

國立雲林科技大學資訊管理系

機器學習-作業三

Department of Information Management

National Yunlin University of Science & Technology

Assignment

貨櫃號碼資料集

Container number data set

巫宇哲、鄭皓名、翁振洋

指導老師：許中川 博士

Advisor: Chung-Chian Hsu, Ph.D.

中華民國 112 年 6 月

June 2023

摘要

本研究主要的動機是想要幫助碼頭人員可以快速辨識貨櫃的編號，如果能善用物件偵測與辨識的技術來進行貨櫃辨識將大幅降低人力成本。希望能藉由提供正確的編號資訊，來幫助碼頭人員成功執行相關的任務。該研究主要透過貨櫃號碼資料集來進行物件偵測與辨識的訓練，以 YOLO 物件偵測與 OCR 辨識技術來訓練模型，YOLO 模型可以判斷影像中貨櫃號碼的位置，而 OCR 則是辨識 YOLO 所偵測到貨櫃編號。使用這兩種深度學習的方法，來訓練和驗證貨櫃影片(影像)的辨識績效，並觀察影片(影像)當中辨識的誤差有多少，來調整模型當中的一些參數，並達到更準確的預測。在資料預處理中做了resize還有將圖片和標註好的 xml 檔做合併讓模型進行學習，在 OCR 辨識預處理中，將偵測出來的貨櫃編號圖片轉灰階和圖像增強讓模型辨識度能提升，而針對貨櫃影片的辨識，則是將每一幀影像皆進行物件偵測和辨識。在訓練模型部分，使用 SiLU、EarlyStopping 等 callbacks 來動態調整超參數。從實驗結果的一些可視化指標也能發現 Recall 和 Precision 之間的負向關係，尤其當 Recall 大到一個值後 Precision 就會明顯下降，score_threshold 的設定可以控制模型檢測結果的準確性和召回率之間的平衡，雖然模型在貨櫃編號辨識的準確率上還不錯，但是對於給予整部影片然後丟給模型一幀一幀來辨識貨櫃編號的準確率並沒有想像中的好，準確率才 50%左右，可能是因為影片的場景較為多樣性導致辨識率降低，因此如果能針對辨識的模型自己訓練一個模型，或許準確率會有所提升，這可以作為未來研究。

關鍵字：YOLO、OCR、SiLU、EarlyStopping、Recall、Precision

一、緒論

1.1 動機

本研究透過貨櫃號碼資料集來進行物件偵測與辨識的訓練，以 YOLO 物件偵測的技術與 OCR 辨識的技術來進行模型訓練，YOLO 只需要對圖片作一次卷積神經網路架構便能夠判斷圖形內的物體位置與類別，而 OCR 則是去辨識 YOLO 所偵測到的位置的編號，本研究想要藉由這兩項技術來進行研究，會做這項研究主要動機是想要幫助碼頭人員可以快速辨識此貨櫃的編號，如果模型成效不錯，就能節省人工辨識的成本，這件事情對於碼頭貨櫃相關的工作人員相當的重要，每一天進進出出許多的貨櫃船，想當然就會有許多的貨櫃車需要辨識號碼，如果能善用物件偵測與辨識技術來進行貨櫃辨識將大幅降低人力成本。

1.2 目的

本研究想要藉由深度學習的方式，還有運用 YOLO 物件偵測的技術與 OCR 辨識的技術，來訓練和驗證貨櫃影片的辨識績效，並觀察影片當中辨識的誤差有多少，在從誤差當中來對模型進行修改與調整，主要是調整模型當中的一些參數，來達到更準確的預測，有準確的編號資訊，碼頭人員的負擔也能減輕許多，貨櫃辨識是一項相當費時的工作，本研究希望能藉由提供正確的編號資訊，來幫助碼頭人員成功執行相關的任務。

二、方法

2.1 實作說明

本研究工作主要用於辨識貨櫃編號，收集了貨櫃車車牌照片和標註好的 xml 檔，以使用於訓練物件偵測模型，本研究使用了 YOLOv7 架構作為我們物件偵測的模型再加上 EasyOCR 作為後續我們抓到貨櫃編號後的辨識，上述兩者技術都是屬於深度學習的框架，且也都有使用預訓練模型，在物件偵測預處理上面，本研究做了 resize 還有將圖片和標註好的 xml 檔做合併讓模型進行學習，在 OCR 辨識預處理上面，將偵測出來的貨櫃編號圖片轉灰階和圖像對比度增強讓模型辨識度能提升，最後拿測試資料來評估我們辨識的績效如何，針對貨櫃車牌影片的辨識，本研究則是將每一幀數都抓出來進行物件偵測和辨識，最終辨識結果為影片中辨識出最多次的號碼牌。

2.2 操作說明

本研究執行環境採用 Python3.7.16，以 Visual Studio Code 作為開發工具，利用 Pytorch、Keras、網路上公開的 YOLOv7 模型及 EasyOCR 文字辨識模型來達成貨櫃號碼偵測的任務，並使 Numpy、PIL、cv2、Matplotlib 等函式庫來讀取圖片，貨櫃影像前處理、以及模型偵測所抓取的區域圖像和績效以視覺化的形式呈現，透過影像壓縮、調整超參數等方式來增加模型泛化能力。在訓練模型時，使用 SiLU、EarlyStopping 等 callbacks 來動態調整超參數，最後在文字辨識前，針對物件偵測所抓取的區域圖像轉灰階以及做對比度增強等方式，使文字辨識能更精準，並計算其偵測貨櫃號碼的準確率。

三、實驗

3.1 資料集

名稱:貨櫃號碼資料集

訓練資料筆數：2125

測試資料筆數：35

驗證資料筆數：536

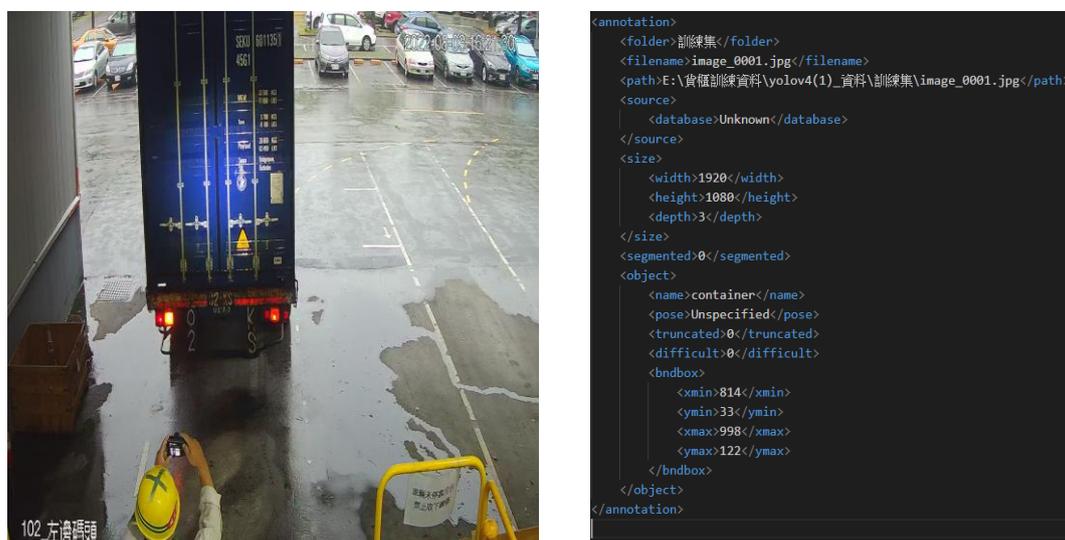


圖 1 訓練資料的貨櫃車照片與 xml 檔照片



```

<annotation>
  <folder>測試集</folder>
  <filename>image_0020.jpg</filename>
  <path>E:\貨櫃訓練資料\yolov4(1)_資料\測試集\image_0020.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>1920</width>
    <height>1080</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>container</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>838</xmin>
      <ymin>69</ymin>
      <xmax>1095</xmax>
      <ymax>158</ymax>
    </bndbox>
  </object>
</annotation>

```

圖 2 測試資料的貨櫃車照片與 xml 檔照片



```

<annotation>
  <folder>驗證集</folder>
  <filename>image_0077.jpg</filename>
  <path>E:\貨櫃訓練資料\yolov4(1)_資料\驗證集\image_0077.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>1920</width>
    <height>1080</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>container</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>801</xmin>
      <ymin>72</ymin>
      <xmax>1070</xmax>
      <ymax>167</ymax>
    </bndbox>
  </object>
</annotation>

```

圖 3 驗證資料的貨櫃車照片與 xml 檔照片

3.2 前處理

● 資料前處理

■ 物件偵測預處理

將原始的輸入圖片和標註好的 xml 檔做整合，檔案格式分別為儲存圖片的路徑名稱、經過人工標註框選出來的四個座標點，最後就是框選出來後的那個類別名稱所對應的數字，表 1 為經過整合後的輸入格式範例，在丟進模型訓練之前也做了 resize 的動作，將圖片壓縮到 640*640 像素，以便後續 YOLOv7 訓練。

表 1 YOLOv7 輸入格式範例

圖片路徑	C:\Users\MB20705\Desktop\machine_learning\ML_HW3\yolov7-pytorch-master\VOCdevkit/container/JPEGImages/訓練集/image_0001.jpg
座標	814,33,998,122
類別	0

■ 文字辨識預處理

首先透過 YOLOv7 模型預測出來的車牌框進行裁剪，因為只需要保留被框出來的貨櫃編號就好，其餘地方是我們不需要的，再來就是將裁剪後的圖片進行轉灰階和圖像增強等技術，讓模型能辨識的更好，圖 4 為經過預處理後的圖片。

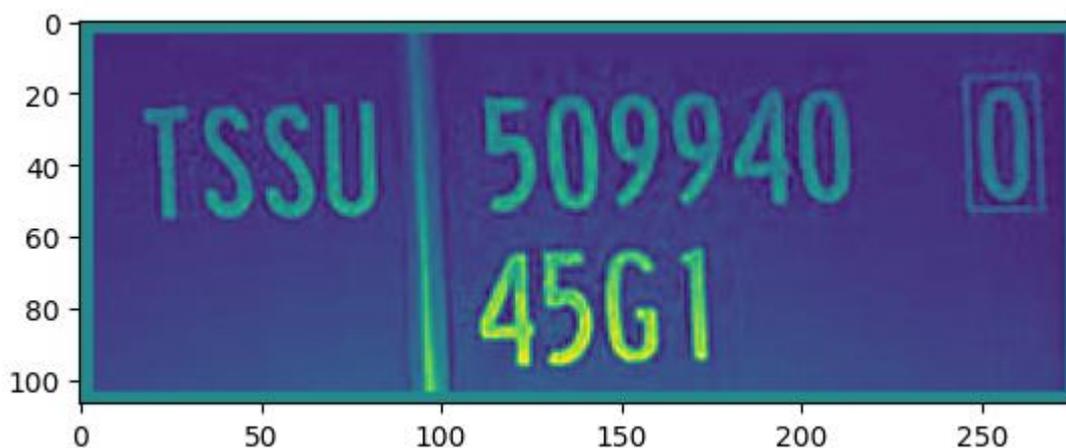


圖 4 預處理後的圖片

3.3 實驗設計



圖 5 實驗設計流程圖

本實驗設計流程分為兩個部分：

- I. 使用貨櫃號碼資料集內訓練和驗證集進行物件偵測的訓練，再透過測試資料集檢測模型性能。最後，將物件偵測所抓取的區域圖像，透過文字辨識（本實驗採用 EasyOCR）將該區域的貨櫃號碼辨識出來，並計算準確率。
- II. 讀取影片資料集內的資料，並根據第一部分，使用物件偵測模型和文字辨識來辨識影片內每幀的圖片的貨櫃號碼，最後計算其準確率。

實驗流程一開始以貨櫃車的圖片作為物件偵測模型(YOLOv7)的輸入，模型訓練前，將圖片壓縮至適合模型訓練的大小，訓練完成後，使用測試資料集檢測模型性能，並根據模型績效調整其超參數的設定。隨後將物件偵測所抓取的區域圖像，使用 EasyOCR 轉成文字，並用人工比對的方式計算其準確率，在進行文字辨識前，針對所抓取的區域圖像做對比度、飽和度等圖像增強技術，此外還有使用濾波器對圖像做平均濾波。接著，針對影片資料集中的貨櫃車影片，對每幀影像做物件偵測及文字辨識並計算偵測的貨櫃號碼的準確率。

3.4 實驗結果

3.4.1 物件偵測模型

本研究使用 YOLOv7 模型做為抓出貨櫃貨櫃編號，在超參數設定上面，首先使用了網路上提供的預訓練模型作為模型基底，在訓練模型中也使用了凍結參數的方式來訓練，因為使用了預訓練模型就代表模型其實已經在大量的資料上學習過了，加上很多研究都認為不管是什麼樣的任務，前面幾層網路所學習到的特徵都是差不多，因此要用在自己的任務上面其實不需要重頭開始學習。

在 YOLOv7 中，本研究設定的最大 epoch 為 300，前 50 個 epoch 我們將模

型的 Backbone 設為凍結，代表它是不進行訓練的，只訓練 Backbone 其它的網路層，50 個 epoch 之後才是整個模型進行訓練，本研究也設定了 early stopping 將 patient 設為 5，在優化器方面選擇的是隨機梯度下降，學習率初始值設為 0.01，會隨著訓練過程動態調整，batch_size 則是根據是否有凍結參數，有的話 batch_size 設為 8，沒有的話則設為 4，損失函數使用的是 SiLU。

圖 6 為 YOLOv7 模型架構圖，分別展示 Backbone、Head 和每一層裡面分別代表有哪些 layer 組成，表 2 為 early stopping 停止在 epoch 為 57 時模型訓練最好的各個指標績效，圖 7 為模型的訓練過程，圖 8 為觀察設定的閾值對各個指標的影響和類別的目標檢測模型在平均精度方面的性能表現。

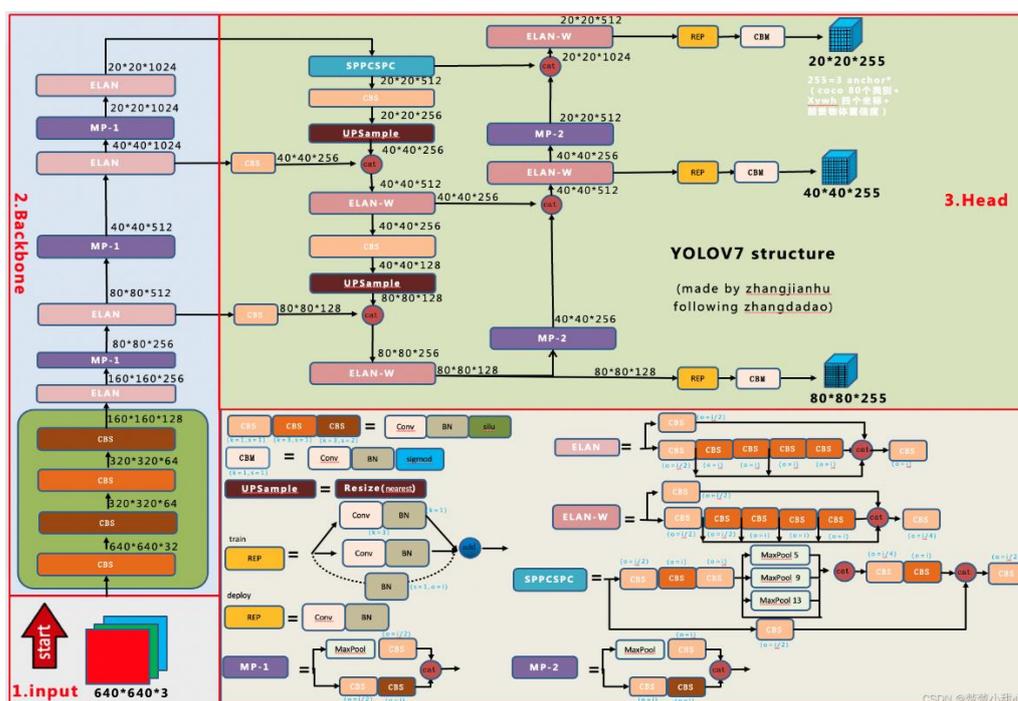


圖 6 YOLOv7 模型架構圖

表 2 模型最好績效(驗證集為準)

loss	Precision	Recall	mAP	F1-score
0.0107	97.1014	100.0	98.9072	0.9853

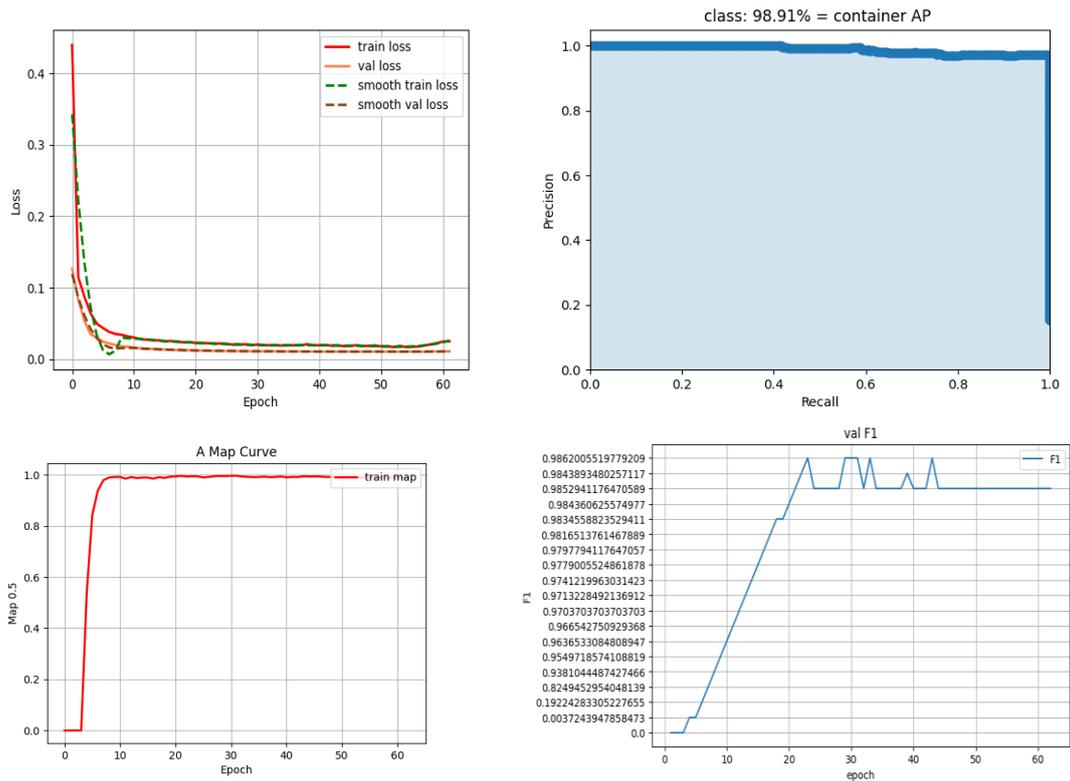


圖 7 模型訓練過程的各個指標

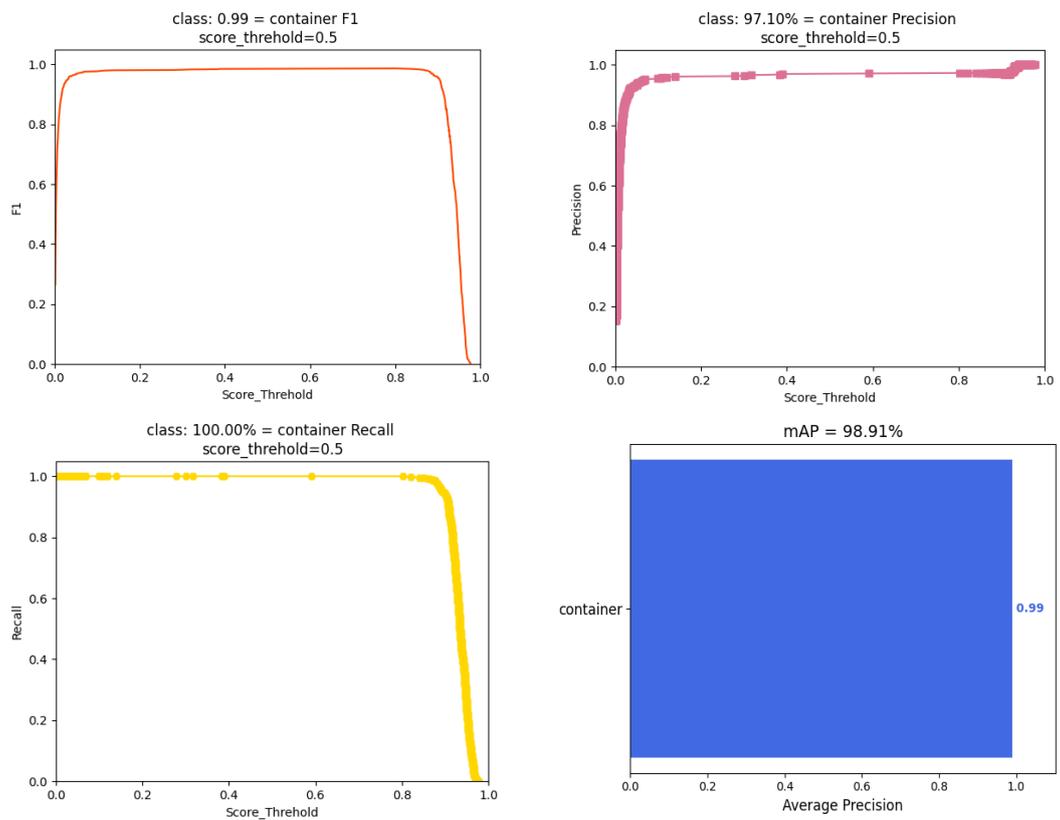


圖 8 閾值對指標的影響以及這個類別在平均精度的性能

訓練好模型後，再來拿測試資料集上的資料來評估模型的績效好不好，圖 9 為前 6 張測試集的預測結果。

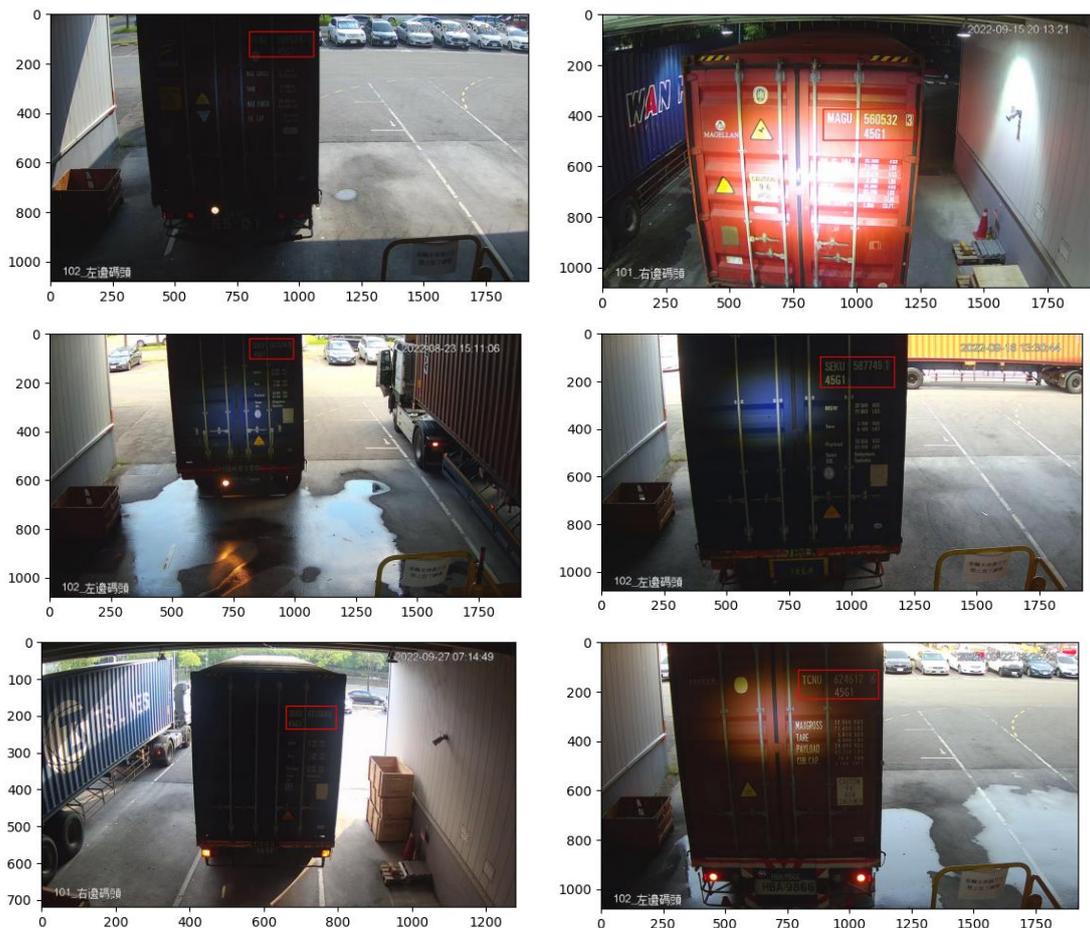


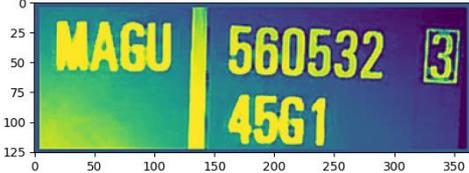
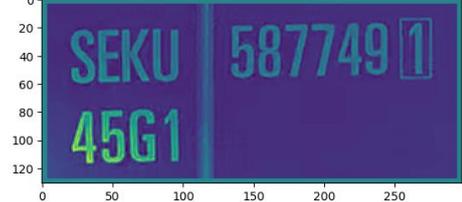
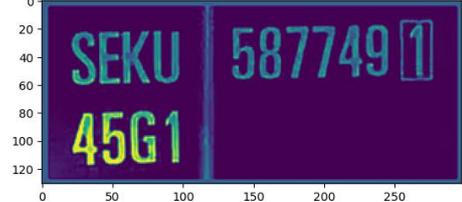
圖 9 前 6 張測試集的預測結果

3.4.2 OCR 模型辨識

在偵測到車牌之後，再來就是需要去辨識車牌裡面的號碼，本研究使用 EasyOCR，它是一個開源的光學字符識別（OCR）庫，它基於深度學習技術從圖像中提取文本。在進行辨識之前也做了前處理包括轉成灰階和圖像對比增強，希望模型可以不受其外在干擾影響辨識，像是照明度或是雜訊等，由於辨識貨櫃編號只需辨識前面 11 個號碼，加上第 11 個號碼是可以透過前面 10 個號碼推出來的，因此本研究主要關注在前 10 個號碼有無預測正確，第 11 個號碼是透過公式推出來的，表 3 為經過處理後的影響、預測出來的號碼和辨識出來的號碼。

表 3 預測出來的號碼和實際答案的比對(抽取 4 張來比對)

Image	Predict numbers	True numbers
	IFFAU289597	FFAU2895947

	MAGU5605323	MAGU5605323
	SEKU5875349	SEKU5875349
	SEKU5877491	SEKU5877491

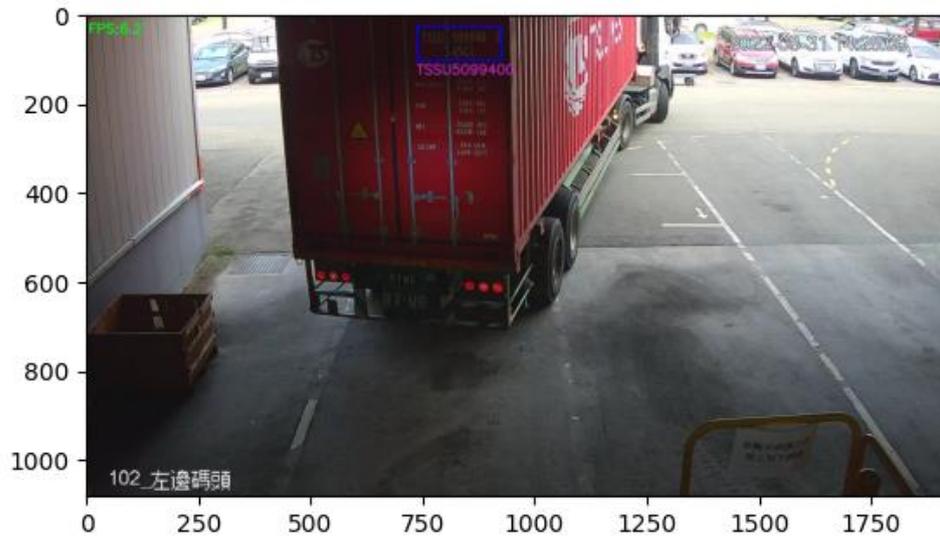
最後將總體測試集的資料拿來算準確率有多少，只取前 11 個號碼做比對。

- 測試及資料筆數：35
- 前 11 個號碼都一樣的數：32
- 準確率：91%

3.4.3 貨櫃車影片測試

最後本研究將拿 10 部貨車停車的影片來測試 OCR 辨識模型的績效，前處理的部分都如同前面測試集的图片一樣，在跑影片的時候，本研究將影片切成每一幀數丟進模型進行預測，在計算影片每一幀準確率時，是當物件偵測模型有偵測到車牌的時候，才會去看辨識的號碼是否跟標準答案是一樣的，然後去計算準確率反之則不列入計算，圖 10 為拿影片 video_0002 這部來測試辨識效果如何，每次取 30 幀來當作一張圖片顯示。完整的影片會上傳至 github。





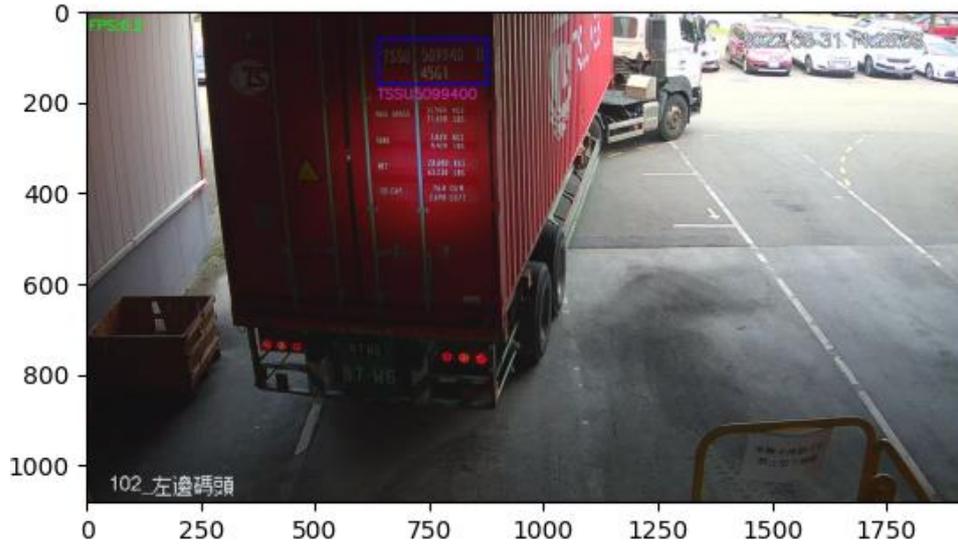




圖 10 影片辨識結果(每 30 幀取一次)

下表 4 為分別 10 部影片跑的結果，包含偵測到車牌且辨識正確車牌的準確率和整支影片跑完出現過最多次的辨識號碼做為我們最終預測結果。

表 4 10 部影片經多數決結果

	偵測到車牌的幀數有幾張	偵測到車牌且辨識正確車牌的準確率	多數決後產生的最終辨識結果	真實答案
video 0001	105	61.9%	SEKU5875349	SEKU5875349
video 0002	195	90.4%	TSSU5099400	TSSU5099400
video 0003	175	36.6%	WHSU5927851	WHSU5927851
video 0004	305	68.9%	WHSU6728690	WHSU6728690
video 0005	467	57%	WHSU5368199	WHSU5368199
video 0006	100	0%	GN24317213	WHSU2483178
video 0007	338	29%	WHSU6167120	WHSU6167120
video 0008	108	90.7%	WHSU6892256	WHSU6892256
video 0009	69	14.5%	WIISU522370	WHSU5223791
video 00010	441	76.6%	TLLU4080736	TLLU4080736

經過測試了 10 部影片，其中有 8 支影片的最終多數決答案是和標準答案一致的，單純只算偵測到車牌且辨識正確車牌準確率的平均為 52.56%，就結果來說其實還不錯了，但是可以更好。

結論

本研究的主要目標是運用 YOLOv7 模型進行物件偵測，以框出貨櫃車的編號，再利用 EasyOCR 模型進行被框起的貨櫃編號識別。對於物件偵測模型的表現，無論從指標或實際測試資料的預測結果來看，其效果均相當不錯。由實驗結果的視覺化指標可觀察到，召回率和精確率呈現負相關，當召回率較高時，精確率明顯下降。透過設定 `score_threshold` 閾值，可以平衡模型檢測結果的準確性和召回率，較高的閾值確保保留高置信度的檢測結果，提高準確性，但可能會錯過一些低置信度但仍正確的檢測結果；而較低的閾值則可增加召回率，但可能引入一些置信度較低的錯誤檢測結果。因此，選擇適當的閾值能獲得更好的 F1-score。

在 EasyOCR 模型的識別實驗中，發現其效果良好，僅有 3 張圖片出現錯誤識別。其中，有一張圖片在物件偵測階段未能正確框出車牌，導致識別錯誤。另外兩張圖片中，一張屬於較模糊的號碼，另一張則是多預測了一個號碼。在前處理階段，我們採用了圖像對比、閉運算和影像平滑等技術。

儘管整體準確率相當高，但目前仍無法解決影像模糊的問題，對於將完整影片逐幀輸入模型進行車牌識別，其準確率並未達到預期，僅約 50% 左右，可能的原因是影片場景的多樣性導致識別率下降。因此，如果能針對識別模型進行自我訓練，或許能提升準確率。無論如何，以多數決的方式讓模型對完整影片進行識別，並選擇出出現次數最多的號碼作為結果，結果顯示在 10 部影片中有 9 部識別結果正確，效果已達到相當良好的水準。

參考文獻

Bubbliiiing(2022)。這是一個 YOLOv7 的庫，可以用來訓練自己的數據集。

<https://github.com/bubbliiiing/yolov7-pytorch>

JaidedAI(2023).Ready-to-use OCR with 80+ supported languages and all popular writing scripts including Latin, Chinese, Arabic, Devanagari, Cyrillic and etc.

<https://github.com/JaidedAI/EasyOCR>

Tommy Huang(2018)。深度學習-物件偵測:You Only Look Once (YOLO)。

<https://chih-sheng->

[huang821.medium.com/%E6%B7%B1%E5%BA%A6%E5%AD%B8%E7%BF%92-%E7%89%A9%E4%BB%B6%E5%81%B5%E6%B8%AC-you-only-look-once-yolo-4fb9cf49453c](https://chih-sheng-huang821.medium.com/%E6%B7%B1%E5%BA%A6%E5%AD%B8%E7%BF%92-%E7%89%A9%E4%BB%B6%E5%81%B5%E6%B8%AC-you-only-look-once-yolo-4fb9cf49453c)