

基於深度學習之物件辨識技術輔助四軸機械手臂進行抓取與整列功能開發

指導老師：連啓翔老師

組員：林宇竹、林子揚

摘要

本專題採用YOLACT深度學習模型演算法搭配上銀科技RS406四軸機械手臂，結合電磁閥吸取目標物件，再加上輸送帶、D435i深度相機，透過兩階段式的研究模擬真實產線上之異常檢測、排除與物件整列之技術開發，透過Python撰寫深度學習結合C++控制手臂以及相機，搭配程式撰寫手法，使程式能完美整合，完成產線自動化動態辨識物件及作動之目的。在YOLACT模型評估上，初步可以得到模型辨識率高達93.1%、Mask AP達到98.10%，BOX AP達到83.57%，另外，本專題使用D435i深度相機判斷深度抓取異常物件成功率可以達到97.5%，使本系統達到快速而準確之結果。

關鍵字：YOLACT、D435i深度相機、HIWIN RS406

1. 前言

工業人工智慧的崛起，為企業帶來了更多利益，不僅可以改善品質穩定性、提高生產的效率、降低能耗成本、提升設備穩定性...等，將人工智慧應用於工程系統上訓練，不僅具有快速性、可傳承性等特性，使得工業生產以及設備機器在原本的基礎上差異化提升，而應用的層面主要聚焦在工業設備以及製造業、交通運輸、能源行業、生產設備以及自動化，在算法上主要有深度學習、模糊學習等。綜觀上述，而工業人工智慧也將掀起第四次工業革命，本次工業革命被稱為「智慧化」的革命，主要解決的核心問題包括1.提升知識作為核心生產要素的生產力邊界、使得知識的產生、利用和傳承過程中的效率化和規模化，從而獲得本質的提升；2.使整個產業鏈具有高效的協同方式組織生產活動，希望未來工業人工智慧更加成熟，為人類們帶來一大福祉。故本專題針對工業自動化之技術加以開發並自行研究，透過深度學習模擬真實產線前端與後端，整列與異常檢測並加以剔除之真實模擬技術，透過NC程式、C++、Python程式整合，在程式上我們運用許多構想，例如多執行緒、佇列排程、物件導向等等，另外，在電腦視覺的部分我們也利用所學知識校正相機、霍夫轉換，最後將相機、輸送帶、手臂座標整合，達成我們的目的，即利用深度學習及深度相機透過軟體整合達成動態異常檢測及剔除、整列物件，使使用者可以應用於實際工業場所，使生產更有效率、提升產品穩定性以及降低人力成本之目的。

本次專題分為兩階段的方式進行，配合模擬自動

化產線之運作方式，透過輸送帶、機器手臂、深度相機來達成我們所需之要求，在第一階段我們將物件整列，此過程類似自動化產線之前端，能使後端處理物件的夾取、辨識等更佳容易。在第一階段，首先我們需要知道此系統該如何檢測動態物件之位置，取得像素級之遮罩，以便於判斷物件位置資訊，在基底模型我們採用Daniel Bolya等人所提出的YOLACT可即時解析實例分割之模型[3]來做為辨識物件之基礎模型，其特點為可快速偵測物件，很適合用來作為動態辨識之模型，其運作原理將在後面加以說明，知道物件位置後，我們需要將手臂世界座標、相機座標、輸送帶座標整合，再使用機械手臂搭配末端治具達成所需之作動方式。

2. 研究方法及原理

本專題在初期開始研究便探討如何模擬真實產線上之動態運作情形，故本專題初期便使用霍夫圓偵測加入深度資訊做基礎測試，以偵測圓的方式探討手臂與形心位置之關係，並實際測試上機之狀況，發現為可行之研究，故加以深入研究並加入深度學習辨識物件，模擬真實產線上之作動情況。

2.1 實驗架構

本次專題硬體架構使用上銀科技的RS406關節型四軸機械手臂，結合電磁閥使用氣壓進行吸取，使用麗晨皮帶平面式輸送機BN-1.5m*300mm模擬真實產線上之動態輸送帶，在視覺上使用Intel D435i深度相機作為判斷物件位置及深度之感測器，實際硬體配置圖如圖1所示，在軟體架構我們使用了三種不同的開發語言來進行開發功能之軟體，最後將其整合達成所需之功能，在不同的開發環境下分別有其所需之功能及任務，針對其任務由圖4所示，我們使用Python撰寫YOLACT即時解析實例分割之模型，便於辨識動態物體位置及邊界資訊，並且將取得之數據集進行訓練得到權重模型；使用NC程式進行機械手臂的控制；使用Visual C++搭配RealSense SDK使我們可以取得相機資訊。最後將以上利用Visual C++整合為一，達成程式整合之目標，可使使用者搭配電腦視窗查看物件移動情形及作動狀況。

本次專題分為兩階段，其系統架構與運作流程圖如圖2、3所示，第一階段物件整列的治具為木板，在第二階段異常檢測的治具為橡膠吸頭，搭配氣壓進行吸取，另外，我們使用圖5之圖示區分正常品以及異

常品之差異，其標籤代號分別為A及X、Y、Z，在本次專題需抓取排除之物件即為X、Y、Z。

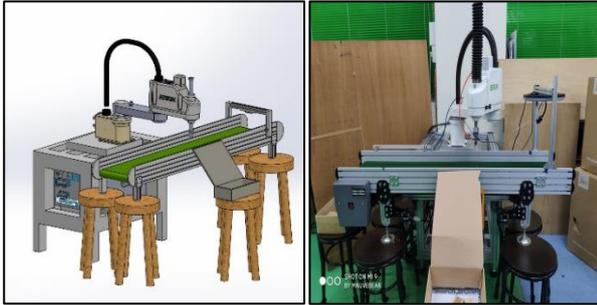


圖1.硬體架構示意圖

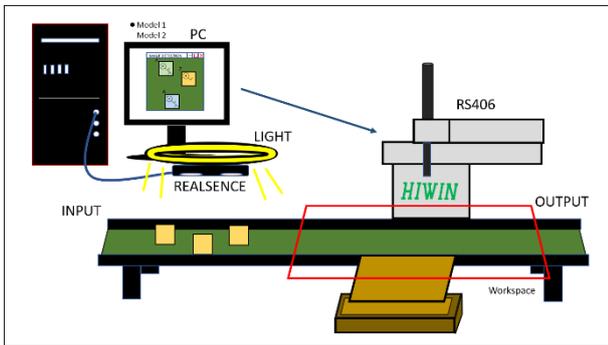


圖2.系統架構示意圖

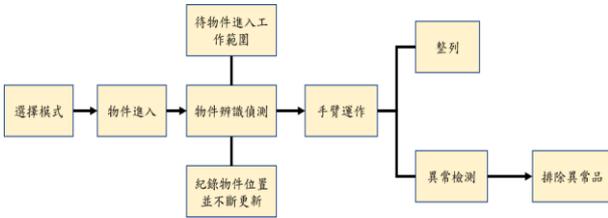


圖3.運作流程圖

開發語言	應用功能
	1.視覺後處理 2.RealSense SDK 3.Hiwin Scara API 4.Python/C API
	1.深度學習 2.視覺辨識 3.訓練數據集 4.視覺前處理
	1.控制機械手臂

圖4.開發語言及其對應功能

正常品(A)	異常品(Z)	異常品(Y)	異常品(X)

圖5.異常檢測圖形示意圖

2.2 實驗分析及原理

2.2.1 電腦視覺-影像處理

電腦視覺(CV)是一個涉及多領域的學科，其目的可用來進行圖片的處理以及分析，包含圖像增強、去噪、校正等，因本次專題需要使用電腦視覺方面之技術作為此系統之眼睛，幫助系統得知物件位置使系統能“看”見，其圖片前處理非常重要，也是本次專題核心之一，以下將敘述本次專題在影像處理所涉略之技術分析。

已知一張640x480的RGB彩色圖片以數學形式可表示成三維矩陣，單位為像素(pixel)在影像處理上我們通常先將其轉換為灰階圖片以減少數據計算量，可使計算更有效率，其轉換關係大多為下列(1)、(2)式表示：

$$\text{Gray}=0.299R+0.587G+0.114B \quad (1)$$

$$\text{Gray}=(\text{Red}+\text{Blue}+\text{Green})/3 \quad (2)$$

其中又以(1)式最為廣泛應用，此方程式可將三維之矩陣轉換為單一矩陣，即將RGB訊息轉換為單一亮度，提供灰階之訊息，其Pixel大小範圍為0-255，隨著灰階亮度而有所變化，越趨近黑色灰階值越小，反之越大，使用這種做法，可有效提升計算效率、降低計算元素。一般我們在獲取圖片時一定會受到外在環境雜訊干擾，這時我們可以使用高斯濾波(Gaussian Filter)或是中值濾波(Mean Filter)來進行影像平滑化之工作，改善影像之品質以便於後續分析，在本次專題我們也運用到感興趣區域(Region of Interest, ROI)之技術，可將此區域獨立分割出來供我們使用，減少很多不必要檢測的區域，使辨識的工作可以更完美。在我們將RGB圖片轉為灰階化時，我們可以使用二值化區分前景與後景，使被偵測物件更加凸顯，方便後續邊緣偵測、型心計算等，若以數學型式可表示為：

$$\text{dst}(x,y) = \begin{cases} \text{maxVal} & \text{if } \text{src}(x,y) > \text{thresh} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

其中x、y表示像素位置，若像素大小大於我們所設定之閾值，則將該位置大小設為255，反之則將該位置設成0，二值化後我們可以使用Canny邊緣偵測Canny邊緣偵測在1986年由John Canny提出，其運作原理先將圖片以5*5的Gaussian Filter濾除雜訊，其數學表示式為：

$$B = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * A \quad (4)$$

此方法為在濾波器平移時將中心像素值透過整個濾波器的加權平均加以代換，可濾除像素值過高之雜訊位置，取得濾除雜訊後的圖片後，透過Sobel濾波器濾出邊緣的強度以及方向，其濾波器型式如下所示：

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \quad (5)$$

$$G = \sqrt{G_x^2 + G_y^2}, \quad \theta = \arctan\left(\frac{G_x}{G_y}\right) \quad (6)$$

利用(6)式即可得出平滑影像中之梯度大小以及方向，在尋找最大梯度方向最後將最大值抑制後，設定雙閾值將邊緣點連線，得出物件邊界。此過程後可透過偵測影像輪廓求得此物件形心位置，具體形心位置可用數學形式表示為：

$$M(\bar{x}, \bar{y}) = \begin{cases} \bar{x} = \frac{\int_A \bar{x} dA}{\int_A dA} \\ \bar{y} = \frac{\int_A \bar{y} dA}{\int_A dA} \end{cases} \quad (7)$$

2.2.2 霍夫轉換

霍夫轉換為一種特徵提取的演算法，1981年由 Dana H. Ballard[2]提出後，在電腦視覺廣泛應用，其轉換可簡易分辨物體形狀，並找出物件中的特徵，在本次專題初期便使用此種方式評估系統可行性，以便規劃後續研究，以下將簡略敘述其霍夫變換演算法之原理。

經過前述邊緣偵測後，我們可以取得物件邊緣資訊，使用霍夫轉換將原本影像空間轉換為或霍夫空間，影像空間中一條直線可表示為：

$$y_0 = m_0 x_0 + b_0 \quad (8)$$

若將其轉換為霍夫空間之形式則可表示為

$$b_0 = -m_0 x_0 + y_0 \quad (9)$$

若 (x_0, y_0) 與 (x_1, y_1) 在圖片空間中同一斜率上，則在霍夫空間則會兩條直線會交於一點，經過變換後再進行最大值投票，即可獲得物體真實邊緣之數值，可更加方便於後續分析。若在影像空間中為圓，則數學形式可表示為：

$$(x - x_c)^2 + (y - y_c)^2 = r^2 \quad (10)$$

在(10)式中未知數為 (r, x_c, y_c) ，在遍歷二值化圖片影像找出非0的點後，找出其梯度方向統計其圓心位置，再利用我們所設最大半徑及最小半徑選取最適合半徑，求得霍夫轉換圖。

2.2.3 人工智慧(Artificial Intelligence, AI)

人工智慧若照字面上來解析即為模仿人類之

智能、智慧所建立之一套系統，此系統類似人類大腦，有神經元，而神經元若受到刺激後則負責傳遞訊息，而最後決定是否需要作出相對應動作抑或是辨識所看見物體，其架構分為輸入層、隱藏層、輸出層，即配對到人之受器、神經元、肌肉。本專題使用之深度學習是機器學習的分支，常聽到的深度學習是指多層隱藏層之神經網路，而深度學習以人工神經網路(ANN)為基本架構，對取得之數據進行表徵學習的演算法，本專題所使用基本神經網路即為捲積神經網路(Convolutional Neural Networks, CNN)，由楊立昆(Yann LeCun)所奠定其基礎，其圖片辨識能力比人類辨識還精準，其運作後可取得圖片之特徵，例如：紋理、線條、顏色...等，用以後續判斷物件分類，其計算方式如圖6所示，計算公式如(11)式所示

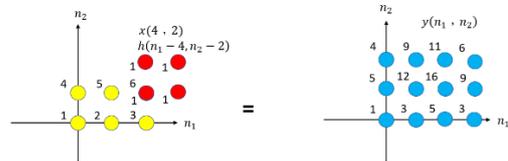


圖6. 圖片捲積示意圖

$$y(n_1, n_2) = x(n_1, n_2) ** h(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) h(n_1 - k_1, n_2 - k_2) \quad (11)$$

將圖片捲積過後再經過池化層後我們可得出此張影像上之特徵訊息，在經過後續的全連接層以及輸出層，即可辨識出影像之類別，然而單純的捲積神經網路有一個缺點，便是無法得知物體實際位置以及物體實際形狀，故我們需要應用到圖片分割分類，其優點為可得知物體形心位置以及物體實際形狀。

2.2.3 實例分割(Instance Segmentation)

在圖片分割分類上，可區分為實例分割(Instance Segmentation)以及語義分割(semantic segmentation)，語義分割如圖7左，為在影像中做像素級之分類，由分割分類所得出之結果，賦予每一個像素意義，但無法得知物體個數。本次專題使用實例分割的分式，如圖7右，可在影像中做像素級分類並得出實際物體個數，以便於個別處理。

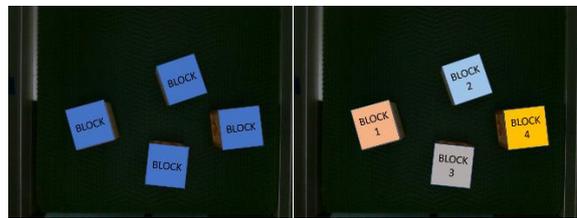


圖7. 語義分割(左)實例分割(右)

2.2.4 YOLOCT

在物件辨識上我們藉由Daniel Bolya等學者在2019年提出之YOLACT[3]，此模型為一個可即時解析實例分割模型，若再Titan XP上運行其速度可

達到33fps，使用COCO數據集驗證時，可由圖8觀察可得，其快速性高又不大幅失去其準確性，故為本次動態產線上之辨識深度模型。

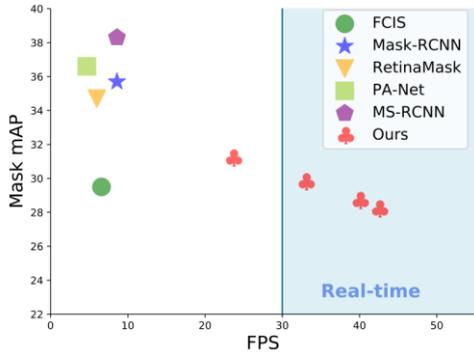


圖8.不同演算法速度性能及Mask準確度比較圖[3]

在過去實例分割檢測模型非常多種，舉凡像是Mask-RCNN[4]、FCIS[5]之類專注於物體之檢測準確度，也有像是Fast-RCNN[6]、SSD[7]之類追求快速性，目前我們使用的模型YOLACT是目前為止追求速度但準確度卻也還不錯的一個模型，其模型類似於SSD與YOLOv3，其速度與MASK-RCNN不同之處由圖9可以看出，將原本的兩階段分類及定位改為一階段就能完成，如圖8所示，其優點為可大幅提升其辨識速度，但同時會降低其辨識準確度。

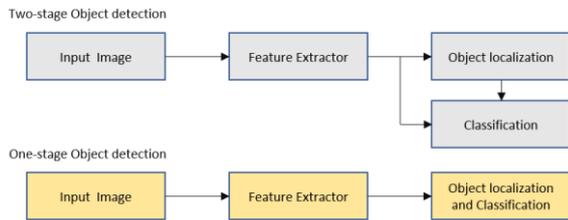


圖9.one stage和two stage物件偵測結構

YOLACT架構圖如圖10所示，其特點為在初期利用影像金字塔之技術，探討不同大小影像之辨識，可以有效提升模型穩健性，並利用RPN網路得到分類類別以及Mask Coefficients後，同時利用prototypes找出最底層影像的多個像素遮罩，因最底層影像擁有最多訊息，保持良好的空間資訊，能處理較複雜特徵，再將兩者相乘可得出真實物件位置及像素遮罩資訊，最後透過閾值分離前後景及可得到實例分割遮罩影像，並達成物件辨識之功能。

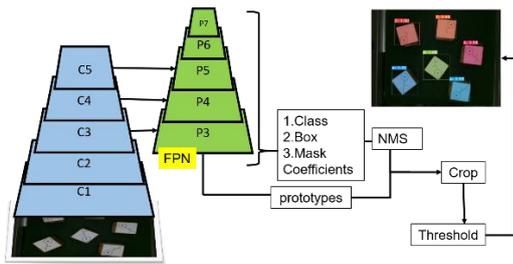


圖10.YOLACT架構示意圖[3]

另外，本專題採用之YOLACT為分類模型，故在結果與討論其辨識率時我們以F1-Score作為模型辨識率之評估指標，F1-Score需先由混淆矩陣 (confusion matrix) 舉例，圖11為其示意圖，其中Precision為精確率，代表對於模型之正向預測信賴程度，Recall為召回率，代表預測正確佔總體的比例值，可利用此二式得出F1-Score的數學表示式，如式(12)可知若模型之準確率與召回率達到100%時，其F1-Score將會達到100%，可用此評估模型之效能。

混淆矩陣	實際YES	實際NO
預測YES	True Position (TP)	False Negative (FN)
預測NO	False Position (FP)	True Negative (TN)

$$\text{Recall} = \frac{TP}{TP+FN} \quad \text{Precision} = \frac{TP}{TP+FP}$$

圖11.混淆矩陣

$$F1_{\text{Score}} = \frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}} \quad (12)$$

2.2.5 座標轉換(Coordinate Transformation)

座標之觀念為幾何運算的基礎，最常使用的為卡氏座標系統，座標轉換(Coordinate Transformation)為固定座標系統(Fixed Coordinate)與移動座標系統(Movable Coordinate)間之運算。位置向量可經由座標轉換-座標平移與座標旋轉，轉換到不同位置。在本專題的平面座標轉換我們利用棋盤方格法使輸送帶座標與手臂座標一致，再將相機座標與輸送帶座標整合，使三者x、y、z能在同一卡式座標系統之下。

在位置轉換上，因需要考慮手臂移動單位為厘米，而物件座標點單位為Pixel，故須探討兩者之關係，已獲取手臂末端點須到達之目標位置，如13式所示其中 R_x 、 R_y 、 R_z 為手臂末端點須到達位置， R_{x0} 、 R_{y0} 、 R_{z0} 為手臂原點，對應相機原點為 P_{x0} 、 P_{y0} 、 P_{z0} ， P_x 、 P_y 、 P_z 為偵測到形心位置像素點，最後factor為像素與厘米的轉換因子，經過13式即可得出手臂對於靜態目標之位置。

$$\begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} = \begin{bmatrix} R_{x0} \\ R_{y0} \\ R_{z0} \end{bmatrix} + \begin{bmatrix} P_x - P_{x0} & 0 & 0 \\ P_y - P_{y0} & 0 & 0 \\ \frac{P_z - P_{z0}}{factor} & 0 & 0 \end{bmatrix} \begin{bmatrix} factor \\ 0 \\ 0 \end{bmatrix} \quad (13)$$

若目標物件為動態時，則需另外考慮其速度以及時間，才能知道機械手臂需要提前多少時間出發，才能在某一時刻剛好對應目標點位置，我們專題使用14式做為本次動態核心方程式，其中我們假設輸送帶以及手臂為等速狀態， X_R 、 Y_R 、 Z_R 為手臂當下

世界座標位置， X_f 、 Y_f 、 Z_f 為目標點的位置，透過此式可以得出機械手臂末端點在何時開始移動才能與目標點在同一時間吻合並抓取。

$$\frac{S_{輪}}{V_{輪}} - \frac{\sqrt{(X_R - X_f)^2 + (Y_R - Y_f)^2 + (Z_R - Z_f)^2}}{V_{手}} = t_{move} \quad (14)$$

2.2.6 IoU(Intersection over Union)

為了評估模型像素遮罩或是邊界框是否精準，可利用IoU進行評估，此標準用於測量真實遮罩和預測遮罩之間的相關度，相關度越高代表越精準，也因遮罩會影響形心值的判斷，與是否能抓取到物件息息相關，而邊界框會影響整列的安全性，故在此討論此模型之IoU。

由圖12假設可知真實遮罩與預測遮罩之交集與聯集，則透過式(13)可得出其IoU，最後將所需驗證之影像分別預測後，即可得到每張影像之IoU，最後將其平均，可得出此模型之平均IoU，若越趨近於1則此模型辨識準確度相對越高。

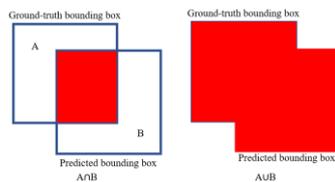


圖12.真實遮罩與預測遮罩聯集與交集

$$IoU = \frac{A \cap B}{A \cup B} \quad (13)$$

2.2.7 程式整合運作流程

在程式整合的部分我們利用C++合成一套名為Tungsten的系統，並將系統劃分為Tungsten Vision與Tungsten Motion兩大部分，再分別包含Tungsten Python Helper與Tungsten Program作為整合的輔助，上述架構如圖13所示，其中Tungsten Python Helper負責管理本系統與Python間的資料互通，因此可以方便將某些以Python開發的演算法整合進Tungsten系統；Tungsten Program則是作為一種介面負責整合多種NC程式的啟動與數值控制。此外，Tungsten Vision負責處理視覺感測並將視覺影像透過Tungsten Python Helper將資料傳入Yolact來解析實例分割並將結果繪製出來；最後Tungsten Motion負責與Scara 機器手臂的連線並結合Tungsten Program達到控制機器手臂的目的。

Tungsten系統廣泛使用多執行緒與物件導向程式設計來提升此系統之性能與擴充性，以上述架構來說Tungsten Vision與Tungsten Motion即為兩個並行的執行緒，因此Tungsten Vision會不停辨識影像將結果推送至佇列等待處理，而Tungsten Motion將持續的依序處理佇列中的結果，從而達到本專題所強調動態追蹤之特色。此外Tungsten系統中包含了兩個重要的資料結構TgPoint與TgObject，其中TgPoint作為記錄三軸向點座標的結構，主要用於

紀錄視覺系統三個維度的座標系，其衍生類別TgWorld則主要用於紀錄機器手臂所需四個維度的世界座標資訊，並且有相關函式可對兩種座標系進行空間轉換；TgObject作為佇列中待處理的物件資料型態，提供Tungsten Vision到Tungsten Motion之間處理時所需的資訊，如物體被捕捉到當下的時間、形心位置、物體外框；此外，TungstenProgram_Base則作為一介面，可建立數個衍生類別個別針對不同NC程式進行封裝，以快速整合多個機器手臂運動模式，本專題所包含的動態整列及動態瑕疵物件控管兩種技術就是基於TungstenProgram_Base所建立的兩種運動模式，最後透過以上規劃成為Tungsten系統運作的流程。

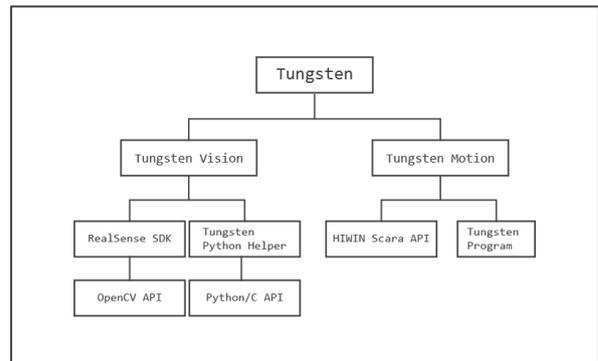


圖13.程式流程架構圖

3. 結果與討論

3.1 YOLACT

3.1.1 模型辨識率(Model Accuracy)

本次評估模型辨識率所採用之方法如2.2節所提之F1-Score，在其分類上做出評估準則，我們在50次不定量的放置上給予主機辨識，實際辨識零件如圖14所示，並記錄每一筆正確以及預測標籤，每組實驗標籤測試如圖14所示。由表1可以觀察到此模型辨識之平均準確率以及平均召回率之狀況，最後F1-Score到達0.93，當F1-Score越趨近於1代表我們的模型穩健度越高，可信度也越高，所以本次模型評估經分析發現我們可以已高達93.1%的機率的分辨出兩種不同的零件，而不會有所遺漏，模型可信度非常高。

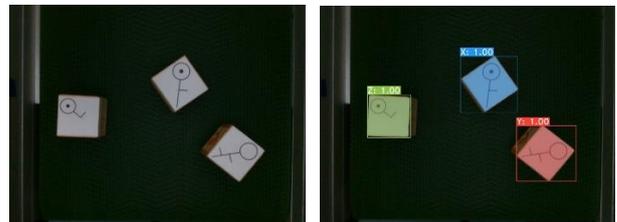


圖14.實際辨識物件示意圖

表1.本專題實驗影像辨識結果

影像序號	實際影像標籤	預測影像標籤	準確率	召回率	F1-Score
1	Y,A,Z	Y,A,Z	1.00	1.00	1
2	X,Z	X,Z	1.00	1.00	1
3	X,Z	X,Z	1.00	1.00	1
4	A,Y,X,Z,A	A,Y,A,Z,A	0.80	0.80	0.80
5	A,X,Z,Y	A,X,Z,Y	1.00	1.00	1
...
49	X,Z,Y	X,Z,Y	1.00	1.00	1
50	A,Y,Z,A	A,Y,Z,X	0.75	0.75	0.75

3.1.2 mAP(mean average precision)

在2.2.6中我們提到的IoU可以幫助我們計算遮罩以及邊界框的準確度mAP，透過設定IoU閾值可以得到在當前閾值下IoU的通過率，本次測試之驗證集共有60組照片，表2為在當前IoU閾值下Mask、Box通過之平均，可觀察得出我們將閾值由0.5開始慢慢地增至0.95，也可藉由評估整體IoU平均得到模型的mAP，Mask為98.10%、Box為83.57%，由此可得知本模型的mAP在Box上較為不穩定，探討其原因認為是因為數據量不足，但是本模型在Mask擁有非常好的成績。

表2. IoU閾值下Mask、Box之通過率

IoU	0.50	0.55	0.60	0.65	0.70
Box	99.88%	99.88%	99.88%	99.88%	99.03%
Mask	99.88%	99.88%	99.88%	99.88%	99.88%
IoU	0.75	0.80	0.85	0.90	0.95
Box	96.55%	95.04%	89.95%	50.45	5.18%
Mask	99.88%	99.88%	99.19%	96.33%	86.27%

3.2相機深度誤差

本次專題使用Intel D435i深度相機檢測物件位置以及深度的感測器，其深度資訊會隨著物件之遠近而有所影響而產生偵測誤差，故由實驗在此討論其深度誤差之結果及實驗過程，圖15為其實驗架構，經過不同距離的擺放紀錄在板子上深度值之上下界限，紀錄並分析。由表3可觀察出若距離相機越遠誤差值相對越高，若物體距離相機1.8公尺誤差最大為20公分，相對若物體距離相機0.3公尺則誤差最大為6mm，本次專題相機距離輸送帶為0.2公尺，其深度值誤差非常小，幾乎可以省略。

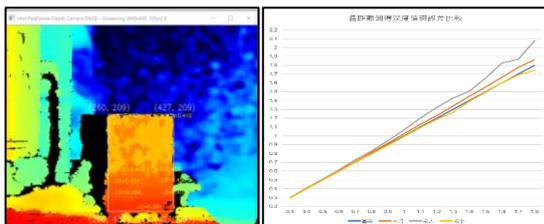


圖15.深度測量及數據示意圖

表3.各距離所測之值(單位為公尺)

基準	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8
平均	0.294	0.402	0.502	0.603	0.707	0.813	0.917	1.024	1.130	1.228	1.338	1.450	1.553	1.664	1.775	1.862
最大	0.300	0.408	0.511	0.616	0.726	0.829	0.944	1.067	1.202	1.322	1.427	1.503	1.656	1.825	1.869	2.078
最小	0.294	0.397	0.499	0.598	0.698	0.793	0.892	0.993	1.092	1.185	1.267	1.390	1.494	1.599	1.689	1.744
誤差+	0	0.008	0.011	0.016	0.026	0.029	0.044	0.067	0.102	0.122	0.127	0.103	0.156	0.225	0.169	0.278
誤差-	-0.006	-0.003	-0.001	-0.002	-0.002	-0.007	-0.008	-0.007	-0.008	-0.015	-0.033	-0.010	-0.006	-0.001	-0.011	-0.056

由上述之結論應用於實測吸取上，我們實驗20組不定量正常品，每組10顆異常品進行異常排除功能，記錄每一組吸取異常品之成功率，如表4所示，可觀察出因深度資訊誤差非常小，以至於整體吸取成功率可達到97.5%，我們探討其失敗原因認為因機械手臂極限位置有限，另一方面也因機械手臂已使用多年，有時候會有奇異點產生，且無法如期等速運行，若將硬體更新，相信會有更好的成績

表4.吸取成功率

次數	1	2	3	4	5
成功率	100%	90%	100%	100%	100%
次數	6	7	8	9	10
成功率	100%	100%	100%	80%	100%
次數	11	12	13	14	15
成功率	90%	100%	100%	100%	100%
次數	16	17	18	19	20
成功率	100%	100%	90%	100%	100%

3.2實際運行

在實際運行時，我們將物件置放於輸送帶上，透過視覺辨識能得知物件位置與邊框，再利用不同治具作出我們需要的功能。在此分為兩階段討論，第一階段為前端的物件整列技術，第二階段為後端的異常檢測排除，以氣壓的方式吸取物件，一方面可加快作動速度，另一方面也能避免使用夾爪力度過大造成零件損壞之情形，以下將一一討論其實際運行作動情形。

在第一階段整列的部分我們取得物件的邊框位置，以利於使手臂判斷該到何處下去撥動物件，並判斷物件偏離中心距離，利用木板治具將物件推向中間，達成整列之效果，實際作動情形如圖16下的連續圖陣所示。在第二階段異常檢測排除我們取得物件的形心位置，開啟電磁閥後，使手臂末端點能在同一時刻形心接觸並吸取，我們也自行設定手臂工作範圍，在物體進入範圍時，開始吸取，一方面可以控制手臂工作極限範圍，另一方面也可以防止危安產生，實際作動情形如圖16上的連續圖陣所示，可排除異常物品丟出輸送帶外側的斜板進入收集區。

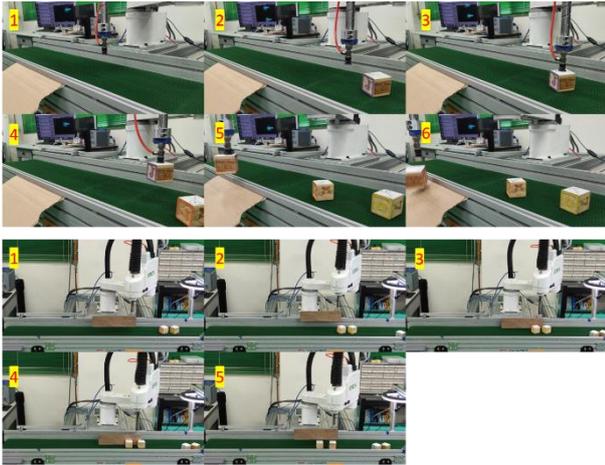


圖16.物件異常排除(上)物件整列(下)

4. 結論

透過本次專題實作，我們開發出具有YOLACT之深度學習整列以及異物檢測吸取之系統，在異物檢測上我們的抓取成功率達到97.5%。在視覺辨識上，我們經過F1評估模型指標整體辨識率達到93.1%，偵測的Mask AP達到98.10%，在程式整合上也成功將C++以及Python整合，最終達到將機械手臂應用於整列以及異常檢測排除之目的。本系統之目的已完成，但針對本次專題的目標物件的選取，仍有待改善，因本系統具有適應性，在未來可針對更多真實產線上之產品研究並探討，因此在未來還需要更多努力，更多實驗以及實測，使系統本身更加完備。

5. 誌謝

在本次專題過程中，雖遇到許多難關以及阻礙，但最後都能迎刃而解，其中最重要還是要謝謝連啓翔老師不辭辛勞地指導我們，在專題的初期奠定我們最基礎的能力，使我們得以在後續更深入實作時有能力初步排除所遇到的問題，以及有能力完成指派之任務，且老師也花非常多的時間與我們討論並修改，毫不保留的傾囊相授。再者，也要十分感謝實驗室的學長們也給予我們非常多的幫助，其中也要特別感謝黃品高學長以及劉建寬學長，扮演著一個輔導者的角色，不時地給予我們建議，並透過他們熟練的實務經驗慢慢的從旁輔助我們，成為我們的幫助者，透過本次專題也可使我們增長許多實務經驗以及知識，對未來的助益無可限量。

6. 參考文獻

1. John Canny, A Computational Approach to Edge Detection, IEEE
2. Dana H. Ballard, Generalizing the Hough transform to detect arbitrary shapes
3. Daniel Bolya, Chong Zhou, Fanyi Xiao, Yong Jae Lee, University of California, Davis YOLACT: Real-time Instance Segmentation
4. Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick, Mask R-CNN, 2017

5. Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie. Feature pyramid networks for object detection. In CVPR, 2017.
6. Ross Girshick, Fast R-CNN, arXiv 2015
7. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, SSD: single Shot MultiBox Detector, arXiv 2015