

中華民國第 65 屆中小學科學展覽會

作品說明書

高級中等學校組 工程學科(一)

第二名

052313

AI 無人機檢測系統在建築磁磚剝落風險評估中的應用

學校名稱： 新北市立新北高級中學

作者：	指導老師：
高二 李芷棻	謝進生
高二 廖姿婷	張宇熙

關鍵詞： 無人機、磁磚剝落、熱成像

AI 無人機檢測系統在建築磁磚剝落風險評估中的應用

摘要

本研究開發了一套結合無人機、紅外線熱顯像技術與 YOLO 模型的建築物外牆磁磚剝落檢測系統。該系統利用無人機搭載樹莓派與紅外線雙光雲台相機，對建築外牆進行檢測，以精確辨識空鼓與水泥裸露區域。我們透過紅外線熱顯像技術識別牆面溫度異常區域，並運用 YOLO 模型標示裸露水泥區域，最終將標示後的圖像進行分析與疊合比較，以評估磁磚剝落風險，並生成評估報告。

此外，本系統整合了 3D 建模技術，以重建大樓外觀，並結合風險評估報告，直觀標記出具有磁磚剝落風險的區域，提升使用者對建築維護的理解與掌握。該系統除了能提高檢測的準確性與效率，還有效降低時間與人力成本，為建築維護提供可靠的數據支持。

壹、前言

一、研究動機

台灣許多老舊建築的外牆使用磁磚作為主要覆蓋材料，然而，隨著建築年久失修，磁磚脫落的問題日益嚴重，不僅影響市容，更對公共安全構成重大威脅。近年來，磁磚剝落導致的事故屢見不鮮，例如台北市忠孝東路等繁忙路段，多起磁磚掉落事件造成行人受傷，甚至導致交通意外，讓社會輿論對建築安全產生高度關切。部分事件雖未造成人員傷亡，但已形成「空中殺手」的隱憂，讓民眾對老舊建築的安全性產生疑慮。為此，政府針對特定條件下的建築物訂定外牆檢測政策，要求定期檢修，以降低事故風險。

然而，目前普遍使用的傳統人工敲擊檢測法，依賴人員高空作業，不僅存在極大的安全風險，也因其費時且成本高昂，特別是在大面積建築外牆檢測時，費用往往動輒數十萬元，使得許多建築所有者望而卻步，導致潛在危險未能及時發現並處理。此外，人工檢測的準確度受限於技術人員的經驗，難以做到全面、精確的評估，增加了外牆安全管理的困境。

因此本研究計畫運用無人機結合紅外線技術與影像辨識，開發自動化外牆檢測系統，相對於傳統人工提高 90%的檢測效率、並確保作業人員的安全性。本組希望所發展模式能快速偵測磁磚剝落或結構異常的潛在風險，並透過機器學習建模提供精確的數據分析，協助建築管理者及早進行維護修繕，從而預防事故發生。

二、研究目的

- (一) 開發無人機檢測系統：結合無人機、紅外線與影像辨識技術，提升外牆檢測自動化程度。
- (二) 提升檢測準確性與效率：運用機器學習技術，提高磁磚剝落與結構異常的識別精度與速度。
- (三) 優化建築安全管理與應用效益：提供即時檢測數據，協助維護決策，並評估無人機檢測的可行性與成本效益。

三、文獻探討

根據內政部資料，截至2023年第2季，屋齡30年以上的老舊房屋已達483.8萬戶，占全國住宅總戶數比率約為54%，亦即全臺有一半以上的房子為超過30年以上的老宅。以臺北市為例，依臺北市建築管理工程處統計，104年到109年間，平均每年約有761件大樓外牆磁磚剝落的通報案；110年為345件，其中190件有立即危險，47件列管中。此外，臺灣地震頻繁，且近年來氣候劇烈變化，推動老舊住宅外牆的補強及維護等措施，刻不容緩。然而，老舊建築雖多，檢測與維修的普及率卻偏低，且成本高昂。

縣市／地區	0-15年 (%)	16-30年 (%)	超過30年 (%)	平均屋齡 (年)
全 國	16.29	30.66	53.03	33
新北市	17.18	33.64	49.16	31.44
台北市	10.52	17.16	72.32	37.78
桃園市	22.46	40.05	37.49	27.86
台中市	19.41	36.03	44.56	29.87
臺南市	15.88	29.61	54.51	34.33
高雄市	14.89	30.80	54.29	33.19

資料來源：內政部 2023年第2季數據 整理：馬揚異

圖 1、全國及六都住宅屋齡統計

(資料來源：馬揚異。(2023)，全台平均屋齡33年，奪命磁磚兩10年後未改善，今周刊1399期)

《台北市建築物外牆安全診斷檢查及申報辦法》規範十一層以上、屋齡十五年以上建築物，每六年申報一次；三十年以上建築每三年申報一次。須由管委會（若無管委會則由全體住戶）找市政府核可的廠商進行檢診，若有危險需進一步修繕，檢修完成後提交報告完成申報。外牆維護雖是住戶自身的責任，但過去民眾沒有出資檢修外牆的觀念，尤其修繕的費用往往數十萬以上。

■ 修正後外牆檢查計畫 費用計算



建物規模：
地上15層建物，雙拼(30戶)
補助：檢查6萬元(依臨道路建物寬度)
診斷8千元
以200元/m²單價計算
3面申報費用117萬(單面約39萬)
依收費參考表，金額如右

收費 參考表	檢查打診	檢查打診 +鬆脫敲除	檢查打診 +鬆脫敲除 +防水塗料
洗窗機	35元/m ²	43元/m ²	48元/m ²
蜘蛛人	40元/m ²	48元/m ²	53元/m ²
吊車	47元/m ²	56元/m ²	62元/m ²

		檢查打診	檢查打診 +鬆脫敲除	檢查打診 +鬆脫敲除 +防水塗料
洗 窗 機	三面	17.64萬	21.672萬	24.192萬
	單面	5.88萬	7.224萬	8.064萬
蜘蛛 人	三面	20.16萬	24.192萬	26.712萬
	單面	6.72萬	8.064萬	8.904萬
吊 車	三面	23.688萬	29.736萬	31.248萬
	單面	7.896萬	9.408萬	10.416萬

診斷費用2萬元/件

圖 2、建築物外牆磁磚修檢費用計算

(資料來源：郭家安。(2020)。北市外牆安檢參考價價差逾20倍。自由時報。)

以一棟大樓為例，該建築長 35 公尺、寬 37 公尺、高 48 公尺，單面外牆面積達 1680 平方公尺。根據市府提供的市價，每平方公尺檢測費用為 200 元，則完整檢測費用高達 117 萬元，單面平均約 39 萬元。上述資訊顯示，傳統外牆檢測方式成本高昂，且價格波動較大，也就是說現行的檢測模式存在改善空間。

台灣近 10 年來常有建築物外牆磁磚掉落的意外，像是台北萬華、新北蘆洲、高雄楠梓、岡山等地區，就陸續傳來外牆大面積掉落、下起「磁磚雨」的事故。如下圖 3，整理了部分 2013 年到 2023 年發生磁磚剝落所導致的砸傷事件。

時間	事件經過
2013/05	台北市京華大廈外牆剝落砸傷女童，致其顱骨骨折、腦損跛足，台北地檢署根據《建築法》規範，依過失傷害罪嫌起訴全體 12 名住戶。
2013/10	基隆市麗榮皇冠大樓外牆大理石墜落，砸死路過婦人。大樓管委會三人遭判有期徒刑三個月、易科罰金 9 萬元；施工人員有期徒刑四個月、易科罰金 12 萬元。
2015/03	台北市聯合報大樓外牆花崗石板墜落，一名補習班老師遭砸死。 北市府依《建築法》開罰聯合報大樓所有人 30 萬元。
2016/01	霸王寒流來襲、磁磚受熱脹冷縮，台北市三天內發生三十多起外牆剝落事故，一名女子頭部遭砸傷。
2021/09	高雄市前鎮區大樓磁磚剝落砸傷路人腿部。
2022/09	台北市中山區商圈大樓磁磚掉落，砸傷路過民眾頭部。
2022/12	新北市新店區大樓磁磚掉落砸傷路人頭部。
2023/09	新北市新莊區公寓外牆磁磚與水泥脫落，砸傷行經機車騎士。新北市工務局處建築物所有權人 6 萬元罰鍰。

圖 3、近10年外牆剝落死傷事故

(資料來源：馬揚異。(2023)，全台平均屋齡33年，奪命磁磚雨10年後未改善，今周刊/1399期)

除了建築物外牆磁磚掉落造成的砸傷事件，現在大多數是採用人工敲擊檢測法，仰賴高空作業，存在極大的安全風險，其中意外包括以下幾種：

（一）高處墜落

在高處作業時，如果未正確佩戴安全帶或使用不當，容易發生墜落事故。

（二）吊籠事故

吊籠未經適當檢查，或固定支架不穩定，可能導致吊籠掉落。

（三）施工架問題

施工架未妥善設置防墜設施，或組配作業主管未確認安全狀況，即造成事故。

在台灣有類似的案例，113年工人於施工架上從事外牆補磚作業發生墜落災害致死，導致其從高差7公尺外牆施工架外側開口墜落，其原因就是在高度2公尺以上外牆施工架外側開口未設置防墜設施，而造成憾事。

根據不同縣市進行外牆磁磚檢測受損評估表，目前地方政府執行上所使用之基準，多將發生剝落及隆起面積達「 $5m^2$ 」或「 $2500cm^2$ 」以上視為「重度劣化」。

表1、各縣市外牆飾面檢修評估表

表1-1、臺北市老屋健檢外牆安全項目評估表

項目	分項	評定內容		評定等級	評定勾選	細項等級	分項等級
外牆	外牆構造	外牆面飾材料剝落、鼓脹現象	4 處以下且面積合計未超過 $5m^2$	D			
			5 處以上或面積合計達 $5m^2$ 以上	E			
		外牆有明顯裂縫、混凝土塊剝落或鋼筋裸露露鑽蝕情形	2 處以下	D			
			3 處以上	E			
		共用部分之窗框或窗扇呈現嚴重變形現象、啟閉困難		D			
		外牆呈現嚴重滲漏水白華現象		D			

表1-2、桃園市老屋健檢外牆磁磚收損項目評估表

項目	分項	評定內容		評定等級	評定勾選	細項等級	分項等級
外牆安全	外牆磁磚受損調查	外牆面飾材料剝落、鼓脹現象	4 處以下且面積合計未超過 $5m^2$	C			
			5 處以上或面積合計達 $5m^2$ 以上	D			
		外牆有明顯裂縫、混凝土塊剝落或鋼筋裸露露鑽蝕情形	2 處以下	C			
			3 處以上	D			
		共用部分之窗框或窗扇呈現嚴重變形現象、啟閉困難		C			
		外牆呈現嚴重滲漏水白華現象		C			

備註

1. 表列每一「剝落」之評定等級，據從嚴認定方式評定（即單項分別有 C、D 等級時，從嚴評定為 D 等級）。

2. 無表列 C、D 等級情形者，分項等級評定為 B 等。

3. 符合 B 等，外牆面無設置任何掛物，或依公寓大廈規約統一規範設置標準並落實管理，評定為 A 等。

4. 符合 B 等，且定期清洗外牆，有顯著管理成效者，評定為 A 等。

評估結果

□ A 級（較微）
□ B 級（尚可）
□ C 級（差）
□ D 級（嚴重）

（簽名） （蓋章）

表1-3、臺中市外牆飾面及附掛誤勘檢項目評估表

項目	分項	評定內容		評定等級	評判勾選	細項等級	分項等級
建築物直接面臨建築線或供公眾	外牆飾面	外牆飾面有零星剝落、鼓脹現象		C			
		外牆飾面有剝落、鼓脹現象	4 處以下且面積合計未超過 $5m^2$	D			
			5 處以上或面積合計達 $5m^2$ 以上	E			
		外牆飾面剝落已達危及行人或車輛之虞		E			
		飾面材質種類：					
		□磁磚	零星現象	C			
		□石材	外牆有明顯裂縫、變形、混凝土塊剝落或鋼筋裸露露鑽蝕情形	D			
		□洗（抿）石子	4 處以下未達嚴重程度				
		□水泥粉光	5 處以上或達非常嚴重程度	E			
		□其他	外牆開口部已有明顯變形或龜裂現象	D			
			外牆呈現嚴重滲漏水白華現象	C			

表1-4、高雄市老屋健檢計畫-外牆健檢項目評估表

項目	評定內容	評定等級	評判勾選
外牆飾材安全	外牆面飾材完整且未有剝落或隆起之情形。	A	
	外牆飾材未有範圍性剝落（三塊以內）。（輕度）	B	
	外牆面飾材材料有範圍性剝落或隆起，其範圍超過三塊磁磚，但面積小於 2500 平方公分或多處小範圍馬賽克磚剝落。（中度）	C	
備註	1.外牆有明顯裂縫、混凝土塊剝落或鋼筋裸露露鑽蝕情形。 2.單處剝落（隆起）面積大於 2500 平方公分。 3.三處以上中度危險範圍性剝落。 符合上述任一項者，為 D 等級。（重度）	D	
	1.本表係以目測檢視並以面臨道路之外牆為主。 2.本表之評定等級，採從嚴認定方式評定，即單項分別有 B、C 等級時，則評定為 C 等，其餘以此類推。如有 2 面以上之外牆面臨道路者，則評定等級最高者（較嚴重者）勾選本表。 3.評定為 B 等以上者，請記錄外牆位置與照片並於補充說明欄位說明，如建物（該棟門牌號）臨○○路外牆約 7 樓左側。		

（資料來源：厲娓娓。(2017)。建築物外牆磁磚飾材評估檢查及診斷標準之研究。《建築物外牆飾面材料評估檢查及診斷標準之研究》(頁21-35)。內政部建築研究所。）

從以上文獻分析發現，現在台灣擁有許多老舊建築物外牆有磁磚剝落的現象，又因地震頻繁，嚴重的剝落甚至有造成傷亡，許多縣市也開始設定相關法規，要求各老舊建築物進行檢修，但現在大多仍是採取傳統人工敲擊檢測法，依賴人員高空作業，不僅費時、成本高昂，同時也會產生許多安全風險，特別是在大面積建築外牆檢測時，費用往往高於數十萬元，使得磁磚安全檢修無法得到普及，導致潛在危險未能及時發現並處理。此外，人工檢測受限於技術人員的經驗，難以做到全面、精確的評估，增加許多外牆安全管理的困境。

YOLO (You Only Look Once) 是一種即時物件偵測演算法，能夠在單次前向傳播中同時識別影像中的多個物件並進行分類與定位。它的核心概念是將影像劃分割成網格，並對每個網格預測邊界框與物件類別。相較於傳統的 R-CNN (Region-based CNN) 系列演算法，YOLO 的速度較快，適用於即時辨識，如自動駕駛、監控系統。

其中 YOLOv8採用了更先進的架構和訓練技術，其網路結構主要由 **Backbone** (使用卷積和反卷積層來提取特徵並使用了殘差連接和瓶頸結構來減少網路的大小)、**Neck** (使用多尺度特徵融合，將 backbone 不同階段的特徵圖融合增強特徵表現能力)、**Head** (負責最後的目標檢測和分類任務，分別為檢測頭和分類頭) 組成。

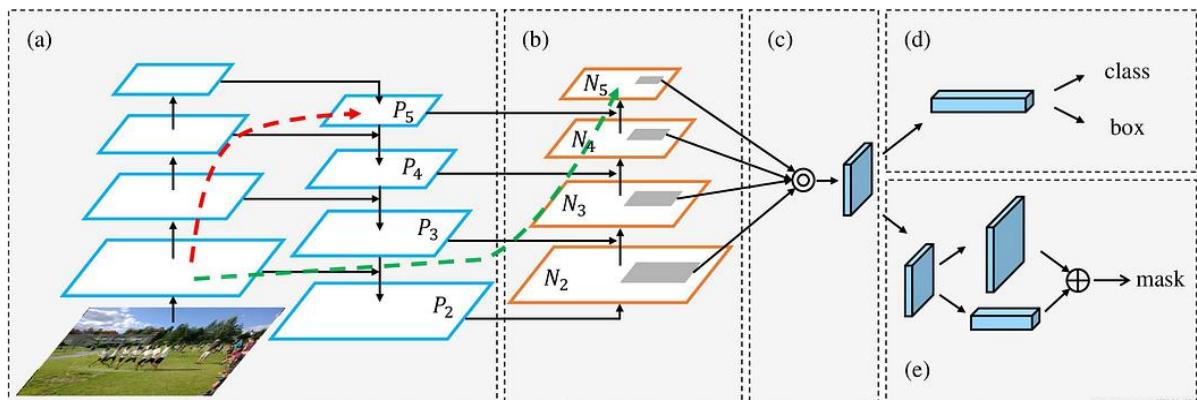


圖 4、YOLOv8 – Neck 階段

(資料來源：泓宇 (2024)，YOLOv8詳解，Medium)

在本研究中，我們選擇 YOLO 作為影像辨識技術，主要原因在於其能夠精確標示建築物外牆上的水泥裸露區域，並提供物件的座標資訊，進一步與紅外線熱成像技術結合，以提高建築磁磚剝落檢測的準確度。YOLO 的應用不僅能夠強化自動化檢測的效能，也可以為建築維護提供更精確的數據支持與風險評估能力，因此我們選用 YOLO，實現即時且準確的外牆剝落辨識，並與熱成像技術結合，以提升磁磚剝落風險評估的精確

性與可行性。

貳、研究設備及器材

表 2、研究設備與器材

名稱	規格	數量
開發板	Raspberry Pi 4 Modle B	1 個
TF卡	32GB	1 張
紅外線雙光雲台相機	S301	1 台
數圖傳模塊	UDVR001	1 塊
地面站遙控器	MK15	1 台
無刷馬達	2216	4
螺旋槳	1047	4
鋰電池	3S 5200mAH	1
無人機機架+腳架	F450	1
無人機遙控器	AT9S	1
無人機遙控接收器	R9DS	1
電子變速器	20A	4
飛控+GPS	APM2.8	1
Pixhawk	Pixhawk 2.4.8	1
AirSim	v1.5.0	1
QGC	v4.2.3	1

(資料來源：由作者自行繪製)

參、研究過程或方法

為了提供更高效、準確，以及降低安全風險的建築物外牆磁磚檢測方案，我們致力於開發一套基於 AI 與無人機技術的建築磁磚剝落風險評估系統，進行磁磚內部溫差判斷以及水泥裸露辨識，應用大數據和機器學習偵測辨識技術自動偵測並標記潛在的剝落區域。因此我們選用樹莓派(Raspberry Pi 4 Model B) 做為開發板，並結合紅外線熱成像鏡頭以及可見光鏡頭，識別建築物外牆的損壞程度，將高機率有磁磚剝落的損壞區域標記在 3D 建模後的建築物模擬模型上，並生成一份風險評估分析報告，協助管理者進行維護決策，再透過長期數據累積進行模型優化，提高系統進行檢測時的準確性與可靠性。

一、研究方法設計

研究開發自動化建築磁磚剝落檢測系統，利用無人機搭載紅外線雙光雲台相機，提供快速且安全的檢測建築外牆。系統中，YOLO 模型識別水泥裸露區域，並以紅外線檢測異常溫度，將資料進行比較，最後，生成風險評估報告搭配可視化的3D 模型圖，提升檢測效率、準確性，並降低安全風險與成本。

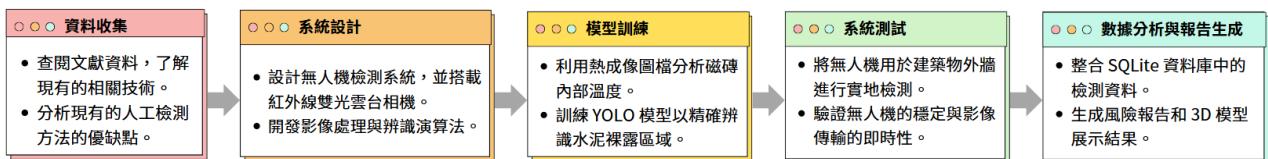


圖 5、研究流程圖 (資料來源：作者自行繪製)

二、無人機設備選用

我們選用四軸無人機與無刷馬達，主要考量其穩定性與較高負載能力，能有效支撐雙光鏡頭並確保飛行穩定，同時我們也在飛行的底下加裝了起落架，以防止底下的攝影鏡頭因降落力道過猛而損壞，如下圖 6。



圖 6、飛行裝置完成圖 (資料來源：由作者自行拍攝)

這款 UDVR001 數圖傳模塊是一款專為無人機、機器人或其他遠程控制應用設計的通訊設備，具有長距離、低延遲、高安全性、低功耗的特點。它能夠支持 WiFi、LoRa、MAVLink、IP 通訊，並配備強大的 ARM Cortex-A7 處理器，適合用於無線影像與數據傳輸應用，例如無人機視訊串流、遠程監控等。



圖7、數圖傳模塊 (資料來源：由作者自行拍攝)

三、數據傳遞

我們利用樹莓派(Raspberry Pi 4 Model B)進行無人機數圖傳的中介站，將攝影機所收集到的數據資料存到 SQLite 資料庫中，再透過疊圖比對後，將數據整合，生成一份風險評估報告。

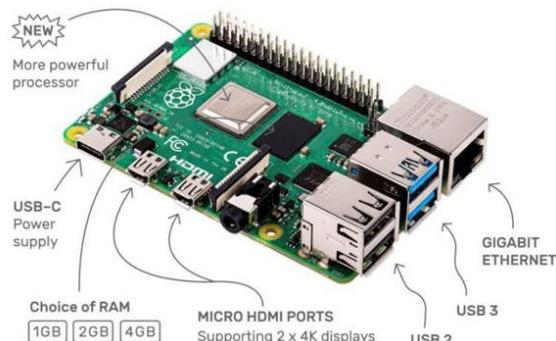


圖 8、樹莓派示意圖 (資料來源：Google)

四、紅外線雙光雲台相機



圖 9、S301_紅外線雙光雲台相機 (資料來源：由作者自行繪製)

（一）紅外線鏡頭

紅外線攝影機（熱成像相機）是一種能夠偵測物體表面溫度的設備，透過捕捉紅外線輻射來生成熱圖像。熱成像相機的核心元件是微測輻射熱計（Microbolometer）或量子探測器，能將不同溫度的物體轉換為影像，其中高溫區域通常顯示為紅色或白色，而低溫區域則為藍色或黑色。

我們利用紅外線鏡頭的特性，針對大樓外牆磁磚進行熱成像偵測，因為磁磚在與水泥牆有縫隙的時候，內部會有空氣存留，會導致其表面溫度較其他磁磚高，基於空鼓的這個理由，我們利用紅外線鏡頭，找到存在掉落危險的磁磚。

（二）可見光鏡頭

可見光鏡頭是捕捉人眼可見光線的裝置，它的原理是利用鏡片組將光線聚焦在感光元件上，感光元件再將光線轉換成電子訊號，最後由影像處理器將電子訊號轉換成我們看到的影像。

有水泥裸露證明建築物外牆的磁磚已經正在剝落，且有繼續擴大面積脫落的風險，因此，我們利用可見光鏡頭搭配物件偵測技術，辨識建築物外牆上的水泥裸露位置。

五、實驗與實際測試

為了驗證紅外線雙光雲台相機系統整合至四軸無人飛行載具後的飛行動力學穩定性，我們進行了實地飛行測試。透過測試去評估無人機載具在實際飛行條件下的性能，特別是關注其是否能夠克服因額外搭載雙光雲台相機所產生的重心變化，確保飛行載具在起飛、懸停及機動飛行等階段的穩定性與可控性，避免因過重而導致的失控風險。



圖 10、實際無人機組裝圖（資料來源：由作者自行拍攝）

我們為了驗證當磁磚空鼓時，經過太陽照射後，磁磚內部會有熱量殘留，具有空鼓的磁磚較其他正常的磁磚熱，因此我們模擬建築物外牆，設計了一套可驗證熱異常的磁磚實驗板，製作四種不同空鼓程度的磁磚，並在充足照射太陽且稍微冷卻後用紅外線鏡頭拍攝並記錄數據。



圖11、實驗磁磚製作 (資料來源：由作者自行拍攝)

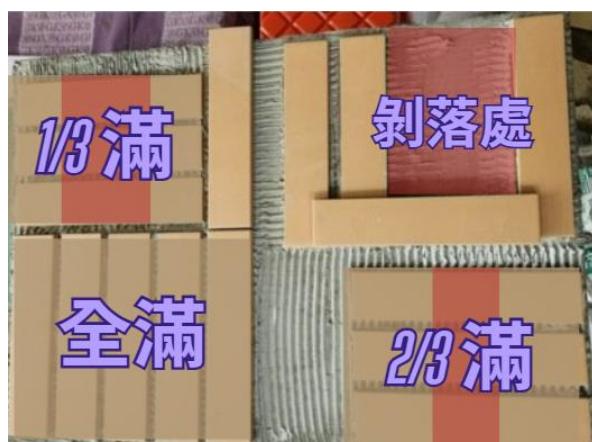


圖12、實驗磁磚成果



圖13、實驗過程 (曬太陽)

(資料來源：由作者自行拍攝)

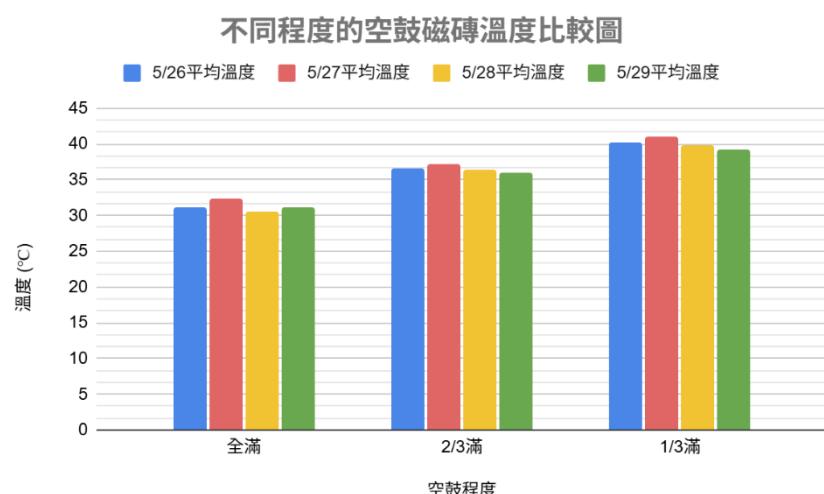


圖 14、不同程度的空鼓磁磚溫度比較圖 (資料來源：由作者自行繪製)

從上圖14可以得知，空鼓程度較高的區域，溫度整體都比較高，因此，從我們的實驗中可以得知當磁磚某部分區域溫度相較其他區域的磁磚來得高時，可能會有空鼓的現象產生，發生剝落的風險較大。

六、物件偵測模型選用：CNN & YOLO

(一) CNN 卷積神經網路

CNN 是一種深度學習架構，主要用於圖像處理和分析，CNN 基本組成包括卷積層、池化層、全連接層。卷積層使用多個卷積核（filters）提取局部特徵，如邊緣、角落和紋理。池化層跟隨在卷積層之後，用於降低特徵圖的維度，減少計算量，同時保留重要信息。全連接層則是在網路的末端，負責綜合所有提取的特徵。

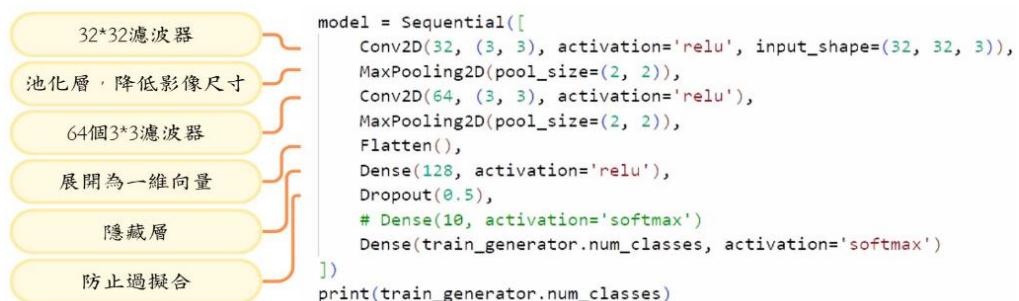


圖 15、CNN 訓練程式圖（資料來源：由作者自行繪製）

```

model.fit(
    train_generator,
    validation_data=val_generator,
    epochs=10 # 訓練 10 個 Epoch
)

Epoch 1/10
8/8 295ms/step - accuracy: 0.7133 - loss: 0.4914 - val_accuracy: 0.9167 - val_loss: 0.2809
Epoch 2/10
8/8 201ms/step - accuracy: 0.9216 - loss: 0.2690 - val_accuracy: 0.9167 - val_loss: 0.2747
Epoch 3/10
8/8 196ms/step - accuracy: 0.9084 - loss: 0.2493 - val_accuracy: 0.9167 - val_loss: 0.2305
Epoch 4/10
8/8 201ms/step - accuracy: 0.9124 - loss: 0.1902 - val_accuracy: 0.9167 - val_loss: 0.2847
Epoch 5/10
8/8 284ms/step - accuracy: 0.8968 - loss: 0.1598 - val_accuracy: 0.9167 - val_loss: 0.3161
Epoch 6/10
8/8 237ms/step - accuracy: 0.9223 - loss: 0.1123 - val_accuracy: 0.7667 - val_loss: 0.4761
Epoch 7/10
8/8 190ms/step - accuracy: 0.9272 - loss: 0.1277 - val_accuracy: 0.8167 - val_loss: 0.4159
Epoch 8/10
8/8 197ms/step - accuracy: 0.9801 - loss: 0.0955 - val_accuracy: 0.7167 - val_loss: 0.4274
Epoch 9/10
8/8 204ms/step - accuracy: 0.9657 - loss: 0.0886 - val_accuracy: 0.7000 - val_loss: 0.4464
Epoch 10/10
8/8 193ms/step - accuracy: 0.9464 - loss: 0.0882 - val_accuracy: 0.4833 - val_loss: 0.6432
<keras.src.callbacks.history.History at 0x7be369f9b410>

```

圖 16、CNN 訓練過程（資料來源：由作者自行繪製）

```

from tensorflow.keras.preprocessing import image
import numpy as np

img_path = "/content/dataset/train/peel/download-19.jpg" # 設定待測的圖片
img = image.load_img(img_path, target_size=(32, 32)) # 調整大小
img_array = image.img_to_array(img) / 255.0 # 轉換為範例要正常化
img_array = np.expand_dims(img_array, axis=0) # 增加 batch 速度

# 進行預測
prediction = model.predict(img_array)

# 計算類別名稱
class_names = list(train_generator.class_indices.keys())
predicted_class = class_names[np.argmax(prediction)]
print("預測結果: " + predicted_class)

```

圖 17、CNN 訓練成果（資料來源：由作者自行繪製）

我們使用 Keras 訓練簡易卷積神經網路 (Convolutional Neural Network, CNN) 模型，用來辨識外牆是否由水泥裸露。我們使用兩層卷池層（分別使用 32 個和 64 個 3×3 濾波器），然後訓練 10 個 epoch，最後去測試 CNN 模型，輸出結果顯示「脫落」(peel)，表明模型成功識別出該區域存在磁磚脫落問題。

(二) YOLO 物件偵測

YOLO 是一種即時的物體檢測系統，主要特點是將物體檢測問題轉成回歸問題。YOLO 用單一的卷積神經網絡來同時預測多個邊界框和類別概率，再將輸入圖像劃分為 $S \times S$ 的網格，每個網格負責預測固定數量的邊界框以及其置信度（該框內是否有物體及準確性），由於 YOLO 將整個檢測過程整合為一個單一模型，因此能夠實現快速檢測，適合實時應用。

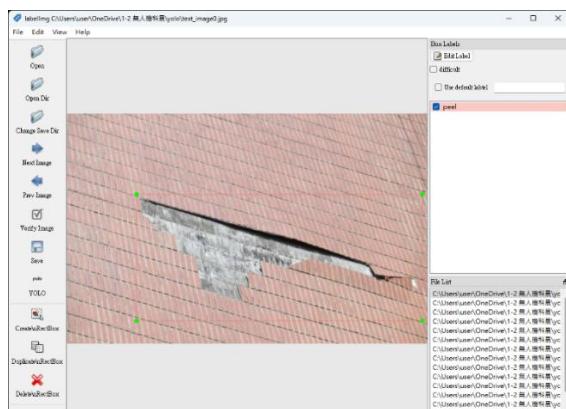


圖 18、LabelImg 框選訓練（資料來源：由作者自行繪製）

LabelImg 是一個開源的工具，有支援 VOC 與 YOLO 格式，可以用於影像手動標記物件的位置與分類，我們用 LabelImg 將 YOLO 模型訓練需要的圖像資料進行標記，框選出有缺陷的區域，再為其指定對應的標籤名稱，例如「peel」來表示磁磚剝落等。完成標記後，LabelImg 會自動將每張圖片的類別編號和圖片中心點與寬高的相對座標資訊儲存，匡列好的資料可直接用於 YOLO 模型的訓練過程中。

```

1  from ultralytics.models import YOLO
2
3  if __name__ == '__main__':
4      model = YOLO("yolov8n")
5      model.train(data='./test-download/data.yaml',
6                  epochs=500,
7                  batch=1,
8                  device='0',
9                  imgsz=640,
10                 workers=0,
11                 optimizer='auto',
12                 lr0=0.001,
13                 lrf=0.1,
14                 momentum=0.9,
15                 freeze=22)
  
```

圖 19、YOLO 模型權重（資料來源：由作者自行繪製）

我們使用 Ultralytics 所提供的 YOLOv8 進行模型訓練，訓練參數如訓練輪數 (epochs=500)、影像尺寸 (imgsz=640)、批次大小 (batch=1)、學習率 (lr0=0.001) 等皆可依據不同需求進行調整。同時我們也設定了鎖定前22層網路參數 (freeze=22)，能使模型在保留原本特徵學習能力的同時，針對我們的特定資料進行微調學習。最後去測試 YOLO 模型，輸出結果成功匡列水泥裸露的區域，我們訓練的模型成功識別出該區域存在水泥裸露問題。



圖 20、YOLO 偵測畫面 (資料來源：由作者自行繪製)

我們最後選用 **YOLO 模型** 來做我們系統的水泥裸露辨識，是因為 CNN 較適用於做分類任務，但我們的系統更需要的是辨識水泥裸露的部分並標示其位置座標，因此我們選擇 YOLO 來做我們的水泥外牆辨識的模型。

七、可視化風險呈現

3D 掃描技術結合了可見光圖像(RGB)和深度圖像(Depth)，使得可以同時獲得顏色信息和場景的三維結構。將 RGB 和深度數據轉換為3D 點雲的基本原理是將深度圖像和 RGB 圖像中的數據結合，將每個像素的深度值轉換為對應的三維坐標，並附加顏色信息。

(一) 轉換深度數據為3D 坐標

給定深度圖像中的像素位置(u, v)和對應的深度值 $d(u, v)$ ，我們可以使用相機的內部參數來計算該像素在3D 空間中的位置(X, Y, Z)。

$$X = (u - cx) \cdot \frac{d(u, v)}{fx} \quad Y = (v - cy) \cdot \frac{d(u, v)}{fy} \quad Z = d(u, v)$$

cx, cy 是相機光心的像素坐標 fx, fy 是相機的焦距

(u, v) 是深度圖像的像素坐標 $d(u, v)$ 是該像素的深度值

(二) 將 RGB 顏色信息附加到3D 點

在深度圖像中，每個像素(u, v)都對應於 RGB 圖像中的一個像素。所以我們可以從 RGB 圖像中提取該像素的顏色信息(R, G, B)，並將其附加到對應的3D 點上。

(三) 組成點雲

每個點 (X, Y, Z) 都會對應一個顏色值(R, G, B)。將所有 3D 點和顏色信息組合在一起，形成點雲。點雲中的每個點都表示3D 空間中的一個位置，並附有該位置的顏色信息。



圖 21、點雲圖效果圖（資料來源：由作者自行繪製）

肆、研究結果與討論

一、系統運作流程圖

經過不斷研究與規劃，我們研究透過無人機進行溫度檢測 (紅外線鏡頭)、裸露水泥辨識 (可見光鏡頭) 及 3D 建模。拍攝完畢，會將影像分割成單張圖片進行分析。系統通過熱成像偵測分析異常溫度 (代表磁磚與水泥之間存在不正常的縫隙) 與 YOLO 影像辨識技術判斷水泥裸露情況 (顯示建築出現大面積損壞)。

所有數據透過樹莓派儲存缺陷數據的座標與時間於 SQLite 資料庫中。最後，整合模組會評估問題的嚴重程度，並產生一份風險評估報告，提供建築檢測與維修的參考資訊，此外，我們還配備 3D 模型，可以更直觀地呈現缺陷位置與範圍。

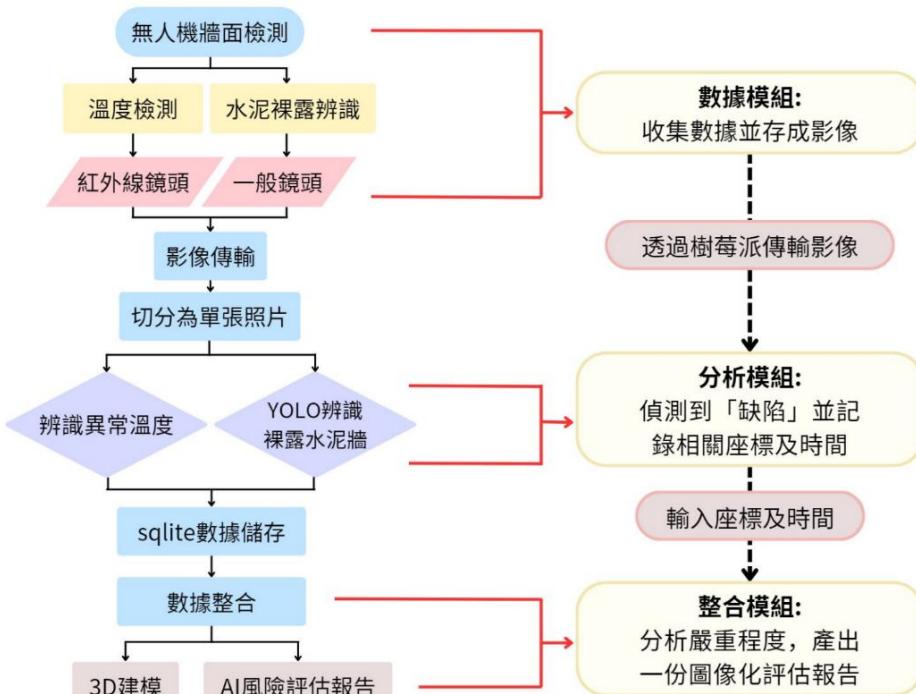


圖22、系統流程圖（資料來源：由作者自行繪製）

二、資料收集 & 模型訓練階段

(一) 紅外線熱顯像檢測

在實驗中，我們設計了兩塊模擬建築物外牆的磁磚板，一組是分成四個不同的區塊，分別代表不同程度的空鼓狀況，另一組則是真實模擬建築物外牆的交錯排列，其中板子紅色部分為空鼓。如下圖 23，由圖片可以得知，有空鼓的區域經過日曬降溫後，內部會有熱空氣殘留，因此溫度較周圍磁磚來的高。我們選擇在12:00 – 16:00進行紅外線的拍攝作業，得以獲得最明顯的溫差與最佳的檢測效果。

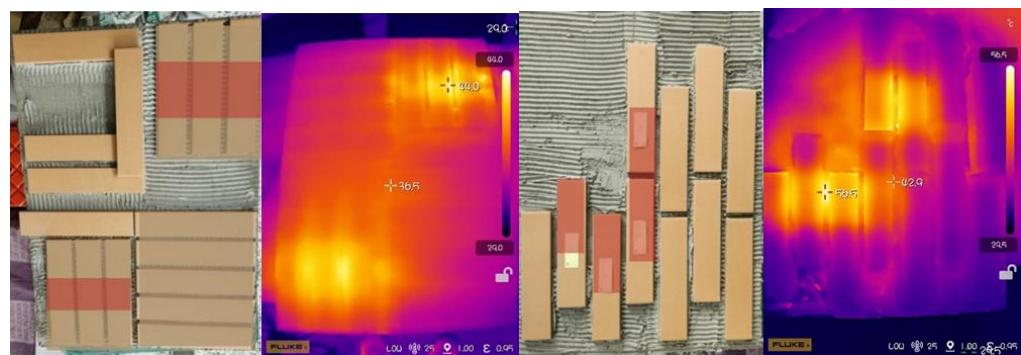


圖 23、磁磚空鼓熱成像畫面（資料來源：由作者自行拍攝）

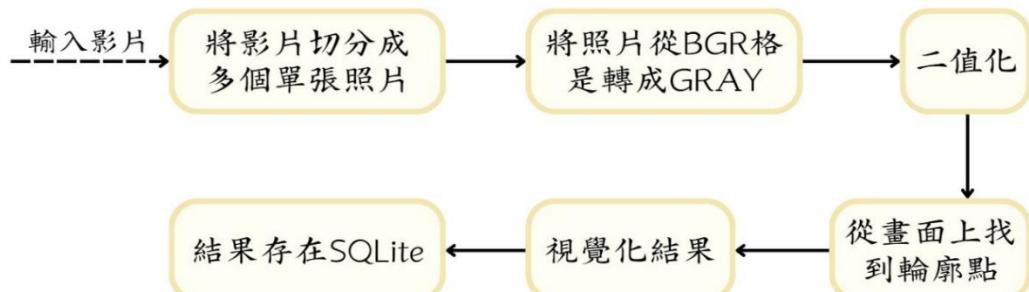


圖 24、紅外線熱顯像檢測流程圖（資料來源：由作者自行繪製）

我們輸入影像將其切分成多張單獨的圖片，接著這些圖片從 BGR 格式轉換成灰階 (GRAY)。接著將圖片進行二值化處理，強調對比並去除不必要的細節，再透過二值化影像，系統能夠從畫面上找到輪廓點，用於識別溫度過高磁磚的特徵點。最後，系統會將偵測結果儲存至 SQLite 資料庫，並提供視覺化結果。

```
video_path = 'tokyo_walking-2024-10-31_16.17.41.mp4'
cap = cv2.VideoCapture(video_path)

if not cap.isOpened():
    print("Error: Could not open the video file!")
    exit()

fps = cap.get(cv2.CAP_PROP_FPS)
frame_count = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
```

圖 25、紅外線檢測部分程式圖 – 1 (資料來源：由作者自行繪製)

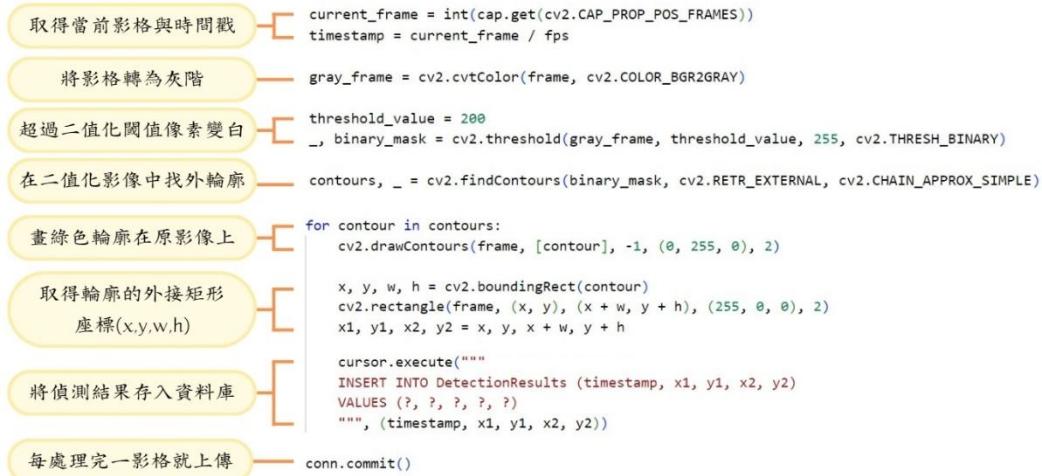


圖 26、紅外線檢測部分程式圖 - 2 (資料來源：由作者自行繪製)

在圖 26 中，我們是使用閾值（設為 200）進行處理，將超過該值的像素變為白色，把其輪廓畫在原始影像上。後來發現磁磚溫度容易受到自然環境的影響，因此我們將閾值調整為「相對溫差」，依據同一區域內磁磚之間的差異自動調整閾值，如下圖 27，去判斷磁磚是否存在空鼓現象。最後每個輪廓的邊界框座標 (x_1, y_1, x_2, y_2) 會被計算出來，最後，偵測結果會存入資料庫（SQLite），並在處理完每幀影像後進行上傳，如下圖 28。

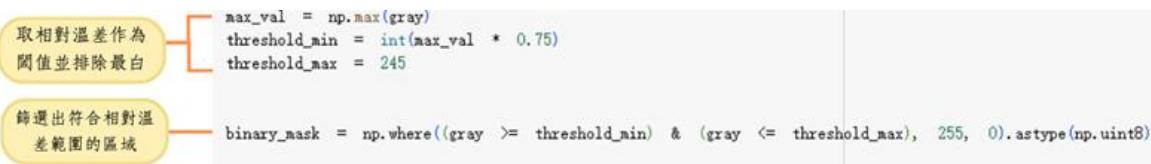


圖 27、紅外線程式圖修正 (資料來源：由作者自行繪製)

	timestamp	x1	y1	x2	y2	Temperature
1	0.03332749766573296	279	1079	280	1080	203
2	0.03332749766573296	282	1078	284	1080	228.66666666666666
3	0.03332749766573296	1085	1072	1086	1073	204
4	0.03332749766573296	147	1070	148	1071	201
5	0.03332749766573296	296	1062	297	1063	202
6	0.03332749766573296	274	1061	275	1062	210
7	0.03332749766573296	241	1055	242	1057	220.5
8	0.03332749766573296	247	1053	251	1056	225.5
9	0.03332749766573296	255	1051	256	1052	215
10	0.03332749766573296	1175	1041	1176	1042	201
11	0.03332749766573296	1153	1041	1154	1042	203
12	0.03332749766573296	230	1032	231	1033	213
13	0.03332749766573296	1067	1020	1068	1021	205

圖 28、IR SQLite 實際資料庫畫面 (資料來源：由作者自行繪製)

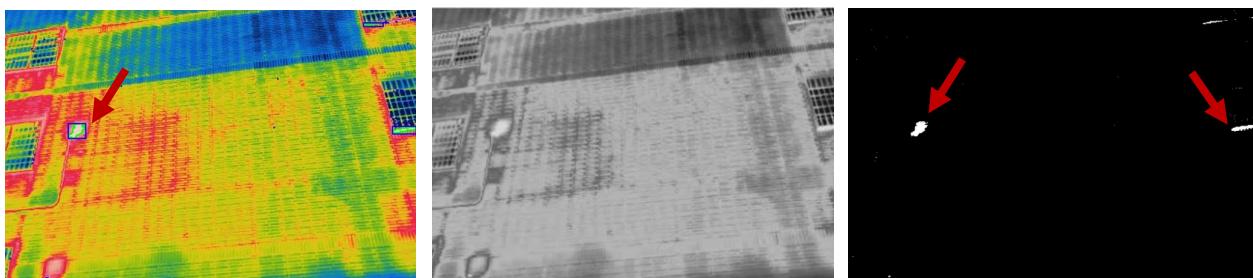


圖 29、紅外線檢測匡列示意圖 (資料來源：由作者自行繪製)

磁磚空鼓現象是由於內部充滿空氣，導致其熱傳導性會遠低於周圍結構材料，導致影響了熱量的傳遞與分佈。在日照或環境溫度變化的影響下，空鼓區域的升溫或降溫速率與周圍正常的磁磚不同，使其產生明顯的溫差。紅外線熱成像技術能夠準確捕捉這些溫差，再將其轉化為可視化的熱圖，協助進行異常區域的辨識。

紅外線影像經過前處理後，會轉換為灰階影像（像素值範圍0~255），為了自動識別可能存在空鼓的區域，系統設定了一個相對門檻值（假設是最高值的 75%），系統能夠篩選出亮度異常偏高的區域，進行二值化處理，並將這些區域自動框選出來。藉由這種方法，可以快速選出可能存在磁磚空鼓的區域，進一步進行精細分析或維修評估，如上圖 29。

(二) 水泥牆裸露辨識

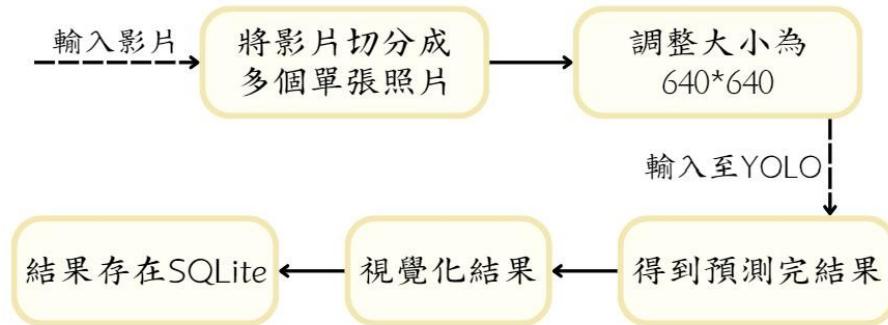


圖 30、執行 YOLO 流程圖（資料來源：由作者自行繪製）

我們首先將輸入影像拆分為多張圖片，接著將其大小調整為640x640，調整後的圖片輸入到 YOLO 模型，進行物件偵測或分類，然後取得預測結果，再將結果視覺化，存到 SQLite 資料庫。

首先先載入訓練好的 YOLO 檔案，利用 OpenCV 讀取可見光鏡頭所拍攝的影片並取得影片的 FPS (Frames Per Second，每秒幀數)，之後進行物件偵測。

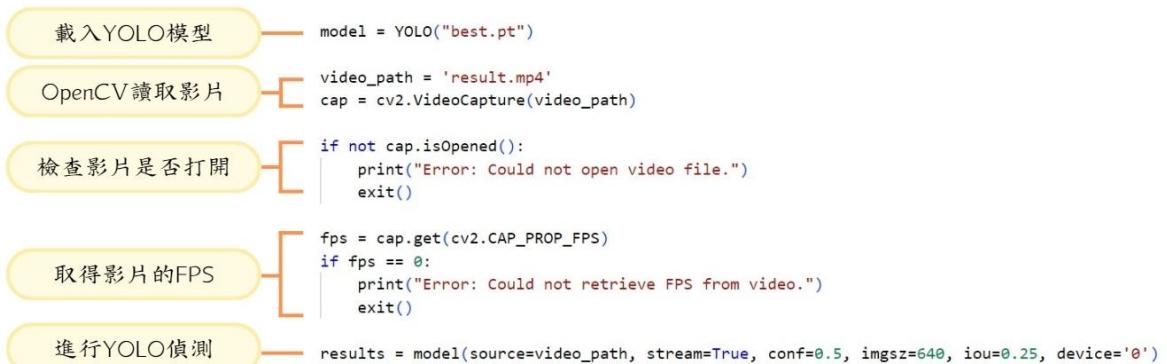


圖 31、水泥辨識部分程式圖 - 1（資料來源：由作者自行繪製）

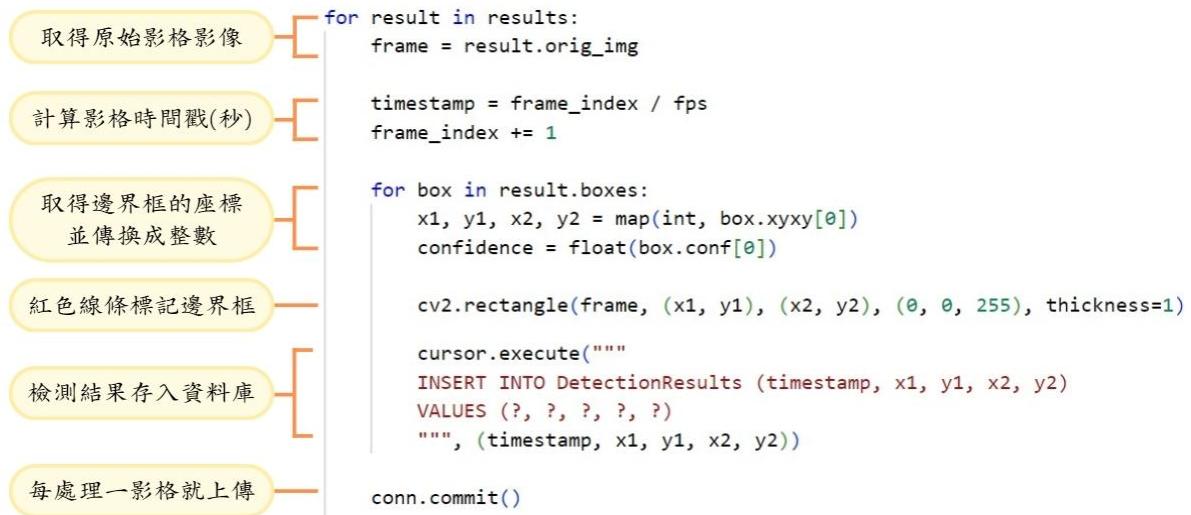


圖 32、水泥辨識部分程式圖 – 2 (資料來源：由作者自行繪製)

從 YOLO 模型的結果中取得偵測到的物件邊界框座標，並將其轉換為整數，在原始影格影像上，則是以紅色線條標記出偵測到的物件邊界框，最後，將偵測到的物件資訊 (時間、邊界框座標) 存入資料庫中。

	timestamp	x1	y1	x2	y2
1	0.03331557449831291	1081	395	1082	396
2	0.03331557449831291	908	398	909	399
3	0.03331557449831291	1063	443	1064	444
4	0.03331557449831291	434	0	1049	321
5	0.03331557449831291	1072	413	1156	433
6	0.03331557449831291	1383	881	1384	882
7	0.03331557449831291	946	540	947	541
8	0.03331557449831291	696	767	697	768
9	0.03331557449831291	1014	365	1056	376
10	0.03331557449831291	391	0	420	34
11	0.03331557449831291	1248	360	1249	361
12	0.03331557449831291	415	331	416	332
13	0.03331557449831291	475	559	497	571
..	0.03331557449831291	586	580	588	580

圖 33、YOLO SQLite 資料庫 (資料來源：由作者自行繪製)

裸露的水泥牆通常是建築物外牆磁磚發生更多剝落或損壞的徵兆，一般鏡頭拍攝的影像在經過我們訓練的 YOLO 模型偵測後，可以即時的標示出裸露的水泥牆區域，這些資訊也更好的警示我們那塊區域存在的安全隱憂。

三、數據判斷

在我們的分析報告中，我們將紅外線以及 YOLO 所偵測到的數據，進行疊合，其中疊合的部分我們是使用 IOU (Intersection over Union) 作為我們檢測的標準。IoU 是一個測量特定資料集中檢測物體準確度的一種標準，可以應用在許多不同的物件檢測方面中，像是物件偵測或是圖片分割甚至是語意分割模型，IoU 都被作為是其中一種檢測的方法。

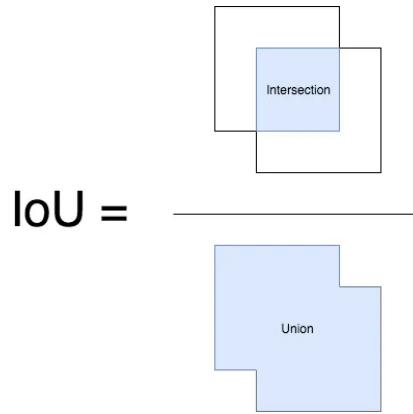


圖 34、IOU 定義圖（資料來源：Google）

交集(Intersection)是預測框與真實框的重疊區域。

聯集(Union)是預測框和真實框的總面積（即兩者面積之和減去交集面積）。

將座標和時間等資料存儲於資料庫中，並根據需求從資料庫中提取相關數據進行後續處理。運用 IoU (Intersection over Union) 面積計算方法，對比不同區域的重疊情況，並生成分析結果，將其轉化為 HTML 格式的評估報告網頁，方便用戶查看和分析。

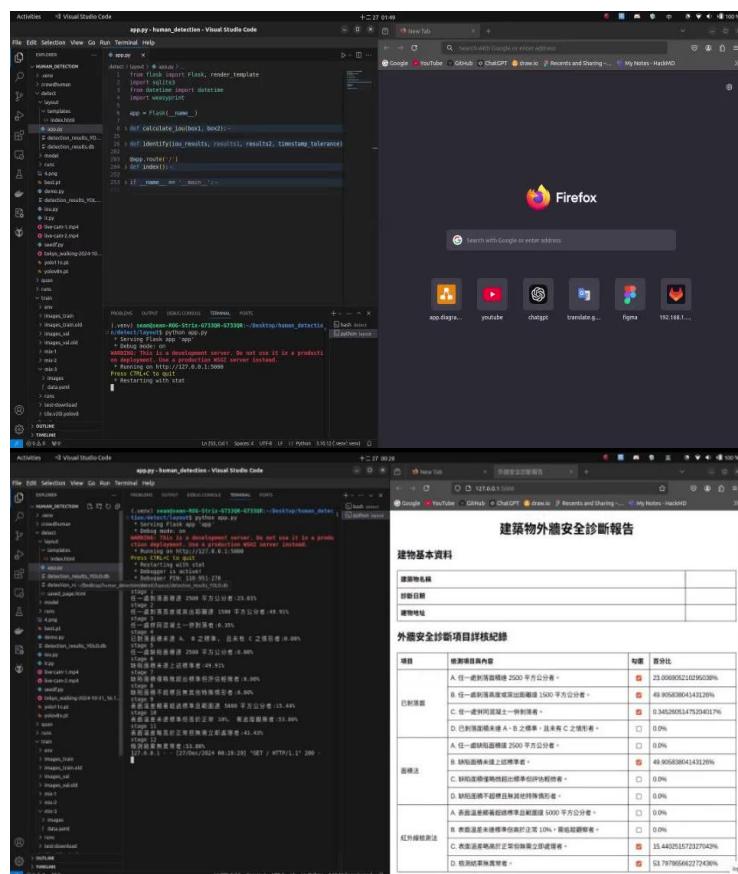


圖 35、AI 評估報告生成過程（資料來源：由作者自行繪製）

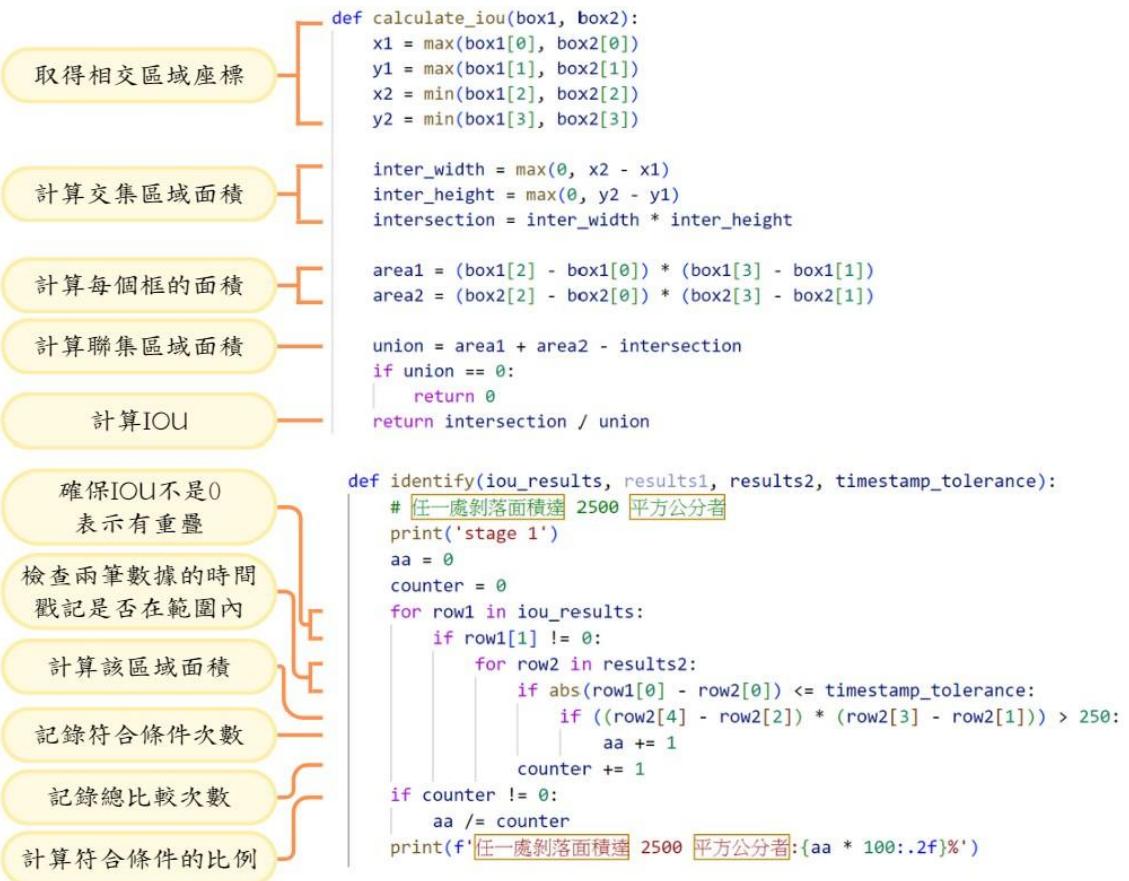


圖 36、評估報告程式碼 (資料來源：由作者自行繪製)

這個程式用來計算 IOU 和識別並計算符合條件的比例，剛開始先從資料庫中取得交區域的座標，再計算其 IoU，優先選擇 IoU 不為0的資料，然後檢查兩筆數據的時間戳記是否在範圍內，並計算面積，選擇任一處超過2500平方公分的剝落面積，最後記錄符合條件的數據，符合條件的比例。最後將數據全部統整，並生成風險評估報告，如下圖 37。

當 IOU 為 0 時，則是表示磁磚空鼓區域與水泥牆裸露區域沒有重合，但這並不代表沒有危險，因此我們在風險評估報告以及可視化的 3D 模型中，加入磁磚空鼓區域和水泥裸露區域單獨的數據呈現，更仔細的去記載建築物所有可能造成磁磚剝落風險的區域，提高系統在呈現數據時的準確度。

外牆安全診斷項目詳細紀錄列出診斷的各個項目，包含「已剝落」、「面積法」和「紅外線異常區域」三大類，每個項目都有：

- 檢測項目內容：具體檢查的條件
- 勾選框：是否符合該項目
- 百分比 (%)：該情況發生的比例

建築物外牆安全診斷報告

建物基本資料

建物名稱	新光大樓
診斷日期	2025-02-19 00:07:26
建物地址	台北市信義區光復南路123號

外牆安全診斷項目詳核紀錄

項目	檢測項目與內容	勾選	百分比
已剝落面	A. 任一處剝落面積達 2500 平方公分者。	<input type="checkbox"/>	1.4%
	B. 任一處剝落高度或突出距離達 1500 平方公分者。	<input type="checkbox"/>	1.77%
	C. 任一處併同混凝土一併剝落者。	<input type="checkbox"/>	0.35%
	D. 已剝落面積未達 A、B 之標準，且未有 C 之情形者。	<input type="checkbox"/>	99.65%
面積法	A. 任一處缺陷面積達 2500 平方公分者。	<input type="checkbox"/>	1.4%
	B. 缺陷面積未達上述標準者。	<input checked="" type="checkbox"/>	48.51%
	C. 缺陷面積僅略微超出標準但評估輕微者。	<input type="checkbox"/>	0.63%
	D. 缺陷面積不超标且無其他特殊情形者。	<input checked="" type="checkbox"/>	47.24%
紅外線檢測法	A. 表面溫差顯著超過標準且範圍達 5000 平方公分者。	<input type="checkbox"/>	0.0%
	B. 表面溫差未達標準但高於正常 10%，需追蹤觀察者。	<input checked="" type="checkbox"/>	27.24%
	C. 表面溫差略高於正常但無需立即處理者。	<input checked="" type="checkbox"/>	24.59%
	D. 檢測結果無異常者。	<input checked="" type="checkbox"/>	46.32%

檢測結論

在檢測過程中，可見光檢測透過 YOLO 模型分析後，未發現明顯缺陷。然而，紅外線檢測顯示多處潛在缺陷，建議進一步追蹤與處理以確保建築物安全性。

報告生成時間：2025-02-19 00:07:26

圖 37、風險評估報告 (資料來源：由作者自行繪製)

這份報告是我們收集了各縣市的檢測報告，選擇其中重要的幾個共同點，再搭配上我們系統多元的功能，提供了一棟建築外牆的安全診斷標準，其中包括已剝落區域、面積法、紅外線異常區域，報告的標準是由不同縣市比對而來的，主要關注在外牆是否即將要剝落或是擴大剝落的危險，透過 AI 來自動偵測與分析外牆狀況。

四、3D 建模



圖 38、3D 建模部分流程圖 (資料來源：由作者自行繪製)

我們使用官方提供的 SDK 來驅動 Astra S 鏡頭，確保設備可以正常運行，再透過 RViz 進行即時視覺化，檢查攝影機捕捉的畫面並驗證數據輸出是否符合預期。

再來，我們利用 SLAM 技術來融合可見光與深度鏡頭的數據，以實現高精度的環境建模。先透過深度鏡頭獲取場景的三維結構資訊，並同步使用可見光鏡頭進行紋理捕捉，得以提高點雲的細節與真實感。將這些數據輸入至 SLAM 系統，透過特徵匹配與位姿估計來確保點雲的準確性與連續性後，利用點雲處理技術（如 Voxel Grid 濾波與 ICP 配準），生成精細且一致的 3D 點雲地圖。

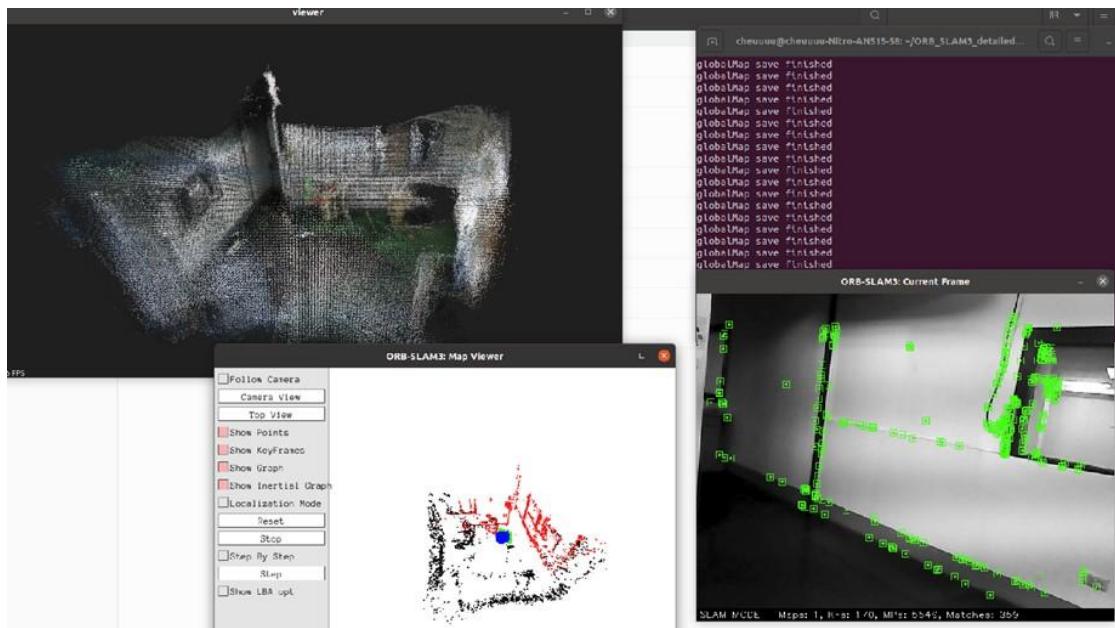


圖 39、點雲圖 (資料來源：由作者自行繪製)

獲取 PLY 格式的點雲圖後，為了將其轉換為可在 Godot 中使用的 3D 模型，我們需要進行網格化處理。先透過三角網格重建（如 Poisson Surface Reconstruction 或 Delaunay Triangulation），把離散的點雲轉換為連續的 3D 表面模型，對模型進行簡化與優化，減少不必要的面數並提升渲染效率。

之後，將處理後的 3D 模型匯出為 FBX 格式，導入至 Godot，引擎內部可進一步調整材質與燈光設定，以達成最佳視覺效果，再加上我們所評估的風險區域，形成下圖 40，標記紅色的地方為缺陷處。

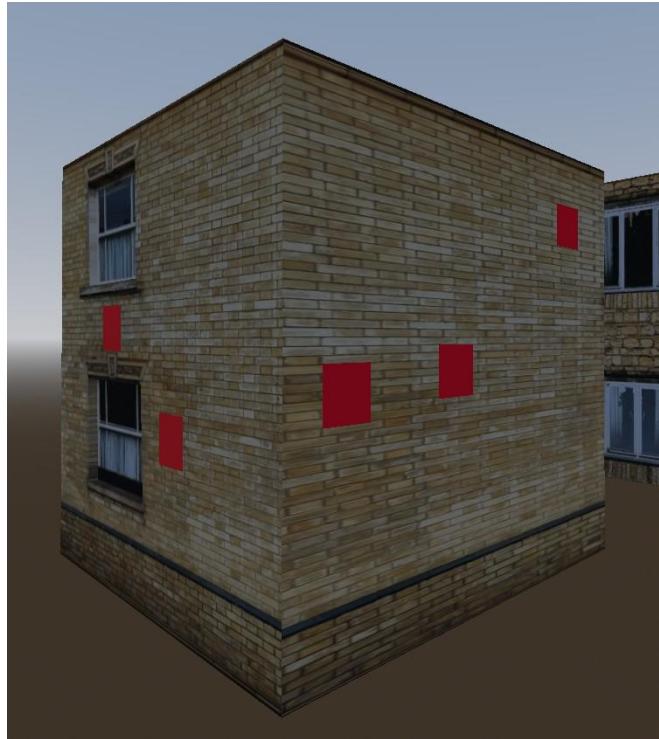


圖 40、3D 建模模擬圖（資料來源：由作者自行繪製）

在研究的初期，我們是嘗試使用結合 RGB 與深度資訊的深度相機來進行近距離獨棟的3D 建模，藉由無人機逐步掃描建物外牆來建模。然而我們發現深度鏡頭存在許多的不穩定性，像是容易受到戶外的光線干擾，導致深度數據出現錯誤，同時深度鏡頭的建模範圍較小，當遇到太大面積的外牆或高樓，需要大量的時間掃描，出來的效果也不一定符合預期，效率不高。

為了精進我們的系統，我們希望可以做到更完善的可視化成果，甚至是一次可以呈現一整個社區，因此我們使用大疆無人機配合 PIX4Dcapture app 進行航線規劃與自動拍攝。我們到可航拍區域進行操作，透過 PIX4Dcapture 事先設定飛行高度、重疊率、拍攝範圍與路徑，無人機就可以依照路徑自動飛行並拍攝地面影像。這個過程能夠確保拍攝到高品質、重疊率足夠的航拍照片，利於後續的3D 建模。



圖 41、PIX4Dcapture 操作畫面（資料來源：由作者自行繪製）

當拍攝完成後，將所有影像匯入至 PIX4Dmapper 軟體中進行處理，PIX4Dmapper 利用影像對位（Structure from Motion, SfM）技術進行點雲生成、網格重建與紋理貼圖，最後輸出3D 模型。這個方法不但可以提升建模效率，也可以讓建物的外觀和結構特徵在3D 模型中真實重現。

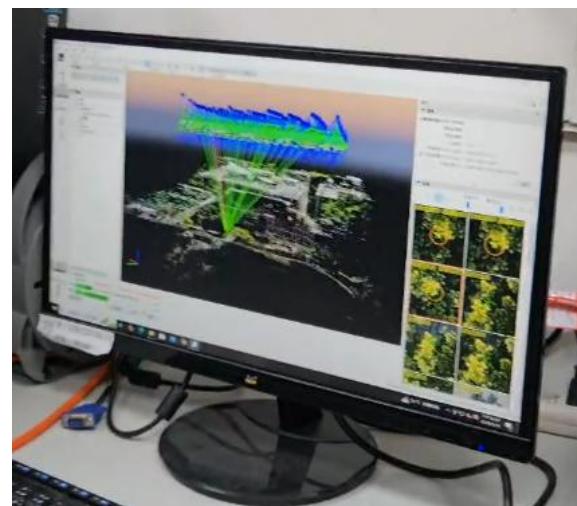


圖 42、PIX4Dmapper 建模畫面（資料來源：由作者自行繪製）



圖 43、3D 模型呈現（資料來源：由作者自行繪製）

為了加入磁磚空鼓區域和水泥裸露區域單獨的數據呈現，我們決定在模型上分成兩種標記方式，當空鼓和裸露區域重疊時標記為紅色區域，若兩者並沒有重疊，將以紅色區域作為標記。



圖 44、3D 模型可視化呈現（資料來源：由作者自行繪製）

五、系統整合

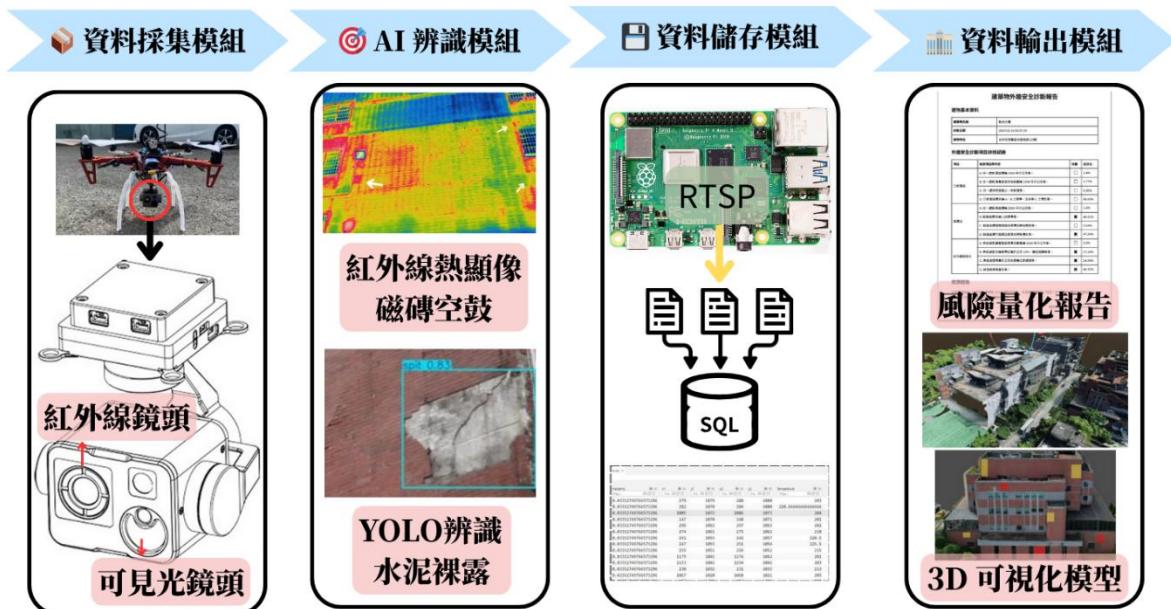


圖 45、系統整合概念圖（資料來源：由作者自行繪製）

我們的系統以模組化方式運作，無人機搭載雙光鏡頭拍攝建築外牆，回傳資料將透過自行訓練的紅外線進行磁磚空鼓辨識，YOLO 模型進行水泥裸露偵測，再將儲存於 SQLite 的資料疊合後生成 3D 可視化模型與風險報告。整體架構具備彈性擴充，可應用於多棟建築長期監測與維護決策，提升檢測效率與實用性。

伍、結論

一、高效自動化檢測

以一般人工檢測而言，一面建築物需約35名工人搭配吊車作業，耗時約達5-10天，平均每天可處理面積僅 $300 - 400\text{m}^2$ 。我們的系統使用無人機約可在1.5小時內完成同樣區域的影像蒐集與初步辨識，預估整體檢測時間縮短至原本的1/30，檢測的成本也可降低至10萬元以下，約為原系統成本的1/10。

二、AI 影像辨識

結合紅外線熱成像技術與 YOLO 物件偵測模型，對牆面進行異常檢測，包括溫度異常區域（分析潛在的空鼓區域或結構問題）與水泥裸露區域（磁磚剝落或老化跡象）。最後將紅外線與可見光影像資訊，進行數據融合與 IoU 疊合技術，提升風險評估準確度。

三、可視化與數據異常決策

進行數據整合與分析之後，系統可自動生成可視化影像，包括缺陷座標、時間標記與嚴重程度標註，自動生成詳細的 AI 風險評估報告，量化缺陷嚴重程度，並建立警示機制，加上3D 建模技術，建立建築外牆點雲模型，將檢測結果視覺化，使檢測結果更直觀易讀，方便後續維修團隊進行處理與決策。

四、預防性風險評估

所有檢測數據存入雲端或本地資料庫，透過維護計劃與長期監測，建立智慧化預測性機制，協助制定預防性維護策略，降低建築物損害風險。

我們的系統使用無人機載具搭配紅外線熱成像技術來做磁磚空鼓的檢測，透過 YOLO 模型來檢測外牆磁磚剝落導致的水泥裸露，我們可實現在建築物外牆的智能巡檢，自動識別潛在風險區域，並進行標記與可視化分析。透過數據的分析與機器學習演算法，系統能預測維護需求，還可以 3D 建模以及生成風險評估報告，幫助使用者做出更好的決策。這項技術應用建築檢測，可以提升檢測效率與準確性，同時降低維修成本與管理負擔，進一步提升建築安全與檢測技術的普及率。

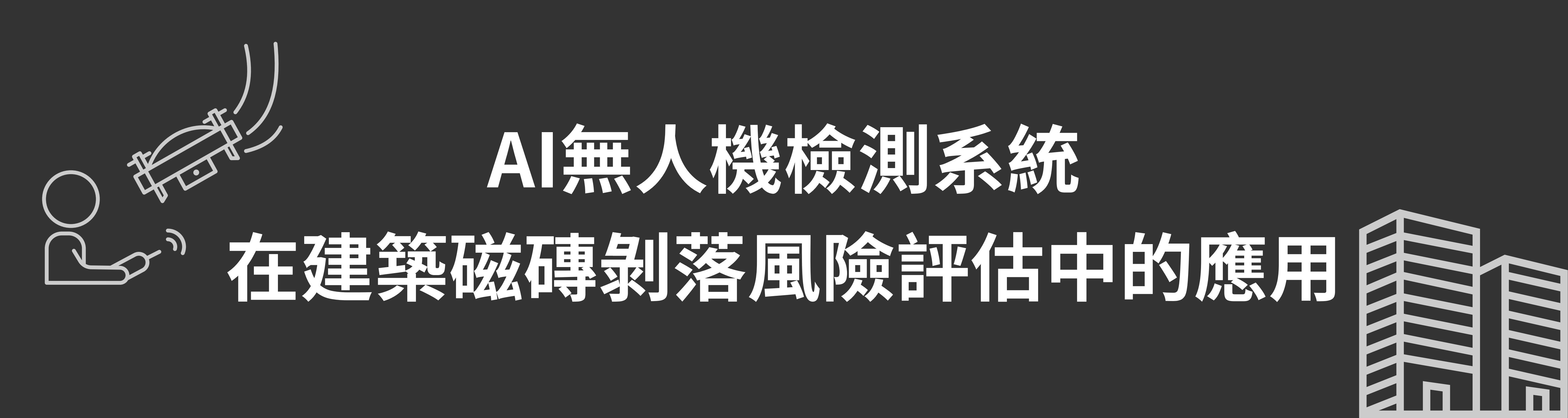
陸、參考文獻資料

- [1] 馬揚異 (2023) , 全台平均屋齡33年，奪命磁磚雨10年後未改善，今周刊1399期，
<https://www.businesstoday.com.tw/article/category/183027/post/202310110029/>
- [2] 吳淑青 (2022) , 公寓大廈外牆維護問題研析，立法院議題研析，
<https://www.ly.gov.tw/Pages/Detail.aspx?nodeid=6590&pid=225597>
- [3] 郭家安 (2020) , 北市外牆安檢參考價價差逾20倍，自由時報，
<https://news.ltn.com.tw/news/life/breakingnews/3326817>
- [4] 北區中心 (2024) , 於施工架上從事外牆補磚作業發生墜落災害致死職業災案，勞動部職業安全衛生署，<https://www.osha.gov.tw/48110/48417/48427/173230/post>
- [5] 盧珽瑞 (2010) , 高層集合住宅外牆磁磚剝落原因與解決對策探討 (一) , 內政部建築研究所工程技術組，
https://www.abri.gov.tw/News_Content_Table.aspx?n=807&s=38815
- [6] 泓宇 (2024) , 用 YOLOv8訓練自己的資料集+詳解，Medium，<https://reurl.cc/26Rbl6>
- [7] 泓宇 (2024) , YOLOv8詳解，Medium，<https://reurl.cc/Q5K7D9>
- [8] Xiang (2024) , 物件檢測的標準 IoU(Intersection over Union) , IT 邦，
<https://ithelp.ithome.com.tw/m/articles/10350081>
- [9] Brandon (2016) , 卷積神經網路的運作原理 , Data Science and Robots ,
<https://brohrer.mcknote.com/zhHant/how machine learning works/how convolutional neural networks work.html>
- [10] 有好多問題 (2024) , 揭秘3D模型：從原理到應用 , 智能雲 ,
<https://cloud.baidu.com/article/3266247>
- [11] Realibox3D (2024) , 3D設計的基本原理和技術 , SOHU ,
https://www.sohu.com/a/668852130_121668539
- [12] 霖思科技 (2024) , 3D點雲圖 , LINX ,
<https://www.linx.tw/tech-info/technique/1216/2024-12->

【評語】052313

1. 作品整合無人機、紅外線熱顯像技術與 YOLO 模型，建置一建築物外牆磁磚剝落檢測系統，具實用價值。
2. 建議加強文獻探討，例如應用溫度曲線的降溫特徵值擬合、熱傳導係數比較，以提升對磁磚黏結層老化程度或界面缺陷的量化判讀能力，以彰顯本作品之效益。
3. 作品同時整合了 3D 建模技術，以重建大樓外觀，並結合風險評估報告，可標記出具有磁磚剝落風險的區域，可以提升檢測效率與準確性。
4. 建議實際應用於不同老舊建築外牆測試，與目前建物檢測方式來比對驗證檢測結果的準確性。

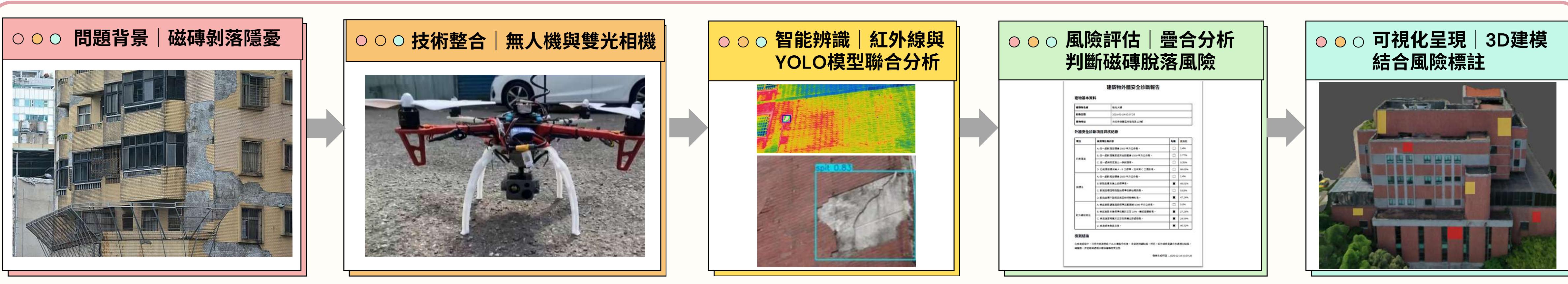
作品海報



AI無人機檢測系統

在建築磁磚剝落風險評估中的應用

摘要



壹、研究動機

一、台灣老舊建築磁磚剝落問題

年久失修造成磁磚剝落問題嚴重，構成公共安全威脅。

二、現行檢測方法的侷限

傳統人敲擊需高空作業，安全風險高且費時、高成本。

三、缺乏系統化資料紀錄

缺乏紀錄歷史比對，不利於長期追蹤外牆健康狀況。

貳、研究目的

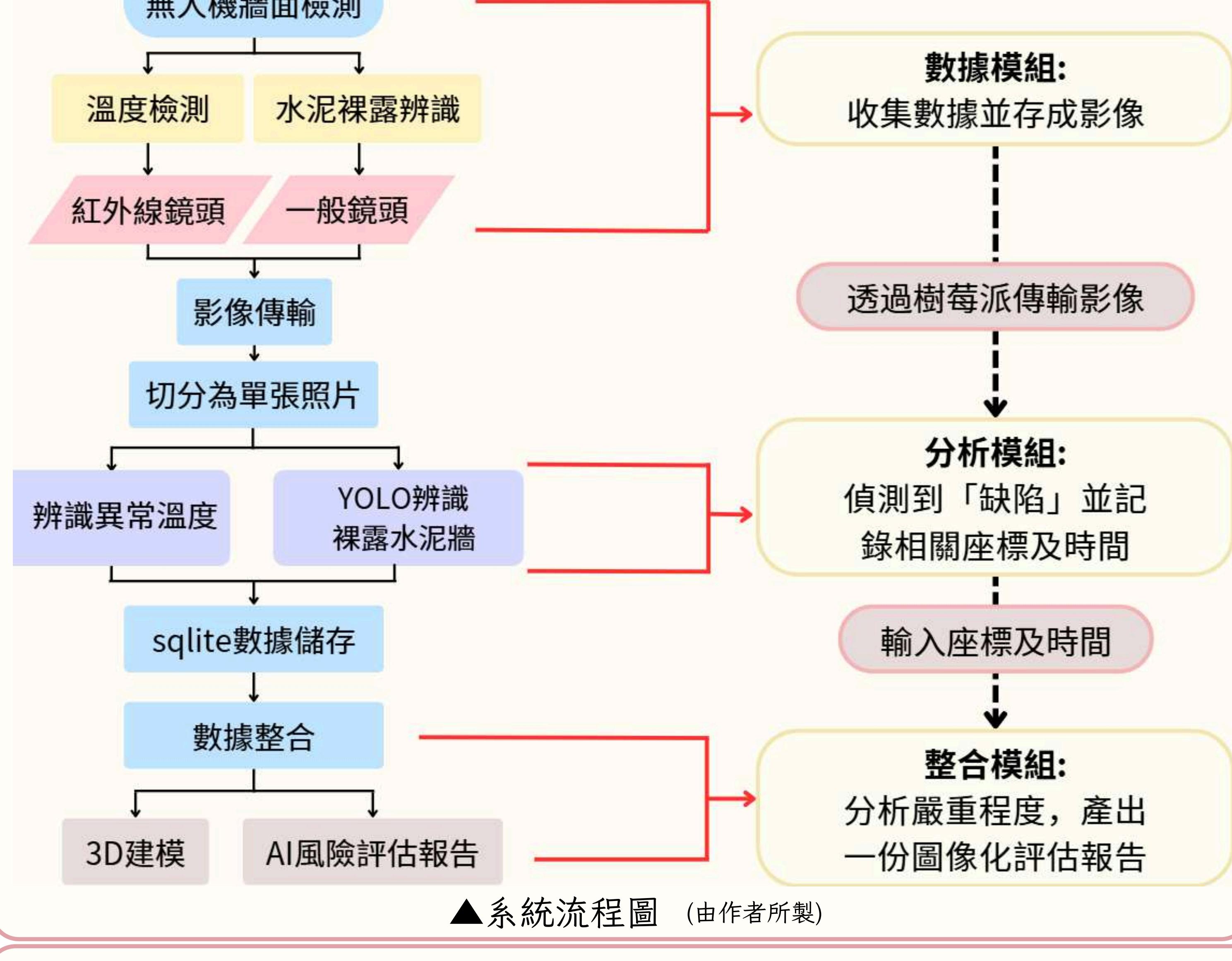
一、開發無人機檢測系統：結合無人機、紅外線與影像辨識技術，提升外牆檢測自動化程度。

二、提升檢測準確性與效率：運用機器學習技術，提高磁磚剝落與結構異常的識別精度與速度。

三、建構可視化維護輔助：結合 3D 建模，直觀呈現檢測結果，方便後續觀察及維修決策。

參、研究過程與結果

一、系統架構



▲系統流程圖 (由作者所製)

二、設備使用

我們選用樹莓派做為開發板，利用紅外線鏡頭的特性，針對大樓磁磚外牆進行熱成像的空鼓偵測，空鼓意指磁磚在與水泥牆有縫隙的時候，內部會有空氣存留，導致其表面溫度較其他磁磚高，在有水泥牆裸露的地方證明外牆的磁磚已經正在剝落，有繼續擴大的風險，因此可利用可見光鏡頭搭配物件偵測，辨識建築物外牆上的水泥裸露位置。



▲裝置示意圖 (由作者所攝)



(由作者所製)

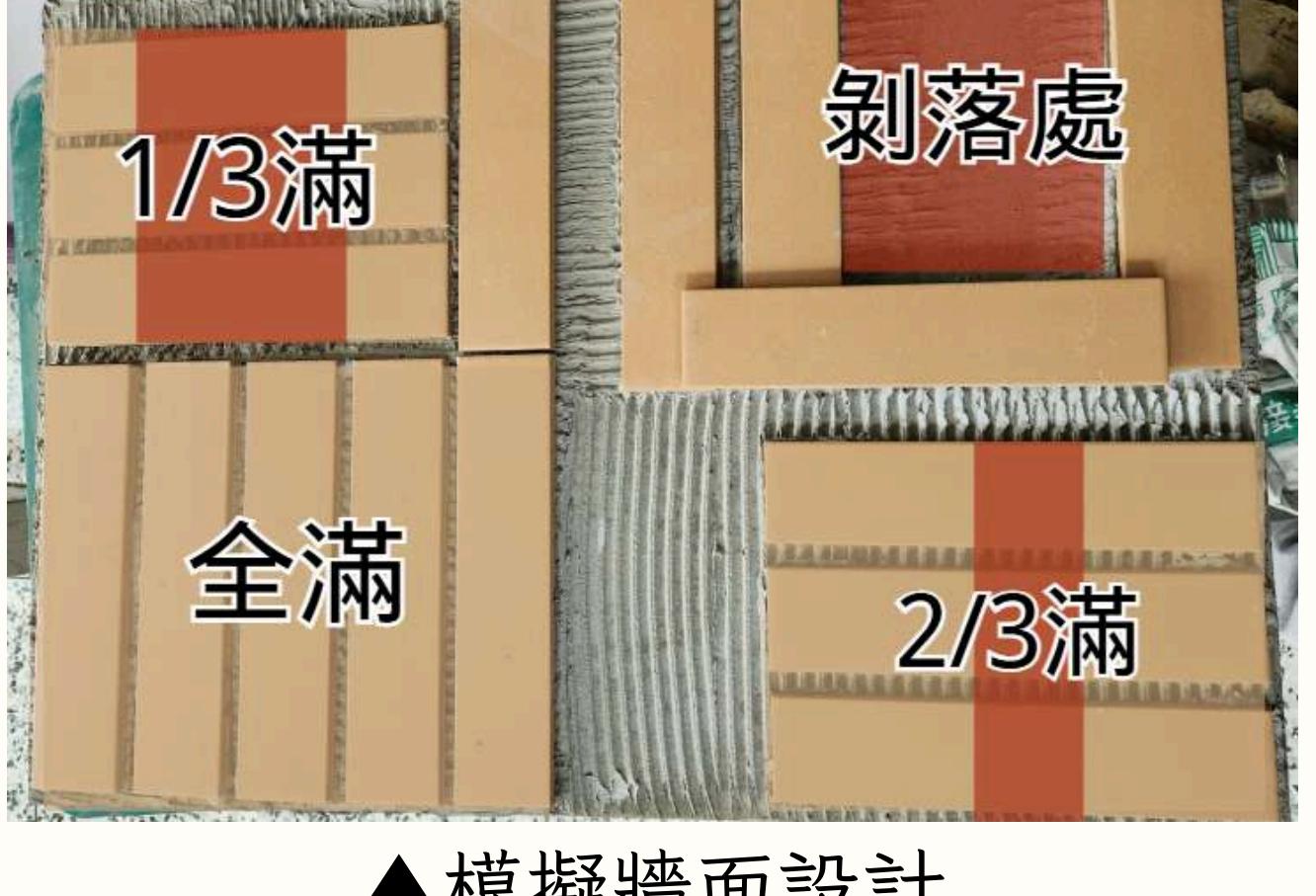
三、實驗過程

當磁磚空鼓時，經過太陽照射後，磁磚內部會有熱量殘留，具有空鼓的磁磚較其他正常的磁磚熱，因此我們模擬建築物外牆，設計了一套可驗證熱異常的磁磚實驗板，製作四種不同程度空鼓程度的磁磚，並在充足照射太陽且稍微冷卻後用紅外線鏡頭拍攝並記錄數據。

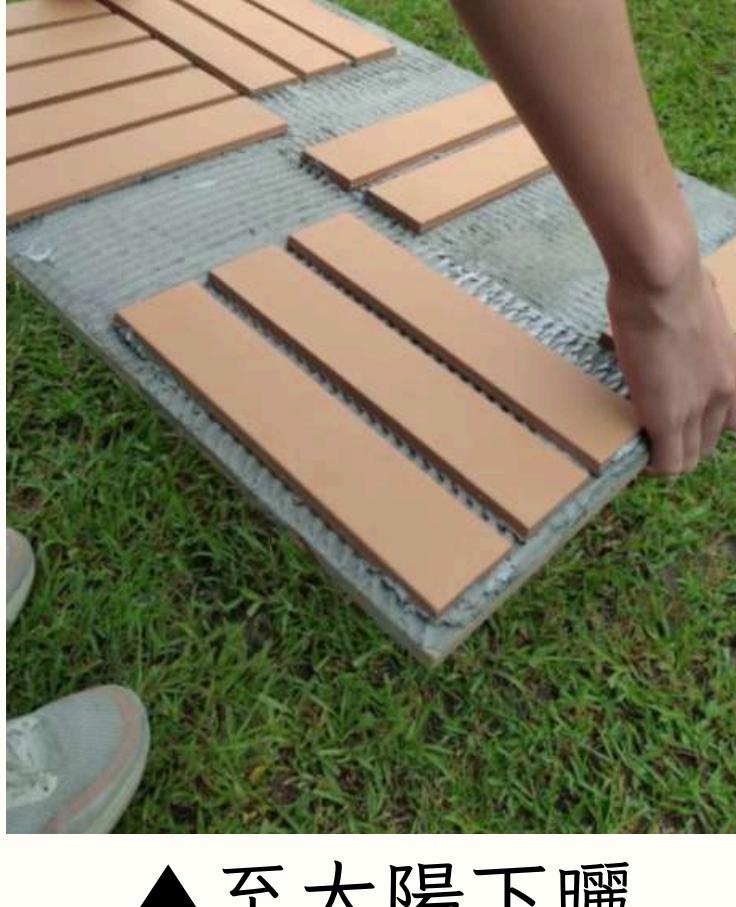
從下圖可以得知，空鼓程度較高的區域，溫度整體都比較高，因此，從我們的實驗中可以得知當磁磚某部分區域溫度相較其他區域的磁磚來得高時，可能會有空鼓的現象產生，發生剝落的風險較大。



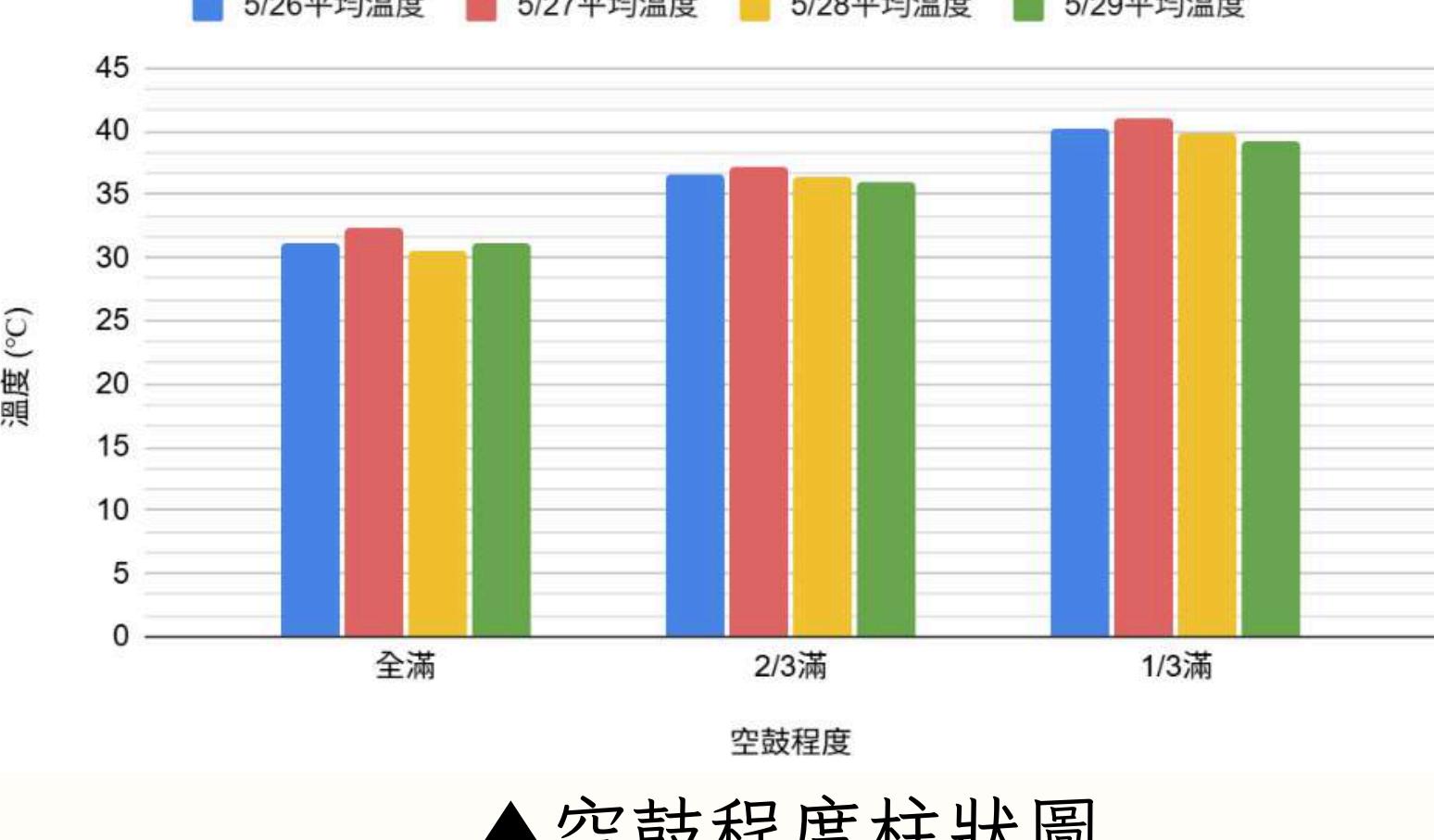
▲製作模擬牆面 (由作者所攝)



▲模擬牆面設計 (由作者所攝)



▲至太陽下曬 (由作者所攝)

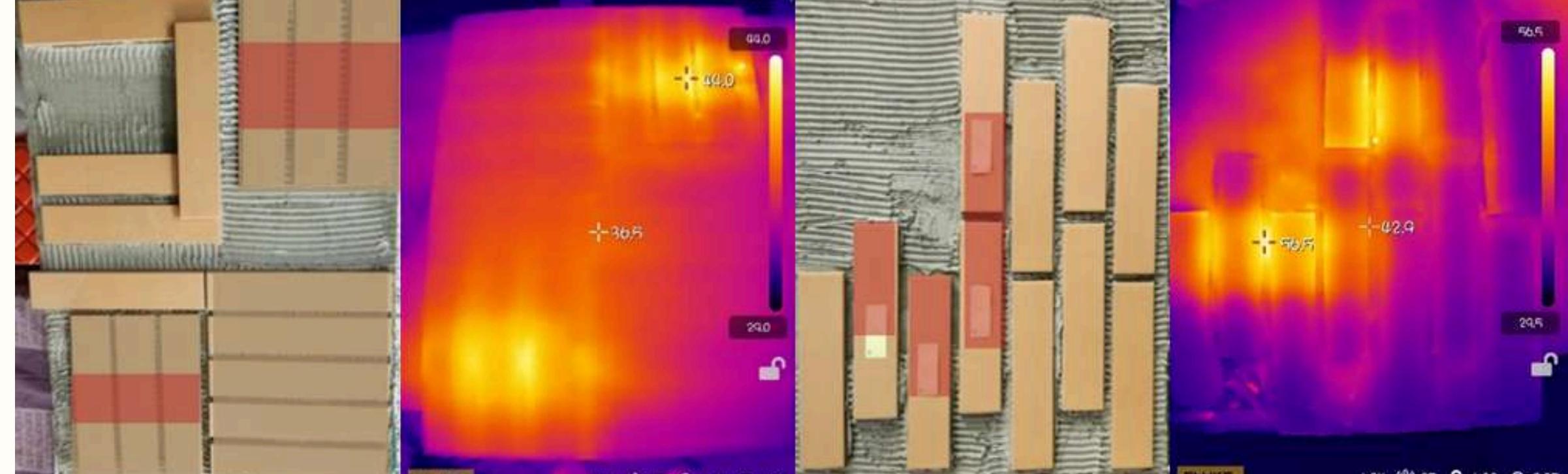


▲空鼓程度柱狀圖 (由作者所製)

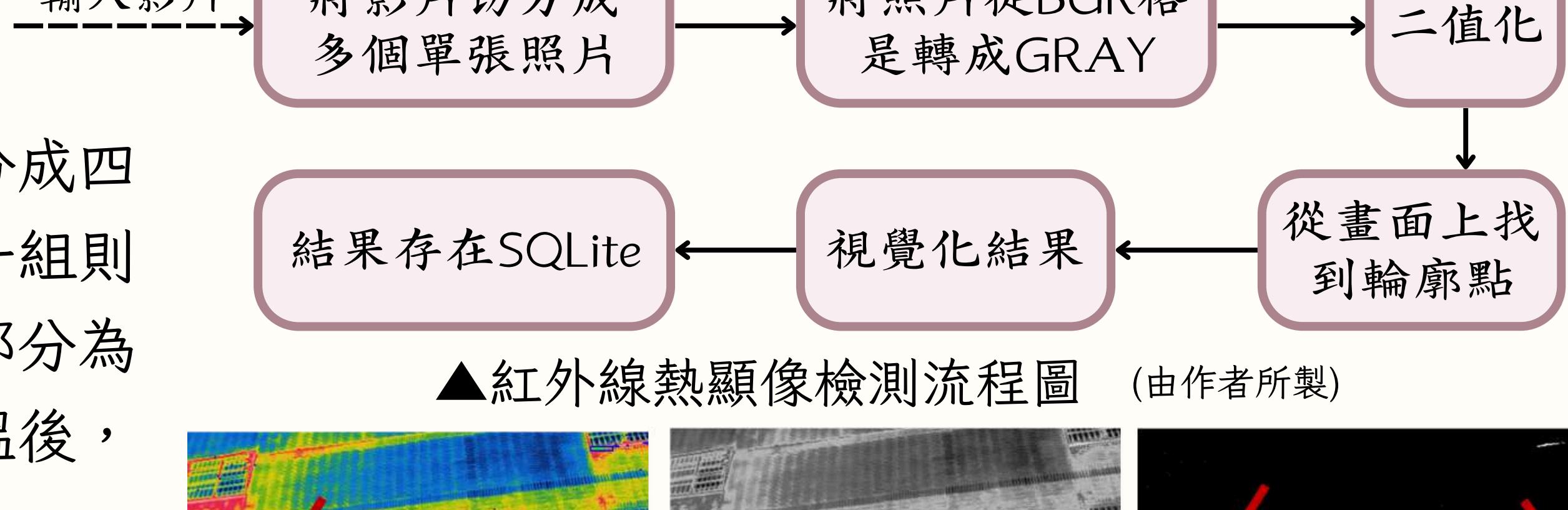
四、模型訓練階段

(一)紅外線熱顯像檢測

我們設計了兩塊模擬建築物外牆的磁磚板，一組是分成四個不同的區塊，分別代表不同程度的空鼓狀況，另一組則是真實模擬建築物外牆的交錯排列，其中板子紅色部分為空鼓。由圖片可以得知，有空鼓的區域經過日曬降溫後，內部會有熱空氣殘留，因此溫度較周圍磁磚來的高。



▲磁磚空鼓熱成像畫面 (由作者所攝)



▲紅外線熱顯像檢測流程圖 (由作者所製)



▲紅外線這部分程式圖-1 (由作者所製)

```

取得當前影格與時間戳
    current_frame = int(cap.get(cv2.CAP_PROP_POS_FRAMES))
    timestamp = current_frame / fps

將影格轉為灰階
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

超過二值化閾值像素變白
    threshold_value = 200
    _, binary_mask = cv2.threshold(gray_frame, threshold_value, 255, cv2.THRESH_BINARY)

在二值化影像中找外輪廓
    contours, _ = cv2.findContours(binary_mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

畫綠色輪廓在原影像上
    for contour in contours:
        cv2.drawContours(frame, [contour], -1, (0, 255, 0), 2)

取得輪廓的外接矩形
    x, y, w, h = cv2.boundingRect(contour)
    cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
    x1, y1, x2, y2 = x, y, x + w, y + h

cursor.execute("""
    INSERT INTO DetectionResults (timestamp, x1, y1, x2, y2)
    VALUES (?, ?, ?, ?, ?)
    """, (timestamp, x1, y1, x2, y2))

將偵測結果存入資料庫
    conn.commit()

每處理完一影格就上傳
    conn.commit()

```

▲紅外線這部分程式圖-2 (由作者所製)

(二)水泥牆裸露辨識

1. CNN模型訓練

我們使用Keras訓練簡易卷積神經網路CNN模型，用來辨識外牆是否由水泥裸露。我們使用兩層卷池層(分別使用32個和64個3x3濾波器)，然後訓練10個epoch，最後去測試CNN模型，輸出結果顯示「脫落」(peel)，表明模型成功識別出該區域存在脫落問題。

```

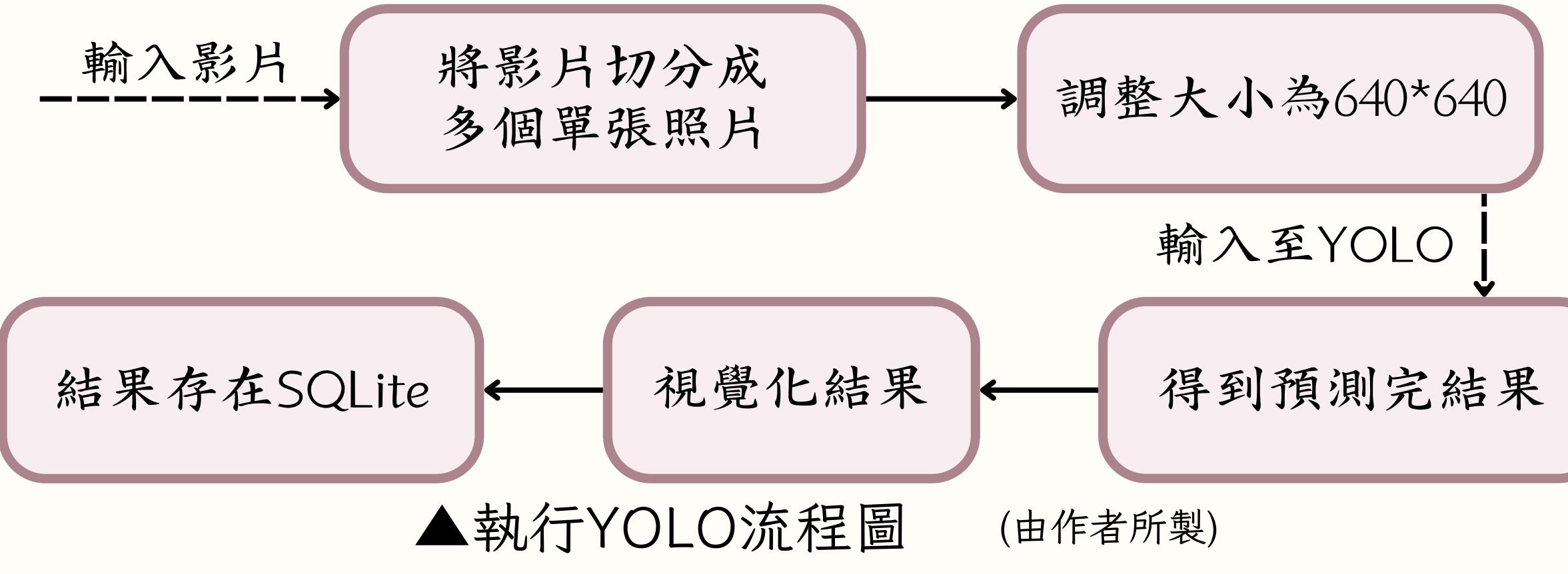
32*32濾波器
    model = Sequential([
        Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)),
        MaxPooling2D(pool_size=(2, 2)),
        Conv2D(64, (3, 3), activation='relu'),
        MaxPooling2D(pool_size=(2, 2)),
        Flatten(),
        Dense(128, activation='relu'),
        Dropout(0.5),
        # Dense(10, activation='softmax')
        Dense(train_generator.num_classes, activation='softmax')
    ])
    print(train_generator.num_classes)

```

▲CNN訓練程式圖 (由作者所製)

2.YOLO模型訓練

我們用LabelImg將YOLO模型訓練需要的圖像資料進行標記，框選出有缺陷的區域，以「peel」來表示磁磚剝落等。標記完成後，會自動將每張圖片的類別編號和圖片中心點與寬高的相對座標資訊儲存再開始進行YOLOv8的模型訓練，訓練參數如訓練輪數、影像尺寸、批次大小等皆可依據不同需求調整。最後測試YOLO模型，輸出結果可以成功匡列水泥裸露的區域。



▲執行YOLO流程圖 (由作者所製)

```

取得原始影格影像
    for result in results:
        frame = result.orig_img

    timestamp = frame_index / fps
    frame_index += 1

    for box in result.bboxes:
        x1, y1, x2, y2 = map(int, box.xyxy[0])
        confidence = float(box.conf[0])

        cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 255), thickness=1)

    cursor.execute("""
        INSERT INTO DetectionResults (timestamp, x1, y1, x2, y2)
        VALUES (?, ?, ?, ?, ?)
        """, (timestamp, x1, y1, x2, y2))

    conn.commit()

    每處理一影格就上傳
    conn.commit()

```

▲水泥辨識部分程式圖-2 (由作者所製)

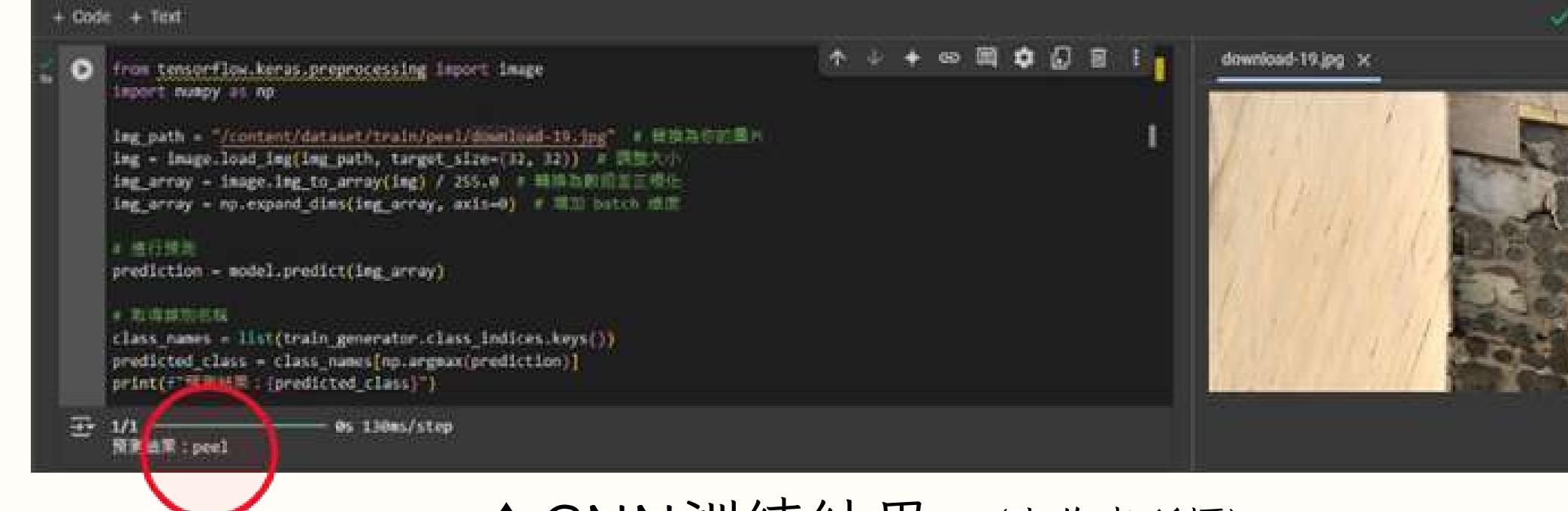
timestamp	#	x1	#	y1	#	x2	#	y2	#	Temperature	#
1	0.0333274976573295	279		1079		280		1080		203	
2	0.0333274976573296	282		1078		284		1080		228.66666666666666	
3	0.0333274976573296	1085		1072		1086		1073		204	
4	0.0333274976573296	147		1070		148		1071		201	
5	0.0333274976573296	296		1062		297		1063		202	
6	0.0333274976573296	274		1061		275		1062		210	
7	0.0333274976573296	241		1055		242		1057		220.5	

▲IR SQLite實際資料庫畫面 (由作者所製)

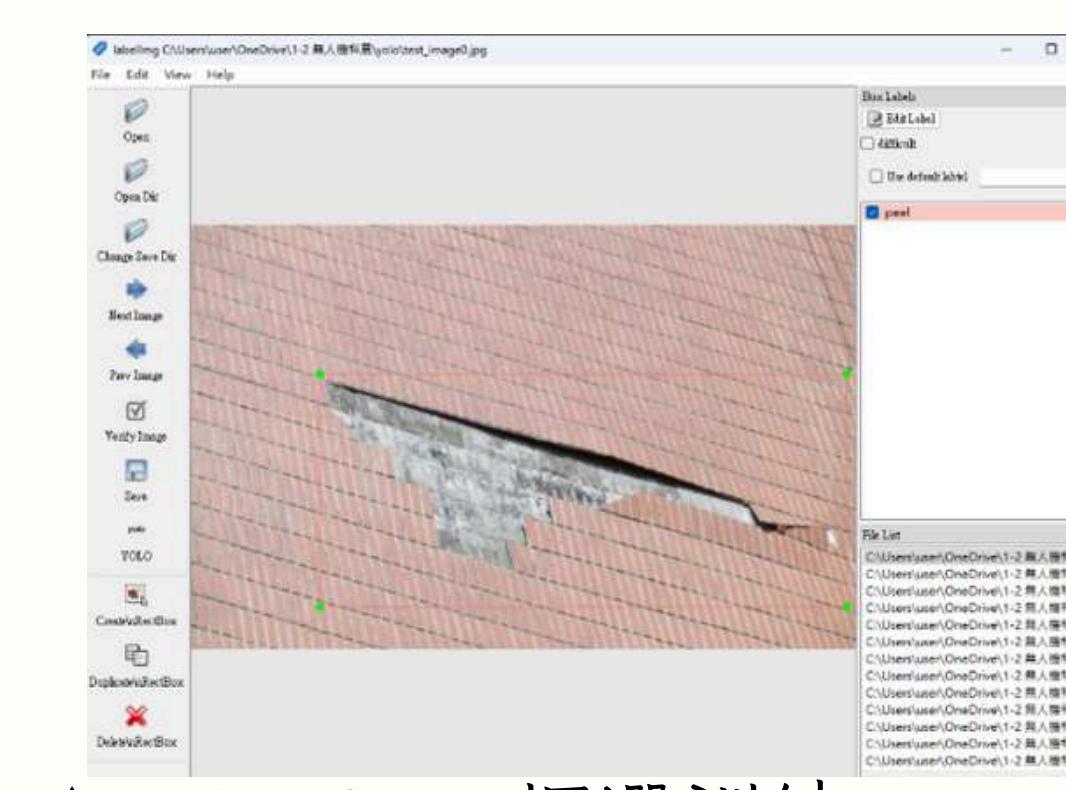
▲紅外線程式圖修正 (由作者所攝)

model.fit(train_generator,	validation_data=val_generator,	epochs=10 #訓練 10 個 Epoch)
Epoch 1/10	4s 295ms/step - accuracy: 0.7133 - loss: 0.4914 - val_accuracy: 0.9167 - val_loss: 0.2809			
Epoch 2/10	2s 201ms/step - accuracy: 0.9216 - loss: 0.2690 - val_accuracy: 0.9167 - val_loss: 0.2747			
Epoch 3/10	2s 196ms/step - accuracy: 0.9084 - loss: 0.2493 - val_accuracy: 0.9167 - val_loss: 0.2385			
Epoch 4/10	2s 201ms/step - accuracy: 0.9124 - loss: 0.1902 - val_accuracy: 0.9167 - val_loss: 0.2847			
Epoch 5/10	2s 204ms/step - accuracy: 0.8968 - loss: 0.1598 - val_accuracy: 0.9167 - val_loss: 0.3161			
Epoch 6/10	2s 284ms/step - accuracy: 0.9223 - loss: 0.1123 - val_accuracy: 0.7667 - val_loss: 0.4761			
Epoch 7/10	2s 237ms/step - accuracy: 0.9272 - loss: 0.1277 - val_accuracy: 0.8167 - val_loss: 0.4159			
Epoch 8/10	2s 198ms/step - accuracy: 0.9801 - loss: 0.0955 - val_accuracy: 0.7167 - val_loss: 0.4274			
Epoch 9/10	2s 197ms/step - accuracy: 0.9801 - loss: 0.0955 - val_accuracy: 0.7167 - val_loss: 0.4274			
Epoch 10/10	2s 204ms/step - accuracy: 0.9657 - loss: 0.0886 - val_accuracy: 0.7000 - val_loss: 0.4464			
Epoch 11/10	2s 193ms/step - accuracy: 0.9464 - loss: 0.0882 - val_accuracy: 0.4833 - val_loss: 0.6432			

▲CNN訓練過程 (由作者所攝)



▲CNN訓練結果 (由作者所攝)

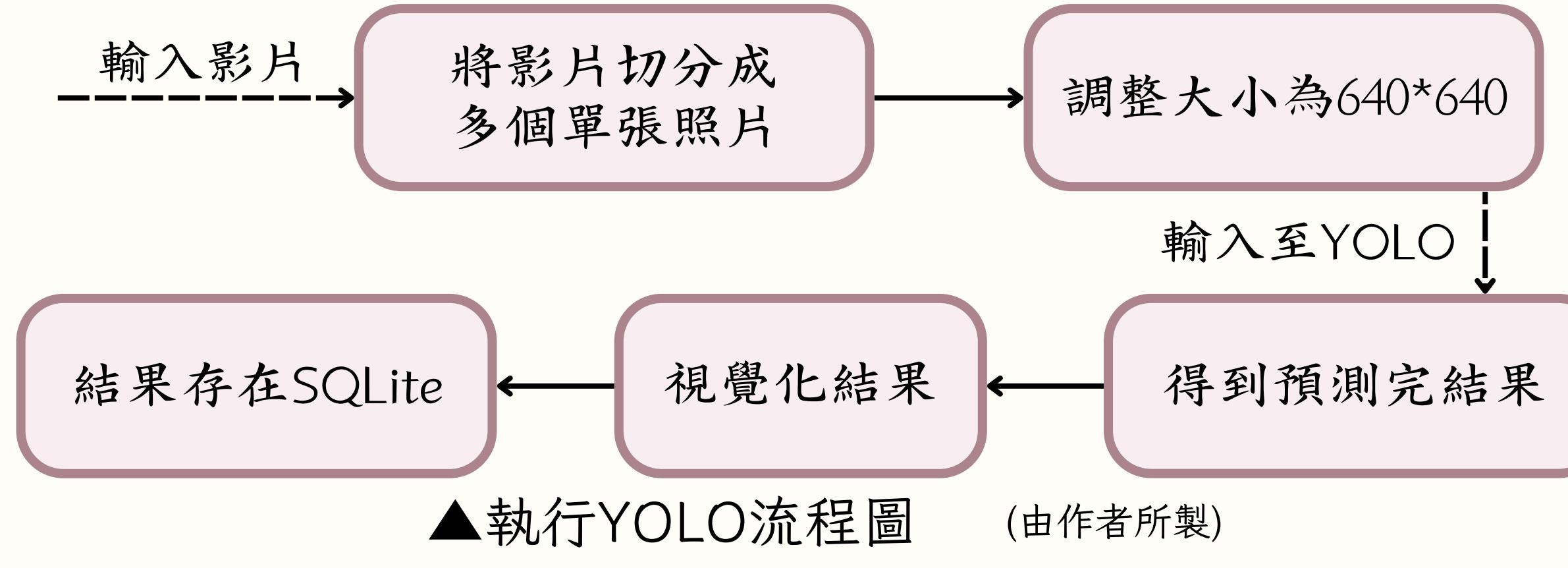


▲LabelImg 框選訓練 (由作者所製)

▲YOLO 模型權重 (由作者所製) ▲YOLO模型訓練結果 (由作者所製)



我們最後選用YOLO模型來做我們系統的水泥裸露辨識，是因為CNN較適用於做分類任務，但我們的系統更需要的是辨識水泥裸露的部分並標示其位置座標，因此我們選擇YOLO來做我們的水泥外牆辨識的模型。



▲執行YOLO流程圖 (由作者所製)

```

取得原始影格影像
    for result in results:
        frame = result.orig_img

    timestamp = frame_index / fps
    frame_index += 1

    for box in result.bboxes:
        x1, y1, x2, y2 = map(int, box.xyxy[0])
        confidence = float(box.conf[0])

        cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 255), thickness=1)

    cursor.execute("""
        INSERT INTO DetectionResults (timestamp, x1, y1, x2, y2)
        VALUES (?, ?, ?, ?, ?)
        """, (timestamp, x1, y1, x2, y2))

    conn.commit()

    每處理一影格就上傳
    conn.commit()

```

▲水泥辨識部分程式圖-2 (由作者所製)

```

    載入YOLO模型
    model = YOLO("best.pt")

    video_path = 'result.mp4'
    cap = cv2.VideoCapture(video_path)

    檢查影片是否打開
    if not cap.isOpened():
        print("Error: Could not open video file.")
        exit()

    取得影片的FPS
    fps = cap.get(cv2.CAP_PROP_FPS)
    if fps == 0:
        print("Error: Could not retrieve FPS from video.")
        exit()

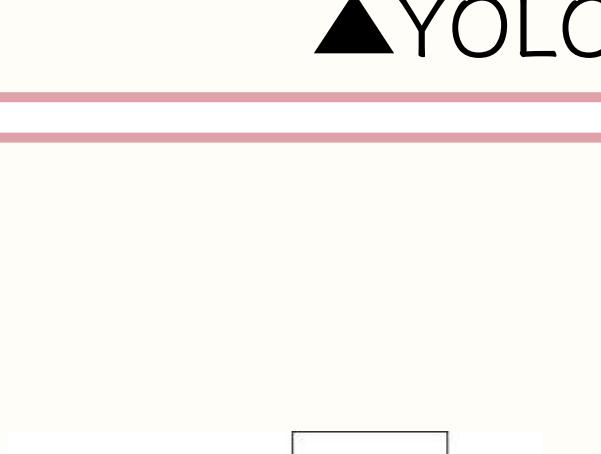
    進行YOLO偵測
    results = model(source=video_path, stream=True, conf=0.5, imgsz=640, iou=0.25, device='0')

```

▲水泥辨識部分程式圖-1 (由作者所製)

timestamp	#	x1	#	y1	#	x2	#	y2	#
1	0.03331557449831291	1081		395		1082		396	
2	0.03331557449831291	908		398		909		399	
3	0.03331557449831291	1063		443		1064		444	
4	0.03331557449831291	434		0		1049		321	
5	0.03331557449831291	1072		413		1156		433	
6	0.03331557449831291	1383		881		1384		882	
7	0.03331557449831291	946		540		947		541	
8	0.03331557449831291	696		767		697		768	
9	0.03331557449831291	1014		365		1056		376	
10	0.03331557449831291	391		0		420		34	

▲YOLO SQLite實際資料庫畫面 (由作者所製)



▲IOU圖(聯集分之交集) (圖片來源: Google)

取得相交區域座標
計算交集區域面積
計算每個框的面積
計算聯集區域面積
計算IOU

def calculate_iou(box1, box2):
 x1 = max(box1[0], box2[0])
 y1 = max(box1[1], box2[1])
 x2 = min(box1[2], box2[2])
 y2 = min(box1[3], box2[3])

inter_width = max(0, x2 - x1)
inter_height = max(0, y2 - y1)
intersection = inter_width * inter_height

area1 = (box1[2] - box1[0]) * (box1[3] - box1[1])
area2 = (box2[2] - box2[0]) * (box2[3] - box2[1])

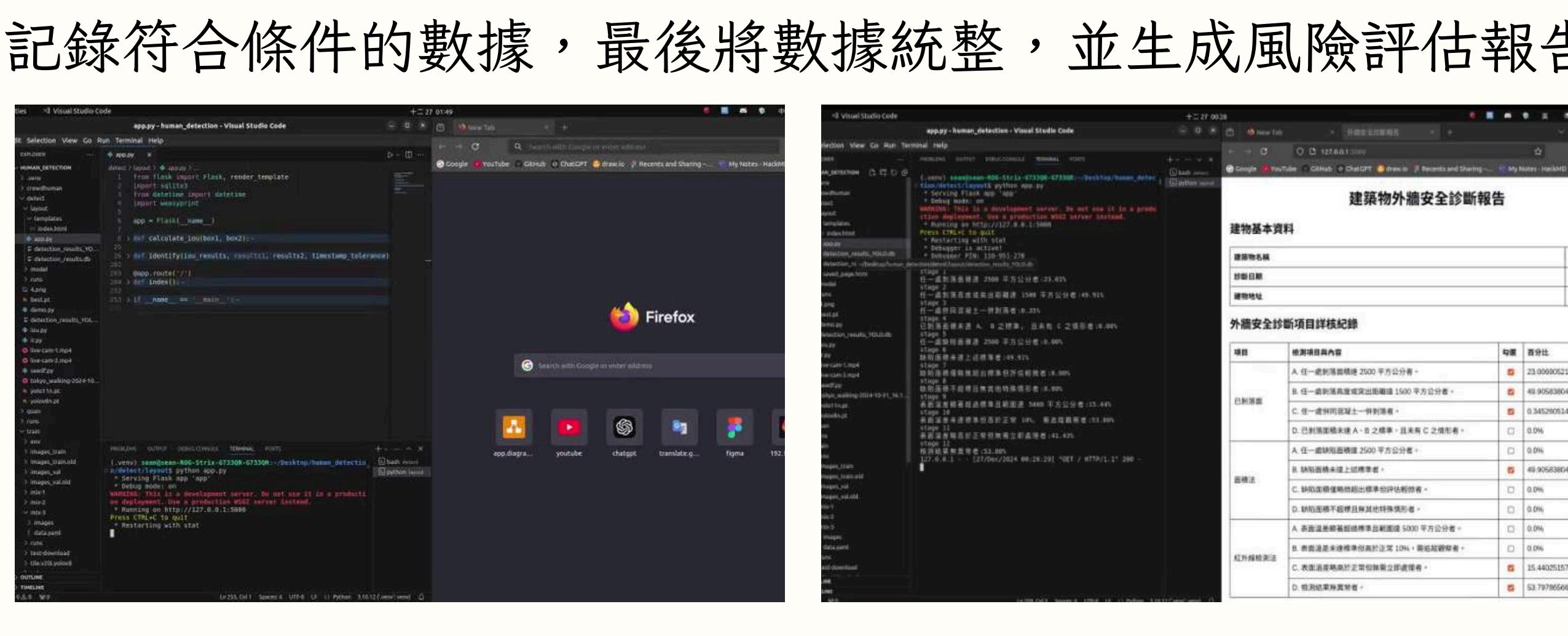
union = area1 + area2 - intersection
if union == 0:
 return 0

return intersection / union

確保IOU不是0
表示有重疊
檢查兩筆數據的時間
截記是否在範圍內
計算該區域面積
記錄符合條件次數
記錄總比較次數
計算符合條件的比例

def identify(iou_results, results1, results2, timestamp_tolerance):
 # 檢查剝落面積是否大於 2500 平方公分
 print("stage 1")
 aa = 0
 counter = 0
 for row1 in iou_results:
 if row1[1] != 0:
 for row2 in results2:
 if abs(row1[0] - row2[0]) <= timestamp_tolerance:
 if ((row2[4] - row2[0]) * (row2[3] - row2[1])) > 2500:
 aa += 1
 counter += 1
 aa /= counter
 print(f"一處剝落面積大於 2500 平方公分者: {(aa * 100):.2f}%")

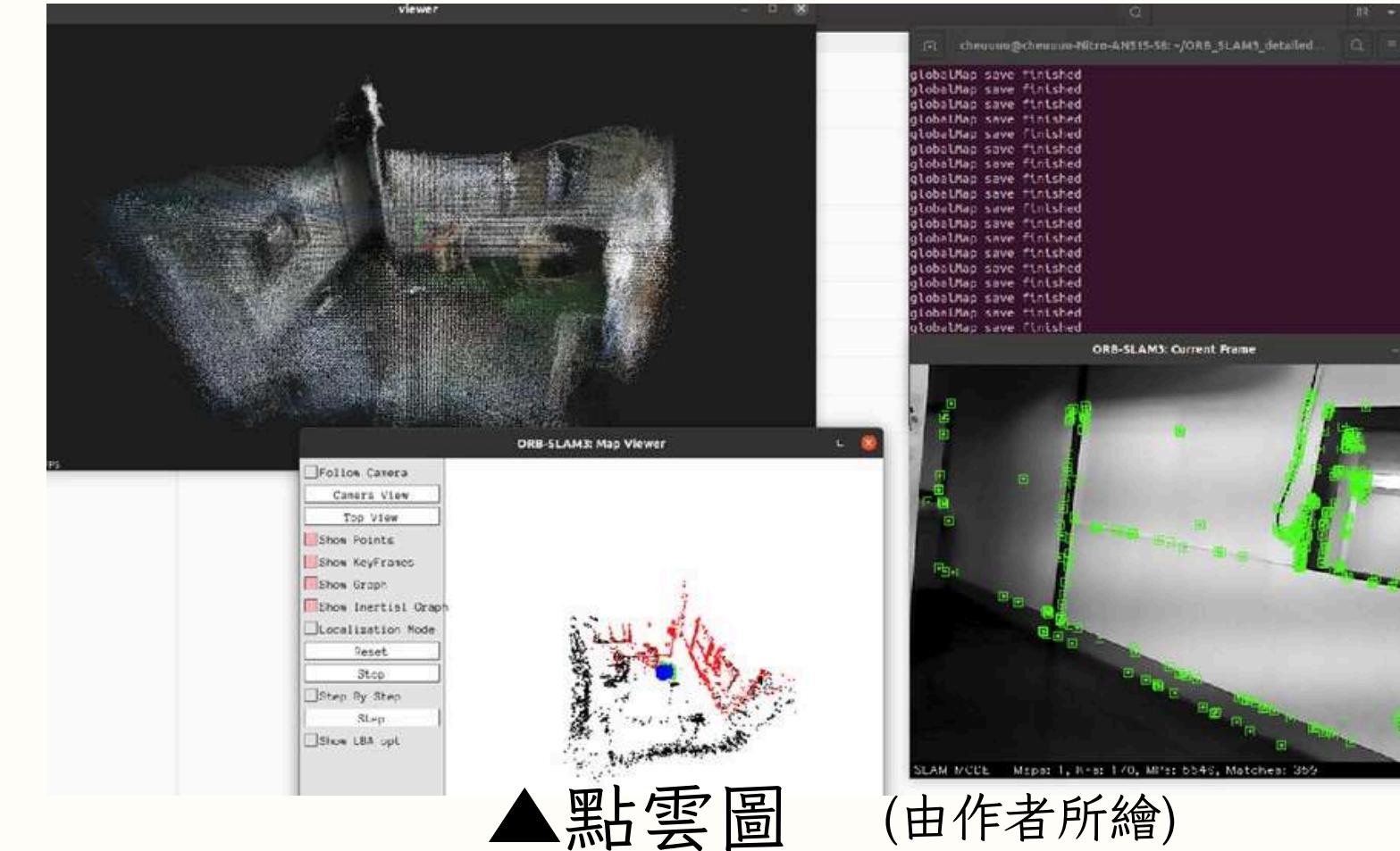
▲評估報告程式碼 (由作者所製)



▲AI評估報告生成過程 (由作者所攝)

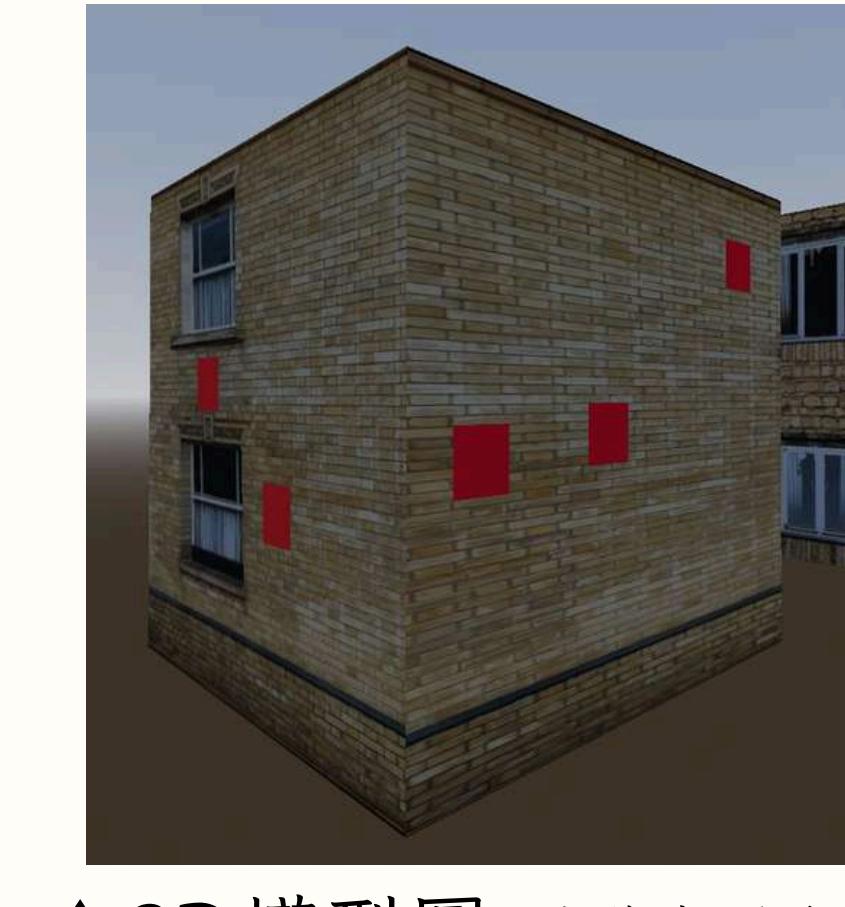
六、3D建模

研究初期，我們嘗試結合RGB與深度資訊的深度相機，搭配無人機近距離掃描建築外牆進行3D建模。然而，深度相機在戶外環境中容易受強光干擾，導致深度資料不精確，同時深度鏡頭有效建模範圍亦有限，對大面積牆體與高樓結構無法高效覆蓋，建模品質與效率皆受限。



▲點雲圖

(由作者所繪)



▲3D模型圖 (由作者所繪)

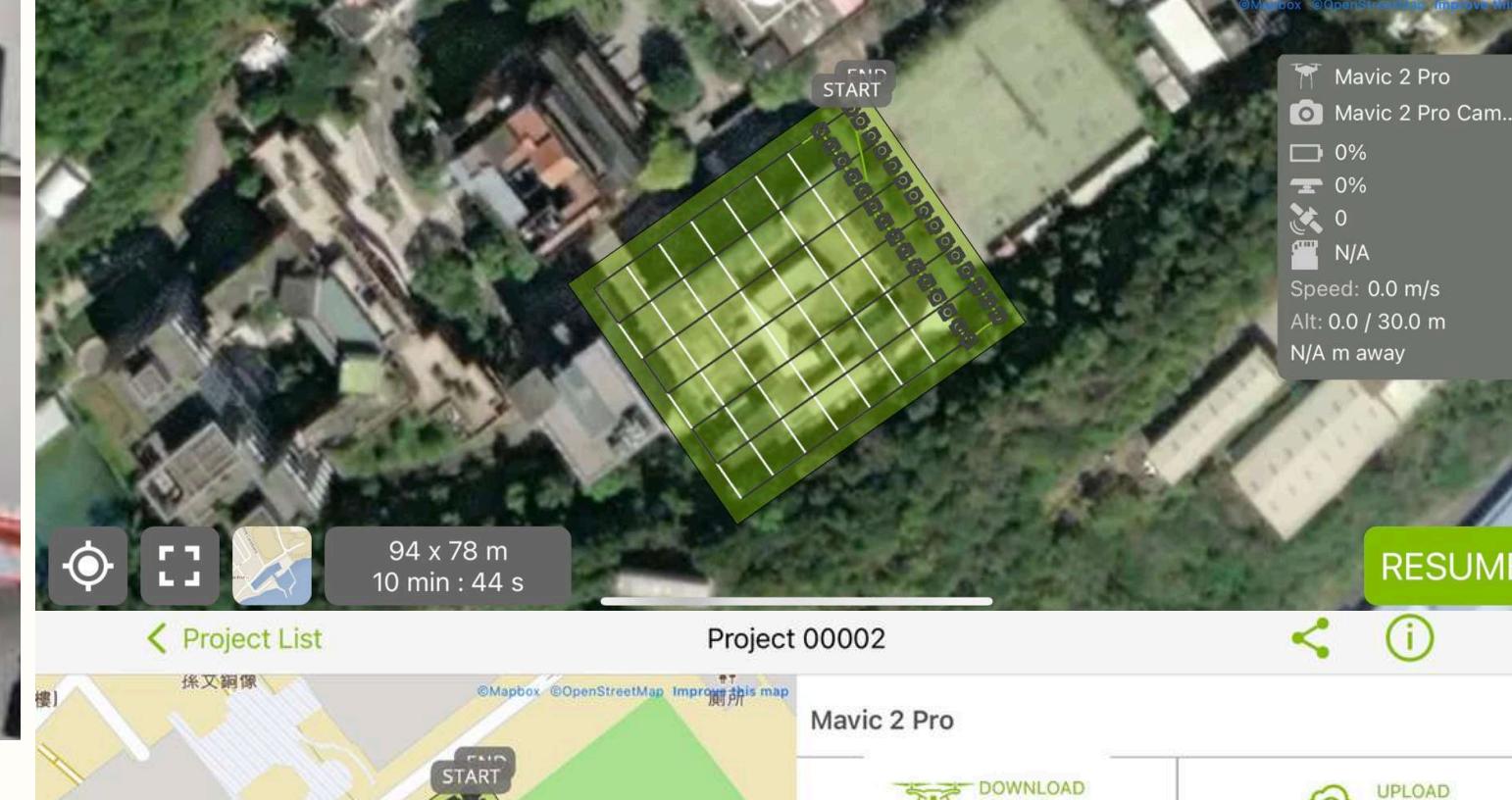
為提升整體可視化效果與作業效率，我們改採使用無人機結合 PIX4Dcapture app 進行自動化航線規劃與影像拍攝，目標擴展至整個社區。可自行設定飛行高度（30公尺）、覆蓋面積（94x78公尺）、影像重疊率等參數，無人機可預估任務時間（10分鐘44秒）。此類標準化參數設定有助於提升影像品質與後續建模精度。



▲PIX4Dcapture 操作畫面



▲PIX4Dmapper 建模畫面



▲PIX4Dcapture 操作介面

94 x 78 m
10 min : 44 s

Mavic 2 Pro
Mavic 2 Pro Cam...
0%
0%
0
N/A
Speed: 0.0 m/s
Alt: 0.0 / 30.0 m
N/A m away



▲3D 模型可視化呈現



▲3D 模型呈現

(此區圖片皆由作者所製、所攝、所繪)

針對磁磚空鼓與水泥裸露區域的辨識與標示，我們於模型上採用雙分類顯示：兩者重疊區以紅色標記，無重疊區則以黃色標示，以利後續風險分析與維修判斷。

七、系統整合

我們的系統以模組化方式運作，無人機搭載雙光鏡頭拍攝建築外牆，回傳資料將透過自行訓練的紅外線進行磁磚空鼓辨識，YOLO模型進行水泥裸露偵測，再將儲存於SQLite的資料疊合後生成3D可視化模型與風險報告。整體架構具備彈性擴充，可應用於多棟建築長期監測與維護決策，提升檢測效率與實用性。



▲系統整合概念圖 (由作者所繪)

肆、結論

一、高效自動化檢測

無人機取代人工高空作業，搭配自動影像分析，大幅提升外牆磁磚檢測效率與安全性。

二、AI影像辨識

結合紅外線熱成像與YOLO模型，偵測溫度異常(空鼓)與水泥裸露區域，提升缺陷辨識準確度。

三、可視化數據

自動生成缺陷可視化影像，量化磁磚剝落嚴重程度，並結合3D模型，將檢測結果視覺化。

四、預防性風險評估

資料雲端儲存與長期監測，建立智慧化預測機制，提前預警並優化維修策略。

我們的系統使用無人機載具搭配紅外線熱成像技術來做磁磚空鼓的檢測，透過 YOLO模型來檢測外牆磁磚剝落導致的水泥裸露，我們可實現在建築物外牆的智能巡檢，自動識別潛在風險區域，並進行標記與可視化分析。透過數據的分析與機器學習演算法，系統能預測維護需求，還可以 3D建模以及生成風險評估報告，幫助使用者做出更好的決策。這項技術應用建築檢測，可以提升檢測效率與準確性，同時降低維修成本與管理負擔，進一步提升建築安全與檢測技術的普及率。

伍、參考文獻資料

1. 馬揚異 (2023)，全台平均屋齡33年，奪命磁磚雨10年後未改善，今周刊1399期，<https://reurl.cc/OY218r>
2. 北區中心 (2024)，於施工架上從事外牆補磚作業發生墜落災害致死職業災案，勞動部職業安全衛生署，<https://reurl.cc/mxWn1G>
3. 盧班瑞 (2010)，高層集合住宅外牆磁磚剝落原因與解決對策探討（一），內政部建築研究所工程技術組，<https://reurl.cc/jqX702>
4. 洪宇 (2024)，用YOLOv8訓練自己的資料集+詳解，Medium，<https://reurl.cc/26Rbl6>
5. Realibox3D (2024)，3D設計的基本原理和技術，SOHU，https://www.sohu.com/a/668852130_121668539
6. 霖思科技 (2024)，3D點雲圖，LINX，<https://www.linx.tw/tech-info/technique/1216/2024-12->