

靜 宜 大 學

資 訊 工 程 學 系

畢 業 專 題 成 果 報 告 書

自 動 化 生 產 線

學 生：

資工三A 411147500 李紘緯

指導教授：翁添雄 教授

西元二0二四年十二月

目錄

一、摘要.....	3
二、研究動機.....	3
三、文獻探討.....	4
四、研究方法.....	4
1. 系統設計與方法概述	4
(1) 架構圖	4
(2) 系統功能	5
(3) 系統數據流向圖	5
(4) 開發環境	6
2. 系統實作結果	6
(1) YOLOv7	6
(2) DOBOT	8
(3) SOCKET	9
(4) WEB	10
(5) NODE. JS	10
(6) MYSQL	10
五、成果展現.....	11
六、結論與未來發展.....	11
1. 結論	11
2. 未來發展.....	12

一、摘要

在台灣有許多傳統工廠中，儘管機器逐漸取代人力，但仍有多數廠房需要人工操控手臂進行物體搬運。這雖然節省了人力，卻仍需投入時間和人力進行操控。提出了將人工智慧應用於機器手臂的想法，使其能夠像人一樣判別物體，應用於處理客戶訂單，實現自動化搬運與訂單處理的目標。

然而，目前機器成本仍然是一個阻礙自動化的問題。因此使用免費的開源程式碼和Python套件，以解決成本高昂的問題，從而提升自動化的速度。這一解決方案有望使更多企業接受自動化技術，促進工業升級。

二、研究動機

在許多產業中，機器逐漸取代人力，台灣許多工廠尚未達到完全自動化的程度，這些工廠仍需依賴人力操控機械手臂來完成搬運工作。雖然這種方式減少了直接的人力投入，但仍需人工操作，導致時間與人力的額外消耗。

如果能將人工智慧（AI）技術應用於機械手臂，使其具備類似人類的判別能力，則可大幅提升自動化水平。例如，透過電腦視覺技術讓手臂識別並分類物體，並進一步結合訂單管理系統，機器手臂可自動抓取符合客戶訂單需求的物品並將其打包，實現真正的無人工干預作業流程。這樣的創新將不僅提高效率，還能減少錯誤率，為企業創造更高的經濟效益。

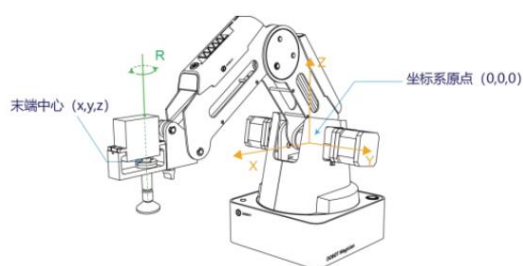
然而，另一個重要的挑戰在於機器設備的高成本。這對於中小型企業而言尤其是一道門檻。為了降低企業進入自動化的障礙，我們可以充分運用當前許多免費的開源資源，例如開源的程式碼庫和基於Python的相關套件。這些工具不僅能顯著降低開發成本，還能讓企業在無需大量投入的情況下快速實現基礎自動化。例如，使用開源的深度學習框架（如TensorFlow或PyTorch）訓練物體識別模型，結合ROS（Robot Operating System）來控制機械手臂，企業可在短時間內部署可行的解決方案。

未來，透過不斷改進程式碼和技術，這些機械設備不僅能更靈活地執行複雜的任務，還能逐漸普及至更多的產業與場景，最終助力產業升級並促進經濟發展。

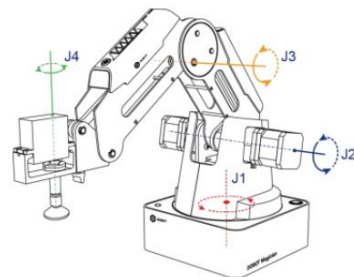
三、文獻探討

DOBOT Magician的介紹

DOBOT Magician是一款四軸的機器手臂，座標軸就分成兩種，第一種就是笛卡爾座標，第二個則是關節座標，這兩個各有自己的優缺點，笛卡爾座標如圖(一)，此座標系，如果是面對同個方向移動時，不會對機器產生卡損，若今天搬運物體需要轉移個面向時，則使關節座標如圖(二)會是最好的選擇，若使用錯誤的坐標系時對機器都是一種損壞，但並不是關節座標就是絕對的優勢，在垂直移動時，笛卡爾座標只需要改變z軸，而關節座標需要改變x、y、z，寫程式時，透過思考動作來做出最快且最易維護的程式碼。



圖(一)笛卡爾座標



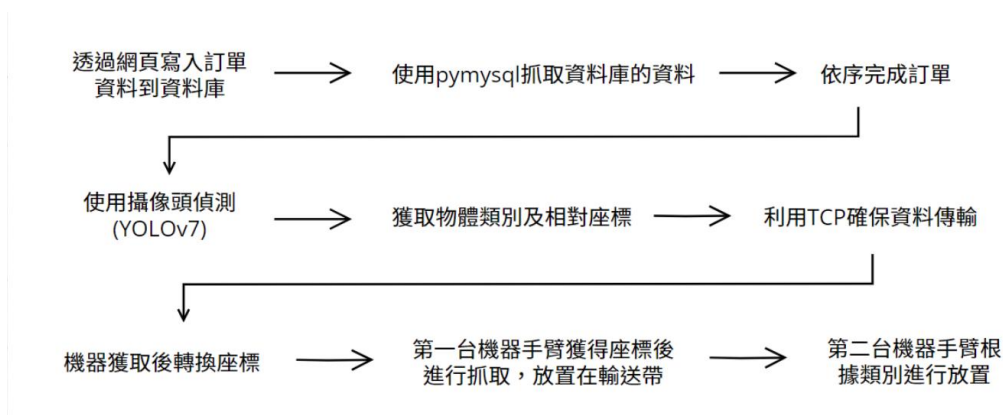
圖(二)關節座標

(圖片來源: <https://blog.csdn.net/GUA8122HOU/article/details/119626113>)

四、研究方法

1. 系統設計與方法概述

● 架構圖



● 系統功能

這套系統結合了 YOLOv7 物體辨識技術、Python Socket 通訊模組、Dobot 機器手臂控制模組、MySQL 資料庫 及 JavaScript，實現了一個高效且智能化的訂單處理解決方案。

- YOLOv7 物體辨識技術作為核心，能夠快速且精準地識別訂單中的各種物品，即使面對複雜場景也能提供準確的辨識結果，為後續自動化操作提供關鍵數據支持。
- Python Socket 通訊模組是系統的數據傳遞中樞，負責模組間的即時協作，確保 YOLOv7 的偵測結果和指令能高效傳遞至其他模組。
- Dobot 機器手臂控制模組根據辨識結果執行分揀與分類，展現高效且準確的自動化能力，靈活應對不同訂單需求，顯著提升處理效率並降低錯誤率。
- MySQL 資料庫負責存儲訂單數據，支持快速查詢和管理，提供高度安全性與數據完整性，為整體系統提供穩定可靠的數據支撐。
- JavaScript 則構建了直觀友好的用戶界面，讓用戶能輕鬆進行訂單操作、監控系統狀態，並快速了解處理進度，提升了整體使用體驗。

● 系統數據流向圖



1. 資料流概述：

1. 從網站接收訂單資料：
2. 資料庫抓取資料並傳給 YOLOv7：

2. YOLOv7 模型的設計與優化：

1. 訓練與測試集準備
2. 模型訓練與調參
3. 測試與部署

3. 資料傳輸：使用 Python 的 Socket TCP 通信

1. Python 端 Socket 實現
2. 機械手臂接收並處理指令

4. 機器間的協作：Thread 實現：

1. 多線程協調控制

5. 最終整合架構：

1. Node.js 後端處理訂單
2. 資料庫與 Python 互動
3. YOLOv7 進行物件偵測
4. 多手臂協作執行訂單

• 開發環境

- 框架：Pytorch 1.x
 - 語言：Python
 - 硬體：個人筆電、機器手臂、輸送帶、紅外線感測器
 - 軟體：PyCharm、MySQL、VScode
-

2. 系統實作結果

A. YOLOv7

(1)前置設定

YOLOv7在本研究中算是最重要的一個部分，首先需要樣本收集，因為需要以大量的圖片進行訓練，且須將這些圖片進行標記，而標記軟體使用的是www.makesense.ai網站以及labeling，這兩個標記工具都十分簡單上手，完成標註後就需要

1. 設定好訓練和測試資料的路徑。
2. 更改yaml檔裡面的分類。

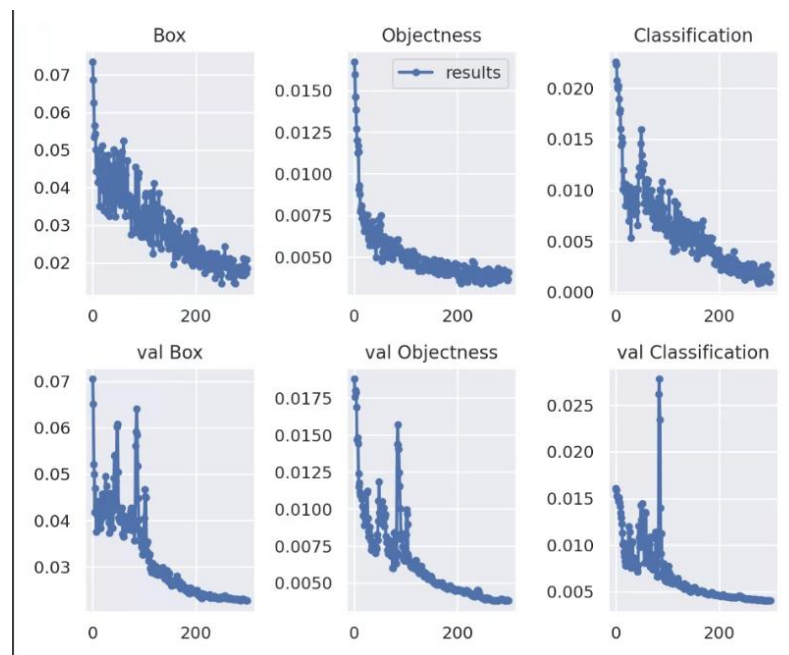
(2)在train.py裡面設置好路徑

```
if __name__ == '__main__':
    parser = argparse.ArgumentParser()
    parser.add_argument('--weights', type=str, default='yolo7.pt', help='initial weights path')
    parser.add_argument('--cfg', type=str, default='', help='model.yaml path')
    parser.add_argument('--data', type=str, default='data/coco.yaml', help='data.yaml path')
    parser.add_argument('--hyp', type=str, default='data/hyp.scratch.p5.yaml', help='hyperparameters path')
    parser.add_argument('--epochs', type=int, default=300)
    parser.add_argument('--batch-size', type=int, default=16, help='total batch size for all GPUs')
    parser.add_argument('--img-size', nargs='+', type=int, default=[640, 640], help='[train, test] image sizes')
```

(3)執行訓練

在應用部分，主要透過調整epoch和batch_size觀察損失函數(訓練損失、驗證損失)結果呈現(收斂情況)

如下圖呈現，由於我們訓練的目標地是普通盒子，且屬於同一類型，實質上對於模型應無太大難度，故結果圖呈現良好。



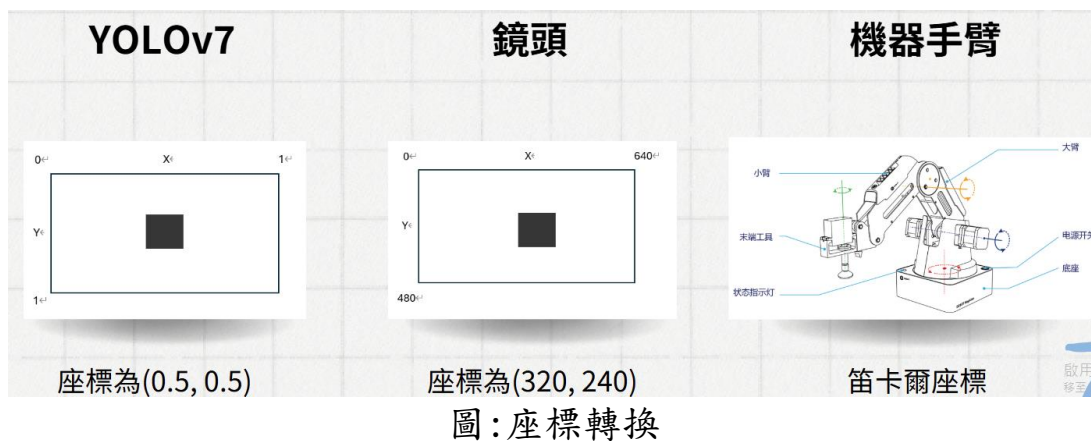
圖：使用YOLOv7訓練結果圖

(4)執行偵測

引入訓練完結果較好的權重執行偵測檔案

```
parser.add_argument('--weights', nargs='+', type=str, default='exp8\weights\best.pt', help='model.pt path(s)')
parser.add_argument('--source', type=str, default='test', help='source') # file/folder, 0 for webcam
```

圖：參數設置



C. SOCKET

在此專題中，將YOLO檔案設定為client，DOBOT檔案設定為server。

```
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect(server_address)
    data = f"{label},{x_c},{y_c}".encode()
    s.sendall(data)
```

圖: client端

```
while True:
    # 等待客戶端連接
    conn, addr = s.accept()
    with conn:
        print('Client connected:', addr)
        while True:
            data = conn.recv(1024) # 接收 client 傳來的資料
            if not data:
                break
            cls, x, y = data.decode().split(",") # 將資料解碼，得到 x, y 座標
            # 將其str轉成浮點數
            # cls = float(cls)
            x, y = float(x), float(y)
            # print(cls, x, y)
```

圖:server端

D. 網頁設計



訂單輸入

在此網站輸入您要的訂單資訊，我們會依序執行

輸入企業名稱

Red Candy Box

輸入數量

Blue Candy Box

輸入數量

Pink Candy Box

輸入數量

提交

圖:網頁

E. node. js

作為網頁後端，接收網頁前端傳遞的資訊，傳送到MySQL。是一個中繼者的角色

```
// MySQL 連線設定
const db = mysql.createConnection({
  host: 'localhost',
  user: 'root',      // 替換成你的 MySQL 用戶名
  password: 'changeme', // 替換成你的 MySQL 密碼
  database: 'orders' // 資料庫名稱，應與你提供的資料庫對應
});
```

圖:node. js

F. MySQL

提供了強大的數據庫建模、開發和管理功能。能夠輕鬆部署、檢查和優化數據庫。更有利的是其豐富的管理功能，如數據傳輸、備份等，保障了數據的安全與完整性。

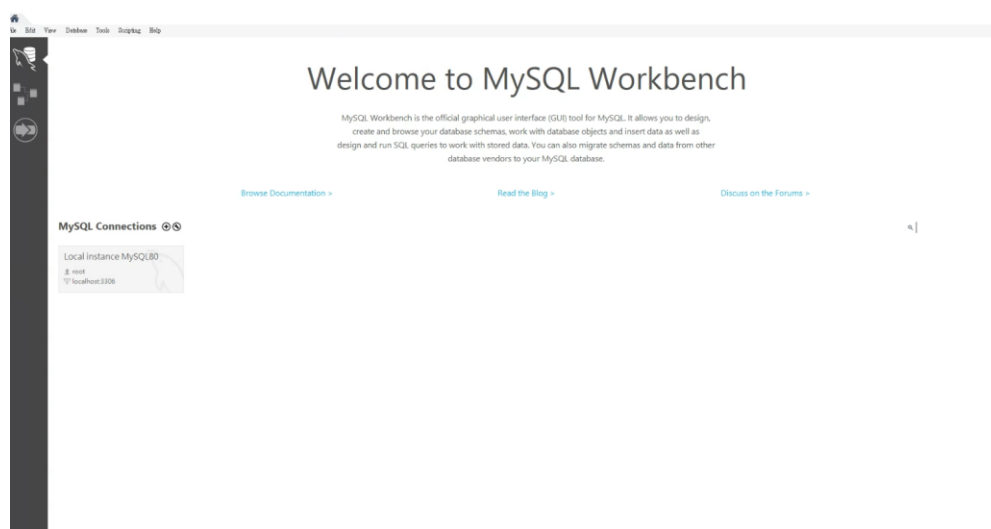


圖:MySQL 應用介面

	id	company_name	red_candy_box	blue_candy_box	pink_candy_box
▶	5	7-11	5	6	7
	6	全家	8	5	9
	7	OKMART	7	8	5
	8	7-11	2	2	2
⌵	HULL	HULL	HULL	HULL	HULL

圖:資料庫內容儲存

五、 成果展現

YOUTUBE連結: <https://youtu.be/fM3IyOYsecw>

六、結果與未來發展

結論

本專題成功實現了自動化訂單處理流程，結合了現代化的技術如 Node.js 伺服器、資料庫操作、YOLOv7 物件偵測模型，以及機械手臂的自動化控制，展示了以下關鍵成果：

1. 訂單到執行的自動化閉環

系統實現從網站接收訂單、將資料存入資料庫、執行物件偵測與結果傳輸、以及機械手臂抓取執行的全流程自動化，大幅提升了作業效率，減少人為操作的錯誤。

2. 技術整合與協作的成功應用

- YOLOv7 模型實現了準確的物件偵測，經過參數調整，模型能適應不同場景，達到高準確度。
- TCP Socket 保證了 YOLOv7 與機械手臂的高效通信，實現了設備間的即時數據交換。
- 多線程程序 確保多台機械手臂協同工作，避免衝突，提升了系統的穩定性與可靠性。

3. 降低自動化門檻

本專題充分利用開源技術（如 Python 和開源的 YOLOv7），顯著降低了自動化解決方案的實現成本，為中小型企業提供了一種可行的技術架構。

未來發展

1. 提升物件偵測的準確性與通用性

- 優化 YOLOv7 模型，加入更多場景與物件類型的訓練數據，提升在多樣化環境中的應用能力。
- 結合 3D 感測技術（如 LiDAR 或深度相機），實現更精確的物體定位與尺寸估算，應用於複雜場景。

2. 進一步強化機械手臂的智能化

- 引入強化學習（Reinforcement Learning）技術，使機械手臂具備自適應能力，能根據不同物體的形狀與材質動態調整抓取方式。
- 開發更智能的路徑規劃演算法，減少手臂運行的時間與能耗。

3. 擴展應用場景

- 將系統應用於其他領域，例如：
 - 電商倉庫的自動分揀。
 - 醫療行業的物資分配。
- 增強系統的可擴展性，支援更多種類的訂單和多樣化的任務需求。

4. 雲端整合與數據分析

- 將資料庫與雲端服務（如 AWS 或 Google Cloud）整合，實現跨地點的實時監控與管理。
- 利用數據分析技術，分析歷史訂單與操作數據，優化系統效能，預測需求，提升供應鏈效率。

5. 降低成本與提高可靠性

- 開發更簡化的硬體控制方案，減少非必要的硬體投入。
- 加強錯誤檢測與故障恢復機制，提升系統的穩定性與使用壽命。

透過不斷優化與創新，本專題的成果不僅可提升當前工業自動化的水準，還可作為基礎架構應用於更多場景，助力企業邁向智慧化與數位化的未來。