

# 自動貨架巡航機器人

## Automated Shelf Inspection Robot

指導教授：李國川教授

學生：林昌明、林冠廷、楊翔鈞

國立聯合大學 資訊工程學系

苗栗市南勢里聯大 2 號

[gcllee@nuu.edu.tw](mailto:gcllee@nuu.edu.tw)

### 摘要

在過去兩三年中，全球疫情和經濟結構的變動已經引發了大規模的離職潮，對各行業造成了嚴重的人力短缺問題。零售業尤其受到影響，迫切的需要應對這些挑戰。為了解決人手不足的問題，我們提出自動化盤點的概念，以解決傳統手動盤點的繁瑣和耗時問題。

本專題的研究目標主要包括「辨別產品種類並計算數量」以及「自動化移動和避障功能」。我們採用 Jetson NX 作為機器人的主機，透過 ESP32-S[1]向馬達發送脈衝訊號實現移動控制。在移動過程中，利用 Webcam 即時獲取影像，接著，我們運用 YOLOv7[2]進行產品辨識並計算總數。同時，在 ROS[3]架構下，使用 YDLidar X2[4]實現偵測障礙物，並透過演算法實現避障功能，藉此製作出一個能夠自動化盤點的機器人。

關鍵詞：YOLOv7、障礙物偵測、自動化盤點

### Abstract

In the past few years, the global pandemic and changes in economic structures have triggered a widespread wave of resignations, leading to a severe shortage of manpower across various industries. The retail sector, in particular, has been significantly impacted and urgently needs to address these challenges. To tackle the issue of manpower shortage, we propose the concept of automated inventory management

to overcome the tedious and time-consuming nature of traditional manual inventory processes.

The primary research objectives of this project include "identifying product types and calculating quantities" and "implementing automated movement and obstacle avoidance functions." We utilize Jetson NX as the robot's main controller, sending pulse signals to the motors via ESP32-S for movement control. During movement, real-time imaging is captured using a Webcam. Subsequently, we employ YOLOv7 for product recognition and quantity calculation. Simultaneously, within the ROS framework, utilizing YDLidar X2 to implement obstacle detection and obstacle avoidance functionality through algorithms, thereby creating a robot capable of automating the inventory process.

**Keyword: YOLOv7, obstacle detection, automated inventory**

### 第一章、背景及目的

本專題的研究目標在於製作出一個能夠自動化盤點的機器人，因此，主要會分成「辨別產品種類並計算數量」以及「自動化移動和避障功能」兩個部分

1.辨別產品種類並計算數量: 透過攝影機來擷取產品的即時影像，並使用物件偵測的方法辨識產品種類，同時，計算每個產品相對應的產品數量。

2.自動化移動和避障功能: 透過對機器人

的馬達發送移動訊號實現移動控制，同時，使用雷達偵測障礙物距離，運用演算法實現避障功能。

## 第二章、系統架構與操作流程

### 1. 系統架構圖



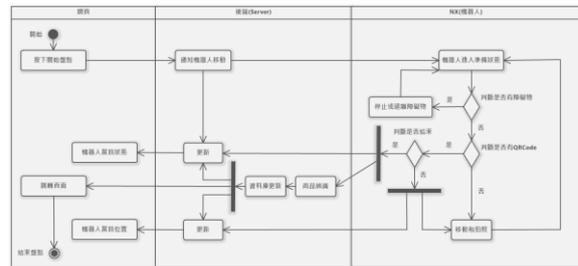
圖一 系統架構圖

在機器人端，使用 Jetson NX 作為機器人的主機，透過 ESP32-S 向馬達發送脈衝訊號實現移動控制。同時，機器人在移動過程中利用 Webcam 獲取產品的即時影像，在 ROS 架構下，使用 YDLidar X2 偵測障礙物距離，透過演算法實現避障功能。

在伺服器端，伺服器主機利用機器人拍攝的影像資訊，透過 YOLO 辨識計算所有產品的總數。接著，將模型預測完的數據存入資料庫，同時建立 RESTful API 供網頁端擷取資料庫中的盤點資訊。此外，透過 socket 傳遞來自網頁端和機器人端的盤點訊號。

最後，在網頁端，使用者可以透過個人電腦或行動裝置進入網頁，操控機器人進行盤點。並且能監測機器人的所在位置和盤點狀態，並查看最新一次盤點的產品明細和待補清單等盤點相關資訊。

### 2. 操作流程圖



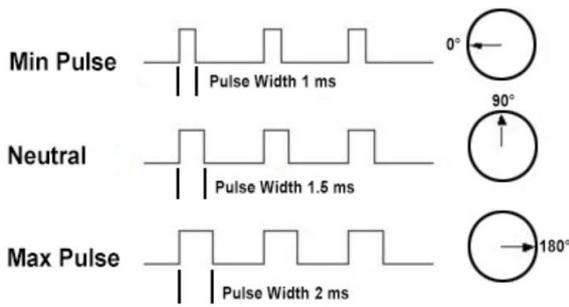
圖二 操作流程圖

## 第三章、專題理論與實作

### 1. 機器人移動控制流程

脈衝寬度調變 (Pulse Width Modulation, PWM) 是一種通過改變訊號的脈衝寬度來調整平均功率的技術，訊號由一系列週期性的脈衝組成，每個脈衝的寬度都可以被調整，因此，這項技術時常被應用在馬達控制、燈光調節和電源管理等領域中。

在我們的專題中，同樣使用 PWM 控制機器人馬達的轉動速度和轉動方向。然而，我們的馬達僅能接受 1ms 到 2ms 之間的脈衝寬度，因此需要控制寫入馬達的脈衝寬度。就如圖 3 所示的馬達轉動角度和脈衝寬度的對照圖，當脈衝寬度為 1.5ms 時，馬達轉動角度維持在 90 度，馬達不會轉動；在 1ms 到 1.5ms 之間，馬達會順時針轉動，數值越低轉速越快，當 1ms 時，馬達轉動角度維持在 0 度，則會達到順時針最快的速度；而在 1.5ms 到 2ms 之間，馬達會逆時針轉動，數值越高轉速越快，當 2ms 時，馬達轉動角度維持在 180 度，則會達到逆時針最快的速度。



圖三 機器人移動控制流程

圖 4 為完整的機器人移動控制流程，首先，配置正確的序列埠 (serial port)，使得 Jetson NX 能夠透過 ESP32-S 對伺服馬達寫入訊號。同時，將每秒傳輸的位元數量 (baud rate) 設置成能夠與 Jetson NX 通訊的數值。接著，在 ESP32-S 上為每個伺服馬達初始化相對應的腳位。最後，持續的從序列埠讀取訊息，並根據寫入的訊息對伺服馬達進行相對應的脈衝寬度，藉此實現機器人的前進、後退、旋轉甚至是橫向的移動。

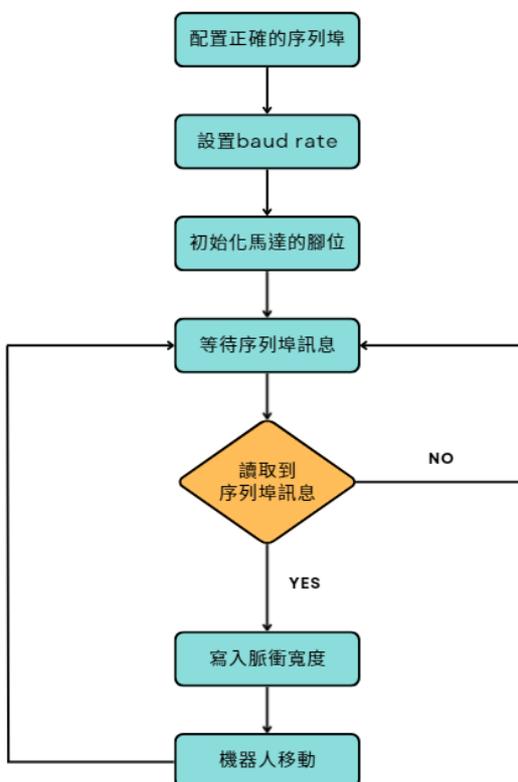


圖 4 機器人移動控制流程

## 2. 障礙物距離偵測

YDLIDAR X2 是一款能夠實現 360 度全方位掃描的 2D 激光雷達，每秒能夠執行 3000 次障礙物距離偵測，可覆蓋 0.12 至 8 公尺的測距範圍。同時，角度的解析度介於 0.6 度至 0.96 度之間，相當於將一個完整的圓周細分成 375 至 600 個點，使其能夠每秒執行至少 5 至 8 次完整的圓周偵測。因此，我們能夠實現即時且連續的獲取雷達的距離資訊，呈現出類似圖 5 的障礙物距離可視圖。

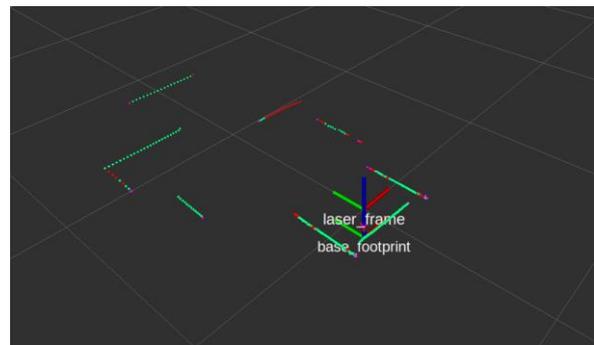


圖 5 障礙物距離可視圖

## 3. 實現避障功能

為了避免雷達受到干擾，導致掃描到的距離資訊產生誤差，我們透過計算一段特定距離的連續出現次數來判斷某個特定方向是否有障礙物。如果在前方檢測到距離小於機器人安全移動距離的障礙物，機器人將會停止移動。若障礙物位於左側，機器人將會在右側沒有障礙物的情況下向右移動，否則將會停止移動。

## 4. 產品辨識

在本專題中，我們採用 YOLOv7 進行產品種類辨識和數量清點。這一選擇是基於 YOLOv7 同時支援 PyTorch 和 TensorFlow 兩個機器學習框架，使得我們能夠更輕鬆的進行模型優化。

我們選用十種產品進行貨架上的產品辨識模擬，這些產品包括可口可樂 (209 筆)、雪碧 (320 筆)、美麗果 (259 筆)、蔓越莓果汁 (135 筆)、維力炸醬麵 (179

筆)、統一肉燥麵(187筆)、綠茶(272筆)、奶茶(104筆)、辛拉麵(115筆)，總共 892 張圖片。

我們使用 Roboflow 進行圖片資料集的標註。根據 70%:20%:10%的比例，我們分配了訓練集、驗證集和測試集的佔比。接著，我們將 Roboflow 標註後的圖片資料集設定檔代入訓練參數，進行模型訓練。最後，我們將訓練後的權重檔存放在 YOLOv7 的環境中，藉此實現產品辨識的功能。

## 5. QRCode 辨識

在本專題中，我們的貨架被劃分為 ABCD 四個區域，每個區域都貼有相對應的 QRCode，提供機器人識別當前位置。在機器人行進的過程中，拍到的的照片會透過 Pyzbar 套件[5]進行 QRCode 偵測，接著，將 QRCode 的資訊解碼成二進位數，最後，將解碼後的文字內容作為產品照片分區存放的依據。

## 6. 伺服器端實作

在 Node.js 的環境下，使用 Express 框架建立 RESTful API，進行 HTTP 的通訊請求，實現資料庫讀寫和盤點照片上傳的功能。同時，經由維持與網頁端和機器人端的 socket 連線，達到即時的盤點訊號傳遞，使得機器人端能夠迅速回應網頁端。此外，機器人端具備發送當前盤點區域和盤點狀態的資訊給網頁端的功能，實現網頁端對機器人狀態的實時監控和即時更新。圖 6 為 WebSocket 和 HTTP 兩種通訊協定的比較圖。

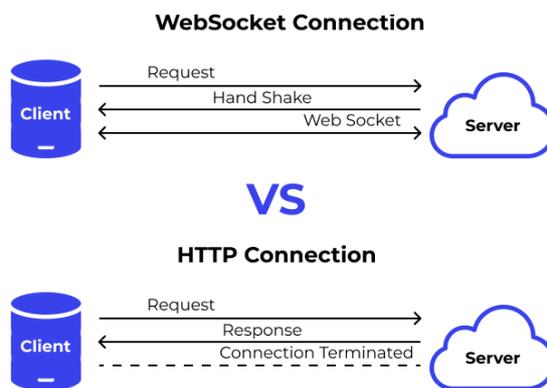


圖 6 通訊協定比較圖

## 7. 網頁端頁面呈現

### 7.1 首頁

如圖 7 所示，頁面左半部有一個機器人所在位置的地圖，而右半部有一個「開始盤點」的按鈕。如圖 8 所示，當按下「開始盤點」的按鈕之後，頁面會顯示「機器人移動中」的文字訊息，同時，以深藍色文字閃爍來標示機器人的所在區域，並以淺藍色背景填滿機器人到達過的區域。如圖 9 所示，當機器人盤點結束，頁面會顯示「產品辨識中」的文字訊息，直到產品辨識完成之後，系統跳轉至產品清單的頁面，呈現最新的產品資訊和待補狀況。

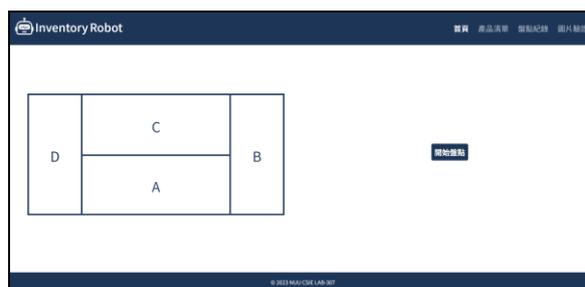


圖 7 首頁(未開始盤點)

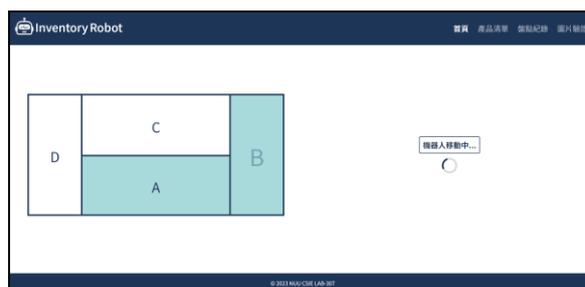


圖 8 首頁(機器人移動中)

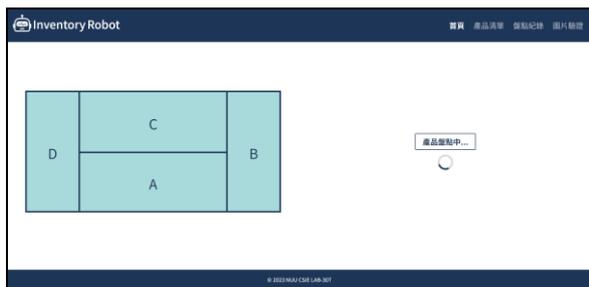


圖 9 首頁(產品盤點中)

## 7.2 產品清單

如圖 10 所示，頁面左半部會以表格的方式陳列出所有產品的資訊，包括「編號」、「產品名稱」、「目前數量」、「貨價容納數量」以及「分區」，若有產品數量不足，則會顯示在右半部的待補產品清單上，同時顯示相對應的待補數量，如果沒有產品需要補貨，則會以灰色的字提示。



圖 10 產品清單

## 7.3 盤點紀錄

如圖 11 所示，頁面會顯示最新一次各區域盤點到的所有產品的「產品名稱」、「盤點總數」以及該區域的「盤點結束時間」。此外，使用者還可以透過下拉選單按鈕查看過往的盤點紀錄。



圖 11 盤點紀錄

## 7.4 圖片驗證

如圖 12 所示，頁面會顯示最新一次各區域盤點的所有產品對應的盤點照片。同時，使用者也可以透過下拉選單按鈕查看並且驗證過往的盤點照片。

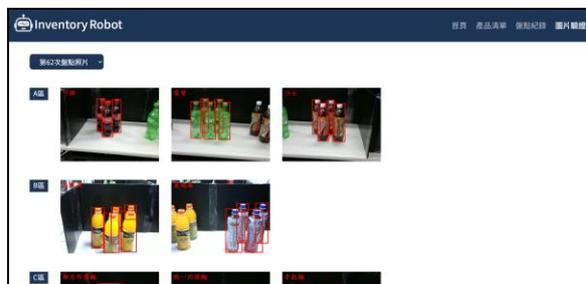


圖 12 圖片驗證

## 第四章、參考文獻

- [1] “ESP32-S,” [線上]. Available: <https://zh.wikipedia.org/zh-tw/ESP32>. [存取日期: 26 11 2023].
- [2] “YOLOv7 Github,” [線上]. Available: <https://github.com/WongKinYiu/yolov7>. [存取日期: 5 7 2023].
- [3] “機器人作業系統,” [線上]. Available: <https://zh.wikipedia.org/zh-tw/%E6%A9%9F%E5%99%A8%E4%BA%BA%E4%BD%9C%E6%A5%AD%E7%B3%BB%E7%B5%B1>. [存取日期: 27 11 2023].
- [4] “Ydlidar X2 使用手冊,” [線上]. Available: [https://www.ydlidar.com/service\\_support/download.html?gid=6](https://www.ydlidar.com/service_support/download.html?gid=6). [存取日期: 3 7 2023].
- [5] “Detect and read barcodes and QR codes with ZBar in Python,” [線上]. Available: <https://note.nkmk.me/en/python-pyqzbar-barcode-qr-code/>. [存取日期: 20 7 2023].