

AI構建自動化分揀與利用系統

作者姓名：劉鑒陞¹、胡舒婷²、黃紫綺³、陳蔡安⁴

職稱：學士，學校：明新科技大學，系所：資工系，Email：a0906608081@gmail.com

教授：陳克任，明新科技大學，資工系，Email：krchen@must.edu.tw

教授：陳美支，明新科技大學，資工系，Email：mindychen@must.edu.tw

摘要

隨著工業自動化的迅速發展，傳統的製造業面臨高效率與高精度的要求，在這樣的環境下，人工智慧(AI)被逐步應用於各行各業，尤其是在生產與質量檢測領域中。在製造業中，螺絲螺帽是常見且關鍵的連接件，廣泛應用於機械、電子、汽車等各類產品中，並且時常重複利用。由於螺絲螺帽種類繁多、體積又小，傳統的人工篩選費時，且無法達到高效的分類。

因此，本計畫透過人工智慧與製造業中的元件結合成識別系統，利用計算機視覺與深度學習，對各種回收的螺絲螺帽進行自動化識別與分類。通過高精度的影像處理技術和 AI 訓練模型，從而實現對螺絲螺帽的分類、檢測與追蹤。

透過此系統能夠分類回收螺絲與螺帽，不僅能大幅提升生產線的工作效率，降低人力成本，還能有效減少錯誤率。隨著 AI 技術的不斷進步，未來分類識別系統將能夠應對更為複雜的需求，為企業帶來更大的效益。

關鍵字：AI 辨識、分類、回收、電子材料、深度學習

1. 前言

在現代製造業中，螺絲螺帽作為基本的機械連接件，廣泛應用於各行各業，如機械、電子、汽車等領域。隨著產品設計的多樣化，螺絲螺帽的類型和規格日益增多，這使得其在生產過程中的識別與分類變得越來越複雜。傳統上，這些操作依賴人工進行，不僅工作強度大，甚至會造成視覺疲勞且容易出現錯誤，難以達到高效且準確的分類要求。

為了解決這一問題，許多企業開始將人工智慧技術引入生產過程中，特別是在質量檢測和物料分類等方面。AI 技術中的深度學習和計算機視覺技術，憑藉其優異的圖像識別能力，成為了提高生產效率、降低成本和提升質量。螺絲螺帽識別與分類系統的研發，正是基於此背景，旨在利用先進的 AI 技術，實現對各種螺絲螺帽的快速、準確、自動化識別。

本系統採用了深度學習中的卷積神經網絡(CNN)模型，並結合多種影像處理技術，對螺絲螺帽的形狀、螺紋以及表面特徵進行特徵提取和分類。通過大量實驗數據的訓練，系

統能夠在不同環境下穩定運行，實現對螺絲螺帽的高效識別。這不僅能提升生產線的運行效率，還能減少人為誤差，確保產品質量的一致性。

本文將介紹該系統的設計思路、技術實現過程以及未來的應用前景。希望透過這項技術的推廣，能夠為智能製造領域帶來更多創新，並促進製造業的進一步升級與發展。

2. 文獻探討

發展至今分類系統有許多各式各樣的種類，我們以家中的行業做延伸發想，想運用至生活當中，讓分類這一項技術能繼續與其他行業做結合。

2.1 物件偵測技術(YOLO, You Only Look Once)

YOLO 物件偵測，採用深度卷積神經網絡(Deep CNN)來同時執行影像辨識與物件定位，YOLO 擁有完善的神經系統及權重數據，為開發者省去了不少時間，只需將蒐集的圖片與相對應的標籤數據丟入即可進行訓練，

並依據訓練結果調整參數來讓模型變得更好，提高準確率。

這項技術分為 v1 到 v9，會因為參數大小而在細分各本版出來，隨著時間演進 YOLO 的版本推陳出新，每個版本都有新的突破，如 v1 將圖像劃分為網格並同時預測邊界框和類別概率但是對於檢測小物件有困難進化到 v3 可以更精準的鑑測不同大小的物件，一直到最新的 v9，提升了在物體檢測速度和精度，雖然沒辦法細講，但是上述的這些就可以知道這項技術不斷地演進，在人工智慧方面幫助不少。

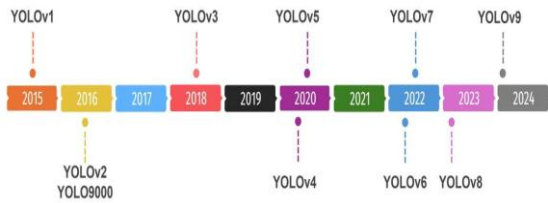


圖 1 YOLO 時間軸(取自文獻四)

2.1.1 YOLO v9 架構圖及說明

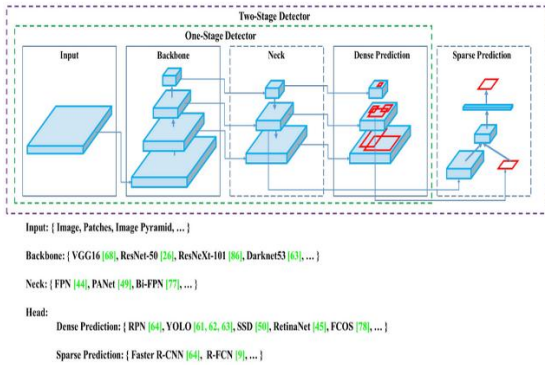


圖 2 YOLO 架構圖(取自文獻一)

第一層 Input Layer：指照片的輸入。

第二層 Backbone Layer：用來進行圖像特徵(Feature)萃取，形成 Feature Map。

第三層 Neck Layer：將上層所獲取的特徵進行融合，進一步增強特徵表現能力。

第四層 Head Layer：依據上層的特徵進行預測，輸出判斷結果及根據物件所處的位置框出界線。

本計畫所採用的 YOLO v9 與以往的版本相較之下，在精度、速度、效率以及泛化能力

等指標上都有不錯的進步。

2.1.2 YOLO v9 重要參數說明

Model	Test Size	APval	AP50 val	AP75 val	Param.	FLOPs
YOLO v9-T	640	38.3%	53.1%	41.3%	2.0M	7.7G
YOLO v9-S	640	46.8%	63.4%	50.7%	7.1M	26.4G
YOLO v9-M	640	51.4%	68.1%	56.1%	20.0M	76.3G
YOLO v9-C	640	53.0%	70.2%	57.8%	25.3M	102.1G
YOLO v9-E	640	55.6%	72.8%	60.6%	57.3M	189.0G

表 1 YOLO v9 各類別之差異重要參數說明(取自文獻五)

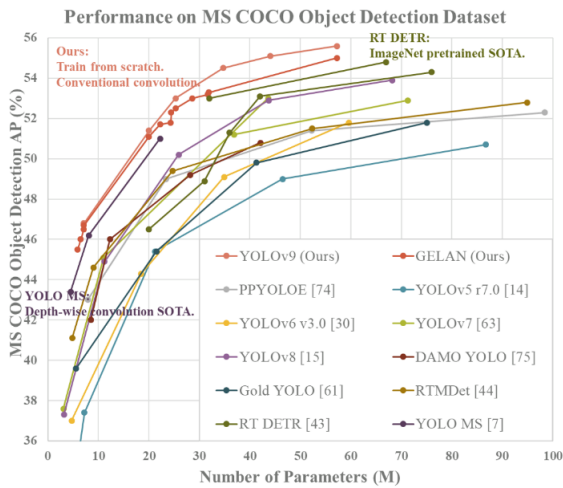


圖 3 YOLO v9 參數量與 AP 比較圖(取自文獻六)

Param.(M)參數量，全名為 Parameters，單位為 MB，模型所需之總參數量大小，其影響使用者在訓練時，須具備足夠的記憶體大小。

FLOPs(G)浮點運算次數，全名為 Floating Point Operations，單位為 GB，用來表示浮點運算數量，值越大，代表所需的浮點運算量就越大，因此，在不同硬體環境下，會有不一樣的完成時間。

IoU 交並比，全名為 Intersection over Union，又稱卡爾指數(Jaccard index)，用來衡量預測框(Predicted Box)與實際框(Ground Truth Box)之間的重疊程度，其值介於 0~1。

若指標為 0.5 IoU，當預測結果的 $\text{IoU} > 0.5$ ，則表示會以框線呈現，但預測不一定對的。

$$\text{IoU} = \frac{A \cap B}{A \cup B} \quad (\text{式 } 1)$$

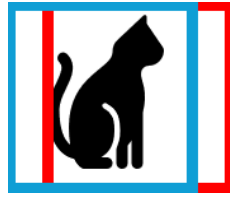


圖 4 $\text{IoU} > 0.5$ (預測成功，自行繪製)

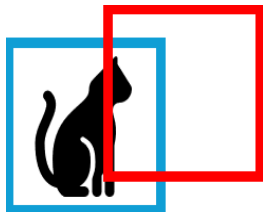


圖 5 $\text{IoU} < 0.5$ (預測失敗，自行繪製)

2.1.3 混淆矩陣與相關公式

透過預測結果與實際結果之間所構成的矩陣，稱為混淆矩陣(Confusion Matrix)有四種不同的指標，分別為 TP、FP、FN、TN，敘述如下：

		預測值	
		Negative	Positive
實際值	Negative	TN	FP
	Positive	FN	TP

表 2 混淆矩陣(取自文獻七)

TP(True Negative): 表示模型預測值與實際值相同，實際值為 Positive，預測值為 Positive，預測正確。

FP(False Positive): 表示模型預測值與實際值不同，實際值為 Negative，預測值為 Positive，預測錯誤。

TN(True Negative): 表示模型預測值與實際值相同，實際值為 Negative，預測值為 Negative，預測正確。

FN(False Negative): 表示模型預測值與實際值不同，實際值為 Positive，預測值為 Negative，預測錯誤。

精確度(Precision): 用來表示模型有預測為某物體，且預測也對，從這指標，我們可以看出模型對該物體的認知程度，是否謊報。以物件 a 的精確率而言其公式如下：

$$P(a) = \frac{TP(a)}{TP(a) + FP(a)} \quad (\text{式 } 2)$$

召回率(Recall): 用來表示模型對於在該物件總數量中，識別出了幾個該物件，從這指標我們可以看出模型是否有漏報。以物件 a 的召回率而言其公式如下：

$$P(a) = \frac{TP(a)}{TP(a) + FN(a)} \quad (\text{式 } 3)$$

針對上述兩點進行舉例，以 100 張貓及 100 張狗為例，模型在 200 張照片的所有答案中(包含回答為貓、回答為狗、不回答)，在回答出貓有 30 次，且正確有 25 張，把貓認為狗有 5 張，可得出該模型對於貓的精確率及召回率為：

$$P(\text{cat}) = \frac{25}{25+5} \cong 0.83 \quad (\text{式 } 4)$$

$$P(\text{cat}) = \frac{25}{25+75} \cong 0.25 \quad (\text{式 } 5)$$

雖然看似精確度很高，模型在回答為貓的謊報率低，但還有 75 張為貓的照片沒有被識別出來，從這指標來看我們可以思考，是否是模型在訓練中可能給予學習的資料不夠好？還是學習的不夠多元？只認為橘貓是貓？灰貓不是貓？

平均精度(AP, Average Precision): 指物件偵測中的對於某物件的準確率指標，其值範圍在 0%~100%，越高越好，反映模型在某物件檢測上的性能指標，其值為 P 為 Y 軸，R 為 X 軸所繪製的曲線圖，在此圖中的曲線下所形成的面積。如表一中的 AP50 為例，意思為 IoU 為 50 時的 AP 值。公式如下：

$$AP = \int_0^1 P(R) dR \quad (\text{式 } 6)$$

平均精度均值 (mAP, Mean Average Precision): 將每個物件的 AP 值在加以平均，為整體平均率，公式如下(c 為物件數, i 為第幾個物件):

$$mAP = \frac{1}{c} \sum_{i=1}^c AP_i \quad (\text{式 7})$$

2.2 Python

Python 是一個跨平台的直譯式語言，擁有豐富的函式庫、物件導向等特色，像 Youtube、Instagram 等網路服務都採用此程式語言，在本計畫中的運作是以 Python 為大腦，結合 Arduino 來實現。

雖然我們是直接使用 YOLO 這項技術，但是在背後的深度學習或者是卷積神經網路都是利用 Python 語言去撰寫，包括調整其中的參數都是可以依照我們的需求去進行調整、測試，得到最佳結果。

2.3 Arduino Uno R3 與步進馬達

Arduino 是一個開源嵌入式硬體平台，用來供使用者製作可互動式的嵌入式專案，可用於各種應用之中，體積雖小卻可連接多種元件。本計畫結合步進馬達，在模型中端擔任過慮零件及尾端擔任分類的重要工作。本計畫畫有用到步進馬達結合 ULN2003AN 驅動版藉由 Arduino 讓馬達可以運作。



圖 6 步進馬達&ULN2003AN
(取自文獻十)



圖 7 Arduino Uno R3
(取自文獻十一)



圖 8 Arduino Mega 2560
(取自文獻十二)

2.4 Kaggle 平台(<https://www.kaggle.com>)

Kaggle 是一個與機器學習與資料科學有關社群平台，企業可以發布競賽，設定問題目標並提供資料向大眾徵求問題解方等等。

在本計畫中，我們進行螺絲與螺帽的分辨，需要收集大量的照片，透過 Google 圖片搜尋需要好費大量時間進行下載，為了節省時間可至可平台一件下載所需的照片，以便更快速結合接下來 2.5 的工具來完成資料集的彙整，送入 YOLO 模型進行訓練。

以下簡單說明若想收集螺絲與螺帽的文字，以便搜尋。

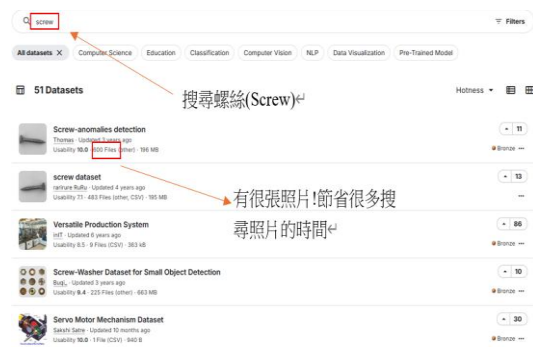


圖 9 Kaggle 平台，搜尋所需的照片集範例(取自文獻十三，實際操作)

2.5 電腦視覺標註工具(<https://www.cvat.ai>)

電腦視覺標註工具 (computer vision annotation tool, CVAT) 是一種開源的計算機標註工具，提供方便的介面，用於創建標註數據集、邊界框等功能，應用在各種領域，用來幫助模型進行圖像學習。下圖為訓練資料的其中一張，紫色框為螺絲，紅色框為螺帽。

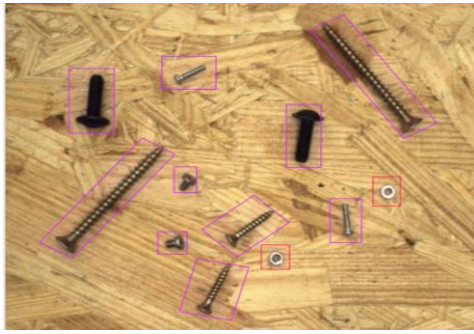


圖 10 標注示意圖
(取自文獻十四，實際操作)

2.6 CUDA & cuDNN

CUDA (Compute Unified Device Architecture) 與 cuDNN(CUDA Deep Neural Network library)是Nvidia顯卡所獨有的功能，如果顯卡是其它品牌，基本上就無法安裝這個功能，在進行訓練中，以GPU為主，相比之大幅減少縮需的時間。

CUDA 是一個並行計算平台和應用程式程式介面(API)，使得開發者可以更有效地利用 GPU 的大規模並行處理能力。

cuDNN 用於深度神經網絡(DNN)的GPU加速庫，在CUDA的基礎上，提供了高度優化的實現，例如對卷積、池化、標準化等操作進行優化。

2.7 扮演運輸零件的 Dobot Conveyor Belt 結合 Dobot Studio 開發環境 (<https://www.dobot-robots.com>)

Dobot 是一家開發機械手臂的公司，尤其是針對教育、工業和研究領域的機械手臂，適合用於自動化、3D 打印、激光雕刻、物料搬運等應用，並且可以通過簡單的編程進行操作，此公司也在商業、醫療和教育領域發揮了重要作用。

本計畫利用 Dobot 公司的產品，利用機械手臂以及運輸帶，在計畫中我們利用此手臂的可動性(上下左右及關節的方向變化)，來與鏡頭相結合，調整鏡頭的最佳位置，以便在進行觀測時，可以提供較清晰的照片，幫助模型進行預測。

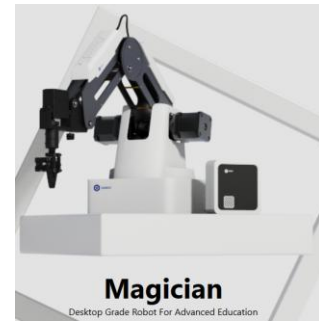


圖 11 本計畫所使用的機器手臂
(取自公司網站)



圖 12 本計畫所使用的傳輸帶
(取自公司網站)



圖 13 Dobot Studio API
(取自公司網站)

3. 研究方法

利用 AI 系統辨識螺絲和螺帽再進行分類，依賴計算機視覺與機器學習技術對圖像中的物體進行分類與識別，先透過樂高零件所組成的零件輸入端，中途經過 Dobot 公司的運輸帶，其中螺絲與螺帽會遇到兩關由 Arduino 所控制的步進馬達，層層被調整所處位置，以便一個個出來，之後好讓末段的攝像頭可以從中提取螺絲和螺帽的特徵，如形狀、大小，而不受並排或彼此距離過於相近，導致分類機沒有足夠的時間進行分類。在訓練階段，AI 系統需要大量的標注資料來學習如何區分不同類型的螺絲與螺帽。通過反向傳播與優化等演算法，模型能夠不斷提升分類的

準確性。經過訓練後，該 AI 系統能夠實現自動分類，以下整理完分為三個部分。

3.1 方法流程敘述

該系統流程分為四大步驟，分別為搭建 YOLO 模型、操作 Dobot Magician & Dobot Mini Conveyor Belt、搭載樂高模型、過濾器、分類機、統整程式碼。

3.1.1 搭建 YOLO 模型

- (1) 從Kaggle中蒐集圖像資料。
- (2) 將資料區分訓練、驗證、測試資料集。
- (3) 利用電腦視覺標註工具(CVAT)，分別框出螺絲、螺帽，並輸出對應座標與標籤名稱(數字)。
- (4) 下載YOLO v9模型，根據自身硬體設備，選擇適合的類別。
- (5) 調整一些重要的參數(如照片集路徑為何？要訓練幾次？)。
- (6) 反覆進行訓練，並根據訓練結果的好壞，適時調整參數。

3.1.2 控制 Dobot Magician & Dobot Mini Conveyor Belt 及 Uno R3、Mega 2560

- (1) 利用程式碼控制機械手臂的起始座標位置，使得搭配的鏡頭能夠有好的辨識距離。
- (2) 利用程式碼控制傳輸帶的運輸速度，依據零件的傳送過程，視情況找尋一個適合的運輸速度。
- (3) 分別撰寫 Uno R3 及 Mega 2560 控制過濾器、Mega 2560 控制分類機。

3.1.3 搭載樂高模型、過濾器、分類機

- (1) 透過學校的智慧機器人研究室的樂高零件，來搭建待識別的入口輸入端。
- (2) 以運輸機為基底，搭配樂高零件及步進馬達，來實現零件過濾功能，使得待識別零件不會過於壅擠，無法順利判斷。
- (3) 在運輸機末端，同樣以樂高零件及步進馬達，構建一個分類機，讓每個零件可以跑到自己的盒子中。

3.1.4 統整程式碼說明

(1) 將機械手臂、運輸機、YOLO等相關程式碼，統整在一個Python程式檔案中，並且再之中添加額外程式碼使得辨識結果可以傳送至以Arduino控制的分類機中，達成單向溝通。以一個程式檔案就能讓所有部件全部動起來的效果，最終達到以Python為大腦實現以AI構建自動化分揀與利用系統。

3.2 流程圖

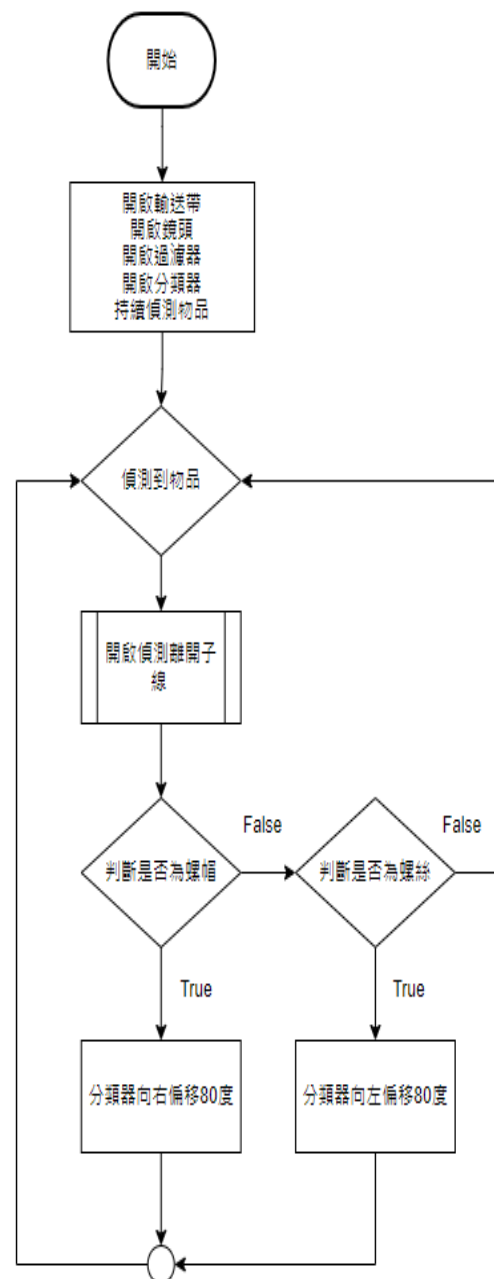


圖 14 主流程

(以 Draw.io 網頁圖表平台自行繪製)

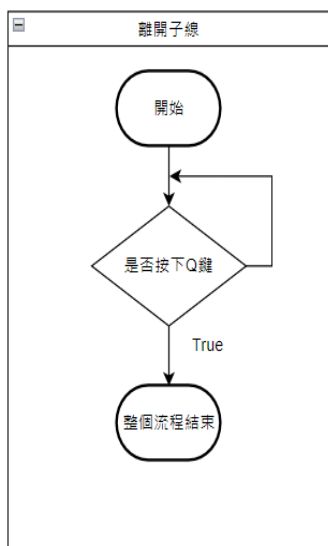


圖 15 離開子線
(以 Draw.io 網頁圖表平台自行繪製)

3.3 示意圖

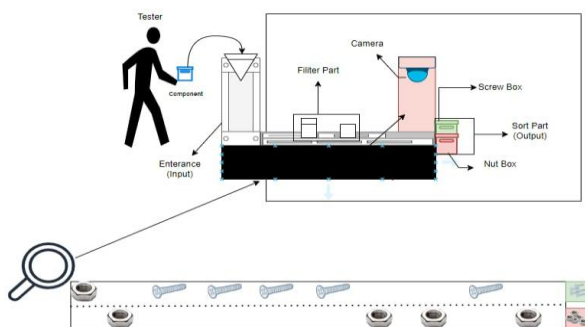


圖 16 流程圖像化(以 Draw.io 網頁圖表平台自行繪製)

4. 結論

在製作樂高模型的輸入端時，遇到很多困難點，例如從上面往下放零件的通道斜度不夠，或是放零件時會因為軌道的寬度而影響到軌道上前進時零件卡住等等的這些問題，隨著每次的測試，我們認為無論怎麼調整都沒有用，畢竟螺絲與螺帽的大小不同，要找到一個合適的寬度是不可能的，後來我們想要以外力的方式去解決，好讓零件可以比較順利的前進，於是就利用 Arduino 控制步進馬達，達到過濾的功能，透過馬達上安裝一個長條的樂高零件，可以使零件的「位置」不斷改變，讓卡住、並排的機率降低，結果已經是令

人相當滿意了。在模型的訓練上是相當順利，能夠在精確度與速度上取得一個不錯的平衡，在一開始因為鏡頭的對焦問題，讓模型看到零件，沒有做出任何反應，辨識正確度只有將近四至五成，之後我們不斷的調整手臂高度，保證鏡頭的對焦問題不再發生，最終辨識度高達九成。模型預判之後需要搭配分類機，來讓零件被倒入到對應的盒子中。我們需要抓準零件掉落與分類機改變方向的速度及角度，在這三個因素中取得對應的平衡，不然會發生零件正在掉落，或已經掉落的時候分類機還沒有把方向調整好，或是受到後面的零件的影響，導致分類機最後變成以後者為主的方向，讓零件跑到錯誤的盒子中，在此環節中我們也花了不少時間在做調整與測試。從零件的輸入到經過運輸帶的傳遞之至模型辨識，最後到分類機的整體效率達八至九成。

在此系統中還有些不足的地方，我們覺得可以更好，如增加額外的「震動感測器」，來偵測零件是否丟入入口區，才開啟鏡頭及軌道等配件，達到省電的效果。

在這個計畫中，除了利用在課堂上的專業技術與知識外，我們對未來的展望是除了針對這些基本的機械元件外，還能擴大至對金屬的回收再利用，例如提去廢棄手機中的黃金等，或是檢測是否有有毒物質的存在。

此外，在本計畫中最重要的是如何與隊友溝通，彼此間如何一起解決問題，團隊精神的強大，才有辦法解決各種挑戰、難關。

5. 成品圖

下列圖為本計畫的成品照片。

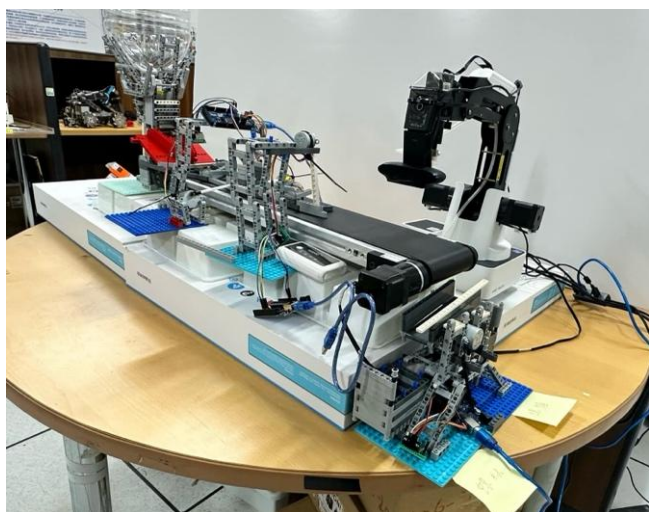


圖 17 自行拍攝

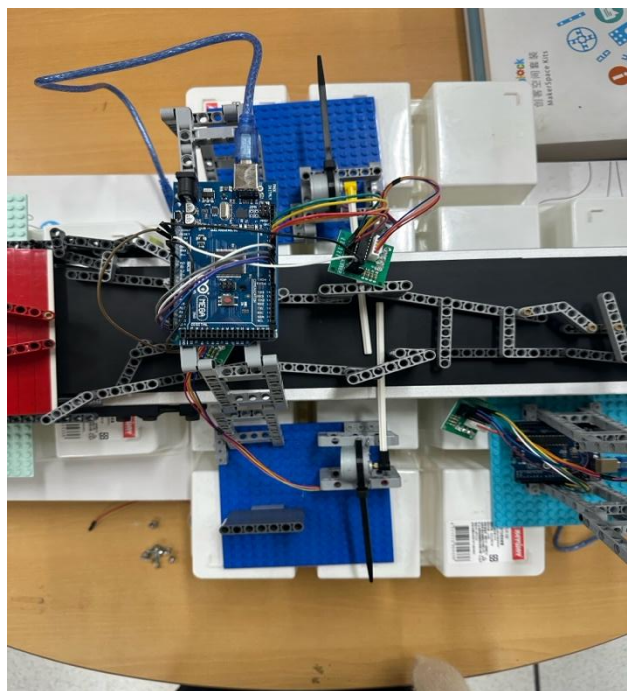


圖 20 自行拍攝

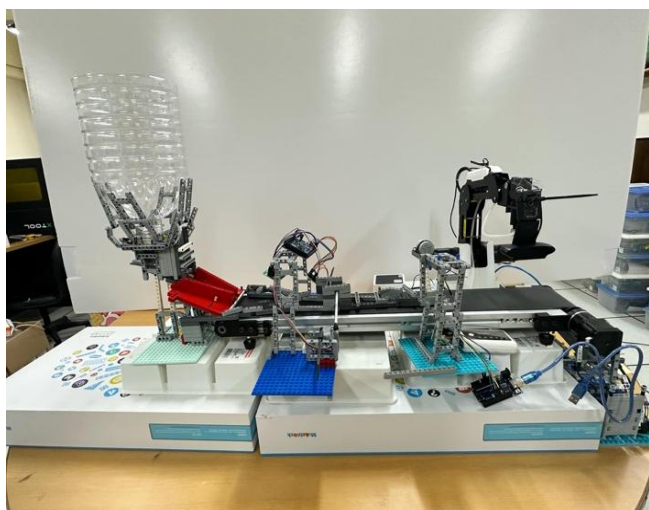


圖 18 自行拍攝

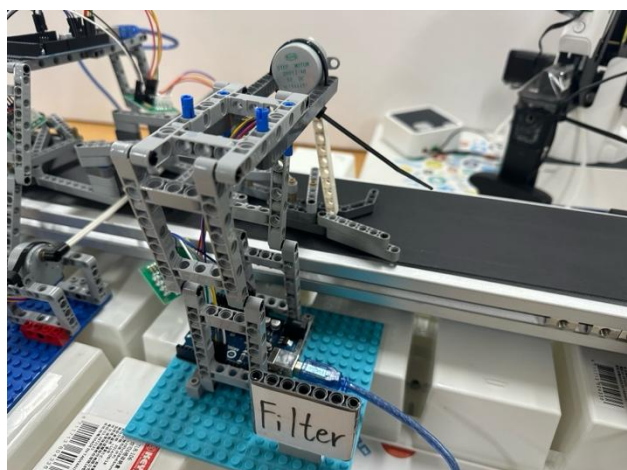


圖 21 自行拍攝

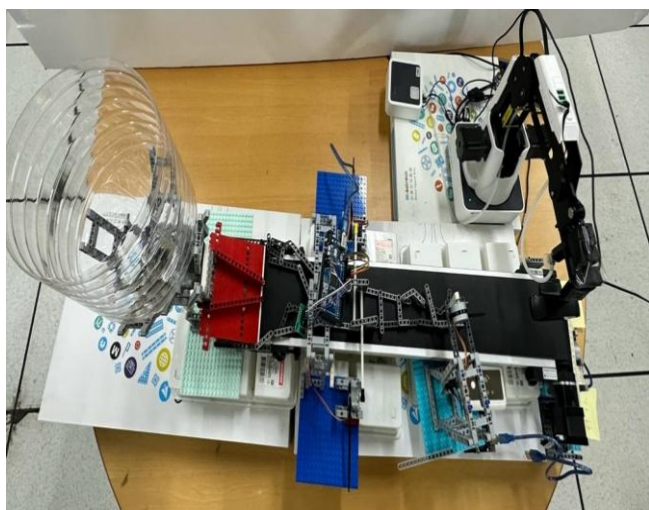


圖 19 自行拍攝

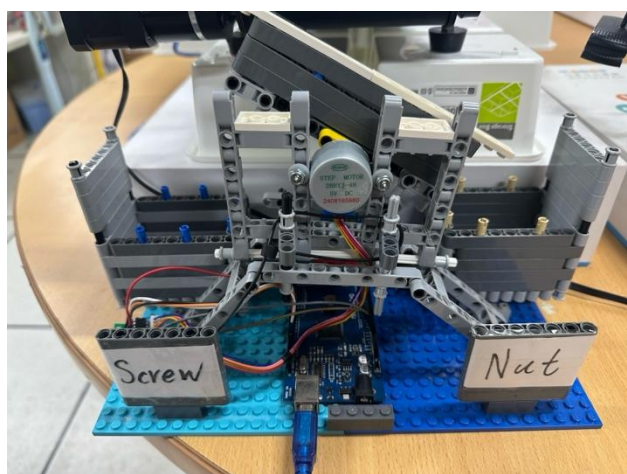


圖 22 自行拍攝

6. 參考文獻

1. Steven Meng(2021), YOLO v4 詳細解讀
2. Xiang(2024), YOLO 的版本迭代介紹之
3. AI4kids(2023), YOLO 是什麼? 3 分鐘了解 YOLO 的演進, 可以應用在生活中哪些地方!
4. Guillaume Demarcq(2023), What is YOLO? An In-Depth Introduction to Object Detection in Computer Vision.
5. WongKinYiu(2024), YOLO9 Introduction.
6. Chien-Yao Wang et al. (2024). YOLOv9: *Learning What You Want to Learn Using Programmable Gradient Information*, 1-2. *Computer Vision and Pattern Recognition*, arXiv: 2402.13616.
7. yehchitsai(2022), YOLO 相關概念說明
8. seanh737(2018), 初識 Python
9. 李心華(2023), Arduino 簡介
10. 米羅(Unknown), 【Arduino 入門教學課程】 第二十篇: 28BYJ48 步進馬達+ULN2003 驅動板, 輕鬆上手! Basic Tutorial for Arduino, 入門篇
11. Jason Chu(2020), Arduino 入門第一篇: 認識 Arduino
12. 曹永忠等人(2022), Arduino 程式教學 (基本語法篇): Arduino Programming (Language & Syntax), 渥瑪數位有限公司, page 256, 2016.8.12, ISBN: 9789865629410
13. syssherry(2022), 累積機器學習實作經驗的好地方—Kaggle 平台介紹
14. Yi Lung Chiu(2023), 安裝影像標註軟體 CVAT
15. Hugo Reyes(2024), CPU vs GPU for Model Training: Understanding the Differences.
16. Colombo Chao(2023), Dobot Studio 下載與安裝
17. SERLatBTH (2019), Dobot Magician with Python - Starter Guide
18. 烏龜 PHOTO(2024), 超硬核教學 YOLO v5 自拍程式實作
19. nighting(2024), Python 環境: CUDA,cuDNN 安裝