PCB 瑕疵檢測

組員：蔡詳羿 (7113056077)

陳信宇 (7113056068)

目次

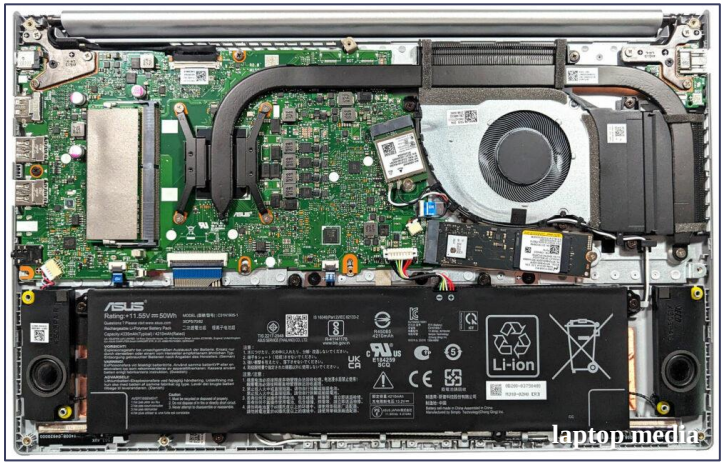
簡介

- 本專題使用 YOLO v11 s 深度學習模型，進行印刷電路板（PCB）瑕疵檢測，mAP50 可達 0.995，並且推論速度極快。
- 相較於傳統的自動光學檢查 (AOI) 檢測方法，深度學習方法是由電腦自動學習瑕疵特徵，不依靠由人定義的模板規則，因此能夠檢測出任意位置的瑕疵，泛用性高。現代 PCB 製造趨勢朝向多樣小量，並且逐漸複雜化，深度學習方法更能因應現代生產環境中快速變化的需求。
- 本專題製作了一款簡易的 PCB 瑕疵檢測程式，讓使用者能夠透過網頁瀏覽器上傳 PCB 圖片，即時獲得瑕疵檢測的結果。

動機 (1/4)

什麼是印刷電路板（PCB）？

- 電子設備的基礎組件
- 負責組織各種電子元件
- 在生活中隨處可見：
  - **電子產品**：電腦、手機
  - **家用電器**：冰箱、電鍋
  - **交通工具**：飛機、汽車

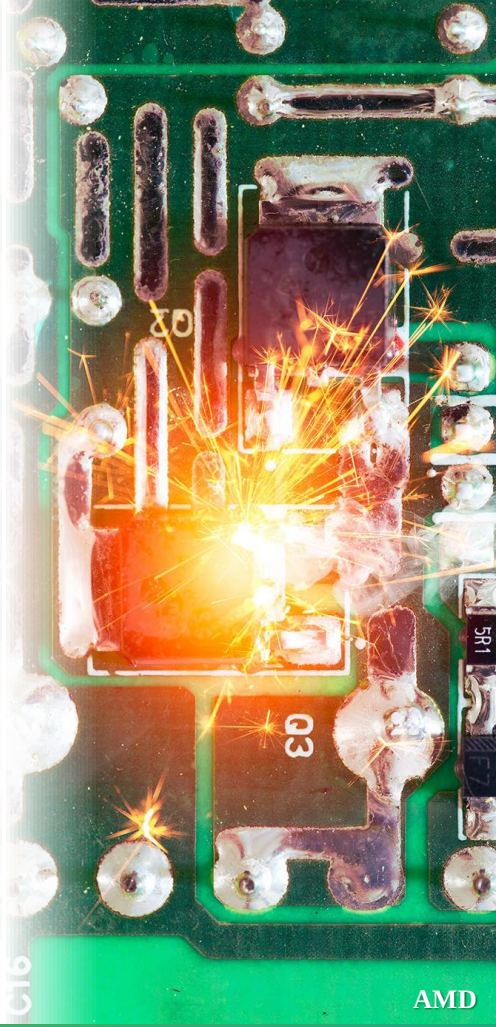


筆記型電腦內部及主機板

動機 (2/4)

PCB 的製造過程中，難免會出現瑕疵

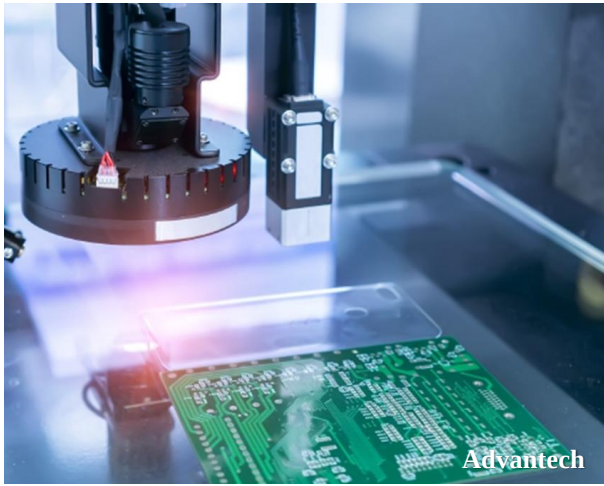
- **瑕疵的種類：**
  - 短路、開路、焊點異常等
- **瑕疵的影響：**
  - **設備功能失效**：設備無法正常運作
  - **營運成本上升**：維修、RMA
  - **造成安全問題**：導致使用者受傷、引發火災



動機 (3/4)

如何降低瑕疵造成的負面影響？

- **在製造過程中進行瑕疵檢測**
- **傳統方法：自動光學檢查 (AOI)**
  - **原理**：將 PCB 照片與模板進行比對
  - **優點**：自動化且快速，適合大批量檢測
  - **缺點**：依賴於模板規則，靈活性不足



動機 (4/4)

現代 PCB 的製造趨勢：

- 少樣大量 → **多樣小量**
- 設計簡單 → **設計複雜**

AOI 傳統方法面臨的問題：

- 基於模板規則，難以應對多樣化的 PCB 型式
- 由人定義規則，難以涵蓋所有瑕疵情況

目前需要一種新式的檢測方法，應對現代 PCB 的製造趨勢

目的

使用基於深度學習的物件辨識方法，

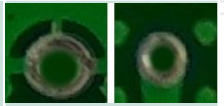
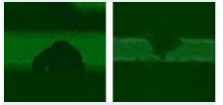
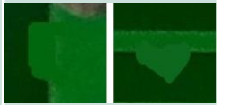
進行 PCB 瑕疵檢測，來實現：

- 不依賴模板規則
- 由電腦自動尋找特徵
- 可檢測出任意位置的瑕疵
- 提升檢測工具的泛用性

方法 (1/7)

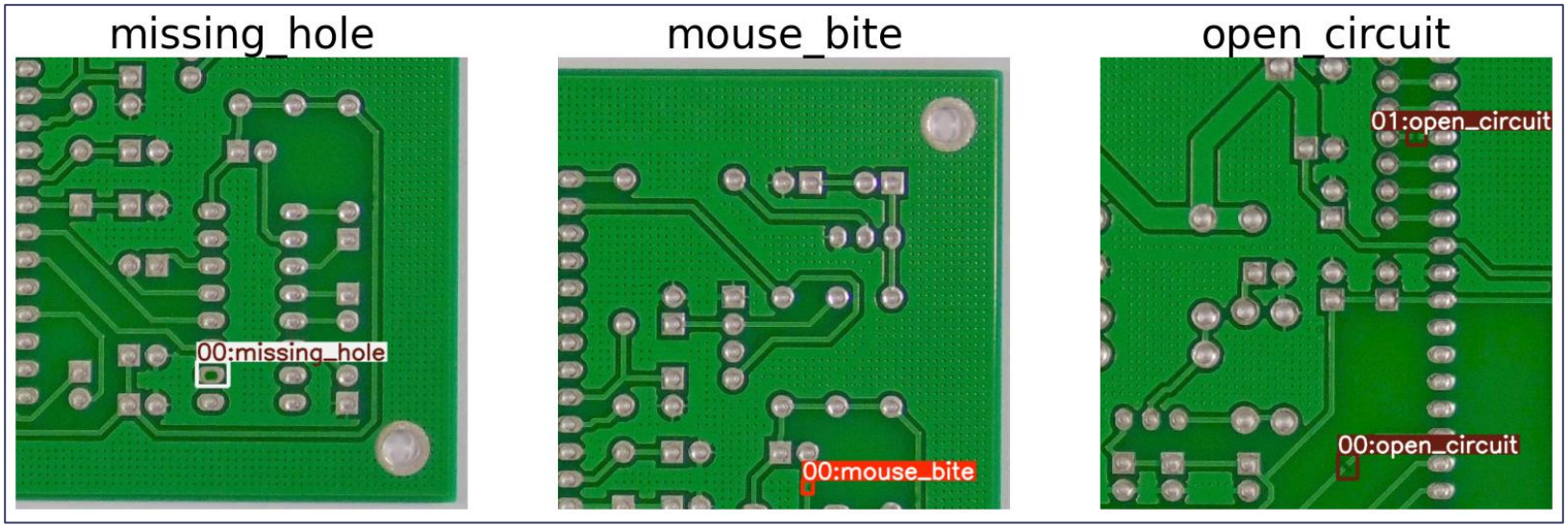
資料集介紹：

- **名稱：**PCB Defect
- **來源與作者：**
  - PKU-HRI
  - Norbert Elter
- **瑕疵種類：**共6種，見右表

瑕疵類別	圖片數量	瑕疵點數量	範例圖片
Missing Hole	1832	3612	
Mouse Bite	1852	3684	
Open Circuit	1740	3548	
Short	1732	3508	
Spur	1752	3636	
Spurious Copper	1760	3676	
總計	**10668**	**21664**	

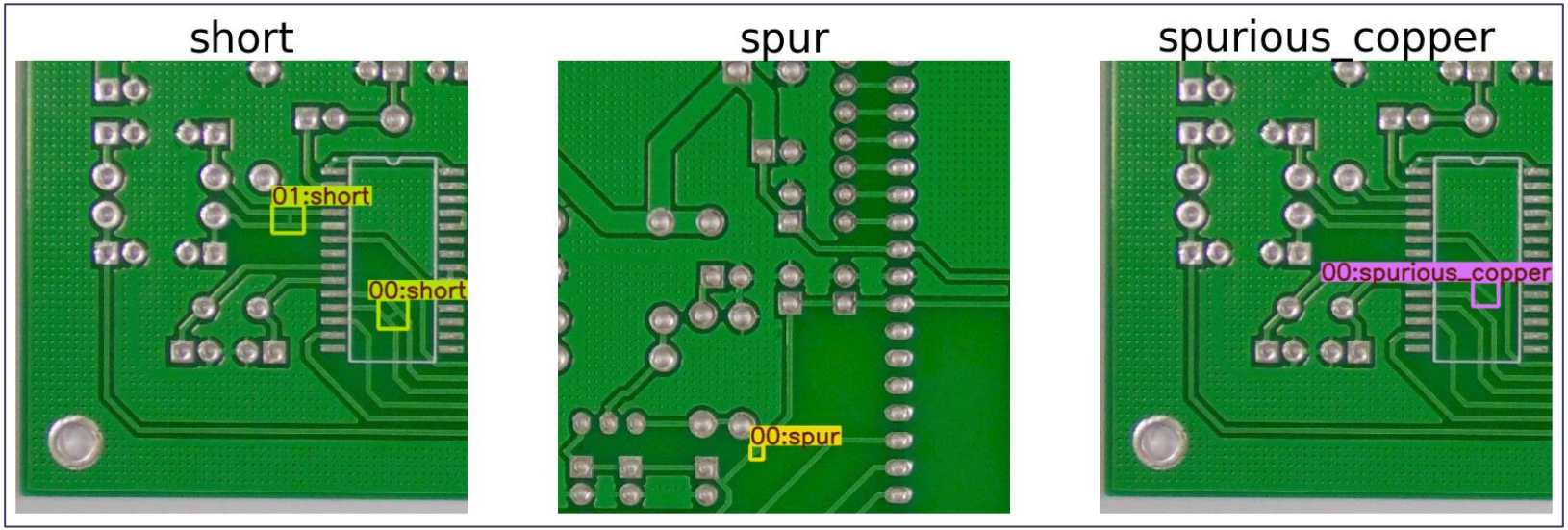
方法 (2/7)

PCB 完整圖：各種瑕疵在 PCB 上



方法 (3/7)

PCB 完整圖：各種瑕疵在 PCB 上



方法 (4/7)

資料增強

- **目的：**
  - 擴充資料的多樣性
  - 提升模型的泛化能力
- **本專題採用的方法：**
  - 翻轉、旋轉、縮放
  - 色調、亮度、飽和度
  - 裁剪、遮擋、線性變換

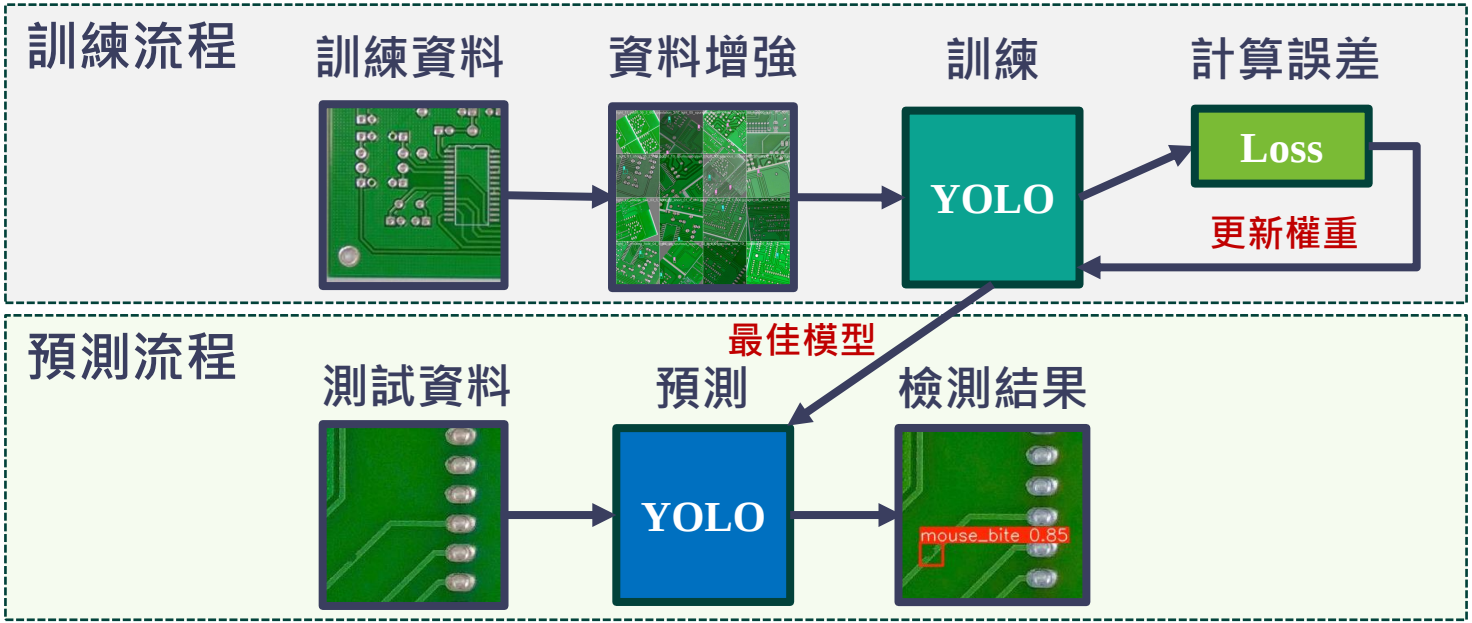
方法 (5/7)

PCB 瑕疵檢測模型：

- **模型：**YOLO v11 s
- **優點 (本專題使用此模型的原因)：**
  - **即時性：** 採用 Single stage 方法，使模型足夠輕量、推論快
  - **平移不變性：**使用 CNN 取特徵，能偵測出任意位置的物件
  - **易於訓練：**提供預訓練權重，能夠快速學習下游任務

方法 (6/7)

模型的訓練及預測流程：



方法 (7/7)

實驗環境

- **OS** ：Linux Mint 22
- **CPU** ：Intel Core i7-10700
- **GPU** ：NVIDIA TITAN RTX
- **RAM**：DDR4 125 GB

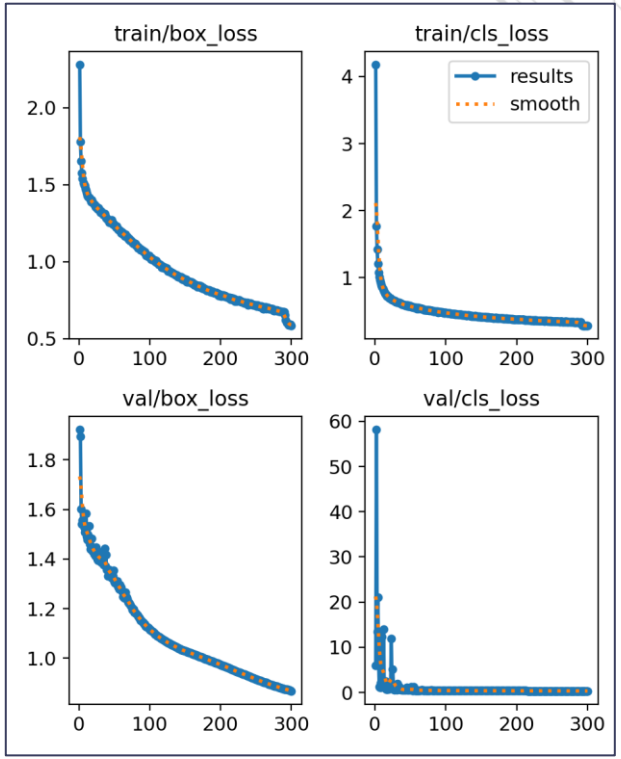
訓練參數設定

- **Batch size**：64
- **Epoch**：300
- **Patience**：10
- **Warm up**：3
- **Loss**： cls_loss、box_loss (ciou)、dfl_loss
- **Optimizer**：AdamW
- **Learning rate**：0.0001
- **Momentum**：0.9

實驗結果 (1/16)

分析訓練結果

- 完整訓練 300 Epochs，耗時 5 小時
- cls_loss 穩定收斂
- box_loss 扔持續下降
  - 推測是因為瑕疵點屬於小型物件，造成收斂較慢。



實驗結果 (2/16)

評估驗證集

- **驗證集**：包含 1066 張圖片、1595 個瑕疵點
- **推論時間**：2.3 ms per image

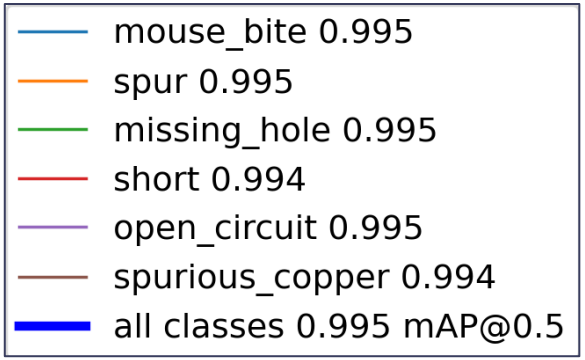
Class	Images	Instances	Box(P	R	mAP50	mAP50-95):
all	1066	1595	0.993	0.995	0.995	0.751
mouse_bite	140	280	0.996	1	0.995	0.768
spur	130	262	0.998	1	0.995	0.755
missing_hole	118	229	0.995	1	0.995	0.76
short	158	327	0.991	0.98	0.994	0.755
open_circuit	135	259	0.999	0.996	0.995	0.708
spurious_copper	121	238	0.979	0.991	0.994	0.763

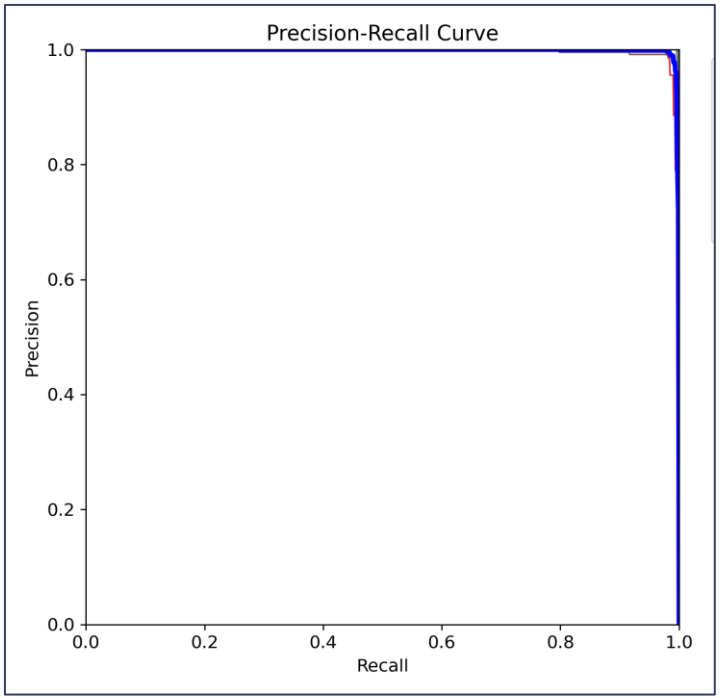
Speed: 0.2ms preprocess, 1.5ms inference, 0.0ms loss, 0.6ms postprocess per image

實驗結果 (3/16)

驗證集的 Precision-Recall Curve

- **mAP50**：0.995

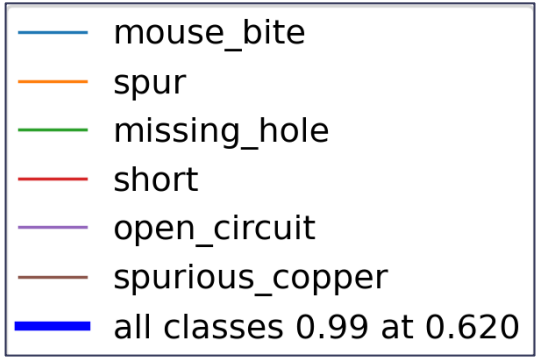


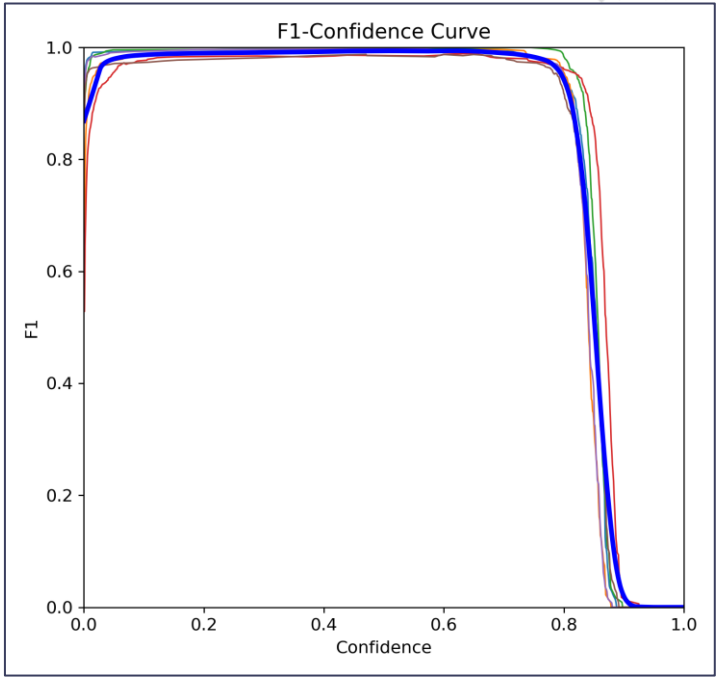


實驗結果 (4/16)

驗證集的 F1-Confidence Curve

- **最佳 Confidence 閾值：**0.62
- **F1 Score：**0.99





實驗結果 (5/16)

評估測試集

- **測試集**：包含 1068 張圖片、1662 個瑕疵點
- **推論時間**：3.9 ms per image

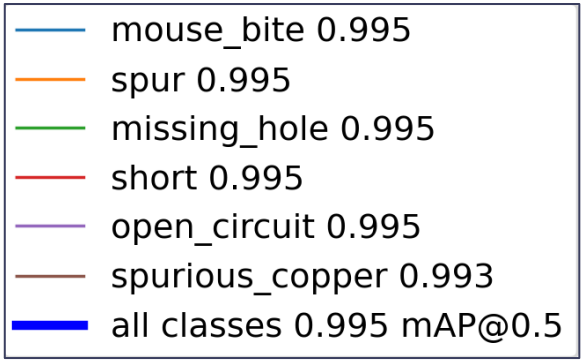
Class	Images	Instances	Box(P	R	mAP50	mAP50-95):
all	1068	1662	0.995	0.996	0.995	0.759
mouse_bite	131	262	0.992	1	0.995	0.767
spur	138	279	0.991	0.996	0.995	0.759
missing_hole	145	283	0.996	1	0.995	0.766
short	142	275	1	0.989	0.995	0.759
open_circuit	128	265	1	0.996	0.995	0.738
spurious_copper	145	298	0.99	0.996	0.993	0.764

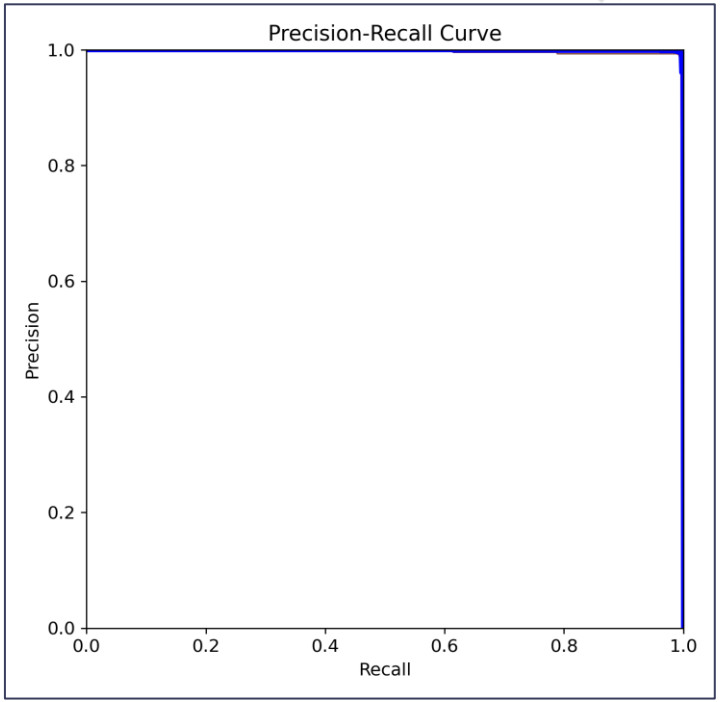
Speed: 0.6ms preprocess, 2.9ms inference, 0.0ms loss, 0.4ms postprocess per image

實驗結果 (6/16)

測試集的 Precision-Recall Curve

- **mAP50**：0.995

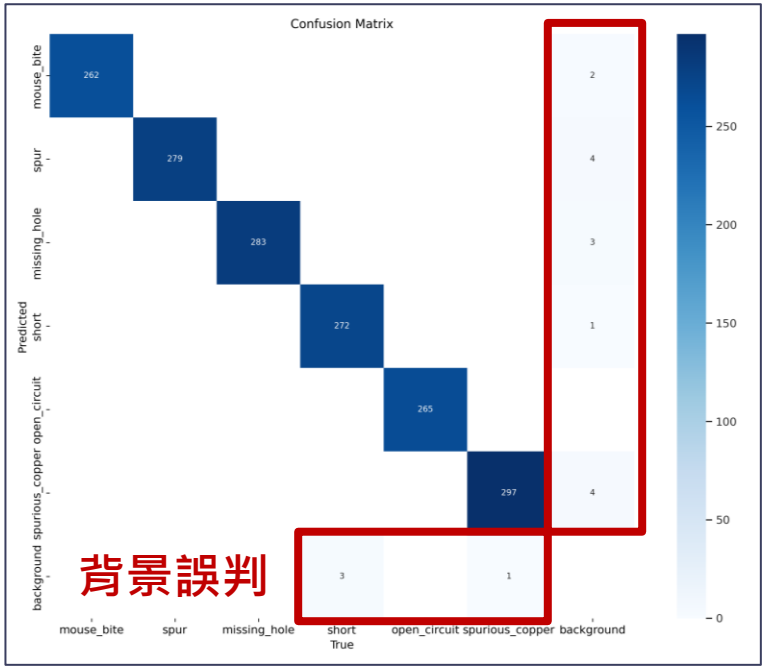




實驗結果 (7/16)

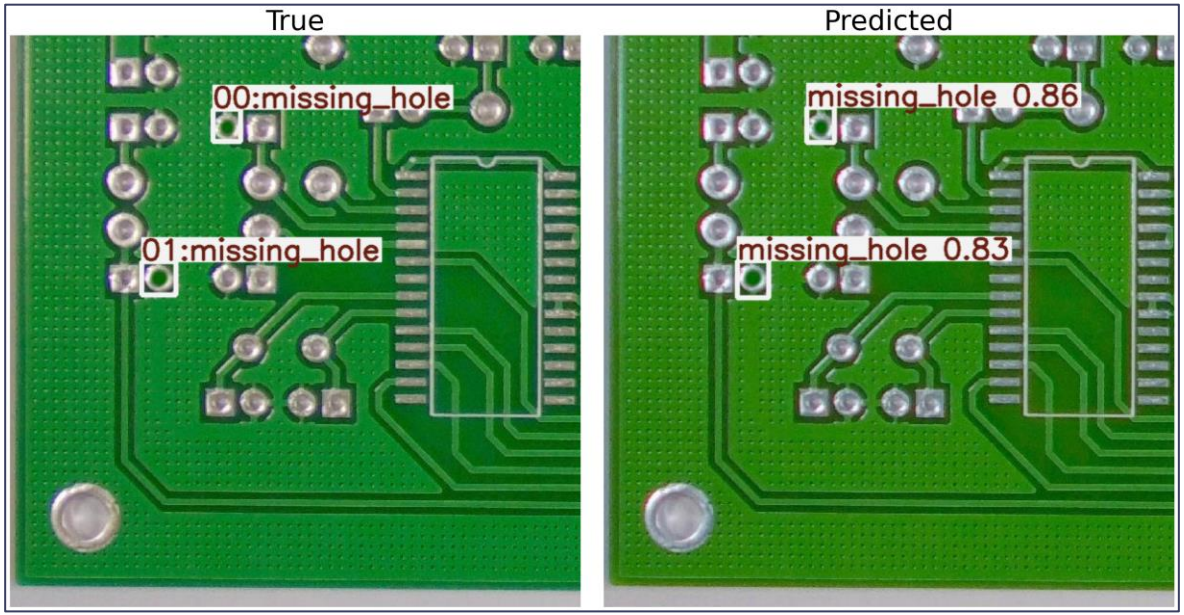
測試集的 Confusion Matrix

- 能正確辨識瑕疵點及類別
- 在背景上 (正常電路)，發生極少數的誤判



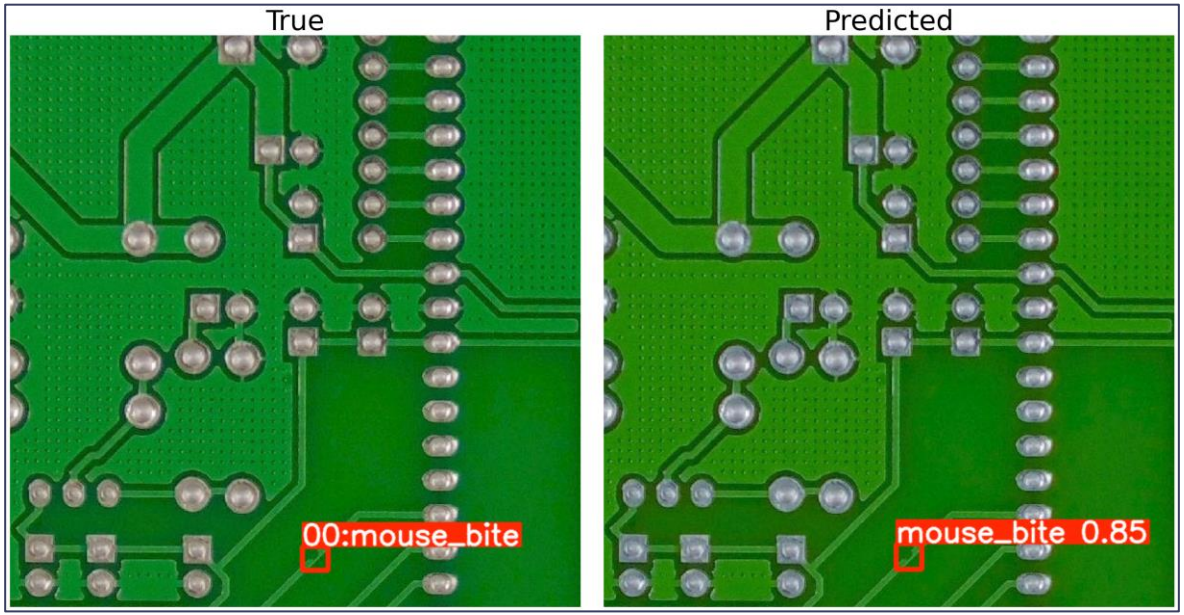
實驗結果 (8/16)

比較真實標籤及預測結果 (missing_hole)



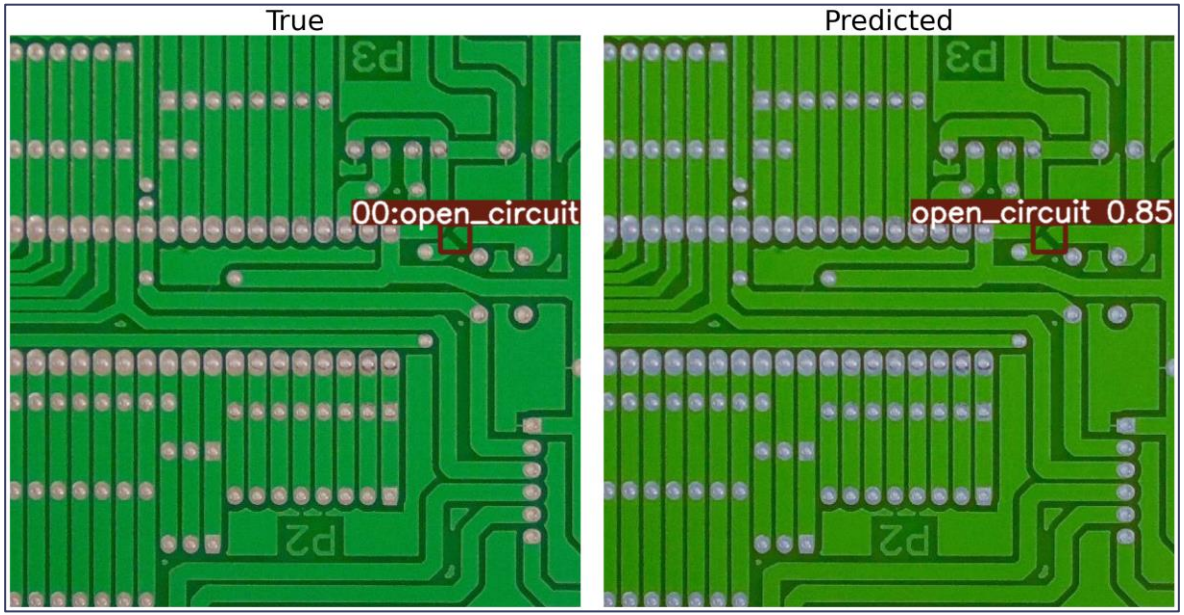
實驗結果 (9/16)

比較真實標籤及預測結果 (mouse_bite)



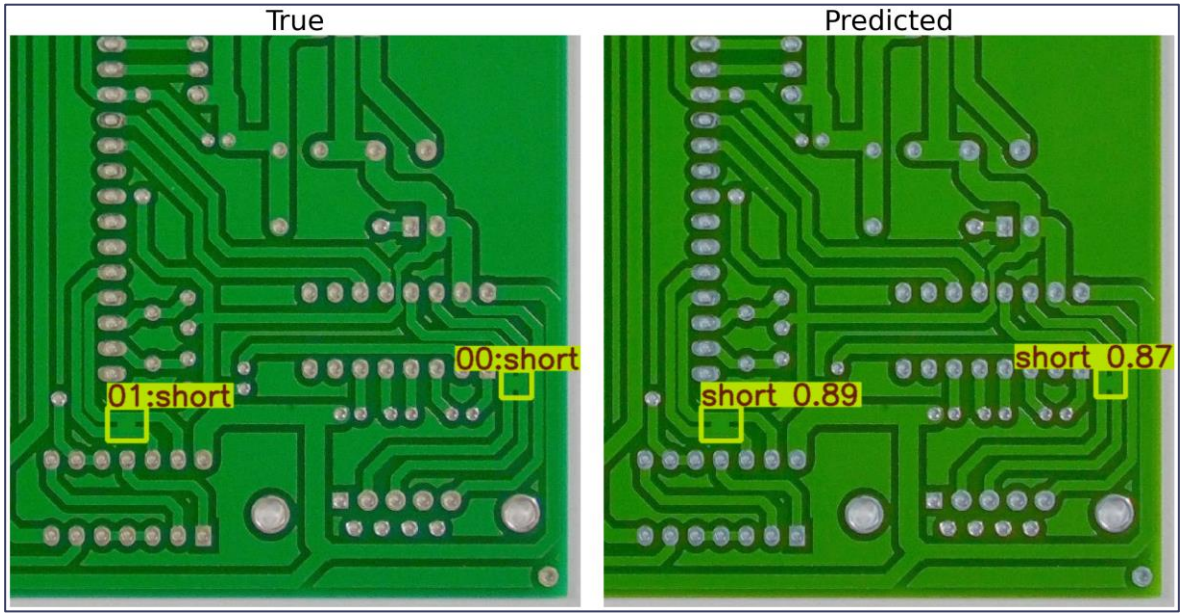
實驗結果 (10/16)

比較真實標籤及預測結果 (open_circuit)



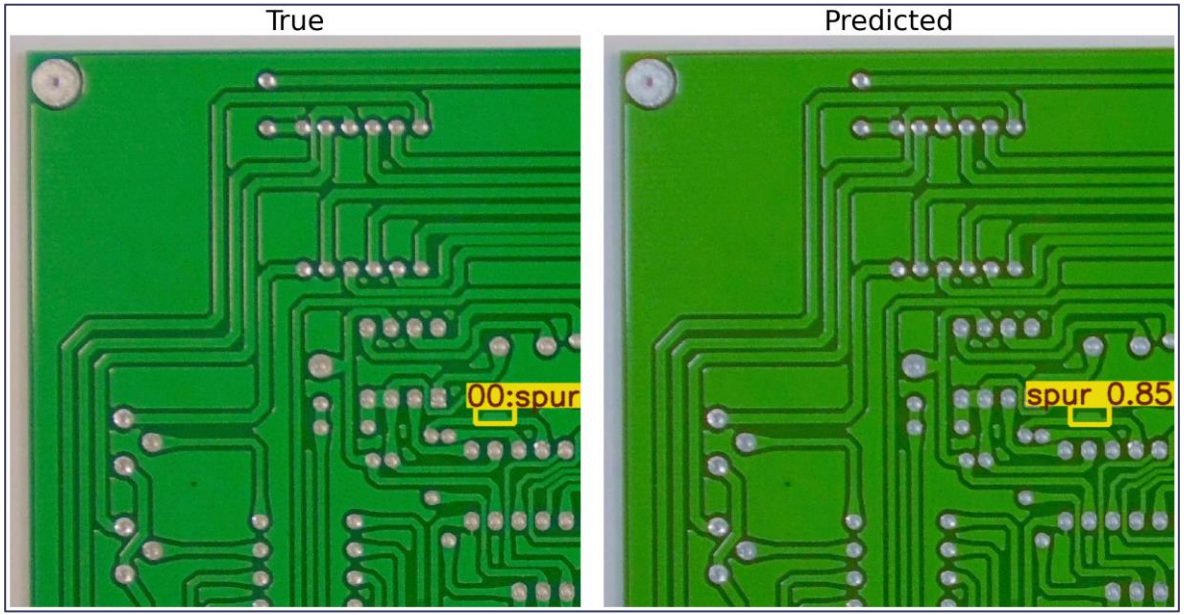
實驗結果 (11/16)

比較真實標籤及預測結果 (short)



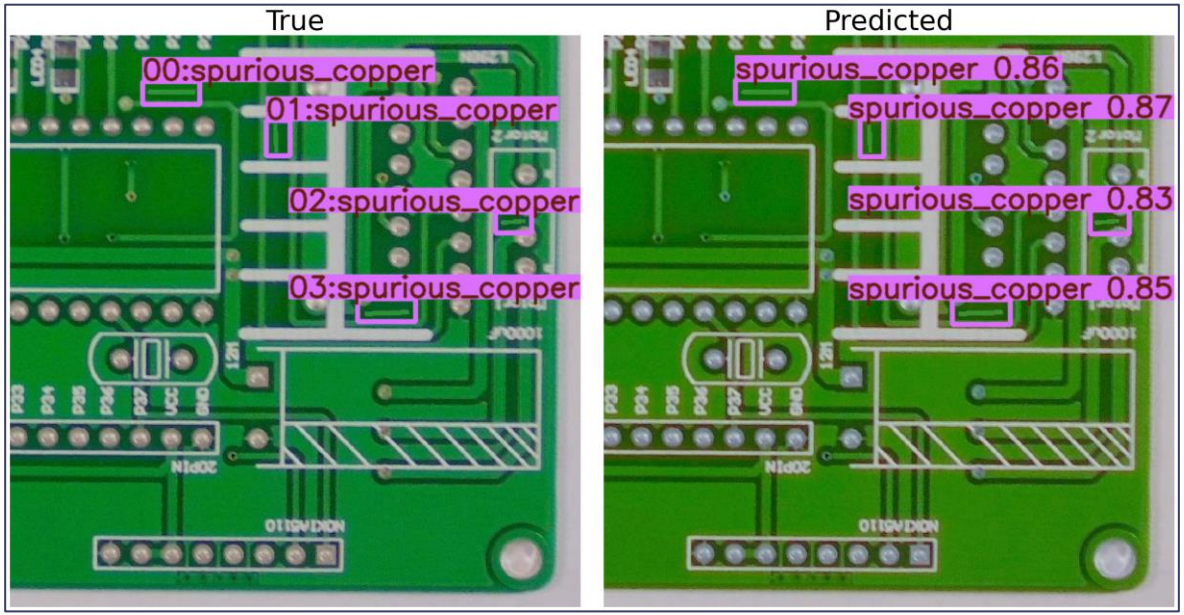
實驗結果 (12/16)

比較真實標籤及預測結果 (spur)



實驗結果 (13/16)

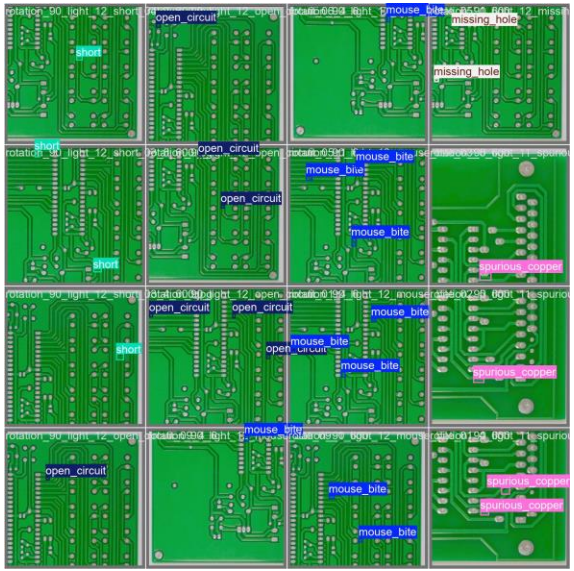
比較真實標籤及預測結果 (spurious_copper)



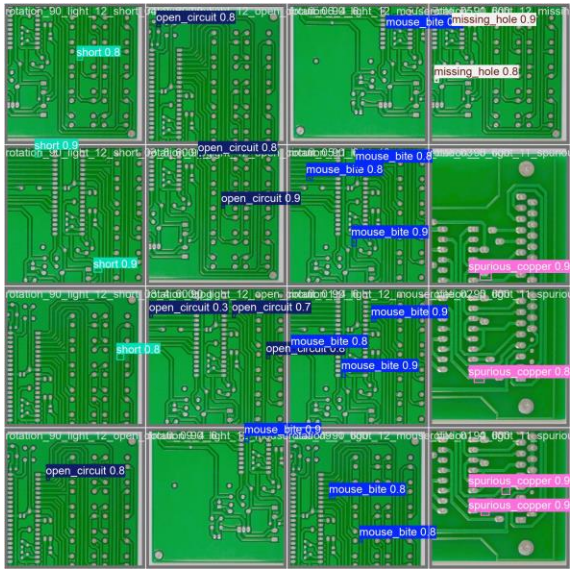
實驗結果 (14/16)

比較真實標籤及預測結果 (Batch 0)

真實標籤



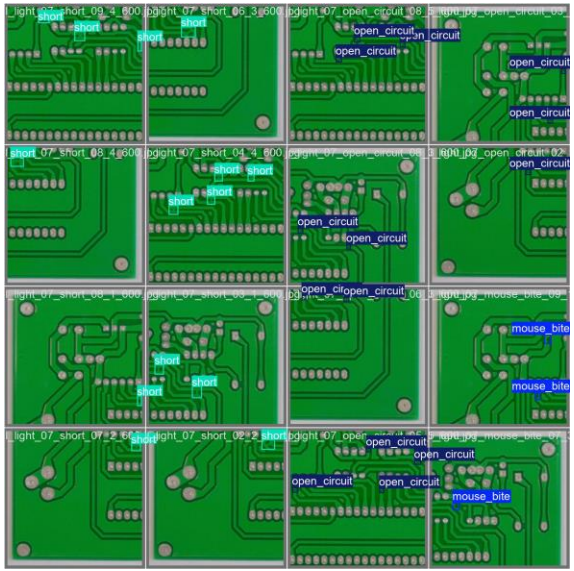
預測結果



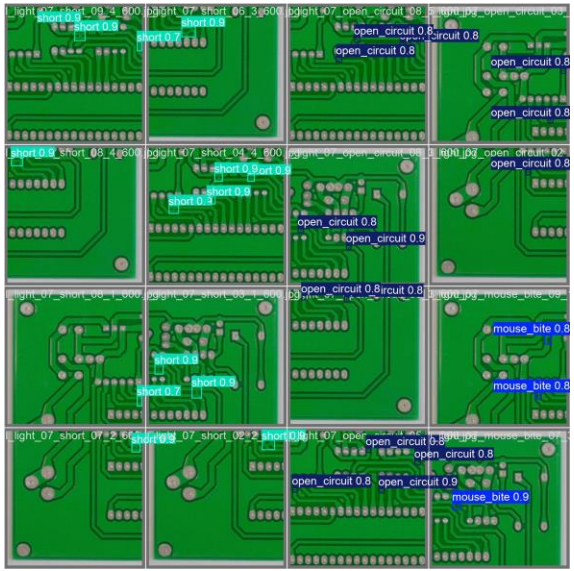
實驗結果 (15/16)

比較真實標籤及預測結果 (Batch 1)

真實標籤



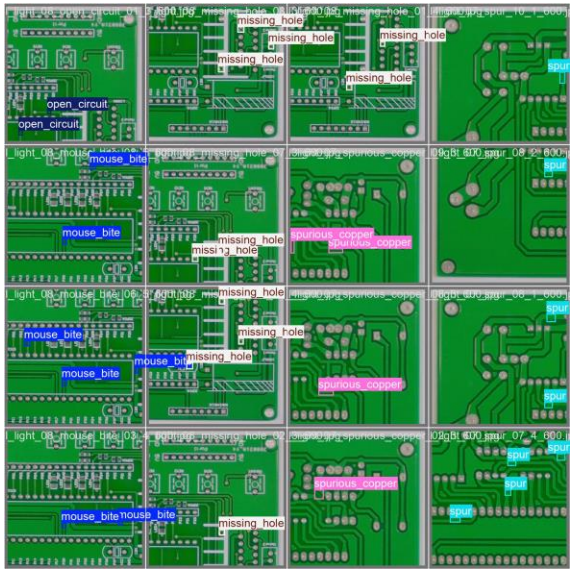
預測結果



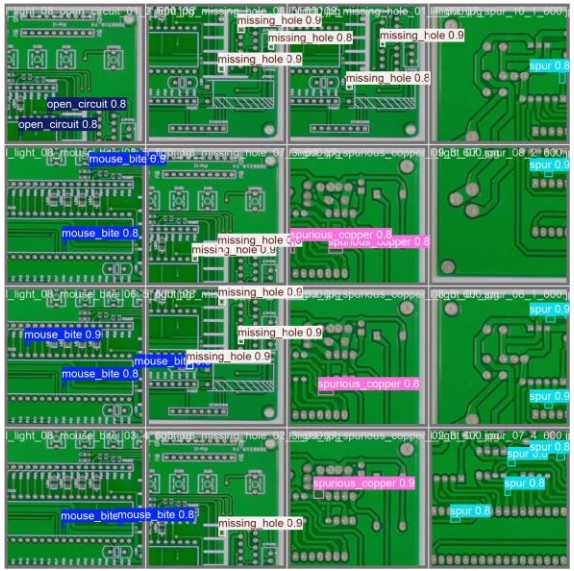
實驗結果 (16/16)

比較真實標籤及預測結果 (Batch 2)

真實標籤



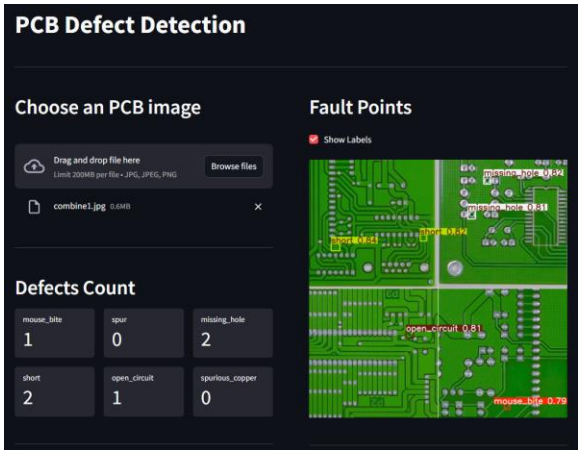
預測結果



成果展示 (1/3)

本專題開發了一款簡易的 PCB 瑕疵檢測程式：

- **功能**：透過網頁瀏覽器上傳 PCB 圖片，
  即時獲得瑕疵檢測結果
- **輸入**： PCB 圖片
- **輸出**：
  - 瑕疵點種類與數量統計
  - PCB 的瑕疵點標記圖片

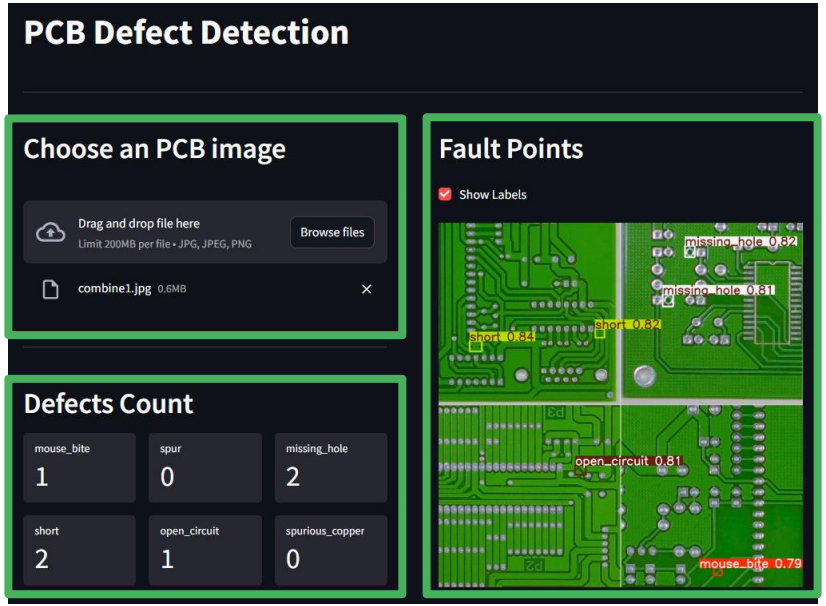


成果展示 (2/3)

- 介面與操作流程介紹：

1. 上傳 PCB 圖片

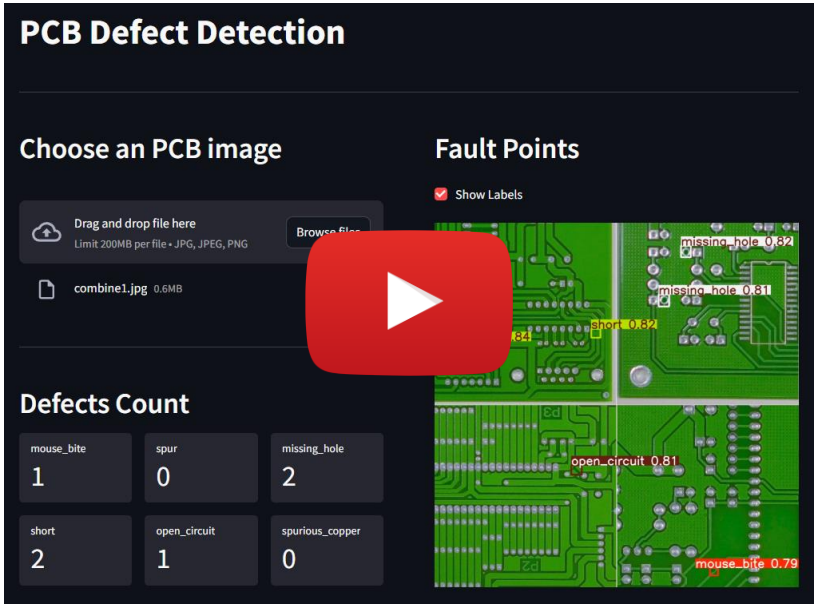
2. 瑕疵點種類與數量統計



3. PCB 的瑕疵點標記圖片

成果展示 (3/3)

- 成果展示影片：



結論

- 本專題成功將 YOLO v11 s 模型應用於 PCB 瑕疵檢測，並展現了卓越的準確性與極快的推論速度，mAP50 達到 0.995。
- 本專題使用深度學習方法進行 PCB 瑕疵檢測，具備高度的靈活性和泛用性，突破了傳統方法依賴於模板規則的限制，更能適應現代 PCB 多樣且複雜的趨勢。
- 本專題製作了一款簡易的 PCB 瑕疵檢測程式，讓使用者能夠透過網頁瀏覽器上傳 PCB 圖片，即時獲得瑕疵檢測的結果，結果包括瑕疵點種類與數量統計，以及 PCB 的瑕疵點標記圖片。

未來展望

- **新增更多元的 PCB 瑕疵種類**
  - 本專題的檢測模型能識別 6 種瑕疵，**現實中有更多瑕疵種類**
- **提升瑕疵點（小物件）定位的準確度**
  - 針對 box_loss 收斂緩慢的問題，未來可嘗試**收集高解析度的 PCB 瑕疵圖片**或**提高瑕疵標記的精度**來解決
- **更準確地分辨背景（正常電路）與瑕疵點**
  - 目前有極少數的背景（正常電路）被誤判為瑕疵點，未來可嘗試**更精細地調整模型超參數**或**修改模型架構**來解決

報告結束

謝謝