

機械手臂視覺抓取系統研發

Research and development of robotic arm visual
grasping system

專題學生： 吳維文

指導教授： 柯春旭 博士

義守大學
電機工程學系
專題研究報告

中華民國 112 年 9 月

摘要

本專題研發基於影像與深度辨識的六軸機器手臂視覺抓取系統，利用 Yolo 影像辨識、OpenCV 深度影像辨識和六軸機器手臂控制方法，實現自動化的物件抓取和移動。由於 Yolo 演算法具有高效率辨識物體類別與位置的優點，但無法直接辨識物體角度，而 OpenCV 則可以利用輪廓計算角度，但無法辨識物體類別，因此，本專題提出結合 Yolo 物體類別位置辨識和 OpenCV 角度計算的方法，達到物體類別、位置和角度的識別，進行手臂即時物體追蹤控制，並根據不同的場景和物體下進行實驗，測試系統性能和穩定性，實驗結果顯示，系統可以準確地識別和抓取不同類別和大小的物體，高效率完成視覺抓取任務。

目錄

摘要.....	i
目錄.....	ii
表目錄.....	iv
圖目錄.....	v
第一章 導論.....	1
1.1 研究背景.....	1
1.2 文獻回顧.....	4
1.3 研究目標.....	10
1.4 專題架構.....	10
第二章 機械手臂視覺抓取系統.....	12
2.1 物件辨識.....	12
2.1.1 Yolo	12
2.1.2 OpenCV	14
2.1.3 Yolo 結合 OpenCV	16
2.2 控制方法.....	18
第三章 系統實現.....	22
3.1 軟體介紹.....	22
3.2 硬體介紹.....	23
3.2.1 Intel RealSense D435.....	23
3.2.2 AR3 六軸機械手臂	25
第四章 實驗結果與討論.....	27
第五章 結論與未來展望.....	31

5.1	結論.....	31
5.2	未來展望.....	31
	參考文獻.....	32

表目錄

表 1.1	機器視覺方法之比較表 [17].....	7
表 2.1	G-code.....	20
表 2.2	AR3 DH model	21
表 3.1	畫面大小與 FPS 之最小深度比較表(單位：公分).....	24
表 4.1	辨識時間比較表(單位：秒).....	29

圖目錄

圖 1.1	2012~2022 年工業機器人年安裝量	1
圖 1.2	1959 至 2070 工作年齡人口推估圖	2
圖 1.3	工業 4.0 的手臂應用 (a)分揀血樣 (b)電池生產 (c)機器人輔助康復 (d)塑膠工業	2
圖 1.4	奧迪 A4 生產	3
圖 1.5	達文西機器手臂輔助微創手術數量	4
圖 1.6	帶有多旋翼和多自由度機械手臂正在抓取物體 [11]	5
圖 1.7	抓蘋果的過程 [12]	5
圖 1.8	移動機器人的六軸機械手臂詳細尺寸與實體圖 [13]	6
圖 1.9	GG-CNN [15]	6
圖 1.10	SSD、YOLO 模型比較 [16]	7
圖 1.11	Yolo 系列比較圖 [18]	8
圖 1.12	Yolo 偵測結果圖 [19]	8
圖 1.13	邊界矩形的類型 [20]	9
圖 1.14	系統架構圖 [22]	10
圖 2.1	機械手臂視覺抓取系統之流程圖	12
圖 2.2	訓練資料 (a)polytab (b)AA box (c)Lutein (d)Charge cable box (e)Converter 3.5mm (f)Foam	13
圖 2.3	Yolo 訓練結果	14
圖 2.4	影像座標轉換圖	15
圖 2.5	遮罩圖	16
圖 2.6	Yolo 結合 OpenCV 流程圖 (a)RGB 圖 (b)Yolo 運算結果 (c)深度圖 (d)裁切後小圖 (e) 遮罩圖 (f)使用 OpenCV 之小圖辨識結果 (g)辨識結果顯示於深度圖 (h)辨識結果顯示於 RGB 圖	17

圖 2.7 輪廓圖 (a)濾雜訊前 (b)濾雜訊後 (框框左上方為其面積，中心為中心點座標，單位像素，左下方為選轉角度)	18
圖 2.8 手臂追蹤物體流程圖	19
圖 2.9 實際手臂 (a)原歸零姿勢 (b)模擬運動學歸零姿勢	19
圖 2.10 正、逆運動學之推導	20
圖 3.1 系統架構圖	22
圖 3.2 操作介面	23
圖 3.3 Intel RealSense Depth Camera D435	23
圖 3.4 對齊前與對齊後比較圖 (a)對齊前-RGB 圖 (b)對齊後-RGB 圖 (c)對齊前-深度圖 (d)對齊後-深度圖 (e)對齊前-疊合圖 (f)對齊後-疊合圖	25
圖 3.5 AR3 六軸機械手臂 (a)手臂本體 (b)手臂控制箱	26
圖 4.1 雙物體抓取過程 (a)原始位置 (b)隨機移動墊板變動物體位置 (c)追蹤並移動鏡頭角度 (d)抓取物體 (e)回歸原始位置 (f)準備下次抓取 (g)再次隨機移動墊板變動物體位置 (h)追蹤並移動鏡頭角度 (i)抓取第二個物體 (j)完成抓取任務	28
圖 4.2 辨識藥品結果比較圖 (a)Yolo+OpenCV(框框上方為物體名稱及信心度，中間為中心點) (b)OpenCV(框框上方為其面積，中間為中心點，下方為選轉角度)	29
圖 4.3 實驗結果比較圖 (a)Yolo+OpenCV (b)OpenCV	30

第一章 導論

1.1 研究背景

隨著機器人技術和自動化能力增長，需求量與應用方面越來越多。根據國際機器人聯合會(IFR)的最新統計數據，如圖 1.1 所示，2022 年所有相關行業的裝機量都大幅增長，且全球工廠安裝的新工業機器人達到 531,060 台 [1]，創歷史新高，比 2018 年機器人安裝的大流行前記錄高出 25%，2012 至 2022 年，全球操作機器人的累計安裝達到約 350 萬台的新紀錄。

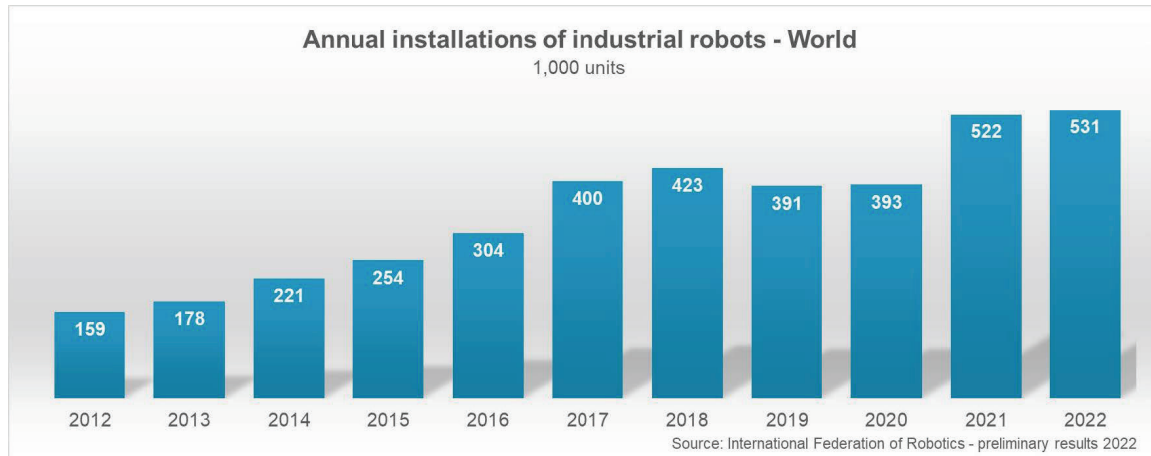


圖 1.1 2012~2022 年工業機器人年安裝量

(資料來源：國際機器人聯合會)

在工業 4.0 的時代，企業面臨人力不足 [2]，如圖 1.2 所示，和品質不良等問題，希望透過智慧工廠的方式來解決。然而，傳統自動化技術只能執行基本的任務，尚未達到能夠取代人類工作的水平，並且工業機器人手臂是當今最普遍使用的機器人類型之一 [3]，其可以提高生產效率、減少勞動強度、改善工作環境等。因此，需要將機械手臂與影像感測器等設備結合，以發揮其最大的效益。其技術在智慧家庭、醫療康復、農業、無人配送、無人巡檢等領域都有廣泛的應用 [4]，如圖 1.3 所示，同時在分揀血樣、復健、電池生產及塑膠工業

也皆有使用 [5]。

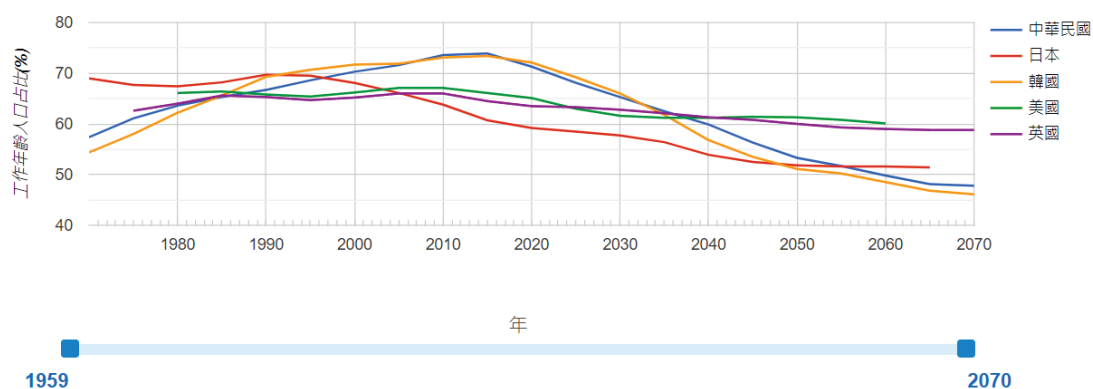
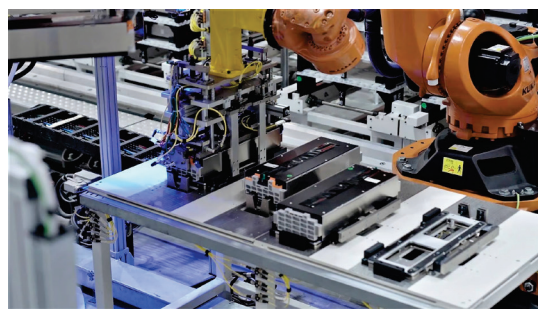


圖 1.2 1959 至 2070 工作年齡人口推估圖
(資料來源：國家發展委員會人口推估查詢系統)



(a) 分揀血樣



(b) 電池生產



(c) 機器人輔助康復



(d) 塑膠工業

圖 1.3 工業 4.0 的手臂應用 (a)分揀血樣 (b)電池生產 (c)機器人輔助康復 (d)塑膠工業

(資料來源：KUKA AG [5])

奧迪生產戰略/系統與技術開發負責人 Karl Unger 提到「作為生產地點，它不再是勞動力成本最低的地方，而是擁有最好的工業機器人和最熟練員工的國家 [6]」。

在奧迪的生產中，像是眾多底盤零件需要上磅數鎖緊的工作，也能靠多隻機械手臂同時進行，而手臂的細膩度還能模仿人手，執行不規則裝物品的拿取，不僅簡單的如電線的條狀物，連數顆螺帽也可透過吸附式機械手臂執行拿取或者是定位的工作 [7]，如圖 1.4 所示。



圖 1.4 奧迪 A4 生產

(資料來源：AUDI AG 奧迪股份公司 [8])

除此之外，在近二十年來，醫學方面有了一台非常先進的設備，它就是「達文西手術系統」，其不僅適用於各種專業病症，並且在最近幾年中使用量大幅增加，差距最高達 139%，如圖 1.5 所示，手術最常見的包括男性的前列腺癌及女性的微創子宮切除術 [9]，並且越來越多的用於心臟瓣膜修復和婦科、消化道等手術過程 [10]，應用非常廣泛，但卻因價格昂貴及學習門檻高，目前仍尚未全面普及，很多病患至今仍無法完全相信讓手臂透過全自動化的方式為自己手術，這讓我意識到了擁有一個優秀的交互界面的重要性。

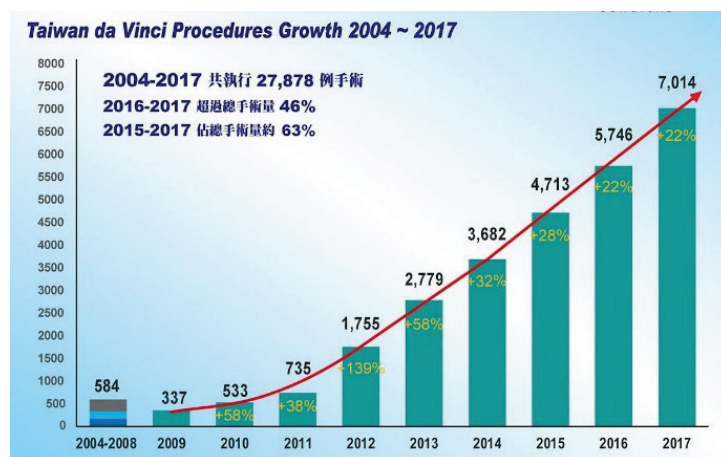


圖 1.5 達文西機器手臂輔助微創手術數量
(資料來源：臺灣楓城泌尿學會)

為了實現這樣的目標，影像辨識是相當關鍵的技術之一，然而，傳統的影像辨識技術容易受到光線強弱、折射等外界因素的干擾，因此，使用深度影像及深度學習的方式可以改善這些問題。即使在沒有燈光的夜晚，也能精準地辨識物體。同時，更多複雜任務所需的動作需要使用者來操控，因此，發展人機互動介面也扮演著重要角色，在發展影像辨識的同時加入人機介面，這樣的功能會讓使用上更加實用和方便。

1.2 文獻回顧

文獻 [11]提出了一種基於手臂視覺的空中操縱方法，它可以讓一個帶有機器人手臂的多旋翼飛行器，在未知環境中自動地抓取和放置物件，如圖 1.6 所示，該圖之右上圖顯示了安裝在末端執行器的魚眼鏡頭檢測到的物體，其利用深度學習和優化算法來訓練一個抓取策略模型，該模型可以根據手臂視覺的輸出，來生成抓取動作和飛行器控制，並且有提到相機坐標系之轉換。文獻 [12]同樣也提出一個抓取策略模型，其通過 sobel 邊緣偵測與影像分割等，達成能夠抓取不同形式的物體，包含圓形物體及條紋物體，並且還講述了相機標定、座標空間變換等手臂運動之相關知識，其過程如圖 1.7 所示。

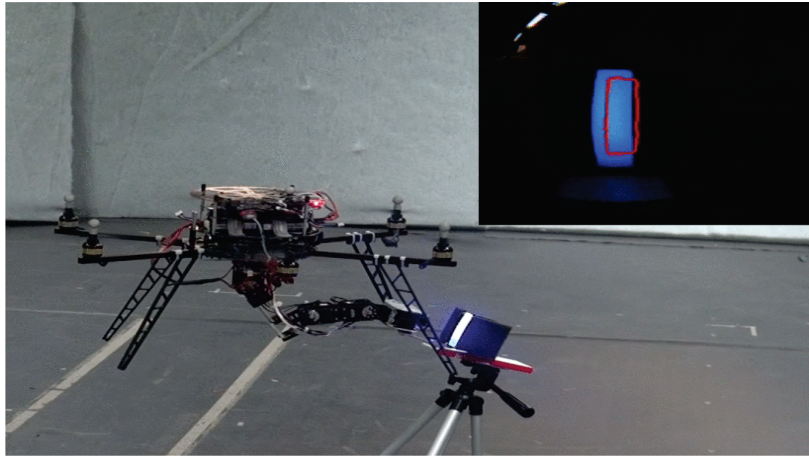


圖 1.6 帶有多旋翼和多自由度機械手臂正在抓取物體 [11]

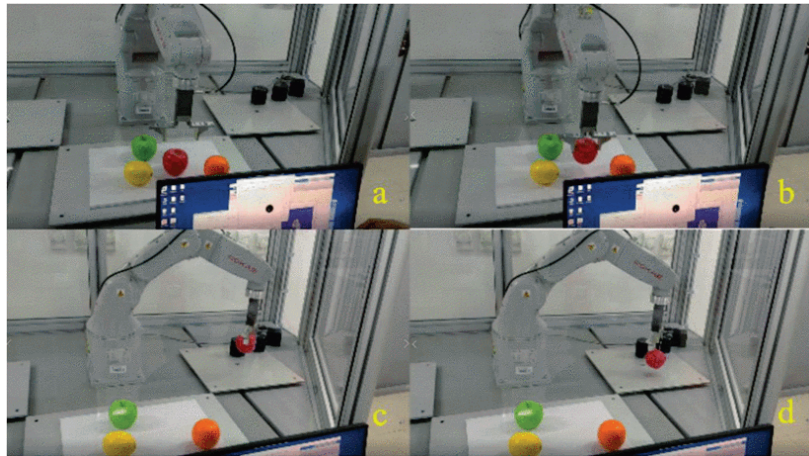


圖 1.7 抓蘋果的過程 [12]

文獻 [13]提出一款麥克納姆輪之家庭式移動機器人，並基於 ROS 作為核心，完成 SLAM、避障、聽從語音指令與自製的六軸機械手臂來取物，如圖 1.8 所示。文獻 [14]提出一種基於模糊邏輯或神經網絡的自適應控制器，利用視覺資訊來控制參數，實現路徑跟蹤和避免碰撞。

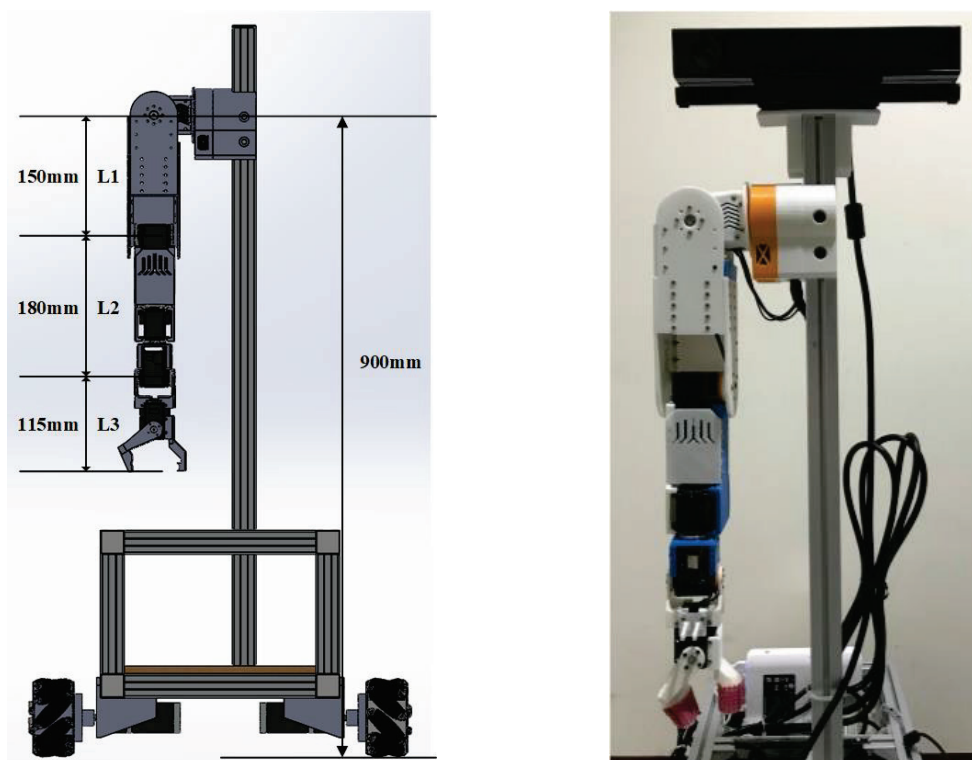


圖 1.8 移動機器人的六軸機械手臂詳細尺寸與實體圖 [13]

同時為了發展一個更優良的抓取系統，因此，本專題透過深度學習的方式，如 r-cnn、SSD、Yolo 或 GG-CNN [15]，而 GG-CNN 從提供的深度圖中生成抓取質量。根據最佳抓取姿勢的位置（由位置、角度和抓取寬度參數化），最後向機器人發出速度命令，如圖 1.9 所示。文獻 [16]中提到了 SSD 及 Yolo 的比較，如圖 1.10 可知 Yolo 演算法相較於其他的演算法在卷積層後接全連接層，並且不論是在速度、準確率和失誤率都相較於其他 r-cnn 來的更加優秀，因此由該演算法學習物體以利夾取。

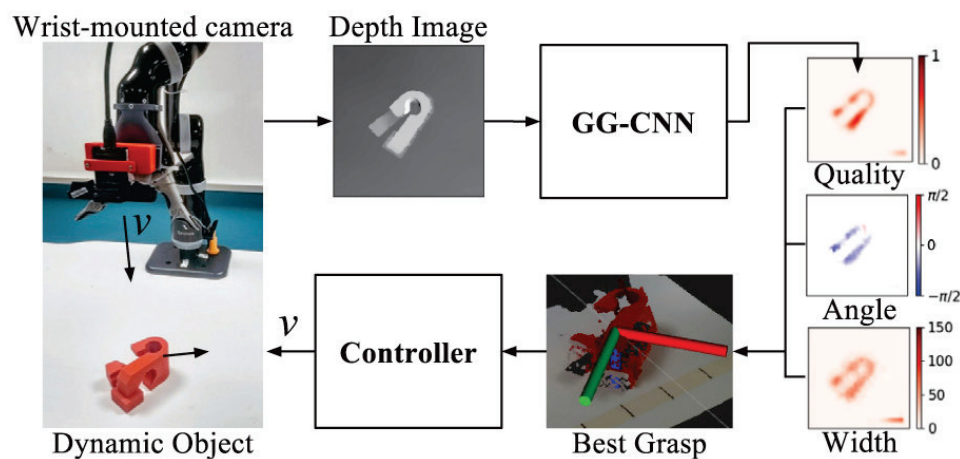


圖 1.9 GG-CNN [15]

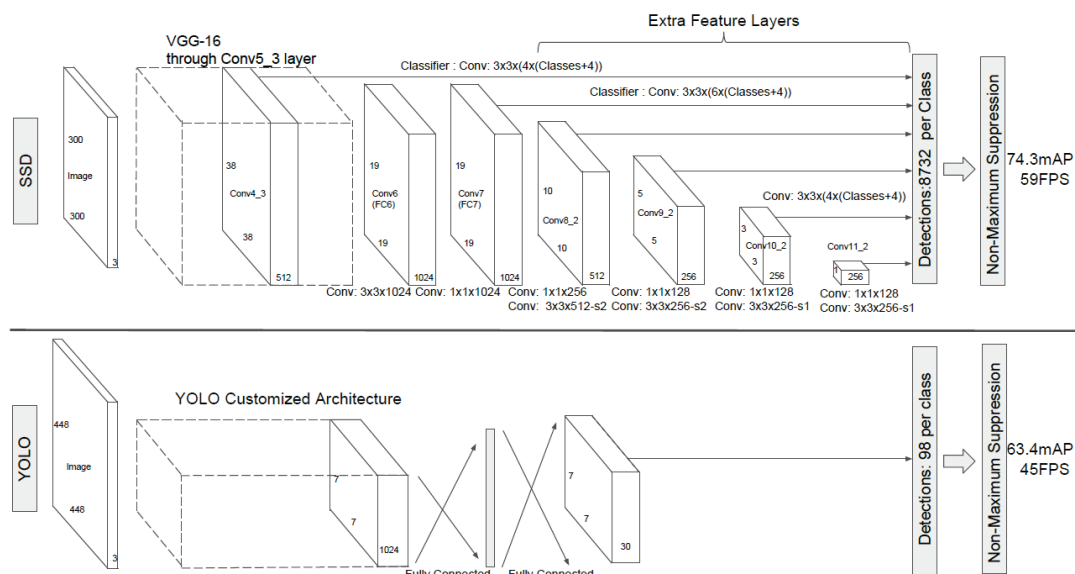
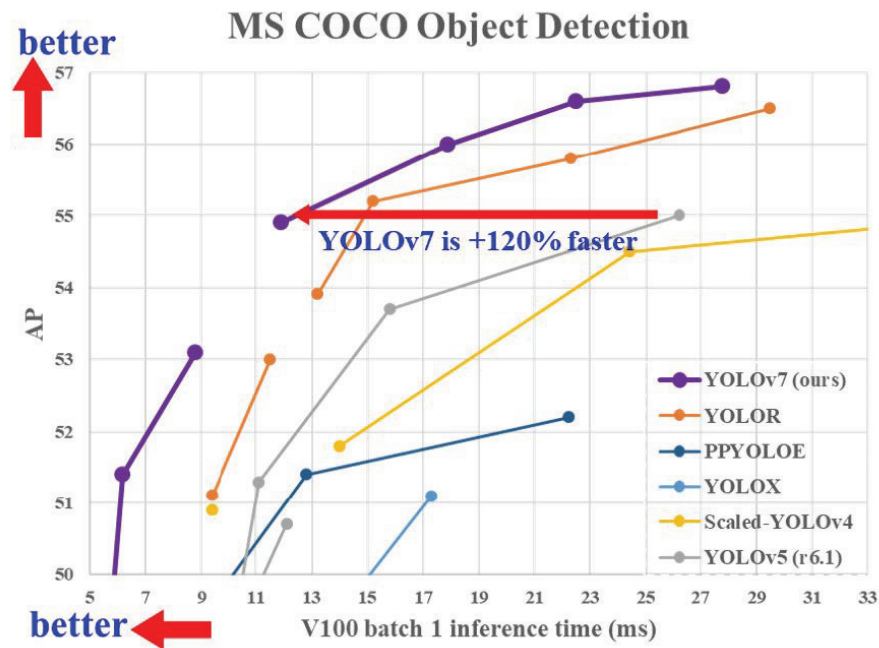


圖 1.10 SSD、YOLO 模型比較 [16]

從文獻 [17]可以知道精度和效率比較結果(表 1.1),Yolo v3 在平均精度(mAP)、每秒幀數 (FPS) 獲得了最好的效果，然而，時至今日 Yolo 已經發展許多版本，在 Yolo 系列比較圖(圖 1.11 [18])中可以發現 YoloV7 無論是在精準度及效率相較於 Yolo 的其他版本上也都有較好的表現。另外，YoloV7 的速度比基於 Transformer 的檢測器 SWIN L Cascade-Mask R-CNN 的速度高出 509%，準確度高出 2%，並且基於卷積的檢測器 ConvNeXt-XL Cascade-Mask R-CNN 速度提高了 551%，準確度提高了 0.7%。

表 1.1 機器視覺方法之比較表 [17]

	Faster R-CNN	YOLO v3	SSD
mAP (GF-1& GF-2)	86.0%	90.4%	84.9%
mAP (GF-1)	64.0%	73.0%	60.9%
mAP (GF-2)	88.3%	93.2%	87.9%
FPS	12	73	35



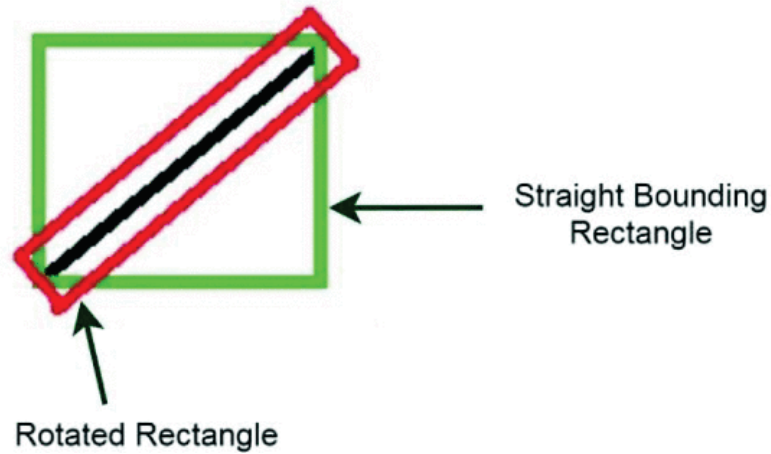


圖 1.13 邊界矩形的類型 [20]

從上述的文獻中，我們可以知道 Yolo 與 OpenCV 之間的優缺點，因此，本專題在研究 Yolo 與 OpenCV 的互相搭配，同時也了解人機互動的重要性，可以透過該介面讓我們的生活更加的便捷，因此，本專題在 Yolo 與 OpenCV 的互相搭配深入研究。然而，我們還需要建構一個完善人機系統，使其更加的便利。文獻 [22]提出了一種基於手臂視覺的機器人手臂介面，其架構如圖 1.14 所示，它可以讓有上肢活動障礙的操作者通過頭部運動來控制機器手臂，該系統使用了一個六軸機器手臂和一個頭戴式攝影機來實現手臂視覺，並設計了一個基於姿勢和眨眼的交互模式，讓操作者可以選擇物件、調整姿勢和執行動作。

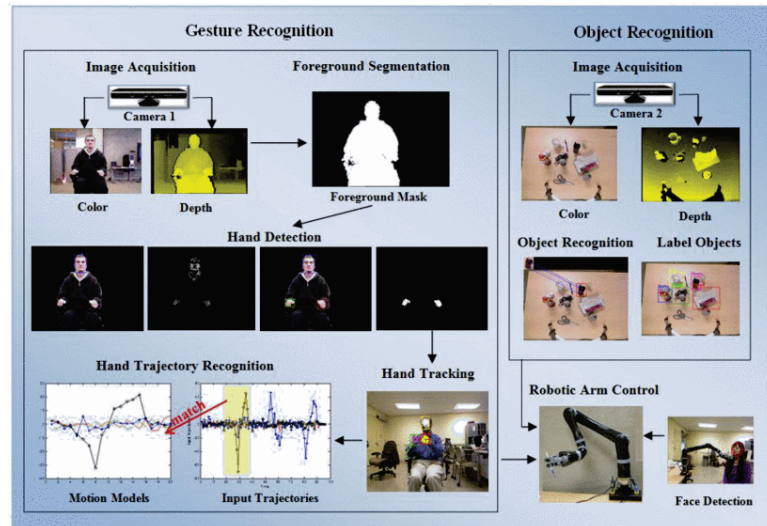


圖 1.14 系統架構圖 [22]

1.3 研究目標

目前標準的 Yolo 演算法，雖然能透過訓練的方式找到物體的名稱，但卻只能框出物體的位置，而無法算出角度，如圖 1.12 所示，然而，若是使用 OpenCV 的話，雖然能透過輪廓算出其角度，但弊病也非常明顯，就是無法辨識物體的類別。因此，本專題使用 Yolo 結合 OpenCV 的方式，同時解決取得物體名稱及辨識物體之角度的問題，並使手臂能實現即時動態追蹤，機器手臂即可協助使用者取得所欲拿取之物體，如可應用於生活中的藥品取放，或使工廠在進行瑕疵檢測時，能擁有更快的效率及準確率。

1.4 專題架構

本專題在第二章介紹手臂視覺抓取系統的概念，包括 Yolo、OpenCV 及深度等影像辨識之方法，並說明手臂控制的方法。第三章詳細說明了系統的硬體配置、軟體架構、影像處理、物件偵測及手臂控制等方法，並展示了系統的流程圖和介面設計。第四章為實驗結果與討論，在不同的場景和物件下，測試了系統的性能和穩定

性。最後第五章總結了本研究的主要貢獻和創新點，並提出了一些未來可以改進和延伸的方向和建議。

第二章 機械手臂視覺抓取系統

機械手臂視覺抓取系統是一種結合影像辨識和機器手臂控制的技術，透過取得空間中目標物體之座標，可以實現自動化的物件抓取和移動，並且符合物件擺放姿態，同時也可以隨時更改物體位置。本系統將 Yolo 影像辨識、深度影像辨識和六軸機器手臂控制方法整合起來，實現了自動化的追蹤物件抓取和移動。具體來說，系統首先使用 Yolo 來偵測待抓取物件的位置和大小，然後使用深度影像來估測其形狀和姿態，最後使用六軸機器手臂來實現即時的物件抓取和移動。

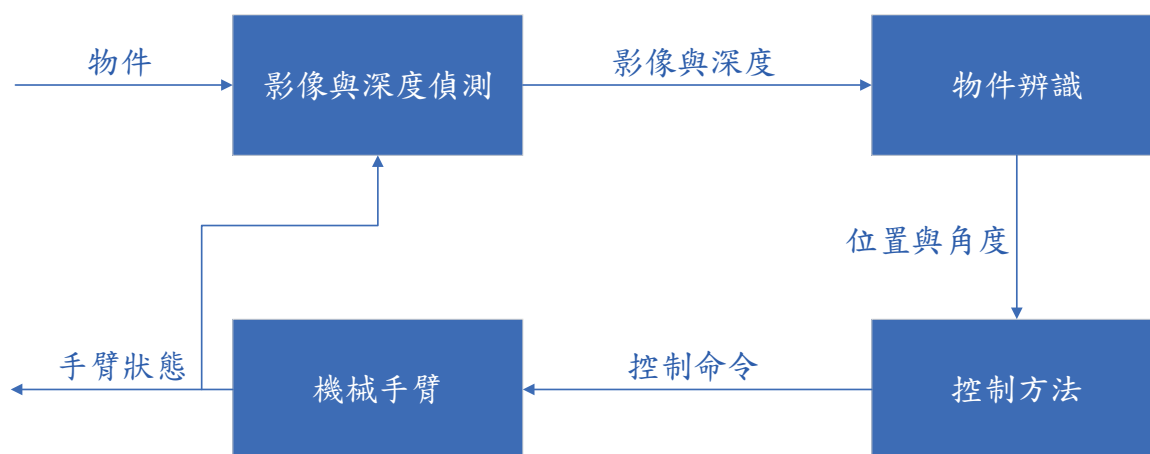


圖 2.1 機械手臂視覺抓取系統之流程圖

2.1 物件辨識

2.1.1 Yolo

You Only Look Once (YOLO)，是一種先進的實時物件偵測演算法，可以實現高效、準確的物

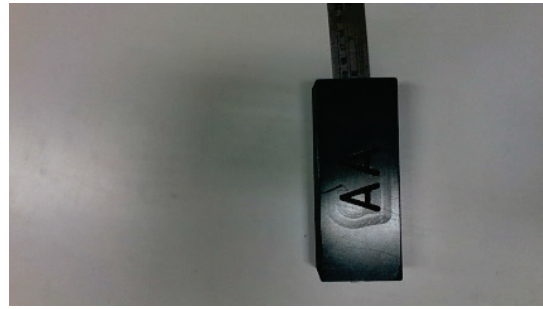
件偵測和定位。此次模型本專題使用 549 張照片和共 6 個標籤訓練而成，如

圖 2.2 所示，其中訓練集有 358 張與驗證集 191 張，共訓練五個多小時。透過 Yolo 訓練後的模型(圖 2.3)來看，在閾值為 0.5 時在訓練 200 次左右，精度已經接近 1，然而，閾值若要達到 0.95 的話，那精度只有 0.9 左右，因此，本專題訓練次數為 1000

次，以提高模型的精度。



(a) polytab



(b) AA box



(c) Lutein



(d) Charge cable box



(e) Converter 3.5mm



(f) Foam

圖 2.2 訓練資料 (a)polytab (b)AA box (c)Lutein (d)Charge cable box (e)Converter 3.5mm (f)Foam

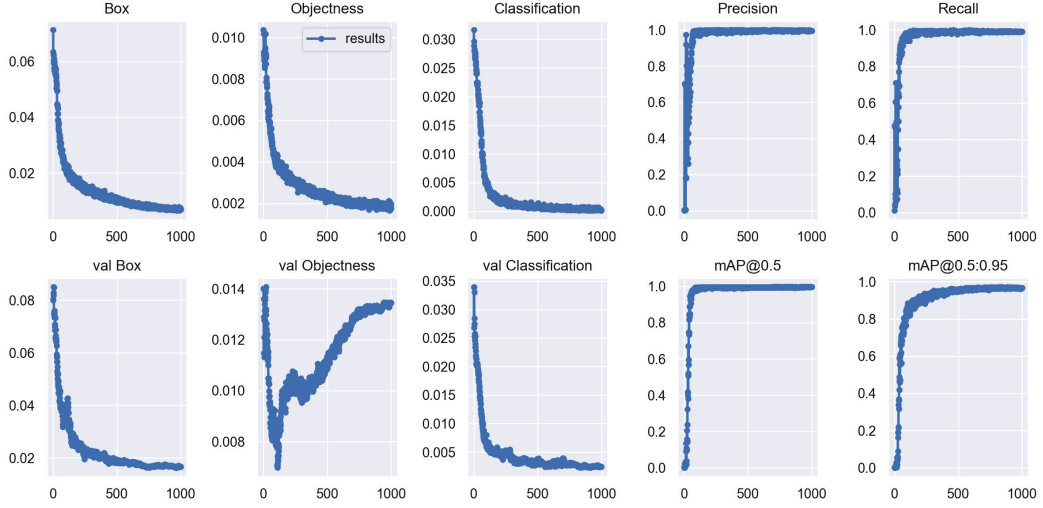


圖 2.3 Yolo 訓練結果

2.1.2 OpenCV

透過 Intel RealSense D435 深度攝影機取得深度資料及 RGB 資料，同時將其對齊，取得其相機參數中的圖像長度(h)、寬度(w)及焦距(f)，並透過(2.1)式及(2.2)式分別計算出對齊後之水平視野(HFOV)及垂直視野(VFOV)。

$$HFOV = 2 \tan^{-1} \left(\frac{w}{2f} \right) \quad (2.1)$$

$$VFOV = 2 \tan^{-1} \left(\frac{h}{2f} \right) \quad (2.2)$$

接著透過圖 2.4 推導出(2.3)式及(2.4)式， r_x 及 r_y 分別為該寬度及長度每像素的長度， n_x 及 n_y 為該方向之總像素，最後取得中心點深度(d)帶入(2.3)式及(2.4)式分別計算出 r_x 及 r_y 。

$$r_x = \frac{d \cdot \tan \left(\frac{HFOV}{2} \right)}{n_x} \cdot 2 \quad (2.3)$$

$$r_y = \frac{d \cdot \tan \left(\frac{VFOV}{2} \right)}{n_y} \cdot 2 \quad (2.4)$$

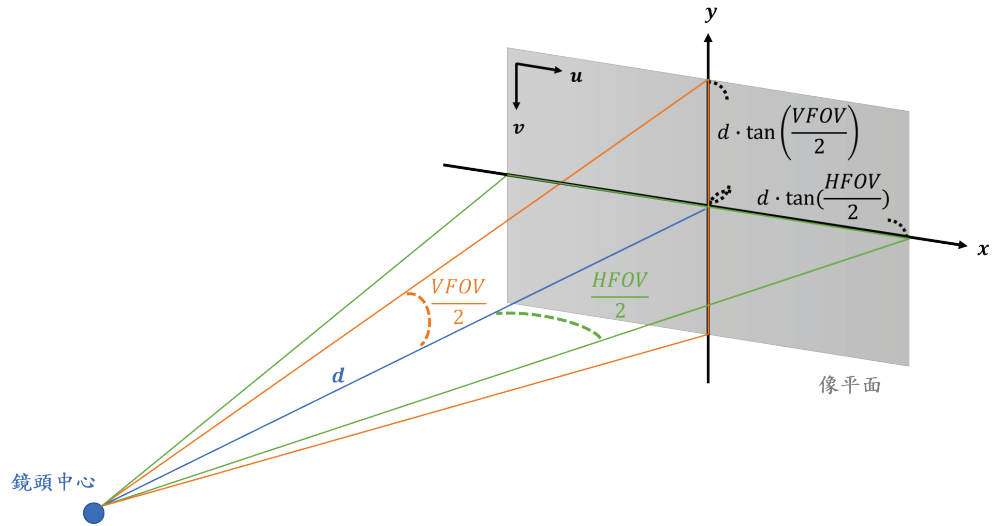


圖 2.4 影像座標轉換圖

當偵測到物體的影像後，可取得該物體之像素點分別為 u_o 、 v_o ，並透過(2.5)式及(2.6) 式計算後，即可得到該物體距離畫面中心點之相對位置，分別為 x_o 及 y_o 。

$$x_o = u_o \cdot r_x \quad (2.5)$$

$$y_o = -v_o \cdot r_y \quad (2.6)$$

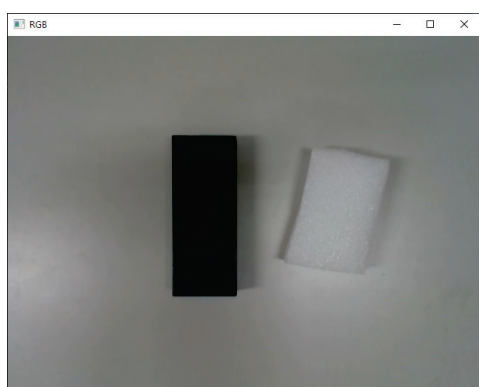
接著透過上述計算出來的深度，計算出物體所在的像素，並透過 OpenCV 將其內容繪製成黑白遮罩圖(圖 2.5)，再透過其遮罩圈出物體輪廓，透過輪廓即可算出物體中心點座標(像素)、該座標之深度及真實長寬。將上下左右各一個像素之深度，來校正鏡頭與桌面平行，並校正攝影機與夾爪之距離，包含長寬高，算出當下物體的中心位置、角度及深度。



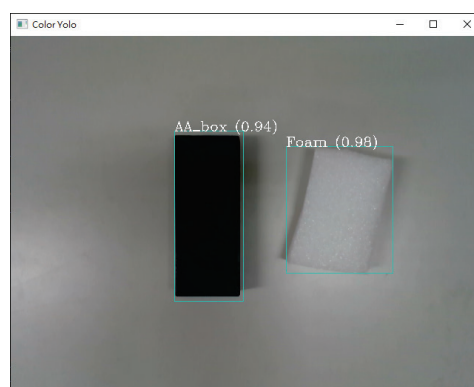
圖 2.5 遮罩圖

2.1.3 Yolo 結合 OpenCV

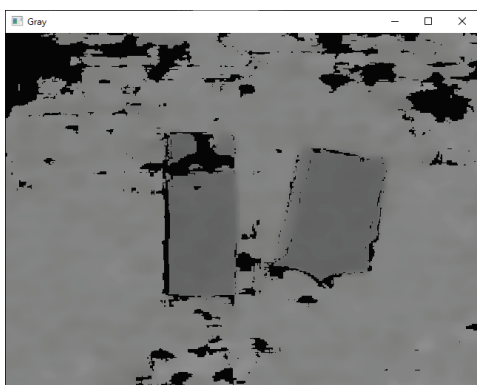
本系統使用訓練好的模型偵測圖 2.6(a)中的物體，並得到圖 2.6(b)的結果，物體名稱及其信心度和該物體的左上及右下之座標。然後根據物體的座標，將深度影像(圖 2.6(c))切割成多個小圖(圖 2.6(d))，以降低運算量。接下來，將裁切圖二值化，得到遮罩圖(圖 2.6(e))，並用 OpenCV 計算其輪廓(圖 2.6(f))。最後，將遮罩的中心點加上校正值(圖 2.6(g))，並用 OpenCV 得到物體的輪廓、位置、角度和深度(圖 2.6(h))。



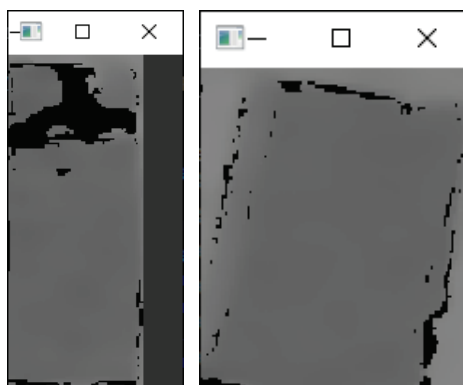
(a) RGB 圖



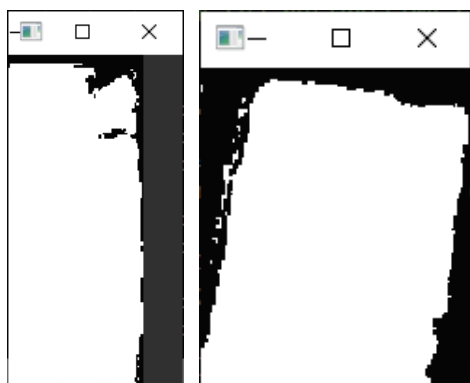
(b) Yolo 運算結果



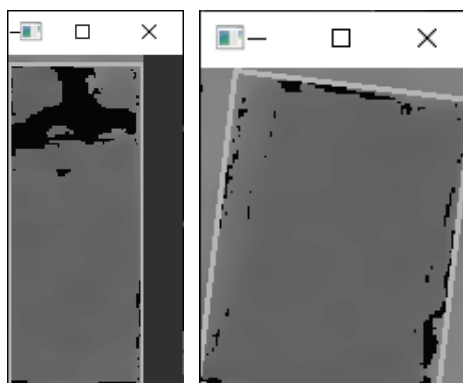
(c) 深度圖



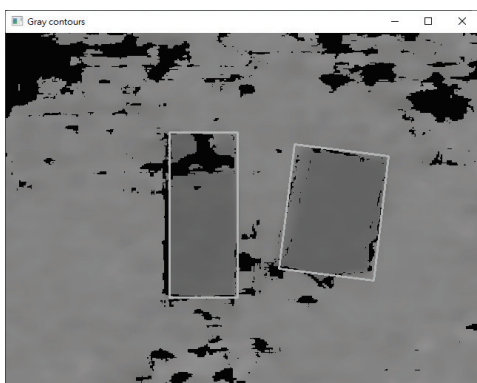
(d) 裁切圖



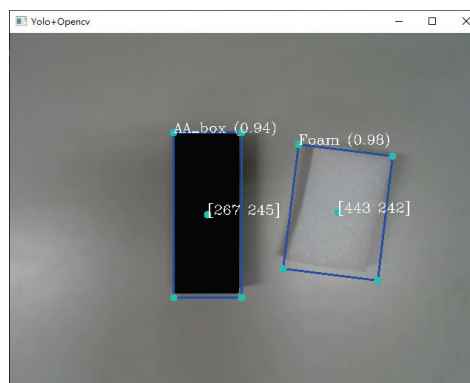
(e) 遮罩圖



(f) 使用 OpenCV 之裁切圖辨識結果



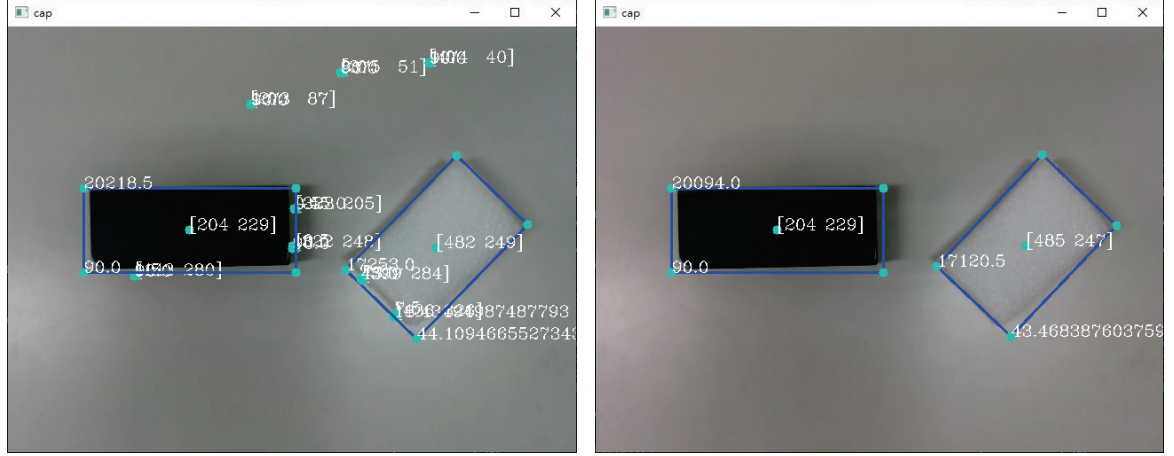
(g) 辨識結果顯示於深度圖



(h) 辨識結果顯示於 RGB 圖

圖 2.6 Yolo 結合 OpenCV 流程圖 (a)RGB 圖 (b)Yolo 運算結果 (c)深度圖 (d)裁切後小圖 (e) 遮罩圖 (f)使用 OpenCV 之小圖辨識結果 (g)辨識結果顯示於深度圖 (h) 辨識結果顯示於 RGB 圖

然而，由物體輪廓(圖 2.7)透過 OpenCV 所計算出來的角度還須經過轉換，首先要辨識物體長寬，若長小於寬的時候，就將角度加上 90 度，因為我們只抓寬度最小的邊，接著如果角度的絕對值大於 90 度，我們就先計算該角度轉了幾次 180 度，最後將該角度減去若干次的 180 度，但若該角度小於 0 度，則加上若干次的 180 度。



(a) 濾雜訊前

(b) 濾雜訊後

圖 2.7 輪廓圖 (a)濾雜訊前 (b)濾雜訊後 (框框左上方為其面積，中心為中心點座標，單位像素，左下方為選轉角度)

2.2 控制方法

透過 2.1 之方法可分別取得物體的中心位置、角度及深度，以上三種參數來實現手臂之物體的追蹤，本專題使用距離比例控制以達成手臂流暢的即時操控，將 x 軸距離、y 軸距離，z 軸距離和物體之度數，分別透過(2.7)式至(2.10)式算出每一次要位移的比例值，其中 z 軸之 v_z 為等速運動。

$$\begin{aligned}\dot{x} &= -k(x - x_T) \\ x_n &= x - k(x - x_T)dt\end{aligned}\tag{2.7}$$

$$\begin{aligned}\dot{y} &= -k(y - y_T) \\ y_n &= y - k(y - y_T)dt\end{aligned}\tag{2.8}$$

$$\dot{z} = -v_z\tag{2.9}$$

$$\begin{aligned}\dot{\theta} &= -k(\theta - \theta_T) \\ \theta_n &= \theta - k(\theta - \theta_T)dt\end{aligned}\tag{2.10}$$

我們先透過攝影機偵測，攝影機座標之目標相對距離與旋轉角度，接著計算目

前攝影機角度、相對距離與旋轉角度之比例值，並設定機器人座標可操作範圍之旋轉角度，再來我們還需要將算出來的比例值加上當前位置計算移動至下一次末端點位置，並計算移動時間及轉速，最後命令手臂移動，除此之外，如果目前末端點與物體之垂直距離小於 15 公分將會執行抓取動作，但若末端點尚未抵達物體上空，則會維持高度，直到末端點抵達物體上空為止。

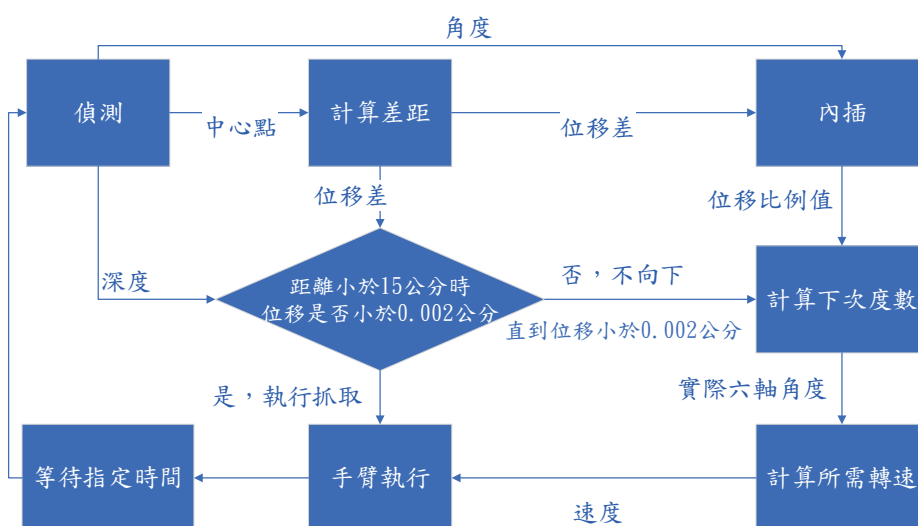


圖 2.8 手臂追蹤物體流程圖

接著透過 Python 的 serial 套件選擇連接埠連接手臂，配合 serial code 與表 2.1 的 G-code 設定手臂，先將真實手臂歸零以校正真實手臂的歸零姿勢，如圖 2.9(a)所示，之後將其校正成運動學歸零姿勢，如圖 2.9(b)所示，最後將三維座標透過逆向運動學算出六軸之角度，並取得當前六軸角度透過正運動學算出當前座標。



(a) 原歸零姿勢



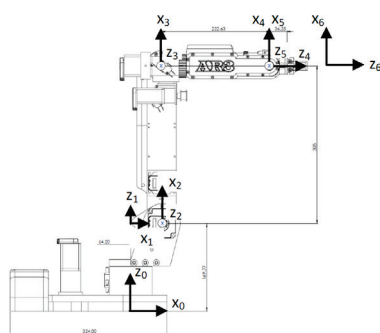
(b) 模擬運動學歸零姿勢

圖 2.9 實際手臂 (a)原歸零姿勢 (b)模擬運動學歸零姿勢

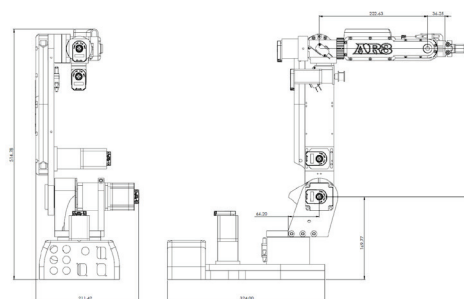
表 2.1 G-code

Code	Detail
G90	絕對位置
G91	相對位置

在推導正、逆運動學前，首先要定義一個六軸角度皆為零的姿勢，此姿勢為圖 2.10(a)，因歸零姿勢與實際手臂歸零不同，所以需先校正歸零姿勢，接著透過其歸零姿勢及圖 2.10(b)的關節長度等該手臂之訊息，如表 2.2 中推導出此手臂之正、逆向運動學。使用方式為輸入一個由三維座標及參考圖 2.10(a)中的第六軸座標(例： x_6 、 z_6)相對於原點座標(例： x_0 、 z_0)之方向的 4x4 矩陣，即可獲得基於此歸零姿勢的六軸角度。



(a)



(b)

圖 2.10 正、逆運動學之推導

表 2.2 AR3 DH model

軸	θ (度)	α (度)	a (公尺)	d (公尺)
1	0	0	0	0.16977
2	-90	-90	0.0642	0
3	0	0	0.305	0
4	0	-90	0	0.22263
5	0	90	0	0
6	0	-90	0	0.15025

第三章 系統實現

圖 3.1 系統架構圖是本系統的整體設計圖，說明了本系統的主要元件和功能。本系統由深度攝影機、電腦及六軸機器手臂組成，深度攝影機將拍攝目標物件的 RGB 影像和深度影像，傳送到電腦中，並於電腦中執行物件偵測、深度影像處理、運動規劃等演算法，並將相關資訊傳送到六軸機器手臂中。

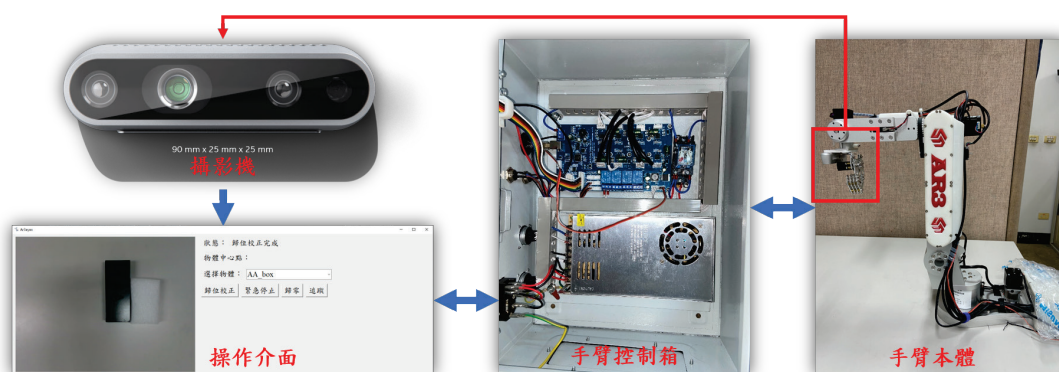


圖 3.1 系統架構圖

3.1 軟體介紹

本系統使用 Python 語言搭配 VSCode 編輯器開發，並結合 Anaconda 來管理環境，主要使用了 OpenCV、PyTorch 等函式庫來實現影像辨識和 Yolo 功能，同時使用 Tkinter 設計操作介面(圖 3.2)，在介面上可完成相對應的操作，並且為了讓所有功能能同時執行，因此採用多執行緒之方法。



圖 3.2 操作介面

3.2 硬體介紹

本系統使用深度攝影機和六軸機器手臂實現物件的抓取和移動。為了提高物件操作的效率和準確性，本系統還利用影像辨識和深度影像辨識的技術，來調整機器手臂的運動軌跡和速度。

3.2.1 Intel RealSense D435

RealSense D435 是 Intel 公司出的深度攝影機，如圖 3.3 所示，透過此深度攝影機來獲取 RGB 影像和深度影像，並使用 RealSense 官網的套件 pyrealsense2 進行 RGB 與深度影像的對齊，同時取得該攝影機之資訊。



圖 3.3 Intel RealSense Depth Camera D435

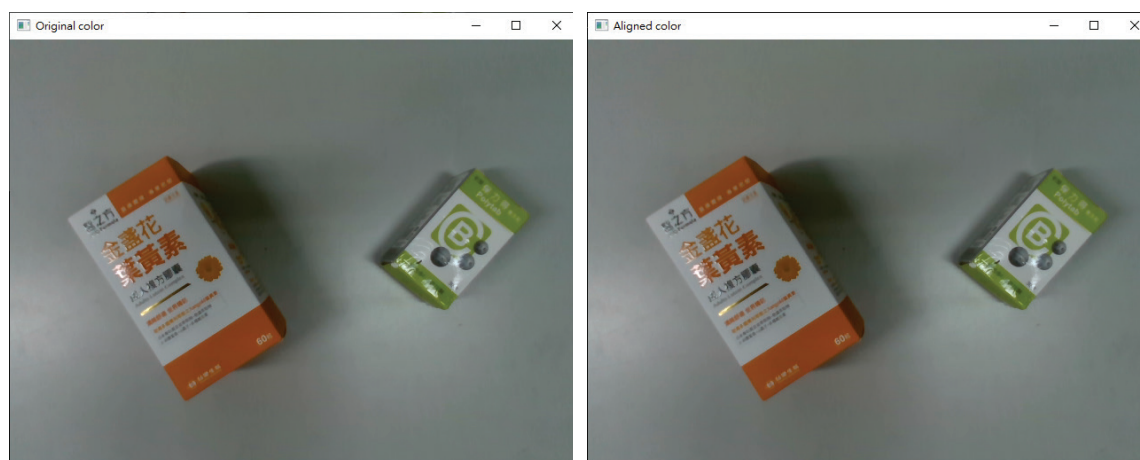
畫面大小為了配合 Yolo 的限制，需為 32 的倍數，因 stride 等於 32，同時 RGB 畫面要與深度畫面同大小，並且參考表 4.1，在 640*480 幀數為 90 情況下，擁有最

小深度較近，還可以加強畫面效率，綜合上述考量，所以決定使用 90 幀，畫面大小為 640*480。

表 3.1 畫面大小與 FPS 之最小深度比較表(單位：公分)

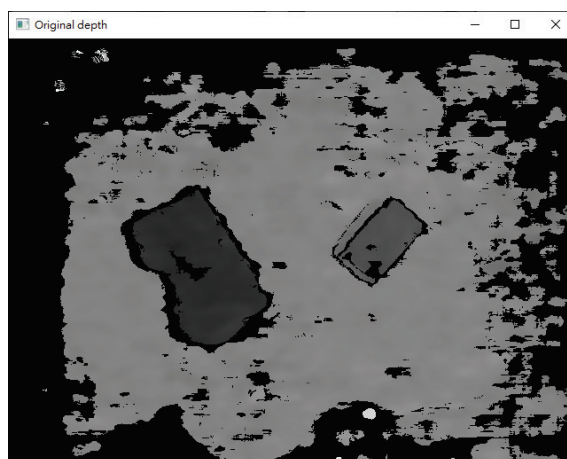
SIZE \ FPS	15	30	90
640*480	15.8	18	15
480*270	10.5	12	10
424*240	8.4		

本系統使用深度攝影機和六軸機器手臂實現物件的抓取和移動。由於 RGB 鏡頭和深度鏡頭的視野和解析度不同，導致 RGB 影像和深度影像有差異，如圖 3.4(a)(c)。為了解決這個問題，我們使用 pyrealsense2 中的函數將兩種影像對齊，得到圖 3.4(b)(d)。這樣可以提高物件偵測的準確度，並減少後續運算的複雜度。圖 3.4(e)(f)展示了對齊前後的差異。

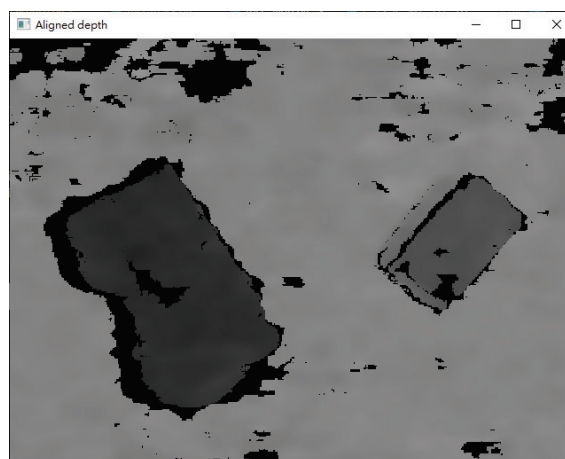


(a) 對齊前 - RGB 圖

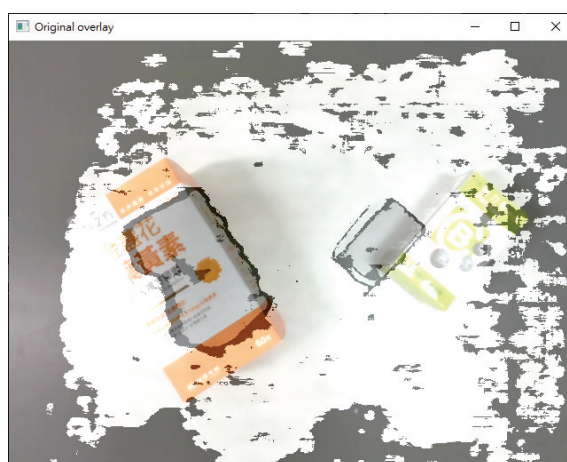
(b) 對齊後 - RGB 圖



(c) 對齊前 - 深度圖



(d) 對齊後 - 深度圖



(e) 對齊前 - 疊合圖

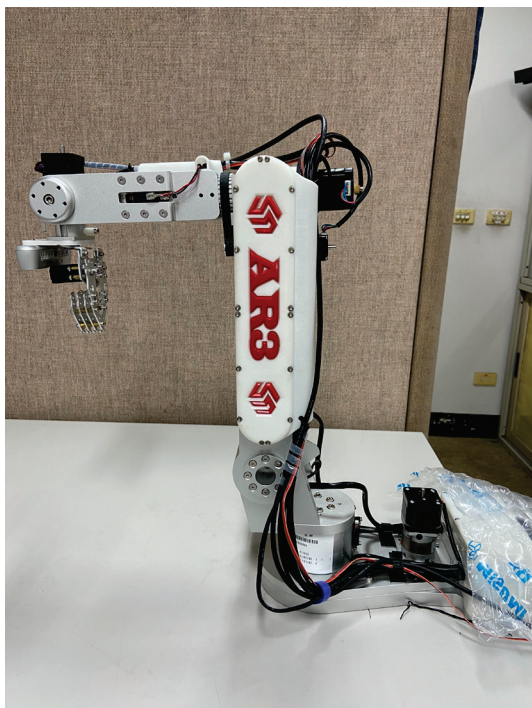


(f) 對齊後 - 疊合圖

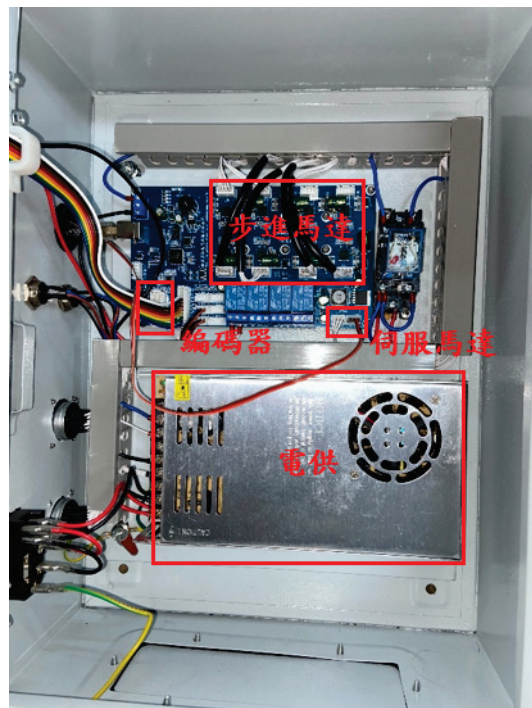
圖 3.4 對齊前與對齊後比較圖 (a)對齊前-RGB 圖 (b)對齊後-RGB 圖 (c)對齊前-深度圖 (d)對齊後-深度圖 (e)對齊前-疊合圖 (f)對齊後-疊合圖

3.2.2 AR3 六軸機械手臂

六軸機器手臂是一種高效、靈活及精確的工業機器人，它能夠實現多自由度運動和高精度定位，適用於各種複雜的工業任務。本系統採用了 AR3 六軸機器手臂(圖 3.5)作為本體，並配備了一個 Swaytail 伺服馬達夾爪，以實現物件的快速抓取和準確移動。



(a) 手臂本體



(b) 手臂控制箱

圖 3.5 AR3 六軸機械手臂 (a)手臂本體 (b)手臂控制箱

第四章 實驗結果與討論

我們對本系統進行了多次實驗，測試了不同種類、大小、形狀的物件。實驗結果顯示，本系統可以實現高效、準確及穩定的物件即時追蹤與抓取，且可以精準判斷雙物體，實驗過程如圖 4.1 所示。



(a) 原始位置



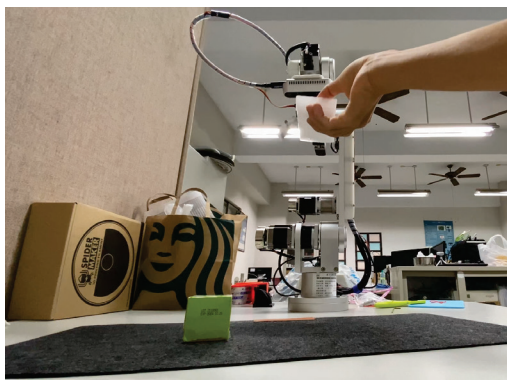
(b) 隨機移動墊板變動物體位置



(c) 追蹤並移動鏡頭角度



(d) 抓取物體



(e) 手臂回歸原始位置



(f) 準備下次抓取



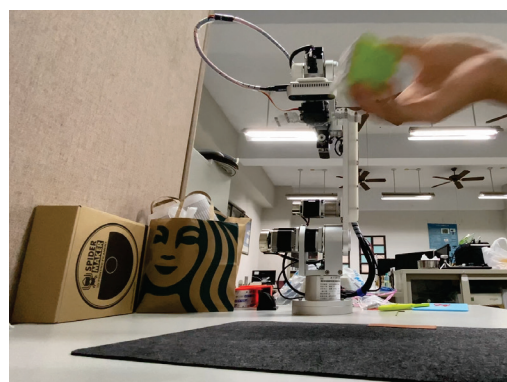
(g) 再次隨機移動墊板變動物體位置



(h) 追蹤並移動鏡頭角度



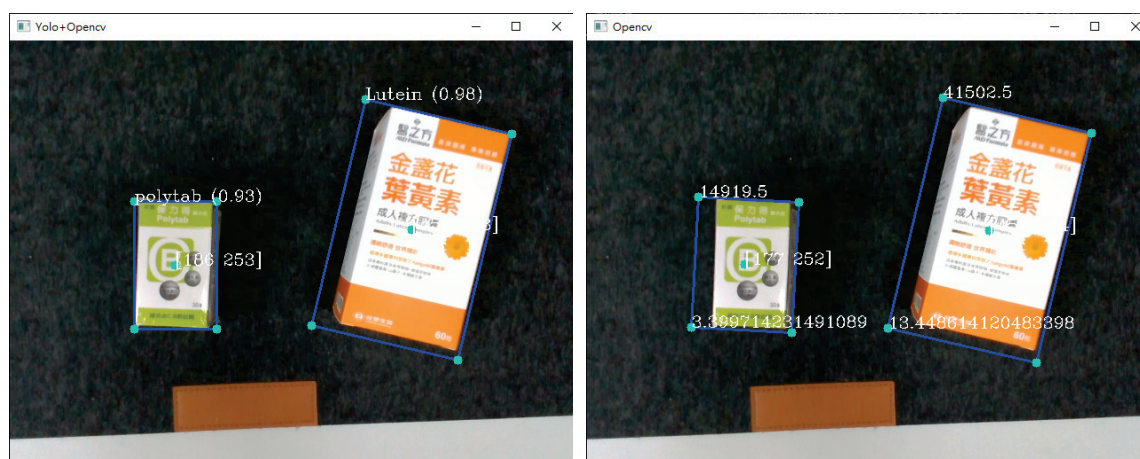
(i) 抓取第二個物體



(j) 完成抓取任務

圖 4.1 雙物體抓取過程 (a)原始位置 (b)隨機移動墊板變動物體位置 (c)追蹤並移動鏡頭角度 (d)抓取物體 (e)回歸原始位置 (f)準備下次抓取 (g)再次隨機移動墊板變動物體位置 (h)追蹤並移動鏡頭角度 (i)抓取第二個物體 (j)完成抓取任務

通過圖 4.2 可以模擬出藥廠透過手臂取放藥品的情境，若只使用 OpenCV 只能辨識出其面積與角度，但若使用 Yolo+OpenCV 的方式，則可以辨識出物體的名稱及信心度，只要將信心度提高，就能提升其準確度，這不僅能減少人力，還能降低醫護人員失誤拿藥的機率。



(a) Yolo+OpenCV

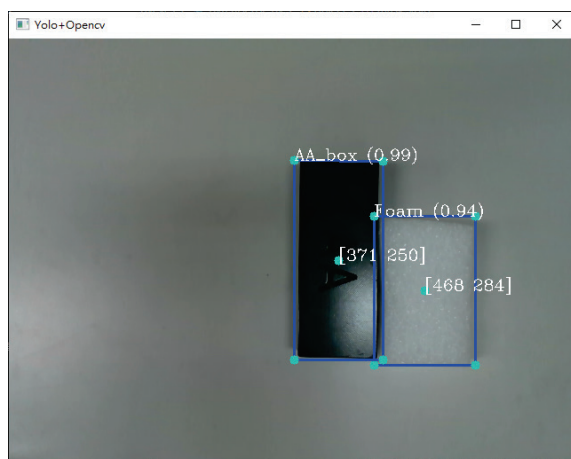
(b) OpenCV

圖 4.2 辨識藥品結果比較圖 (a) Yolo+OpenCV(框框上方為物體名稱及信心度，中間為中心點) (b) OpenCV(框框上方為其面積，中間為中心點，下方為選轉角度)

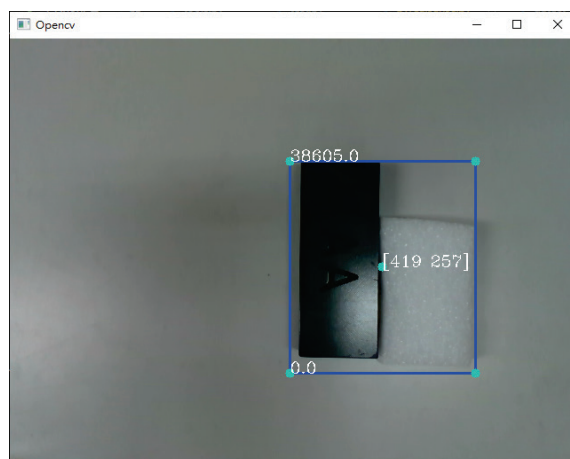
然而，在比較 Yolo+OpenCV 和 OpenCV 的辨識時間時，卻是使用 OpenCV 較為快速，如表 4.1 所示，原因在於 OpenCV 是以 C 語言為底層，所以速度較快，而 Yolo 的部分運算是由 Python 執行。雖然 OpenCV 速度較快，但是當物件相鄰或重疊時，使用 Yolo+OpenCV 可以成功辨識出個別物體，OpenCV 則會視為一個物體，如圖 4.3 所示。

表 4.1 辨識時間比較表(單位：秒)

物體數目	Yolo+OpenCV	OpenCV
1 個	0.093	0.022
2 個	0.087	0.023
3 個	0.089	0.026



(a) Yolo+OpenCV



(b) OpenCV

圖 4.3 實驗結果比較圖 (a)Yolo+OpenCV (b)OpenCV

第五章 結論與未來展望

5.1 結論

本專題提出了一個機械手臂視覺抓取系統，利用 Yolo 影像辨識、深度影像辨識和六軸機器手臂控制方法，實現了自動化的物件抓取和移動，同時使用 Tkinter 設計操作介面，提供了簡單易用的人機互動方式，讓使用者可以方便地設定參數、選擇物體和取得當前狀態。

本系統使用 Yolo 影像辨識方法，可以快速而準確地識別出不同種類、大小、形狀及姿態的物件，並根據物件的特徵選擇合適的抓取策略，同時利用深度影像辨識方法，測量出物件的角度、距離，長寬和高度等物體資訊，提高了抓取的精確度和成功率。

5.2 未來展望

在實際應用中，本系統還須加強一些功能。例如，對於複雜形狀與細小物件，可能無法正確識別出物件或者產生誤判。未來可透過增加更多種類的物件資料庫，以提高 Yolo 影像辨識的涵蓋範圍和鑑別能力，並研究更多種類的控制策略，如模糊控制、神經網路控制等，以增加對物件的抓取穩定性和移動平穩性。

參考文獻

- [1] International Federation of Robotics (IFR), "Building Resilience for Europe through Automation," *IFR Executive Roundtable on 28 June 2023*, 28 6 2023.
- [2] 國家發展委員會, "國際比較- 人口推估統計查詢系統," 22 8 2022. [Online]. Available: <https://pop-proj.ndc.gov.tw/international.aspx?uid=69&pid=60>.
- [3] Intel, "工業機器人手臂：改變工作方式," [Online]. Available: <https://www.intel.com.tw/content/www/tw/zh/robotics/robotic-arm.html>.
- [4] 農業科技決策資訊平台, "遠端控制機器人 吸睛," 11 3 2021. [Online]. Available: <https://agritech-foresight.atrri.org.tw/article/contents/3478>.
- [5] KUKA AG , "KUKA 的自動化解決方案," 2023. [Online]. Available: <https://www.kuka.com/zh-tw/%E8%A1%8C%E6%A5%AD/%E8%A7%A3%E6%B1%BA%E6%96%B9%E6%A1%88%E8%B3%87%E6%96%99%E5%BA%AB?&filters=EE8A90CEAE6D45B3BCC8C290B637ABF0|AB912E9473314B439CE9A5A468A5E663>.
- [6] AUDI AG, "Humans & Machines," [Online]. Available: <https://www.audi.com/en/careers/working-world/humans-machines.html>.
- [7] carstuff, "Audi 展示最新 Smart Factory 智慧工廠 人力越來越不值錢了!," [Online]. Available: <https://www.carstuff.com.tw/car-news/item/21541-audi-smart-factory.html>.
- [8] AUDI AG, "Production of the Audi A4: Assembly – “Marriage”: joining the body and the drive unit together," 28 11 2019. [Online]. Available: <https://www.audi-mediacycenter.com/en/photos/detail/audi-site-ingolstadt-88245>.
- [9] Kristen Gerencher, MarketWatch, "Robots as surgical enablers," marketwatch, 3 2

2005. [Online]. Available: https://www.marketwatch.com/story/a-fascinating-visit-to-a-high-tech-operating-room?dist=msr_2.
- [10] Wikipedia, "da Vinci Surgical System," [Online]. Available: https://en.wikipedia.org/wiki/Da_Vinci_Surgical_System.
- [11] Suseong Kim, Hoseong Seo, Seungwon Choi and H. Jin Kim, "Vision-Guided Aerial Manipulation Using a Multirotor With a Robotic Arm," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 4, pp. 1912-1923, 29 1 2016.
- [12] Haichao Liu, Zhitai Liu, Huanli Liu 且 Weiyang Lin, "Research on Robot Visual Grabbing Based on Mechanism Analysis," *2021 IEEE 11th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, 10 11 2021.
- [13] 劉威德, "基於人工智慧與機器人作業系統可影像辨識自主拿," 7 108.
- [14] A. Koivo, R. Lewczyk and T.-H. Chiu, "Adaptive path control of a manipulator with visual information," *Proceedings. 1984 IEEE International Conference on Robotics and Automation*, 6 1 2003.
- [15] P. C. a. J. L. Douglas Morrison, "Learning robust, real-time, reactive robotic grasping," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 183-201, 3 2020.
- [16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, "SSD: Single Shot MultiBox Detector," 2016.
- [17] M. Li, Z. Zhang, L. Lei, X. Wang, and X. Guo, "Agricultural Greenhouses Detection in High-Resolution Satellite Images Based on Convolutional Neural Networks: Comparison of Faster R-CNN, YOLO v3 and SSD', *Sensors (Basel)*," *Sensors (Basel)*, vol. 20, no. 17, Aug. 2020.

- [18] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao、Institute of Information Science, Academia Sinica, Taiwan, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object” .
- [19] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
- [20] Sakshi; Vinay Kukreja, “Segmentation and Contour Detection for handwritten mathematical expressions using OpenCV,” *2022 International Conference on Decision Aid Sciences and Applications (DASA)*, 2022.
- [21] Mauricio Marengoni; Denise Stringhini, “High Level Computer Vision Using OpenCV,” *2011 24th SIBGRAPI Conference on Graphics, Patterns, and Images Tutorials*, 2011.
- [22] Hairong Jiang, Juan P. Wachs and Bradley S. Duerstock, "Integrated vision-based robotic arm interface for operators with upper limb mobility impairments," *2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*, 31 10 2013.
- [23] Douglas Morrison, Peter Corke and J”urgen Leitner, “Closing the Loop for Robotic Grasping:A Real-time, Generative Grasp Synthesis Approach,” 2018.
- [24] Z. Keita, “YOLO Object Detection ExplainedYOLO,” 2022.
- [25] Juan R. Terven, Diana M. Cordova-Esparza, “A COMPREHENSIVE REVIEW OF YOLO: FROM YOLOV1 AND,” 9 8 2023.
- [26] 陳昭明, “PyTorch 開發者傳授秘笈,” 深智數位股份有限公司, 2022.
- [27] T. Dewi, S. Nurmaini, P. Risma, Y. Oktarina and M. Roriz, "Inverse kinematic analysis of 4 DOF pick and place arm robot," *International Journal of Electrical and*

Computer, vol. 10, no. 2, pp. 1376-1386, 2020.