

# 成果報告書



## 基於YOLO的水稻葉病害識別與分析

指導教師：謝孟諺 教授

專題學生：

資工四 B 411147649 林泳存

資工四 B 411147665 蘇建豪

資工四 B 411147403 許秉豐

資工四 B 411147186 蔡尚廷

---

專題名稱：基於深度學習的水稻葉病害識別與分析

指導教師：謝孟諺 教授

專題學生：

資工四 B 411147649 林泳存 [s1114764@o365st.pu.edu.tw](mailto:s1114764@o365st.pu.edu.tw)

資工四 B 411147665 蘇建豪 [s1114766@o365st.pu.edu.tw](mailto:s1114766@o365st.pu.edu.tw)

資工四 B 411147403 許秉豐 [s1114740@o365st.pu.edu.tw](mailto:s1114740@o365st.pu.edu.tw)

資工四 B 411147186 蔡尚廷 [s1114718@o365st.pu.edu.tw](mailto:s1114718@o365st.pu.edu.tw)

---

# 目錄

目錄 .....	3
圖目錄.....	4
第一章 序論 .....	5
壹、 前言 .....	5
貳、 動機 .....	5
第二章 報告介紹 .....	6
壹、 病徵介紹 .....	6
貳、 系統架構圖： .....	7
參、 功能概述 .....	8
肆、 使用對象 .....	8
伍、 開發工具 .....	8
第三章 網頁介紹 .....	9
壹、 網頁架構圖 .....	9
貳、 系統畫面流程 .....	10
第四章 模型介紹 .....	20
壹、 YOLOv11 訓練過程 .....	20
貳、 模型表現： .....	22
一、 Results - 整體模型表現.....	22
二、 Results - 模型整體結論.....	23
三、 BoxF1_curve (F1-Confidence Curve) .....	24
四、 BoxPR_curve.....	26
五、 labels.....	27
六、 confusion matrix.....	28
第五章 補充說明 .....	29
壹、 遇到的困難 .....	29
貳、 結論及未來發展 .....	29
一、 結論： .....	29
二、 短期目標： .....	29
三、 長期目標： .....	29
參、 經費預算需求表（執行中所需之經費項目單價明細） .....	30
肆、 工作分配（詳述參與人員分工） .....	30
伍、 名詞解釋 .....	31
陸、 參考資料 .....	32

# 圖目錄

圖 1 白葉枯病(Bcterialblight)	6
圖 2 褐斑病(Bownspot)	6
圖 3 稻熱病(Blast)	6
圖 4 系統架構圖	7
圖 5 網頁架構圖	9
圖 6 系統畫面流程	10
圖 7 爬蟲	12
圖 8 排程器	12
圖 9 爬蟲數據	13
圖 10 天氣	14
圖 11 主頁面	15
圖 12 分析畫面	16
圖 13 歷史紀錄	16
圖 14 米價查詢	18
圖 15 關於我們	19
圖 16 關於我們	19
圖 17 YOLOv11 訓練過程	20
圖 18 訓練過程	21
圖 19 訓練過程	21
圖 20 模型表現	22
圖 21 模型整體	23
圖 22 模型整體	23
圖 23 BoxF1_curve	24
圖 24 BoxPR_curve	26
圖 25 labels	27
圖 26 confusion matrix	28
圖 27 改善後的標記方式	29
圖 28 原本標記方式	29
圖 29 甘特圖	30

# 第一章 序論

## 壹、前言

水稻是亞洲地區最主要的糧食作物，其收成狀況直接影響了數億人口的糧食安全與經濟生活。然而，水稻在其生長週期中極易遭受各類病害侵襲，如臺灣嘉南地區曾爆發的大規模稻熱病事件，皆證實了病害對農民收入具有毀滅性的衝擊。

傳統的病害辨識與診斷主要依賴專業農技人員的經驗判斷，此方式不僅人力成本高昂、效率緩慢，且在病害大規模蔓延時難以即時處置，從而錯失最佳診治期。為保障糧食供給並穩定農民生計，尋求一套快速、準確、低成本的病害診斷與防治方案已成為當前農業領域極待解決的關鍵課題。

## 貳、動機

有鑑於傳統診斷的限制以及現代數位工具的普及性，我們決定結合影像分析與深度學習技術來應對水稻病害挑戰。透過機器學習模型，可實現對水稻病害的早期預警和精準防治。

此外，考慮到目前農業人口結構逐漸老化，數位工具的使用門檻對部分年長農民構成挑戰。因此，本專題在網站建構時，特別捨棄複雜功能與華麗介面，採用極簡風格設計，致力於打造一個連不熟悉數位工具的長者也能輕鬆操作的實用型數位工具，確保成果能真正落地服務農民。

## 第二章 報告介紹

### 壹、病徵介紹

#### 1. 白葉枯病(Bacterial blight)

典型葉枯病徵主要發生在分蘖盛期後，葉緣出現水浸狀小斑，逐漸延長成黃色後變為白或灰白色枯斑，病斑處常有黃色菌泥溢出。急性萎凋型病徵（Kresiek）常在幼苗期出現，導致幼葉捲曲萎凋、植株矮小、呈冠腐狀，嚴重時整株枯死，成株期也會在劍葉上形成大白色斑紋並向下延伸。淡黃化型病徵則出現在最幼小葉片上，呈現淡黃或白色，徵狀類似缺鐵。

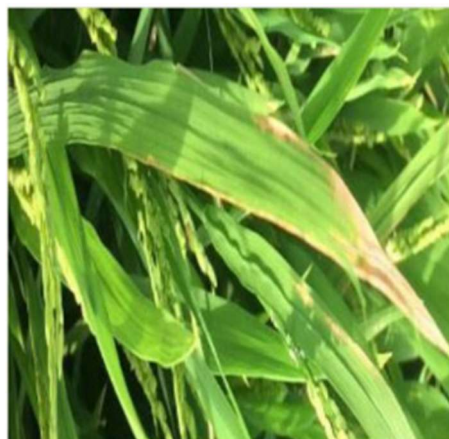


圖 1 白葉枯病(Bacterial blight)

#### 2. 褐斑病(Brown spot)

褐斑病典型病徵是葉片出現淡褐色水浸狀斑點，隨後擴大成褐、暗褐或黑色的不規則凹陷壞疽斑，周圍有明顯黃暈，病斑可相互融合變大；有些感染會擴展成長條形或不規則形的深綠色或黑褐色大型水浸狀斑，病斑周圍常有墨綠色環紋，且病斑處觸摸仍堅硬，與軟腐病不同。濕度高時病斑破裂可見乳白色菌泥溢出。罹病葉片最終黃化乾枯，若病勢擴展至生長點會導致整株死亡；幼苗感染後則迅速軟腐壞死。

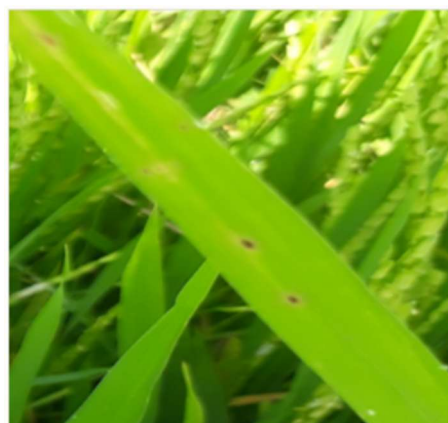


圖 2 褐斑病(Brown spot)

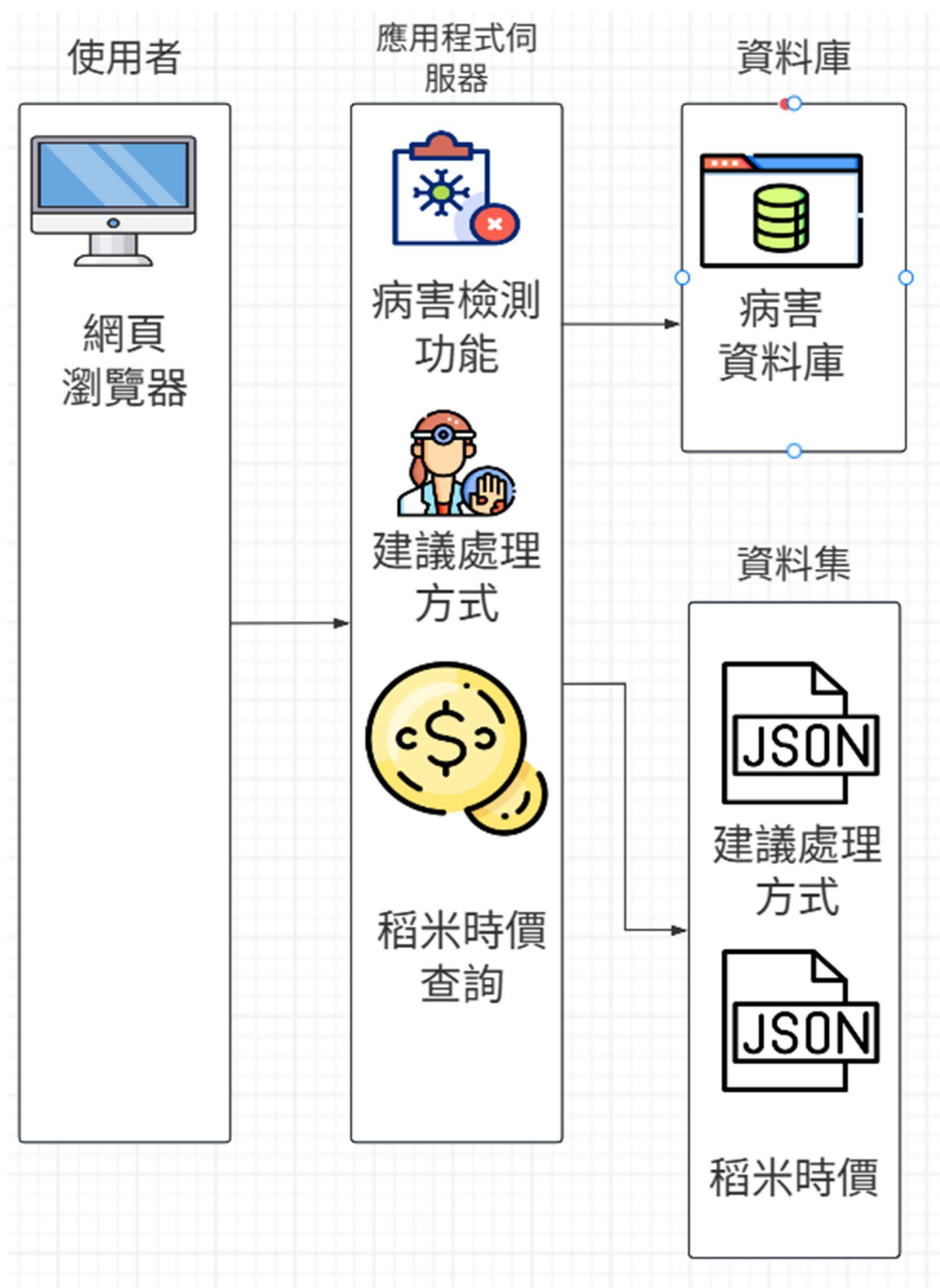
#### 3. 稻熱病(Blast)

發病初期於葉片上產生小型棕褐色病斑，帶有黃暈。發展中的病斑隨葉脈逐漸擴大，呈紡錘型，二端稍尖，周圍出現水浸狀，中央黑灰色，邊緣棕褐色，外圍有薄層黃暈。病斑停止發展時，中央轉為灰白色，常數個病斑連結，造成葉片乾枯。



圖 3 稻熱病(Blast)

貳、系統架構圖：



▲圖 4 系統架構圖

### 參、功能概述

1. 透過訓練深度學習模型，建立高效率、高準確性的水稻病害圖像自動分類系統，解決傳統人工診斷的效率瓶頸。
2. 防治建議：根據影像分析的結果，提供使用者相應的防治建議與處理方法，實現農藥的精準使用，避免農藥濫用，同時保護環境。
3. 建構友善使用者介面：開發操作介面簡化、功能直覺化的網頁應用系統，提供農民、農業相關單位及研究機構等一個易於使用且功能整合的平臺。
4. 整合農業即時資訊：整合網路爬蟲技術，提供水稻時價查詢和最新農業資訊瀏覽功能，方便使用者一站式獲取關鍵農業情報，輔助決策

### 肆、使用對象

1. 農民
2. 農業相關單位或研究機構
3. 學校教師與學生

### 伍、開發工具

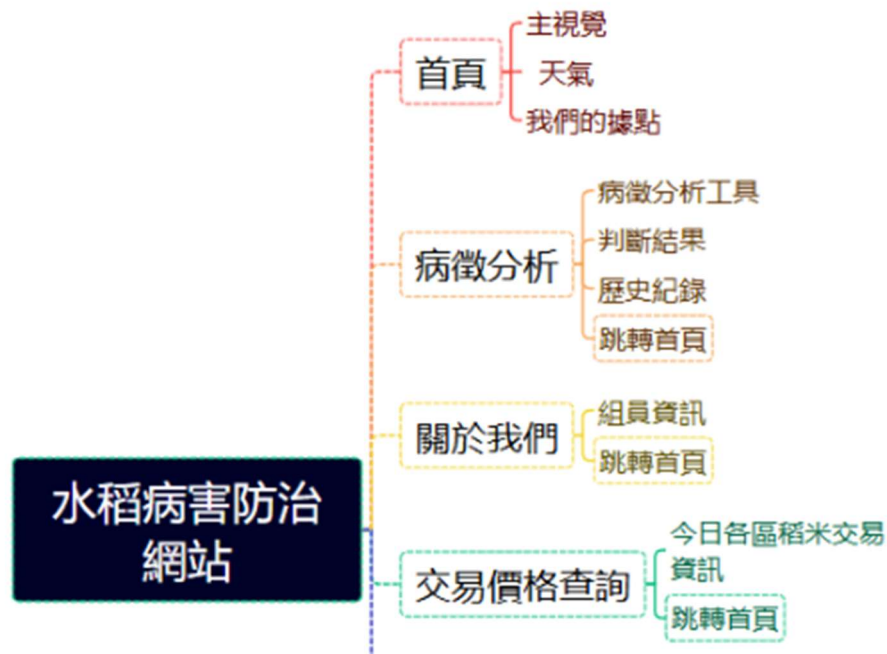
軟體

1. 使用語言：Python
2. 使用套件 YOLO、Django
3. 使用開發環境 IDE：Pycharm，VScode
4. 使用開發環境 IDE：Pycharm，VScode



## 第三章 網頁介紹

### 壹、網頁架構圖

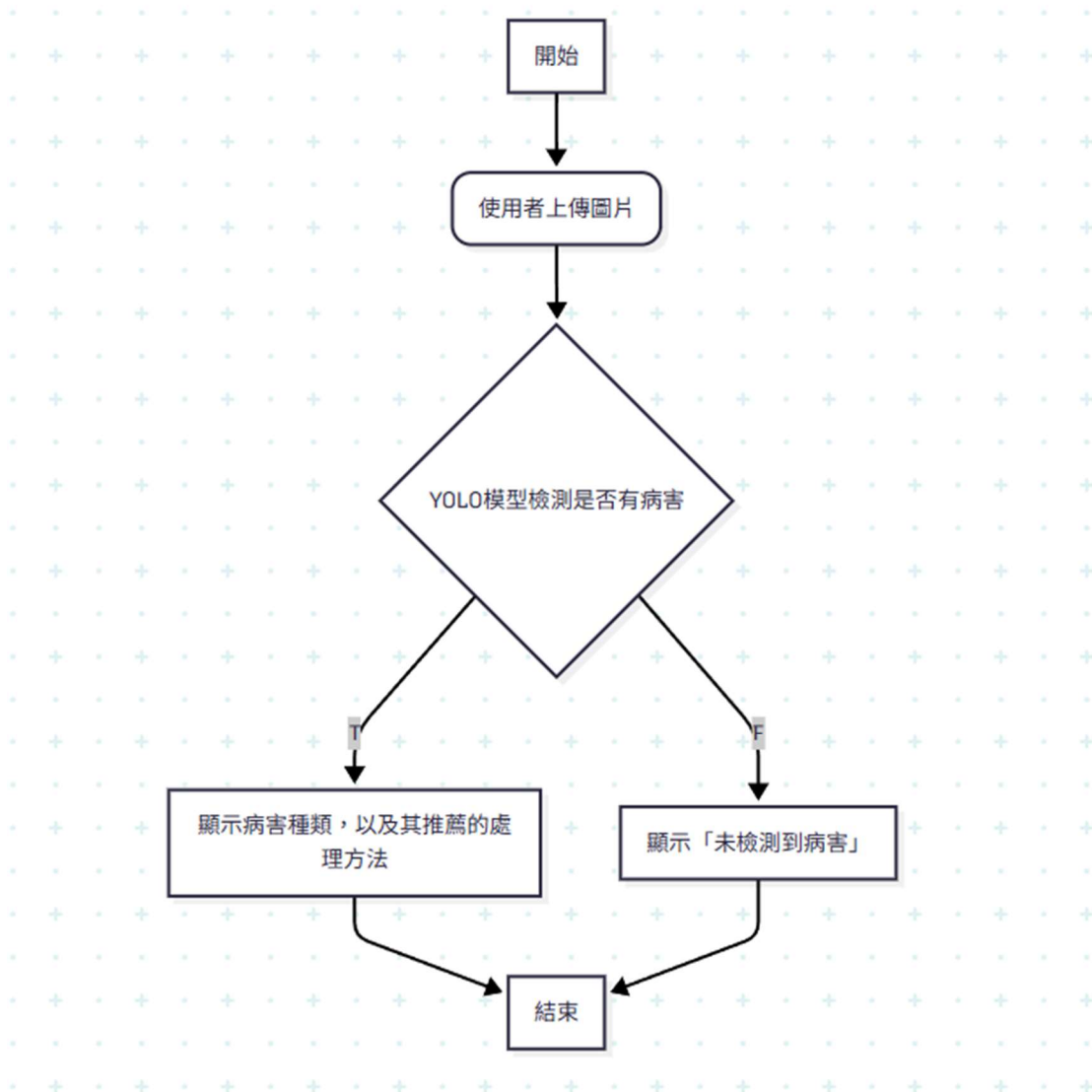


▲圖 5 網頁架構圖

網站的整體功能依據使用者需求劃分為首頁、病徵分析、關於我們及交易價格查詢四大模組（如圖 5 所示）。

- 首頁：主要負責展示系統的核心主題（主視覺），並提供天氣查詢。
- 病徵分析：是網站的核心功能，使用者可進入病徵分析工具上傳圖片，並查看判斷結果 及 歷史紀錄。
- 交易價格查詢：提供今日各區稻米交易資訊，輔助農民決策。
- 關於我們：提供組員資訊，方便使用者聯絡與回饋。

## 貳、系統畫面流程



▲圖 6 系統畫面流程

圖6為我們的「稻米病害辨識系統」在整個辨識過程中的運作邏輯與判斷流程。當使用者進入該頁面後，第一步是透過網頁介面上傳一張稻葉圖片。系統接收到圖片後，會將該影像輸入至事先訓練好的 YOLO 模型中進行分析。YOLO 模型會針對圖片內容進行物件偵測，判斷稻葉上是否出現任何的病害特徵，例如稻熱病（Rice Blast）、褐斑病（Brown Spot）或白葉枯病（Bacterial Blight）。若模型偵測結果顯示圖片中存在病害（流程中的「True」分支），系統便會進一步將辨識出的病害種類顯示於畫面上，並同時提供對應的建議處理方法與防治措施，幫助使用者瞭解病情並採取適當行動。而若模型判斷該圖片中未出現明顯病斑（流程中的

「False」分支)，系統則會回傳提示訊息「未檢測到病害」，表示目前的稻葉影像狀態健康或不屬於已知病害範圍。無論是哪一種結果，系統都會在顯示完成後結束流程。

```

run_crawler.bat
1  @echo off
2  setlocal
3
4  :: === 依你的環境調整 ===
5  set "PY=C:\Users\Mizuyu\.conda\envs\myenv\python.exe"
6  set "APPDIR=D:\code\myproject"
7  set "MANAGE=%APPDIR%\manage.py"
8  set "LOG=%APPDIR%\scheduler.log"
9  set "ARGS=crawl_rice --pages=2"
10
11  echo [%date% %time%] ==== start ==== >> "%LOG%"
12
13  if not exist "%MANAGE%" (
14      echo manage.py not found: %MANAGE% >> "%LOG%"
15      exit /b 1
16  )
17
18  cd /d "%APPDIR%" || (
19      echo CD failed: %APPDIR% >> "%LOG%"
20      exit /b 1
21  )
22
23  "%PY%" "%MANAGE%" %ARGS% >> "%LOG%" 2>&1
24  set "EC=%errorlevel%"
25  echo [%date% %time%] exitcode=%EC% ==== end ==== >> "%LOG%"
26  exit /b %EC%
27

```

▲圖 7 爬蟲



▲圖 8 排程器

圖7，是我們用來自動執行 Django 專案的爬蟲程式的批次檔。 它的目的，是讓我們不用每次都手動開啟 Python 或 Django 指令，而是雙擊檔案即可自動啟動整個爬蟲流程，並且把執行過程與錯誤紀錄下來，方便日後除錯。

圖8，我們使用windows的工作排程器每三小時自動執行圖7的程序以確保資料能即時更新。

```
"2025/10/14": [  
  {  
    "市場": "基隆市",  
    "零售": {  
      "粳種白米": 50.95,  
      "硬秈白米": 55.0,  
      "軟秈白米": 0.0,  
      "圓糯白米": 64.29,  
      "長糯白米": 64.17  
    },  
    "躉售": {  
      "粳種白米": 4430.0,  
      "硬秈白米": 4833.0,  
      "軟秈白米": 0.0,  
      "圓糯白米": 5375.0,  
      "長糯白米": 5333.0  
    }  
  },  
]
```

▲圖 9 爬蟲數據

圖9，為爬蟲將數據轉換成json檔的格式，紀錄各縣市的稻米，如粳種白米、硬秈白米等多種稻米之零售價以及躉售價

```

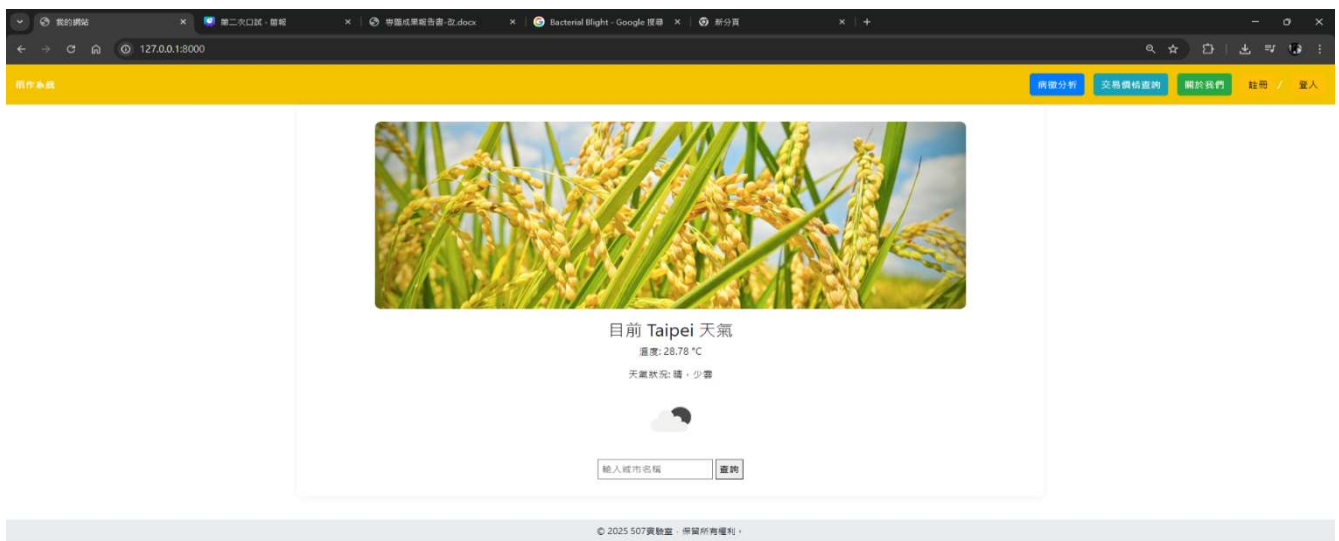
34
35 def home(request):
36     city = request.GET.get('city', 'Taipei')
37     api_key = 'bd7dfbca3f211ee414f9fc82e4a68c91'
38
39     url = f'https://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key}&units=metric&lang=zh_tw'
40
41     response = requests.get(url)
42     weather_data = response.json()
43
44     if response.status_code == 200 and weather_data.get('cod') == 200:
45         context = {
46             'city': weather_data.get('name', city),
47             'temperature': weather_data['main']['temp'],
48             'description': weather_data['weather'][0]['description'],
49             'icon': weather_data['weather'][0]['icon'],
50             'error': None
51         }
52     else:
53         context = {
54             'city': city,
55             'temperature': None,
56             'description': None,
57             'icon': None,
58             'error': '找不到該城市，請重新輸入！'
59         }
60
61     return render(request, 'home.html', context)
62

```

▲圖 10 天氣

圖10，在我們的 Django 專案中，這段 home(request) 程式負責實作整個天氣查詢的主要功能。當使用者進入網頁並輸入城市名稱後，我們的程式會透過 HTTP 請求與 OpenWeatherMap API 進行連線，以取得即時的天氣資訊。首先，我們使用request.GET.get('city', 'Taipei')從使用者的網址參數中讀取城市名稱，如果使用者沒有輸入任何城市，系統會自動預設查詢

「Taipei」。接著我們設定好 API 金鑰 api\_key 以及查詢網址 url，網址中包含城市名稱、授權金鑰、溫度單位（以攝氏為主）以及語言設定（顯示繁體中文）。這樣設計的目的是讓使用者可以靈活查詢各個區域的天氣。



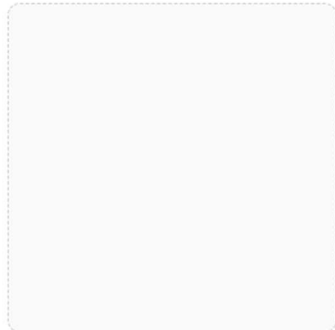
▲圖 11 主頁面

圖11，是我們的「稻米病害辨識系統」的主頁介面，整體設計以「清楚呈現農業主題、友善使用者操作」為核心概念。我們在頁面上方設置了醒目的黃色導覽列，作為系統標題區域，象徵稻穗與農作的色調，同時讓整體風格更具農業意象。中央主視覺區域放置一張稻田或稻穗的圖片，作為系統的封面背景，讓使用者一進入網站就能感受到這是以稻米為核心的辨識平台。下方的主要內容區塊則是天氣查詢系統，詳細內容於第14頁有詳細的介紹其工作原理，在用戶介面上可以在搜尋欄中輸入使用者欲查詢之城市並按下查詢，就會顯示該城市的天氣情報。

## 上傳稻葉圖片進行病害辨識

選擇檔案 未選擇任何檔案

辨識



### 辨識結果：

- Blast
- Brownspot

### 建議解決方法

#### • 稻熱病 (Rice Blast)

##### ▼ 處理建議

1. 發病初期應及時施用對應藥劑（如三環唑、稻瘟靈等），依標籤建議劑量施用。
2. 改善田間排水、避免過度潮濕環境。
3. 發病嚴重時移除病株並清除殘體，避免病原蔓延。

##### ▶ 預防建議

#### • 褐斑病 (Brown Spot)

##### ▼ 處理建議

1. 移除嚴重受害葉片並清園，避免病原殘留。
2. 改善田間通風與日照，降低葉面潮濕時間。
3. 必要時依標籤規定施用對應藥劑（例如：苯醚甲環唑類）。

▲圖 12 分析畫面

## 上傳稻葉圖片進行病害辨識

選擇檔案 未選擇任何檔案

辨識

### 辨識結果：

沒有偵測到任何病害，請換一張圖片試試。

### 辨識歷史紀錄

清除歷史紀錄

- [2025/10/21 上午10:55:59] (未取得檔名) → 沒有偵測到任何病害
- [2025/10/21 上午10:55:14] common\_bacterial\_blight\_of\_beans\_featured.jpg → 沒有偵測到任何病害
- [2025/10/21 上午10:52:32] Bacterial-Blight-angular-shape.jpg → Brownspot, Blast
- [2025/10/21 上午10:52:23] Bacterial\_Blight\_of\_Soybean.jpg → 沒有偵測到任何病害
- [2025/10/21 上午10:52:16] common\_bacterial\_blight\_of\_beans\_featured.jpg → 沒有偵測到任何病害
- [2025/10/21 上午10:50:38] Bacterial\_Blight\_of\_Soybean.jpg → 沒有偵測到任何病害

▲圖 13 歷史紀錄



圖12，這個畫面是我們「稻米病害辨識服務」的頁面，主要功能是讓使用者上傳稻葉圖片，並辨識發生的病害種類。整體介面設計簡潔明瞭，方便農民、研究人員或一般使用者快速操作。使用者只需點擊「選擇檔案」按鈕上傳一張稻葉照片，系統就會自動啟動後端的深度學習模型（YOLO 模型）進行影像分析。模型會根據訓練過的資料集判斷稻葉表面的病斑特徵，辨識出可能的病害種類，例如「Blast（稻熱病）」與「Brown Spot（褐斑病）」等，並將結果以條列方式清楚呈現在「辨識結果」區塊中。在辨識結果下方的「建議解決方法」區域中，系統會根據不同的病害種類，自動顯示對應的防治建議與管理措施。以「稻熱病」為例，說明欄中會列出建議的用藥時機與防治策略，例如使用對應的殺菌劑、注意施肥比例與水分管理等。

圖13，下方有辨識的歷史紀錄，記錄著辨識圖片之檔名以及其偵測出來的病徵種類，該紀錄為瀏覽器的紀錄在更換裝置或是刪除瀏覽器的歷史紀錄時會消失，若想主動刪除也可以手動點擊旁邊的「清除歷史紀錄」按鈕來刪除。

### 全國稻米價格行情

資料更新時間：2025-10-13T16:11:24.421950+00:00

2025/10/13

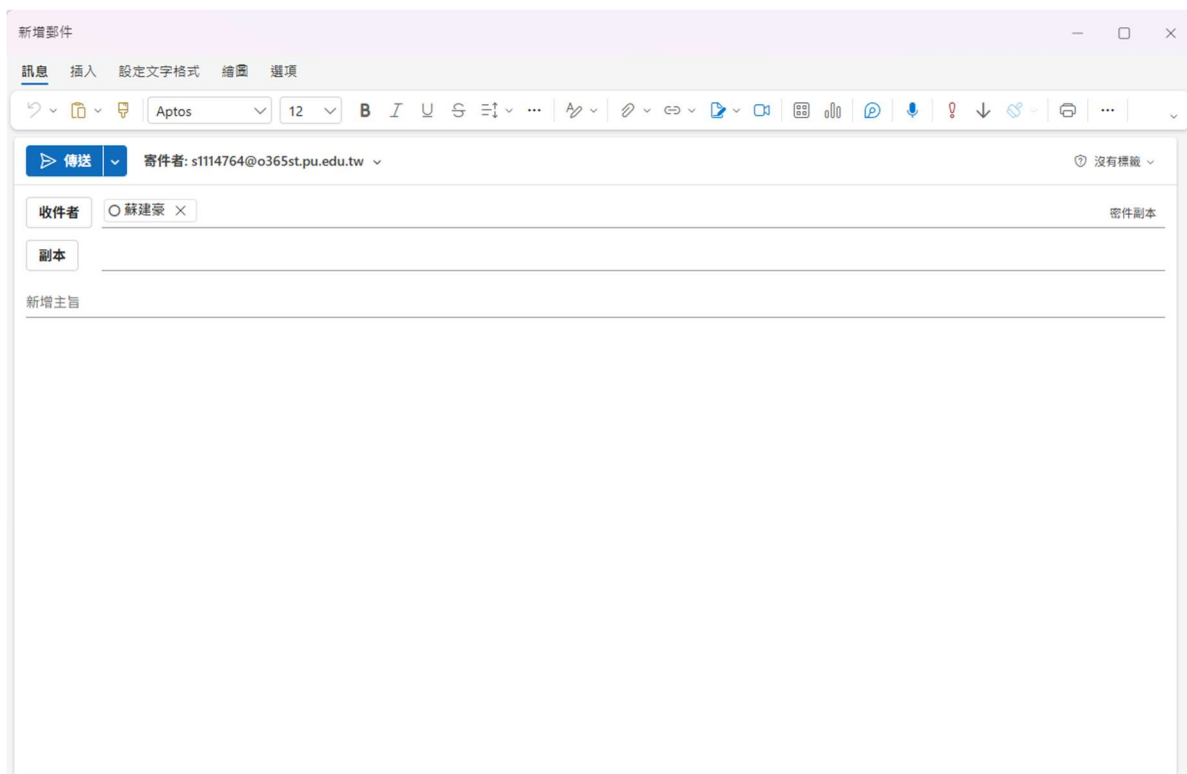
交易日期	市場	零售 (元/公斤)					躉售 (元/百公斤)				
		縹絲白米	縹絲白米	軟粒白米	圓粒白米	長粒白米	縹絲白米	縹絲白米	軟粒白米	圓粒白米	長粒白米
2025/10/13	基隆市	50.95	55.0	0.0	64.29	64.17	4430.0	4833.0	0.0	5375.0	5333.0
2025/10/13	台北市	51.88	57.5	62.92	65.83	63.13	4386.0	4572.0	5084.0	5500.0	5427.0
2025/10/13	新北市	48.74	54.03	57.96	58.57	58.24	4337.0	4542.0	5204.0	5246.0	5204.0
2025/10/13	桃園市	43.49	47.78	56.25	54.79	55.0	3824.0	4167.0	4917.0	4833.0	4833.0
2025/10/13	新竹市	45.75	50.83	0.0	54.98	53.61	4063.0	4345.0	0.0	4905.0	4517.0
2025/10/13	新竹縣	44.58	48.0	0.0	54.37	53.33	3940.0	4217.0	0.0	4903.0	4861.0
2025/10/13	苗栗縣	44.18	0.0	0.0	54.46	54.92	3972.0	0.0	0.0	4833.0	4778.0
2025/10/13	台中市	46.19	54.57	55.51	56.72	56.94	3883.0	4425.0	5113.0	4898.0	4889.0
2025/10/13	彰化縣	44.82	48.39	52.65	53.67	53.47	3907.0	4155.0	4927.0	4718.0	4721.0
2025/10/13	南投縣	46.68	51.63	55.0	56.03	55.55	4000.0	0.0	0.0	4583.0	4250.0
2025/10/13	雲林縣	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2025/10/13	嘉義市	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2025/10/13	嘉義縣	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2025/10/13	台南市	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2025/10/13	高雄市	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2025/10/13	屏東縣	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2025/10/13	宜蘭縣	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2025/10/13	花蓮縣	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2025/10/13	台東縣	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2025/10/13	澎湖縣	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

▲圖 14 米價查詢

圖14，為稻米的價格查詢頁面，在這裡會顯示每天最新的稻米價格，此分頁依靠圖7的主程式來爬取稻米價格資訊並儲存為如圖9的.json檔，最後以表格的方式呈現給使用者。



▲ 圖 15 關於我們

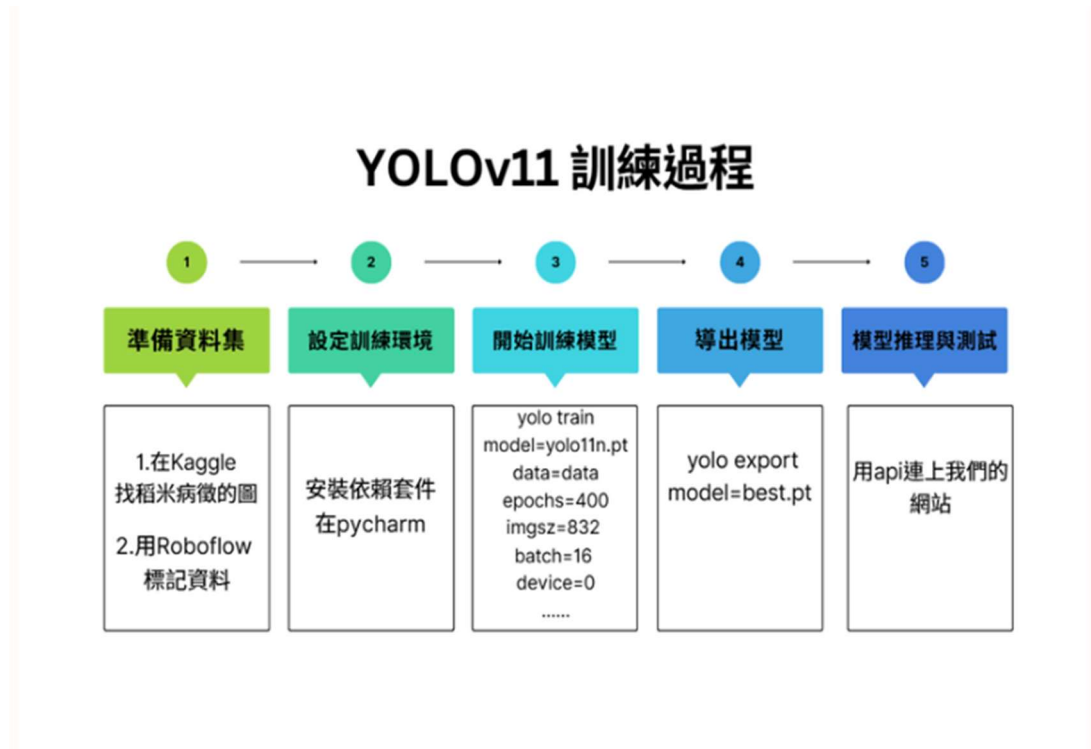


▲ 圖 16 關於我們

圖15，有開發者的相關資訊(姓名及連繫郵箱)，點擊信箱可以直接跳轉到寄送信件的畫面，如圖16(每個裝置頁面可能有所差異，這裡以桌面端outlook為例圖)會直接自動輸入所點擊的郵箱，無須自己輸入郵箱直接輸入內容即可，主要供使用者回饋或是回報問題等

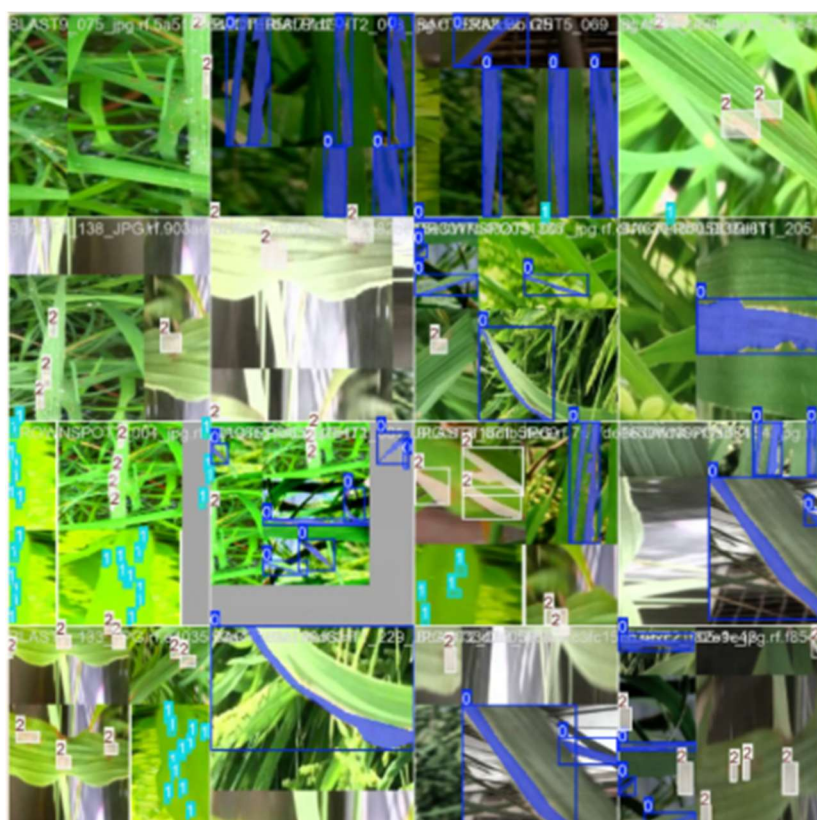
## 第四章 模型介紹

### 壹、YOLOv11 訓練過程

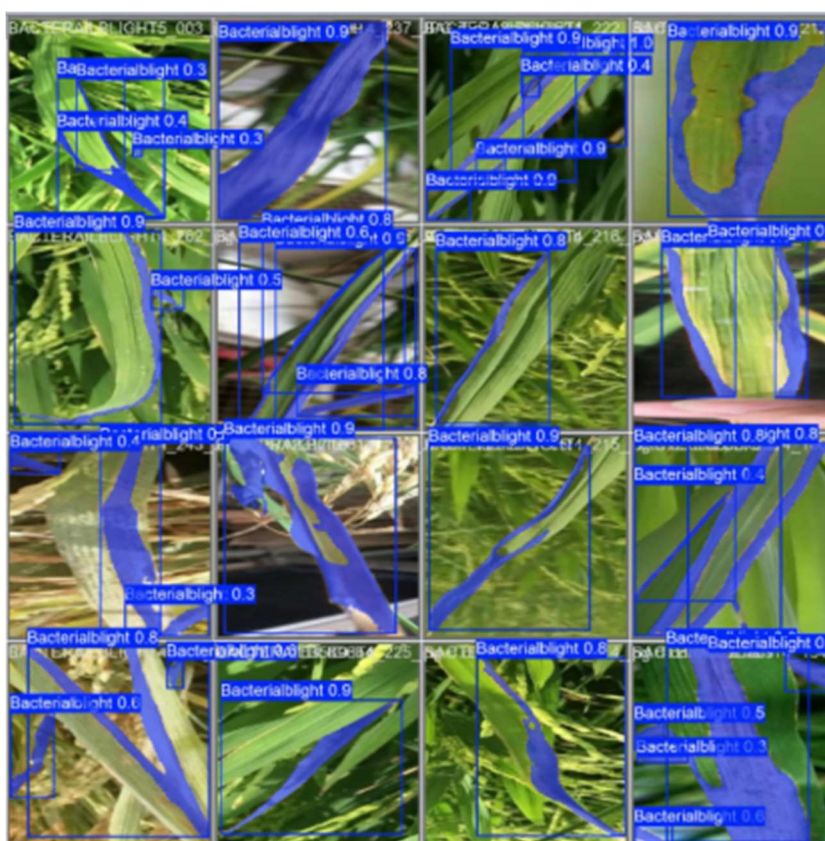


▲圖 17 YOLOv11 訓練過程

1. 準備資料集
  - Kaggle 找稻米病徵的圖
  - Roboflow 標記資料
2. 設定訓練環境
  - 裝依賴套件在 pycharm
3. 開始訓練模型
4. 導出模型
5. 模型推理與測試



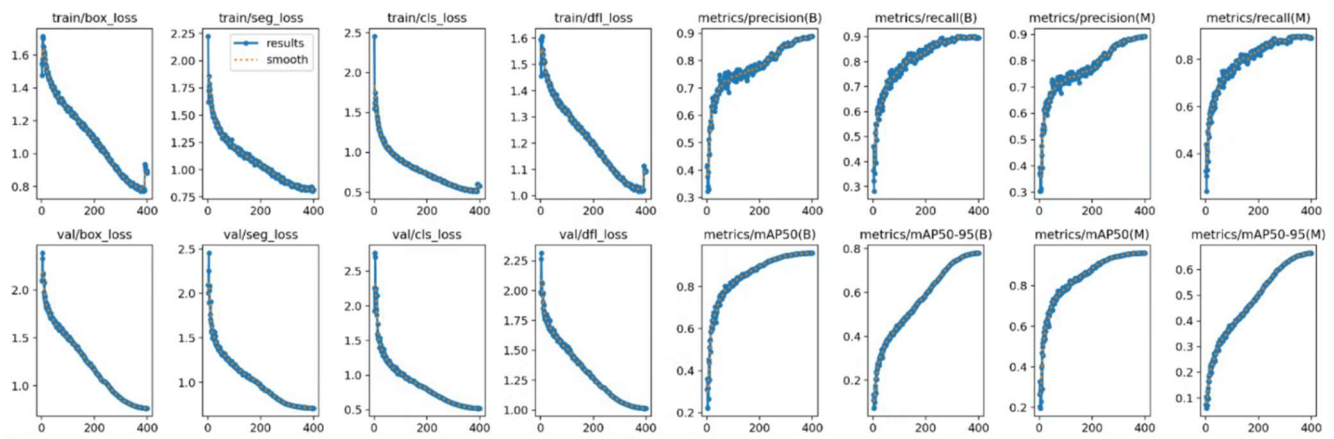
▲圖 18 訓練過程



▲圖 19 訓練過程

## 貳、模型表現：

### 一、Results - 整體模型表現



▲圖 20 模型表現

整體模型表現：

mAP@0.5 : 0.961 mAP@0.5 - 0.95 : 0.664 Precision : 0.893 Recall : 0.893 整體時間190hr

- mAP (mean Average Precision):物件檢測模型的主要效能指標,衡量模型在不同 IoU 門檻下的平均精度。
- Precision(精確率):指模型判斷為缺陷的樣本中,實際正確的比例。
- Recall(召回率):指所有實際缺陷中,模型能正確檢出的比例。
- F1-score:精確率與召回率的調和平均數,用於綜合評估模型效能。
- Confusion Matrix(混淆矩陣):用於檢視分類模型效能的表格,顯示正確與錯誤分類的數量。



## 二、Results - 模型整體結論

```

400 epochs completed in 190.961 hours.
Optimizer stripped from C:\Users\user\PycharmProjects\rice leaf diseases dataset\runs\exp_seg_42\weights\last.pt, 45.3MB
Optimizer stripped from C:\Users\user\PycharmProjects\rice leaf diseases dataset\runs\exp_seg_42\weights\best.pt, 45.3MB

Validating C:\Users\user\PycharmProjects\rice leaf diseases dataset\runs\exp_seg_42\weights\best.pt...
Ultralytics 8.3.121 Python-3.11.9 torch-2.7.0+cu118 CUDA:0 (NVIDIA GeForce RTX 3060, 12288MiB)
YOLO11m-seg summary (fused): 138 layers, 22,337,625 parameters, 0 gradients, 123.0 GFLOPs

```

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95)	Mask(P)	R	mAP50	mAP50-95)	0%
Class	Images	Instances	Box(P)	R	mAP50	mAP50-95)	Mask(P)	R	mAP50	mAP50-95)	100%
all	539	1742	0.888	0.894	0.96	0.779	0.893	0.893	0.961	0.664	
Bacterialblight	171	394	0.86	0.932	0.968	0.866	0.867	0.931	0.971	0.692	
Brownspot	175	591	0.911	0.927	0.973	0.751	0.915	0.925	0.973	0.693	
Blast	193	757	0.893	0.824	0.937	0.722	0.897	0.823	0.939	0.607	

▲圖 21 模型整體

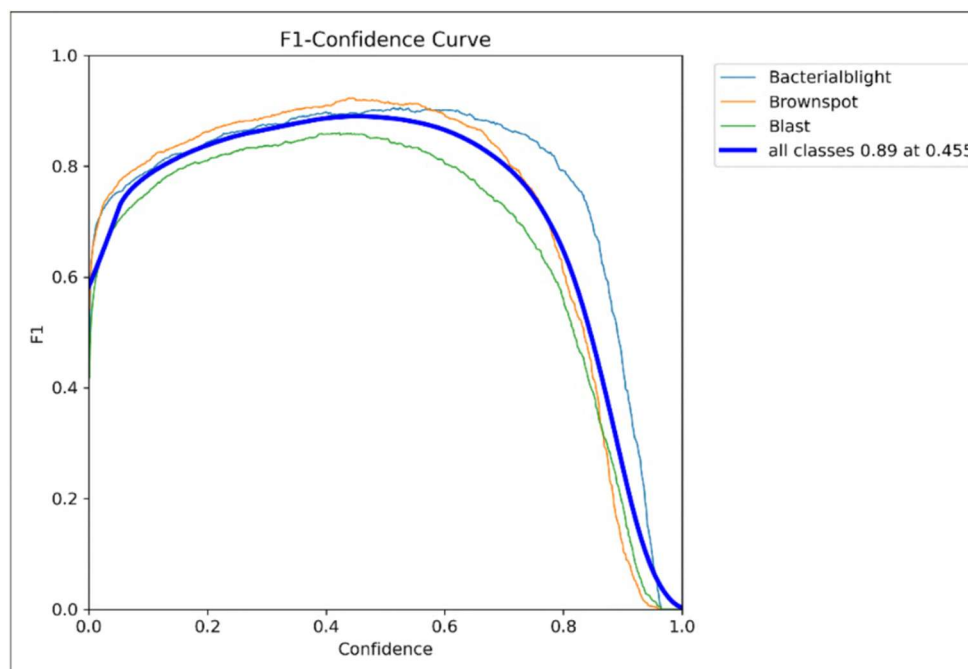
類別	Precision	Recall	mAP@0.5	mAP@0.5-0.95
Bacterialblight	0.86	0.932	0.968	0.866
Brownspot	0.911	0.927	0.973	0.751
Blast	0.893	0.824	0.937	0.722

▲圖 22 模型整體

### 模型整體結論

- 模型對三種稻葉病害皆能準確分割與辨識，整體檢測表現優異。
- Brownspot 類別的表現最穩定；
- Blast 則因外觀變化大、邊界模糊，導致 Recall 略低。

### 三、BoxF1\_curve (F1-Confidence Curve)



▲圖 23 BoxF1\_curve

圖23，顯示的是 YOLO 模型在不同「信心閾值」下的 F1 分數變化曲線。F1 值是 Precision（精確率）與 Recall（召回率）的調和平均，代表模型在「找對」與「不漏找」之間的整體平衡表現。

圖形解讀：

橫軸（X 軸）：模型預測的信心閾值（Confidence Threshold）

→ 代表模型判斷「這個框有多大機率是真的」。

縱軸（Y 軸）：F1 分數（越高代表模型平衡越好）

三條曲線：分別代表三個病害類別的表現（Bacterial blight、Brown spot、Blast）。

主要觀察結果：

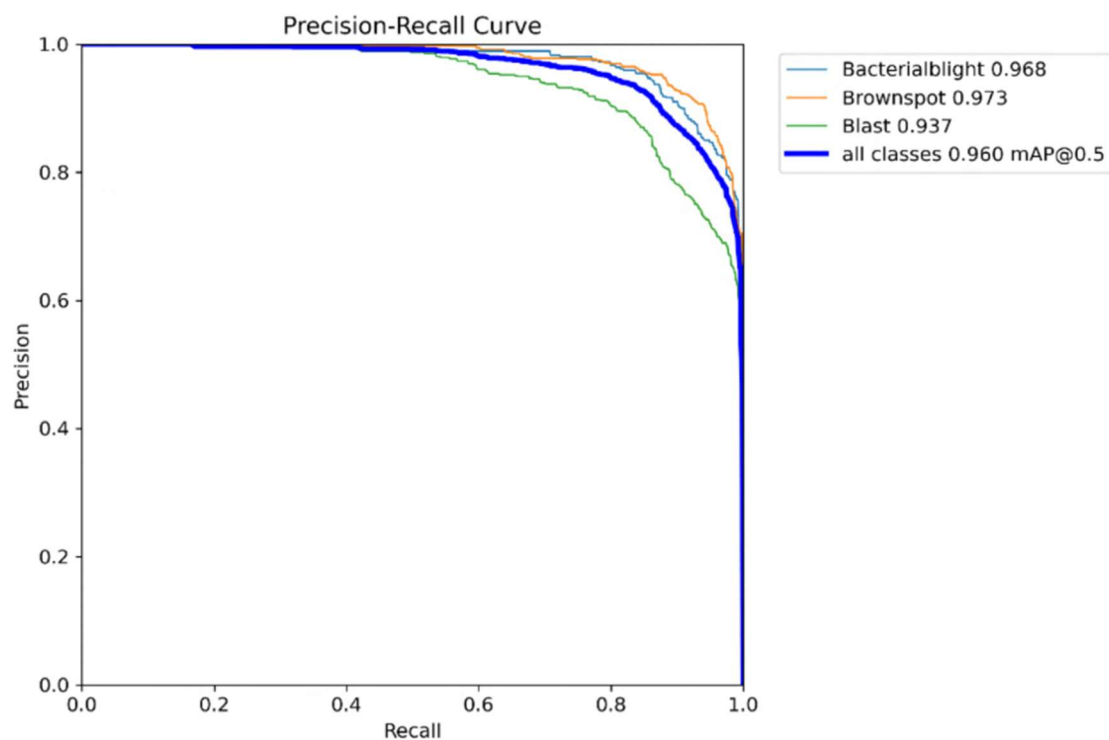
1. 平均 F1 值：約 0.89  
→ 模型整體在 Precision 與 Recall 之間的平衡非常好。
2. 最佳信心閾值：約 0.455  
→ 當信心值設定在約 0.45 時，模型達到整體最佳 F1 表現。
3. 曲線平穩、無劇烈震盪  
→ 表示模型在不同信心條件下表現穩定，不容易受閾值設定影響。
4. 三類別趨勢相近  
→ 模型對不同病害的偵測一致性高，沒有明顯偏好或弱項。



結論：

- ＞ 從 BoxF1\_curve 可以看出，本模型在各病害類別上都維持穩定的 F1 表現，
- ＞ 平均 F1 約為 0.89，代表模型具有良好的精確度與召回率平衡。
- ＞ 當信心閾值設定在約 0.45 時，可獲得最佳偵測效果。
- ＞ 整體曲線平滑，顯示模型具有良好的穩定性與泛化能力。

#### 四、BoxPR\_curve



▲圖 24 BoxPR\_curve

#### PR\_curve

評估內容:在各閾值下的 P-R 關係

用途:顯示整體檢測穩定性

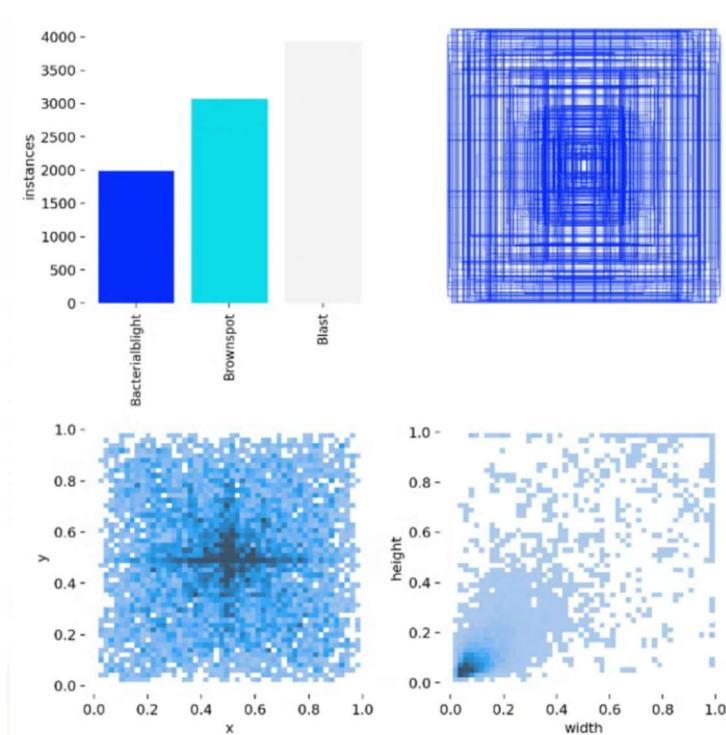
#### 各類別表現:

1. Brownspot:  $AP = 0.973$ , 最佳表現, 分類穩定。
2. Bacterialblight:  $AP = 0.968$ , Precision 與 Recall 均衡。
3. Blast:  $AP = 0.937$ , 在高 Recall 時 Precision 稍降。

#### 結論:

模型對 Brownspot、Bacterialblight 效果佳, Blast 類別仍可加強高召回率下的 Precision。

## 五、labels



▲圖 25 labels

圖25，展示了數據集中不同病害類別的標註數量、標註框位置與大小分布情況。

整體資料分布：

三個病害類別的樣本數量分布不均，Blast 樣本最多，Bacterialblight 最少。

大部分標註框（bounding boxes）集中在影像中央，並且尺寸偏小，顯示病斑多出現在葉片中央區域，且範圍較有限。

類別差異：

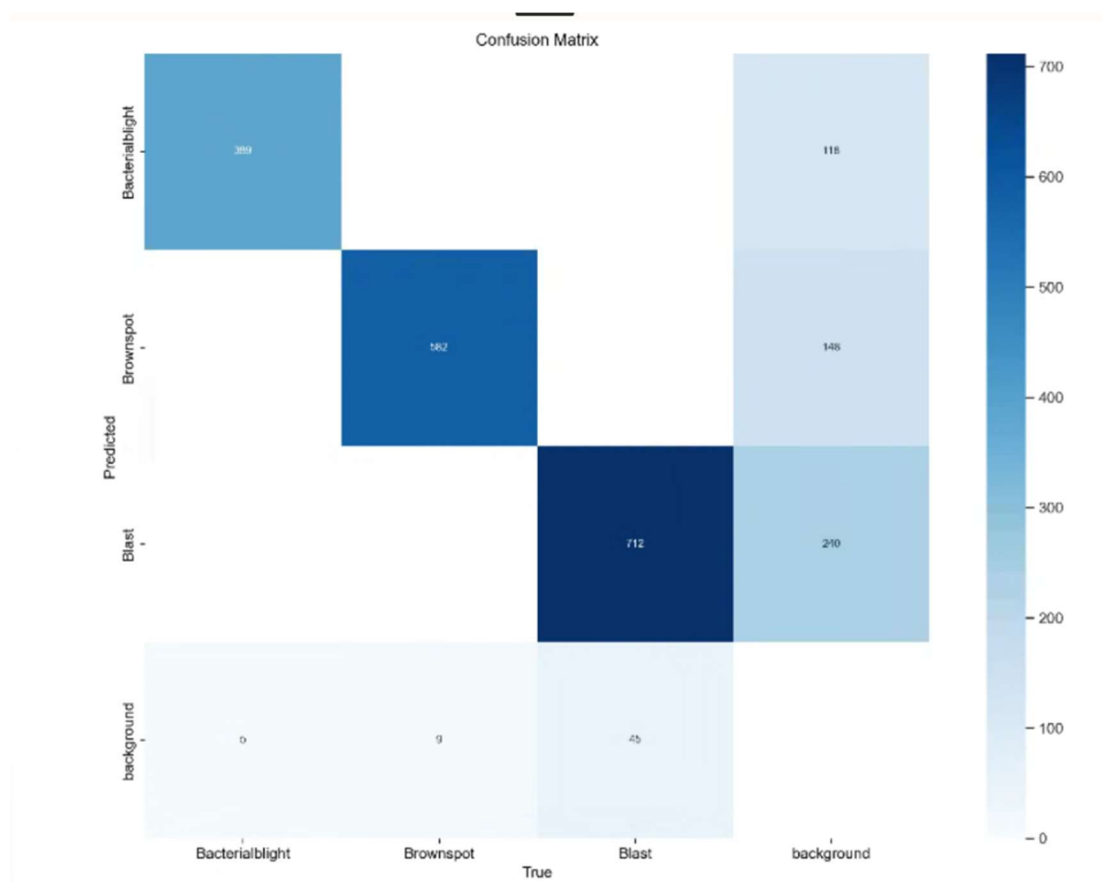
- Bacterialblight：樣本數約 2000，為三類中最少，數據量不足可能導致模型對該類別的泛化能力稍弱。
- Brownspot：樣本數約 3000，數據量適中，標註框分布較均勻，利於模型學習。
- Blast：樣本數約 3800，雖然數量最多，但其檢測表現（F1 與 Recall）仍較差，顯示僅增加數量不足以完全解決辨識困難，可能與樣本特徵模糊或與背景干擾有關。

標註框位置與大小：

標註框的中心點（x, y）多集中於影像中央（0.4 – 0.6 區間），反映拍攝時葉片多在畫面正中。

寬度與高度多小於 0.2，顯示大部分病斑區域相對於整張影像比例較小。

## 六、confusion matrix



▲圖 26 confusion matrix

圖26，展示了模型在各類別的實際標籤與預測結果之間的混淆情況。  
整體模型表現：大部分樣本被正確分類，但仍存在部分與背景（background）混淆的情況。

類別差異：

- Bacterialblight：正確分類數 389，但有 118 張被誤判為背景。
- Brownspot：正確分類數 562，但有 148 張被誤判為背景。
- Blast：正確分類數 712，但與背景混淆最嚴重，有 210 張被錯誤分類。

結論：

模型在 Blast 類別上與背景的混淆最明顯，這也是 Recall 偏低的主要原因。建議補充更多背景負樣本或針對 Blast 特徵進行資料增強，以減少誤判。

## 第五章 補充說明

### 壹、遇到的困難

整體的模型準確率和召回率都偏低。

改善方式：

1. 重新標註（全改成segmentation的格式）
2. 換模型(換成yolo11l-seg.pt)

roberflow的資料格式有問題

因為我們是開三個專案做，所以label的資料會混亂(image跟label對不到)

改善方式:寫一隻轉換程式，讓label分三類



圖 28 原本標記方式  
Bounding Box（矩形框）

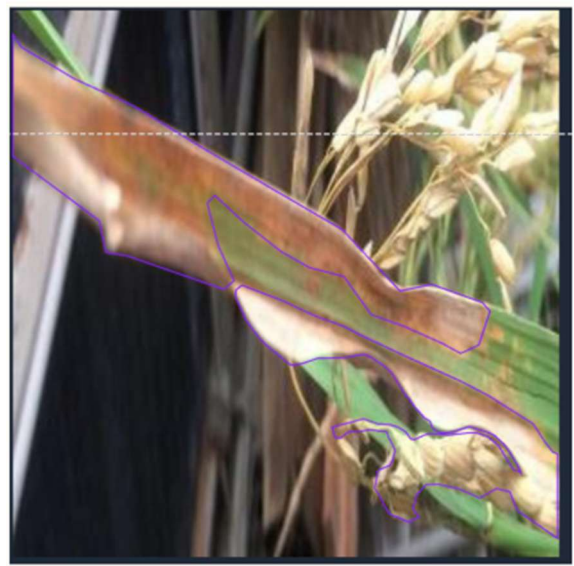


圖 27 改善後的標記方式  
segmentation

### 貳、結論及未來發展

#### 一、結論：

- 偵測準確率高，模型穩定，網頁辨識功能正常

#### 二、短期目標：

- 增加改善圖片解析度功能
- 優化模型結構

#### 三、長期目標：

- 擴展更多種類的稻葉病害辨識能力

### 參、經費預算需求表（執行中所需之經費項目單價明細）

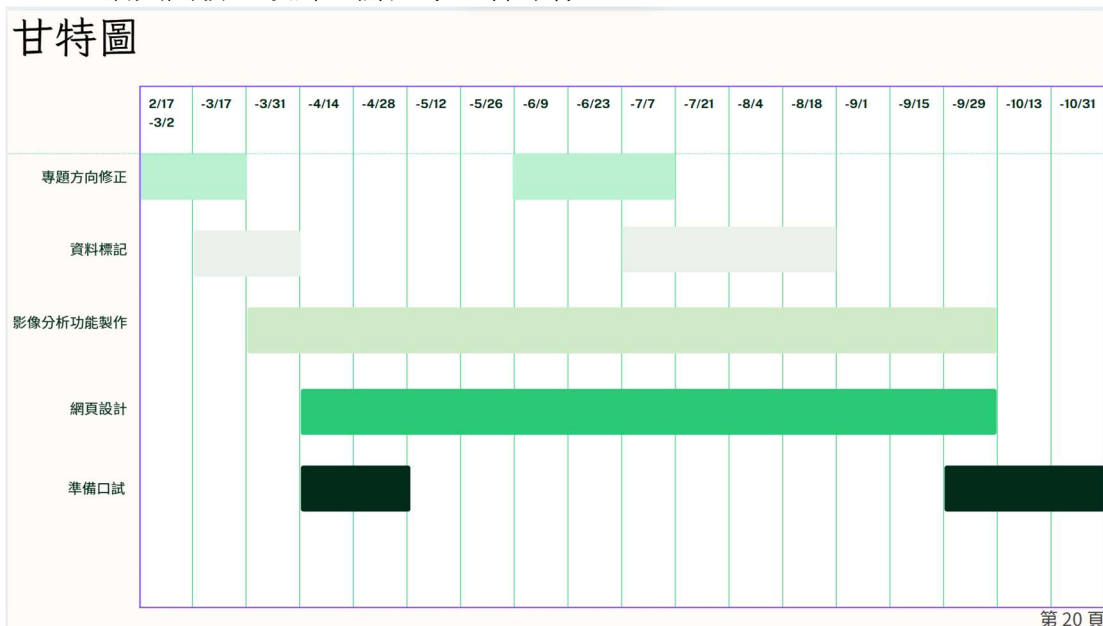
#### 編列預算範本

項目名稱	說明	單位	數量	單 價	小 計	備 註
				臺幣(元)	臺幣(元)	
個人電腦	專案之進行	部	1	26000	26000	由系上實驗室提供
雷射印表機	列印所需文件	部	1	100	100	由系上實驗室提供
消耗性器材	印表機消耗材料、紙張等	批	1	200	200	由系上實驗室提供
消耗性器材	光碟片、隨身碟、外接硬碟等	批	1	300	300	自行負擔
共 計					26600	

### 肆、工作分配（詳述參與人員分工）

AI模型訓練、製作流程圖：許秉豐、蔡尚廷

網頁開發、設計：蘇建豪、林泳存



▲圖 29 甘特圖

## 伍、名詞解釋

### 交集與聯集 (IoU)

真陽性 (TP, True Positive): 正確檢測目標。

假陽性 (FP, False Positive): 錯誤檢測目標 (誤報)。

假陰性 (FN, False Negative): 遺漏目標 (漏報)。

### 精確度 (Precision)

$$\text{Precision} = \text{TP}(\text{正確}) / (\text{TP}(\text{正確}) + \text{FP}(\text{誤報}))$$

### 召回率 (Recall)

$$\text{Recall} = \text{TP}(\text{正確}) / (\text{TP}(\text{正確}) + \text{FN}(\text{漏報}))$$

### AP (Average Precision, 平均精確度)

定義: 對於單一類別, 計算其 Precision-Recall 曲線下的面積。

### mAP (mean Average Precision, 平均均值精確度)

定義: 將所有類別的 AP 值計算出來後, 取其平均值。(衡量模型在所有類別上的整體性能)

## 陸、參考資料

1. Ultralytics YOLO 文件  
<https://docs.ultralytics.com/zh/guides/yolo-performance-metrics/>
2. 訓練週期(epoch) 與 批次(batch)  
<https://ithelp.ithome.com.tw/articles/10298302>
3. epoch和learning rate的通常取值范围？  
<https://blog.csdn.net/qlkaicx/article/details/135886572>
4. 模型评估和微调的启示  
<https://docs.ultralytics.com/zh/guides/model-evaluation-insights/>
5. 發現稻熱病108年  
<https://www.agriharvest.tw/archives/123388>
6. 褐斑病  
[https://www.aphia.gov.tw/Publish/plant\\_protect\\_pic\\_6/orchidPDF/02-7.pdf](https://www.aphia.gov.tw/Publish/plant_protect_pic_6/orchidPDF/02-7.pdf)
7. 病蟲害管理  
[https://kmweb.moa.gov.tw/redirect\\_files.php?theme=knowledgebase&id=457001](https://kmweb.moa.gov.tw/redirect_files.php?theme=knowledgebase&id=457001)
8. 植物葉子出現黑點、黑斑？葉斑病與褐斑病的病徵判斷與防治方法  
<https://www.nafertino.com.tw/technology/disease-black-spot.html>
9. 觀葉植物病害及防治  
<https://book.tndais.gov.tw/Brochure/tech102.htm>
10. 水稻白葉枯病1  
[https://www.aphia.gov.tw/Publish/plant\\_protect\\_pic\\_8/ricePDF/06-15.pdf](https://www.aphia.gov.tw/Publish/plant_protect_pic_8/ricePDF/06-15.pdf)
11. 水稻白葉枯病2  
<https://azai.tari.gov.tw/search/bug/detail?id=897>