

國立高雄科技大學

半導體工程系

專題製作報告

影像辨識-太陽能電池晶片缺陷之研究

指導教授：姚永正

學 生：汪上安、翁偉豪、羅子達、呂重翰、蔡世鈞、

楊云亨、張嘉顯、陳遠碩

中 華 民 國 年 月

國立高雄科技大學

半導體工程系

專題製作報告

影像辨識-太陽能電池晶片缺陷之研究

指導教授：_____（簽章）

中 華 民 國 年 月

摘要

目前 AOI 測量方法，如灰度分析、邊緣檢測等，都存在侷限。灰度分析會因反光和陰影影像、顏色相近、須高反差依賴以及缺乏高度信息。邊緣檢測在邊緣模糊像是說拍攝角度問題造成光照等因數變得模糊，影響檢測準確度等，最大問題是對於細小特徵無法檢測，因是使用邊緣檢測關係所以難以精確檢測。

本研究系統利用 YOLO 以及 CNN 2 個模型去做辨識首先是使用 YOLO 來判斷我們須測量的晶片是否已經在鏡頭裡並把他框出來，接下來使用第二個模型以分析的晶片好壞。

YOLO 演算法以及 OpenCV 去除樣品中不需要的部分。

關鍵字： YOLO、CNN、AOI、人工智慧、視覺辨識、物件追蹤、晶片檢測

Abstract

Currently, AOI (Automated Optical Inspection) methods such as grayscale analysis and edge detection have limitations. Grayscale analysis can be affected by reflections and shadows, similar colors, reliance on high contrast, and lack of height information. Edge detection can suffer from blurred edges caused by factors such as shooting angle and lighting, impacting detection accuracy. The biggest issue is the inability to detect small features because edge detection relies on edge information, making precise detection difficult.

This research system utilizes two models, YOLO and CNN, for recognition. First, YOLO is used to determine whether the chip to be measured is within the camera's view and to frame it. Then, a second model is used to analyze the quality of the chip. YOLO algorithm and OpenCV are employed to remove unnecessary parts of the sample.

Keywords : YOLO, CNN, AOI, artificial intelligence, visual recognition, object tracking, chip inspection

目錄

1. 緒論.....	8
2. 研究動機與研究問題.....	9
2.1 研究動機.....	9
2.2 研究問題.....	9
2.2.1 如何建立晶片辨識的自動化系統.....	9
2.2.2 如何提高晶片辨識的準確性和效率.....	9
2.2.3 如何應對晶片多樣性和變異性.....	9
3. 研究方法及步驟.....	10
3.1 影像特徵判別技術.....	10
3.2 具影像學習光學檢測系統檢測流程.....	11
3.2.1 物品取樣.....	11
3.2.2 資料庫建置.....	11
3.2.3 物件辨識模型選用.....	12
3.2.4 使用 YOLO 初始權重的優點.....	13
3.3 混淆矩陣結構介紹.....	14
3.4 模型性能分析.....	15
3.4.1 精確率(Precision)和召回率 (Recall).....	15
3.4.2 總體準確度 (Overall Accuracy).....	15
3.4.3 訓練損失 (train/box_loss, train/obj_loss, train/cls_loss).....	16
3.4.4 驗證損失 (val/box_loss, val/obj_loss, val/cls_loss).....	16
3.5 性能指標趨勢.....	16
3.5.1 精確度和召回率 (metrics/precision, metrics/recall).....	16
3.5.2 平均精度 (mAP, metrics/mAP_0.5, metrics/mAP_0.5 : 0.95).....	16

3.6 影像預處理.....	17
3.7 卷神經網路.....	18
3.7.1 模型架構.....	18
3.8 影像機器學習.....	20
3.9 物件識別模型.....	20
3.10 晶片檢測模型.....	20
3.10.1 PyTorch：.....	20
3.11 檢測結果輸出.....	21
3.11.1 檢測系統結果展示.....	22
3.11.2 檢測參數設置.....	22
3.11.3 結果輸出與應用.....	22
3.12 專題過程中遇到的問題.....	22
3.12.1 何謂果凍效應?.....	23
3.13 解決方案.....	23
3.13.1 果凍效應：在高速移動下所產生的形變。.....	24
3.13.2 軟體卡頓：辨識程式在樹莓派上執行導致的內存問題。.....	24
4. 軟硬體整合.....	24
4.1 精密電控載台系統(Tr6*1 及 28 型步型步進電機).....	25
4.2 Skr Pico.....	26
4.3 樹莓派.....	27
4.4 顯微鏡.....	28
4.4.1 使用原因.....	29
4.4.2 果凍效應比較.....	29
4.5 優缺點.....	30

4.5.1 優點.....	30
4.5.2 缺點.....	30
4.6 電源供應器.....	30
5.人機介面(GUI).....	31
5.1 使用工具.....	32
5.1.1MobaXterm SSH 終端連線.....	32
5.1.2VNC	32
5.2PyQt 與 Tkinter 的比較與應用.....	34
5.2.1PyQt5 函式庫	34
5.2.2Tkinter.....	34
5.2.3 最終使用 PyQt5 的原因	34
5.3Klipper 控制系統	35
5.3.1Klipper 的優點：	35
5.3.2Klipper 控制系統如何控制載台	36
5.4Mainsail(影像串流).....	36
5.4.1 實時狀態監控.....	37
5.4.2 攝像鏡頭配置.....	37
5.4.3 加熱床控制.....	37
5.4.4XYZ 軸控制	37
5.4.5 G-code 文件管理.....	37
5.4.6 打印控制.....	38
5.5OpenCV	38
5.6GUI 畫面及功能介紹：	39
5.6.1 畫面：	39

5.6.2GUI 功能介紹.....	40
6.預期結果.....	42
6.1 優化檢測.....	42
6.2 操作簡易.....	42
6.3 檢測之精確度調整.....	43
6.4 檢測參數及時調整.....	43
7.未來展望.....	43
7.1 硬體設備升級.....	43
7.2 檢測效能提升.....	43
7.3 系統整合強化.....	44
7.4 應用範圍擴展.....	44

1. 緒論



圖 1 專題研究流程

本專題利用 AI 影像辨識技術來檢測太陽能電池晶片製造過程中的缺陷，以提高生產效率和產品品質。太陽能電池晶片的製造過程中常常面臨裂紋、汙點、結晶不良等缺陷，這些缺陷可能降低電池效率並增加生產成本。本計劃將採用先進的影像辨識技術，包括深度學習和機器視覺算法，對晶片進行快速且準確的缺陷檢測。計劃將首先建立一個包含各種缺陷類型的數據庫，並利用這些數據來訓練和優化影像辨識模型。隨後，我們將開發一個自動化的檢測系統設備，將影像辨識技術整合到生產線中，以實時檢測晶片的缺陷。最後，我們將通過實驗驗證系統的性能和準確性，並對其進行優化，以滿足太陽能電池製造行業對於高效率和高品質的需求。本計劃的實施將有助於提高太陽能電池晶片的製造效率，降低生產成本，推動太陽能產業的發展。

太陽能電池作為清潔、可再生能源的重要來源，已經在全球範圍內得到廣泛應用。然而，太陽能電池的生產過程中常常面臨各種挑戰，其中包括製造過程中可能出現的缺陷問題。這些缺陷可能對太陽能電池的性能和產品品質產生不利影響，進而影響到太陽能產業的可持續發展。

隨著科技的不斷進步，影像辨識技術成為了檢測和分析製造過程中缺陷的一個重要工具。這種技術能夠通過分析圖像和影像來檢測製造過程中的缺陷，具有非接觸式、高效率和高精度的優點。因此，我們利用影像辨識技術來檢測太陽能電池晶片製造過程中的缺陷，將有助於提高生產效率，降低生產成本，並改善產品品質。

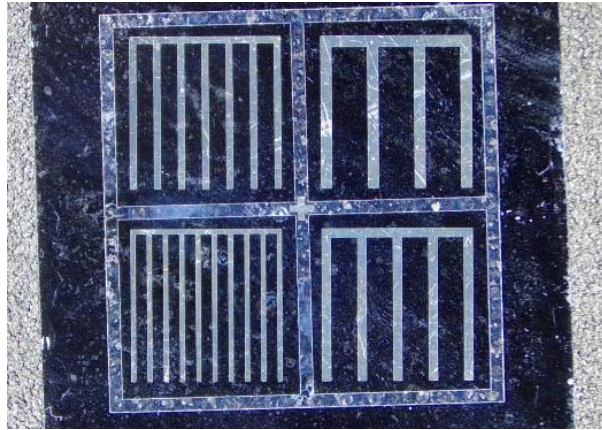


圖 2 晶片照片

2. 研究動機與研究問題

2.1 研究動機

在半導體製程中，最後的步驟往往是人工目視，確保晶片在製程過程中沒有過度蝕刻或是其他的缺陷，但是人工目視需要耗費大量的人力資源以及時間成本，還需考慮人為造成的誤差，所以自動化辨識晶片的想法便成為了一種解決方案，以智慧識別取代人工目視。太陽能電池晶片是一個類別較多且相對容易取得樣本，因此被選為研究對象。其特徵明顯，易於區分，這使得對其進行自動化辨識具有一定的可行性。

2.2 研究問題

2.2.1 如何建立晶片辨識的自動化系統

首要的研究問題是如何開發一個能夠自動辨識太陽能電池晶片的系統。這涉及到選擇合適的技術和方法，可能包括機器學習、計算機視覺、載台控制、人機介面撰寫等技術。

2.2.2 如何提高晶片辨識的準確性和效率

自動化辨識系統需要具備高準確性和高效率，因此需要研究如何通過優化算法、增加特徵標示的精度、載台精密控制等方式來提高辨識的準確性和效率。

2.2.3 如何應對晶片多樣性和變異性

太陽能電池晶片可能存在多樣性和變異性，包括不同型號、不同廠商生產的晶片等。因此，需要研究如何應對這種多樣性和變異性，提升神經網路撰寫讓辨識系統並擁有更新資料

庫的能力，以日後辨識的種類更多，使得辨識系統具有更好的泛化能力。



圖 3 AOI 示意圖

3. 研究方法及步驟

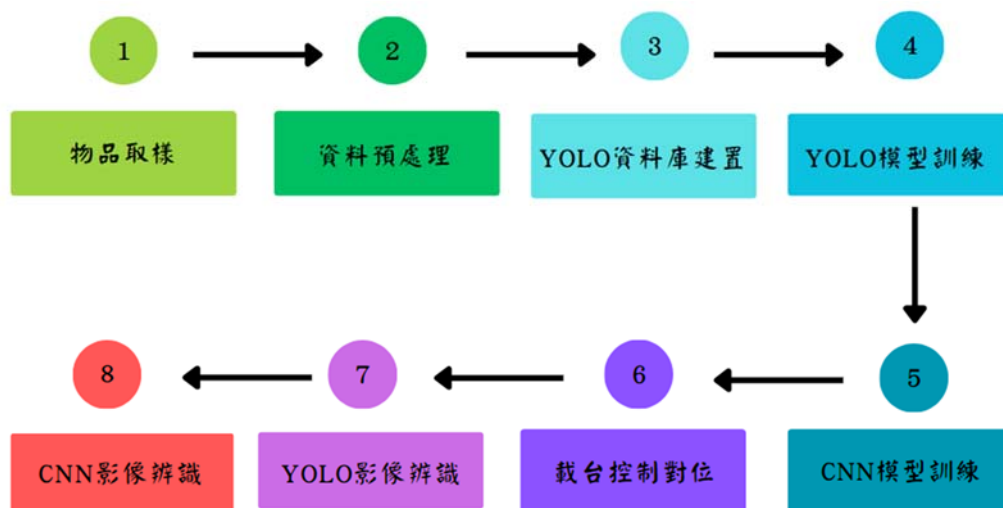


圖 4 整體系統流程圖

3.1 影像特徵判別技術

本專題的核心技術在於設定太陽能電池晶片的檢測範圍，再以大數據及機械學習進行分析，為不同使用者建立專屬的檢測資料庫。在本計畫中，我們選擇使用 PyTorch 軟體來完成

後續影像比對及大數據分析。

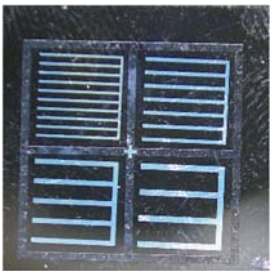
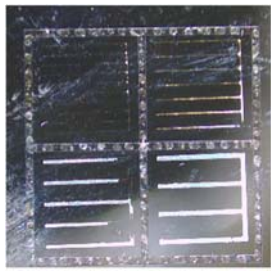
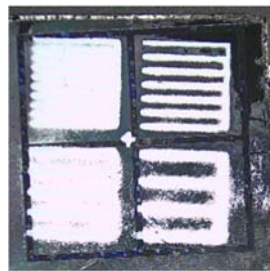

3.2 具影像學習光學檢測系統檢測流程

我們將影像學習光學檢測系統的流程細分為幾個主要步驟：包括檢測物物品取樣、影像預處理、卷神經網路、影像機器學習，以及檢測輸出結果的輸出。以下將對每個步驟進行詳細解說。

3.2.1 物品取樣

取得合適待測物影像，以便後續影像機器學習及數據分析。

表 1 物品取樣圖片展示

			
完好的晶片	曝光顯影的問題	蝕刻(一)	蝕刻(二)

3.2.2 資料庫建置

本專題拍攝 600 張樣品圖片，以比例 80%及 20%拆解成訓練用照片以及測試用照片。

我們使用 CVAT 去標記晶片位置。CVAT(Computer Vision Annotation Tool)是一款免費、開源、基於網頁的圖像和視頻標記工具，用於標記電腦視覺演算法的資料，廣泛應用於各種電腦視覺任務，包括物體檢測、圖像分割和影像追蹤等。



圖 5 CVAT 標記過程

標記的目的是為了進行物件辨識(Object detection)，透過框選目標物取得 XY 座標，將目標照片以及框選物的座標輸入至物件辨識模型網路中訓練。

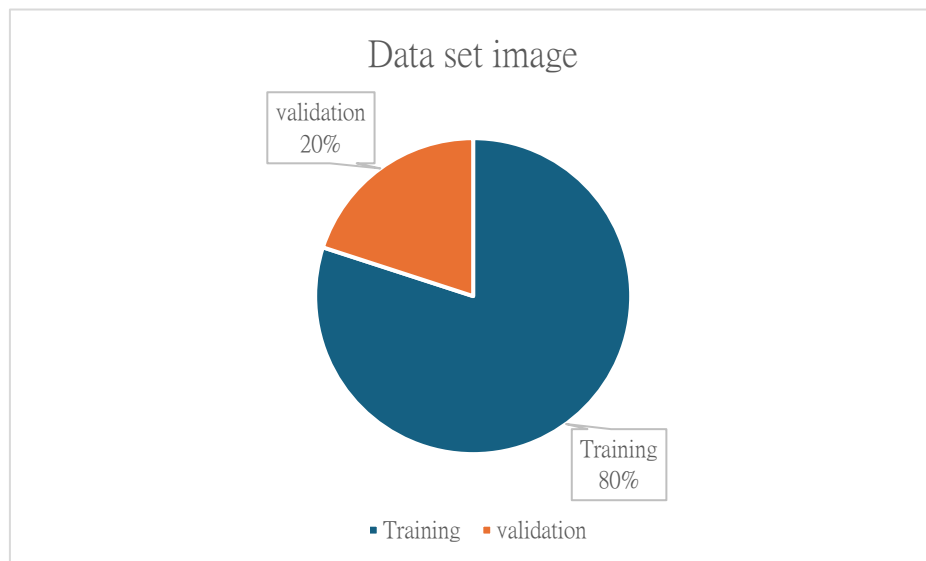


圖 6 資料庫樣品比例

3.2.3 物件辨識模型選用

在晶片檢測上，速度以及精準性是主要的考量因素，因此 One-stage 模型較為適合運用在此專題。One-stage 模型的是只需要一個深度神經網路就可以完成物件偵測，較為著名的模型為 2015 年由 Google 提出的 Single Shot Detector(SSD)以及由 Joseph Redmon 等人在 2015 年提出的 You Only Look Once(YOLO)。

YOLO(You Only Look Once)為 One-stage 物件辨識模型，其特點為在快速的辨識下還具有一定的精準度，相比於相同為 One-stage 的 SSD 模型，YOLO 具有更快的辨識速度如圖...所示，所以成為本專題在物件辨識上使用的模型。

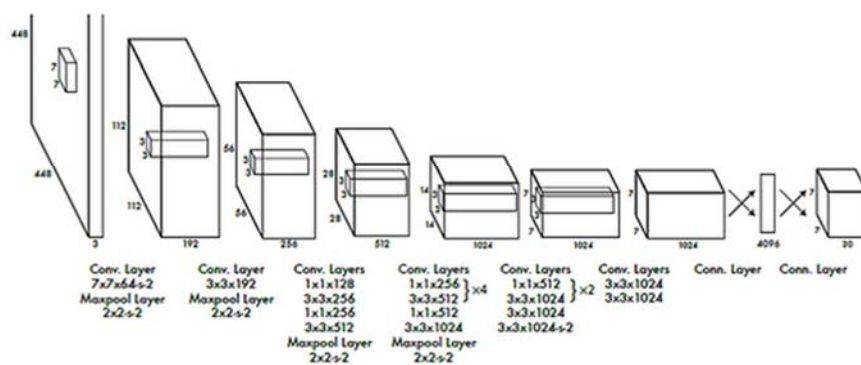


圖 7 YOLO 架構圖(One-stage)

YOLO 在訓練時具有可調整的初始權重，這些參數通常是官方在一些大型數據集上預先

訓練好，YOLO 提供 5 種不同的初始權重給開發者使用。

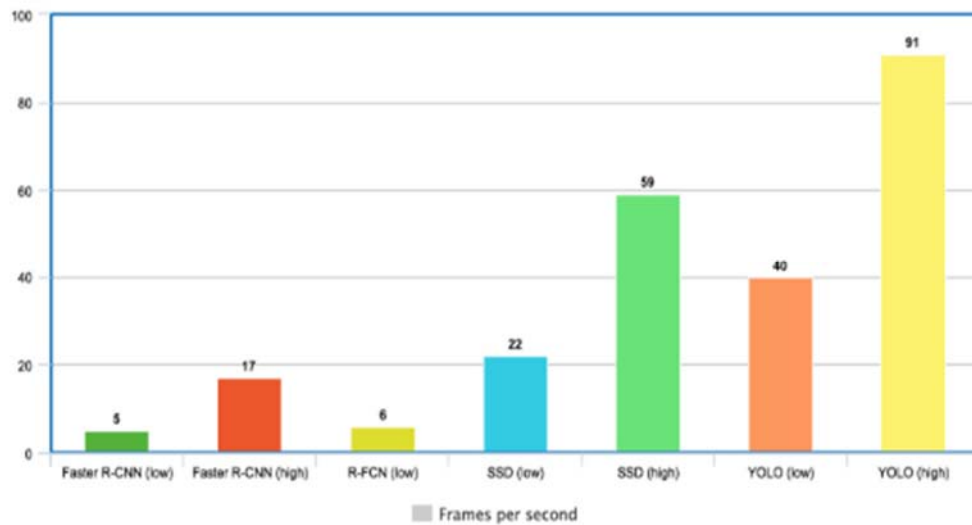


圖 8 模型速度比較

3.2.4 使用 YOLO 初始權重的優點

(1) 加速模型收斂

小資料集從頭開始訓練通常很難收斂。這時預訓練的權重就成為了很好的開始點，可以使模型達到更好的準確度水平。

(2) 避免過擬合

在小資料集上，例如本專題 600 張晶片資料，可有效的規範模型，使其不易過擬合。

(3) 遷移學習

初始權重的選用取決於運用場景，以 YOLOv5n 舉例，nano 擁有最小的模型體積，因次適合部屬於行動裝置上，但準確率也是 5 個之中最低的，只能辨別簡單的物體。本專題使用 YOLOv5s 作為初始權重，因為我們只需要切除晶片的無效區，處理後交給後面的 CNN 做好壞辨識，並不需要做太複雜的判斷。

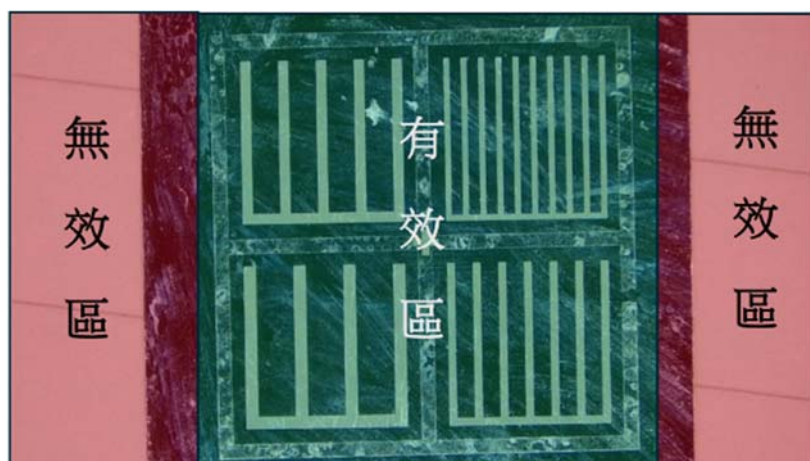


圖 9 無效區及有效區

(4) 訓練結果

混淆矩陣（Confusion Matrix）是一種特殊的表格，用來評估分類算法的性能。它的每一列代表實際的類別，而每一行代表預測的類別。混淆矩陣提供了分類器在不同類別上的正確和錯誤分類的詳細信息。

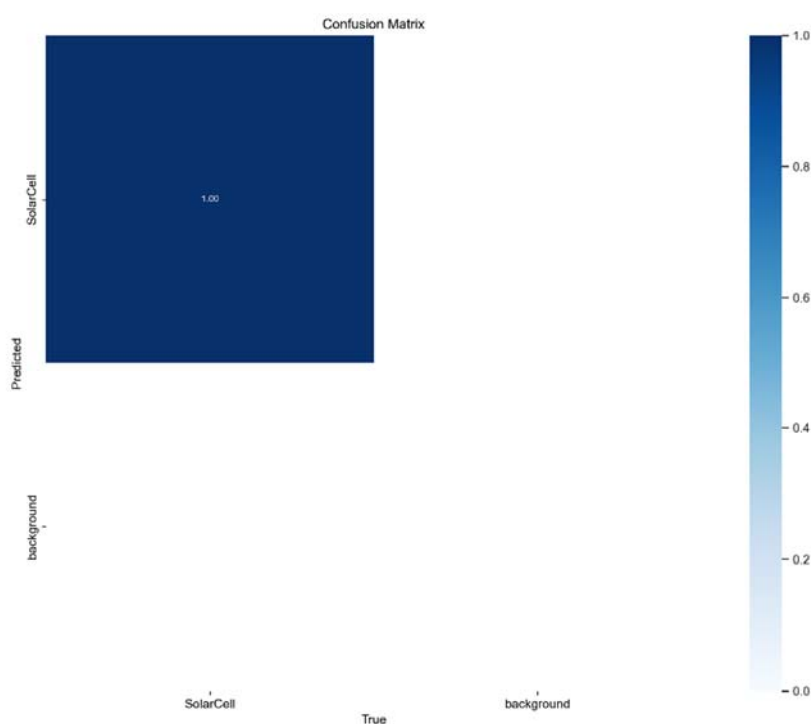


圖 10 混淆矩陣

3.3 混淆矩陣結構介紹

左上角 (SolarCell, SolarCell)：這個方格表示真實標籤為“SolarCell”的樣本數量中，模型正確預測為“SolarCell”的數量。在本例中，這個值為 1.0，表明所有真實為“SolarCell”的樣本都

被模型正確地預測為“SolarCell”。

右下角 (background, background): 這個方格表示真實標籤為“background”的樣本數量中，模型正確預測為“background”的數量。在這個矩陣中，此部分是空白的，表示沒有真實標籤為“background”的樣本，或是模型未能正確預測這些樣本。

其他部分：左下角和右上角的方格代表分類錯誤的情況。這些部分在該圖中是空白的，這表明模型沒有將任何樣本錯誤地分類。

3.4 模型性能分析

3.4.1 精確率(Precision)和召回率 (Recall)

對於“SolarCell”類別，模型的精確率和召回率均為 100%，因為所有真實為“SolarCell”的樣本都被正確預測，且沒有誤報。

由於缺少“background”類別的樣本，我們無法確定模型在這一類別上的表現，因此在報告中需要進一步討論或補充相關數據。

3.4.2 總體準確度 (Overall Accuracy)

總體準確度是正確預測數量除以總樣本數。在這個例子中，由於所有真實標籤為“SolarCell”的樣本都被正確預測，總體準確度達到 100%。

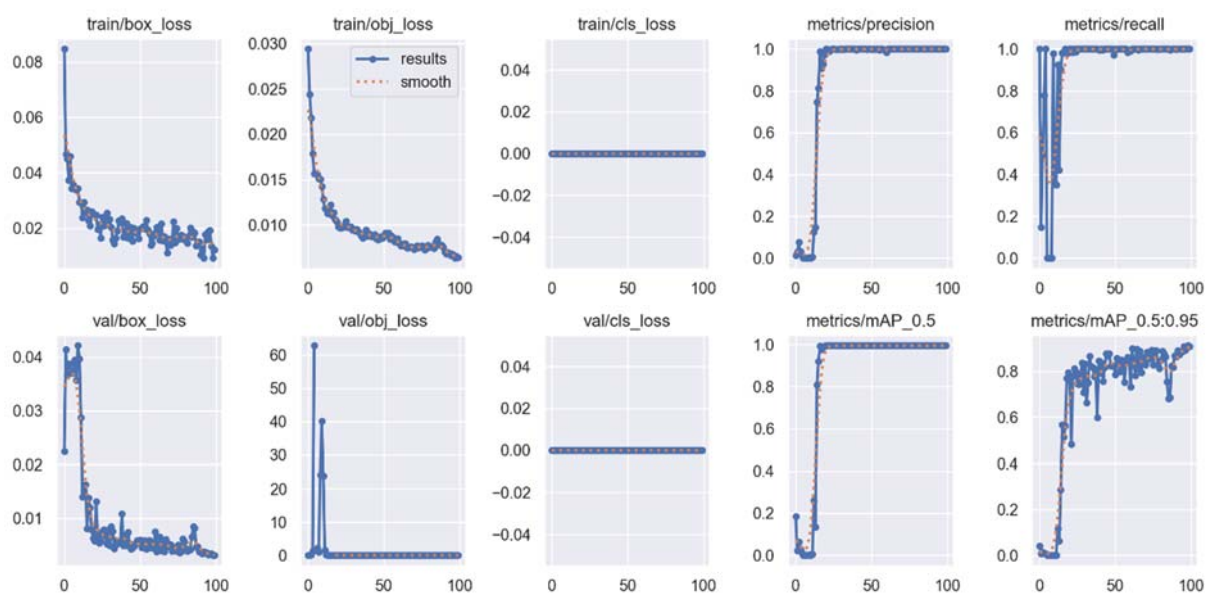


圖 11 損失函數趨勢

3.4.3 訓練損失 (train/box_loss, train/obj_loss, train/cls_loss)

- (1) train/box_loss：邊界框回歸損失隨著訓練的進行穩步下降，顯示模型在預測邊界框位置的精度逐漸提高。
- (2) train/obj_loss：目標檢測損失逐漸減少，表明模型在識別物體存在方面的性能提升。
- (3) train/cls_loss：分類損失保持在零，可能是數據集中不存在分類標籤或模型在這方面表現穩定。

3.4.4 驗證損失 (val/box_loss, val/obj_loss, val/cls_loss)

- (1) val/box_loss：驗證階段的邊界框損失顯示出類似的下降趨勢，驗證了模型的泛化能力。
- (2) val/obj_loss：目標檢測損失在驗證階段也逐漸減少，進一步表明模型的良好性能。
- (3) val/cls_loss：分類損失保持在零，與訓練損失一致。

3.5 性能指標趨勢

3.5.1 精確度和召回率 (metrics/precision, metrics/recall)

- (1) metrics/precision：精確率隨著訓練的進行迅速上升並穩定在接近 1.0 的位置，表明模型在檢測到的對象中正確檢測的比例很高。
- (2) metrics/recall：召回率同樣迅速上升並穩定在接近 1.0 的位置，表明模型能夠檢測到大部分的真实對象。

3.5.2 平均精度 (mAP, metrics/mAP_0.5, metrics/mAP_0.5 : 0.95)

- (1) metrics/mAP_0.5：在 IoU 閾值為 0.5 時的平均精度迅速上升並穩定在接近 1.0 的位置，表明模型在這一閾值下表現非常好。
- (2) metrics/mAP_0.5 : 0.95：在不同 IoU 閾值範圍內的平均精度逐漸上升並穩定在較高的位置，展示了模型在多種評估標準下的穩定性能。

訓練顯示出良好的趨勢，具體包括：train/box loss 逐漸下降，表明模型在預測邊界框位置的判斷逐漸提高。驗證損失同樣表現出相似的趨勢：val/box loss 下降，驗證了模型的適應能力，而 precision 和 recall 迅速上升並穩定在接近 1.0 的位置，表明模型對檢測到的對象具有很高的正確檢測比例，且能夠檢測到大部分的真实對象。metrics/mAP_0.5 在 IoU 臨界值為 0.5 時的平均準確度迅速上升並穩定在接近 1.0 的位置，展示了模型在該判斷值下的表現良好，

而 metrics/mAP_0.5:0.95 則顯示在不同 IoU 臨界值範圍內的平均精度逐漸上升並穩定在較高的位置，顯示出模型在多種評估標準下的穩定性能。

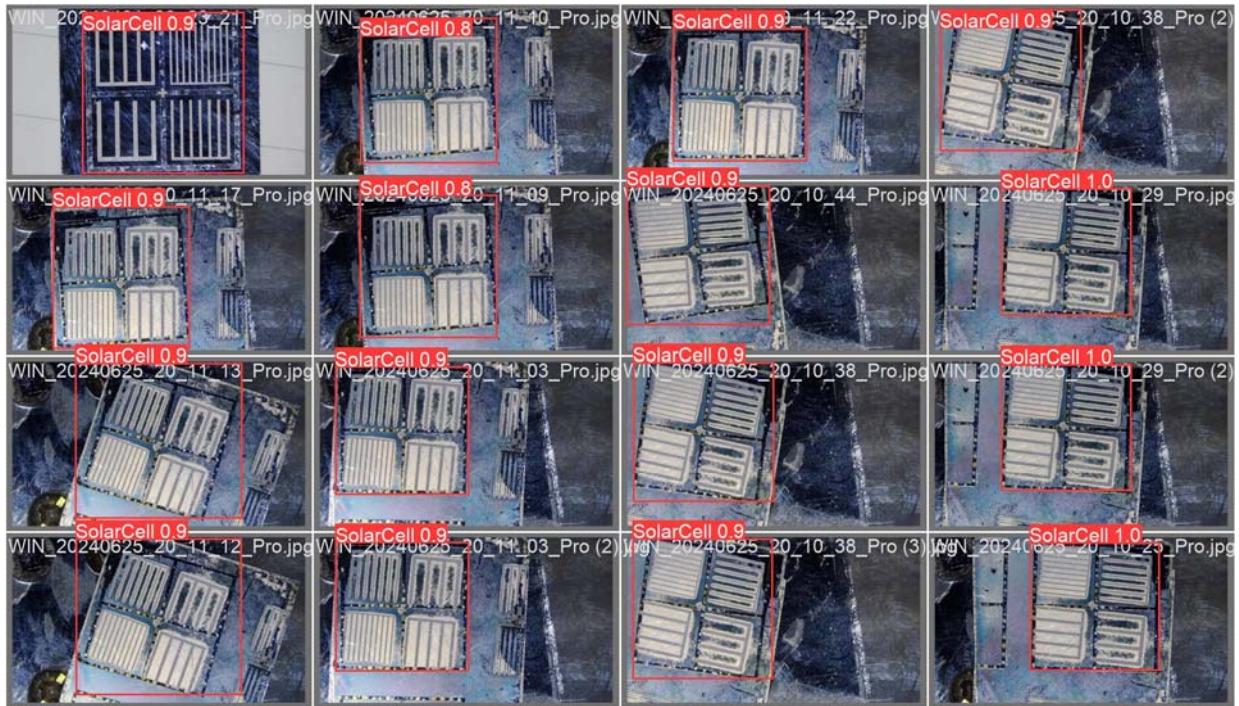


圖 12 實際結果

3.6 影像預處理

取用某些照片時，常常需要進行影像預處理，以確保影像資料的品質、可用性和適應性，從而達到強化學習的目的。例如：調整影像大小、正規化、旋轉、裁切、增加噪點等等。

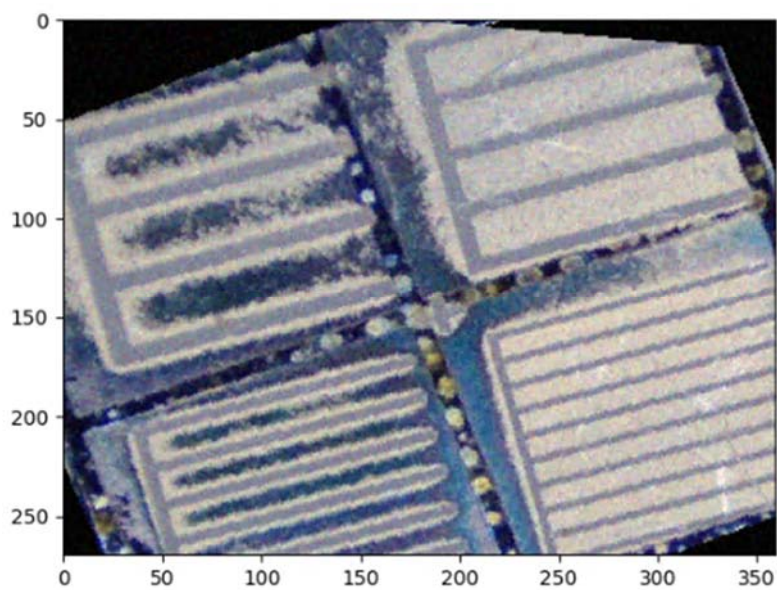


圖 13 影像預處理後的結果

3.7 卷神經網路

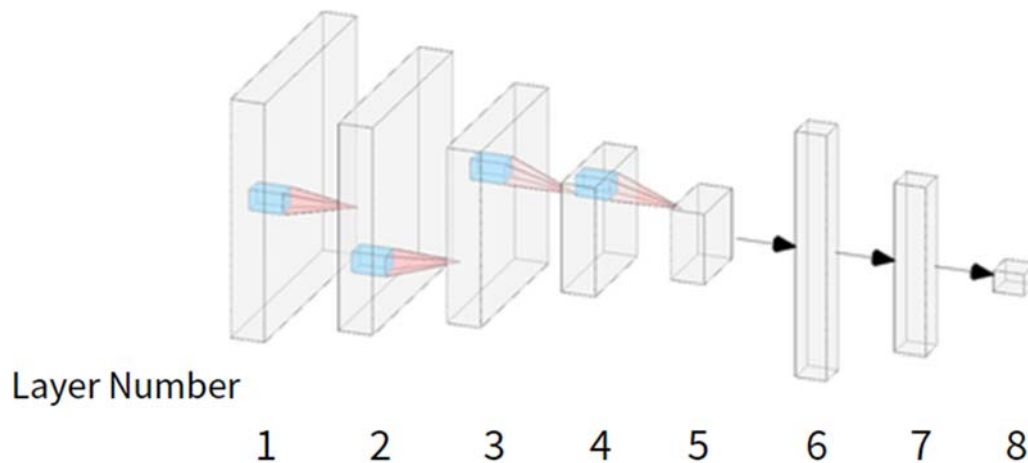


圖 14 CNN 模型架構

3.7.1 模型架構

前五層我們使用卷積層來設計模型，接著再用兩層全連接層進行輸出，完成整個模型設計。

我們使用捲積神經網路（Convolutional Neural Network ;CNN）是為了讓照片能夠更有效地被模型理解和處理。CNN 的強大之處在於它能夠自動學習圖像中的抽象特徵，從而提高對於複雜模式和物體的辨識能力。其中包含了卷積部分(Convolution)、池化部份(Pooling)、全連接層部分(Fully Connection)。

(1) Convolution Layer 卷積層

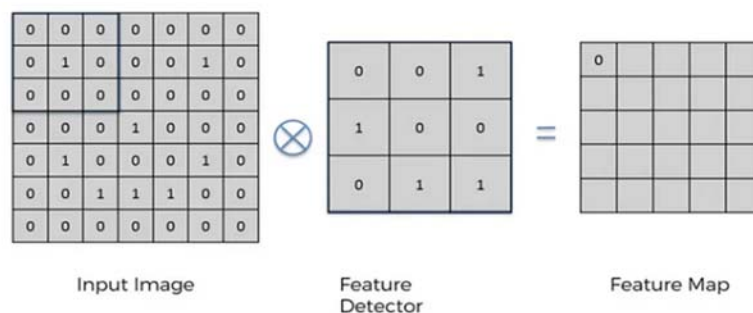


圖 15 利用 Feature Detector 萃取出物體的邊界

卷積運算就是將原始圖片的與特定的 Feature Detector(filter)做卷積運算(符號 \otimes)，卷積運算就是將圖 3 兩個 3x3 的矩陣作相乘後再相加，以圖 16 為例

$$0*0 + 0*0 + 0*1 + 0*1 + 1*0 + 0*0 + 0*0 + 0*1 + 0*1 = 0$$

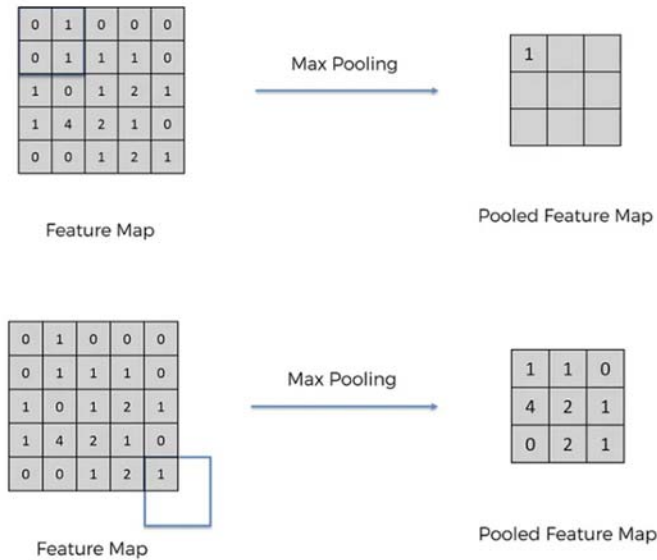


圖 16 Max Pooling

中間的 Feature Detector(Filter)會隨機產生好幾種(ex：16 種)，如圖 17，Feature Detector 的目的就是幫助我們萃取出圖片當中的一些特徵(ex：形狀)，如圖 17，就像人的大腦在判斷這個圖片是什麼東西也是根據形狀來推測。

(2) Pooling Layer 池化層

在池化層中，我們主要使用了最大池化(Max Pooling)的方法。如圖 17，最大池化的概念相當簡單，只需從矩陣中選取最大的值即可。最大池化的主要優勢在於，當圖像整體平移幾個像素時，對於判斷並不會產生影響，同時它還具有良好的抗雜訊能力。

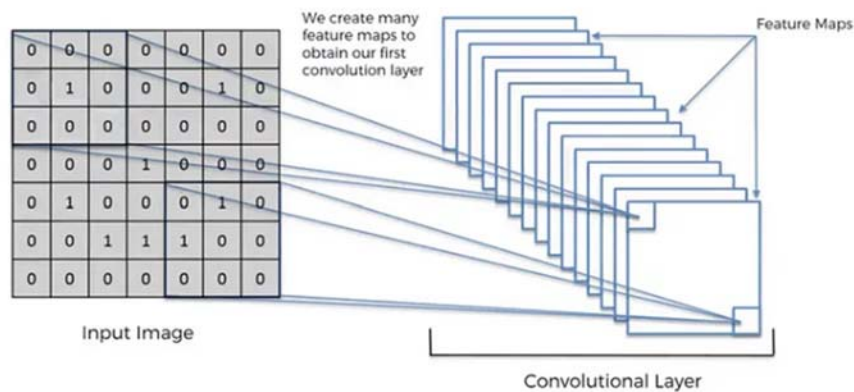


圖 17 ReLu 層

(3) Fully Connected Layer 全連接層

基本上，全連接層的部分就是將之前獲得的結果進行平坦化，然後連接到最基本的神經網絡層，如圖 18。

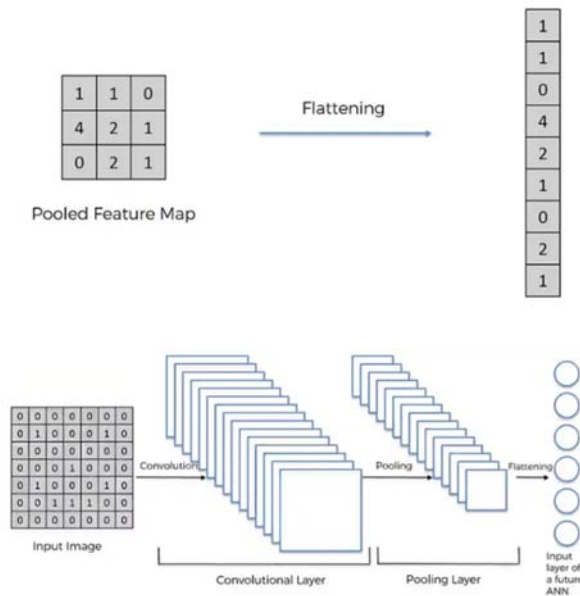


圖 18 Fully Connected Layer

3.8 影像機器學習

我們的專案涉及兩個模型，以達成不同的目標：

3.9 物件識別模型

我們使用 YOLO（You Only Look Once）架構進行物件辨識。這個模型能夠實時檢測並識別圖像中的多個物體，具有高效的處理速度和較高的準確性，非常適合用於我們的應用場景。

3.10 晶片檢測模型

我們利用 PyTorch 框架來開發一個專門用於晶片檢測的模型。PyTorch 提供靈活的動態計算圖和豐富的庫函數，能夠支持我們設計和訓練深度學習模型，以精確地檢測和分析晶片上的缺陷和異常。

這兩個模型的結合，實現了我們專案中對物件識別和晶片檢測的雙重需求，確保了系統的高效性和精確性。

3.10.1 PyTorch：

我們選擇使用 PyTorch Lightning 主要是因為它在影像機器學習領域相對於其他訓練模型的優勢明顯。首先，PyTorch Lightning 提供統一且簡化的訓練循環，減少了大量的繁瑣程式

碼，讓我們能夠更專注於模型架構和實驗設計，同時保持高度的可讀性和可維護性。這對於快速迭代和實驗調整非常有利。

其次，PyTorch Lightning 內建支援自動分佈訓練，使我們能夠輕鬆地擴展到多個 GPU，提高了訓練速度，同時無需手動處理分佈策略的細節。這對於處理大規模的數據集和複雜的模型非常重要。

豐富的內建回調函數也是一個重要的考慮因素。PyTorch Lightning 提供多種回調，如模型檢查點、學習率調整等，這使得在訓練過程中的監控和調整變得更加方便。內建的模型檢查點功能能夠自動保存模型的狀態，確保實驗的可重現性。

總體而言，PyTorch Lightning 在簡化訓練流程、提高可擴展性、提供豐富的回調函數、內建模型檢查點功能以及方便的配置管理等方面優於其他訓練模型方法，使我們能夠更有效率地進行影像機器學習實驗和訓練。

3.11 檢測結果輸出

依據檢測員所調整之參數，將檢測結果輸出至顯示器上。

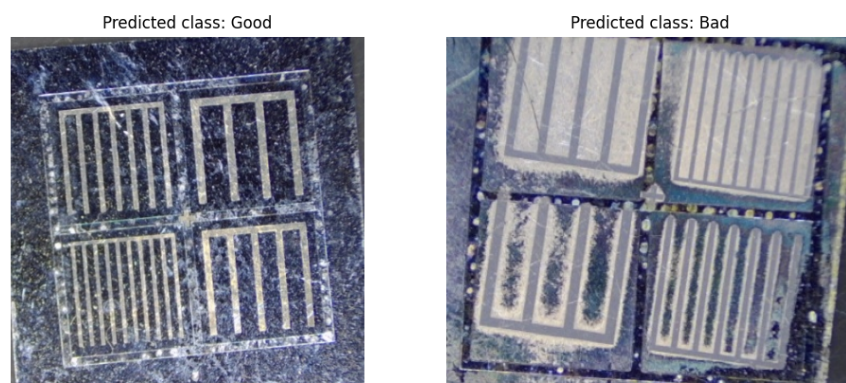


圖 19 模型檢測結果

這張圖片展示了使用卷積神經網絡（CNN）對太陽能晶片進行檢測的結果。圖中每個晶片都被紅色邊框標出，並標註了"SolarCell"的標籤以及一個信任分數（例如 0.8、0.9 等）。這些分數表示模型認為該區域是太陽能晶片的信心程度。這套模型架構使用了多層卷積層、批量歸一化層、活化函數（ReLU）和池化層。簡單來說，這些層用來從圖像中提取有用的特徵。然後，這些特徵通過全連接層進行處理，最終模型輸出了每個類別的機率分佈。模型使用 Softmax 函數來計算這些機率，這些機率反映了模型對每個偵測結果的信心，也就是圖中每個晶片的置信度分數。總的來說，這個 CNN 模型透過提取影像特徵，然後將這些特徵轉換為信任分數，來識別太陽能晶片，圖中的結果就是模型實際運行時的表現，顯示了每個晶片檢測到的準確度。

3.11.1 檢測系統結果展示

檢測系統基於影像處理技術，對樣品進行分類，並將結果輸出到顯示器上。以下為兩個實例結果展示：

圖左（預測類別：良好）

描述：系統對這個樣品進行分析，判定其屬於「良好」類別。這表明該樣品的圖案均勻，無明顯缺陷或不規則性。

特徵：均勻的圖案結構，無明顯缺陷。

圖右（預測類別：不良）

描述：系統對這個樣品進行分析，判定其屬於「不良」類別。這表明該樣品存在圖案不均勻、缺陷或結構異常。

特徵：圖案不均勻，存在明顯缺陷。

3.11.2 檢測參數設置

檢測結果的準確性和可靠性依賴於檢測員所設置的參數，包括：

缺陷大小的閾值：設定可以接受的缺陷尺寸範圍。

均勻性測量：評估圖案的一致性。

圖案識別標準：確定什麼樣的圖案可以被視為良好。

3.11.3 結果輸出與應用

檢測結果輸出到顯示器上，供檢測員進行即時的確認與決策：

即時顯示：每個樣品的圖片和預測類別標籤（「良好」或「不良」）會立即顯示在螢幕上。

應用場景：該系統主要應用於生產線上的品質檢測，通過自動化流程減少人為錯誤，提高生產效率。

3.12 專題過程中遇到的問題

將影像辨識程式和 GUI 程式檔透過 MobaXterm 遠端使樹莓派執行程式時發現該 CPU 使用率過高。

3.12.1 何謂果凍效應？

果凍效應，或稱捲簾現象(Rolling Shutter Effect)。是一種在數碼攝影和影像捕捉過程中的扭曲和變形。

捲簾快門效應 具體描述了相機在捕捉影像時，感光元件（例如 CMOS 傳感器）並非同時對整個影像進行曝光，而是逐行或逐列地進行掃描。這意味著影像的不同部分是在不同的時間記錄的，通常是從上到下或從左到右依次進行。這種掃描方式會導致以下現象：

快速移動的物體變形：當攝像機或拍攝物體快速移動時，會出現垂直方向的彎曲或扭曲。例如，快速移動的車輛或旋轉的風扇葉片會出現變形。

閃光燈效應：如果有快速閃爍的光源（如螢光燈或 LED 燈），影像中可能會出現橫條紋或光影的變化，因為不同的掃描行在不同的時間點被曝光。

影像呈現出像果凍般的波浪形扭曲。這種效應在手持拍攝或快速移動的情況下特別明顯。這種果凍效應導致的視覺效果可能會使螺旋槳看起來像是一個動態、柔軟的物體，而不是一個堅硬的機械部件。



圖 20 車廂的模糊和拉伸效果增強了車廂的高速運動感



圖 21 螺旋槳在高速下捕捉的畫面

3.13 解決方案

主要遇到的問題有以下幾點：

3.13.1 果凍效應：在高速移動下所產生的形變。

在自動識別中，精確度為我們主要的考量因素，特別是在晶片辨識，我們透過將馬達轉速調低，進而使載台移動距離縮短，雖然會減少辨識速度，但精確度以及果凍效應都有所改善。

3.13.2 軟體卡頓：辨識程式在樹梅派上執行導致的內存問題。

我們所使用的樹梅派型號為 Raspberry Pi 4 它所擁有的記憶體內存為 4G，並不足以支撐我們的辨識模型以及 GUI 程式，其佔用率會達到 80~100%，導致辨識時會產生閃退、馬達失控的各種問題。

後續我們直接採用電腦作為控制器，雖然整體設備的體積會變大，但在電腦上操作可以使用 Klipper 更加細部的調整馬達的參數。

4. 軟硬體整合

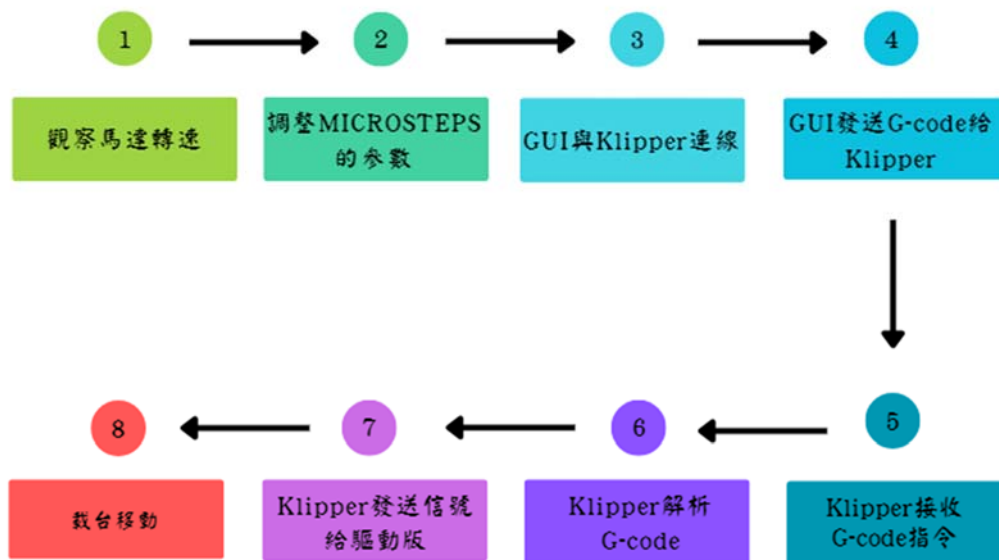


圖 22 載台校準流程圖

為了將待測物移動到鏡頭下方，我們需要一個可移動的雙軸載台。此計畫使用兩個單軸 28 步進馬達滑台，透過樹莓派（Raspberry Pi 4）和馬達驅動板（TMC2209）來實現。控制人機介面發送訊號給 Klipper，以執行預先編寫的 G-Code 來控制載台移動，並記錄當前座標，以便後續配合影像辨識的結果呈現。

4.1 精密電控載台系統(Tr6*1 及 28 型步型步進電機)

(1) Tr6*1 螺桿的特性

高精度螺紋設計：Tr6*1 表示絲桿的直徑(D)為 6 毫米，牙距(P)為 1 毫米以及 30° 的 ISO 公制梯形螺紋。這種螺紋設計每轉一圈的移動距離較小，從而提高了運動的精確度。

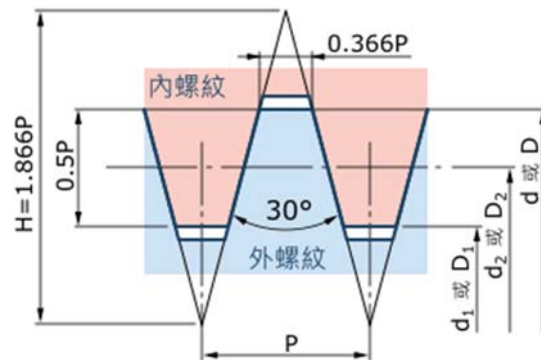


圖 23 ISO 公制梯形螺紋

(2) 28 型步進電機的優勢

基本步數：28 型步進電機通常具有 200 步/轉 (1.8° /步) 的基本步數。

細分驅動：通過使用電機驅動器(TMC2209)，可以將步進電機的步數進一步細化，最大可以達到 1/256 的細分，使每轉步數達到 51200 步 (0.00703125° /步)。馬達的細分數越多，通常會導致馬達的移動速率變慢。這是因為細分數增加後，每一步的移動距離變小，所以在相同時間內，馬達需要執行更多的步數才能完成相同的位移。這樣雖然可以提高定位的精度和平滑度，但會降低整體的移動速度，因此，我們採用 128 細分步數，這樣既能保持適當的速度，又能達到精確的定位效果，實現速度和精度的平衡。



圖 24 高精準電動載台

表 2 TMC2209 規格介紹

型號	TMC2209
----	---------

最大細分	1/256 細分
電壓範圍	4.75V 至 28V
最大電流	2A RMS，峰值 2.8A
UART 接口	支持，便於配置和調試
故障檢測	支持短路、過溫、欠壓等故障檢測
驅動模式	支持靜音模式、SpreadCycle 等多種模式
應用場景	3D 打印機、CNC 機床、機器人等需要精確控制的設備
優勢	提高定位精度和平滑度 降低噪音 節能效果好
適用步進電機	NEMA 17、NEMA 23 等

表 3 TMC2209 的 StealthChop、SpreadCycle 模式比較

比較項目	StealthChop	SpreadCycle
噪音水平	極低，適合安靜環境	較高，但提供更高的性能
平滑度	非常平滑，減少振動和共振	相對較平滑，但不如 StealthChop
扭矩	扭矩較低，適合低速應用	高扭矩，適合高速和高負載應用
效率	效能較低，可能會更熱	效率較高，適合需要持續高效運行的應用
應用場景	家用打印機和需要安靜運行的設備	工業設備和需要高性能的應用
適用範圍	低噪音環境中的低速、低負載應用	高負載、高速和需要高動態性能的應用

4.2Skr Pico

Raspberry Pi Pico 可兼容並利用其強大功能和靈活性來控制 3D 打印機。提供多種接口，包括步進電機驅動器（TMC2209）接口、加熱器接口、風扇接口和熱敏電阻接口，滿足各種 3D 打印機的需求。支持 Marlin 和 Klipper 等開源固件，用戶可以根據需要定制和優化打印機的控制程序。具備過流、過熱保護等安全功能，確保使用過程中的穩定性和安全性。

表 4 Marlin 與 Klipper 比較

比較項目	Marlin	Klipper
性能和速度	主要運行在微控制器上，受限於硬體性能，步進速率相對較低	將計算工作分布到外部主機上，微控制器僅處理時間關鍵的任務，允許更高的步進速率

配置和靈活性	配置需要重新編譯和刷寫韌體，調整相對複雜	使用文本配置文件，修改配置簡單，調整和調試方便
Macros 命令	支持 Macros 命令，但靈活性和功能性不如 Klipper	支持強大的 G-code 功能，允許自定義複雜任務
多平台支持	主要運行在微控制器上，可通過 OctoPrint 等軟體控制	支持多平台的外部主機（Linux、Windows、MacOS），可輕鬆整合到 OctoPrint 等軟體中
實際應用體驗	穩定性和功能性得到廣泛驗證，但在處理高複雜度任務時可能不足	實際使用中表現流暢和穩定，特別是在處理複雜打印任務時表現出色

本專題選擇使用 SKR Pico 的原因是希望精密控制步進電機進行，確保電機能夠準確地運行到使用者所設定的位置，這樣我們可以提升系統的精度和效果，達到更高的運行質量和可靠性。



圖 25 SKR Pico

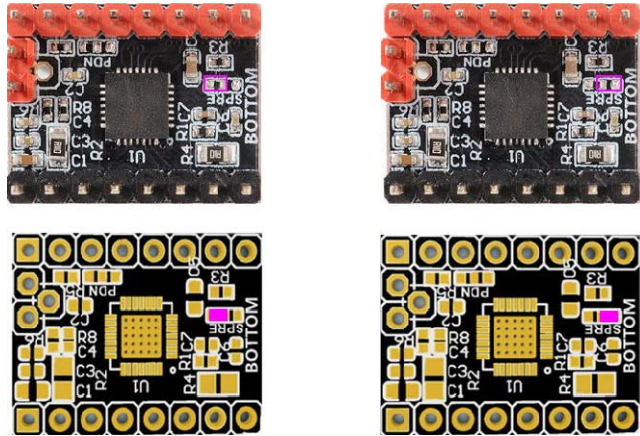


圖 26 TMC2209

4.3 樹莓派

樹莓派是由英國樹莓派基金會開發的微型單板電腦，旨在通過低價硬體和自由軟體促進

學校的基本電腦科學教育。樹莓派系列電腦每一代均使用博通（Broadcom）出產的 ARM 架構處理器。目前生產的機型（樹莓派 5）擁有 2GB 到 8GB 的主記憶體，並使用 TF 卡作為系統儲存媒體（初代使用 SD 卡）。這些電腦配備 USB 介面和 HDMI 影片輸出（支援聲音輸出），內建 Ethernet、WLAN 和 Bluetooth 網路連結功能（依據型號而定），並且可使用多種操作系統。



圖 27 樹莓派

Raspberry Pi OS 是官方推出的操作系統，適用於所有型號的樹莓派。樹莓派基金會的網站還提供了 Ubuntu MATE、Ubuntu Core、Ubuntu Server 和 OSMC 等第三方系統供用戶下載。

經過比較，我們發現 Raspberry Pi 4 在價格、體積和性能方面都完全符合我們的需求。作為微電腦，Raspberry Pi 4 能夠更好地滿足本專題的設備要求。我們選擇使用 4GB 的主記憶體和 Linux 操作系統 Klipper 來控制馬達，因為它的擴充套驗已經安裝配置完成，使用起來非常方便。

4.4 顯微鏡

使用，我們選擇的規格是 48MP 以及變焦 180X。這個產品的使用原因是其能夠透過 HDMI 或是 USB 直接將顯微鏡中的畫面傳輸到 Raspberry Pi 4，方便進行 AI 晶片缺陷辨識。



圖 28 Yizhan 電子螢幕顯微鏡

4.4.1 使用原因

選擇 Yizhan 電子螢幕顯微鏡是因為其可以直接通過 HDMI 或 USB 傳輸高清畫面到 Raspberry Pi 4，這樣我們能夠方便地利用 AI 進行晶片缺陷辨識。其高清畫質和高放大倍率（180 倍）使得觀察晶片細節變得更加容易。

4.4.2 果凍效應比較

果凍效應（Rolling Shutter Effect）主要在 CMOS 感光器中常見，當快速移動或震動時，影像會出現扭曲或波浪狀。CCD 感光器則不會出現這種問題。相比之下，Yizhan 顯微鏡使用的高質量 CMOS 感光器已經在這方面有了很大改善，果凍效應相對較小。

表 5 CCD 與 CMOS 元件比較

特性	CCD (Charge-Coupled Device)	CMOS (Complementary Metal-Oxide-Semiconductor)
感光原理	使用電荷轉移技術	使用半導體技術
圖像質量	高，噪聲較低	一般，高於 CCD
低光性能	優異，低光環境表現出色	一般，在低光環境下噪聲較大
功耗	高	低
成本	高	低
讀出速度	慢	快
果凍效應	無	有，但現代 CMOS 已有改善

耐用性	一般	高
生產難度	高	低
像素密度	一般	高
噪聲控制	良好	較差
靈活性	低	高，容易集成其他功能
應用場景	天文學、醫學成像、 高端攝影	消費級數碼相機、手機攝像頭、 一般攝影

4.5 優缺點

4.5.1 優點

高解析度：48MP 感光器提供了高解析度的影像，適合細節觀察。

高放大倍率：180 倍放大倍率讓我們能夠清楚地觀察到微小缺陷。

方便傳輸：支持 HDMI 和 USB 傳輸，使得影像可以直接顯示和存儲。

即時性：可以即時觀察和分析影像，適合 AI 即時處理。

4.5.2 缺點

果凍效應雖然已有所改善，但在快速移動的情況下仍可能發生。此外，系統依賴外接設備，如 Raspberry Pi 或其他電腦進行影像處理。在傳輸方式和畫質方面，仍有進一步優化的空間。

Yizhan 顯微鏡支持 HDMI 和 USB 兩種傳輸方式。HDMI 可以提供更高的即時性和穩定性，適合即時觀察和顯示。USB 則方便資料傳輸和存儲。畫質方面，由於使用的是 48MP 的高解析度感光器，影像質量高，細節清晰，適合精密分析。

綜合來看，Yizhan 電子螢幕顯微鏡在高解析度和便捷的影像傳輸方面具有明顯的優勢，適合用於 AI 晶片缺陷辨識等需要高精度觀察和分析的應用。

4.6 電源供應器

為了能直接使用 110V 電源來為 Raspberry Pi 4 供電，所以許要一個電源供應器來達成這個目的，電源供應器中有搭載變壓器，以防電壓過高導致 Raspberry Pi 4 燒壞。



圖 29 LRS-100-24 電源供應器

5.人機介面(GUI)

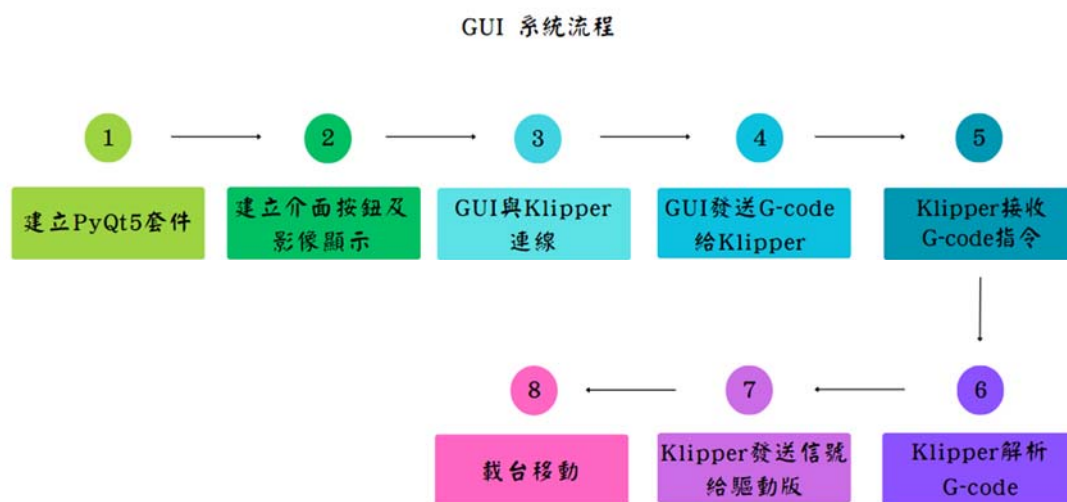


圖 30 GUI 系統流程

在影像辨識-太陽能電池晶片缺陷研究專案中，我們設計了一個直觀的圖形使用者介面（GUI）。這個 GUI 使用 PyQt5 建立，提供了便捷的操作和數據輸入功能。用戶可以使用方向鍵控制載台移動，並通過按鈕選擇移動步數，實現粗調和細調，Home 按鈕可將載台歸位。

介面中包含步數選擇功能，用戶可以輕鬆調整載台的每步移動距離。發送 G-Code 命令功能讓用戶能夠精確控制載台運動，例如輸入 G1 X100 即可移動 X 軸 100 步。

我們還使用 OpenCV 在 GUI 中顯示攝像頭的實時影像，方便用戶即時監控晶片狀況。介面設有拍照按鈕，用戶可以拍攝並保存當前影像，用於記錄和分析。

5.1 使用工具

5.1.1 MobaXterm SSH 終端連線

MobaXterm 是一套功能強大且直觀的視覺化 SSH 客戶端軟體，提供了許多實用功能以提升使用者在操作 SSH 時的效率與便利性。其視覺化資料夾顯示及檔案拖放 SFTP 同步連線功能，使得檔案管理變得簡單且直觀。此外，MobaXterm 還支援輕鬆的複製貼上操作，並能在編輯器中編輯檔案後自動儲存並上傳，極大地提升了使用者操作純終端機的 Linux 系統的便利性。對於剛接觸 Linux SSH 的新手而言，MobaXterm 是一款非常推薦的 SSH 客戶端軟體。

本專題使用 MobaXterm SSH 進行安裝樹莓派上的軟體系統，藉由終端顯示可知道樹莓派的記憶體用量及 CPU 用量，方便於觀察及適時的改變我們在影像辨識的模型參數



圖 31 MobaXterm

5.1.2 VNC

VNC (Virtual Network Computing)，為一種使用 RFB 協定的螢幕畫面分享及遠端操作軟體。此軟體藉由網路，可傳送鍵盤與滑鼠的動作及即時的螢幕畫面。

VNC 與作業系統無關，因此可跨平台使用，例如可用 Windows 連線到某 Linux 的電腦，反之亦同。甚至在沒有安裝使用者端程式的電腦中，只要有支援 JAVA 的瀏覽器，也可使用。



圖 32 VNC

為什麼選擇 MobaXterm：

- (1) 多功能集成：若需要使用多種協議（例如 SSH、SFTP、VNC 等），MobaXterm 可以提供一個統一的平台來管理這些連接，不需要切換不同的工具。
- (2) 便攜性：MobaXterm 的便攜版本可以方便地在不同的電腦上使用，不需要安裝，特別適合經常移動的用戶。
- (3) 圖形化界面：對於不熟悉命令執行操作的人，MobaXterm 提供了易於使用的圖形化界面，

使得遠程管理更加直觀和便捷。

表 6 MobaXterm 與 VNC 比較

功能/特點	MobaXterm	VNC
主要用途	集成多種網絡工(SSH、VNC 等)	桌面共享和遠程控制
便攜性	提供便攜版本，無需安裝即可運行	需要安裝服務端軟件
支持的協議	SSH、SFTP、RDP、FTP、SCP、Telnet 等	RFB 協議
內建工具	提供 Unix 命令和工具	不提供內建 Unix 工具
圖形化界面	提供圖形化界面，便於管理多個遠程會話	主要用於遠程桌面控制，提供基本的圖形化界面
性能和安全性	預設提供 SSH 加密，整體安全性較高	需要額外配置

```
PI4@PI4:~ $ ./kiauh/kiauh.sh

=====
[ KIAUH ]
Klipper Installation And Update Helper
=====

=====
[ Main Menu ]
=====

0) [Log-Upload] | Klipper: Installed: 1(py3)
                  Repo: Klipper3d/klipper
1) [Install]
2) [Update]      | Moonraker: Installed: 1
3) [Remove]
4) [Advanced]   | Mainsail: Installed!
5) [Backup]     | Fluid: Installed!
                  KlipperScreen: Installed!
6) [Settings]   | Telegram Bot: Not installed!
                  Crowsnest: Installed
                  Obico: Not installed!
                  OctoEverywhere: Not installed!
                  Mobileraker: Not installed!
                  OctoApp: Not installed!

                  Octoprint: Not installed!

v5.0.0-101 | Changelog: https://git.io/JnmlX
          Q) Quit

##### Perform action: █
```

圖 33 使用 MobaXterm 安裝於樹莓派的周邊功能

5.2 PyQt 與 Tkinter 的比較與應用

5.2.1 PyQt5 函式庫

PyQt5 是 Python 的一個第三方函式庫，是 Python 用來設計使用者介面（GUI）的函式庫，如果要設計比較美觀，往往會使用 PyQt5 取代 Python 內建的 Tkinter 進行實作，Qt 本身是 C++ 的函式庫，而 PyQt 則是 Qt 的分支，主要是使用 Python 搭配 Qt 進行介面的設計開發，PyQt 除了能利用 Python 語法，也保留了 Qt 的強大功能，大多數使用 Python 所開發出來的功能，都可以搭配 PyQt。



圖 34 PyQt5 函式庫

5.2.2 Tkinter

Tkinter 是 Python 標準庫中的一部分，用於創建圖形使用者介面（GUI），並包含在所有標準的 Python 發行版本中。我們可以使用 Tkinter 來製作桌面應用程式。

傳統上，電腦只有命令行界面（CLI）。圖形使用者介面（GUI）的發明使得電腦變得更加容易學習和使用。



圖 35 Tkinter

5.2.3 最終使用 PyQt5 的原因

(1) 豐富的功能：PyQt5 提供更豐富的小部件、工具和功能，例如佈局管理、圖表繪製、數

據庫連接、網路程式設計等，可以滿足更多複雜的 GUI 開發需求。

(2) 可自訂性：PyQt5 允許自訂 UI 的外觀和風格，讓應用程式擁有更一致的視覺效果。

(3) 活躍的社群：PyQt5 擁有更活躍的社群，可以更容易找到解決問題的答案，以及分享和學習新技術。

表 7 PyQt5 與 Tkinter 比較

特性	PyQt5	Tkinter
跨平台性	支援 Windows、macOS、Linux 等多種平台	支援 Windows、macOS、Linux 等多種平台
學習曲線	相對陡峭，需要學習 Qt 框架	相對平緩，易於入門
效能	高效能，優化性能	效能較低，但仍足以滿足大部分需求
豐富性	功能豐富，提供大量預設控件和自定義選項	功能較少，但仍可滿足基本需求
社群支援	廣泛的社群支援，資源豐富	社群支援較少，但仍有可用資源

5.3Klipper 控制系統

Klipper 是一個開源的 3D 列印機控制軟體，以其高性能和靈活性著稱。它採用微控制器和主機電腦分離的架構，透過串列通訊進行資料傳輸，並利用預先編譯的運動指令，有效降低了處理時間，使列印過程更精確、更快速。



圖 36 Klipper

5.3.1Klipper 的優點：

(1) 高性能控制： Klipper 利用微控制器進行運動控制，有效降低了延遲和抖動，使列印結果

更加精確。

- (2) 靈活配置：Klipper 可以輕鬆配置和自訂各種硬體，包括不同的步進馬達驅動器、熱床控制、感測器等。
- (3) 線性控制：根據速度和加速度進行精確的線性控制。
- (4) 開源且易於使用：Klipper 是一個開源軟體，使用者可以自由地修改和自訂程式碼，並擁有龐大的社群支援。

5.3.2 Klipper 控制系統如何控制載台

Klipper 透過 G-code 指令控制載台的動作，所以我們需要使用 GUI 介面連接 Klipper API 來發送 G-Code。當您將 G-code 檔案傳送至 Klipper 時，它會將指令解析並傳送至微控制器，微控制器則會根據指令控制步進馬達，移動載台。



圖 37 GUI 通訊界面示意圖

以下為我們有使用過的 G-code：

- (1) 移動 (G0 or G1)：G1 [X<位置>] [Y<位置>] [Z<位置>] [E<位置>] [F<速度>]
- (2) 停留時間：G4 P<milliseconds>
- (3) 返回原點：G28 [X] [Y] [Z]
- (4) 關閉步進電機：M18 或 M84
- (5) 使用絕對/相對座標：G90, G91
- (6) 緊急停止：M112
- (7) 獲取目前位置：M114

5.4 Mainsail(影像串流)

Mainsail 是一個專為 3D 打印機設計的網頁界面，提供了豐富的功能來控制和管理 3D 打印機。以下是 Mainsail 的詳細介紹，特別針對你的使用情境進行了說明。



圖 38 Mainsail

5.4.1 實時狀態監控

打印機狀態：顯示打印機的即時狀態，包括噴嘴和加熱床的溫度、XYZ 軸的位置、打印進度等。

溫度曲線：實時顯示噴嘴和加熱床的溫度曲線，便於監控溫度變化。

5.4.2 攝像鏡頭配置

實時視頻監控：通過連接攝像頭，Mainsail 可以實時顯示打印過程的視頻流，便於用戶監控打印進度和質量。

快照功能：支持定時或手動拍攝打印過程的快照，保存和查看打印過程中的關鍵圖片。

5.4.3 加熱床控制

溫度設定：用戶可以通過界面設定和調整加熱床的溫度，確保打印材料的附著性和打印質量。

預熱功能：提供預熱選項，可以預設不同材料的加熱床溫度，以便快速啟動打印。

5.4.4 XYZ 軸控制

手動移動：界面上提供控制按鈕，可以手動調整 XYZ 軸的位置，便於打印前的校準和調整。

運動速度調整：用戶可以設定不同的運動速度，以滿足不同操作的需求，如快速移動或精確定位。

5.4.5 G-code 文件管理

文件上傳：支持通過網頁界面上傳 G-code 文件，便於隨時開始打印新的項目。

文件管理：提供文件瀏覽和管理功能，用戶可以查看、刪除和組織 G-code 文件。

5.4.6 打印控制

開始/暫停/停止打印：用戶可以隨時通過界面控制打印過程，開始新的打印、暫停當前打印或停止打印。

打印參數調整：支持實時調整打印參數，如打印速度、填充密度等，優化打印質量和效率。

本專題使用 Mainsail OS 1.3.2 作為樹莓派的人機操作系統，但此系統為 3D 列印機專用的人機介面系統，非本專題想呈現的人機介面，因此設計另一套適用於 AOI 辨識晶片的 GUI 介面，而影像部分藉由 Mainsail 支援的攝像鏡頭配置，從網頁上能夠獲取影像至本專題的 GUI 畫面。

5.5 OpenCV

OpenCV(Open Source Computer Vision Library) 是一個基於開放原始碼發行的跨平台電腦視覺庫，它同時也支援跨平台，目前現行的各式主流系統諸如 Windows、Linux、Android、iOS 等系統上都有相應的版本，撰寫時以 C/C++ 為主，也同時支援 C#、Python、MATLAB 等程式語言，免費又通用性大的特性使其已然成為最容易取得的影像處理軟體之一。

OpenCV 的主要功能為協助電腦視覺與影像處理效能的增進，它本身自帶高達 500 個以上的以 C 函數為主體的 API，這使它可以不必再依賴其他外部函數庫便能完成一系列的電腦視覺與影像處理等動作，模組化的設計讓使用者在使用這軟體時只需取用自己需要的部分，這大大的減少了匯入函數時所需占用的記憶體空間與時間。與其他影像辨識軟體相比，OpenCV 在開源的理念下，使它有自己的廣大使用者社群，在裡面可以獲得許多開發上的幫助，也能透過此社群讓 OpenCV 保持最新的版本。在實際的生活應用上，幾乎只要有電腦視覺的地方，就會有 OpenCV 的使用，比如人臉辨識、車牌辨識，工業的產品檢測等都有它的存在。



圖 39 OpenCV

5.6 GUI 畫面及功能介紹：

5.6.1 畫面：

為一開始的人機介面，因為 CPU 的使用率高、影像卡頓及美觀的問題，最後我們進行了優化，完成如圖(10)的 GUI 版本，其功能包含利用方向鍵控制載台移動、調整載台移動步數選單、G-Code 命令、顯示實時影像、座標位置、拍照和儲存照片。

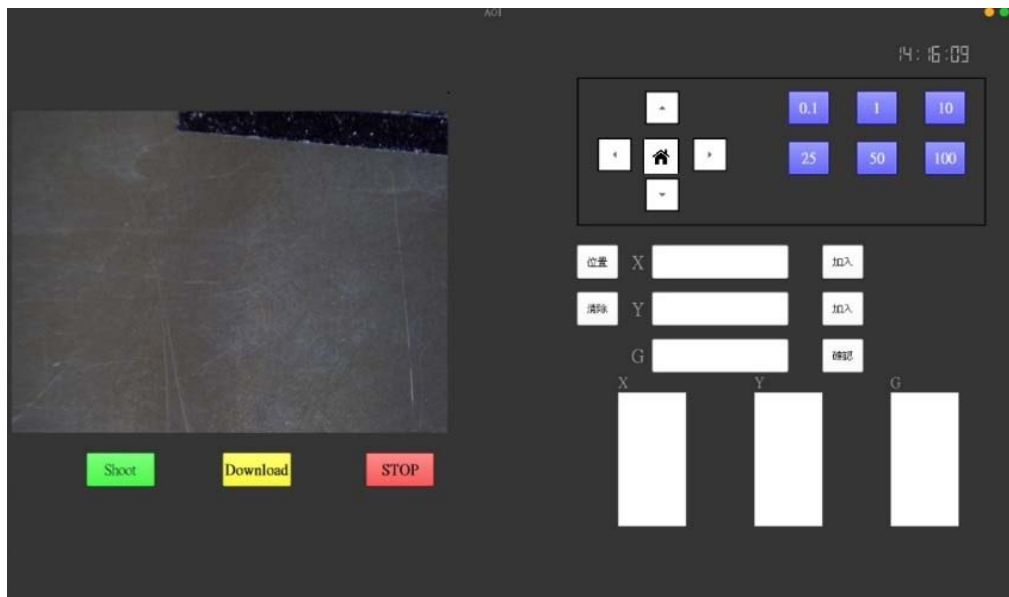


圖 40 初始人機介面

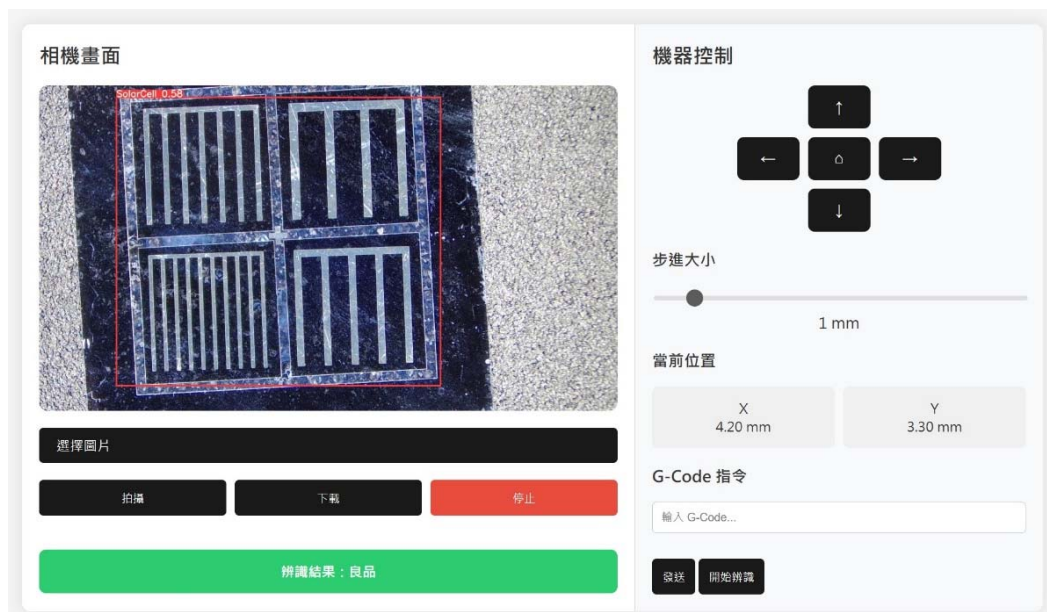


圖 41 最終人機介面

5.6.2 GUI 功能介紹

(1) 建立基本的 GUI 介面：使用 PyQt5 設置基本的介面，並放置各種控件如按鈕、輸入框等。

設置載台控制：方向控制按鈕排列成十字形，用於控制設備的運動方向。向上、向下、向左、向右四個方向的箭頭按鈕可以讓設備沿著相應的方向移動。中間的“房子”按鈕可用來將設備移動到預設的“原點”位置。

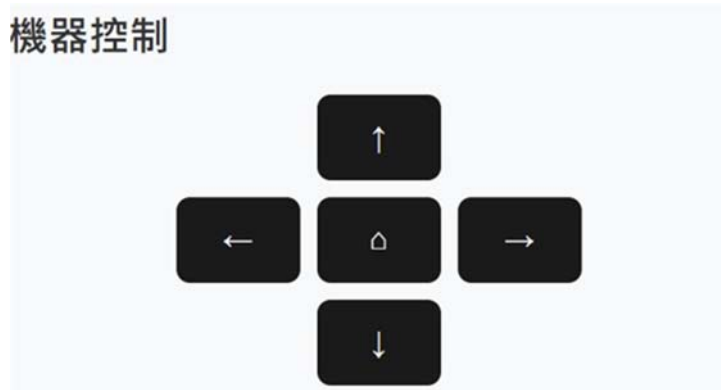


圖 42 控制按鈕

(2) 設置載台控制：有不同的步距選擇每次按下方向控制按鈕時設備的移動距離。，包括 0.1、1、10、25、50、100，單位是毫米或其他合適的單位。



圖 43 設定移動的步距

(3) X 和 Y 位置的顯示：顯示當前 X/Y 軸的位置。

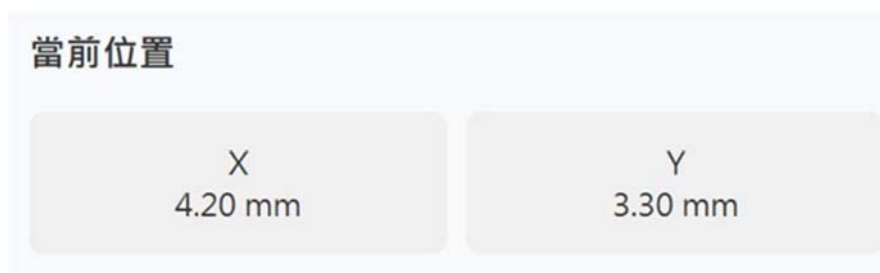


圖 44 X/Y 位置

(4) 發送 G-code 指令和辨識按鈕：在界面中添加一個輸入 G-code 命令的方框，例如輸入 G28 即載台返回起點，按下開始辨識會顯示晶片的好壞。

G-Code 指令

發送

開始辨識

圖 45 輸入 G-code 方框

相機畫面

選擇圖片

拍攝

下載

停止

辨識結果：良品

機器控制

↑

← △ →

↓

步進大小

1 mm

當前位置

X
4.20 mm

Y
3.30 mm

G-Code 指令

發送

開始辨識

圖 46 辨識為良品

相機畫面

選擇圖片

拍攝

下載

停止

辨識結果：不良品

機器控制

↑

← △ →

↓

步進大小

1 mm

當前位置

X
14.10 mm

Y
3.30 mm

G-Code 指令

發送

開始辨識

圖 47 辨識為不良品

(5) 拍照和儲存照片：在界面中添加拍照按鈕，拍攝並保存圖片。

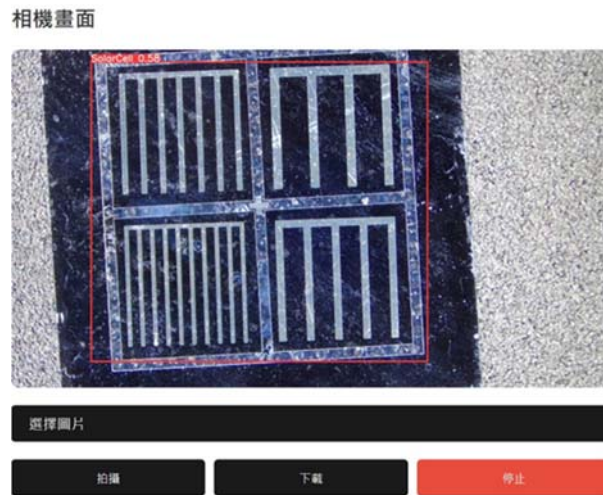


圖 48 GUI 拍照

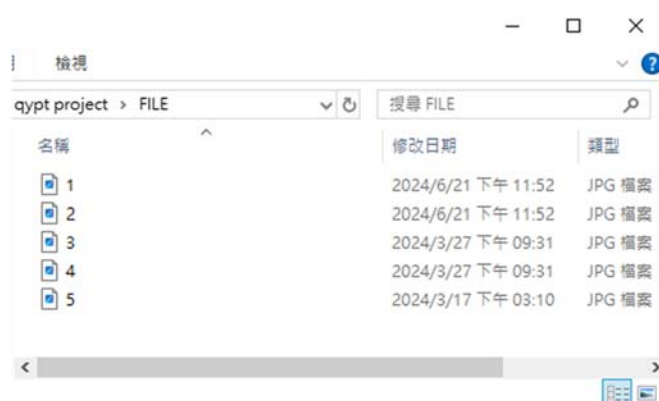


圖 49 GUI 拍照後儲存

6. 預期結果

6.1 優化檢測

精準判斷太陽能電池缺陷，並即時標記，為使用者提供快速而直觀的檢測結果，助力製程優化。

6.2 操作簡易

使用者可經藉由人機介面操作使具影像學習光學檢測系統學習待測物之輪廓後進行檢測。

6.3 檢測之精確度調整

相較於常規光學檢測系統非 1 即 0，具影像學習光學檢測系統在檢測上具有彈性可依照使用者需求來調整。

6.4 檢測參數及時調整

在量測時若要調整參數，例如：改變待測物溫度而造成待測物的些許形變或色變，具影像學習光學檢測系統能及時反應，待使用者修改參數。

7.未來展望

本專題在完成初步的太陽能電池晶片缺陷檢測系統後，仍有許多可持續精進的空間。以下從四個主要面向，說明未來的發展規劃

7.1 硬體設備升級

在硬體設備方面，首要目標是提升整體檢測系統的性能與穩定性。目前使用的 USB 攝影機在高速移動時容易產生果凍效應，影響影像品質，因此計畫升級為工業級高速相機，提高擷取幀率至 120Hz 以上，大幅改善影像品質。同時也將擴充載台尺寸，從現有的 5 吋規格提升至 12 吋晶圓容量，並強化機構剛性，確保大尺寸檢測時的精度與穩定性。照明系統也將進行優化，採用可調式 LED 光源，提供更均勻且穩定的照明效果，有助於提升缺陷檢測的可靠度。

7.2 檢測效能提升

在檢測效能方面，我們規劃透過軟硬體整合的最佳化來提升整體效率。目前載台的移動速度為 20mm/s，未來將通過改善運動控制演算法，將速度提升至 50mm/s 以上，同時優化加減速曲線，減少高速移動時的震動影響。在影像處理與 AI 模型方面，將持續擴充訓練資料集的多樣性，引入更多實際案例進行訓練，提升模型的泛化能力。同時也將優化模型架構，降低運算負載，實現更快速的檢測判定，預期可將單片晶圓的檢測時間從現有的 16 秒縮短至 10 秒以內。

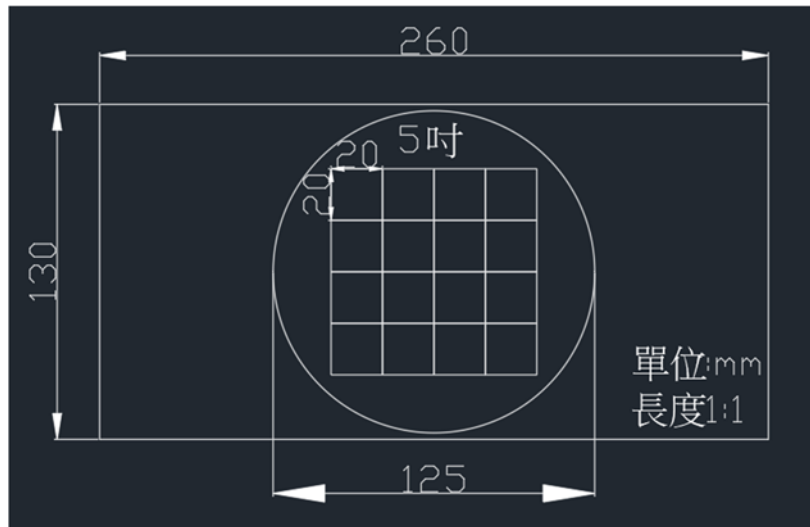


圖 50 檢測範圍示意圖

7.3 系統整合強化

系統整合部分著重於提升操作便利性與數據管理能力。將改善人機介面的操作流暢度，加入更多自動化功能，減少人工操作步驟。同時建立完整的檢測數據管理系統，提供即時良率分析與趨勢追蹤，協助工程師快速掌握製程狀況。此外，也將開發遠端監控功能，讓管理者能夠即時掌握設備運作狀態，提升管理效率。系統也將預留與製造執行系統(MES)的整合介面，為未來導入智慧製造做好準備。

7.4 應用範圍擴展

在應用領域的拓展上，本系統具有很大的發展潛力。首先將在現有的缺陷檢測基礎上，增加更多檢測項目，如表面缺陷、邊緣缺陷等多樣化的檢測能力。系統架構也將朝向模組化發展，除了太陽能電池外，未來可擴展至其他半導體產品的檢測應用。透過建立彈性化的檢測模組，可快速因應不同產品的檢測需求，提供客製化的解決方案。這不僅能擴大系統的應用範圍，也能為相關產業提供更全面的品質管理工具。

參考文獻

- [1] Yeh James , 2017 , [資料分析&機器學習] 第 5.1 講: 卷積神經網絡介紹(Convolutional Neural Network) , <https://reurl.cc/E4Mjov> 。
- [2] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, Soumith Chintala (2019), PyTorch: An Imperative Style, High-Performance Deep Learning Library, <https://doi.org/10.48550/arXiv.1912.01703>.
- [3] 我是土堆 , 2021 , 自製數據集及訓練 | 目標檢測 YOLO v5 項目調試與實戰講解 PyTorch 教程 , https://www.youtube.com/watch?v=AhAMAg5gZ0&list=PLgAyVnrNJ96CqsspB257Seln7_-eIT-Sg&index=11 。
- [4] Computer vision engineer (2023), Object detection with Python FULL COURSE | Computer vision , <https://www.youtube.com/watch?v=UL2cfTTqdNo>.
- [5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin (2023), Attention Is All You Need , <https://doi.org/10.48550/arXiv.1706.03762>.
- [6] STEAM 教育學習網 , 無日期 , 建立 PyQt5 視窗 , <https://steam.oxxostudio.tw/category/python/pyqt5/window.html> 。
- [7] STEAM 教育學習網 , 無日期 , 搭配 OpenCV 實作攝影機拍照和錄影 , <https://steam.oxxostudio.tw/category/python/pyqt5/example-opencv-photo.html> 。
- [8] STEAM 教育學習網 , 無日期 , QPushButton 按鈕 , <https://steam.oxxostudio.tw/category/python/pyqt5/qpushbutton.html> 。
- [9] Gmarull (2020), qtmodern 0.2.0 , <https://pypi.org/project/qtmodern/>.
- [10] 甲客實驗室 , 2021 , Python GUI PyQt 與 TKinter 對比 , https://blog.csdn.net/qz_30404573/article/details/114377477 。
- [11] Shenzhen Bigtree Technology Co., Ltd. (2022), BIGTREETECH SKR Pico V1.0 Operation Manual of the Motherboard, <https://github.com/bigtreotech/SKR-Pico/blob/master/BTT%20SKR%20Pico%20V1.0%20Instruction%20Manual.pdf>
- [12] Klipper (n.d.), G-Codes, <https://www.klipper3d.org/G-Codes.html>.
- [13] 無作者 , 無日期 , [推薦] MobaXterm – 強大方便好操作的 SSH Client 客戶端連線軟體 , <https://richarlin.tw/blog/mobaxterm/> 。
- [14] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg(2016), SSD: Single Shot MultiBox Detector , <https://arxiv.org/pdf/1512.02325>.
- [15] Basilio Pueo(2016), “High speed cameras for motion analysis in sports science ,” *Journal of Human Sport and Exercise*, 11(1): 53-73.