國立勤益科技大學工業工程與管理系
113 學年度實務專題競賽
基於視覺辨識之自動倉儲系統建置

摘要

隨著科技進步與人工智慧的問世，機器自動化結合 AI 是重要的一環。現今自動化系統不只應用於製造業，進一步應用於其他的產業，如物流、醫療、倉儲、服務業等眾多領域之中。本研究基於視覺辨識之倉儲系統中，透過視覺辨識與機械手臂的結合，建立一套具備物件識別與自動分揀功能的智慧自動倉儲系統，透過 YOLOv8 物體捕捉，加上 KNN 和 RGB 轉 HSI 色彩空間演算法輔助辨識，並結合六軸機械手臂，使得能自動辨別物品並完成自動分類作業。

本研究內容涵蓋視覺模組建構、演算法訓練與評估、機械手臂路徑設計及整體倉儲模擬場域建置。實驗結果顯示，系統能有效提升分揀準確率與作業效率，未來可望導入更多顏色與形狀識別項目，並且可放置多鏡頭建構 3D 空間與複雜的路徑程式撰寫，以實現多貨物堆積的場景。

關鍵字：自動倉儲系統、機械手臂、視覺辨識、YOLOv8、RGB/HIS 演算法、KNN 演算法

第一章 緒論

1.1 研究背景動機

隨著科技的快速發展，自動化在許多產業中成為重要的一環，物流產業隨之轉型，傳統的倉儲作業過度依賴大量人力，從進出貨、分揀到盤點皆須人工操作，人力作業存在風險的問題，導致錯誤率偏高，以及無法即時盤點庫存，使企業難以立即調整庫存配置，影響物流效率與市場應變能力，進一步影響企業利益。因此，傳統全人工倉儲作業模式逐漸被市場淘汰，傳統倉儲低效率與高錯誤率難以滿足產業標準，因此企業開始導入自動化設備以提升運作效率並降低人為錯誤。

在此背景下，人工智慧技術的發展為倉儲管理帶來革命性突破，透過視覺辨識技術、RFID 感測器與自動化設備的結合，使倉儲系統能夠準確識別物品、降低存貨成本、減少錯誤、提升分揀及盤點。

目前現有文獻專注於新建自動化倉儲系統的應用，而對於如何在設備上導入人工智慧技術來提升倉儲智慧化程度仍然有限。因此本研究專注於提升智慧化自動倉儲的發展，探討如何在不汰換舊設備的情況下，實現設備智慧化，以彌補傳統倉儲模式的缺陷，幫企業提升效率又能降低成本。

1.2 研究方法

通過 YOLOv8 與演算法結合，用於判斷物體顏色的視覺辨識模型，並且給予一個指定值用於判斷存放倉位位置，使機械手臂執行不同路徑將物體進行存放，以下為自動化倉儲系統建置步驟說明。

1. 確立系統架構：選擇適合的影像處理模組、機械手臂控制模組和倉儲管理模組的系統架構。
2. 訓練視覺辨識模型：採用深度學習方法訓練 YOLOv8 進行物品檢測與分類，並對比不同演算法的性能與準確性；選擇合適的深度學習框架 PyTorch 進行模型訓練，並通過參數調整以提升辨識準確度。
3. 撰寫機械手臂路徑：基於物品位置與倉儲空間，設定點紀錄與撰寫路徑程式，實現無人化倉儲應用。
4. 架設模擬場域：設計一套智慧化倉儲管理系統，對比不同演算法對於系統性能的影響，選擇最合適之配置方案。分析系統在運行過程中的問題，進一步優化辨識模組系統與硬體設計。
5. 實現物品的自動分類儲存，倉儲系統整合與測試實際執行設計之倉儲系統，並檢驗物品辨識準確率，確保系統達到設計之效能。

第二章 文獻探討

2.1 視覺辨識

視覺辨識是將影像輸入至分析儀器或是軟體中進行影像分析，結合機器學習其應用範圍相當廣泛，包含醫療領域、安全防護、工業製造等以下將

介紹各領域的案例介紹。醫療領域應用如將皮膚給機器去做辨識，並以其特徵與狀態辨別出是哪種的皮膚病；安全防護應用如中國的天眼系統，可以清楚的有多少人經過了哪些地方；工業製造應用如利用視覺辨識系統，辨別貨物是否有缺陷或瑕疵從而達到品質管理的目的。

2.1.1YOLO

許瀟傑等人(2024)針對 YOLOv5 與 YOLOv8 模型架構進行比較，並分析其核心模組 C3 與 C2f 的主要差異，以條列式說明兩者的架構特性與模組比較。

1. YOLOv5 由 Ultralytics 於 2020 年底開發，目標是提供輕量、高效且高精度的即時目標檢測模型。其架構包含四大部分：
   (1) Input（輸入層）:
   (a) 使用 Mosaic 增強、動態邊界框與自動縮放。
   (b)提升模型對多樣態場景的學習能力，以避免過度擬合。
   (2) Backbone（主幹網路）：主要作為提取影像特徵，此層包含 Focus 模組、ConvBNSiLU、C3 模組，以下說明各模組功能。
   (a) Focus 模組：切圖後重組以保留空間資訊。
   (b) ConvBNSiLU：結合卷積、正規化與 SiLU 激活。
   (c) C3 模組：基於 CSP 架構，強化特徵融合。
   (d) SPPF 模組：聚合多尺度特徵。
   (3) Neck（特徵融合層）:
   (a) Upsample 與 Concat 融合不同階段特徵。
   (b) 強化多尺度物件檢測能力。
   (4) Head（輸出層）:
   (a) 執行分類、邊界框回歸與置信度預測。
   (b) 採用耦合式設計，同時處理多任務。
2. YOLOv8 由 Ultralytics 於 2023 年推出，在延續 YOLOv5 架構的基礎上，針對準確率與效率進行多項優化，依然包含四大部分：
   (1) Input（輸入層）:與 YOLOv5 類似，進行資料增強與預處理。
   (2) Backbone（主幹網路）:
   (a) 改用 3×3 卷積取代 6×6，降低計算成本。
   (b) C3 模組全面替換為 C2f 模組（源自 ELAN 架構），提供更強梯度流與靈活性，提升訓練效率與準確率。
   (3) Neck（特徵融合層）:
   (a) 精簡融合方式，移除部分 1×1 卷積。
   (b) 直接拼接後上採樣，降低參數與延遲。
   (4) Head（輸出層）:
   (a) 採用解耦式設計，分類與回歸分開處理。
   (b) 提升模型收斂速度與預測精度。

根據上而言，儘管 C2f 在設計上需更多模組，但整體運行時間更短，展現其架構優化效果，因此本研究選擇 YOLOv8 作為訓練模型。

表 1 C3 與 C2f 模組差異分析表

比較項目	YOLOv5-C3	YOLOv8-C2f
設計理念	基於 CSP（Cross Stage Partial）架構	結合 C3 與 ELAN 模組概念，強化梯度資訊流
結構複雜度	相對較高，包含 Bottleneck 重複結構	結構簡潔，支援更多跳層與彈性配置
特徵表達能力	穩定，適合中等深度網路	更豐富，特別適合深層與輕量模型
運算效率	較高資源消耗，運行時間較長	較低資源消耗，訓練與推理更快速
模型應用	YOLOv5 Backbone 與 Neck 的核心模組	YOLOv8 全面取代 C3 的主力模組

2.1.2RGB 演算法

RGB 空間主要是由红(Red)、绿(Green)、蓝(Blue)三原色来表示,他們分別在三維空間中分別各代表一個軸。RGB 模型可用笛卡爾坐標系統來表示，如圖 1 所示。

由於 RGB 在不同環境下隨著亮度變化導致色彩空間分布不均勻，所以兩種不同顏色在色彩空間中的色彩差距，無法用來表示在這個顏色空間中的

距離，因此轉換成 HSI 顏色空間以增加考慮亮度因素；其空間模型如圖 2。

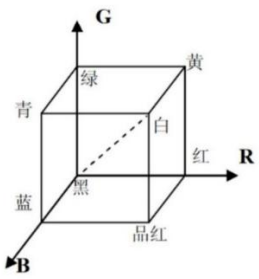


圖 1 RGB 顏色模型圖

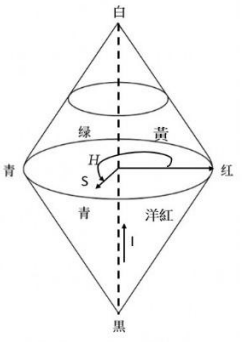


圖 2 HSI 顏色模型圖

HSI 模型是由 Munsell 提出来的，適合人類視覺特性的色彩模型，其中 H(色度)表示不同的顏色、S(飽和度)表示顏色的深淺、I(亮度)表示顏色的明暗程度；HSI 模型通過色度、飽和度和亮度這三個維度提供了一種更加符合人類視覺感知的顏色表示方法，所以相較於 RGB 來說他更易於進行圖像處理中的顏色調整、分割和增強等操作，從 RGB 到 HSI 轉換的公式如下。

$$\mathrm{H} = \begin{cases} \theta, \mathrm{G} \geq \mathrm{B} \\ 2\pi - \theta, G < \mathrm{B} \end{cases} \text{當} \theta$$

$$= \cos^{-1}\left(\frac{(R-G)+(R-B)}{2\sqrt{(R-G)^2+(R-B)(G-B)}}\right)$$

$$\mathrm{I} = \frac{R+G+B}{3}$$

$$\mathrm{S} = 1 - \frac{\min(R,G,B)}{I}$$

2.1.3 分類分群演算法

Thomas(1967) 提出 K- 鄰近法 (K-Nearest Neighbor, KNN)，為機器學習中的監督式學習的一種算法，這個演算法的核心思想是「物以類聚」，透過計算待測樣本與訓練樣本之間的歐幾里得距離(euclidean distance)後根據 K 個最接近者進行分類判斷；KNN 是一種無母數統計方法，分類標準由「多數決」判定，輸入包含特徵空間中的 K 個最接近的訓練樣本，以一個分類類別作輸出，KNN 優點在於機器學習中算是最容易理解的演算法之一，計算方式較易懂，資料的變數型態不受限制，較不容易受到異常值之影響，但 KNN 缺點在於需要計算待測樣本與所有樣本的距離，計算量較大，且需要占用大量記憶體，因此 KNN 適合用於處理資料較少的情況。

2.1.4 開源計算機視覺函式庫

開源計算機視覺函式庫 (Open Source Computer Vision Library, OpenCV)由 Intel 公司所發起並開源，在學術以及商業界都廣泛的應用。OpenCV 支援多種語言，包括 Python、C++、Java、Ruby、MATLAB 等，並且可以在多個作業系統上運行，如 Windows、Linux 和 macOS；而應用範圍相當廣泛，如人臉辨識、邊緣檢測、物體追蹤、自駕車道路監測等。

OpenCV 提供基本影像操作，應用包含濾波、邊緣檢測、直方圖均衡化、物件檢測、影像分割、特徵提取，以及機器學習演算法，如支援向量機（SVM）、K-means 分群等；OpenCV 對於深度學習模型加載，可以直接使用訓練好的模型進行推理，在影像處理中扮演著核心的腳色，雖然基礎的 GUI 功能簡易，但結合其他框架就能做出很完整的應用程式，在商業開發或是學術研究都是可靠的選擇。以下將以三個步驟介紹 OpenCV 如何運作：

1. 讀取圖片：將圖片庫中的圖片導入程式。
2. 處理影像：將導入的圖片轉換成灰階，並識別圖片中有哪些圖樣。
3. 輸出結果：將識別出的圖樣顯示出來，如人臉、車子等。

2.1.5 PyTorch

PyTorch 因其能夠快速上手的 Python 語法，並

支持動態計算圖，讓開發者能靈活修改計算流程，並且具有強大的調適能力，可直接使用 Python 原生工具，降低學習成本。此外，PyTorch 在研究領域影響力巨大，並在學術界和開發者廣泛採用，使其成為深度學習領域的重要選擇。

2019 年時機器學習框架主要分為 TensorFlow 和 PyTorch，其中 PyTorch 從 2017 年在學術界迅速崛起，成為 TensorFlow 的主要競爭對手。相較於多數學習框架，PyTorch 在程式撰寫上更加靈活，並提供更好的使用體驗。其前身 Torch 是基於 Lua 語言的張量計算函數庫，而 Lua 語言以高速運行著稱，被譽為世界上最快的手稿語言；然而 Torch 團隊發現 Python 的使用者數量遠超 Lua，因此決定基於Python 重建並優化函數庫，最終推出 PyTorch。

PyTorch 具備兩大核心特點，一為支援 NumPy 的張量運算，其中 NumPy 作為 Pyhton 的擴充函數庫，主要用於高階數學運算，尤其是在大規模多維陣列與矩陣處理方面，進一步增強 PyTorch 的運算效能與靈活性，並可利用 GPU 或 MPS 等硬體加速運算；二為內建自動微分機制，使其特別適用於深度神經網路的訓練。

TensorFlow 由 Google Brain 團隊於 2015 年推出的開源機器學習和深度學習框架，目前廣泛應用於研究與生產環境，其中主要用於建構和訓練神經網路，支持各種感知和語言理解任務，包含電腦視覺、自然語言處理(NLP)和強化學習等。

2.2 機械手臂

機械手臂(Robotic arm)是模仿人類手臂功能並完成各項作業的自動控制設備，此設備具有多個關節相連得以在平面或三度空間進行運動，或使用線性點位移動。由機械主體、控制器、主機板、伺服馬達和感測器所組成，並撰寫路徑程式根據作業的需求完成指定動作。機械手臂技術之應用，範圍最廣泛的是用於工業製造上，如各種機器之組裝、噴漆、焊接、高溫鑄造等繁重工作皆能以機械手臂取代人工作業。除了主要用於工業製造上，商業農業、醫療救援、娛樂服務、軍事保全甚至在太空探索等領域，以下本研究將介紹機械手臂之構造、原理種類與手臂種類。

1. 機械手臂之架構可分為以下四部份：
   (1) 機構部分：
      (a) 基座：用來固定機械手臂。
      (b) 關節：可旋轉或滑動的部位，決定手臂的距離與活動範圍。
      (c) 連桿：連接關節的部分。
      (d) 末端處理器：機械手臂最前端的裝置。

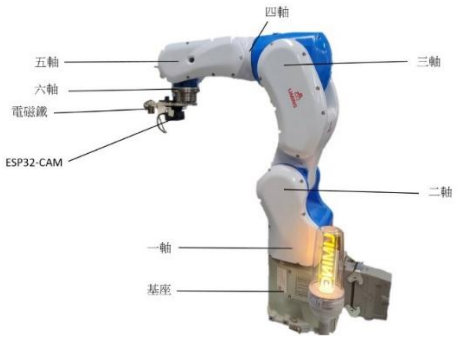


圖 3 六軸機械手臂模型圖

   (2) 驅動部分：
      (a) 伺服馬達：伺服馬達取得各軸之點位紀錄後，最終由伺服馬達去驅動減速機使其機械手臂至各軸點位。
      (b) 減速機：用來降低轉速達到精確控制、提高扭矩使機械手臂可以搬取更重的物體、降低電機的轉動誤差，使手臂在微小角度的調整提高控制精度。
   (3) 感測部分：
      (a) 力矩感測器：確保安全施力，防止損壞工件。
      (b) 視覺感測器：用於物體識別與導航。
   (4) 控制部分：
      (a) PLC：工業機器人常用的控制系統。
      (b) 嵌入式電腦：適合 AI 視覺與機器學習應用。
      (c) 運動控制技術：確保機械手臂能準確移動到目標位置。
2. 機械手臂的原理分類，以下為原理種類與分類表如表 2。

(1) 門型手(直角坐標型)：移動方向都與前手臂方向成 90 度，主要由 2 至 3 軸的馬達所組成，運用負荷較輕的工件重量，常用於塑膠射出產業居多，適用於直線運動的自動化生產，如：搬運、上下物料等。

(2) SCARA(圓柱座標型)：手臂在 X、Y 方向具有靈活、速度快的特性，特別適用於小空間，如：小型零件組裝工廠。

(3) 青蛙腿(極座標型)：是透過多軸的協同動作，實現準確的線性與旋轉運動，這種前後左右移動的線性運動和水平搬運移動，適合應用於面板廠、電視液晶螢幕、半導體、科技等自動化生產。

(4) 一般型(關節型座標)：以實現複雜運動，在不同的方向和平面上都能進行操作，具有多樣化的性能表現，可以進行拿取、搬運重物、換物料等作業，因此可應用的產業相當廣泛，如：半導體、加工、製造、食品、汽車、服務業等。

(5) Delta(蜘蛛型)：將多個機械手臂連接到同一個支撐的控制點，在平行移動上能實現快速移動的效果。

表 2 機械手臂原理分類

名稱 / 項目	門型手	SCARA	青蛙腿	一般型	蜘蛛型
座標系統	直角座標型	圓柱座標型	極座標型	關節型座標	關節連結型
活動範圍	直線	水平快速	平面內	適應多種情境	快速平行移動
操作特性	精確定位	適合高速且精確操作	高速操作	軌道能快速靈活運動	用於特殊的抓取任務
應用範圍	搬運	小型零件組裝	半導體	通用型	食品加工

3. 機械手臂種類：

(1) 四軸機械手臂：適合水平移動，如平面搬運、簡單堆疊等，其中應運用包含電子產品配送、物流揀貨、簡單的物料搬運。

(2) 五軸機械手臂：比四軸多一個自由度，能夠進行更靈活的運動，但仍無法實現全方位的旋轉，應用於包裝、食品包裝、簡單的焊接作業。

(3) 六軸機械手臂：因其目前最高的自由度，在一定的限制下能大部分模擬人類手臂運動，包含旋轉、俯仰、左右擺動等，靈活度高因此特別用於需要多角度操作的工作，如：焊接噴塗與裝配。

由於六軸機械手臂的高靈活度可完成更多不同角度的工作，減少機械臂重新定位的時間提高生產效率，此外他還具備可擴展性強的優勢，能夠適應多變的生產需求。以下為機械手臂種類之比較如表 3。

表 3 手臂種類之比較

手臂種類	四軸機械手臂	五軸機械手臂	六軸機械手臂
自由度（DOF）	4DOF	5DOF	6DOF
運動範圍	平面運動	增加一個旋轉自由度，靈活性提升	可在3D空間內自由移動
應用場景	搬運、分揀、包裝、簡單裝配	機械加工、簡單焊接	精密焊接、噴塗、複雜加工、電子裝配

2.8 自動化倉儲應用

倉儲管理在供應鏈中扮演著重要角色，對於企業營運至關重要；傳統倉儲主要依賴人力與基本機械設備進行物料儲存、揀貨與配送。隨著市場需求快速變化，與科技逐漸進步，傳統倉儲的缺點逐漸顯現，於是企業為了尋求更高效的倉儲解決方案。而導入自動倉儲(Automated Storage and Retrieval System, AS/RS)，透過自動化與智慧化技術，最佳

化貨物存取與管理流程，以下為案例之介紹。

京東作為中國領先的電商平台，其倉儲與物流效率對於提升整體運營能力至關重要。近年來隨著訂單的增長與市場需求變化快速，傳統倉儲產業已經難以滿足需求，因此京東選擇建設智慧化自動倉儲系統，並在多個物流中心導入機械手臂技術，藉此降低人力成本並提高生產效率。

京東的智慧自動化倉儲技術透過機械手臂、自動存取系統（AS/RS）、AGV 機器人、AI、大數據分析與 5G 物聯網等技術，極大提升物流運營的效率，這種高度自動化的倉儲模式，為全球物流行業提供了參考指標。

2.9 使用者資料包通訊協定

使用者資料包通訊協定(User Datagram Protocol, UDP)是一種是和時間敏感應用的通訊協定，如線上遊戲、影音串流撥放或網域名稱系統(DNS)查詢等。由於它在傳輸資料前不需要與接收端建立穩定連線，因此能夠加快傳輸速度。省略連線建立的過程，讓 UDP 更適合需要快速反應的應用場合。不過 UPD 也存在一些風險，如資料封包在傳輸過程中可能會遺失。而且，駭客也較容易利用 UPD 發動分散式阻斷服務(DDoS)攻擊，本研究之機械手臂與視覺辨識模型的相關指令與結果，將採用 UDP 進行通訊。

第三章 視覺辨識模型設計與訓練

3.1 環境建置

本研究設備需具備視覺辨識模組和機械手臂，模擬工廠環境和倉庫分類時，將在全開光、半開光兩種情況收集影像數據，主要模擬實際倉儲環境，並收集的影像數據進行視覺模組訓練，視覺辨識模組訓練完畢後，將於機械手臂結合，建置智慧自動化倉儲系統。

3.2 軟體及硬體選用

建置環境分為軟體與硬體兩個方面，軟體方面透過 PyTorch 建立深度學習框架以此加速視覺訓練進度，再用 YOLOv8 對所選定物體進行即時的追蹤及框選位置，再運用 KNN 結合 RGB/HSI 演算法將收集的資料進行判斷並分類；硬體方面透過使用 ESP32-CAM 當作視覺辨識的鏡頭即時擷取現場畫面，並透過 Wi-Fi 機建立一個區域網路實現無線串接，用以傳輸鏡頭擷取的畫面至視覺辨識模組。本研究採用一般筆記型電腦內程式半段顏色與距離結果，最後透過機械手臂吸取物體進行存放的作業程序。

3.3 視覺辨識模型訓練

此節運用流程圖闡述本研究視覺辨識模型中兩個模組流程建置，模型一為 YOLO 模組；模型二為顏色與距離辨識模組，其架構流程如圖 4。

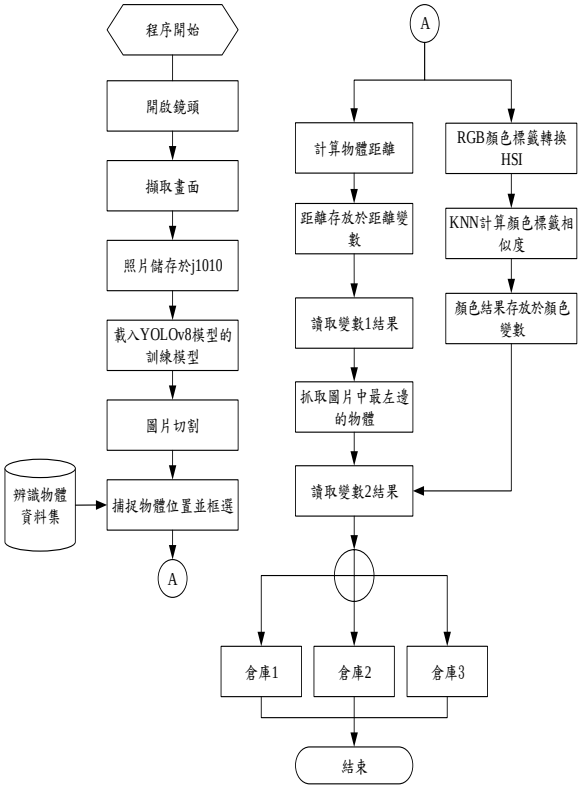


圖 4 視覺辨識流程圖

3.4 視覺辨識原理介紹

鏡頭接收攝影命令後將畫面擷取後，畫面再讀取至模型一將特定物體捕捉並框選，之後再將匡選物體位置的畫面傳給模型二，使其對框選物體進行顏色與距離判斷。以下為模型一及模型二的詳細說明及程式碼運作原理。

1. 模型一為 YOLOv8 物件偵測模型捕捉物體的

位置並框選出來，以下說明前置作業與模型訓練步驟。

(1) 前置準備：

(a) 資料夾建立：建立訓練集、測試集、驗證集並創建 images(儲存訓練用圖片資料)跟 labels(儲存標註資料)資料夾。

(b) 圖片分配：將收集到的圖片資料依照 8:1:1 比例分為訓練集、測試集、驗證集的 images 資料夾中。

(c) 圖片標註：將訓練集、測試集、驗證集中的照片使用標註工具進行標註。

(2) 模型訓練步驟：

(a) 配置文件：YOLOv8 的資料夾中新增 yaml 檔存放訓練集、驗證集及測試集三者圖片的相對路徑、類別數量與類別名稱。

(b) 新增模型放置資料夾：YOLOv8 的資料夾中新增 models 資料夾用於存放訓練用的模型或已訓練的模型，主要是確定本研究要使用的 YOLOv8 模型，因為 YOLOv8 提供了不同的預訓練模型。

(c) 開始訓練：新增一個 Python 檔開始進行訓練，載入 YOLOv8 模型後在裡面調整訓練參數，設定訓練次數、設定影像大小、設定批次大小、設定使用 4 個工作線程、設定使用 CPU 訓練，調整完參數後執行程式會針對訓練集中的圖片一直調整訓練權重，每一次權重的調整會越來越佳。

(d) 成果查看：執行程式查看訓練成果，如果 mAP50 的數字越接近 1 則模型判斷的準確度越高。

2. 模型二為 KNN 結合 RGB 演算法，其中我們將 RGB 顏色環境轉換成 HSI 顏色環境增加了對環境中的亮度進行考量，增加顏色判斷的正確率。顏色判斷模型的程式將分成四個部分，分別是準備要讓它訓練的資料（RGB 顏色樣本）、將 RGB 資料轉換成 HSI、訓練 KNN 模型、儲存訓練完的模型。

(1) 給予顏色資料：將我們設定的紅、藍、綠標籤，在 RGB 空間中定義出來，例如[255, 0, 0]跟[200, 20, 20]是紅色而[0, 0, 255]跟[20, 30, 200]是藍色。

(2) 顏色空間轉換：將 RGB 顏色空間轉換成 HSI 顏色空間，先將 RGB 顏色範圍從 0-255 改成 0-1，然後計算 RGB 三個通道的平均值出來就是 I(亮度)，之後在帶入公式計算 S(飽和度)，最後計算色相 H(色相)，而 H(色相)的計算方式是先求分子 num(這是為了計算色相的角度)、分母 den(這部分是 H 計算中的歸一化因子，確保輸入值不會落在[-1,1]的範圍內)、theta(計算色相角度後回傳一個 0-180 度的角度)，最後判斷 H(色相)在 0-360 度的範圍。以下是我們的計算公式及帶入實作程式的範例圖如圖 5。

$$S = 1 - \min(R, G, B) \div I$$

$$num = 0.5 \times \big((R - G) + (R - B)\big)$$

$$\text{den} = \sqrt{(R - G)^2 + (R - B) \times (G - B)}$$

$$then = \theta = arccos(num/den)$$

```
def rgb_to_hsi(rgb):  1 usage
    r, g, b = rgb / 255.0
    intensity = (r + g + b) / 3.0
    min_rgb = min(r, g, b)
    saturation = 1 - (3 / (r + g + b + 1e-6)) * min_rgb
    numerator = 0.5 * ((r - g) + (r - b))
    denominator = np.sqrt((r - g)**2 + (r - b)*(g - b)) + 1e-6
    theta = np.arccos(numerator / denominator)
    hue = theta if b <= g else 2 * np.pi - theta
    hue_deg = np.degrees(hue)
    return np.array([hue_deg, saturation, intensity])
```

圖 5 程式範例圖

(3) 演算法訓練：透過 KNN 演算法來計算我們所輸入的各個顏色範圍與各個定義的顏色標籤範圍之間的相似度，並透過歐幾里得距離(euclidean distance)衡量兩個點在顏色空間中直線距離，來判斷最接近的 k 個群組。以下為歐幾里得距離公式。

$$d = \sqrt{(R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2}$$

(4) 模組儲存：在顏色判斷的準確率可以達到百分之 90 左右後，將訓練完的顏色辨識程式轉換為一個模組，以後續引用作為視覺

模型的使用。

3.5 數據收集與分析

本研究針對三個物體進行拍照，分別收集到了不同角度跟亮度下三個物體的照片，並且每個物體分別具有 20 張左右的照片，以此做成的資料集進行訓練做成模組。以下是資料集的範例圖如圖 6。

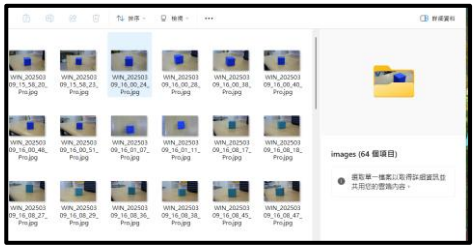


圖 6 資料集範例圖

3.6 選用演算法導入模型之評估

本研究針對 YOLOv5 跟 YOLOv8 各自訓練出了一個模組，發現在相同照片集下 YOLOv8 跟 YOLOv5 訓練他們訓練到第 11 次訓練時，YOLOv8 就已經比 YOLOv5 準確率(mAP50)高，因此本研究選擇 YOLOv8。以下是 YOLOv5 及 YOLOv8 的訓練對照圖如圖 7、圖 8。

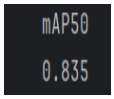


圖 7 YOLOv5 訓練圖

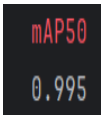


圖 8 YOLOv8 訓練圖

演算法方面原先只使用 RGB 顏色空間與 KNN 演算法的進行模組訓練，但是訓練出的模組在測試時發現物體顏色在不同亮度的環境條件下它的顏色辨識會出現錯誤，所以在程式中新增 HSI 顏色空間，將原本的 RGB 顏色空間轉換成 HSI 顏色空間後再帶入到 KNN 演算法進行訓練，發現在不同亮度下顏色辨識錯誤的問題已修正。以下是有無增加 HSI 的對照圖，如圖 9 圖 10。

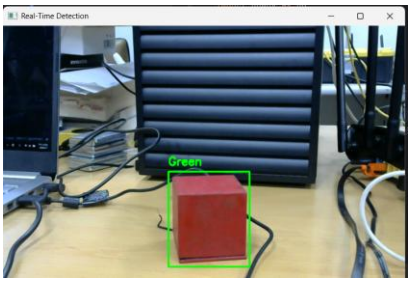


圖 9 無 HSI 時之圖片

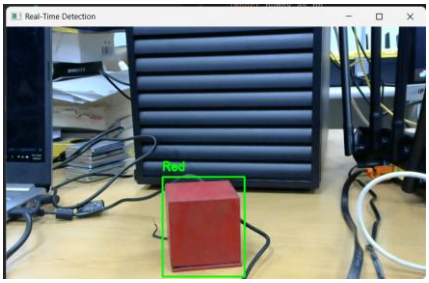


圖 10 加入 HSI 時之圖片

第四章 機械手臂設計與應用

4.1 倉儲環境建置與夾治具設計

本研究在自動化倉儲系統實作中，針對倉儲環境進行合理規劃與建置，確保整體系統在實際操作中能夠穩定且高效率的執行。倉儲環境設計以模擬實際倉儲作業為目標，規劃出物品待放區、機械手臂路徑與分類儲存區，而 ESP32-CAM 模組則安裝於機械手臂下方，作為影像擷取裝置，並考量機械手臂的活動範圍、奇異點與路徑流暢性，以確保機械手臂能夠順利完成夾取與放置作業。

在夾治具設計方面，所選用之物體含有金屬部件，因此選用電磁鐵作為夾治具，以實現快速且準確的物品吸取功能。電磁鐵具有控制簡易以及適合吸取金屬物品的優點，可透過電流控制其磁力開關，執行「吸起－釋放」之動作。與機械式夾爪相比，電磁鐵能夠在不需複雜機構設計下依舊能穩定吸取指定物品，特別適用於金屬材質之物件。

4.2 機械手臂作動路徑程式

俯視圖及正視圖和路徑結合說明。如圖 11 及圖 12。

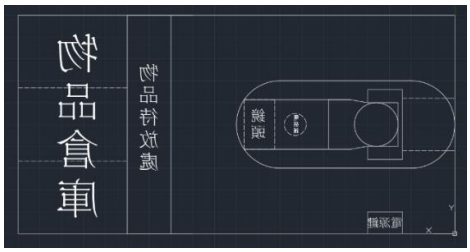


圖 11 機械手臂俯視圖

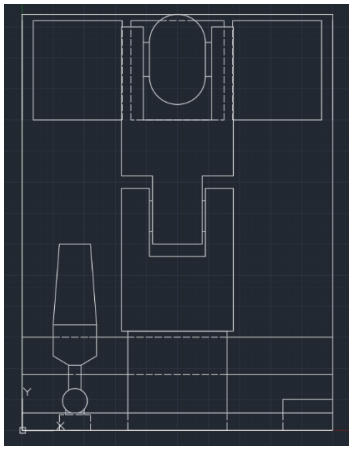


圖 12 機械手臂正視圖

首先，預先固定三個物品的待放處位置，並將待處理的物品分別擺放於這三個指定的位置上。啟動整體系統電源後，機械手臂根據所撰寫的控制程式，會先自動回歸至預設的「安全點」位置，確保機械手臂在移動前處於初始且安全的狀態。隨後機械手臂從安全點開始移動至指定拍攝處，搭配安裝於機械手臂末端的 ESP32-CAM 鏡頭進行定位，拍攝該處的畫面並擷取圖片資訊。此階段中，電腦會接收由機械手臂發送 R101 的值，當 R101＝1 時，鏡頭便啟動拍照，並擷取該位置的影像資訊，再回傳至電腦執行視覺辨識模型。將擷取到的影像將進行圖像處理，以辨識畫面中所有物體的顏色資訊。

系統會優先擷取從左至右排列中最左側的待放物體，並將其顏色與位置轉換為機械手臂的判斷執行指令。根據辨識結果，物品的待放位置將以 R102 參數表示，從左至右分別為 1、2、3；而物體的顏色則以 R103 表示，分別對應為 R103 等於 1 代表為紅色物料，分類至倉庫存放口 1；R103 等於 2 代表為綠色物料，分類至倉庫存放口 2；R103 等於 3 代表為藍色物料，分類至倉庫存放口 3。

4.3 實際機械手臂程序運作說明

手臂到達拍攝指定位後，將發送 R101 為 1 的指令給予視覺辨識模型，視覺辨識模型即通過 URL 開啟 ESP32-CAM 鏡頭，拍攝照片後進行後續判斷，機械手臂指令如圖 13 紅框區。



圖 13 機械手臂發送指令開啟鏡頭程式圖

完成分類判斷後，電腦會將對應的 R102 與 R103 值傳送給機械手臂。視覺辨識結果如圖 14。



圖 14 電腦端判斷程式圖

電腦傳送至機械手臂之 R 值，能夠透過手持盒內之資源選項，確認物體辨識結果是否正確，如圖 15。



圖 15 機械手臂資源圖

當 R102 ＝ 1 時，機械手臂程式便會執行跳躍

指令，使機械手臂執行標號 1 之指令，即電磁鐵移動至待放區 1 進行吸取動作，如圖 16。



圖 16 流程判斷程式圖

該物體的顏色判定紅色時，則設定 R103＝1，程式執行跳躍指令，使機械手臂執行標號 5 內部指令，即將物體送入指定倉庫之指令，將物體放置於倉庫 1(紅色存放區)，以此類推。如圖 17

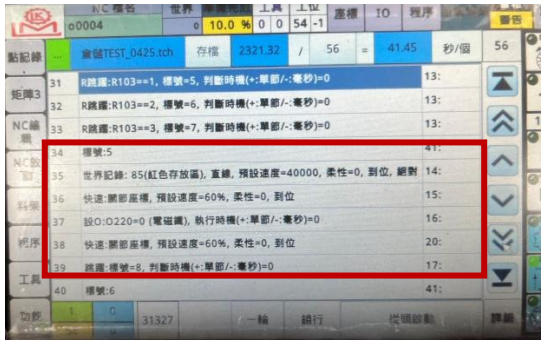


圖 17 顏色儲存判斷程式圖

到達指定位後關閉電磁鐵，即完成一次物體存放作業。每完成一輪作業，機械手臂會自動回到安全點，並清空 R101 至 R103 參數，供下一個物品的辨識與處理流程，如圖 18。此流程將不斷重複執行，直到所有物品皆完成分類與存放作業為止。

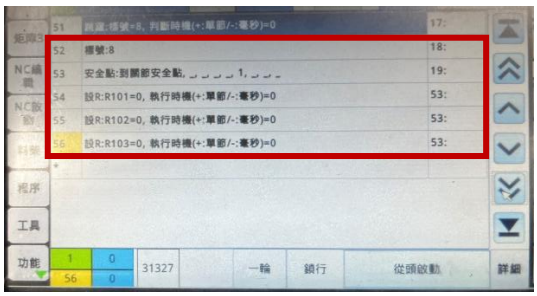


圖 18 手臂回至安全點與清空相關參數程式圖

第五章 結論

本研究之目的在於研究視覺辨識與機械手臂之結合，目標在於透過加強現有設備的自動化功能，實現智慧自動化倉儲。研究中訓練視覺辨識模型，以提升辨識準確性減少人為錯誤；同時設計機械手臂的操作路徑，實現最高效率之倉儲流程，機械手臂結合視覺辨識模組實現自動抓取與分類儲存功能，並建立完整的倉儲作業流程。實驗結果驗證本系統在實際應用中能夠有效提升作業效率、降低人為錯誤與運作成本，以達到自動化倉儲之目的。

透過攝影鏡頭接收指令後，擷取影像，並將擷取影像傳送至 YOLOv8 模型進行物體偵測與框選，接著將框選完之影像傳送至結合 RGB 與 HSI 色彩轉換的 KNN 模型進行顏色判斷與分類，最終根據分類結果判定物品的存放路徑，並由機械手臂執行自動抓取與存放流程。實驗結果證明，此系統能有效提升倉儲作業之效率、減少人為錯誤並降低成本，成功實現了自動化倉儲的目標。

本研究主要分為兩部分，以下說明

1. 視覺辨識模組之訓練本模組：主要負責物件之偵測與特徵判別，並作為後續自動化搬運與分類儲存的基礎。視覺辨識模組分為兩個模型進行訓練與設計。

   (1) 模型一：YOLOv8 物件偵測模型：採用 YOLOv8（You Only Look Once）進行即時影像之物體捕捉與框選。該模型具有高速且高準確率的特性，能快速辨識畫面中多種不同物件，並標註其位置，適用於即時倉儲辨識。模型訓練流程包括資料蒐集、使用 Label Studio (labelsy)進行標註，透過預訓練模型進行訓練，加速訓練與提升其準確率，最終模型具備即時多物件捕捉與框選之能力，並將處理後之影像傳送至後續色彩辨識模組。

   (2) 模型二：KNN 結合 RGB 演算法：在模型一框選之基礎上，模型二進一步分析物件之色彩特徵。透過將 RGB 資料轉換為 HSI 色彩空間，以色相（Hue）作為主要分類依

據，經過轉換後的色彩資料輸入 KNN 模型進行訓練，透過計算樣本間的歐幾里得距離判斷最相近的顏色類別。當模型準確率達到 90%後，將其儲存為模組，使其運作中能夠即時辨識物品顏色並進行分類。

2. 自動倉儲系統建置：其建置規劃包括倉庫、點位位置與機械手臂之控制設計，以下說明。
   (1) 倉儲空間規劃：依據目標應用設計合理的倉庫格局與物品待放區域，確保機械手臂操作之可行性與效率。
   (2) 鏡頭位置設計：鏡頭裝設至機械手臂前端，能夠隨機械手臂移動，確保畫面資訊完整性與辨識準確性。
   (3) 待放點與分類區設計：規劃物品放置初始區域與最終儲存點之座標，並考量機械臂之奇異點和活動範圍與路徑流暢性進行設計。
   (4) 機械手臂控制程式：根據視覺模組提供之物件座標與分類資訊，設計機械手臂移動路徑，實現自動倉儲的流程。控制程式應避免碰撞並最佳化移動路徑，以提升整體倉儲效能。

視覺辨識不僅僅是物體的分類輔助工具，透過大量運用 AI 與各項機械化設備的整合達到產線接近零人工，或是減少人力支出；未來本研究將從下述方向進行改善優化。

1. 路徑的判斷：透過整合深度的路徑規劃演算法讓移動的路徑可以變得更加多元；以及利用物品隨機擺放的方式訓練資料集，讓系統適應非規則或混亂的擺放情境。
2. 物品的判斷：擴展更多顏色甚至是不同形狀的物品，如正方形、長方形、圓形等。
3. 增加 RFID 身分識別及 UI 介面：結合 RFID 在每個商品上一個標籤，這樣商品的生產歷程就能完整的記錄下來，在開發一個 UI 介面讓生產者或管理者能一目瞭然的了解到此商品的生產情況，包含哪些是良品或是不良品哪些需要報廢等，便能提前去做商品的庫存管理等。

參考文獻

1. FORTINET，什麼是使用者資料包通訊協定（UDP）？，https://www.fortinet.com/tw/resources/cyberglossary/user-datagram-protocol-udp
2. FENED，機械手臂原理是什麼？5 大機械手臂類型分析、機械手臂推薦！，https://www.fened.com.tw/blog/robotic-arm-principle。
3. Julian Liu，Python 程式設計實務應用：YOLOv8 安裝，https://www.youtube.com/watch?v=ydC9x1L5SJU。
4. Medium，Pytorch 基本介紹與教學，https://medium.com/ching-i/pytorch-%E5%9F%BA%E6%9C%AC%E4%BB%8B%E7%B4%B9%E8%88%87%E6%95%99%E5%AD%B8-ac0e1ebfd7ec。
5. Medium，[NextGen SC] 邁向電商倉全無人化，我們需要機械手臂隨機揀貨方案，https://peichialiu.medium.com/nextgen-sc-%E9%82%81%E5%90%91%E9%9B%BB%E5%95%86%E5%80%89%E56. Medium，【機器學習筆記】KNN(K Nearest Neighbor)，https://medium.com/@SCU.Datascientist/python%E5%AD%B8%E7%BF%92%E7%AD%86%E8%A8%98-knn-k-nearest-neighbor-531a95336f71。
7. pin luarn，*2022 9.1* 樂老師暑期課程 s9.1 集❤️YOLO v5 安裝與簡單應用 1😊，https://www.youtube.com/watch?v=SU_XvZKrClA&t=1270s。
8. pin luarn，*2022 9.3*樂老師暑期課程 s9.3 集❤️YOLO v5 GPU 環境安裝與設定😊，https://www.youtube.com/watch?v=0V5EXMw2wsI。
9. 大和有話說，【圖解】工業機器人之五大機械結構及關鍵零組件分析｜大和有話說，https://dahetalk.com/2018/02/19/%E3%80%90%E5%9C%96%E8%A7%A3%E3%80%91%E5%B7%A5%E6%A5%AD%E6%A9%9F%E5%99%A8%E4%BA%BA%E4%B9%8B%E4%BA%94%E5%A4%A7%E6%A9%9F%E6%A2%B0%E7%B5%90%E6%A7%8B%E5%8F%8A%E9%97%9C%E9%8D%B5%E9%9B%B6%E7%B5%84%E4%BB%B6/。
10. 楊康葉、鄔春學(2013)，基于 RGB 模型颜色相似性的彩色图像分割，*中國科技核心期刊，22(3)*，128-131，https://www.c-s-a.org.cn/csa/article/abstract/20130330?st=advanced_search。
11. 林義楠、吳亞芬、孫子恩、許瀟傑(2024)，基於動物辨識的 YOLOv5 與 YOLOv8 之性能比較，明新學報，47(2)，https://acade.must.edu.tw/admin/files/47%E5%8D%B72-%E6%98%8E%E6%96%B0%E5%AD%B8%E5%A0%B1(%E5%9F%BA%E6%96%BC%E5%8B%95%E7%89%A9%E8%BE%A8%E8%AD%98%E7%9A%84YOLOv5%E8%88%87YOLOv8%E4%B9%8B%E6%80%A7%E8%83%BD%E6%AF%94%E8%BC%83)e024002-%E8%A8%B1%E7

%80%9F%E5%82%91__%E5%90%B3%E4%BA%9E%E8%8A%AC__%E5%AD%AB%E5%AD%90%E6%81%A9_%E6%9E%97%E7%BE%A9%E6%A5%A0_20240416083826.pdf。

%85%A8%E7%84%A1%E4%BA%BA%E5%8C%96-%E6%88%91%E5%80%91%E9%9C%80%E8%A6%81%E6%A9%9F%E6%A2%B0%E6%89%8B%E8%87%82%E9%9A%A8%E6%A9%9F%E6%8F%80%E8%B2%A8%E6%96%B9%E6%A1%88-2656ba74fcf0。

12. 林義楠、吳亞芬、孫子恩、許瀟傑(2024)，基於動物辨識的 YOLOv5 與 YOLOv8 之性能比較，明新學報，47(2)，https://acade.must.edu.tw/admin/files/47%E5%8D%B72-%E6%98%8E%E6%96%B0%E5%AD%B8%E5%A0%B1(%E5%9F%BA%E6%96%BC%E5%8B%95%E7%89%A9%E8%BE%A8%E8%AD%98%E7%9A%84YOLOv5%E8%88%87YOLOv8%E4%B9%8B%E6%80%A7%E8%83%BD%E6%AF%94%E8%BC%83)e024002-%E8%A8%B1%E7%80%9F%E5%82%91__%E5%90%B3%E4%BA%9E%E8%8A%AC__%E5%AD%AB%E5%AD%90%E6%81%A9_%E6%9E%97%E7%BE%A9%E6%A5%A0_20240416083826.pdf。

13. 葉焦明(2019)，結合 OpenCV 視覺辨識系統之倉儲物料智慧化管理系統開發，東海大學，碩士論文。

14. 郭彥助(2022)，OpenCV 影像處理與機器學習應用於桌上空氣曲棍球之互動控制系統，國立雲林科技大學，碩士論文。

15. 維基百科，K-近鄰演算法，https://zh.wikipedia.org/zh-tw/K-%E8%BF%91%E9%82%BB%E7%AE%97%E6%B3%95。

16. 維基百科，機械手臂，https://zh.wikipedia.org/zh-tw/%E6%A9%9F%E6%A2%B0%E6%89%8B%E8%87%82。

17. 騰訊云，YOLO 系列算法全家桶——YOLOv1-YOLOv9 詳細介绍！！，https://cloud.tencent.com/developer/article/2406045。