基於 Darknet 框架以 Yolov3 演算法實現智慧產線檢測系統

指導老師：魏清煌 教授

參賽組員：黃泰源、胥景然、黃柏儒

摘要

目標檢測演算法的主要任務是將影像或圖像中有興趣的物件辨識出來，並確定它的位置、大小。近年來深度學習技術蓬勃發展，目標檢測技術也隨著這股潮流不斷進步，在現在的電腦計算能力下，深度學習目標檢測演算法開始能夠滿足即時檢測任務的需求。本專題希望能將 Yolov3 演算法使用在生產線即時檢測產品的工作上，檢測生產線上產品是否有無缺陷、將檢測結果進行分析，並與資料庫結合把檢測數據紀錄下來，以便生產管理與品管人員了解生產線狀態。

1.前言

1.1 研究動機

目標檢測演算法的主要任務是將影像或圖像中有興趣的物件辨識出來，並確定它的位置、大小。近年來深度學習技術蓬勃發展，目標檢測技術也隨著這股潮流不斷進步，在現在的電腦計算能力下，深度學習目標檢測演算法開始能夠滿足即時檢測任務的需求。本專題希望能 Yolov 演算法使用在生產線即時檢測產品的工作上，檢測生產線上產品是否有無缺陷、將檢測結果進行分析，並與資料庫結合把檢測數據紀錄下來，以便生產管理與品管人員了解生產線狀態。

在工業 3.0 後，生產線上大部分的工作都改以自動化生產以降低人力成本，機器相對於人力的優勢在於機器耐久能強、效率高，能執行精密、單純和重複的工作，但面對較為複雜的組立或檢測步驟就顯得相對地無能為力，在生產線後端較為複雜的組立組裝與檢測步驟仍多是由人力來負責，藉助檢測儀器設備提供的檢測信號，讓具備判讀信號經驗的品管人員決定產品是否有缺陷。

目前大部分的中小型工廠組立、包裝、檢測階段仍交由人力負責，在長時間的生產線工作上，人往往會因疲勞而無法保持專注，此時就有可能因疏失導致作業員在組立階段遺漏零件或沒有篩選出瑕疵零件，而有缺陷的產品依然在生產線上進行包裝、封裝，完成後開始裝箱。對企業而言，一來此種有缺陷之產品如果在市面上銷售會造成公司負面形象；二來瑕疵品如果幸運地在裝箱階段被抽驗的品管人員挑出，通常會將同批已裝箱產品全部拆封重新檢查，會造成不小的人力與時間成本。

1.2 研究目的

為了避免浪費上述不必要的成本，如果在生產線的最終檢測階段增加智慧自動檢測系統，並在發現瑕疵品後發出警示燈光與警示音效來提醒品管人員，便能精準地處理缺陷品。另外，將通過自動檢測的產品所獲得的數據資料及記錄蒐集到資料庫裡，方便產線管理人員了解生產線的狀況並適當地調整生產步驟或工序，也能進一步提高良率。

2.文獻探討

目標檢測演算法主要任務是在影像或圖像中圈選出有興趣的目標，確定他們的種類、大小與位置。由於各項目標具有不同的形狀、大小、顏色、光照、角度等各種參數的影響，一直都是電腦視覺頗具挑戰性的問題。以深度學習為基礎的目標檢測演算法主要分成兩類方法：

(a) One-stage Learning　(b) Two-stage Learning

Two-stage Learning 的方法主要是要先利用 Region proposal 產生可能有目標的 Region 再進行辨識，最經典的就是 R-CNN 系列方法；而 One-stage Learning 的方法則是在一個神經網路上直接完成位置與類別的判定。

Yolo(You Only Look Once)論文公佈於2015年，刊登於arXiv公開論文網站上，掛名的作者分別是JosephRedmon、Santosh Divvala、Ross Girshick以及Ali Farhadi。Yolo計算快速，能夠達到即時檢測的速度需求，缺點是對位置的預測不夠精確，且小物體預測效果較差。Joseph Redmon也在隨後幾年發布了Yolo的改善版本Yolo v2、Yolo v3，皆是為了改善準確率所做的改進。Yolo類神經網路演算法一系列方法的演進是值得研究與學習的。

Yolo 屬於 One-stage Learning 的目標檢測演算法，相較於 R-CNN 系列的方法，Yolo 並不像 R-CNN 需要做 Region proposal，例如 Fast R-CNN 使用獨立於神經網路之外的 Selective Search 模組、Faster R-CNN 將 RPN (Region Proposal Network)整合到神經網路等。Yolo 將物體檢測作為迴歸問題求解，基於一個單獨的 End-to-End 網路，完成從原始影像的輸入到物體位置和類別的輸出。

2-1 Yolo 運作原理

圖19為Yolo神經網路架構圖，前面卷積層負責接收輸入圖像，並對圖像進行特徵提取，後面為全連接層用來預測目標位置並給予類別概率值。例如圖20，檢測目標有兩

個，後面20層為屬於各類別的機率。

基本運作流程如圖21，首先將一張圖像劃分成SxS的網格(S在此設為7)，在每個網格中預測各種類別的機率，並挑選機率最大的類別，同時在網格中預測B個Bounding Box(B在此設為2)的大小與是否有物體的置信度(Confidence)。最後使用NMS (Non-Maximum Suppression) 將不必要的Bounding Box去除掉，獲得最終預測結果。訓練時其使用迴歸方法的Loss Function，如圖22所示。

3.研究方法

本專題採用IKEA宜家家居公司販售的產品HUSET迷你臥室家具(如圖1)作為生產線上即時檢測目標，檢測是否有缺項；以Books W10網路HD高畫質LED燈攝影機(如圖2)做為檢測工具；使用學校實驗室提供的110V、最高速12公尺/分鐘PVC輸送帶(如圖3)進行模擬。

在實做之前，先閱讀目標檢測演算法相關資料，了解檢測方法與相關理論；研究Yolo系列，了解其神經網路架構與設計理念與背後原理。有概念之後，學習如何使用Darknet框架訓練Yolo v3模型，並閱讀Darknet框架原始碼，這樣便能修改原始碼，加入擷取檢測結果、記錄等功能，如圖4之檢測結果範例。

實際進行目標檢測前，必須先收集大量圖像作為訓練數據，將產品放置在生產線上，針對不同的角度、位置、方向、光照，如日光燈、上午或下午、晴天或陰雨天的自然光線、LED燈等各種情境下，進行大量的樣本圖像數據擷取，並利用開源軟體LabelImg圈選目標框(如圖5)。經過框選後會產生以Yolo格式為標準的訓練數據集合，就可以使用這些訓練數據對Yolo v3模型進行訓練以達到目標檢測的功能，此模型將作為檢測系統的辨識核心，通過即時產品檢測，測試其辨識的準確度。

除了辨識核心，根據系統架構圖(如圖18)同時使用Qt開發使用者介面，顯示生產線的生產情況並將產品狀態記錄到資料庫中、提供可視化圖形介面，方便使用者查看。

圖 1、HUSET 迷你臥室家具

圖2、E-Books W10網路HD高畫質LED燈攝影機

圖 3、110V、12公尺/分鐘PVC輸送帶

圖4、檢測結果

圖5、利用LabelImg框選後的結果

4. 神經網路訓練過程

本專題使用HUSET迷你臥室家具作為辨識使用，拍攝時著重在各種不同的光照、角度、方向、缺項、位置。

拍攝完畢後，以「物件特徵肉眼可辨識」、「邊框盡量貼近物件」這兩個標準，使用LabelImg將一張圖片中的各個物件框選出來。框選完後，可以直接輸出成Yolo需要的label檔。

Yolo所需要的label檔採用Yolo format的text文字檔，如圖6，第一欄為類別的代碼，後面欄位為物件框相對於整張圖的比例。

類別代碼	物件中心的x位在整張圖的x的比例	物件中心的y位在整張圖的y的比例	物件寬度佔整張圖片寬度的比例	物件長度佔整張圖片長度的比例

[category number] [object center in X] [object center in Y] [object width in X] [object width in Y]

```
0 0.6784060846560847 0.33630952380952384 0.2767857142857143 0.2837301587301587
0 0.35383597883597884 0.6729497354497355 0.26455026455026454 0.26785714285714285
0 0.23792989417989419 0.3169642857142857 0.22453703703703703 0.18617724867724866
```

圖6、Yolo專用label檔

轉換公式：

X = [Xmin + (Xmax - Xmin)/2] * 1.0 / image_w

Y = [Ymin + (Ymax - Ymin)/2] * 1.0 / image_h

W = (Xmax - Xmin) * 1.0 / image_w

H = (Ymax - Ymin) * 1.0 / image_h

訓練時Yolo會擷取label和圖片，但還需要五個設定檔，分別為：obj.names、obj.data、train.txt、test.txt、yolov3.cfg或yolov3-tiny.cfg。

obj.names為物件的列表，以本專題為例，內容為：bed、box、table、chair、wardrobe、carpet、dinosaur，Yolo在訓練與預測時皆需要讀取此檔。

obj.data定義了物件數目、訓練集路徑、測試集路徑及權重目錄的位置，以本專題為例，內容為：

```
classes= 7
train  = E:\dataset\Project\cfg\train.txt
valid  = E:\dataset\Project\cfg\test.txt
names = E:\dataset\Project\cfg\project.names
backup = E:\dataset\Project\weights\
```

train.txt為訓練集列表，我們設為數據集的80%，訓練時Yolo會依次讀取該檔內容取出相片進行訓練。

test.txt為測試列表，我們設為數據集的20%，可視需求調整比例，訓練時Yolo會依次讀取該檔內容取出相片進行測試。

yolov3.cfg為Yolo神經網路架構的模型設定檔，由於yolov3.cfg模型權重參數龐大且運算效能不足，會造成影像檢測畫面禎數低落，因此本專題最後採用yolov3-tiny.cfg。

圖7為數據分類圖，bed至wardroble均放入image及label檔供Yolo讀取，weights用來儲存Yolo專用的權重檔。

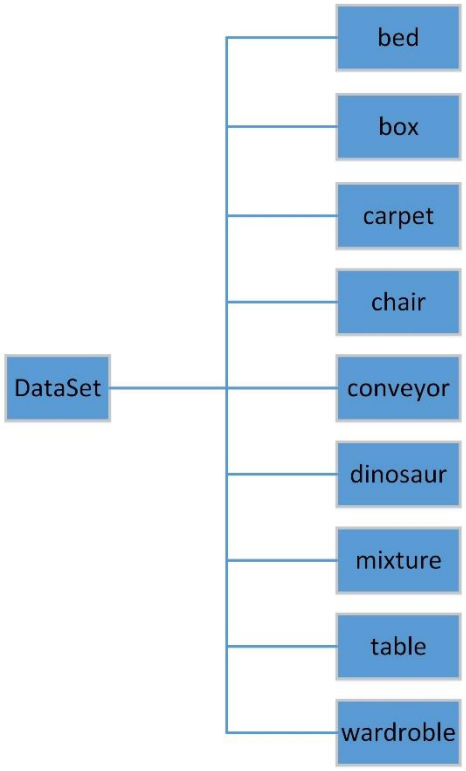

圖7、數據分類圖

以上步驟皆完成後便可執行darknet進行訓練，訓練過程會持續秀出各種數值，如圖8，並每隔100批次產生一個權重檔。最需要注意的是紅框的average loss error，如果訓練的圖片數目有數千以上，那麼average loss error約0.06便可停止；如果僅有數百張，則大約0.6，便可載入權重檔測試辨識效果。最後檢測效果如圖9。

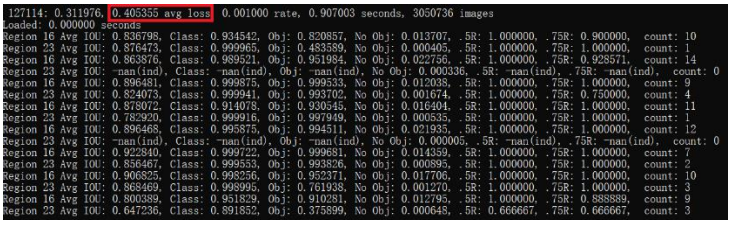

圖8、Yolo訓練畫面

圖9、最後檢測效果

5. 計數模組與錯誤判別

此系統進行計數與錯誤判別的流程圖如圖23。計數主要是以包裝盒為目標進行計數，如果包裝盒中點有通過中間的計數線便計數，示意圖如圖10。當包裝盒通過中間的計數線時需要判別零件的歸屬，即判別哪些零件的中點落入包裝盒中示意圖如圖11，黑點表示個別零件之中點，如果黑點落入box中即歸屬於此box中。做完零件的歸屬判別後，將辨識結果進行檢測，檢測是否缺陷或缺項。最後將檢測結果記錄到資料庫中。

圖 10、計數示意圖

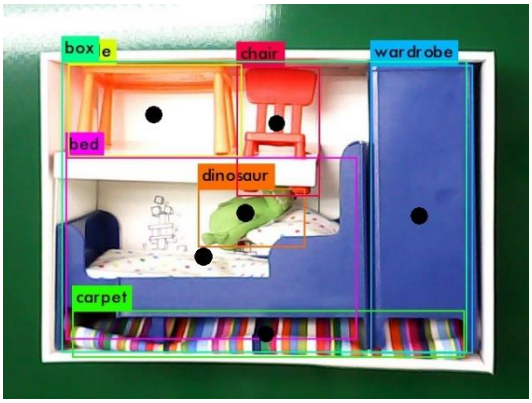

圖 11、零件歸屬示意圖

6.實作成果

此系統主要功能為即時的辨別生產線上的產品是否有無缺陷，並可以查看之前所記錄的數據。圖12顯示了檢測畫面的功能，畫面上會顯示現在時間、目前檢測的目標 、今日的計數、有缺陷產品的數目。按下【開始檢測按鈕】之後便能開始檢測生產線上的產品並記錄在資料庫中，按下【停止】便能停止檢測。按下【查看數據】按鈕便會跳出數據頁面，數據頁面會顯示之前所生產的數據相關資料如圖14。按下【統計圖表】便會顯示長條圖來顯示最近幾天的生產狀況，也能根據不同的條件來進行顯示。

7.結論與未來展望

在輸送帶上實驗後獲得以下結論：在不同光照下，攝影機拍攝到的零件色澤會有所不同，辨識結果會受到影響，目前我們的解決方法是收集不同光照下的圖片來進行訓練，來提高辨識的穩定度；但此方法耗力費時，也許有其他相關影像處理的方法能夠改善此問題。

本專題最有難度的地方在於整合Darknet框架API撰寫的辨識核心與Qt撰寫的介面，Darknet框架的API並無詳細的介紹，在撰寫時必須閱讀原始碼來了解函式的功能，並依所要之功能對原始碼進行修改。Qt是我們首次接觸的開發工具，許多功能不熟捻，經常是邊做邊學，造成撰寫上並無嚴謹的架構，對於之後維護上可能會造成不小的問題。

我們這次的實驗僅能辨識出是否缺項，希望未來能收集到零件在生產線上會發生的瑕疵來進行辨識，提升神經網路辨識瑕疵的能力，並開發簡單易用的訓練介面，讓使用者可以針對不同的產品進行辨識。

圖12、檢測頁面

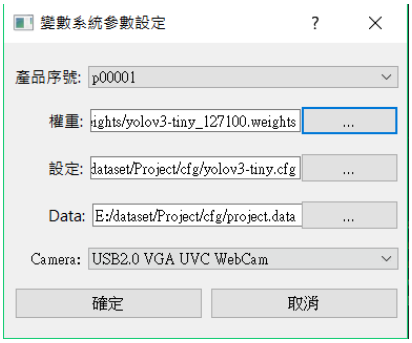

圖13、參數設定對話框

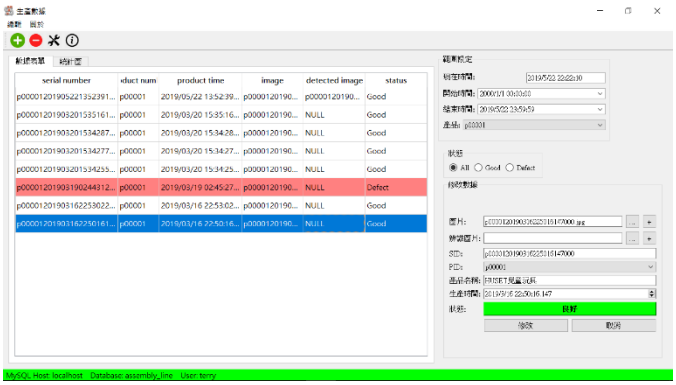

圖14、數據頁面

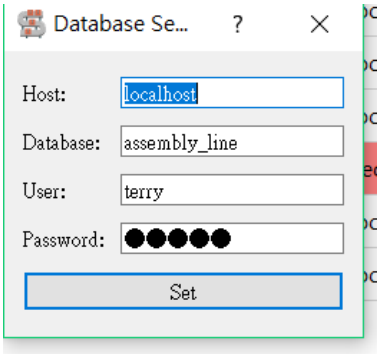

圖15、資料庫設定對話框

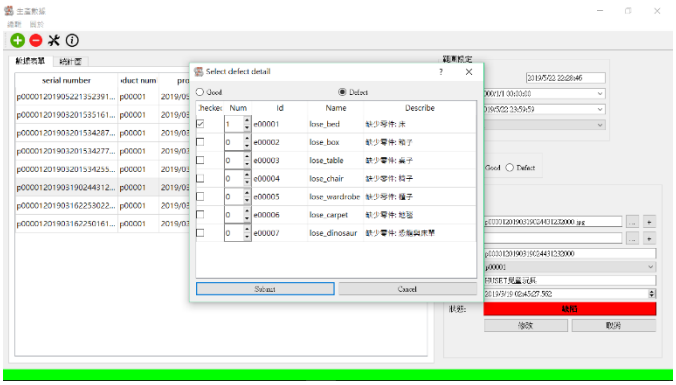

圖16、瑕疵設定對話框

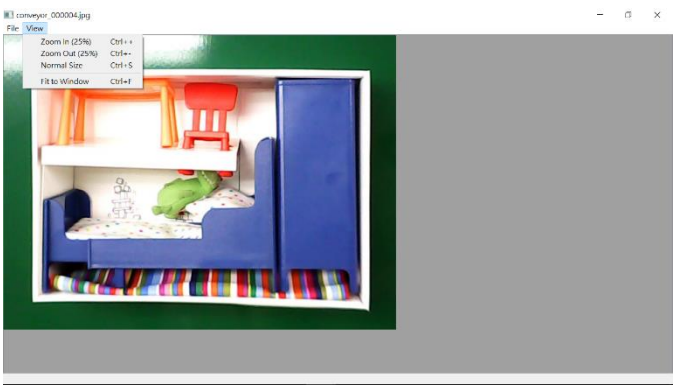

圖17、圖片檢視頁面

參考文獻

[1] J. Redmon and A. Farhadi, Yolo v.3: An Incremental Improvement, ArXiv, pp. 1-6, CVPR 2018.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, ArXiv, pp. 1-10, CVPR 2016.

[3] J. Redmon and A. Farhadi, YOLO9000: Better, Faster, Stronger, ArXiv, pp. 1-9, CVPR 2017.

[4] Darknet: Open Source Neural Networks in C, https://pjreddie.com/darknet/

[5] Z. E. Lee, Hands-On GUI Programming with C++ and Qt5, Packt, 2018.

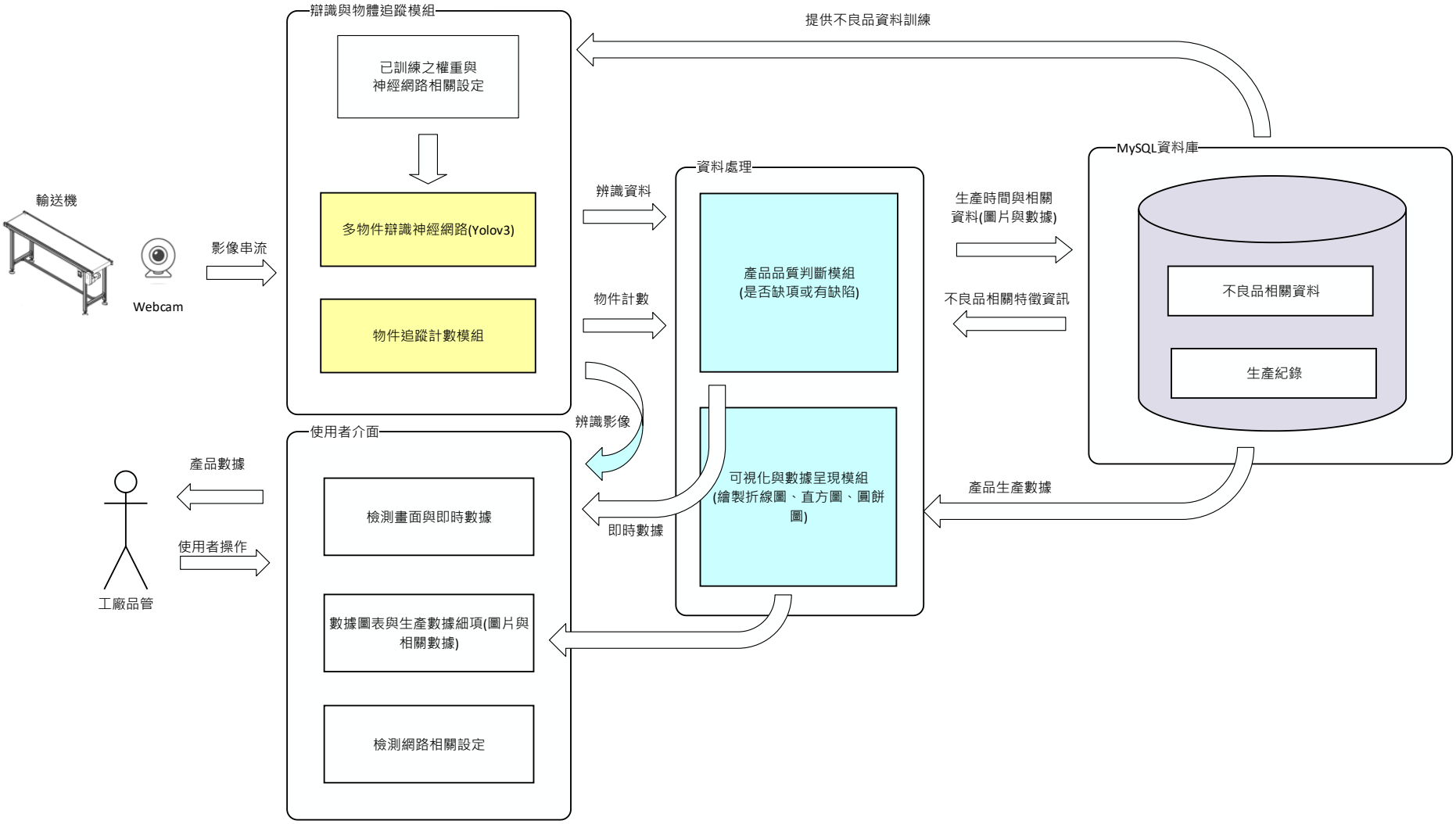
辯識與物體追蹤模組
已訓練之權重與
神經網路相關設定
多物件辯識神經網路(Yolov3)
物件追蹤計數模組
提供不良品資料訓練
MySQL資料庫
不良品相關資料
生產紀錄
資料處理
辨識資料
物件計數
辨識影像
即時數據
產品品質判斷模組
(是否缺項或有缺陷)
可視化與數據呈現模組
(繪製折線圖、直方圖、圓餅圖)
生產時間與相關
資料(圖片與數據)
不良品相關特徵資訊
產品生產數據
輸送機
Webcam
影像串流
使用者介面
檢測畫面與即時數據
數據圖表與生產數據細項(圖片與相關數據)
檢測網路相關設定
產品數據
使用者操作
工廠品管

圖 18、系統架構圖

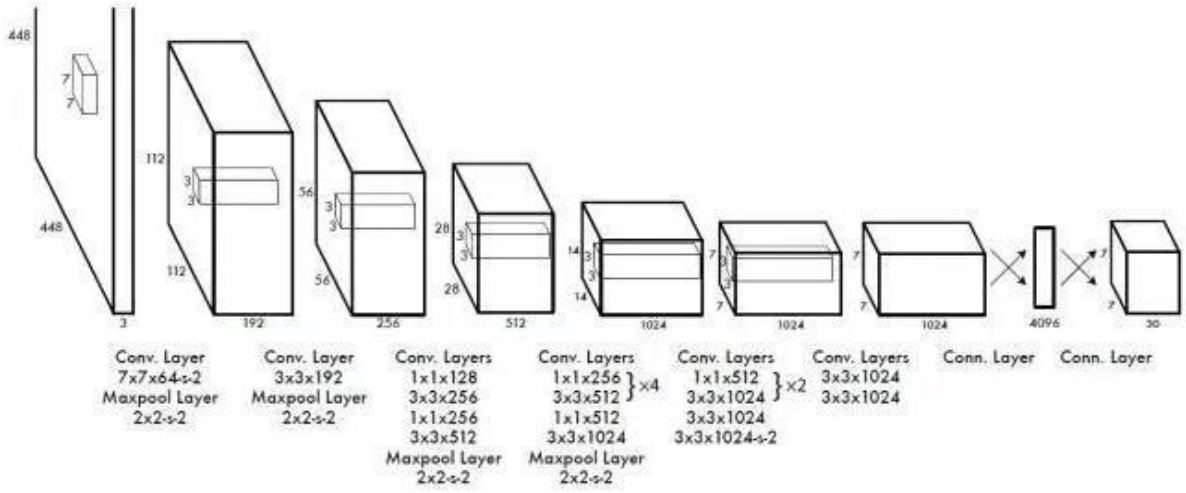

圖 19、Yolo 神經網路架構

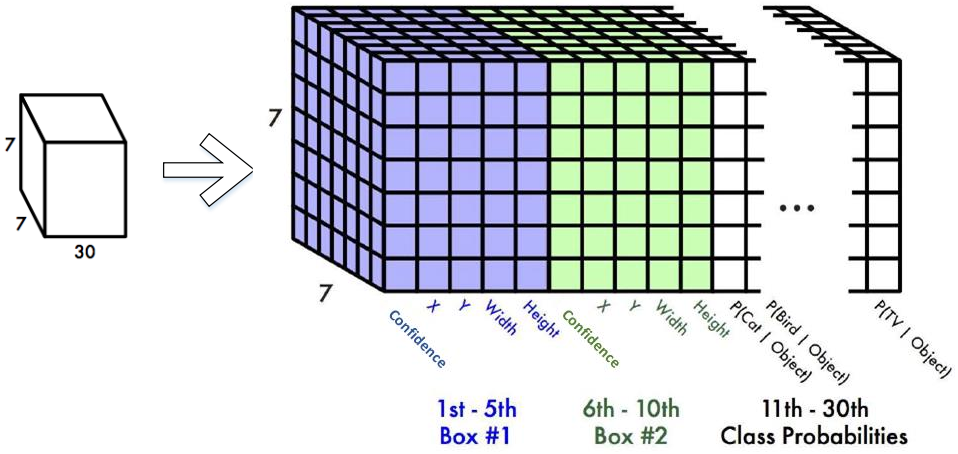

圖 20、Yolo 輸出層

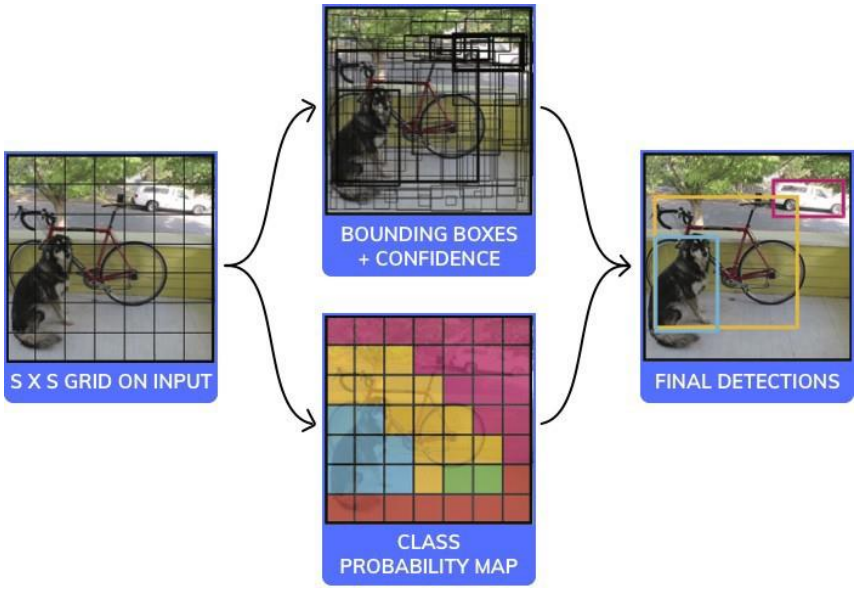

圖 21、Yolo 處理示意圖

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

Bounding Box的大小與位置的lossfunction

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

Bounding Box是否含有物體的置信度的lossfunction

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

在網格裡各類別機率訓練的lossfunction

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2$$

$\mathbb{1}_{ij}^{\text{obj}}$ 如果網格i裡的Bounding box j有物體需要被檢測則為1，否則為0

$\mathbb{1}_{ij}^{\text{noobj}}$ 如果網格i裡的Bounding box j沒有物體需要被檢測為1，否則為0

$\mathbb{1}_{i}^{\text{obj}}$ 如果有物體在網格i中為1，否則為0

圖 22、Yolo Loss Function

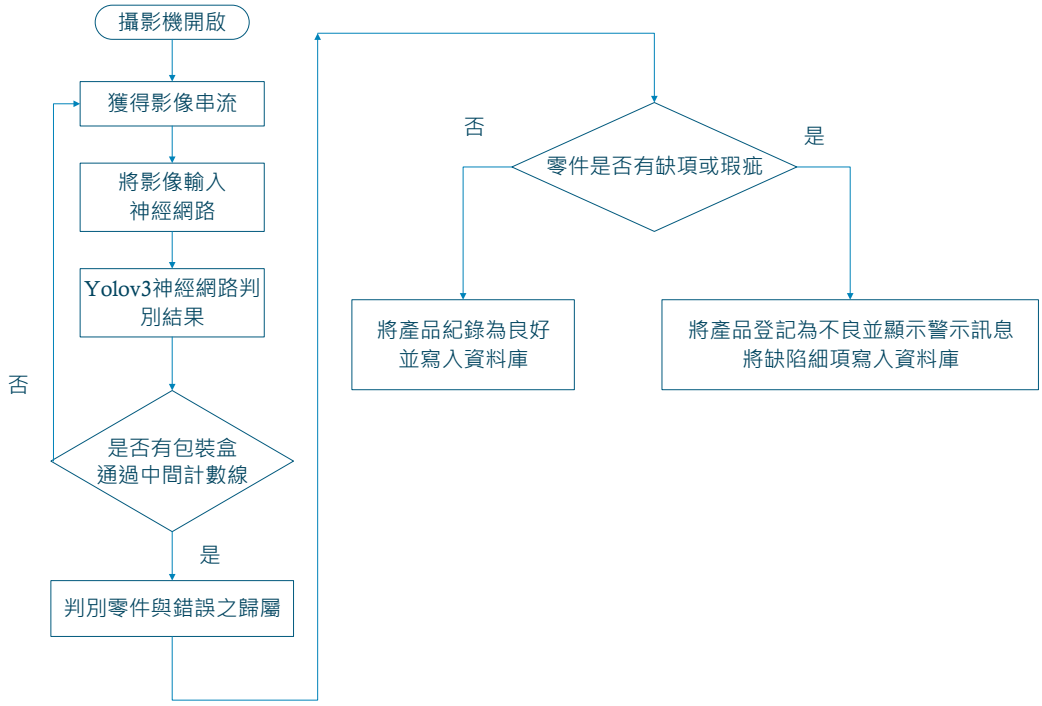

圖 23、計數與錯誤辨識流程圖

附錄

基於 Darknet 框架以 Yolov3 演算法實現智慧產線檢測系統

指導老師：魏清煌 教授

參賽組員：黃泰源、胥景然、黃柏儒

材料成本

項次	項目	規格	數量	單價(元)	金額(元)
1	HUSET 迷你臥室家具	長度: 32 公分 寬度: 21 公分 高度: 17 公分 重量: 0.83 公斤	3	400	1200
	合計				1200

設備折舊 (使用設備折舊金額以購買價格的 1/5 計算)

項次	項目	規格	數量	單價(元)	金額(元)
1	皮帶式輸送帶	W400 x L2000 x H900 單位:mm 最大速度: 12 米/min	1	5460	5460
2	E-Books W10 網路 HD 高畫質 LED 燈攝影機	動態解析度：1920X2560 VGA 靜態解析度：5000 萬畫素 色彩：24 位元 電壓：3.3 V／電流:小於 200mA 工作頻率：60Hz USB 2.0 連接埠介面 鏡頭尺寸：1/7 英吋 尺寸：6.5x6x5 cm	1	150	150
3	ASUS 筆記型電腦 X550V	CPU: Intel Core i5-6300HQ GPU: NVIDIA GTX 950M Memory: 8GB	1	5400	5400
	合計				11010

人力成本

項次	工作項目	說明	工時	單價(元)	金額(元)
1	無				
2	雜項工作				
	合計				0

組員貢獻度 (全部組員貢獻度合計 100%)

學號	姓名	主要工作項目	貢獻度(%)
0551076	黃泰源	神經網路訓練、辨識核心撰寫、資料庫設計與實作、數據處理模組撰寫	60%
0451102	胥景然	程式介面撰寫、文件撰寫	20%
0551034	黃柏儒	數據收集、圖片標註、海報製作	20%