細節。第四章將呈現實驗結果。最後，第五章將對本研究進行總結，提出未來發展方向和相關建議。



第二章 文獻回顧

本研究將針對裂縫影像辨識之現況進行文獻探討，以及即時性裂縫偵測對於現況的不可缺失性，再往下探討 ROS 系統、無人機影像辨識的方法，以及提高辨識小物件準確度的方法。

2.1 影像辨識於建物裂縫之發展

物件辨識的概念早在 2001 年被提出，但基於深度學習的物件辨識是在 2012 年才開始快速發展，在此之前，學業界主要是使用一些傳統的手工特徵提取和機器學習方法來進行物件偵測 [8]，而在災害防範領域，無人機物件辨識技術則是在近年來才逐漸被應用。自 2019 年起，陸陸續續出現了一些相關研究工作，探討利用無人機採集影像數據，結合先進的深度學習模型進行災害現場設施的自動監測和損壞檢測 [7]。

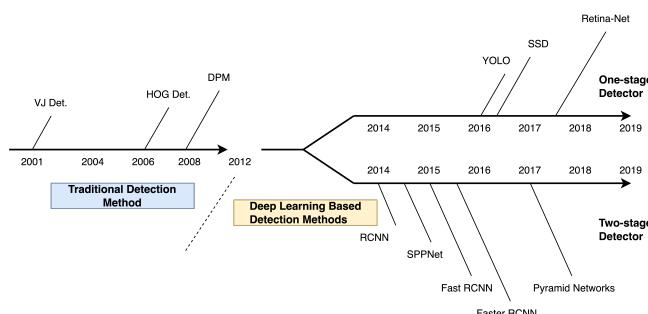


圖 2.1: 物件偵測里程碑圖
[8]

而近年來，國內的相關研究機關也將無人機的裂縫辨識納入發展的範疇之中，例如饒見有等人提出了利用無人機拍攝橋梁影像，再進行後續的 3D (3-Dimensional) 橋梁建模，最後再將劣化利用語義分割 (segmentation) 的方式標註在模型上 [9]；王姿樺等人則是利用 Faster-RCNN (Faster Region-Convolutional Neural Network) 模型搭配 ResNet50 (Residual Neural Network 50)，將無人機在外收集回來的橋梁照片進行影像辨識，其研究結果已經可以用於不同材質的裂縫上面，[10]；張永辰等人利用無人機於後龍大橋拍攝橋梁影像，後續再以預訓練之 YOLOv4 (You Only Look Once version 4th) 模型將影像進行裂縫的辨識，並利用比例尺法進行像素的計算，以 Canny 邊緣檢測及形態學得出裂縫輪廓，計算出裂縫的寬度。[11]。

在國外早已有許多相關的研究，早在 2016 年，Qianyun Zhang et al. 就發表了一篇利用深度學習進行混凝土橋面裂縫即時檢測的研究論文，研究人員利用離散傅立葉變換 DFT (Discrete Fourier Transform) 將橋面圖像轉換到頻域表示，這種轉換使得像素位置和值之間的關係更加直觀，有助於提取圖像中的特徵訊息。為了進一步優化特徵，他們還應用了頻率濾波技術，保留高頻成分並抑制低頻成分。基於這些頻域特徵，研究人員設計了一個結合 1D-CNN (1-Dimensional Convolutional Neural Network) 和 LSTM (Long Short-Term Memory) 的深度學習模型。1D-CNN 負責從頻域信號中提取時序特徵，而 LSTM 則用於建模時間依賴關係，從而達到更高的裂縫檢測精度 [12]。

Kui Luo et al. 提出利用影像視覺技術，用以檢測橋梁表面缺陷、測量振動、識別車輛參數等變得更加智能化和自動化，提高了檢測效率和準確性 [13]；Matthew Norman Gillins et al. 提出無人機系統為橋梁檢測帶來了新的可能，透過快速獲取高解析度的橋梁表面影像，檢測人員可以立即在現場檢視這些影像，大大提高了裂縫辨識的及時性。同時，結合電腦視覺與影像處理技術，這些影像可以

進行自動化的裂縫辨識分析，進一步加快了整個檢測流程。不過若要進行更完整詳細的裂縫評估，仍需將影像傳回實驗室，由專家進行深入的判讀與分析。總的來說，無人機技術的應用，為橋梁檢測的及時性帶來了明顯的改善，但要實現完全的即時辨識，仍需自動化技術的進一步發展 [14]。

Najihah Izzaty Mohd Yusof et al. 這篇論文評估了三種 YOLO (You Only Look Once) 模型 YOLOv5、YOLOv6 和 YOLOv7 在檢測和分類道路缺陷方面的性能。研究發現，YOLOv5 和 YOLOv7 模型在所有 12 個評估的模型中表現最佳，它們的性能非常相似，但 YOLOv7 的推理速度是 YOLOv5 的兩倍，結論為 YOLOv7 整體的表現在無人機的靜態辨識具有精度以及準確性的優勢 [15]。

Jian Zhang et al. 提出了一種基於輕量級視覺模型的自動化橋面裂縫檢測方法。應用 YOLOv4 物件檢測算法進行橋梁鋪面裂縫檢測。為了加速模型，研究替換了 YOLOv4 中的特徵提取網路，使用 DenseNet (Dense Convolutional Network)、MobileNet (Mobile Convolutional Network) 和 GhostNet (Ghost Convolutional Network) 等較輕量的網路架構。為了在實時運行時不犧牲精度，提出了改進的 YOLOv4 裂縫檢測方法。改進包括最佳化錨框、調整網路架構、引入注意力機制以及訓練技巧等。其提出的基於輕量級視覺模型的橋面裂縫檢測方法適用於有限計算資源的實際場景應用，例如無人機、便攜式檢測設備和智能機器人等。它利用了深度學習的優勢，避免了傳統方法需要大量人工參與和工作量巨大的缺陷。通過輕量級網路設計和算法優化，這種橋面裂縫檢測方法不僅在精確度上有優異表現，且達到實時檢測運行速度，並將模型大小控制在合理範圍內，具有良好的實用價值和潛力廣泛應用於基礎設施健康監測 [16]。



2.2 無人機影像辨識

影像辨識技術近年來飛速發展，已被廣泛應用於各式各樣的設備和領域之中。從自動駕駛車輛識別障礙物，到智慧手機上的人臉解鎖功能，再到工業自動化設備中物品的檢測與分類，影像辨識技術正在深刻改變我們的生活和工作方式。其高度自動化、精確性和靈活性，使其成為各類應用中不可或缺的關鍵技術，而在基礎設施管理這個領域，影像辨識技術的強大潛力也逐漸被挖掘。

將無人機應用於影像辨識的技術中，相較於一般影像處理，存在一些困難。首先是影像品質問題，高速飛行和廣泛航拍容易產生模糊、失焦等失真，加上惡劣天氣條件下的干擾，都會降低影像的清晰度和解析度。其次是視角偏斜問題，無人機拍攝的影像往往呈現斜視或仰角視角，與一般正交拍攝大不相同，這給後續的影像配準、三維重建等處理帶來了一定難度。此外，無人機所處的環境也較為複雜，存在大量的陰影、遮擋物等干擾因素，這對目標檢測演算法的泛化性提出了挑戰。同時，基礎設施檢測需要即時的狀況掌握，但無人機影像處理過程中存在一定的時間延遲，如何實現快速高效的分析也是一大難題。

Xin Wu et al. 提及基於無人機的影像辨識，相較於一般的影像辨識在影像方面有著其他的問題，例如物體尺寸小、衛星和載人飛機物體的透視特異性、背景複雜性等問題。另外無人機獲得的影片中的物體檢測或跟蹤需要快速準確地定位移動的地面上物體，因此實時處理的效能至關重要。此篇也比較 SSD(Single Shot MultiBox Detector)、Faster R-CNN、FPN、ClusDet(Clustered Detection)、DMDet(Density-Map guided object detection Network)、UAV-YOLO、ComNet(Combinational Neural Network)、SlimYOLOv3、DAGN(Depthwise-separable Attention-Guided Network)、FS SSD(Feature fusion and Scaling-based Single Shot MultiBox Detector) 在無人機的物體辨識結果，所提出的結論是若要進行無人機的靜態物體偵

測，YOLO 和 SSD 等基於單階段檢測的模型在速度上具有優勢 [17]。

Alison Cleary et al. 利用無人機收集空拍影像，再透過 ROS 系統將所拍攝的影像傳輸至無人機上的開發板，透過訓練好的 Faster R-CNN 進行目標檢測，並採用了轉移學習和資料增強技術來提高模型性能，以達到即時的車輛辨識 [18]；Julian Kedysa et al. 中有提及有研究人員利用無人機影像，搭配 YOLOv3 模型，針對 RC 柱子的損害，以及鋼筋暴露去進行影像辨識，實際應用在建物損害的評估中 [7]。

另外 Qiwen Qiu et al. 也利用一系列的 YOLO 模型應用至無人機影像的影像辨識上，以用於人行道磁磚裂縫的即時裂縫檢測，該文獻訓練了至少 40 個 epoch 跟導入至少 3200 張的影像做訓練，其精準度至少有 80% 以上，也在文章中建議無人機需以離監測物體 1-2 米的距離進行拍攝，以免造成危險 [19]；Chen Xing et al. 採用了修改後的 VGG-16 (Visual Geometry Group 16) 網路，與原始 VGG-16 不同，這個網絡保留了較淺層的特徵圖，並將其輸入到邊界框回歸器，以更好地捕捉小型物體的訊息。同時，論文還採用了深度壓縮技術來減少網絡參數，提高檢測速度。在數據增強方面，透過縮小圖像垂直方向的尺度來模擬無人機拍攝角度的特點，增加了模型對小物體的泛化能力。此外，論文還在損失函數設計上做了相應的調整，以使得位置偏差在小物體檢測中佔有更大的權重。結果顯示，在 PASCAL VOC (PASCAL Visual Object Classes)2012 數據集上，提出的網絡達到了 61.8% 的平均精度 (Mean Average Precision, mAP) [20]。

從上述的研究實例可以看到，針對無人機的辨識模型可以分為兩大類，一種是利用像是 DLNNs (Deep Learning Neural Networks)、Faster-RCNN 的 One-stage 檢測器，是一種直接在單個步驟中檢測和定位物件的方法。這意味著它們不需要額外的階段或者過程，而是使用單個神經網絡模型來執行。這使得 One-stage 檢測器非常簡單高效，因為它們的訓練和推斷速度都較快。此外，在實時應用中，它們

通常表現良好，因為只有一個步驟。然而，One-stage 檢測器的定位準確性可能較差，特別是對於小型或者密集的物件。在複雜場景或者多尺度物件檢測中，其性能可能不如 Two-stage 檢測器。

另一種是像 YOLO 的 Two-stage 檢測器，將物件檢測分為兩個獨立的階段。首先，它們會生成候選區域 (Region Proposals)，然後對這些候選區域進行分類和定位，這使得 Two-stage 檢測器在準確性方面表現出色，因有兩個階段，可以更好地定位和分類物件。由於可以在第一個階段生成不同尺度和形狀的候選區域，因此對於複雜場景和多尺度物件檢測更為適用。然而，Two-stage 檢測器的訓練和推斷複雜度都較高，這可能會導致相對較慢的推斷速度，並且在一些實時應用中可能不太適用。總而言之，One-stage 檢測器具有簡單高效和實時性好的優點，但準確性較低；而 Two-stage 檢測器則具有準確性高和適應性強的優點，但複雜度和效率較低，但在無人機的影像辨識上，效率跟準確性應達到良好的平衡。

無人機可以快速、靈活地對廣闊範圍內的橋梁進行航拍，獲取高清晰度的影像資料。配合成熟的影像辨識系統，就能夠自動識別出橋梁表面的各種損壞情況，如裂縫、剝落等。這不僅大幅提高了檢測的效率和覆蓋面，更能夠及時發現並定位問題，為後續的維修工作提供關鍵訊息，可以預見，隨著影像辨識和無人機技術的不斷進步，它們必將在基礎設施管理領域發揮越來越重要的作用，為提升相關工作的智能化水平貢獻力量，但因為無人機需跟拍攝物離一定的距離，加上現在許多市面上的無人機都有包含防撞的安全措施，所以做無人機裂縫的影像辨識，就需要克服距離以及偵測目標過於渺小的問題。



2.3 小物件影像辨識

除了要克服無人機的影像品質，沒辦法像普通的影像那麼高之外，由於裂縫是較微小的物件，當物件較小時，要取得特徵值相對於大型物件困難很多，所以有許多研究學者專門優化模型，或是進行相對應的解決方案，以達到該目的。

Qianyun Zhang et al. 提出了一些針對提取小物件特徵的方法。採用了一維卷積神經網路 (1D-CNN) 來提取圖像頻域的特徵，相比傳統的二維卷積，1D-CNN 能更有效地捕捉細微的裂縫特徵，另外將 1D-CNN 與長短期記憶網路 LSTM 相結合，用於融合 1D-CNN 提取的特徵序列。LSTM 擅長建模特徵間的長時依賴關係，這有助於檢測出狹窄的裂縫。在預處理方面，作者提出將圖像轉換到頻域進行處理。這一策略可以顯著提高計算效率，因為頻域更能突出小物件的邊緣特徵。具體是採用高通濾波去除低頻背景，保留高頻的邊緣訊息。為了更連續地檢測出細小的裂縫，使用了重疊的滑動窗口方法，而非單一的窗口檢測。這樣可以避免僅檢測到斷斷續續的裂縫片段 [12]。

Tianjie Zhang et al. 提出在不改變深度學習模型架構的情況下，顯著提高了公路裂縫檢測任務的性能。該解決方案為在 YOLOv5 模型中嵌入注意力機制模塊 (Attention module)，提升模型對道路裂縫特徵的關注度，也提出了一種基於裂縫類別的圖像增強策略 (Class-Specific Image Augmentation, CSIA)，根據不同裂縫類別的特點和模型性能自適應生成相應數量的訓練樣本，最後利用正交試驗法優化模型超參數，大幅縮短了參數調優所需的實驗次數，在不影響模型精確度的情況下提高了訓練效率。這些手法大幅提高了模型在公路裂縫檢測任務上的性能 [21]。

也有一些文獻是在專注於無人機的小物件影像辨識上，Siling Feng et al. 該論文針對傳統的 YOLOv7 架構在處理無人機影像時存在一些問題，如小目標訊息容易丟失等問題。為解決這些問題，作者對 YOLOv7 的網路結構做了的改進，首先，他們移除了 YOLOv7 骨幹網路中的 P5 特徵提取層，避免過度下採樣導致小目標訊息流失。其次，設計了一種新的併行骨幹網路結構，包含兩條近乎相同的骨幹網路，可以融合高層語義訊息和低層細節訊息，更好地處理無人機影像的複雜場景。最後在特徵融合層引入了 MLVC (Multiple Frames Prediction for Learned Video Compression) 模塊，利用輕量級的 MLP (Multi-Layer Perceptrons) 捕捉全局上下文訊息，同時使用可視化編碼捕捉局部特徵，增強特徵表徵能力。相比原始 YOLOv7，新架構的 mAP50 提高了 5.5%，mAP 提高了 4.3% [22]；Yingkun Wei et al. 也是利用改進的 YOLOv7 算法，稱為 YOLOv7-UAV，專門用於在無人機影像情境中檢測小物體，其也是移除了 YOLOv7 骨幹網路中的 P5 特徵提取層，並添加了 P2 特徵提取層，以更好地採樣小目標的特徵訊息 [23]。

2.4 小結

從上述文獻中可以得出無人機影像辨識技術已經可以應用於橋梁裂縫辨識、車輛辨識、建物裂縫的辨識，其技術在各種場景中展現出了其價值，另外運用影像視覺的技術檢測橋梁缺陷，可以提高橋梁檢測的效率以及準確性。

現有的橋梁裂縫影像識別研究，大多集中於算法開發和模型優化等軟體層面的創新，而對於如何將這些先進技術與無人機等硬體平台實際集成，卻鮮有深入探討。要真正將裂縫識別應用於實際的橋梁檢測，單純的算法開發是遠遠不夠的，需要解決如何在有限的資源條件下，將影像採集、處理和傳輸等功能高效集成於無人機載荷之中的關鍵問題。缺乏軟硬體融合創新，將嚴重制約裂縫識別技術的

實際應用和推廣，像是在上述的文獻中就有提及可以使用 ROS 系統來達成軟硬體整合的目的。



若能夠解決目前裂縫影像識別系統在軟硬體整合方面的問題，那麼即時性的層面也會迎刃而解。一旦實現了軟硬體的融合，將無人機拍攝的影像，即時傳輸至需做預測的硬體中，就能充分發揮先進的影像分析算法在小型無人機平台上的性能潛力。透過對深度學習模型進行優化、修改架構，或是對圖像進行一系列的預處理，以符合針對裂縫辨識的需求，可以大幅提高裂縫檢測的即時性和可靠性。只有在軟硬體高度融合的基礎上，裂縫識別系統才能真正發揮出即時、高效的監測能力，為進一步實現智慧化橋梁管理奠定基礎。



第三章 研究方法

過去相關研究聚焦於裂縫辨識，通常是利用無人機在現場進行影像拍攝，隨後將影像帶回進行後續處理，包括影像跟演算法的應用，以得到最終的辨識成果。然而這種後續處理的方法意味著使用者無法在第一時間獲取到辨識結果。在面對諸如地震等緊急情況時，即時獲取警示是至關重要的。因此對於這類型的緊急情況，即時得到裂縫辨識結果的能力變得至關重要，而這也需要更加迅速和有效的影像處理技術和演算法，以實現即時警示和應對。

因此本研究總結目前裂縫影像辨識的現況，將利用具有便攜性的開發板，透過 ROS 系統，串連起無人機的影像進行影像辨識，會先訓練好針對於裂縫辨識的影像模型，並將該模型導入系統中，在無人機載具拍攝到影像的同時，透過 ROS 系統將影像傳輸至開發板，在開發板上的 ROS 系統中進行即時的影像處理跟演算法處理，並對應其辨識結果提供使用者警示。



3.1 硬體介紹

3.1.1 NVIDIA Jetson Orin Nano

Jetson Orin Nano 是為 NVIDIA Orin 架構的嵌入式系統，由 1024 個 CUDA 核心和多達 32 個處理人工智慧運算工作的 Tensor 核心組成，相較於一樣是 NVIDIA 發展的 Jetson Nano、Jetson TX2 NX 和 Jetson Xavier NX 有較好的影像處理效能，因此相當適合發展影像處理的機器人，也因為它的便攜性，所以本研究用此做為在外即時影像辨識的主機使用，以下是本研究建設的環境版本 (表3.1)。

表 3.1: NVIDIA Jetson Orin Nano 環境版本表

名稱	版本
Ubuntu	20.04
ROS	Noetic
CUDA	11.4
Pytorch	2.1.0a0
torchvision	0.16.1
OpenCV	4.8.0



3.1.2 無人機載具

本研究所使用的無人機有兩種，分別為 DJI Tello EDU 以及 DJI Inspire 2，會使用兩種無人機載具的原因是由於 DJI Inspire 2 價格昂貴的緣故，且 DJI Tello EDU 的硬體條件緣故，所以在室外進行無人機飛行的泛用性不高，故先透過 DJI Tello EDU 進行整體架構流程的疏通，將其整合於 ROS 系統中，再將整個系統架構轉移至 DJI Inspire 上。

3.1.2.1 DJI Tello EDU

DJI Tello EDU (圖3.1a) 重達 87 克，最高速度為 8 公尺每秒、最長飛行時間為 13 分鐘，最好的環境下訊號距離可以達到 100 公尺，其鏡頭解析度為 720p，影像格式為 HD30FPS，可以使用 Wifi 即時傳輸，因為 DJI Tello EDU 已有許多 ROS 開源的相關套件，所以本研究拿來作為前期整體架構的梳理，並試圖將架構轉移至實用性更高的無人機上。



3.1.2.2 DJI Inspire 2

DJI Inspire 2 (圖3.1b) 重達 3440 克，最高速度為 94 公里每小時、最長飛行時間為 27 分鐘、訊號距離為 7 公里，其鏡頭要另外搭載，本研究使用的相機是 DJI ZENMUSE X5，其鏡頭解析度為 1080p，影像格式為 2.7K30FPS、4K30FPS，以下為 DJI Tello EDU 及 DJI Inspire 2 的規格比較 (表3.2)。

表 3.2: DJI Tello EDU 及 Inspire 2 規格比較表

項目	DJI Tello EDU	DJI Inspire 2
重量	87 克	3340 克
最高速度	8 公尺/秒	26 公尺/秒
最長飛行時間	13 分鐘	27 分鐘
訊號距離	100 公尺	7 公里
鏡頭解析度	720p	1080p
影像格式	HD30FPS	2.7K30FPS、4K30FPS



(a) DJI Tello EDU



(b) DJI Inspire 2

圖 3.1: 無人機圖



3.2 研究流程圖

本研究流程與方法將依照分為以下 (一) 影像蒐集及處理、(二) 模型建置及訓練、(三) 無人機即時影像辨識、(四) 裂縫警示。一開始我們將利用外部公開的裂縫資料集影像進行篩選，並進行手動標註，以建立我們自己的裂縫資料集。同時將採用多種影像處理手法以擴充資料集，提高模型的多樣性。之後將測試針對無人機小物件辨識進行優化的 YOLOv7-UAV 模型架構，以及 YOLOv7 原始的架構，並對兩種架構都進行訓練。通過調整模型參數和優化演算法，對兩個模型架構進行模型分數的比較。最後將把訓練好的模型權重和最適宜的模型架構，導入開發板上的 ROS 系統中。

無人機於實地飛行所取得之影像會透過影像處理跟轉換，將影像資訊傳輸至開發板上的 ROS 系統中，以便進入 YOLOv7 模型中進行辨識，隨後會利用彈跳視窗對於辨識的結果提供相關的警示，以達成在 ROS 平台上實現裂縫的即時辨識和警示功能。

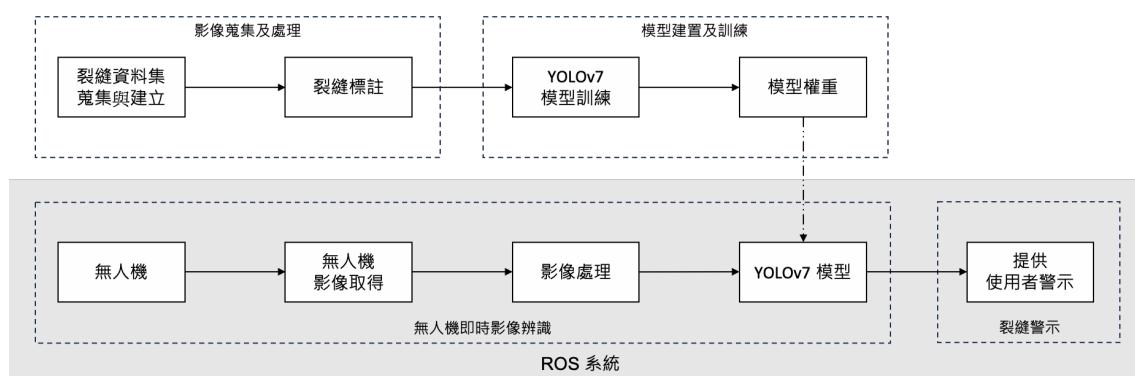


圖 3.2: 研究流程圖



3.3 影像蒐集及處理

本研究將影像蒐集分為兩大部分 (一) 裂縫資料集蒐集與建立 (二) 裂縫標註 (圖3.3)，在第一部分，我們將利用外部公開的資料集進行裂縫影像的收集，由於針對於橋梁裂縫的資料集不好取得，所以採用鋪面或是混凝土裂縫的資料集來進行資料集的建立。我們將蒐集並整理外部公開的裂縫資料集，這些資料集可能包含不同類型、大小和形狀的裂縫影像，是建立模型所需的重要資源之一，以增加其泛用性。

第二部分是為了使判斷標準統一，本研究將所有需訓練的影像進行重新標註，纖細的裂縫以邊界框 (Bounding Box) 的形式進行標註，若為大面積剝落的話以多邊形 (Polygon) 的形式進行標註，以得出更精準的邊界框，本研究的標註平台使用的是 Roboflow (Dwyer B. et al., 2024)。另外利用影像處理方法，增加資料集的影像以及提升資料集的多樣性，以達到良好的泛化效果。

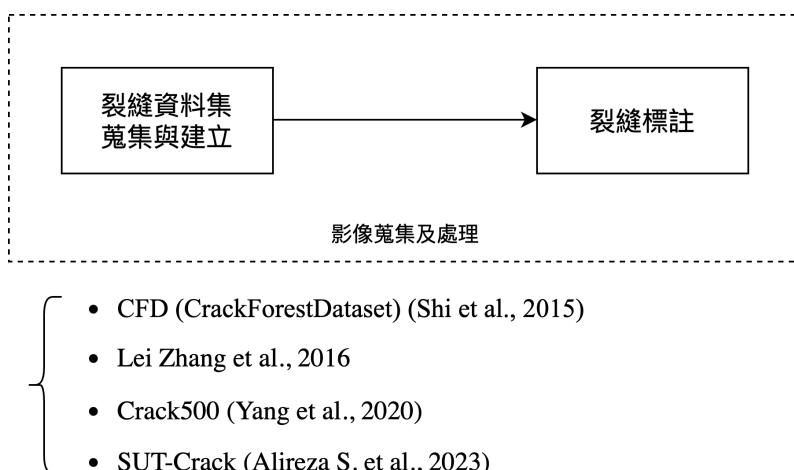


圖 3.3: 影像蒐集及處理流程圖



3.3.1 裂縫資料集蒐集與建立

本研究利用多個公開的裂縫資料集作為基礎，以構建全面且多樣化的裂縫模型訓練資料。其中包括 Yong Shi et al. 於 2015 發表的 CFD (CrackForestDataset) 資料集 [24]、Lei Zhang et al. 於 2016 年發表的混凝土裂縫公開資料集 [25]、Fan Yang et al. 於 2020 年發表的 Crack500 資料集 [26]，以及 Alireza Sepidbar et al. 於 2023 年發表的 SUT-Crack 鋪面裂縫資料集 [27] (圖 3.4)、(表 3.3)，在選擇資料集時，我們專注於挑選出符合橋梁裂縫特徵的影像，以確保資料集的相關性和適用性。透過這樣的資料篩選和整合，我們將能夠建立一個更具代表性和可靠性的裂縫模型訓練資料集，為後續的研究和應用提供可靠的基礎，最後篩選出 979 張影像作為原始資料集使用。

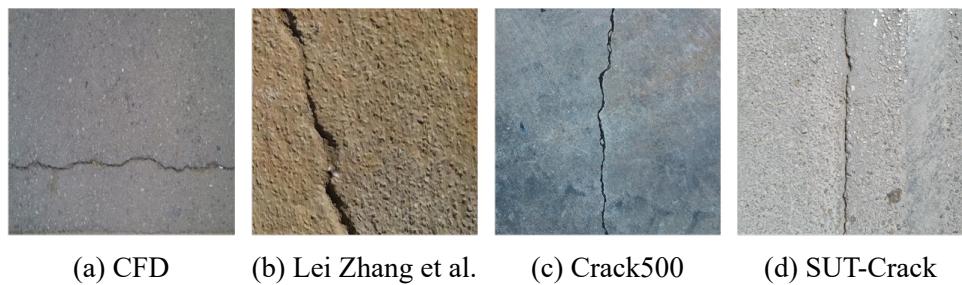


圖 3.4: 公開資料集影像示意圖

表 3.3: 公開裂縫資料集影像資料表

	CFD (CrackForestDataset) (Shi et al., 2015)	Lei Zhang et al., 2016	Crack500 (Yang et al., 2020)	SUT-Crack (A. S. et al., 2023)
簡述	道路鋪面裂縫 彩色影像	道路鋪面裂縫 彩色影像	人行道鋪面裂縫 彩色影像	人行道鋪面裂縫 彩色影像
使用張數	79	200	400	300
影像範例	(圖 3.4a)	(圖 3.4b)	(圖 3.4c)	(圖 3.4d)



3.3.2 裂縫標註

裂縫標註在進入模型訓練前的關鍵意義，在於確保訓練資料的品質和準確性。這是由於標註過程的準確性直接影響著後續模型的訓練效果和應用效能。通過精心的裂縫標註，可以建立出高品質的訓練資料集，從而提高模型的泛化能力和準確性。但為了使判斷標準統一，本研究將資料集進行重新的標註，讓資料集的標註較為平衡。

本研究利用網路上資料庫編輯平臺 Roboflow [28]，來進行資料集的標註跟影像增強，Roboflow 是一個圖像數據管理和準備平台，它幫助用戶將圖像資料整理得更加有效和準備好用於訓練模型。其主要優點在於它提供了自動化的工作流程，可以輕鬆地進行數據標註、轉換和增強。此外，Roboflow 還提供了方便的數據集管理功能，讓用戶可以輕鬆地管理和共享他們的數據集，方便後續要做相似研究的人可以使用。並以邊界框 (Bounding Box) 的形式標註細小的裂縫，有明顯裂縫用多邊形 (Polygon) 的形式進行標註，以取得更精確的邊界框 (圖3.5)。

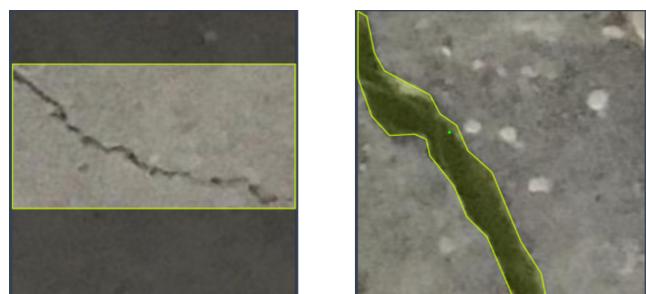


圖 3.5: 標註示意圖



3.3.3 影像處理

影像處理有助於使模型更好地應對各種不同的影像場景和變化，使影像增加數量以彌補資料集不足之處。

本研究為了增加資料集的多樣性，使用了使用了多種影像處理技術，包括將影像做順時鐘 90° 以及逆時鐘 90° 的旋轉 (90° Rotate)，另外也進行了 -15° 到 $+15^\circ$ 範圍內的隨機旋轉 (Rotation)。此外，我們還對影像進行水平和鉛直方向上 $\pm 10^\circ$ 的錯切 (Sheer)，等等方法以進一步擴展資料集 (圖3.6)。最後為了模擬實際拍攝情況下的光線變化，我們隨機調整了影像的亮度 (Brightness)，範圍從 -15% 到 15% 。這些方式不僅可以有效地提高模型的泛化能力，還可以使其更好地適應不同場景下的影像輸入。透過影像處理的手法，本研究將原始資料集 979 張影像擴增至 2937 張影像，作為後續資料集分割使用，第四章節會再針對資料集做細微的調整，以更貼合實地的環境。

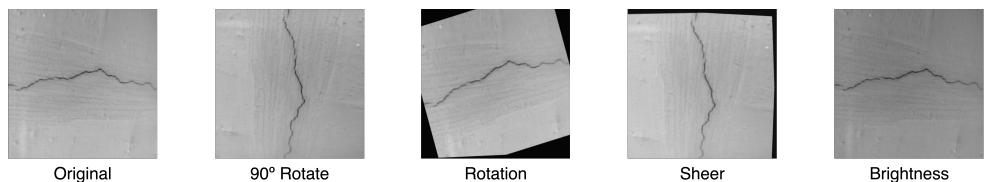


圖 3.6: 影像處理手法示意圖



3.3.4 資料集分割

資料集分割是將一個整體的數據集分成幾個部分的過程，通常是訓練集、驗證集和測試集。這樣的操作有助於確保模型的訓練和評估過程更加準確和可靠。

首先將數據集分成訓練集和測試集的目的是為了訓練和評估模型的性能。訓練集用於訓練模型，而測試集則用於評估模型在未見過數據上的表現，這樣可以確保模型對於不同的數據樣本都有良好的泛化能力。有時還會將數據集進一步分成訓練集和驗證集，這是為了在訓練模型的同時，過度擬合是指模型在訓練集上表現很好，但在測試集上表現很差的情況，透過驗證集可以即時檢測並解決這個問題，在本研究的資料集中，將其拆成訓練集 75%、測試集 15%、驗證集 10% (圖3.7)，其資料增強前後的資料分割為下表 (表3.4)。



圖 3.7: 資料集分割比例圖

表 3.4: 資料集影像分配表

資料集種類	訓練集 Training Set	測試集 Testing Set	驗證集 Validation Set
百分比	75%	15%	10%
影像增強前張數	734	147	98
影像增強後張數	2202	441	294



3.4 模型建置及訓練

在這部分主要分為模型挑選跟架構調整兩大部分，第一部分的模型挑選涵蓋模型的主要選擇，以及後續參數的微調，另外需經由模型分數進行訓練效果的量化判斷；架構調整是當模型沒辦法有顯著提升的時候，就需要針對架構去進行特徵提取層的微調，以更好符合資料集跟模型任務需求，在最後也應需要進行訓練參數的調整以達到該架構下的優化結果。

第二部分是通過反覆調整模型架構和參數，不斷優化訓練過程，以達到最佳的效果和性能。這種迭代的過程將使模型能夠更好地擬合目標，並在實際應用中取得更好的表現，當模型訓練完成後，將進行詳細的性能評估和測試，以驗證其在各種情況下的準確性和穩定性。這可能涉及使用各種指標和測量方法來評估模型的性能，例如精度、召回率、F1 分數等。透過這些評估，可以更全面地了解模型的表現，並在需要時進行進一步的調整和優化。

3.4.1 模型選擇

由於本研究的最終模型需在開發板上執行並進行即時辨識，經過文獻回顧發現，Xin Wu et al. 比較多種模型針對無人機影像辨識的結果，提及 YOLO 系列模型在無人機的靜態影像辨識上具有良好的速度以及具有一定的精確度 [7]，另外，也有文獻針對 YOLOv7 小物件的影像辨識方面進行了研究 [22]，由於對於即時性的需求，因此本研究決定採用 YOLOv7 來作為裂縫影像辨識的模型。

在選定模型後，需要將先前標註好的資料集匯入模型中，以進行特徵提取。然而，從頭訓練一個模型需要耗費大量的資源。為了節省時間和資源，本研究採用預訓練模型。這些模型通常是在大型數據集上進行了長時間的訓練，並且已經

達到了良好的性能。因此，透過微調或遷移學習，可以在現有模型的基礎上進行訓練，從而節省了大量的訓練時間和計算資源。此外，預訓練模型具有較好的準確性和泛化能力，這對於訓練數據較少的情況尤其有用，可以有效地避免過擬合問題，提高模型的性能。本研究也會以 YOLOv7 做為預訓練模型，雖然相較於其他預訓練模型，YOLOv7 的預訓練模型需要較多的資源，但由於其較大的模型規模，更能涵蓋不同種類的資料集，進而提高預訓練模型的效果。



3.4.1.1 損失函數選擇

通常損失函數的選擇也會嚴重影響到模型訓練的結果，因為在訓練過程中，模型的參數會根據損失函數的值進行調整，以最小化損失函數的值。因此，在訓練過程中，損失函數的選擇對於模型的性能和收斂速度都會產生重大影響。以下將列出本研究訓練過程中有使用過的損失函數並進行介紹。

- IoU (Intersection over Union)

是用來衡量目標檢測任務中預測框 (Bounding Box) 和真實目標框之間重疊程度的一個指標。其計算方式是真實目標框與預測框的交集面積除以它們的聯集面積 (式3.1) (圖3.8)，其交集面積的計算方式為 (式3.2)，但 IoU 有一個很明顯的缺點，就是當預測框和真實框沒有交集時，IoU 將無法被計算出來，所以後續的損失函數就有針對這一點進行改善。

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|} \quad (3.1)$$

$$\begin{aligned} \text{交集面積} &= \max(0, \min(x_{\text{true}} + w_{\text{true}}, x_{\text{pred}} + w_{\text{pred}}) - \max(x_{\text{true}}, x_{\text{pred}})) \\ &\quad \times \max(0, \min(y_{\text{true}} + h_{\text{true}}, y_{\text{pred}} + h_{\text{pred}}) - \max(y_{\text{true}}, y_{\text{pred}})) \end{aligned} \quad (3.2)$$

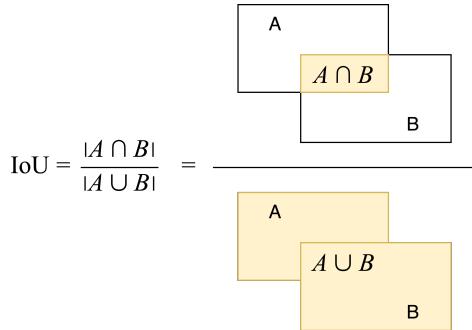


圖 3.8: IoU 示意圖

[29]

- GIoU (Generalized-IoU)

Hamid Rezatofighi et al. 在 2019 年提出了 GIoU [30]，GIoU 解決了先前若真實目標框與預測框沒有交集的問題，也就會導致 IoU 為零的情形，取而代之的是，GIoU 加入了真實目標框與預測框的聯合框面積，比 IoU 更能有泛用性，以更準確地衡量預測框和真實框之間的匹配程度 (式3.3) (圖3.9)，但是 GIoU 也有缺點，當目標框與預測框的高度相同，且處於同一平面時，GIoU 就會退化成 IoU。

$$\text{GIoU} = \text{IoU} - \frac{|C/A \cup B|}{|C|} \quad (3.3)$$

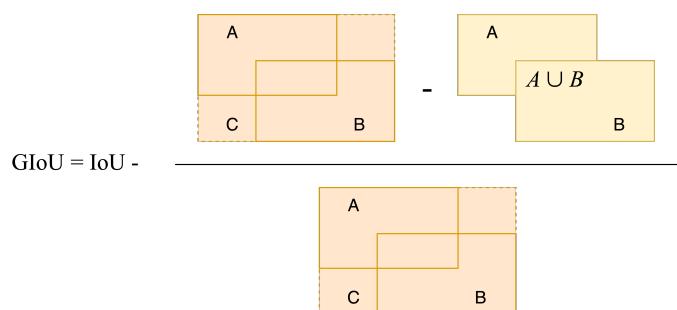


圖 3.9: GIoU 示意圖

[31]



- CIoU (Complete-IoU)

Zhaohui Zheng et al. 在 2022 年提出了一個新的損失函數 CIoU [32]，其除了包含以往真實框與預測框的重疊面積，還有中心點的距離之外，這次還包含了方框的長寬比 (式3.4) (圖3.10)，以避免當真實框與預測框的邊長都相對小時，所造成 IoU 的不準確。

$$CIoU = 1 - IoU - \frac{\rho^2(B, B_{gt})}{c^2} - \alpha v \quad (3.4)$$

其中 $\rho^2(b, b^{gt})$ 是預測框跟真實框中心點的歐幾里得距離；

c 是預測框跟真實框聯集的方框的對角線距離；

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w_{gt}}{h_{gt}} - \arctan \frac{w}{h} \right)^2;$$

$$\alpha = \frac{v}{1 - IoU + v} \text{ } ^\circ$$

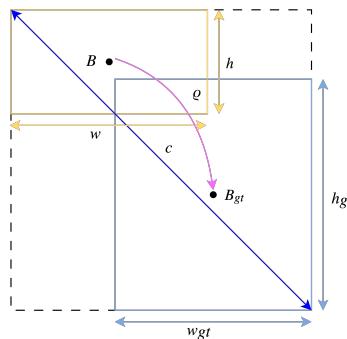


圖 3.10: CIoU 示意圖
[33]

- SIoU (SCYLLA-IoU)

Zhora Gevorgyan 在 2022 提出 SIoU 的概念 [34]，由於當預測邊界框和真實邊界框的長寬比相同時，CIoU 損失函數中的長寬比項為常數 0，導致收斂過程波動較大。而 SIoU 損失函數在計算損失時引入了角度成本，通過考慮預測邊界框和實際邊界框之間連線與水平方向夾角的最小值 (式3.5) (圖3.11)，不僅解決了 CIoU 的問題，也降低了損失函數的自由度，加速了收斂速度。



$$SIoU = 1 - IoU + \frac{\Delta + \Omega}{2} \quad (3.5)$$

其中 $\Lambda = 1 - 2 \times \sin^2(\arcsin(x) - \frac{\pi}{4})$ ；

$$x = \frac{c_h}{\delta} = \sin \alpha ; \delta = \sqrt{(b_{c_x}^{gt} - b_{c_x})^2 + (b_{c_y}^{gt} - b_{c_y})^2} ;$$

$$c_h = \max(b_{c_y}^{gt}, b_{c_y}) - \min(b_{c_y}^{gt}, b_{c_y}) ; \Delta = \sum_{t=x,y} (1 - e^{-\gamma \rho_t}) ;$$

$$\rho_x = \left(\frac{b_{c_x}^{gt} - b_{c_x}}{c_w} \right)^2 ; \rho_y = \left(\frac{b_{c_y}^{gt} - b_{c_y}}{c_h} \right)^2$$

$$\gamma = 2 - \Lambda ; \Omega = (1 - e^{-\frac{|w - w^{gt}|}{\max(w, w^{gt})}})^{\Theta} + (1 - e^{-\frac{|h - h^{gt}|}{\max(h, h^{gt})}})^{\Theta}$$

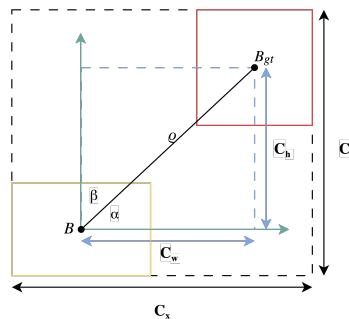


圖 3.11: SIoU 示意圖
[35]



3.4.2 模型驗證指標

通常為了要判斷一個模型的好壞，我們會查看某些模型的數據，來判定是否有過擬合的情形，或是模型是否有足夠的訓練，以下將說明常見的模型數值。

3.4.2.1 精確度 (Precision)

在模型訓練中，精確度 (Precision) 是一個用於衡量模型預測中正確預測的樣本所佔的比例的指標。精確度通常用於二元分類問題中，其中模型的預測結果可以被歸類為正例 (Positive) 或負例 (Negative)，精確度的計算方法為 (式3.6)，簡單來說，精確度就是被模型判斷成正例中，其中真實是正例樣本的比例。

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3.6)$$

3.4.2.2 召回率 (Recall)

召回率 (Recall) 是用於衡量模型在所有實際正例中成功檢測到的比例，即模型能夠正確找出的正例的比例，其計算方法為 (式3.7)，但召回率與精確度是有負相關性的，召回率和精確度之間通常是一種權衡關係，增加召回率可能會導致精確度的降低，反之亦然。例如，如果分類器過於激進，可能會提高召回率但降低精確度。相反，如果分類器過於保守，會導致召回率降低但精確度提高。

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3.7)$$



3.4.2.3 mAP (mean Average Precision)

每當模型訓練完成時就會得到每一種類的精確度以及召回率，我們可以精確度與召回率化在一個座標平面上，AP 就是計算 Precision-Recall curve 底下的面積，將所有類別的 AP 進行平均，就會得到 mAP 值，mAP 是物件偵測中常用的評價指標，它能夠全面評估模型在不同類別上的性能，一個高的 mAP 值代表著模型在識別目標時更加準確和可靠。



3.4.3 調整模型架構

3.4.3.1 YOLOv7

YOLOv7 的架構可以描述為一系列連接的神經網路，Backbone 網路負責從輸入影像中提取特徵表示，這個網路通常由多個卷積層和池化層組成，用於捕捉影像中的局部和全局特徵。接著連接到 Neck，Neck 網路接收來自 Backbone 的特徵表示，並進行進一步的處理和增強。Neck 網路通常包括一系列卷積層和上下採樣操作，以及一些跳連結構，用於融合不同層次的特徵表示。最後連接到 Head 層，Head 網路是 YOLOv7 模型的核心部分，負責進行目標檢測的預測。這個網路包括多個卷積層和激活函數，以及一個特殊的檢測層。檢測層生成目標的檢測框並預測其類別和位置資訊(圖3.12)。

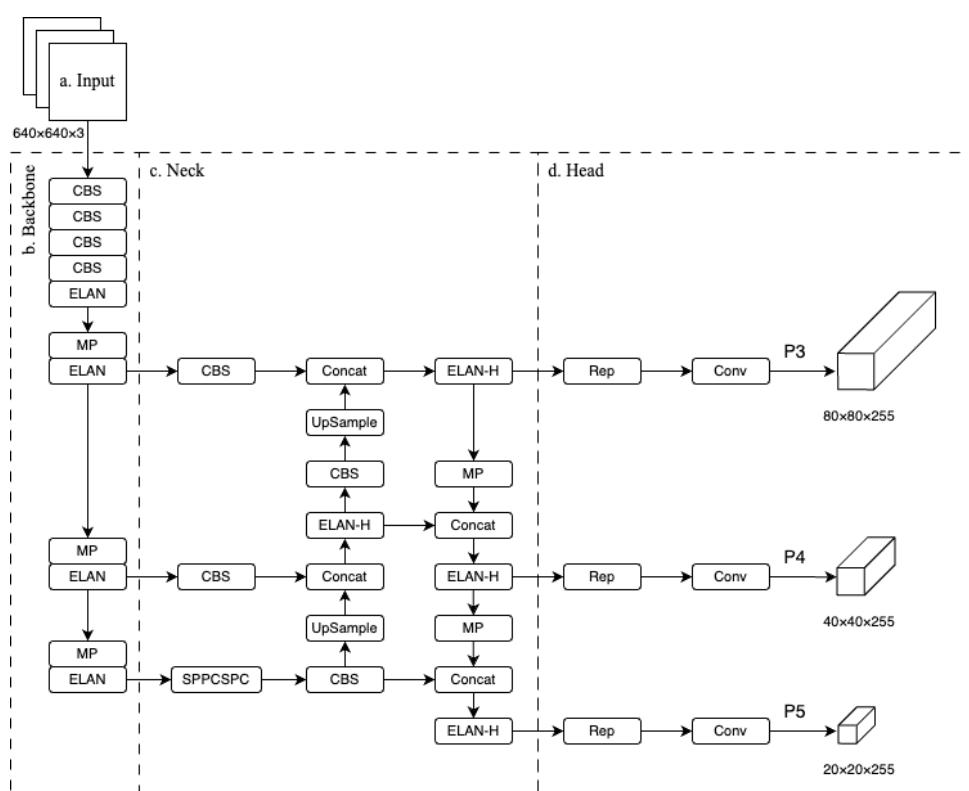


圖 3.12: YOLOv7 架構圖

[36]



3.4.3.2 YOLOv7-UAV

由於無人機本身拍攝時就要跟目標物有一段距離，且本研究所要偵測的是裂縫，裂縫又相較於一般偵測的物件，例如車輛或是人物，要來的細小許多，調整參數可能無法提升其模型分數，所以考慮到可能面臨的情形，參考了多篇文獻，發現 Siling Feng et al. [22] 跟 Yingkun Wei et al. [23] 都是以剔除 P5 的特徵提取層，避免過度下採樣導致小目標訊息流失，甚至在第二篇的文獻中有增加 P2 的特徵提取層，以建立 YOLOv7-UAV 的改良架構(圖3.13)，以更全面的提取裂縫的目標訊息，以達到更好的辨識結果。

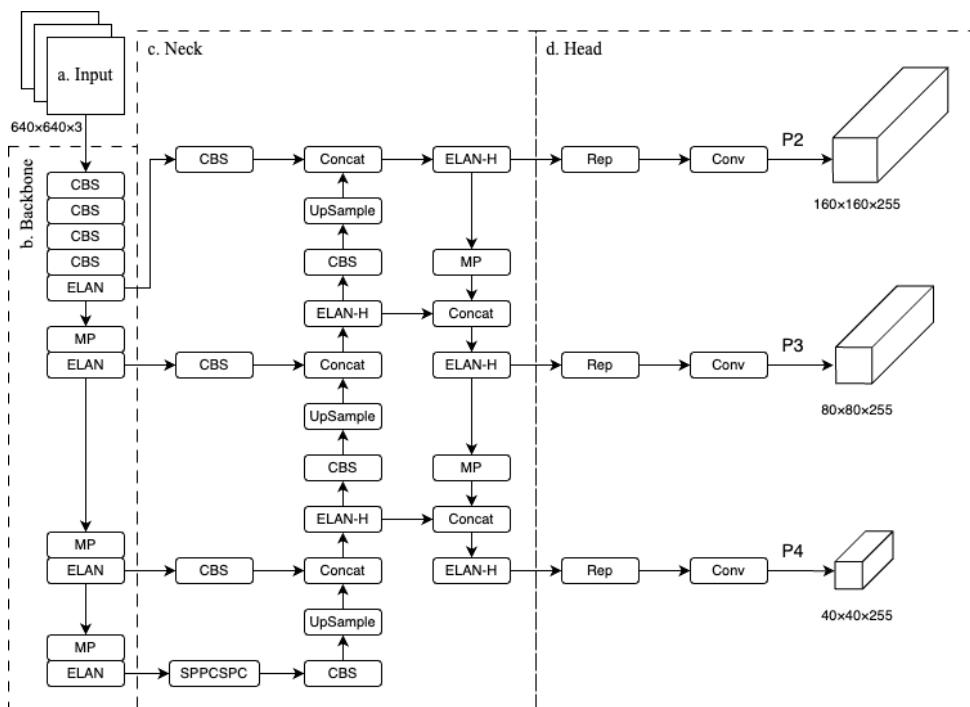


圖 3.13: YOLOv7-UAV 架構圖

[23]



3.4.4 調整模型參數

模型的適配性除了跟資料集有關係，另外架構也會有很大的影響，模型的骨幹網路決定了其對輸入圖像的特徵提取能力。如果骨幹網路設計能夠有效地提取出圖像中的關鍵特徵，那麼模型就有較好的性能表現，所以尋找適合目標偵測任務的架構是很重要的，本研究將先利用手動調整先粗略調整 YOLOv7 以及 YOLOv7-UAV 模型架構的超參數，找到最好的超參數組合。再利用 YOLOv7 的超參數進化方法 (Hyperparameter Evolution) ，也就是採用基因演算法 (Genetic Algorithm, GA) [37] 來進行迭代。

基因演算法 (Genetic Algorithm, GA) 是由 J.H. Holland 所提出，其方法是受遺傳學中自然進化的啟發，模擬了自然界中的適者生存達爾文理論。GA 的基本元素包括染色體表示、適應度選擇和受生物啟發的運算子。Holland 還引入了一個新元素，即反轉 (Inversion)，通常在實現中使用。通常，染色體採用二進制字符串格式。在染色體中，每個位點 (染色體上的特定位置) 有兩種可能的等位基因 (基因的變體形式) -0 和 1。染色體被視為解空間中的點。這些染色體通過遺傳運算子進行處理，通過迭代地替換其群體。適應度函數用於為群體中的所有染色體分配一個值，其流程以及圖示 (圖3.14) 為下：

- 產生初始群集 (Initial Population)

隨機生成一個包含 n 個染色體的初始族群。

- 計算適應度 (Fitness)

計算每個染色體的適應度值。

- 選擇 (Selection)

根據染色體的適應度值，選擇出兩個染色體作為父母染色體。



- 交叉 (Crossover)

對選擇出的父母染色體進行交叉操作，生成子代染色體。

- 突變 (Mutation)

對生成的子代染色體進行突變操作，引入一定程度的隨機變化。

- 新族群形成

將經過選擇、交叉和突變操作後得到的子代染色體形成新的族群。

- 終止條件檢查

檢查是否滿足終止條件，如達到最大迭代次數或適應度達到某個閾值，即停止算法輸出最佳解。

- 繼續執行

如果未滿足終止條件，則繼續進行適應度計算和遺傳操作，以達到滿足條件為止，才得以結束算法並輸出最優解。

將其應用在參數的優化中的話，就是將染色體更改為各種超參數，透過該方法找到超參數組合。本研究將測試兩種架構，透過手動調參以及超參數進化 (Hyperparameter Evolution)，以期找出最適合裂縫資料集的架構，並於第四章節進行討論。

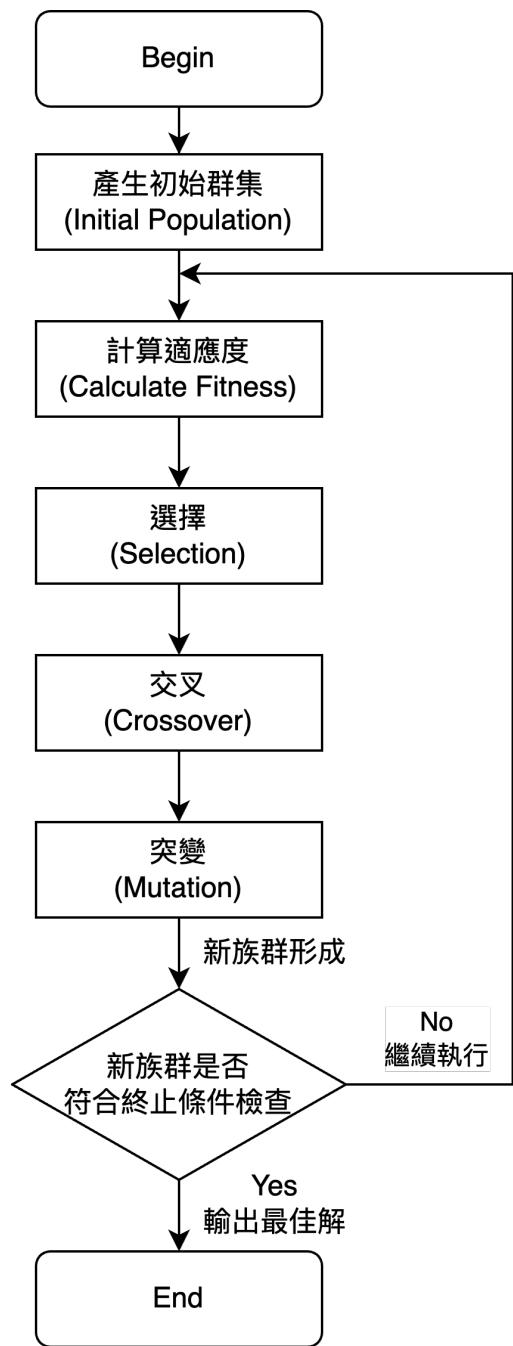


圖 3.14: 基因演算法流程圖

[38]



3.5 無人機即時影像辨識

為了在 ROS 系統中建立即時的 YOLOv7 物件辨識系統，我們需要取得無人機的影像，並對先前得取的無人機影像進行影像資訊的轉換，讓其可以在 ROS 系統中被使用。在 ROS 系統中架設節點訂閱並處理影像，然後執行模型，將辨識結果儲存在開發板中，以便進行後續的警示。

在本研究中，將採用兩種不同的無人機載具，分別是 DJI Tello 和 DJI Inspire 2。使用輕量級無人機 DJI Tello EDU 的目的，在於在初期階段確立整個系統的架構和流程，由於 DJI Tello EDU 具有輕巧便攜、易於操控且有多方開發套件的特點，因此非常適合用於快速驗證和測試裂縫影像擷取的流程和方法，透過 DJI Tello EDU 的使用，可以迅速獲取裂縫影像，並將其傳輸至後續的影像處理系統中進行進一步的處理和分析。接著，在確立了整個系統的架構和流程後，將架構轉移至量級較大的 DJI Inspire 2，DJI Inspire 2 具有較大的有效載荷和更長的飛行時間，同時也具有更高的影像品質，因此非常適合用於進一步的裂縫影像擷取和細節捕捉。通過將流程轉移至 DJI Inspire 2，可以提高裂縫影像的品質和準確性以及在外飛行的穩定性，進一步優化整個系統的性能。



3.5.1 無人機影像取得

為了有別於以往研究，於現地以無人機將影像拍攝後再進行後續的處理；本研究將無人機的影像即時傳輸至後續的系統中進行影像辨識，為了達到到該目的，將會使用一些硬體設備或 ROS 上的套件。這些設備或套件將允許將無人機攝取的影像即時地傳輸至開發板上的系統中，以供後續的影像辨識和分析使用。透過這樣的設置，可以實現對無人機影像的即時處理和辨識，從而為裂縫辨識系統的建立和應用提供更好的支持。

3.5.1.1 DJI Tello EDU

在使用 DJI Tello EDU 進行影像蒐集的部分，由於無人機本身支援多種程式語言，因此可以輕易地連接到 ROS 系統中，並利用 ROS 系統中的各項套件進行操作。例如 DJI Tello EDU 的飛行操作可以透過 ROS 中的 xone [39] 套件實現，這個套件可以連接 Xbox One 的搖桿，並透過開源的 flock [40] 套件將無人機連接到 ROS 系統中。通過由搖桿傳送飛行指令，再透過 ROS 系統將指令傳送給無人機，就可以控制無人機進行飛行，並利用 flock 套件取得其影像資訊，發布影像資訊於系統中，方便往後將影像資訊在開發板中供後續處理和分析使用。

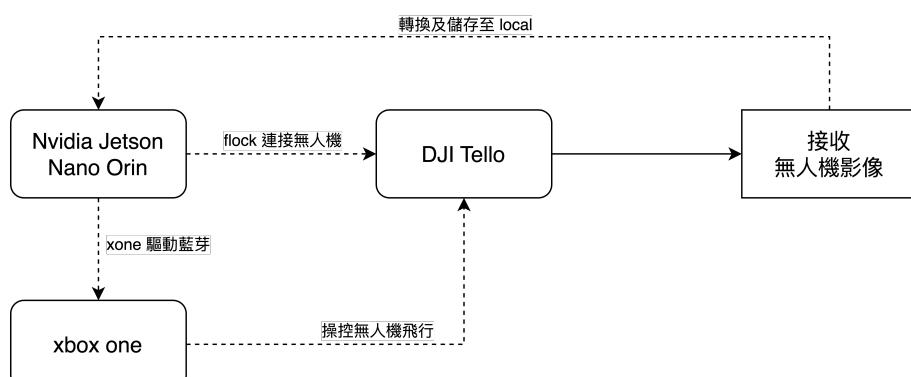


圖 3.15: DJI Tello EDU 影像蒐集流程圖

3.5.1.2 DJI Inspire 2



由於 DJI Inspire 2 本身硬體並不支援 ROS 系統，因此在進行影像蒐集時，必須利用遙控器本身的影像導出功能。DJI Inspire 2 的遙控器具有影像輸出端口，無人機可以在飛行時將即時影像，透過 2.4GHz 或是 5.8GHz 的無線電頻段傳輸至搖桿上，並透過 HDMI 將影像投影至外部設備。在這種情況下，可以使用影像擷取盒，將無人機的影像訊號轉換成數位訊號，並將其傳輸至開發板或其他相關系統中。一旦影像訊號被傳輸至開發板中，可以利用 ROS 系統中的相應套件，將影像接收並發布在系統中。這可以通過設置 ROS 節點或利用 ROS 提供的影像話題進行實現。透過訂閱影像話題，ROS 系統可以接收並處理來自無人機的即時影像，為後續的處理和分析提供必要的影像資料。

這個應變措施不僅僅能夠解決無人機在飛行時影像傳輸的問題，還可以實現即時地面端接收到影像的功能。通過將 DJI Inspire 2 的影像訊號轉換成電腦可以接收的訊號，我們可以在地面端即時接收到無人機的影像資訊。這樣一來，我們就能夠利用開發板，進行即時的影像辨識和後續的警示工作。

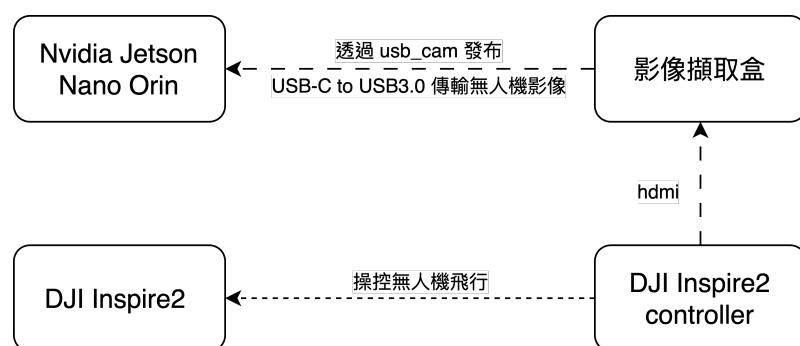


圖 3.16: DJI Inspire 2 影像蒐集流程圖



3.5.2 影像處理

拿到無人機影像之後，需經過影像格式的轉換，才能將影像提供給 ROS 系統做使用，由於硬體的限制不同，處理的方式跟應變的措施也會有所差異，以下將說明 DJI Tello EDU 及 DJI Inspire 2 的影像處理方法。

3.5.2.1 DJI Tello EDU

為了要將無人機的影像轉換為 ROS 系統可以讀取的形式時，需要進行一系列的步驟。首先需要設置 ROS 節點來訂閱 DJI Tello EDU 無人機的即時影像流，首先使用 TelloPy 函式庫去取得無人機的即時影像流，之後利用 Python 中的 PyAV 套件，PyAV 函式庫將 H.264 編碼格式的影像流進行解碼，轉成 YUV 像素格式。再透過 PIL 函式庫 (Python Imaging Library) 轉換為 RGB8 的像素格式。再利用 NumPy 轉換成 array 格式，這樣就能透過 OpenCV，調整其色彩通道為 BGR 且為 OpenCV Mat 的影像格式，最後利用 ROS 提供的 CVBridge 庫，將 OpenCV Mat 物件轉換為 ROS 影像資料格式，利用節點發布至系統上以供後續使用。

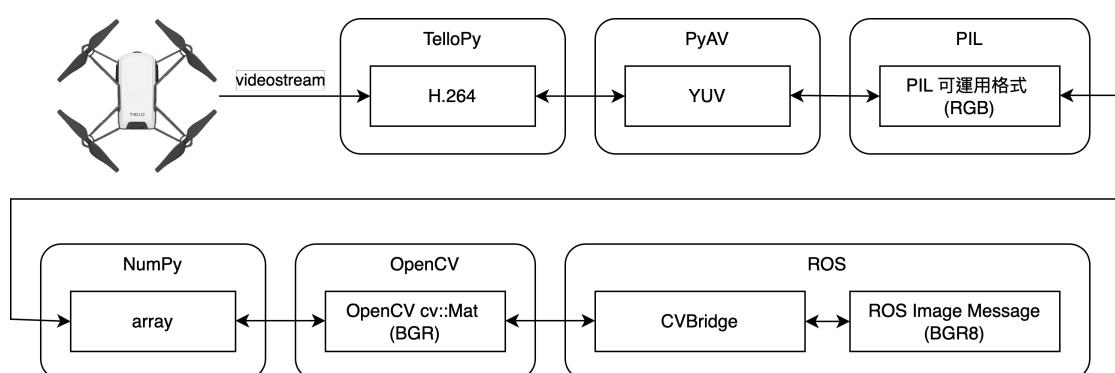


圖 3.17: DJI Tello EDU 影像格式轉換流程圖



3.5.2.2 DJI Inspire 2

由於無法直接使用 ROS 套件來獲取 DJI Inspire 2 的影像，所以採取一些應變措施以實現對影像的取得和處理。DJI Inspire 2 的遙控器具有 HDMI 輸出影像功能，這使得我們可以將無人機相機的數位影像訊號，透過其本身的技術轉為類比訊號，經由 2.4GHz 或 5.8GHz 這兩個不同的頻段傳輸至 DJI Inspire 2 的遙控器中，最後可以通過 HDMI 端口，將類比訊號轉換成數位訊號的影像，輸出到外部設備上。為了將這個影像訊號轉換成電腦可以接收的訊號，我們可以使用影像擷取盒。這個擷取盒能夠將 HDMI 輸出的影像訊號轉換為電腦可以接收的視訊訊號，透過 USB 接口將影像資訊傳輸至電腦，並會以 YUYV 像素格式傳輸至開發板中。一旦將影像轉換成電腦可以接收的訊號，接下來我們可以使用 ROS 系統從 USB 攝像頭或其他視訊設備中獲取視訊流，再將影像資訊透過解碼，並以函式轉換成 ROS 系統中支援且影像通道為 RGB 的影像訊息，並發佈到 ROS 系統上，供後續的影像處理和分析使用，在這邊我們還會將無人機原始的影像，透過 ROS 進行裁切，將本身無人機影像會干擾的部分裁剪掉，並重新發佈。

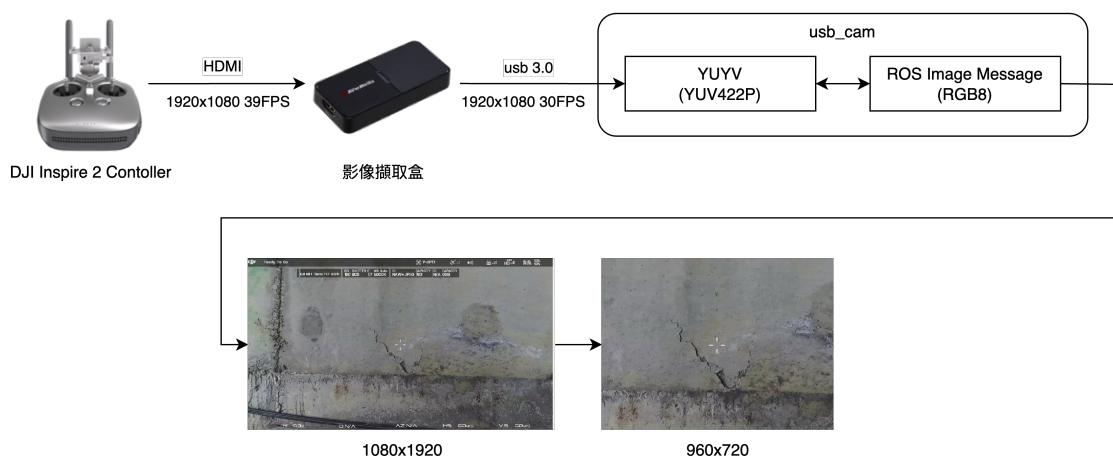


圖 3.18: DJI Inspire 2 影像格式轉換流程圖



3.5.3 YOLOv7 模型

為了需要在 ROS 系統上進行即時的 YOLOv7 物件辨識，就需要透過套件包裝一系列的節點才能施行，本研究使用以開源的套件 yolov7_ros [41] 進行修改來達成，修改指定的影像主題，並將原先已訓練好，且將表現最佳的權重匯入工作區中，讓辨識時可以取用該權重，本研究是將經過格式轉換後的影像，透過節點的訂閱進行即時影像的辨識，最後透過含有辨識框的影像儲存至開發板中。

3.6 裂縫警示

本研究的警示功能是以 Tkinter 庫 [42] 呈現。Tkinter 提供了豐富的 GUI 元素，包括按鈕、標籤、文本框、列表框、滾動條等常用的組件，以及各種容器組件如框架、窗口等。透過搭配相關的參數設定，可以實現警示的功能。在本研究中，若裂縫的信心水準超過一定閾值，系統將彈出視窗供使用者確認，另外在這邊特別說明，信心水準就是為 IOU，其範圍介於 0 到 1 之間，計算方式於 3.4.1.1，在彈出視窗的同時，模型還是會持續地進行影像的辨識，以達到警示使用者的效果（圖 3.19）。

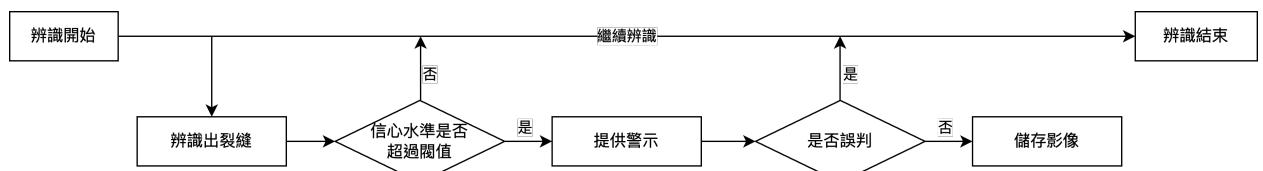


圖 3.19: 裂縫警示流程圖



3.7 流程圖

在這一部分，繪製硬體跟整體流程的硬體流程圖，並進行粗略的說明，以使整體架構更加清晰。

3.7.1 DJI Tello EDU

本研究會預先訓練好 YOLOv7 模型，並將表現最好的權重導出，匯入於在開發板上的影像辨識套件，由於 DJI Tello EDU 支援 ROS 系統的特性，所以可以利用 ROS 系統進行 XBox One 搖桿的串連，並利用搖桿進行飛行的操控，將無人機的影像傳輸回開發板，並進行影像辨識及提供對應的警示。

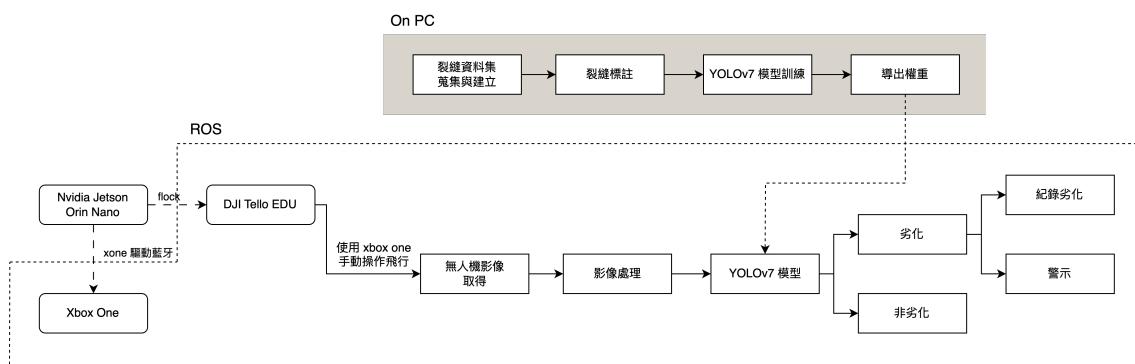


圖 3.20: DJI Tello EDU 流程圖



3.7.2 DJI Inspire 2

由於 DJI Inspire 2 不支援 ROS 系統，所以透過無人機本身的無線電圖傳功能，搭配影像擷取盒，將影像進行訊號轉換傳輸回開發板中，並透過先前訓練好的權重以其架設好的 ROS 系統套件，進行自動化的影像處理，隨後影像辨識以及提供對應的警示。

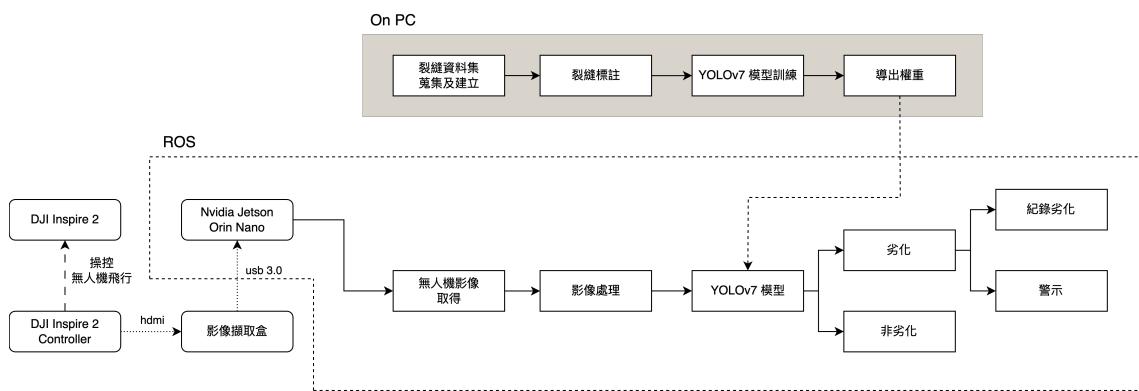


圖 3.21: DJI Inspire 2 流程圖



3.8 節點圖

透過 ROS 的節點圖，更能得知各項主題在 ROS 系統中的傳遞方式，以更明確的方式釐清系統之間節點跟節點間的訊息傳遞，以及跟硬體的連接方式。

3.8.1 DJI Tello EDU

在 ROS 系統中，Xbox One 搖桿透過 joystick 以及 joy 節點連結至控制 DJI Tello EDU 的 flock 節點中，並利用 /flock_base_node，接收來自多個飛行指令的輸入數據，最後飛行指令的輸出連接到 /flock_driver_node 節點，該節點負責實際驅動飛行指令來執行系統的運動。另外也透過該節點，提供了原始影像數據 /image_raw，之後被傳輸到 /tello_yolo_image_processor 節點處理，進行影像分析和物件偵測（圖3.22）。

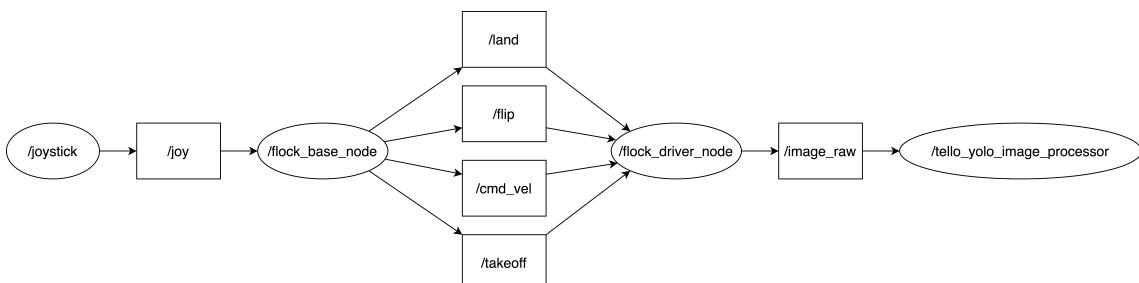


圖 3.22: DJI Tello EDU 節點圖



3.8.2 DJI Inspire 2

DJI Inspire 2 的影像傳輸是透過影像擷取盒傳輸至開發板中，並經由 `/usb_cam` 的節點發布至 ROS 系統當中，並輸出相機的數據資訊以及提供相機捕捉的原始影像數據的 `/usb_cam/image_raw` 主題，同時也會使用 `/usb_cam/image_proc`，將原始影像數據從 `/usb_cam/image_raw` 傳送到 `/camera_crop` 節點，該節點輸出一個被裁剪過的影像，發布在 `/camera_crop/image_raw` 主題。最後這個裁剪過的影像數據隨後被傳送到 `/yolo_image_processor` 節點，進行後續的影像辨識（圖3.23）。

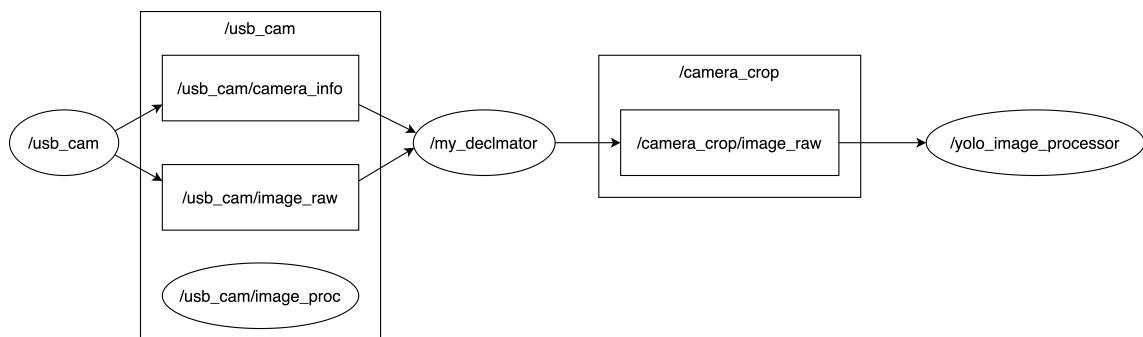


圖 3.23: DJI Inspire 2 節點圖

3.9 小結

本研究利用 ROS 系統或是以硬體的手法，初步架設無人機的即時影像取得，並透過於開發板的 ROS 系統中架設模型，以達成即時的無人機影像辨識，並對相對應的辨識結果提供警示，後續會在第四章進行成果的說明。



第四章 研究結果

本研究根據上述研究方法，將成果以兩個部分呈現，裂縫辨識模型結果以及無人機的影像辨識，於裂縫辨識模型的部分，會詳細比較兩種模型架構，以及超參數的設置，還有其訓練出來的結果，並作模型分數以及應用在 bag 檔上的成果比較。無人機影像辨識的部分會呈現於現地辨識的結果，以及現地的相關資訊。

4.1 裂縫辨識模型

4.1.1 設備規格

本研究所使用的影像辨識的模型，是以桌機的方式進行訓練，最後再將效果呈現最佳的模型權重移至 ROS 系統中的 YOLOv7 套件中。以下簡略說明本研究訓練模型的軟硬體規格 (表4.1)。

表 4.1: 桌機軟硬體規格

項目	規格
作業系統	Ubuntu 20.04
顯示卡	RTX 4080 16GB
處理器 CPU	AMD Ryzen 5-7600 6-Core
CUDA	12.1
PyTorch	2.1.1



4.1.2 模型架構

本研究利用相同資料集針對不同架構，將模型訓練都設定相同的超參數，來進行模型架構的比較，正如 3.4.2 所說明的，資料集的影像先初步以順時鐘 90° 以及逆時鐘 90° 的旋轉 (90° Rotate)，另外也進行了 -15° 到 $+15^\circ$ 範圍內的隨機旋轉 (Rotation)。另外利用資料增強的方式，將原始資料集 1186 張影像擴增至 3559 張影像。

本章節為了要比較 YOLOv7 及 YOLOv7-uav 的架構，會先分別將兩個架構搭配 YOLOv7 的預訓練模型，利用手動調整將 Epoch、Batch Size、Loss Function、Learning Rate、調整凍結層，以初步調整至該模型的最佳狀態，再利用 YOLOv7 本身的 evolve 方法，其方法是使用基因演算法 (GA, Genetic Algorithm) 調整其他細微的超參數至最佳的狀態，以確保模型都有經過良好的訓練，再進行模型架構的比較，以下會詳細介紹模型的設定參數以及成果。



4.1.3 YOLOv7 初步參數微調

4.1.3.1 Batch Size

由於 Batch Size 的設定會影響到資料集輸入，所以在進行其他參數調整時，選擇調整 Batch Size，分別測試 4、8、16、20，其餘的參數設定使用的是 YOLOv7 預設之參數組合，但為了使模型有足夠的訓練，先以訓練 100 次為基礎設定，詳細的參數表在下方 (表4.2)。

從圖4.1及圖4.2可以得知，訓練期間當 Batch Size 為 20 時，擁有最小的 Loss 值跟最高的 mAP 曲線，但從表4.2 可知，在 Batch Size 為 16 時，雖然其訓練集以及測試集的 Precision、Recall、mAP 都來到分數最高點，但訓練集和測試集的模型分數有不小的差距。理論上來說，較大的 Batch size 可能會加快訓練速度，因為它允許更多的樣本在一次反向傳播過程中被處理。但是這也可能會導致性能下降或過度擬合的問題，再由於本研究測試 Batch Size 為 32 的時候，因為內存不夠而導致訓練無法進行，所以於該架構資料導入的部分，本研究決定以 Batch Size 為 16，來做後續的訓練。

表 4.2: YOLOv7 架構模型 Batch 參數表

Epoch	Batch	LR	Opt.	Loss	Set	Precision	Recall	mAP@50
100	4	0.005	SGD	GIoU	train	0.837	0.674	0.609
					test	0.835	0.629	0.575
100	8	0.005	SGD	GIoU	train	0.793	0.336	0.283
					test	0.732	0.199	0.151
100	16	0.005	SGD	GIoU	train	0.954	0.953	0.96
					test	0.875	0.871	0.862
100	20	0.005	SGD	GIoU	train	0.965	0.96	0.969
					test	0.87	0.868	0.856

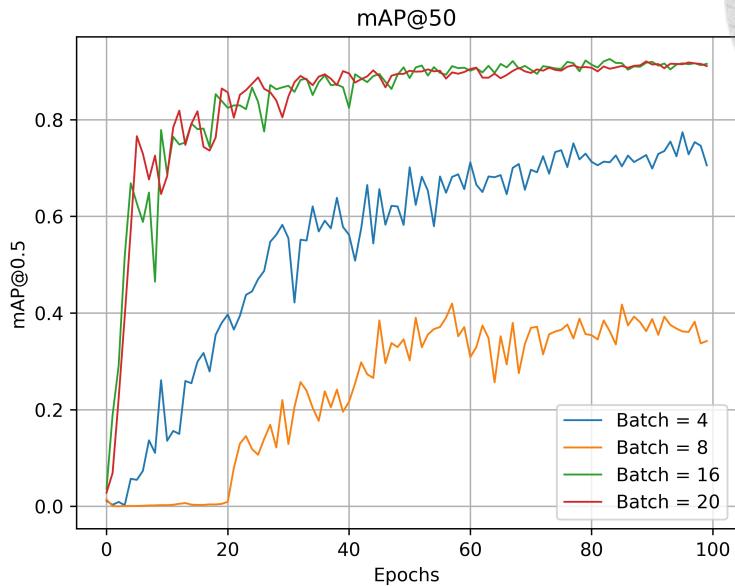
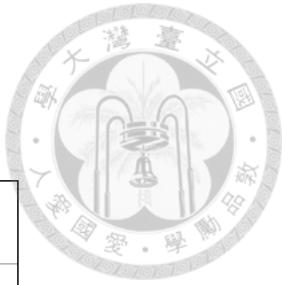


圖 4.1: YOLOv7 Batch 參數調整 mAP 圖

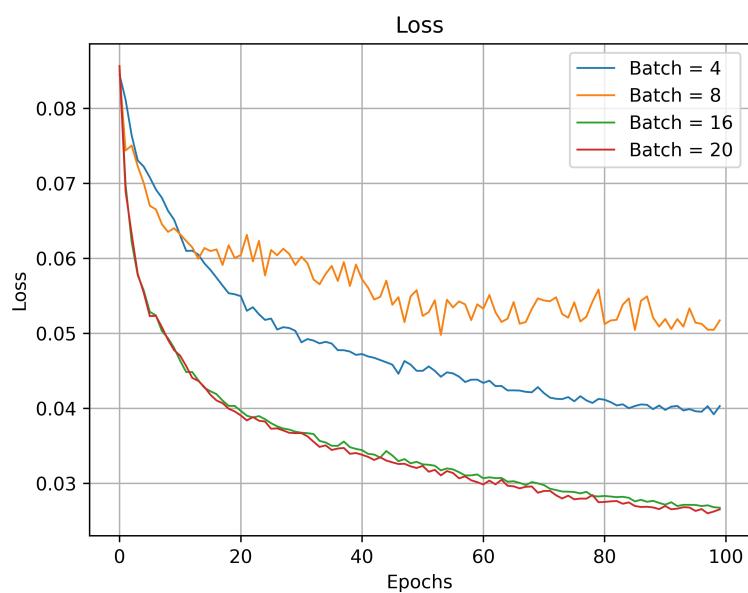


圖 4.2: YOLOv7 Batch 參數調整 Loss 圖



4.1.3.2 Loss Function

本研究測試損失函數對應模型架構的適配度，則使用目前主流的損失函數進行分次的訓練，分別是 GIoU、CIoU、SIoU，其詳細的損失函數說明已在第三章節中說明。以 Batch Size 為 16，以及固定其他超參數，僅變動不同損失函數進行測試（表4.3）。

從圖4.3、圖4.4可以得知，其實這三個損失函數在訓練過程中沒有太大的差異，但從表4.3可以知道，CIoU 損失函數在訓練過程中的表現，相對於 GIoU 和 SIoU 損失函數表現更好，但在測試集的表現上 SIoU 的精確度稍顯突出，但在測試集 mAP 的分數是 CIoU 得到最好的成績，所以綜合評估，決定以 CIoU 做為後續超參數調整的優先選擇，這將有助於提高模型在實際應用中的準確性和泛化能力。

表 4.3: YOLOv7 架構模型 Loss 參數表

Epoch	Batch	LR	Opt.	Loss	Set	Precision	Recall	mAP@50
100	16	0.005	SGD	GIoU	train	0.954	0.953	0.96
					test	0.875	0.871	0.862
100	16	0.005	SGD	CIoU	train	0.963	0.957	0.965
					test	0.907	0.848	0.863
100	16	0.005	SGD	SIoU	train	0.961	0.961	0.965
					test	0.912	0.846	0.853

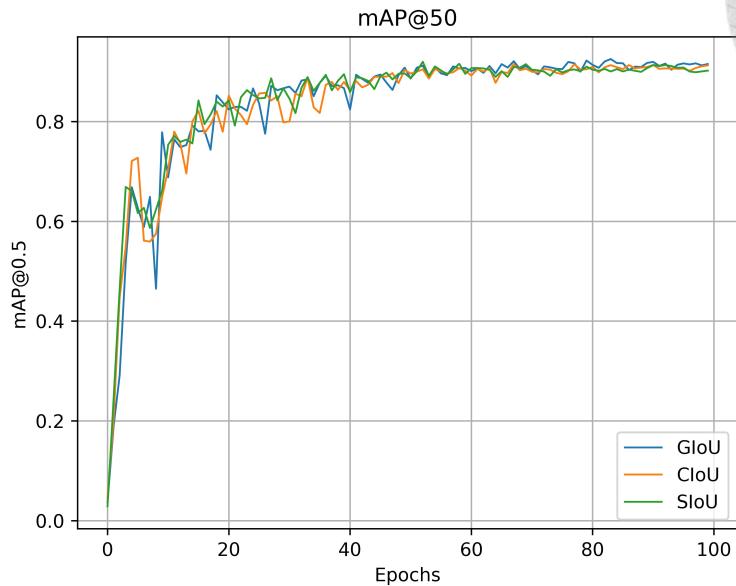


圖 4.3: YOLOv7 Loss 參數調整 mAP 圖

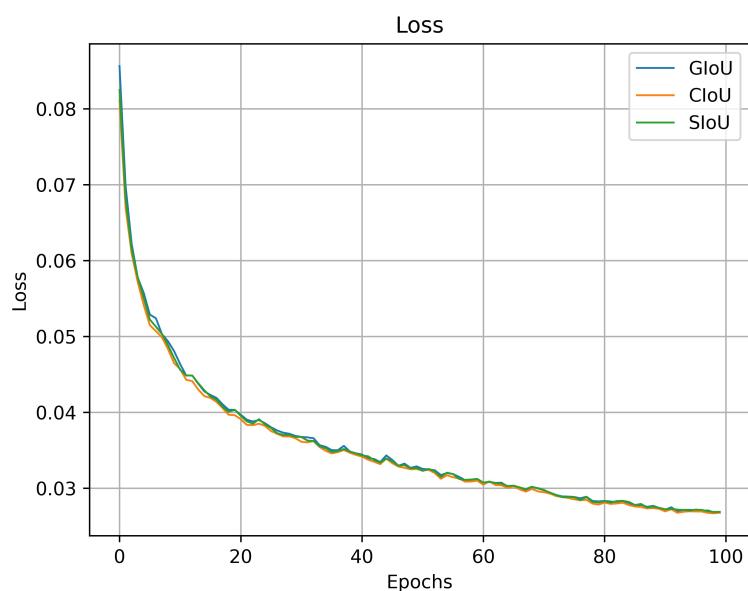


圖 4.4: YOLOv7 Loss 參數調整 Loss 圖



4.1.3.3 Learning Rate

在調整學習率參數時，本研究嘗試了 0.001、0.002、0.003、0.004、0.005 的取值（表4.4）。除了將批量大小設定為 16、損失函數為 CIoU 外，其他參數均沿用了 YOLOv7 的預設設置。從圖4.5及圖4.6可以得知，這五種學習率在訓練過程中，mAP 跟 loss 沒有太大的差距。

而從表4.4可知，這五個參數在測試集和驗證集的表現差異都相當接近。在學習率為 0.004 時，訓練集取得了最高的 Recall (0.962) 和 mAP (0.967) 指標，但其訓練集的分數跟測試集的分數相較於其他的不同的參數組合，差距要來得大。在學習率為 0.002 時，其訓練集的 mAP 也是最高分 mAP (0.967)，且測試集的 Recall 跟 mAP 都拿到最高的分數，在考慮測試集跟訓練集分數的狀況下，其相差值也是其他組合中最小的，為了兼顧訓練集和測試集的表現，選擇學習率為 0.002，繼續進行參數的調整。

表 4.4: YOLOv7 架構模型 Learning Rate 參數表

Epoch	Batch	LR	Opt.	Loss	Set	Precision	Recall	mAP@50
100	16	0.001	SGD	CIoU	train	0.944	0.957	0.96
					test	0.903	0.868	0.863
100	16	0.002	SGD	CIoU	train	0.963	0.958	0.967
					test	0.907	0.876	0.876
100	16	0.003	SGD	CIoU	train	0.958	0.955	0.961
					test	0.917	0.834	0.853
100	16	0.004	SGD	CIoU	train	0.958	0.962	0.967
					test	0.885	0.865	0.847
100	16	0.005	SGD	CIoU	train	0.963	0.957	0.965
					test	0.907	0.848	0.863

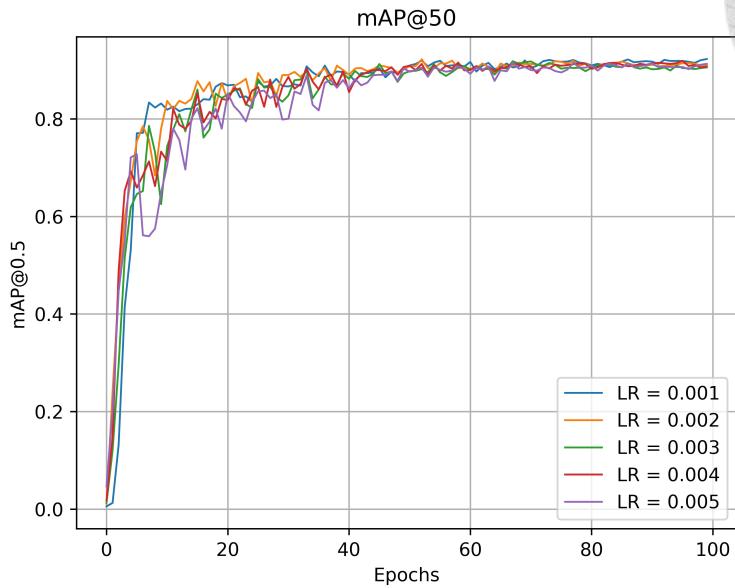
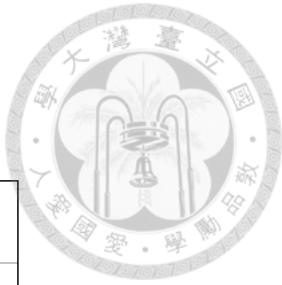


圖 4.5: YOLOv7 LR 參數調整 mAP 圖

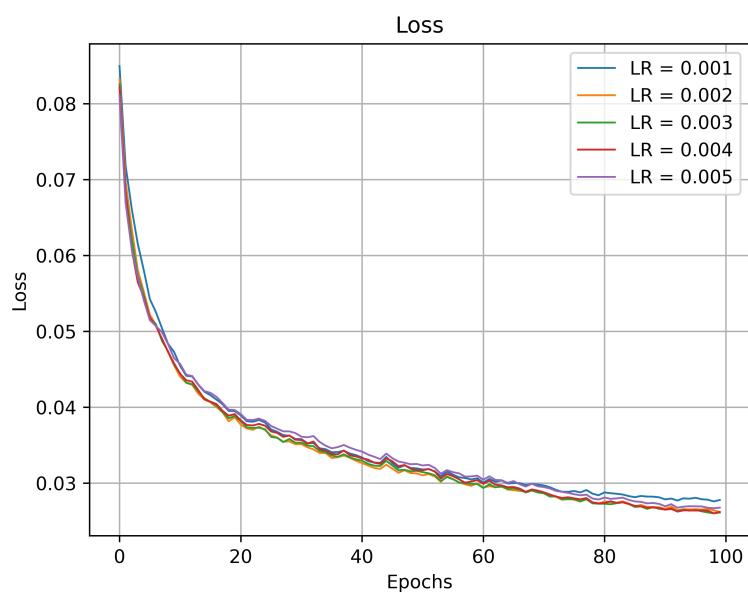


圖 4.6: YOLOv7 LR 參數調整 Loss 圖



4.1.3.4 Epoch、凍結層

在這部分以調整 Epoch 以及凍結層作為測試，Epoch 以 100、120、140，三個參數進行測試，凍結層是以調整 Backbone 的層數，以常用的凍結層數 0、2、4、6，進行交叉組合測試，詳細數據表以及圖像呈現於圖4.7、圖4.8、表4.5。

從圖4.7、圖4.8，可以看出在訓練時，其實 mAP 跟 Loss 都沒有太大的差異，但由表4.5中可以看到，在考慮測試集的 mAP 分數時，前三高的參數組合為：在沒有任何凍結層，且 Epoch 為 100 的時候 (0.876)、凍結 0、2、4、6 層，Epoch 為 120 時 (0.875)、凍結 0、2 層且 Epoch 為 140 時 (0.872)，但若將 Precison、Recall、mAP 的差距考慮進去，凍結 0、2 層且 Epoch 為 140 的模型，其訓練集跟測試集沒有誤差太大的情形，考慮到後續的可能的過擬合情形，本研究後續會利用該組參數進行後面的超參數進化。

表 4.5: YOLOv7 架構模型 Epoch、凍結層參數表

Epoch	Batch	LR	freeze	Loss	Set	Precision	Recall	mAP@50
100	16	0.002	none	CIoU	train	0.963	0.958	0.967
					test	0.907	0.876	0.876
120	16	0.002	none	CIoU	train	0.979	0.97	0.978
					test	0.896	0.868	0.863
140	16	0.002	none	CIoU	train	0.976	0.967	0.972
					test	0.893	0.871	0.859
100	16	0.002	0, 2	CIoU	train	0.968	0.962	0.972
					test	0.913	0.86	0.866
120	16	0.002	0, 2	CIoU	train	0.981	0.966	0.978
					test	0.897	0.86	0.853
140	16	0.002	0, 2	CIoU	train	0.976	0.973	0.98
					test	0.913	0.885	0.872
100	16	0.002	0, 2, 4	CIoU	train	0.964	0.967	0.97
					test	0.885	0.862	0.848
120	16	0.002	0, 2, 4	CIoU	train	0.974	0.965	0.972
					test	0.89	0.885	0.865
140	16	0.002	0, 2, 4	CIoU	train	0.967	0.97	0.975
					test	0.865	0.879	0.863
100	16	0.002	0, 2, 4, 6	CIoU	train	0.968	0.963	0.972
					test	0.911	0.837	0.863
120	16	0.002	0, 2, 4, 6	CIoU	train	0.976	0.968	0.976
					test	0.881	0.89	0.875
140	16	0.002	0, 2, 4, 6	CIoU	train	0.983	0.962	0.976
					test	0.913	0.851	0.852

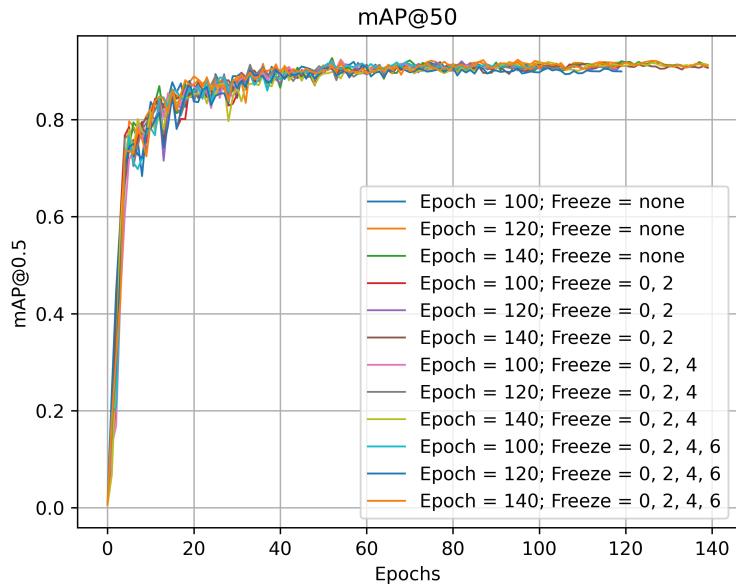


圖 4.7: YOLOv7 Epoch、凍結層參數調整 mAP 圖

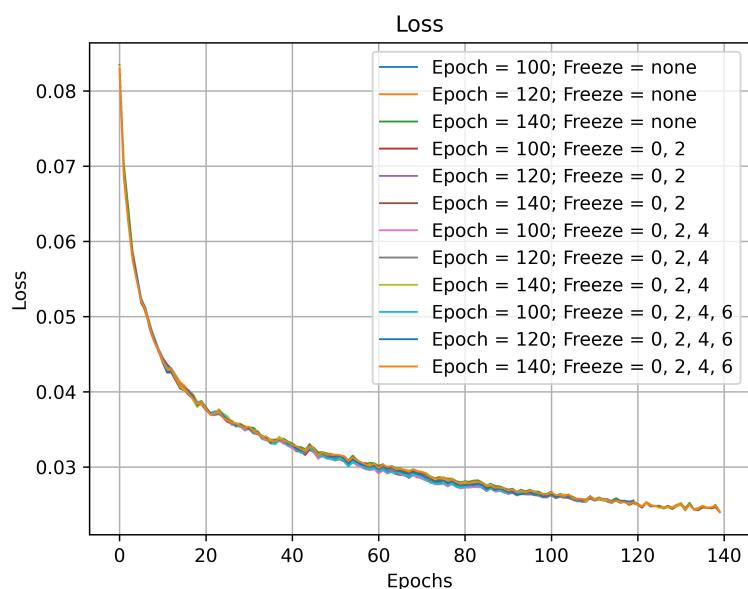


圖 4.8: YOLOv7 Epoch、凍結層參數調整 Loss 圖



4.1.3.5 Evolution

由以上的初步調整超參數可知，YOLOv7 架構的最佳模型分數為在測試集上取得 0.872 的 mAP (表4.5)，再後續經過 50 代的超參數進化，將模型的表現提升至 0.877 mAP (表4.6)。

表 4.6: YOLOv7 超參數進化模型分數表

	Epoch	Batch	LR	freeze	Loss	Set	Precision	Recall	mAP@50
Pre-evolve	140	16	0.002	0, 2	CIoU	train	0.976	0.973	0.98
						test	0.913	0.885	0.872
evolved	140	16	0.002	0, 2	CIoU	train	0.978	0.976	0.983
						test	0.892	0.882	0.877



4.1.4 YOLOv7-UAV 初步參數微調

4.1.4.1 Batch Size

由於 Batch Size 的設定會影響到資料集輸入，所以在進行其他參數調整時，選擇調整 Batch Size，其餘的參數設定使用的是 YOLOv7 預設之參數組合，但為了使模型有足夠的訓練，先以訓練 100 次為基礎設定 (表4.7)。

由圖4.9及圖4.10可以得知，當 Batch Size 為 16 時，具有最低的 Loss 值，但在 mAP 的部分，各個參數的表現沒特別突出，另外，由表 4.7 可以得知，當 Batch Size 為 16 時，其測試集以及訓練集都有最高分，所以選擇以 Batch Size 為 16 進行後續的訓練。

表 4.7: YOLOv7-UAV 架構模型 Batch 參數表

Epoch	Batch	LR	Opt.	Loss	Set	Precision	Recall	mAP@50
100	4	0.005	SGD	GIoU	train	0.941	0.597	0.585
					test	0.881	0.5	0.475
100	8	0.005	SGD	GIoU	train	0.936	0.631	0.616
					test	0.904	0.452	0.428
100	16	0.005	SGD	GIoU	train	0.938	0.709	0.697
					test	0.876	0.596	0.568
100	20	0.005	SGD	GIoU	train	0.953	0.638	0.626
					test	0.87	0.506	0.483

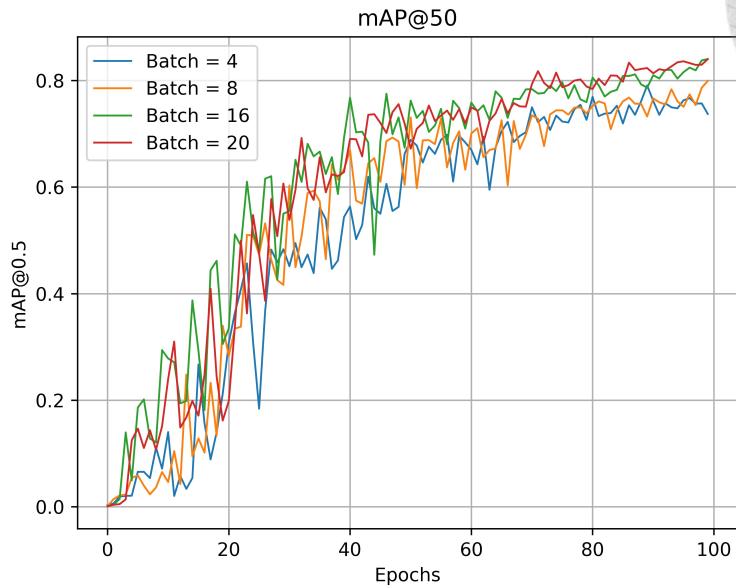


圖 4.9: YOLOv7-UAV Batch 參數調整 mAP 圖

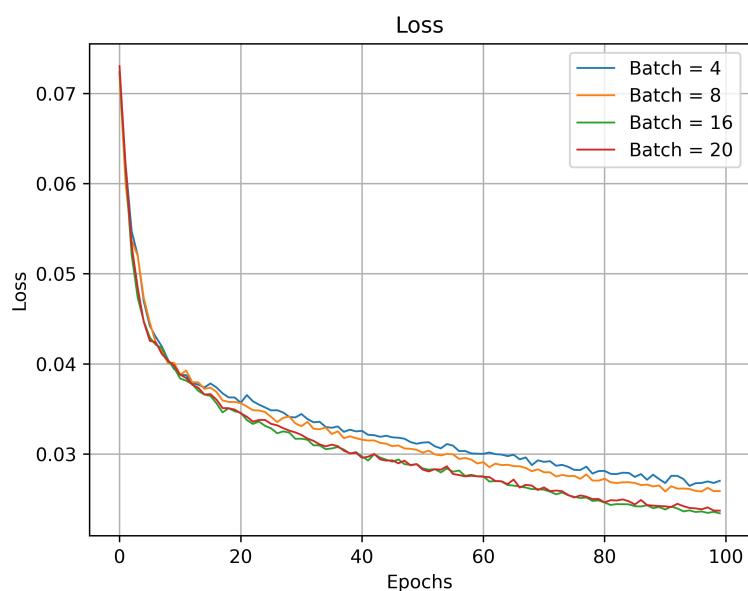


圖 4.10: YOLOv7-UAV Batch 參數調整 Loss 圖



4.1.4.2 Loss Function

本研究測試損失函數對應模型架構的適配度，則使用目前主流的損失函數進行分次的訓練，分別是 GIoU、CIoU、SIoU，其詳細的損失函數差異已在第三章節說明。以 Batch Size 為 16，以及固定其他超參數，僅變動不同損失函數進行測試（表4.8）。

從圖4.13及圖4.14可以得知，這三種損失函數沒有太大的差異，但從表4.8可以得知，SIoU 損失函數相對於 GIoU 和 CIoU 損失函數，不管是在測試集還是訓練集的分數上都表現更好，且測試集跟訓練集的分數是在這三種損失函數中相差最小的，代表在該損失函數下，較不會有過擬合的情況發生，基於以上理由，所以選擇以 SIoU 做後續的參數調整。

表 4.8: YOLOv7-UAV 架構模型 Loss 參數表

Epoch	Batch	LR	Opt.	Loss	Set	Precision	Recall	mAP@50
100	16	0.005	SGD	GIoU	train	0.938	0.709	0.697
					test	0.876	0.596	0.568
100	16	0.005	SGD	CIoU	train	0.938	0.702	0.689
					test	0.852	0.551	0.519
100	16	0.005	SGD	SIoU	train	0.942	0.718	0.708
					test	0.891	0.62	0.599

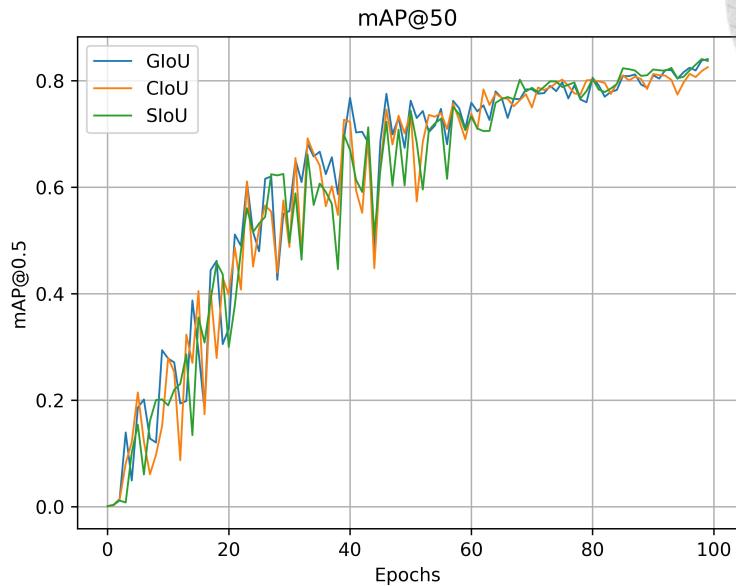


圖 4.11: YOLOv7-UAV Loss 參數調整 mAP 圖

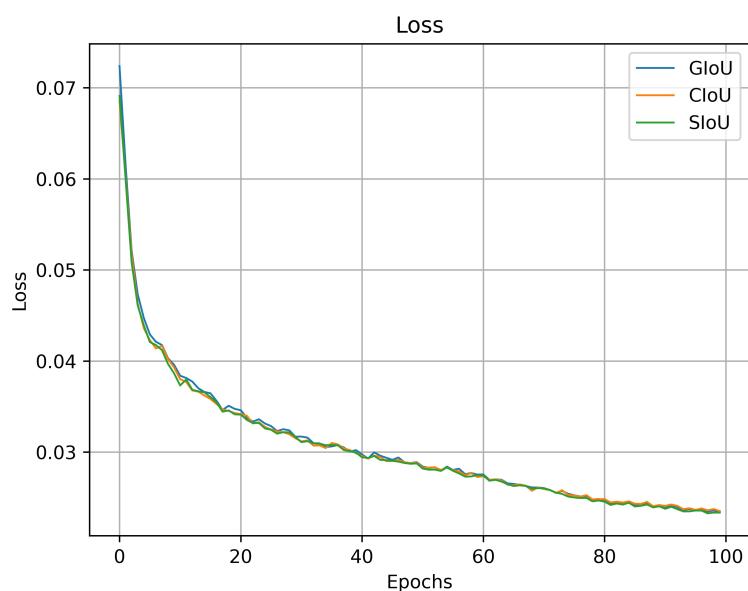


圖 4.12: YOLOv7-UAV Loss 參數調整 Loss 圖

4.1.4.3 Learning Rate

此部分也是選擇以分別調整學習率為 0.001 至 0.005 共五個參數，除了批量次數為 16 以及損失函數設定為 SIoU 外，其他參數都為 YOLOv7 預設的超參數。

從圖4.13及圖4.14可知，在訓練期間，這五種學習率的 mAP 跟 Loss 沒有太大的差異，但從表4.9可以看到，當學習率為 0.005 時，測試集跟驗證集的分數除了 Precision 表現沒那麼出色之外，其他的分數例如 Recall 跟 mAP，在測試集跟驗證集都拿到最高的分數，且兩者的分數差距十分相近，可以尤其判斷應該沒有出現過擬合的情形，基於以上的 reason，選擇以學習率為 0.005 繼續做後續的參數調整。

表 4.9: YOLOv7-UAV 架構模型 Learning Rate 參數表

Epoch	Batch	LR	Opt.	Loss	Set	Precision	Recall	mAP@50
100	16	0.001	SGD	SIoU	train	0.968	0.621	0.616
					test	0.93	0.52	0.505
100	16	0.002	SGD	SIoU	train	0.963	0.648	0.642
					test	0.925	0.517	0.503
100	16	0.003	SGD	SIoU	train	0.951	0.672	0.662
					test	0.904	0.579	0.553
100	16	0.004	SGD	SIoU	train	0.942	0.711	0.703
					test	0.857	0.539	0.509
100	16	0.005	SGD	SIoU	train	0.942	0.718	0.708
					test	0.891	0.62	0.599

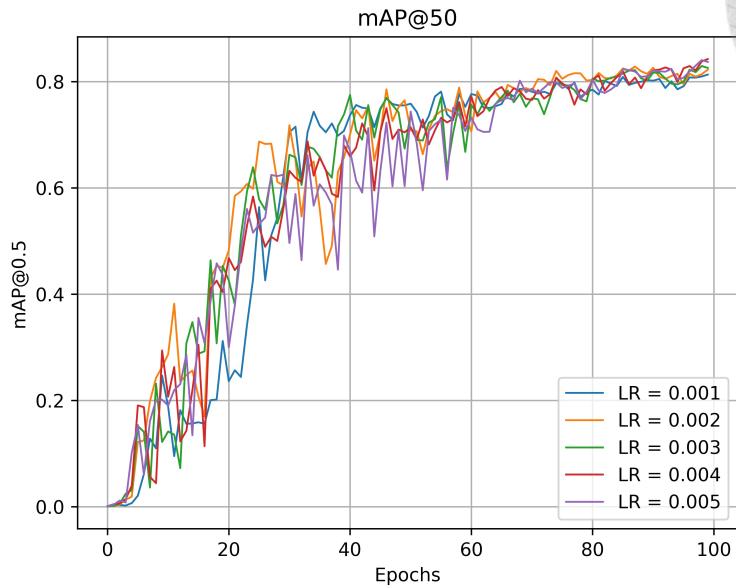


圖 4.13: YOLOv7-UAV LR 參數調整 mAP 圖

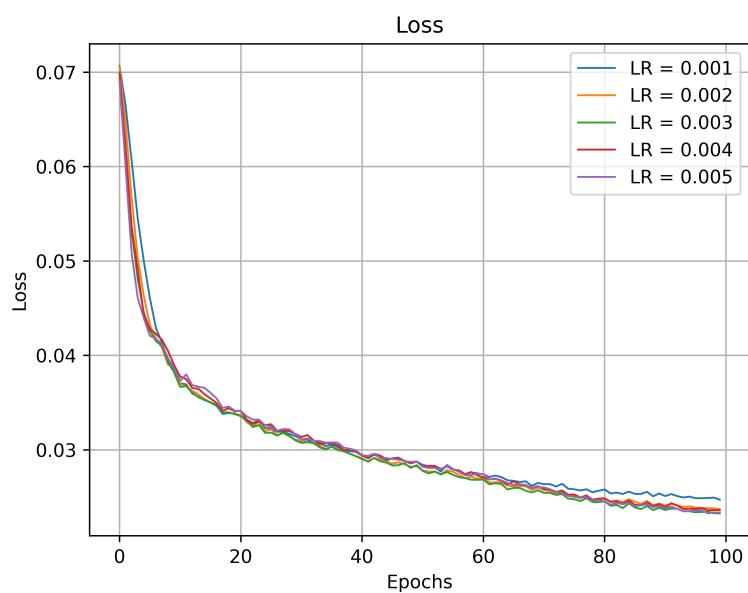


圖 4.14: YOLOv7-UAV LR 參數調整 Loss 圖



4.1.4.4 Epoch、凍結層

在這部分以調整 Epoch 以及凍結層作為測試，Epoch 以 100、120、140，三個參數進行測試，凍結層是以調整 Backbone 的層數，以常用的凍結層數 0、2、4、6，進行交叉組合測試，詳細的訓練數據以及結果為圖4.15、圖4.16以及表4.10。

從圖4.15、圖4.16，可以看出在訓練時，其實 mAP 跟 Loss 都沒有太大的差異，但由表4.10中可以看到，考慮測試集的 mAP 分數時，在沒有任何凍結層，又 Epoch 為 140 時，其 mAP 表現最好 (0.618)，且相較於其他的參數組合，其訓練集以及測試集的分數差異沒有那麼大，所以這部分會以沒有凍結層，且 Epoch 為 140 的參數組合，進行後續的超參數進化。

表 4.10: YOLOv7-UAV 架構模型 Epoch、凍結層參數表

Epoch	Batch	LR	freeze	Loss	Set	Precision	Recall	mAP@50
100	16	0.005	none	SIoU	train	0.942	0.718	0.708
					test	0.891	0.62	0.599
120	16	0.005	none	SIoU	train	0.933	0.739	0.723
					test	0.872	0.632	0.595
140	16	0.005	none	SIoU	train	0.945	0.738	0.73
					test	0.881	0.643	0.618
100	16	0.005	0, 2	SIoU	train	0.946	0.679	0.67
					test	0.871	0.511	0.497
120	16	0.005	0, 2	SIoU	train	0.943	0.747	0.737
					test	0.87	0.621	0.591
140	16	0.005	0, 2	SIoU	train	0.948	0.761	0.752
					test	0.857	0.64	0.613
100	16	0.005	0, 2, 4	SIoU	train	0.95	0.737	0.731
					test	0.868	0.573	0.551
120	16	0.005	0, 2, 4	SIoU	train	0.947	0.705	0.697
					test	0.871	0.57	0.547
140	16	0.005	0, 2, 4	SIoU	train	0.939	0.737	0.727
					test	0.864	0.629	0.599
100	16	0.005	0, 2, 4, 6	SIoU	train	0.948	0.689	0.681
					test	0.918	0.567	0.551
120	16	0.005	0, 2, 4, 6	SIoU	train	0.944	0.744	0.734
					test	0.869	0.635	0.591
140	16	0.005	0, 2, 4, 6	SIoU	train	0.937	0.733	0.723
					test	0.873	0.618	0.588

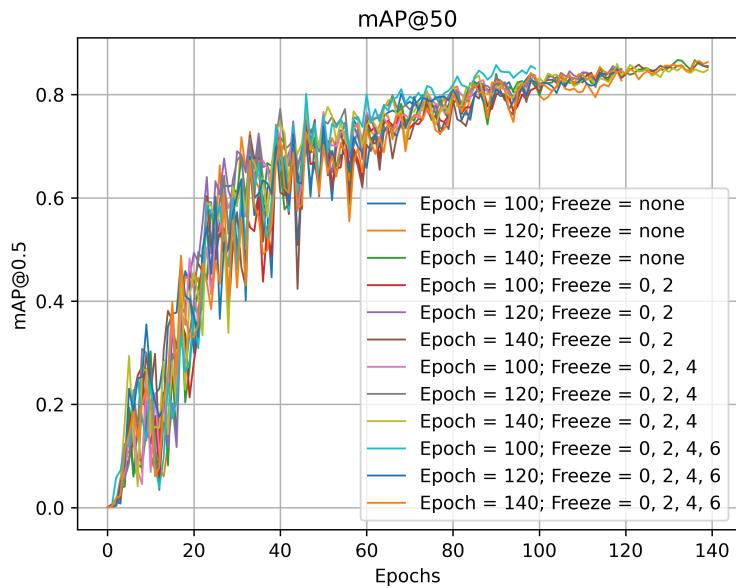


圖 4.15: YOLOv7-UAV Epoch、凍結層參數調整 mAP 圖

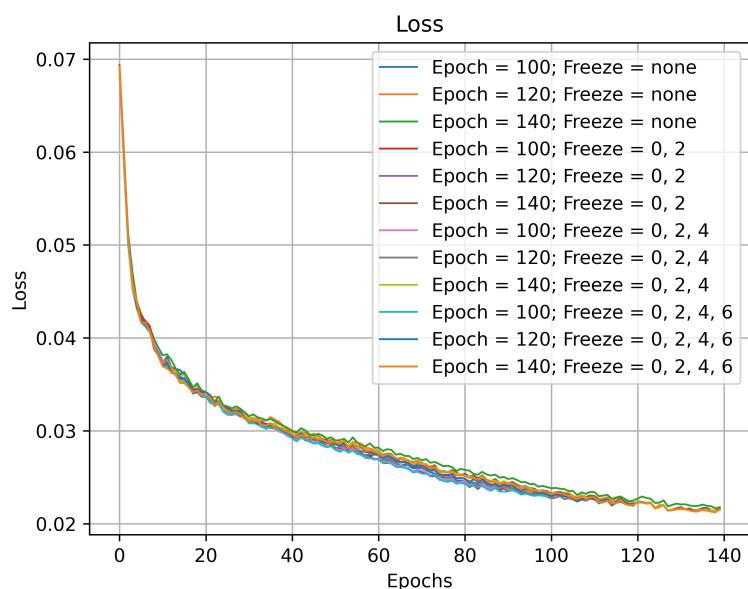


圖 4.16: YOLOv7-UAV Epoch、凍結層參數調整 Loss 圖



4.1.4.5 Evolution

由以上的初步調整超參數可知，YOLOv7-UAV 架構的最佳模型分數為在測試集上 mAP 取得 0.618 的分數 (表4.10)，再經過超參數進化，將模型的表現提升至 0.663 的 mAP 分數 (表4.11)。

表 4.11: YOLOv7-UAV 超參數進化模型分數表

	Epoch	Batch	LR	freeze	Loss	Set	Precision	Recall	mAP@50
Pre-evolve	140	16	0.005	none	SIoU	train	0.945	0.738	0.73
						test	0.881	0.643	0.618
evolved	140	16	0.005	none	SIoU	train	0.945	0.786	0.774
						test	0.859	0.702	0.663



4.1.5 實際裂縫影像測試

由於影像辨識模型最終是用來辨識實際的裂縫，所以利用測試集的模型分數評斷模型的準確率缺乏一定的公信力，所以本研究在這邊另外使用先前去現場拍攝的影片，截取部分幀的影像建立資料集，拍攝地點分別是於一壽橋以及後龍觀海大橋，一共 133 張影像，不使用任何的影像增強手法，來測試模型分數 (表4.12)。另外從表 4.12 中可以看到 YOLOv7 的架構整體的分數都比 YOLOv7-UAV 高出 0.13 的 mAP。

但比較於利用訓練集照片所得到的模型分數 (表4.13)，YOLOv7 以及 YOLOv7-UAV 分別有著 0.417 以及 0.333 的差距，這部分所造成的原因是由於模型資料集為解析度較高的影像所構成，而實際裂縫資料集為利用無人機拍攝的影像進行每一幀的截取所得出，無人機相較於一般的相機影像不同之處，是為容易受到無人機的晃動或是相機視角的影像，導致影像的畫質沒那麼高，所以這也解釋了模型訓練分數跟實際裂縫資料集測試分數的差距。

表 4.12: 實際影像資料集測試模型分數表

	Epoch	Batch	LR	freeze	Loss	Precision	Recall	mAP@50
YOLOv7	140	16	0.002	0, 2	CIoU	0.596	0.523	0.46
YOLOv7-UAV	140	16	0.005	none	SIoU	0.567	0.338	0.33

表 4.13: 超參數進化後架構模型分數表

	Epoch	Batch	LR	freeze	Loss	Precision	Recall	mAP@50
YOLOv7	140	16	0.002	0, 2	CIoU	0.892	0.882	0.877
YOLOv7-UAV	140	16	0.005	none	SIoU	0.859	0.702	0.663



4.1.6 實際裂縫影片測試

在最後利用 YOLOv7 以及 YOLOv7-UAV 表現最佳的模型權重，利用先前錄好的影片進行影像辨識的復現，詳細拍攝資訊以及拍攝結果分別展示於(圖4.14)、(圖4.15)，這樣不僅能避免拍攝所造成的影像差異的變因，也能更好的比對權重的結果(圖4.17)。從以下的結果可以看到，YOLOv7 在同一幀的影像辨識結果下有比較好的表現，可以偵測到更細微的裂縫，且在 mAP 上，YOLOv7 (0.877) 相較於 YOLOv7-UAV (0.663) 高上 0.244 (表4.13)，也所以這可以說明 YOLOv7 的架構在該資料集上有比較好的辨識結果，那後續的實測也利用該權重進行模型的辨識。

表 4.14: 影片拍攝資訊表

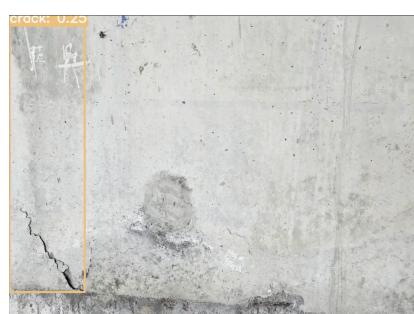
拍攝地點	苗栗縣後龍鎮後龍觀海大橋
無人機載具	DJI Inspire 2

表 4.15: 影像測試結果表

架構種類	影像總長度 (秒)	FPS	影片總幀數	含有辨識框幀數	辨識花費時間 (秒)
YOLOv7	34.72	4.46	155	79	43.73
YOLOv7-UAV		5.60	194	12	45.42



(a) YOLOv7-UAV



(b) YOLOv7-UAV

圖 4.17: 實際裂縫影片測試圖

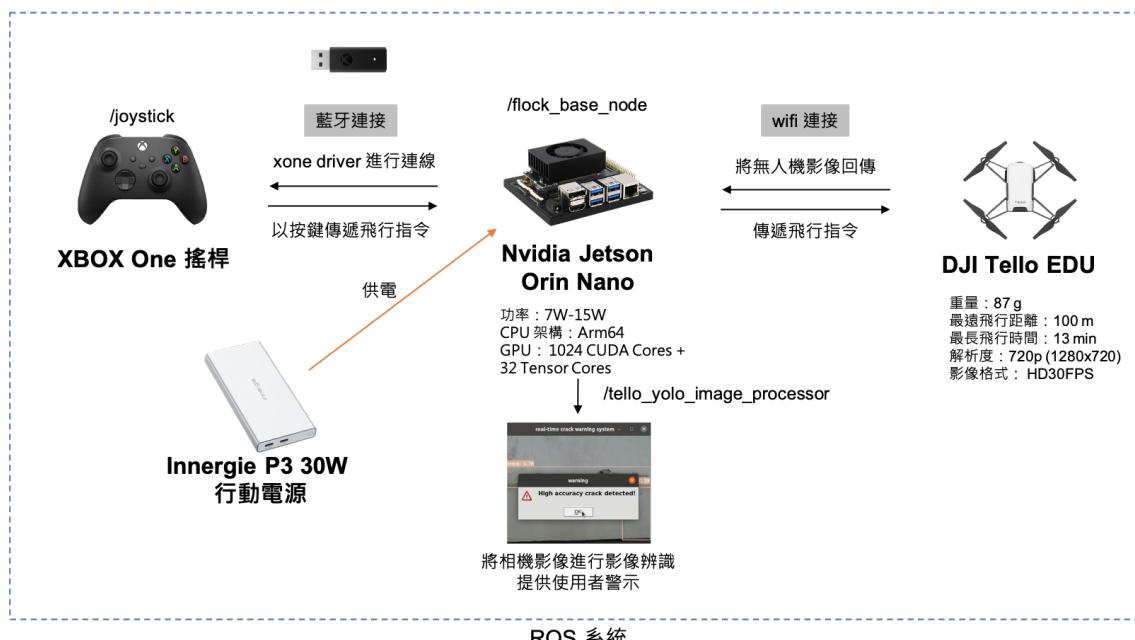


4.2 無人機影像辨識系統實測

本研究利用兩種無人機進行無人機自動影像辨識系統的建置，第一部分是用 DJI Tello EDU 將進行系統整體架構的梳理，以確保整體架構的可行性跟穩定性，最後再將架構移至 DJI Inspire 2 上。並於台北市文山區以及苗栗縣後龍鎮的兩座橋梁進行系統的實測，確認系統的在外在環境的可靠性跟穩定性，並於下面兩個子章節展現其成果。

4.2.1 DJI Tello EDU

在研究中，DJI Tello EDU 無人機的輕量級和有限通訊距離帶來了一些限制，特別是在室外環境下，即使是輕微的風也可能導致無人機飛行不穩定。因此，我們決定將該無人機用來進行系統的測試跟梳理，為了降低飛行風險，我們選擇在建築物外部和橋梁連接路面等相對較為穩定的區域進行實地測試。





4.2.1.1 建物裂縫辨識

由於前面提及 DJI Tello EDU 飛行穩定度的緣故，本研究選擇於台北市大安區某研究大樓的外牆進行無人機影像辨識的測試，並利用 XBOX One 遊戲搖桿進行 DJI Tello EDU 的飛行，以測試整體系統的可行性，辨識結果為（圖 4.19、圖 4.20）。



圖 4.19: 建物裂縫辨識結果圖

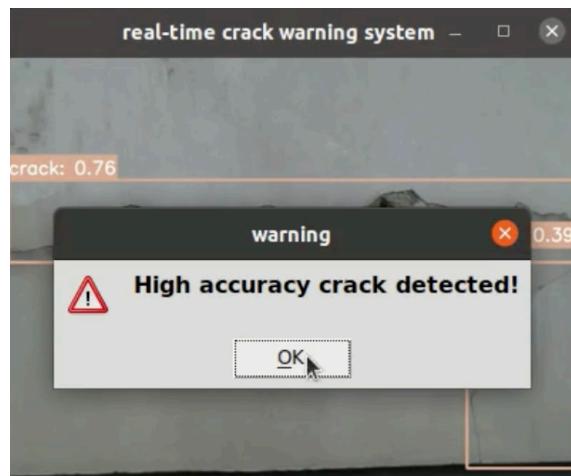


圖 4.20: 建物裂縫辨識警示圖



4.2.1.2 一壽橋

一壽橋位於台北市文山區，橫跨景美溪，連接樟新街以及寶高路，本研究於樟新街右側橋梁連接路面處，利用 DJI Tello EDU 進行裂縫的影像辨識（表4.16），由於 DJI Tello EDU 的硬體限制，所以於天氣良好的情況下進行系統的實測，，並利用 XBOX One 把手進行 DJI Tello EDU 的飛行（圖4.21a），影像辨識的結果如（表4.17）、（圖4.21b）所示。

表 4.16: DJI Tello EDU 一壽橋拍攝資訊表

拍攝地點	台北市文山區一壽橋
拍攝日期	2024 年 05 月 31 日
無人機載具	DJI Tello EDU

表 4.17: DJI Tello EDU 一壽橋測試結果表

影像總長度 (秒)	FPS	影片總幀數	含有辨識框幀數	辨識花費時間 (秒)
37.09	4.44	165	43	43.69



(a) DJI Tello EDU 飛行

(b) 一壽橋辨識結果

圖 4.21: DJI Tello EDU 一壽橋現地實測示意圖



4.2.2 DJI Inspire 2

本研究選用 DJI Inspire 2 作為實驗機具，由於其為高性能無人機，能夠在外部環境中實現相對穩定的飛行。其懸停系統表現良好，使其非常適合用於實踐我們建立的影像辨識系統。考慮到民航局法規的限制以及樣本獲取的困難，我們選擇了台北市文山區的一壽橋和苗栗縣後龍鎮的後龍觀海大橋作為實地測試的地點，以進行硬體和系統的實測。

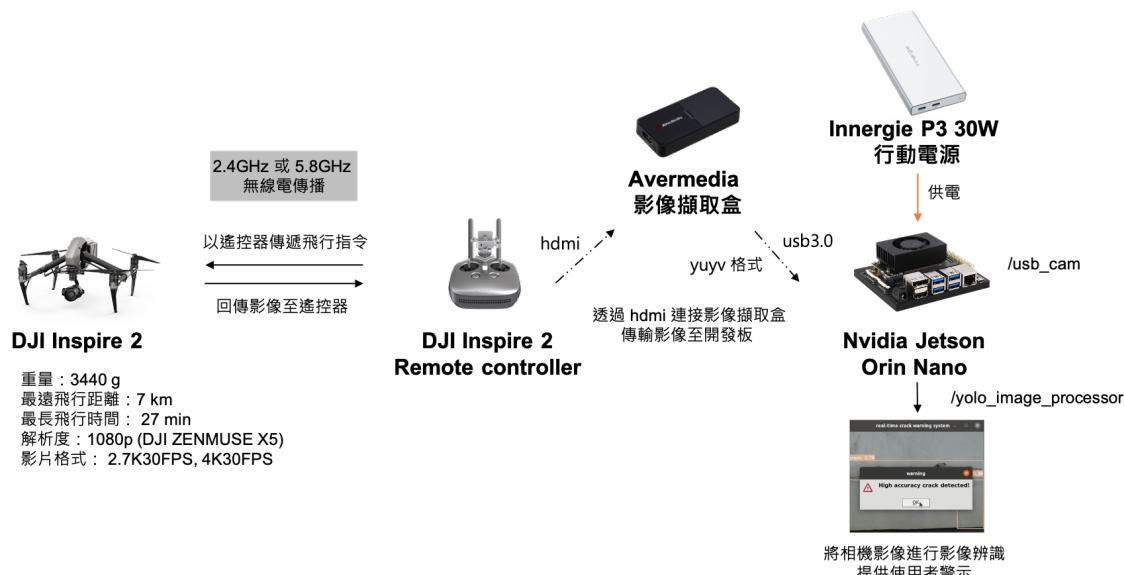


圖 4.22: DJI Inspire 2 硬體架構圖



4.2.2.1 一壽橋

一壽橋位於台北市文山區，橫跨景美溪，連接樟新街以及寶高路，由於 DJI Inspire 2 本身飛行限制的緣故，由於橋墩有裂縫的位置太靠近橋面版下方，無法靠近橋墩拍攝，所以選擇在橋梁連接樟新街的側面處進行拍攝（表4.18），利用 DJI Inspire 2 自身的遙控器進行飛行（圖4.23a），並進行裂縫的影像辨識，其結果為（表4.19）、（圖4.23b）所示。

表 4.18: DJI Inspire 2 一壽橋拍攝資訊表

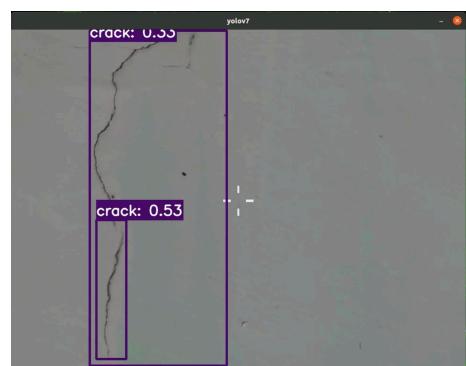
拍攝地點	台北市文山區一壽橋
拍攝日期	2024 年 05 月 31 日
無人機載具	DJI Inspire 2

表 4.19: DJI Inspire 2 一壽橋測試結果表

影像總長度 (秒)	FPS	影片總幀數	含有辨識框幀數	辨識花費時間 (秒)
102.45	5.56	569	175	137.16



(a) DJI Inspire 2 飛行



(b) 一壽橋辨識結果

圖 4.23: DJI Inspire 2 一壽橋現地實測示意圖



4.2.2.2 後龍觀海大橋

後龍觀海大橋位於苗栗縣後龍鎮，橫跨後龍溪，並位於後龍溪與台灣海峽的出海口處，是為東北至西南向的橋梁。本研究於後龍觀海大橋的 P1 橋墩處進行無人機的飛行 (表 4.20)、(圖 4.24a)，並進行 DJI Inspire 2 的影像辨識系統測試，辨識結果為 (圖 4.24b) 所示。

表 4.20: DJI Inspire 2 後龍觀海大橋拍攝資訊表

拍攝地點	苗栗縣後龍鎮後龍觀海大橋
拍攝日期	2024 年 05 月 03 日
無人機載具	DJI Inspire 2

表 4.21: DJI Inspire 2 後龍觀海大橋測試結果表

影像總長度 (秒)	FPS	影片總幀數	含有辨識框幀數	辨識花費時間 (秒)
102.45	5.56	569	175	137.16



(a) 後龍觀海大橋橋墩圖



(b) 後龍觀海大橋辨識結果

圖 4.24: DJI Inspire 2 後龍觀海大橋現地實測示意圖



4.3 實驗結論與探討

本研究首先進行了 YOLOv7 和 YOLOv7-UAV 模型的架構比較，並對其進行了參數調整，以使其達到最佳狀態。另外以免在訓練模型時取得高分，但在實際辨識時效果不佳的情形發生，研究使用了以先前利用真實的橋樑裂縫影像建立測試集，但圖片的辨識情形也無法完全符合影片辨識的情形，所以本研究另外使用預先錄製的橋梁裂縫影像 bag 檔，以復現成果。但比較訓練模型分數的結果、真實裂縫資料集的結果，以及 bag 檔的辨識結果，YOLOv7 架構下的權重表現都比較好。造成這樣結果的可能原因是，YOLOv7-UAV 最初是設計用於高空無人機的車輛辨識，其特徵相對於本研究資料集更難抓取，因此在該架構下模型表現不佳。

為驗證系統的可行性，研究在三個地點進行了 DJI Tello EDU 和 DJI Inspire 2 的影像辨識系統實測，以驗證系統的穩健性和適用性。實驗結果表明，系統在不同環境下都能正常運行，為未來橋梁檢測的自動化提供了寶貴的參考。



第五章 結論與未來發展

5.1 結論

本研究成功建立了一個自動化的即時裂縫影像辨識系統，這個系統可以在無人機拍攝影像後立即進行裂縫辨識，為未來橋梁檢測自動化奠定了基礎，其主要貢獻為不同於傳統方法讓該系統達成三大目的，分別為影像辨識及處理的自動化、即時的影像辨識及軟硬體的整合，這三者有其相輔相成的關係，在以下做說明。

為了進行裂縫的影像辨識，經過文獻回顧後選擇使用 YOLOv7 作為基礎模型，並進行了不同的模型架構、手動調參數及超參數進化等優化方法的測試，以確保在這個應用場景下獲得最佳模型表現。在純圖片影像辨識方面，超進化後的模型參數所得到的 YOLOv7 架構訓練集的 mAP 分數比 YOLOv7-UAV 高出了 0.209；測試集的 mAP 也高出了 0.214，尚未經過超參數進化的表現 YOLOv7 也都是優於 YOLOv7-UAV 的架構。此外，本研究也使用實際裂縫的資料集進行模型的比較，為了反應真實的辨識情形，也使用 bag 檔進行模擬實際辨識，以能夠更公平地評估模型權重的優劣表現，其結果為 YOLOv7 的架構比較能偵測到細小的裂縫，且分數表現上都優於 YOLOv7-UAV 架構，其資料集使用的是公開的鋪面相關的資料集。

接著，本研究將表現最好的架構權重導入 ROS 系統中，使其能夠在開發板上進行即時的影像辨識，實現影像處理的自動化。這樣的成果也證明了無人機可以配合影像辨識模型檢測橋梁狀態，用於監測安全性。同時，利用開發板的低功率和方便攜帶特性，建立了高靈活度的系統，方便部署在戶外的環境中，並實現了即時辨識系統，提高了辨識和警報的即時性。

在硬體整合方面，ROS 系統成功整合了 DJI Tello EDU 的軟硬體，並透過 wifi 傳輸影像資料至開發板中。在 DJI Inspire 2 無人機上，由於硬件限制無法透過 ROS 系統控制飛行，因此利用無人機自帶搖桿進行飛行控制，影像傳輸是透過無線電的方式進行傳輸至開發板進行處理及辨識。儘管無法實現全自動化，但結果證明了即時影像辨識的可行性。在即時的方面，相較於傳統橋檢人員來回於現地以及辦公室登錄系統，或是利用橡皮艇的方式去查看橋梁裂縫，本研究的系統只需要運轉十幾秒，就可以進行即時的辨識結果進行檢視，相對於原先的方法還是快速許多，且使用 ROS 系統也方便後續演算法或是硬體的添加。

本研究系統實現了自動化即時裂縫影像辨識，為未來橋梁檢測自動化提供了有力支持，但在模型即時性或是模型上的改良，以及 ROS 系統部分的硬體集成等方面仍有進一步優化的空間。

5.2 限制與討論

本研究雖然已建置初步的即時影像辨識系統，但還是有許多的限制，以至於系統沒辦法真正的完善，以下將以條列式的方式將詳細列舉本研究的限制，並進一步地進行說明。

- 資料集缺乏：由於針對於遠距離橋梁裂縫的資料集找尋不易，所以目前研究

只對於近距離的裂縫資料集進行訓練，模型缺乏排除背景或是雜物的能力。

- DJI Inspire 2 硬體限制：由於 DJI Inspire 2 硬體不同於 DJI Tello EDU，不支援 ROS 系統，所以無法像 DJI Tello EDU 一樣，可以使用非官方的遙控器進行操控，也無法直接利用 ROS 系統整合得取無人機的影像的方式，進行後續的處理，所以沒辦法像 DJI Tello EDU 一樣利用 ROS 貫徹整個系統，進行完全的自動化。
- 訓練模型硬體限制：由於本研究所訓練模型的效能並非為最理想狀態，故無法以最大的限度進行模型的訓練，像是在進行超參數的優化時，無法進行高次數的迭代，所以在本研究中盡可能以現有的硬體條件進行模型訓練，以得到最佳的權重。
- 相機硬體限制：由於本研究所使用的 DJI Inspire 2 無人機相機是為 DJI 自產的相機 DJI ZENMUSE X5，為二軸雲台相機，但因為相機位於無人機下方的緣故，所以視角受到限制，且 DJI Inspire 2 的量體太大，不方便太過靠近橋梁本身，拍取到更細節的裂縫影像，當橋梁的跨度太近，或是橋梁高度太低時，無人機也不太方便進入到其中進行拍攝。然而在許多橋檢報告書中，橋梁的裂縫大多位於橋梁的橋面板下方，所以辨識的範圍就受到許多的限制。
- 法規限制及安全考量：由於民航局的法令規定，禁航區、限航區及航空站或飛行場四周之一定距離範圍內，禁止從事遙控無人機飛航活動；航空站或飛行場四周之一定距離範圍由民航局公告之。但就以台北市而言，因為腹地狹小，很多老化的橋梁都位於機場附近，且都市人口稠密，飛行無人機難免會有飛安風險，所以本研究無法於台北市區內進行實際橋梁案例的實驗。



5.3 未來發展

因為本研究導入了 ROS 系統，所以可以方便進行後續硬體的整合，使得整個系統更加完善，更能輔助目前的橋檢系統，以達到真正自動化的目的，在下面以條列式的方式，詳細列出未來發展的可能項目。

- 無人機自航：可於系統中另外添加深度相機以及光達，利用 SLAM 演算法以及路徑規劃演算法，結合 ROS 系統以達成無人機的自動導航，除了增加其飛行的安全性，也可以讓整體的系統架構都完全建立於 ROS 系統中，以達到完全的自動化。另外由於使用了深度相機及光達，可以另外計算裂縫實際長度，以取得更全面性的裂縫資料。
- 裂縫的語義分割標註：未來可以利用訓練實例分割 (Instance Segmentation) 的模型來用以標註裂縫，實例分割相較於邊界框的方式，精確度較高，相對於傳統的分類或語義分割，更能夠區分不同裂縫之間的邊界，這對於在裂縫繁多的情況下，更能標記出裂縫的位置跟範圍，讓使用者更能了解裂縫的資訊。
- 結合點雲的裂縫標註：結合 3D 點雲 (3D Point Cloud)，可以將裂縫標註於 3D 點雲模型中，除了可以融合真實的空間資訊讓使用者可以更加了解裂縫所在的位置，另外也可結合數位孿生的概念方便往後更新。
- 軟硬體的後續進化：由於該系統使用了 ROS 系統進行軟硬體整合，除了像前面所提及的可以添加支援 ROS 系統的深度相機以及光達，也可以後續添加往後更精進的演算法，讓整個系統可以隨著時間和硬體發展保持同步，不斷提升其泛用性跟性能。



參考文獻

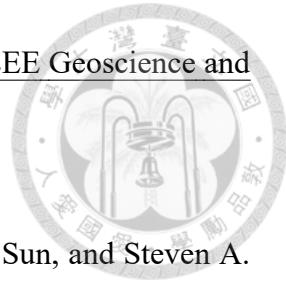
- [1] 交通部運輸研究所. 全國橋梁統計資訊網, 4 2024.
- [2] Jeroen Guinee. Handbook on life cycle assessment. an operational guide to the iso standards. The International Journal of Life Cycle Assessment, 7:311–313, 09 2001.
- [3] 國家運輸安全調查委員會. 重大運輸事故調查報告第一冊：南方澳大橋斷裂重大公路事故. Technical report, 國家運輸安全調查委員會, 2020.
- [4] 運輸計畫及陸運組. 我國橋梁檢測方式之發展探究. Technical report, 交通部運輸研究所, 2018.
- [5] 王炤烈, 宋裕祺, 林曜滄, 彭康瑜, and 黃炳勳. Morandi 橋崩塌帶來的省思與建議. 土木水利, 45(5):44–45, 2018.
- [6] Linjie Xing, Xiaoyan Fan, Yaxin Dong, Zenghui Xiong, Lin Xin, Yang Yang, Haicheng Bai, and Chengjiang Zhou. Multi-uav cooperative system for search and rescue based on yolov5. SSRN Electronic Journal, 01 2021.
- [7] Julian Kedys, Igor Tchappi, and Amro Najjar. Uavs for disaster management - an exploratory review. Procedia Computer Science, 231:129–136, 01 2024.
- [8] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. Proceedings of the IEEE, PP:1–20, 03 2023.

- [9] 饒見有, 李志清, 劉光晏, and 林昭宏. 無人機搭配 ai 影像辨識應用於橋梁檢測之研究 (2/2)-無人機自動化檢測架構探討. Technical report, 交通部運輸研究所, 2023.



- [10] 王姿樺 (Zi-Hua Wang), 高書屏 (Szu-Pyng Kao), and 林志憲 (Jhih-Sian Lin). 應用深度學習技術輔助橋梁裂縫辨識. 航測及遙測學刊, 27(4):247–257, Dec 2022.
- [11] 張永辰, 高書屏, 王豐良, and 林志憲. 應用 uav 影像及深度學習技術輔助橋梁裂縫量化分析. 國土測繪與空間資訊, 11(1):15–34, Jan 2023.
- [12] Qianyun Zhang, Kaveh Barri, Saeed Babanajad, and Amir Alavi. Real-time detection of cracks on concrete bridge decks using deep learning in the frequency domain. Engineering, 7, 11 2020.
- [13] Kui Luo, Xuan Kong, Jie Zhang, Jiexuan Hu, Jinzhao Li, and Hao Tang. Computer vision-based bridge inspection and monitoring: A review. Sensors, 23(18), 2023.
- [14] Matthew Gillins, Daniel Gillins, and Christopher Parrish. Cost-effective bridge safety inspections using unmanned aircraft systems (uas). 02 2016.
- [15] Najiha Yusof, Ali Sophian, Hasan Firdaus Mohd Zaki, Ali Bawono, Abd Halim Embong, and Arselan Ashraf. Assessing the performance of yolov5, yolov6, and yolov7 in road defect detection and classification: a comparative study. Bulletin of Electrical Engineering and Informatics, 13:350–360, 02 2024.
- [16] Jian Zhang, Songrong Qian, and Can Tan. Automated bridge crack detection method based on lightweight vision models. Complex Intelligent Systems, 9:1–14, 09 2022.
- [17] Xin Wu, Wei Li, Danfeng Hong, Ran Tao, and Qian Du. Deep learning for unmanned

aerial vehicle-based object detection and tracking: A survey. IEEE Geoscience and Remote Sensing Magazine, 10(1):91–124, 2022.



- [18] Alison Cleary, Kristopher Yoo, Paul Samuel, Sean George, Fei Sun, and Steven A. Israel. Machine learning on small uavs. In 2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), pages 1–5, 2020.
- [19] Qiwen Qiu and Denvid Lau. Real-time detection of cracks in tiled sidewalks using yolo-based method applied to unmanned aerial vehicle (uav) images. Automation in Construction, 147:104745, 03 2023.
- [20] Chen Xing, Xi Liang, and Yanna Ma. A solution to improve object detection for images captured by uav-mounted camera. In 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT), pages 317–320, 2019.
- [21] Tianjie Zhang, Donglei Wang, and Yang Lu. A data-centric strategy to improve performance of automatic pavement defects detection. Automation in Construction, 160:105334, 2024.
- [22] Siling Feng, Yuanlong Wang, Mengxing Huang, and Guanjun Wang. Aerial object detection by uav based on improved yolov7. In 2023 IEEE 6th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), pages 63–67, 2023.
- [23] Yingkun Wei, Jiahui Li, Wenwen Duan, Xinmin Li, Xiaoqiang Zhang, and Yi Huang. Yolov7-uav: Improved yolov7 algorithm for small object detection in uav image scenarios. In 2023 International Conference on Artificial Intelligence of Things and Systems (AIoTSys), pages 64–70, 2023.

- [24] Yong Shi, Limeng Cui, Zhiqian Qi, Fan Meng, and Zhensong Chen. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3434–3445, 2016.



- [25] Lei Zhang, Fan Yang, Yimin Daniel Zhang, and Ying Julie Zhu. Road crack detection using deep convolutional neural network. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3708–3712, 2016.

- [26] Fan Yang, Lei Zhang, Sijia Yu, Danil Prokhorov, Xue Mei, and Haibin Ling. Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, 21(4):1525–1535, 2020.

- [27] Mohammadreza Sabouri and Alireza Sepidbar. Sut-crack: A comprehensive dataset for pavement crack detection across all methods. *Data in Brief*, 51:109642, 10 2023.

- [28] Dwyer B., Nelson J., and Hansen T. Roboflow, 2024.

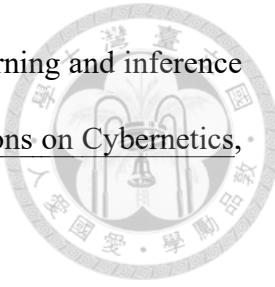
- [29] Adrian Rosebrock. Intersection over union (iou) for object detection, 2023.

- [30] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 658–666, Los Alamitos, CA, USA, jun 2019. IEEE Computer Society.

- [31] Meixia Fu, Qu Wang, Jianquan Wang, Lei Sun, Zhangchao Ma, Chaoyi Zhang, Wan-qing Guan, Qiang Liu, Danshi Wang, and Wei Li. Deep cnn-based materials location and recognition for industrial multi-crane visual sorting system in 5g network. *Applied Sciences*, 13:1066, 01 2023.

- [32] Zhaojun Zheng, Ping Wang, Dongwei Ren, Wei Liu, Rongguang Ye, Qinghua Hu,

and Wangmeng Zuo. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. IEEE Transactions on Cybernetics, 52(8):8574–8586, 2022.



- [33] Hong Vin Koay, Joon Huang Chuah, Chee Onn Chow, Yang-Lang Chang, and Keh Yong. Yolo-rtuav: Towards real-time vehicle detection through aerial images with low-cost edge devices. Remote Sensing, 13, 10 2021.
- [34] Zhora Gevorgyan. Siou loss: More powerful learning for bounding box regression, 2022.
- [35] Xiangqian Xu and Xing Li. Research on surface defect detection algorithm of pipeline weld based on yolov7. Scientific Reports, 14, 01 2024.
- [36] Dengjie Yang, Changyun Miao, Xianguo Li, Yi Liu, Yimin Wang, and Yao Zheng. Improved yolov7 network model for gangue selection robot for gangue and foreign matter detection in coal. Sensors, 23:5140, 05 2023.
- [37] John H. Holland. Genetic algorithms. Scientific American, 267(1):66–73, 1992.
- [38] Musatafa Albadr, Sabrina Tiun, Masri Ayob, and Fahad Al-Dhief. Genetic algorithm based on natural selection theory for optimization problems. Symmetry, 12:1–31, 10 2020.
- [39] medusalix. medusalix xone: Linux kernel driver for xbox one and xbox series x|s accessories, 2021.
- [40] Clyde Mcqueen. Clydemcqueen flock: Ros driver for dji tello drones, 2018.
- [41] Phuoc Nguyen. Phuoc101 yolov7_ros: Ros package for official yolov7, 2022.
- [42] Fredrik Lundh. An introduction to tkinten, 1999.