

二、研究計畫內容（以 10 頁為限）：

（一）摘要

近年因為極端氣候、全球暖化等現象的發生，人類的環保意識也跟著抬頭，其中再生資源便是一項備受關注的議題。再生資源的其中一個來源為舉凡廢電池、玻璃、塑膠等可回收再利用之垃圾，它們能夠重新被加工再使用，同時也能減少燃燒垃圾對環境的破壞。本計畫希望能提高資源垃圾的回收率，在既有的 YOLO 技術架構下，實行物件偵測，期望能找出最適合的方式進行模型的訓練，以便透過相機分辨資源回收物。接著使用訓練好的模型計算轉換，將目標物座標透過 ROS 傳送給機器手臂，再進一步調整並完善程式架構，使其能針對環境的變化做出適當的反應，完成夾取與放置於特定位置的動作。

（二）研究動機與研究問題

甲、研究動機

垃圾是人類文明的發展的副產物，與人類活動密不可分。然而，垃圾在自然界中容易破壞生態環境，被動物誤食使汙染物參雜於食物鏈中，而垃圾的處理亦耗費大量資源成本，同時燃燒垃圾排放的廢氣又會回歸到環境中，增加溫室效應加速全球暖化。

根據環境部全國一般廢棄物產生量統計[1]，2021 年全台一般廢棄物總量共 1104 萬 9062 噸，其中包含一般垃圾 38.7%，資源垃圾 56.3%；而 2022 年的全台一般廢棄物總量更是到達了 1123 萬 8654 噸，其中一般垃圾占了 42.7%而資源垃圾占了 52.9%，由此可見我國的垃圾量與一般垃圾占比正在增加中。然而我國的廢棄物處理設施有限，再加上近年公有焚化爐陸續歲修，垃圾的處理速度趕不上其增加的速度，導致一般廢棄物暫存量始終壓不下來，2022 年的暫存待處理一般廢棄物高達 75.2 萬噸，相較 2021 年增加了 6.4 萬噸[2]。不過一般垃圾中仍有一定比例的可回收再生資源垃圾，若能將一般垃圾減量，不僅能減少焚化爐的負擔，降低一般廢棄物暫存量，更能減少對環境的污染，實現資源再利用。

減少一般垃圾的方式，除了宣導一般民眾「主動」做好垃圾分類外，假想若能利用垃圾貯存的時間「被動」以機器挑取出可再被利用的資源垃圾，定能有效再將一般垃圾減量。

而智慧型機器人一直是一個具有發展性的主題，隨著世界各地機器人相關的技術發展與人力成本的提升，機器人進入人類的社會並參與服務的趨勢與需求逐漸提高；同時，智慧型機器人屬於高整合性的項目，集結控制、人工智慧、通訊等領域於一身，我想利用實驗室已有的協作機械手臂設備實際操作，學習相關知識與研究，最後將 Eye-to-Hand 自動資分類夾放技術實現。

乙、研究問題

我希望能將資收分類設定為本計畫協作機器人的任務，藉由取出一般垃圾中的可再利用資源，提高資源垃圾的回收率，同時幫助減少焚化爐燃燒一般垃圾的負擔。

本計畫欲使用 YOLO（You Only Look Once）物件偵測技術[3]，訓練模型偵測一般民眾易混淆為一般垃圾的可回收垃圾，並想辦法降低誤判物品的機率，接著再透過機器人作業系統 ROS（Robot Operating System）作為整合平台[4]，連結 Gazebo 模擬軟體[5]，模擬機器手臂操作的環境與情況並測試在不同環境中機器的運作成效，完善程式使機

器人能針對環境的變化做出適當的反應。最後再結合實驗室的協作機械手臂，透過 eye-to-hand 鏡頭實際操作，辨識一般垃圾中的可回收物並夾取，放置在指定位置。

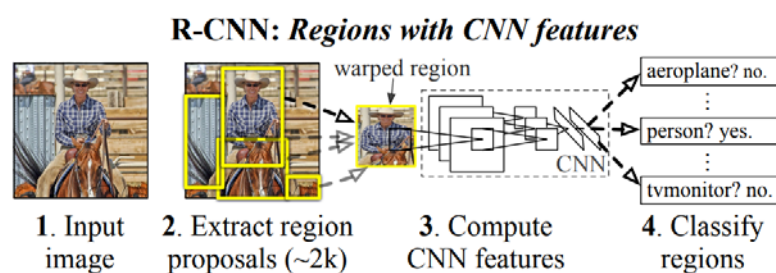
本計畫欲以體積小，容易被忽視，數量卻龐大的「瓶蓋」、「塑膠筆管」、「塑膠餐具」作為範本訓練，再將鏡頭拍攝的畫面藉由 YOLO 即時偵測，辨識出目標物的外型框架後，從框架的四個邊角求得夾取點資訊，再將座標校正，從拍攝畫面的像素座標系轉換成以機器手臂為中心的座標系，經過 ROS 將座標傳送給機器手臂，使其完成移動與夾取的動作。

(三)文獻回顧與探討

為達成本計畫的目標，應先認識使用的工具，若能參考其他文獻與資料認識其背後工作原理，定能使計畫更具意義，執行起來更順遂。

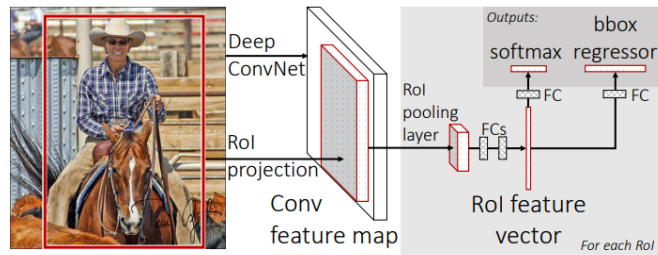
物件偵測是指在照片或影片等圖像內容中，以「框」標出有興趣的目標物件位置，並針對該物件進行分類並猜測機率。此技術可應用於工廠瑕疵檢測、人臉五官定位、醫療影像分析等各種智慧應用中。早期的物件偵測演算法以 two stage 為大宗，需要先用演算法生成樣本的候選框，再透過卷積神經網路做影像辨識，例如 R-CNN 便是 two stage 演算法的代表之一。

R-CNN[6]的運算方式為，在輸入的圖片中以 Selective Search 的方式先取出約 2 千個候選區域 (region proposals)，Selective Search 的理論概念是使用圖像分割 (segmentation) 後的結果，計算影像中所有相臨區域之間的相似性，然後把相似性高的區域合併起來。接著將這些區域使用卷積神經網路求得特徵集合 (feature vector)，再將其各自的特徵向量再丟入 SVM 分類器中去分類，並將候選區域送進 Bounding Box Regressor 中迴歸與基準真相 (ground truth) 的差異量，判斷當前區域所對應的實物是 background 還是所對應的物體類別，利用非極大抑制 (non-maximal suppression) 獲得最終的 bounding-box。



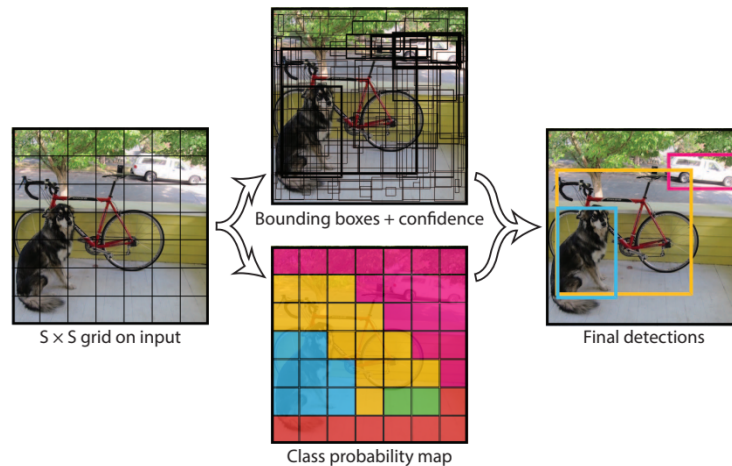
圖一、R-CNN 運作架構[6]

然而，R-CNN 將模型的訓練分成許多階段的方式非常沒有效率，再者，在訓練時還需先把所有候選區域的特徵集合都存到硬碟上，不僅佔用硬體資源，也耗時許多，於是便有了 Fast R-CNN [7]與 Faster R-CNN[8] 的出現。Fast R-CNN 與 Faster R-CNN 致力於利用運算共享以及神經網路來代替 Selective Search 找出候選區域，加速 R-CNN 上。Fast R-CNN 改變了候選區域的提取邏輯，將原本繁瑣的提取改為只對整張圖像全區域進行一次特徵提取，映射到卷積神經網路最後的 feature map 上才切出真正要拿來使用的 ROI(Region Of Interest)，加快了執行的速度；而 Faster R-CNN 運用 RPN(Region Proposals Network)搭配 anchor box 的設計提前定義區域，使執行速度有更明顯的提升。



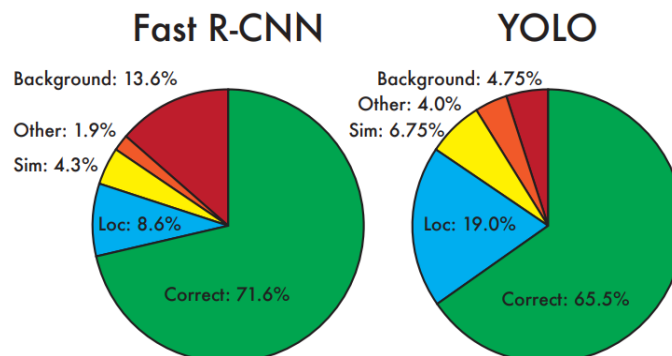
圖二、Fast R-CNN 運作架構[7]

在 2016 年，J. Redmon 等人發表了一篇關於 YOLO，一個嶄新的物件偵測方式的論文[9]。與以往使用分類器進行的 two stage 物件偵測演算法不同，YOLO 將物件偵測視為一個迴歸任務（regression problem），在空間中分割出邊界框（Bounding Box）並計算出其類別的機率（associate class probability）。YOLO 僅需要一個卷積神經網路進行一次計算便能同時進行多邊界框與其類別機率的預測，其運作方式為將輸入圖像內容切成 $s \times s$ 個單位網格，再同時進行在每個網格內預測邊界框和各自的信賴指數（confidence score）以及針對每個網格進行類別機率預測，最後利用非極大抑制取得最終的偵測結果。



圖三、YOLO 運作架構[9]

與 two stage 物件偵測需要先用演算法生成樣本的候選框，再透過卷積神經網路做影像辨識不同，YOLO 為 one stage 的物件偵測，物件類別和位置偵測及物件辨識一步到位，雖然精準度會略低於如 R-CNN 與其變體的 two stage 演算法，但 YOLO 提升的速度以及其降低背景誤判機率的特性，在現實考量上非常划算。再加上 YOLO 是開源且普遍被使用的模型，本身非常容易打造並且可以直接在整張圖像上訓練，其便成為了此計畫中物件偵測系統的理想選擇。



圖四、與 Fast R-CNN 相比，YOLO 大幅降低背景誤判的機率[9]

為了捕捉工作平台上的物件進行偵測，本計畫欲使用眼在外（eye-to-hand）方法，也就是將相機固定於機器手臂外，再透過校正取得機器人座標與相機座標之間的關係。在 1999 年的一篇論文[10]提出一種利用平面的棋盤方格進行校正的方法，接著在 2000 年，同一作者又發布了一篇論文[11]將此方法完整化，變作實用的嶄新校正方法。此校正法首先需採集以相機拍攝的二維方格棋盤各個位姿之圖片，再計算求得相機本身的內參（Intrinsic），也就是相機內部的各個參數，接著再放置校正用的棋盤方格於工作平面上並拍攝一張照片，透過照片中校正棋盤的位姿計算求得相機的外參（Extrinsic），也就是相機本身擺放相對於世界座標系旋轉平移的向量。

世界座標系為使用者定義的三維世界座標系，用以描述目標物在真實世界的位置，相機座標系（Camera coordinate system）為以相機為原點的座標系，幫助溝通世界座標系與圖像及像素座標系，兩者的單位皆為公尺(m)。圖像座標系（Image coordinate system）和像素座標系（Pixel coordinate system）都成像在平面上，只是各自的原點和單位不同：圖像座標系的原點在相機光軸與成像平面的焦點，單位為公釐(mm)；而像素座標系的原點在圖像左上角，像點（0，0）處，單位為 pixel。世界座標系可以透過乘上外參矩陣與內參矩陣得出其像素座標系，同理，藉由將像素座標乘上內參的反矩陣與外參的反矩陣，我們便可得知其世界座標系中的座標。最後再將此座標系平移為以機器人為中心的座標系，便能提供機器手臂夾取目標物的座標。此種校正法的優點在於，在已知使用同一台相機的前提下，每當相機或工作平台位置改變時，使用者只需重新拍一張校正用棋盤方格的圖片便能迅速重新校正。

求得目標物座標後，為了實現軟硬體之間的溝通，我們便需要使用到 Robot Operation System (ROS)。ROS 是一個專為機器人軟體開發所設計出來的開源作業系統框架[4][12]，為一種中介軟體提供類似作業系統的功能。ROS 可視作一個軟體框架，提供軟硬體溝通的基礎架構，並且也支援多種程式語言，降低了機器人的開發門檻，便於開發者專注於演算法與應用的研究。同時它也擁有眾多強力的開發工具，其功能模組都被封裝於獨立套件包（package）中，便於開發者共享，這使得 ROS 得以有效簡化開發機器人平台的複雜任務建立與穩定版本控制。另外 ROS 也整合了許多第三方工具，諸如 Gazebo 等常見開發工具，幫助開發者快速完成機器人應用的建立、設計與多機整合。

ROS 主要依靠中心化 P2P（Centralised Peer to peer）實作。中心化 P2P 的概念為使用一個中心服務器（Server）連結其他節點（Node）的訊息資料並對請求做出回應；而節點負責處理與發布訊息資料，具有 index 便於找到絕對地址。在 P2P 架構中，節點指的是程序（Process），一個機器人控制系統通常包含多個節點；Master 是一個特殊的節點，其作為 Server，負責查找節點並傳遞資料與溝通；訊息（Message）是節點之間傳遞的資料，為一種資料結構；Topic 是 message 發布時的標籤；發佈者（Publisher）為發布消息的節點，而訂閱者（Subscriber）則為接收消息的節點，節點會透過 topic 找尋需要接收的訊息，同一種 topic 的訊息可以由不同的節點發布或接收，因此存在複數個發佈者和訂閱者，或是一個節點同時為發佈者與訂閱者都是有可能的；套件包（package）為一個 ROS 軟體的基本單位，其包含許多節點與相關函式庫、資料集（data set）、配置文件（configuration file）或其他和專案有關的檔案。透過 ROS 我們便能實作機器人的操作。

不過，機器人要實機操作時，可能會因為各種硬體機構的問題而突然出

狀況，此外，也為了確保實行時的安全性，在確認演算法沒有問題前先在模擬器上測試會是相當安全且便利的方法。本計畫欲使用 Gazebo 作為模擬軟體，在實機運作前先行測試。Gazebo 是一套能真實模擬各種模型、物體的軟體，很適合用來實驗機器人如何與周遭環境互動，同時其也能夠與 ROS 連結，方便開發者將機器人資訊匯入並存取。而在實作方面，本計畫欲使用達明機器人公司的 TM5 常規負載機器人[13]作為實作，達明機器人亦提供在 Gazebo 上模擬所需要的資源檔案，方便模擬測試，最後再實際應用。



圖五、TM5 常規負載機器人[13]

(四)研究方法及步驟

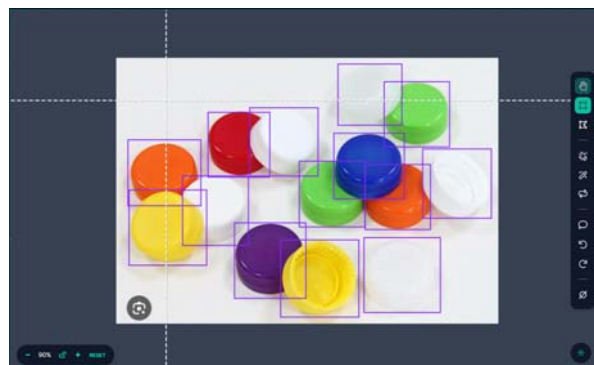
本計畫的步驟流程如圖六所示，主要分為「框定目標物件」、「訓練模型」、「建立環境」、「控制手臂移動與夾取」、「校正座標」、「程式整合」。以下將對各流程一一介紹。



圖六、計畫流程圖

甲、 框定目標物件

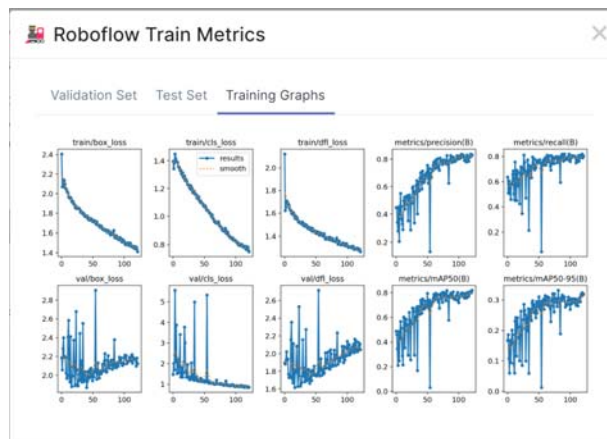
Roboflow 是一個線上框定工具，除了能夠線上繪製邊界框並進行分類標籤外，更能對圖像數據進行轉換增強，改善資料集的多樣性，提高模型的泛化能力，並訓練模型對模型進行評估與改善。我們將透過公開網站所提供圖片，以及自行拍攝來收集大量「瓶蓋」、「塑膠筆管」、「塑膠餐具」的圖片，應用在 Roboflow 上處理圖像，做成資料集。



圖七、框定物件示意圖

乙、 訓練模型

Roboflow 同樣可以支援模型的訓練，同時它可以將框定完成的圖片分為訓練用以及測試用，以便模型訓練完後測試實際成效。



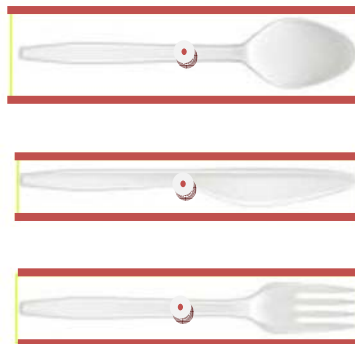
圖八、測試集的偵測狀況示意圖

丙、 建立環境

本計畫將會在 Linux 作業系統下作業。為確保測試時的安全性，避免各個開發環境彼此干擾，實作會使用 conda 建立虛擬環境，並在虛擬環境下執行。除此之外，也需要先建立 ROS，YOLO，Gazebo 等工具的環境。

丁、 控制手臂移動與夾取

各別先測試使用程式控制手臂移動位置與夾爪夾取。先藉由 YOLO 即時偵測辨識出目標物的外型框架後，將框架四個邊角的座標值計算求得平均，即可獲得目標物的中心點座標，也就是協作機器人的夾具需抵達的目標位置。至於夾具之角度則會透過計算外型框架的長短邊求得，利用畢氏定理計算各點間的距離，即可判斷出長短邊結果。最後，將夾具移動至中心點座標並將夾爪兩側對齊框架之長邊即為夾取點。



圖九、夾取點示意圖

得知夾取點後，本計畫將使用 ROS 為平台，傳輸目標資訊給協作機器人。ROS 提供了兩種網路通訊方式，分別為 TCP (Transmission Control Protocol) 協議與 UDP (User Datagram Protocol) 協議。UDP 的優點是速度快，但數據是否會被接收以及數據傳輸的順序無法被保證，相對的，雖然 TCP 速度較慢，但他具有安全可靠的優點，也能夠檢查與糾正錯誤，使數據確實且按照順序的被接收。為確保數據傳輸的準確性，本計畫將參考 TM ROS Driver[14]上的範例程式碼「send_script」，透過 TCP 協議進行數據的傳輸，藉由傳送座標控制手臂移動到目標位置。

當程式被呼叫時，會先建立一個傳輸座標用的節點 (node)，然後當發現傳輸座標的服務 (service) 後，創建一個服務客戶端 (client) 請求服務調用並輸入請求數據，請求的數據為欲移動至的座標。接著數據會透過 TM ROS Driver 內的 Service Server 提供的運動指令，對手臂接

收端節點「Listen node」下指令移動到指定位置，而若過程中出現問題，便將問題回報列印並終止程式運作。

控制手臂夾爪夾取的方式與移動手臂相仿，參考 TM ROS Driver 上的範例程式碼「set_io」，對 Listen node 發送數位或類比輸出值的指令。當程式被呼叫時，先透過指令將控制手臂夾爪的端子（pin）設為低電位以打開夾爪，接著等待兩秒，再將該端子設為高電位闔上夾爪，同樣的，若過程出錯程式會將問題回報列印並終止運作。

戊、校正座標

列印出方格長寬已知的棋盤方格，並選定相機從各個角度拍攝之，參考 GitHub 上的程式碼[15]，計算求得該相機內參。求得內參後將內參更新至程式中，再固定相機，拍攝同棋盤方格一起的工作平台，求得外參。將內參與外參相乘得到轉移矩陣 H ，給予任一個待轉換的二維像素座標 (u, v) ，其與世界座標系 (x, y, z) 之間的關係如下：

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = ZH^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = Z \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

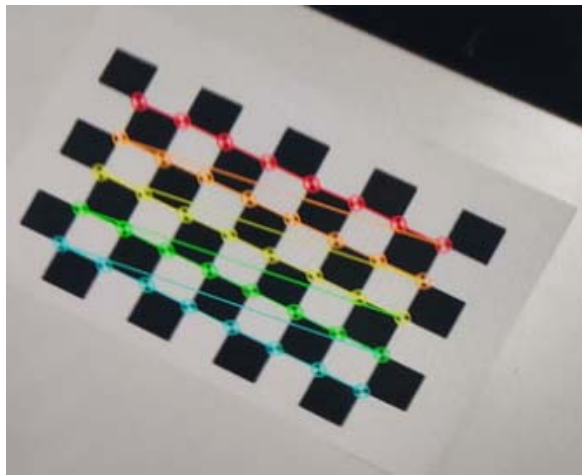
圖十、從像素座標轉換至世界座標

其中， Z 為尺度因子，可透過以下式子將之消去

$$\begin{cases} x = \frac{H_{11}u + H_{12}v + H_{13}}{H_{31}u + H_{32}v + H_{33}} \\ y = \frac{H_{21}u + H_{22}v + H_{23}}{H_{31}u + H_{32}v + H_{33}} \end{cases}$$

圖十一、消去尺度因子

最後求得的便是校正後的真實世界座標，將此座標單位轉換並加上手臂本身到像素座標 $(0, 0)$ 處所需的平移量，即為手臂待移動至的目標座標。



圖十二、棋盤校正示意圖

己、程式整合與改善

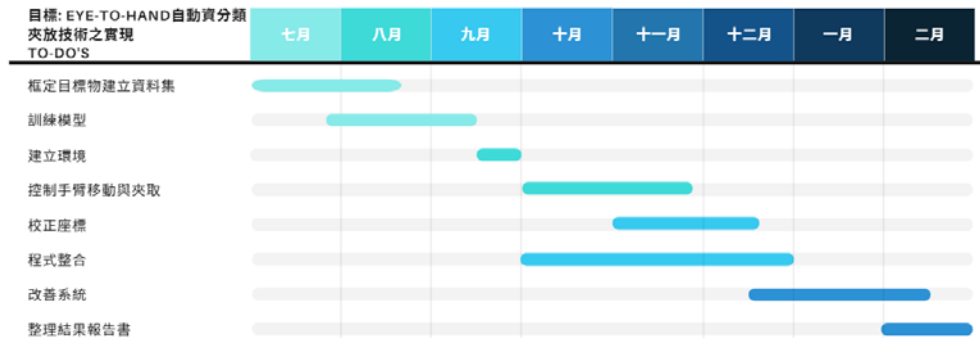
參考 YOLOv5 的物件偵測程式「detect.py」[16]，將之修改加入上述控制手臂的程式並整合，在 Gazebo 上測試執行，執行成功再於 TM5 機器人上實際執行。

機器手臂的操作流程如下圖所示，先開啟手臂，接著在終端（terminal）進入 ROS 的資料夾啟動 roscore，再開啟一個終端連結到手臂的 IP 位址，然後打開達明機器人提供的 GUI(Graphical User Interface) 介面確認手臂的連線狀況，最後進入虛擬環境中執行整合的程式。



圖十三、系統執行流程

確認機器人任務可以順利完成後，再回到 Gazebo 中模擬不同的環境，例如外在光源，工作平台的位置，或是物件擺放的方式，修正並將智慧型機器人的功能完善。



圖十四、計畫甘特圖

(五)預期結果

甲、短期目標

完成資源回收分類系統的實作：在擺放不同物件的工作平台上以相機成功辨識「瓶蓋」、「塑膠筆管」、「塑膠餐具」，從標定框的四個邊角計算出夾取點，並使用機械手臂陸續將其夾取，擺放至指定位置。

乙、長期目標

以軟體模擬不同的環境，增加任務完成的正確性與手臂夾取的準確性，完善智慧型機器人的功能並使其最有效率的執行。

(六)需要指導教授指導內容

甲、如何提升 YOLO 模型的偵測準確度

資料集的製作，例如樣本數、照片拍攝中的環境、物品擺放方式等都會影響物件偵測的結果，需要向老師請教如何才能使模型具有最佳的可靠性。

乙、TM5 機器人的操作

包含 ROS 的操作以及 TM5 設備的使用以及功能都是我們必須學習的。特別是 ROS 作為軟硬體之間的整合平台，如何以程式控制手臂移動完成任務會是需要被指導的地方。

丙、夾具夾取點的改良

一件長型的待取物，例如塑膠筆管和塑膠餐具，其重心所在位置其實不競相同。塑膠筆管最恰當的夾取位置會是在物體的中心點，然而塑膠湯匙或叉子等外型及重量分布不均勻的物件重心則會偏離中心點。倘若夾取物品時夾取點不理想，即會有夾取失敗的可能性，如何修正此點會是需要跟老師研究討論的地方。

(七)參考文獻

[1]環境部全國一般廢棄物產生量

https://data.moeenv.gov.tw/dataset/detail/STAT_P_126

[2]環境資訊中心

<https://e-info.org.tw/node/237224>

[3] Ultralytics YOLO

<https://www.ultralytics.com>

[4]ROS Wiki

<http://wiki.ros.org/>

[5]Gazebo

<https://gazebo.org/home>

[6] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 580-587, doi: 10.1109/CVPR.2014.81. keywords: {Proposals;Feature extraction;Training;Visualization;Object detection;Vectors;Support vector machines},

[7] R. Girshick, "Fast R-CNN," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 1440-1448, doi: 10.1109/ICCV.2015.169. keywords: {Training;Proposals;Feature extraction;Object detection;Pipelines;Computer architecture;Open source software},

[8] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017, doi: 10.1109/TPAMI.2016.2577031.

keywords: {Proposals;Object detection;Convolutional codes;Feature extraction;Search problems;Detectors;Training;Object detection;region proposal;convolutional neural network},

[9] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91. keywords: {Computer architecture;Microprocessors;Object detection;Training;Real-time systems;Neural networks;Pipelines},

[10] Zhengyou Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 1999, pp. 666-673 vol.1, doi: 10.1109/ICCV.1999.791289. keywords: {Cameras;Calibration;Computer vision;Layout;Lenses;Nonlinear distortion;Computer simulation;Testing;Voltage control;Robustness},

[11] Z. Zhang, "A flexible new technique for camera calibration," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330-1334, Nov. 2000, doi: 10.1109/34.888718.

keywords: {Cameras;Calibration;Computer vision;Layout;Lenses;Nonlinear distortion;Closed-form solution;Maximum likelihood estimation;Computer simulation;Testing},

[12] Quigley, Morgan, et al. "ROS: an open-source Robot Operating System." *ICRA workshop on open source software*. Vol. 3. No. 3.2. 2009.

[13]達明機器人公司

<https://www.tm-robot.com/zh-hant/>

[14]TM ROS Driver

https://github.com/TechmanRobotInc/tmr_ros1/tree/melodic

[15] Calibration_ZhangZhengyou_Method

<https://github.com/zhiyuanyou/Calibration-ZhangZhengyou-Method>

[16] ultralytics/yolov5
<https://github.com/ultralytics/yolov5/blob/master/detect.py>

表 C802