

# 整合深度學習與自動化技術於半導體瑕疵檢測之解決策略

洪宗貝<sup>1</sup>, 李詠騏<sup>2</sup>, 蘇子仁<sup>3</sup>

<sup>1</sup> 國立高雄大學資訊工程學系

E-mail: tphong@nuk.edu.tw

<sup>2</sup> 正修科技大學資訊管理系

E-mail: yeongchi@gcloud.edu.tw

<sup>3</sup> 國立高雄大學資訊工程學系

E-mail: l1115502@mail.nuk.edu.tw

## 摘要

本研究結合自動流程機器人(RPA)、物件偵測技術、資料擴增、貪婪樣本搜尋、及基於主動學習的半自動資料標記方法,提出一個半導體瑕疵檢測之有效策略,可以解決從項目開發至後期維護階段的挑戰。我們所提的方法可以大幅減少資料準備與標記時間,並增強開發流程的靈活性。其利用 RPA 技術,將部分開發工作交由專家客製化,顯著縮短了開發時程。此外,應用品質管控標準與增量學習策略,有效維持模型性能,適應產業需求變化。我們也實際建構出一個系統,實驗結果顯示整體開發時程縮短了 65.7%。

**關鍵字:** 深度學習、瑕疵檢測、自動流程機器人、半自動標記、模型效能管控。

## Abstract

This study integrates robotic process automation (RPA), object detection technologies, data augmentation, greedy sample search, and semi-automatic data annotation based on active learning to address these challenges from project development to post-deployment maintenance. The proposed framework significantly reduces the time required for data preparation and annotation while enhancing the flexibility of the development process. The development timeline is notably shortened by utilizing the RPA technology to delegate certain development tasks to domain experts for customization. Furthermore, applying quality control standards and incremental learning strategies can effectively maintain model performance, adapting to the evolving demands of the industry. We have also developed a practical system based on this approach, and experimental results demonstrate a 65.7% reduction in overall development time.

**Keywords:** Deep Learning, Defect Detection, Robotic Process Automation, Semi-Automatic Annotation, Model Performance Control.

## 1. 前言

近年來,半導體業逐漸依賴深度學習技術進行瑕疵檢測 [6]。然而,在導入此檢測機制的同時,也逐漸暴露出一些問題和挑戰。包括資料標記的困難、多物件的交互關係判斷、如何發現未知物件並快速具備偵測能力、開發時程的冗長、流程修改的困難、以及模型穩定性的驗證等。本研究旨在針對上述問題,設計一套涵蓋從開發、實際生產到後續維護各階段的整合解決系統,以應對包括訓練資料的準備、物件交互及未知物件的偵測、縮短開發時程、偵測模型完成後的效能檢驗與維護,以及具備靈活調整工作流程的系統整合方法。

本文首先將回顧相關文獻以明確理論基礎,包括深度學習應用於瑕疵檢測之相關研究、並在偵測到物件後透過交互條件進行瑕疵判斷,結合異常偵測技術使用於偵測未知物件、以降低瑕疵遺漏風險,透過資料擴增以增加樣本多樣性,利用基於主動學習的半自動化標記並結合代表性樣本搜尋、以縮短資料標記時程、並提升模型性能,透過檢驗方法監控模型穩定性、並在發現性能下降時使用增量學習恢復模型偵測能力,最後透過 RPA (Robotic Process Automation, 自動流程機器人) [13] 流程自動化工具連結不同工作流程。接下來,我們將展示實驗結果,討論遇到的挑戰和相應的解決策略,最後總結本研究的主要發現並指出未來的研究方向。

## 2. 相關研究

在半導體業的瑕疵偵測應用中,物件偵測技術 (Object Detection) 因為能夠對瑕疵物件進行精準的分類和定位而受到歡迎 [10]。而此技術又分為兩階段式與一階段式偵測 [3],其中,YOLO (You Only Look Once) 因為具備高效 [5]、資源需求低的特性而受到歡迎 [3]。到目前為止已有多個版本的 YOLO 系列模型 [16],從 YOLOv5 開始改為 Pytorch 架構 [4],為方便實驗及開發,使用同樣架構的 YOLOv5、YOLOX、YOLOv7、YOLOv8 及

YOLONAS，並在進入方法研究工作時，以實際產品資料及準確度結果作為最適用模型評估。

在生產環境中可能因為生產條件的變化而出現未知的外觀模式，而主要的偵測模型是基於已知的類別物件進行偵測，這可能帶來潛在的瑕疵遺漏風險，因此發現未知的外觀現象是一個重要議題；這裡探討了異常偵測 (Anomaly Detection)，在已知的方法中，PatchCore 是 IAD (Industry Anomaly Detection) 任務中最有效的偵測方法 [9]。它基於一個想法：先將輸入的圖像會被劃分成多個瓦片、並輸入到神經網絡中，當單個區塊 (Patch) 是異常的，整個圖像就被分類為異常。

為了提升偵測模型的效能，也探討了資料擴增，這是提高深度學習模型表現的一種有效方法，尤其在圖像識別領域，其效果已獲得廣泛認可。而基於實際產品可能的外觀變化，使用的方法如：旋轉、縮放、翻轉、鏡像、色彩增強、PatchShuffle Regularization、隨機擦除 [15]。

為了縮短標記資料的工作時程，針對基於主動學習的半自動化標記方法也進行了探討，這是一種選擇高價值的訓練資料方法，能夠在較少的訓練資料下得到相對於隨機資料訓練的模型、以及半監督式學習訓練的模型等方法更好的識別結果 [1]。

如果使用於訓練的資料非常相似，將導致模型泛化能力不足。依靠人工搜尋圖片，可能因為資料量龐大、而需要花費大量的時間。因此，不依賴人工找到具代表性的資料方法，是模型能在短時間內從少量資料中獲得最佳效能的關鍵。而基於 PatchCore 方法所萃取的核心樣本在 Anomaly Detection 應用的有效性 [9]，將參考此方法搜尋 Coreset 的想法作為搜尋代表性樣本的方法。其流程包括：將樣本圖片進行特徵圖萃取、並進行 Patch 特徵萃取，接著計算 Patch 特徵之間的分數後，以貪婪近似演算法搜尋一部分的代表性樣本作為核心集 [14] 來提升搜尋效率。

在既有類別出現變化導致模型效能下降以後，需要對已完成訓練工作的模型再次進行訓練，以使其能恢復應有的偵測效能或偵測出新的瑕疵類型。在 [11] 中使用了增量學習，它使模型能在新的資料上繼續學習，同時保留以前的知識。其中，iCaRL 方法可透過類別增量學習的方式逐步增加新的類別 [12]，當需要加入新類別時，先合併舊類別代表樣本，並更新特徵提取參數，接著根據分配的記憶體數量來調整新舊類別資料集。此方法選擇最代表類別的資料，除了能有效地管理和利用有限的記憶體資源，也能防止因增加新類別而導致的災難性遺忘。

由於實際上的問題不僅是單純檢測出物件，而是物件與物件間的交互比較後結果，在已知的物件交互關聯性方法中，主要分成兩個議題：人-物件互動偵測、物件偵測後處理。忽略前者主要關注於人與物件這一點，兩者主要的差異是：前者在偵測出物件後、會再經過深度學習模型判斷兩者間的互動關係 [7]。而後者僅是輕量級的清理

工作，不涉及對物件互動的分析或識別。

### 3. 研究方法

為了解決包括物件交互條件的瑕疵檢驗、代表性資料的選擇與加速標記、縮短開發時程、模型效能檢驗及維護、以及工作流程的整合等問題，本研究將整個工作流程劃分為三個主要階段：專案開發、生產應用和後期維護。

在專案開發階段，首先選定特定時期範圍內的瑕疵樣本，並由專家進行類別設定和挑選少量訓練樣本。隨後，進行資料擴增操作以增加訓練資料的多樣性。接著，基於主動學習半自動標記的想法，使用 YOLOv7-X 作為標記資料使用的模型，以偵測後的信心分數搭配貪婪搜尋找到代表性樣本並逐步增加訓練資料，並透過 YOLOv5、YOLO-X、YOLOv7、YOLOv8、YOLO-NAS 等物件偵測模型進行評估，以選擇最適合的模型。

在生產階段，偵測物件後會透過物件交互處理和 PatchCore 進行判斷，並將結果傳遞外部系統。

在維護階段，根據瑕疵類別和嚴重程度設定 AQL (Acceptable Quality Levels, 抽樣品質允收標準) 抽樣標準 [8]，並結合貪婪搜尋選取樣本進行品質檢驗。每次檢驗後記錄的準確度分數將透過 SPC (Statistical Process Control, 統計製程管制) [2] 進行穩定性評估。當檢驗結果不符合預期時，利用 iCaRL 進行增量學習，以恢復模型性能。

最後，為了縮短開發及流程討論時程、我們將各階段與外部系統的連結工作，透過非程式化的 RPA 工具、由領域專家進行開發及流程整合。依據模型功能及流程功能，將分為模型準備與流程方法兩個階段說明。

#### 3.1 模型準備

模型準備階段包括五個步驟：資料準備、圖片標記、模型評估、異常偵測模型訓練、以及增量學習方法準備。在資料準備與圖片標記的部分，首先透過人工方式對少量瑕疵圖片進行分類，對於少量的樣本使用資料擴增技術進行處理。經人工標記後，參考基於主動學習的半自動標記方法，訓練 YOLOv7-X 用於推論未標記圖片。在推論過程中，利用貪婪搜尋樣本從未標記圖片中取得訓練圖片；需要使用此方法的重要原因是：因為候選的資料量非常龐大、使得人工難以進行完整的檢視，而同類別瑕疵圖片也存在相似樣本，因此，利用此方法選出代表性樣本、以達到提升模型泛化偵測能力的目的；這個方法包含兩個步驟：特徵萃取、樣本選擇，如圖 1 Extraction 與 Selection 所示。

在特徵萃取階段，利用模型推論後獲得的物件位置切割出物件，然後使用 Wide ResNet50-2 作為特徵萃取模型，並以中間層進行 Patch 特徵的提取，再經過合併後進行降維與扁平化處理，最後利用物件偵測的信心度進行過濾、並儲存其向量資訊，如圖 1 Feature Processing 所示。

在樣本選擇階段，先計算同類別、不同向量的相似度分數，再將結果建立成一個相似度矩陣。另外建立一個代表各向量的選擇旗標、且預設為 false，當該向量對應的選擇旗標被設置為 true 時，所有向量的平均相似度計算結果將不包含此向量，如圖 1 Similarity Score Matrix by Category 所示。

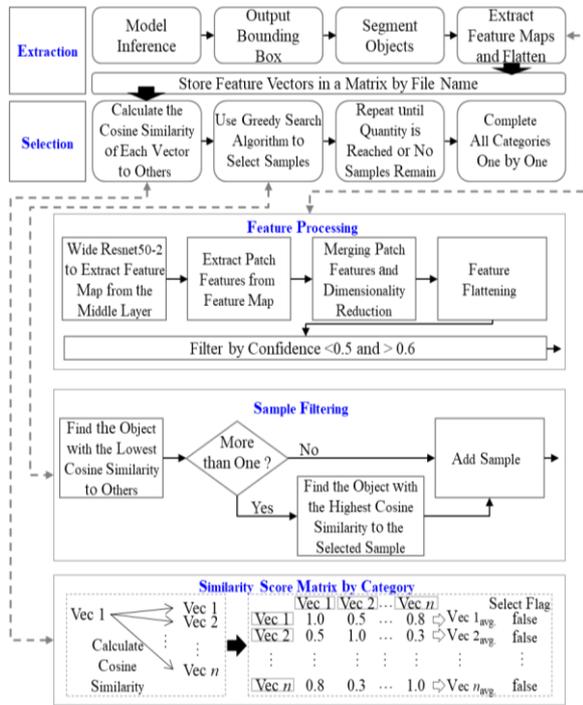


圖 1. 貪婪搜尋樣本整體流程

接著進行貪婪近似樣本搜尋，首先選擇平均相似度最低的向量，因為初始狀態不存在任何選擇旗標為 true 的向量，就直接加入此向量對應的檔案、再將此向量的選擇旗標設定為 true、並更新平均相似度。在開始搜尋第二個向量時，如果存在超過一項最低平均相似度的候選向量，就把選擇旗標設定為 true 的向量納入候選向量的平均相似度計算、並選擇其中相似度最高的向量之檔案，如圖 1 Sample Filtering 所示。

在獲得訓練資料後，使用多個版本的物件偵測模型、並依據 F1 分數選出最佳的模型。接著，為了避免因為出現未知物件而導致潛在的缺陷遺漏，這裡加入未知物件的偵測。首先搜尋出不同類別的代表性樣本，再逐一將不同的類別視為未知物件、並結合 K-Fold 驗證法進行評估，從而確定最佳的異常偵測模型。對於模型性能出現下降或需要加入未知類別偵測時，採用 iCaRL 的增量學習策略進行應對。

### 3.2 資料準備

初步的訓練資料準備流程包含幾個步驟：自動蒐集圖片、依據已知的瑕疵定義進行圖片分類、透過領域專家篩選外觀差異較大的同類別瑕疵。首先蒐集共 3,196,801 張產品圖片，其中的 10% 為瑕疵

圖片，圖片尺寸範圍介於 500×500×3 至 1000×1000×3 像素，接著透過領域專家進行人工過濾。由於瑕疵圖片的資料量要以人工執行分類過於龐大，因此先根據已知的瑕疵類型設定 13 種類別，再進行隨機的篩選，這個階段需要確保所有類別都能找到。對於樣本量較少的類別，使用實際可能存在的外觀現象進行資料擴增，最終所有類別的都超過 30 張，將以此作為初步的模型訓練資料。

### 3.3 多階段資料標記法

圖片標記階段是最耗時且勞力密集的一環。不同標記人員或同一標記人員在不同時間的標記結果可能會有所不同，這種差異性可能導致標記品質不一，進而影響模型的準確性。為了減少人力投入和加速整個流程，這裡採用了基於主動學習的半自動標記方法。首先以人工取得的初始訓練集對 YOLOv7-X 進行遷移學習，再以訓練後的模型對未標記的資料進行推論，接著對推論後的結果以貪婪搜尋樣本、搭配信心度小於 0.5 或大於 0.6 的閾值規則篩選出圖片，並利用 YOLO 推論得到的物件框產生標記檔；使用小於 0.5 作為篩選閾值的目的是要使模型學習未學好的樣本，使用大於 0.6 作為篩選閾值的目的是要維持已學習好的知識、同時快速增加訓練樣本。經過人工校正後、加入前一次的訓練資料繼續訓練模型，反覆此流程。

經過幾次的迭代流程後，各類別的資料集樣本累計超過 800 張。至此也與手動標記相同數量的資料集、所耗費時間進行比較。條件包括：相同資料改以三人進行標記、在標記前先以不連續標記同類別的方式熟悉標記動作、僅計算人工作業時間但不包含模型訓練時間、以訓練過的物件偵測模型對人工標記結果進行測試，以確保結果可信。

最後將半自動標記與完全人工標記所耗費時間進行比較，發現前者相對於完全手動標記方法，能顯著減少工作時間，具體來說減少了 93.7% 的標記時間，如表 1 所示。這一結果強調了半自動標記方法在效率提升方面的顯著優勢。

表 1. 半自動與全自動標記在相同數量耗費時間

Round	Total Datasets	Semi-automatic Annotation	Manual Annotation
1	> 390	2.5 Hrs	
2	> 1300	1.8 Hrs	35.5 Hrs
3	> 3900	5.3 Hrs	101.4 Hrs
Final	> 10400	15.1 Hrs	253.5 Hrs
Total		24.7 Hrs	392.9 Hrs

以此階段透過半自動搜尋獲得的資料所訓練後的物件偵測模型為基礎，對所有的瑕疵圖片進行推論，其中 83% 集中於 2 種類別，17% 包含 11 種類別，類別之間呈現不平衡資料集比例關係，這部分也說明若要透過人工在大量的資料中進行搜尋，尤其是少量類別要能被搜尋到、而且還需要是同類別彼此差異較大的樣本，將會是非常困難的。

### 3.4 模型評估

評估的模型選項包含：YOLOv5(x6)、YOLOX(x)、YOLOv7(-X、W6、E6、D6、E6E)、YOLOv8(N、S、M、L、X)、YOLONAS(S、M、L)。為了避免隨機性影響且加速評估的流程，停用所有訓練期間的資料擴增超參數，並且使用完全相同的資料集及預訓練權重進行遷移學習，在訓練 400 個回合後，以最佳權重進行測試，並依據最佳的 F1 分數作為模型選擇的指標。在找到最佳模型後，加入訓練期間的資料擴增超參數進型完整訓練。

### 3.5 異常偵測

除了使用物件偵測模型作為主要的檢測方法以外，也加入了 Anomaly Detection，目的是要發現未知的外觀物件。由於 PatchCore 是從正常資料中萃取出指定比例的 Coreset 以後，在測試階段將未知圖片用來和 Coreset 進行比較、藉此判斷是否存在缺陷。基於此想法，這裡將已知的瑕疵視為正常樣本，當出現與已知瑕疵不同的物件時，該物件就是未知物件。這裡使用訓練物件偵測模型的 13 種類別資料進行拆分，其中的 12 種類別視為正常樣本，另外一種類別作為未知類別物件的測試樣本，接著使用 K-Fold 驗證方法逐一對不同類別物件進行測試。在嘗試不同的 Coreset 百分比、Coreset 萃取模型組合後，透過最佳的 F1 分數找到適用於此資料集的模型，接著，將其加入生產檢測流程中。

### 3.6 增量學習

由於在訓練資料不足、短時間內需要能偵測、且需要維持舊有類別檢測能力的前提下，參考 iCaRL 方法對已上線工作的物件偵測模型進行增量學習、使其能快速具備對新增的外觀模式具備偵出能力。首先，使用合適的資料擴增方法對新資料進行擴增並進行標記工作，對於舊類別，先取得代表性的樣本，並使新舊類別維持數量平衡，接著依序載入 Backbone、最後訓練的權重，並加入訓練期間資料擴增進行訓練，待訓練完成，對包含新舊資料的測試集進行測試，如果結果符合期望水準，則更新代表性樣本資料，作為下次增量學習使用。

## 4. 流程方法

此節將說明模型以外的工作，包含：物件偵測模型推論後的物件交互演算法處理、加入人工抽樣檢驗的 AQL、定期檢驗物件偵測模型效能的 SPC、以及透過 RPA 在開發/生產/維護階段等時期連結不同系統流程的整合機制。

### 4.1 推論後物件交互處理

在某些情況下，瑕疵與否是藉由不同物件的交互關係來決定的。例如底下六種情況：指定類別的至少其中一項物件必須存在（如圖 2 Situation 1 所示），指定的所有類別必須同時存在（如圖 2

Situation 2 所示），指定類別數量需要符合指定數量（如圖 2 Situation 3 所示），物件不能完全重疊、且面積需超過某一類別一定比例（如圖 2 Situation 4 所示），其中一物件需完全被包覆在另一物件、且面積之比超過一定比例（如圖 2 Situation 5 所示），多個相同類別中、大於或小於同類別面積中位數一定比例的單一物件（如圖 2 Situation 6 所示）。針對此六種情況加入模型推論後的交互判斷。

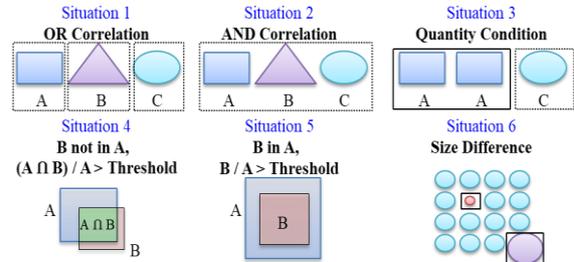


圖 2. 六種物件交互關係情境

### 4.2 抽樣檢驗

由於實際資料可能發生與訓練資料不同的變化，因此在偵測後加入抽樣檢驗，目的是為了進一步提高對偵測結果的信心水準。首先依據不同批次產品的偵測母數、抽樣計劃，透過 AQL 表獲得抽樣數量與允收標準。檢驗樣本包含兩個部分：模型偵測含有瑕疵的樣本中、利用貪婪搜尋取出指定的比例，與未被偵測到瑕疵的樣本中、貪婪搜尋取出剩餘的比例；對於未被偵測到瑕疵的貪婪搜尋方法，此處在萃取特徵的部分修改為從完整圖片進行萃取，其餘步驟皆相同。接著在完成檢驗樣本儲存後，模型系統透過 API 方法呼叫 RPA，RPA 從樣本池取得檢驗樣本後、以內部信件的方式通知檢驗人員。

### 4.3 穩定性監控與增量學習

除了透過 AQL 進行人工抽樣檢驗以外，也利用抽樣後並校正標記檔的結果進行模型測試、再以 SPC 檢驗來管理模型穩定性，並根據 AQL 和 SPC 結果判斷是否進行增量學習。模型一次的測試結果代表一點，那麼 SPC 異常判定規則包括：任何點超出三倍標準差管制界限外，連續七點以上出現在中心線同一側，連續六點呈現上升或下降趨勢，連續十四點呈上下交互跳動，連續三點中、有兩點出現在兩倍標準差外，連續五點中、有四點出現在一倍標準差外，連續十五點集中在一倍標準差內。首先進行人工結果驗證，驗證後將正確結果儲存到正確樣本池，錯誤結果校正標記檔案後、儲存到錯誤樣本池，並將兩種結果合併後進行模型測試。如果 AQL 及 SPC 驗證 F1 分數皆通過，就不進行增量學習；如果任何一項未通過，就執行增量學習。

### 4.4 流程開發

整個專案中，包含了開發、生產及維護階段，在不同階段中有一些重複的流程，這裡由最熟悉流程的領域專家透過自動流程機器人 (RPA) 將各項跨

系統的任務進行串聯，以建立完整的流程機制。

## 5. 研究結果

本節將介紹本研究使用的開發流程應用於實際專案開發、並與工廠內的專案進行比較。項目包含：對專案時程的影響，相同模型前提下、不同資料準備方式對準確度指標的影響，與專案上線後、經由人工驗證模型偵測的結果。

### 5.1 開發時程與所需人力影響

將工作任務分成資料準備、資料標記、工作討論、IT 開發工作、領域專家開發工作、模型訓練、上線前測試。其中本研究方法應用於資料準備工作（搭配 RPA、貪婪搜尋且不需人工篩選，傳統方法依賴領域專家人工搜尋），資料標記工作（搭配貪婪搜尋、半自動標記與人工校正，傳統方法依賴領域專家標記），討論工作的減少（需要人工智慧專家、領域專家、針對人工智慧技術討論，傳統方法需要多個領域專家共同參與，且著重在問題、想法、流程、可行性的討論），減少資訊及自動化專家在流程的開發工作（僅需開發 API 及少量流程功能、並由人工智慧專家完成，傳統方法由資訊、自動化領域專家開發），增加領域專家開發工作（使用 RPA 開發，傳統流程僅提供想法）。

以三件專案為題目、與本研究方法同步進行，以比較在不同流程中的影響。以工作日為單位，資料準備工作減少 53.3%，標記時間減少 73%，討論時間減少 89.5%，IT 開發時間減少 92%，整體開發時程減少 63.4%；此外，由於流程開發工作由領域專家進行，因此減少了程式化流程開發工作、而進一步減少多個部門參與，整體而言減少了 53.8% 開發人力，如圖 3 所示。

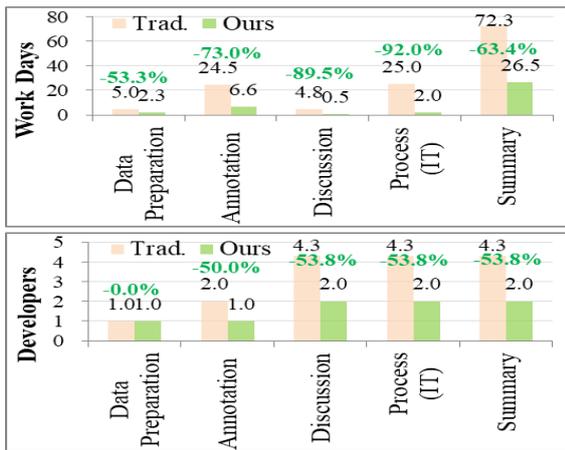


圖 3. 傳統與本研究方法開發時程與人力差異

計算至八項專案，整體開發時程平均為 24.8 個工作日，相較於既有流程平均耗時 72.3 個工作日，減少了 65.7% 的開發時程，如圖 4 所示。

### 5.2 隨機與貪婪搜尋方法對準確度的影響

除了一開始都使用相同少量人工搜尋獲得的訓練資料以外，接著分別比較隨機搜尋與貪婪搜尋獲得的資料、對模型的準確度差異。在相對較少的訓練資料、相同模型進行準確率的比較，指標包含：精確度、召回率、F1 分數。三項專案結果中，本研究方法精確度提升 3.1%，召回率提升 3.5%、F1 分數提升 3.5%，如表 2 所示。進一步檢視訓練樣本差異，發現隨機搜尋的樣本中，存在較多的相似樣本，而造成相對較差的模型泛化能力。

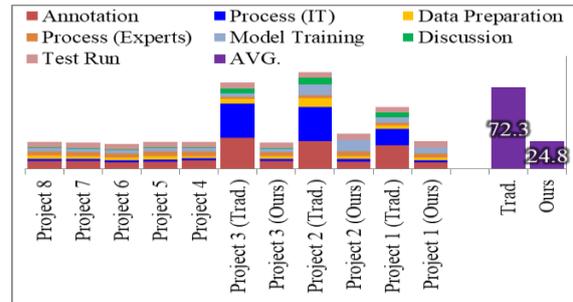


圖 4. 傳統與本研究方法整體開發時程差異

表 2. 隨機及貪婪搜尋樣本對準確率的影響

No.	Model	Methods	mP@.5	mR@.5	mF1@.5
1	YOLOv5-X	Trad.	88.7%	86.5%	87.6%
		Ours	<b>91.0%</b>	<b>89.8%</b>	<b>90.4%</b>
		Diff.	<b>+2.3%</b>	<b>+3.3%</b>	<b>+2.8%</b>
2	YOLOv7-E6E	Trad.	90.5%	88.5%	89.5%
		Ours	<b>94.2%</b>	<b>93.2%</b>	<b>93.7%</b>
		Diff.	<b>+3.7%</b>	<b>+4.7%</b>	<b>+4.2%</b>
3	YOLOv7-X	Trad.	90.2%	89.2%	89.7%
		Ours	<b>93.5%</b>	<b>91.7%</b>	<b>93.1%</b>
		Diff.	<b>+3.3%</b>	<b>+2.5%</b>	<b>+3.4%</b>
Ours V.S. Trad. AVG.			<b>+3.1%</b>	<b>+3.5%</b>	<b>+3.5%</b>

### 5.3 實際上線後的準確率驗證

整合 YOLO、PatchCore、Post Processing 等方法的偵測結果，以每張圖片為單位進行人工驗證。判斷瑕疵標準包括：物件偵測模型偵測出物件，並透過物件交互演算法判斷為瑕疵，PatchCore 偵出該圖片出現未知物件，當同一圖片偵測結果包含非瑕疵與真實瑕疵時，判定瑕疵；若否，則視為偽瑕疵。基於此規則計算精確度、召回率、F1 分數等指標，共進行 30 次不同模型結果、總計 60,000 張圖片的檢驗樣本，其中 57% 沒有瑕疵，43% 存在瑕疵。檢驗結果的各項指標為：精確度 95.4%，召回率 99.5%，F1 分數 97.4%，如表 3 所示。

表 3. 三十次的模型判斷並經過人工驗證結果

Result By Manual Validation				P (%)	R (%)	F1 (%)
TP (Fail)	TN (Pass)	FP (Pass)	FN (Fail)			
25,583	33,035	1,238	144	95.4	99.5	97.4

## 6. 結論與未來展望

本研究透過整合多種技術與策略來應對製造業中瑕疵檢測的多重挑戰。藉由 RPA、YOLO 物件偵測、資料擴增、貪婪搜尋樣本、以及基於主動學習的半自動標記機制，減少資料準備及標記時間。此外，透過將 RPA 融入開發流程中，有效地將複雜的流程開發任務分配給領域專家，進而減少了開發人力、會議時間並縮短整體開發時程。這種策略加速專案進度、提高流程變更彈性，從而能迅速適應生產變化。在檢測策略方面，本系統結合 YOLO 和物件交互處理以應對複雜的條件場景，並整合異常檢測、AQL 及 SPC 等方法，增強對偵測結果的信心度。此外，當模型偵測效能下降時，運用增量學習進行模型強化，確保持續的檢測能力。綜上所述，本研究提供一個完整的解決方案，構建了一個高效、靈活且可持續的工廠自動化瑕疵檢測系統。

未來，我們計劃進一步優化自動標記、瑕疵檢測及自動化流程。首先，改進自動標記系統，因資料相關工作在開發和維護階段仍占用大量人力，可探索更高效的標記方法，提升品質與效率、並減少人員投入。其次，對於異常偵測僅假設出現一種，但實務上仍可能同時出現一種以上的未知物件，未來可以探討如何識別更多未知的瑕疵，以應對複雜多樣的實務情境。最後，對於自動化流程仍需要相對熟悉現況的領域專家進行開發，未來可將這些流程建立知識，以自動化探索更智能化和自適應的流程控制策略，來提高生產流程的智能化效率和品質。

## 7. 參考文獻

- [1] Ao, B., & Fan, B. (2022). Overview of the development of AI dataset annotation. *The International Symposium on Artificial Intelligence and its Application on Media*, pp. 174-181.
- [2] Bottani, E., Montanari, R., Volpi, A., & Tebaldi, L., (2023). Statistical process control of assembly lines in manufacturing. *Journal of Industrial Information Integration*, Vol. 32, p. 100435.
- [3] Carranza-García, M., Torres-Mateo, J., Lara-Benítez, P., & García-Gutiérrez, J. (2020). On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data. *Remote Sensing*, Vol. 13, No. 1.
- [4] Hussain, M. (2023). YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection. *Machines*, Vol. 11, No. 7.
- [5] Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A review of YOLO algorithm developments. *Procedia Computer Science*, Vol. 199, pp. 1066-1073.
- [6] Keleko, A. T., Kamsu-Foguem, B., Ngouna, R. H., & Tongne, A. (2022). Artificial intelligence and real-time predictive maintenance in industry 4.0: a bibliometric analysis. *AI and Ethics*, Vol. 2, No. 4, pp. 553-577.
- [7] Kim, B., Lee, J., Kang, J., Kim, E. S., & Kim, H. J. (2021). Hotr: end-to-end human-object interaction detection with transformers. *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 74-83.
- [8] Kurniadi, D., & Syahputra, A. (2023). Penerapan AQL (Acceptable Quality Level) Pada Proses Trimming Baju Tidur Untuk Perbaikan Mutu Di CV. Aliya. *Jurnal Teknik dan Teknologi Tepat Guna*, Vol. 2, No. 1, pp. 10-26.
- [9] Liu, J., Xie, G., Wang, J., Li, S., Wang, C., Zheng, F., & Jin, Y. (2024). Deep industrial image anomaly detection: A survey. *Machine Intelligence Research*, Vol. 21, No. 1, pp. 104-135.
- [10] Liu, X. P., LI, Y. Q., Liu, L., & Wang, Z. (2019). Improved YOLOv3 target recognition algorithm for adaptive edge optimization. *Microelectronics & Computer*, Vol. 36, No. 7, pp. 59-64.
- [11] Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A. D., & Van De Weijer, J. (2022). Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 45, No. 5, pp. 5513-5533.
- [12] Rebuffi, S. A., Kolesnikov, A., Sperl, G., & Lampert, C. H. (2017). iCaRL: incremental classifier and representation learning. *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2001-2010.
- [13] Ribeiro, J., Lima, R., Eckhardt, T., & Paiva, S. (2021). Robotic process automation and artificial intelligence in industry 4.0—a literature review. *Procedia Computer Science*, Vol. 181, pp. 51-58.
- [14] Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., & Gehler, P. (2022). Towards total recall in industrial anomaly detection. *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14318-14328.
- [15] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 1-48.
- [16] Terven, J., Córdova-Esparza, D. M., & Romero-González, J. A. (2023). A comprehensive review of YOLO architectures in computer vision: from YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, Vol. 5, No. 4, pp. 1680-1716.