

# 機器學習 專案作業三

組員

M11221004 侯郡凌

M11223040 邱琳恩

中華民國一一三年五月三十日

## 摘要

本研究旨在比較 YoloV5 和 YoloV8 兩種物件偵測模型在貨櫃號碼識別上的表現。隨著貨櫃流通量的增加，準確識別貨櫃號碼對提高物流效率至關重要。本研究使用貨櫃影像資料集，並評估了兩種模型的 Precision、Recall、F1-score 和平均精度 (mAP)。結果顯示，兩種模型在大多數指標上表現良好，但 mAP 略遜。此外，文字辨識的準確度也不盡理想，但這些發現提供了優化貨櫃號碼識別系統的重要參考資訊。研究中使用 Yolo 系列的模型進行物件偵測，並使用 OCR 技術實現貨櫃號碼辨識任務，OCR 文字辨識技術中準確率不如預期。未來的改進方向包括微調模型及引入語言模型來校正辨識出的文字，以提升整體績效。

關鍵字：物件偵測、YoloV5、YoloV8、貨櫃號碼、OCR 文字辨識

## 一、緒論

### 1.1 動機

本研究的動機在於探討不同物件偵測模型在貨櫃號碼資料集上的實際表現。隨著全球貿易的快速發展，貨櫃的使用變得愈加普及，而對貨櫃號碼的準確識別和管理顯得尤為重要。準確的貨櫃號碼識別不僅能提升物流效率，還能減少錯誤和成本。因此，本研究將使用物件偵測模型，如 Yolo 模型，來對貨櫃影像進行自動化的偵測，準確框出包含貨櫃號碼的位置。

為了進一步提高識別準確度，本研究結合 OCR 技術，從偵測到的區域中提取並辨識貨櫃號碼。這種方法不僅可以大幅提高工作效率，還能在一定程度上替代人工作業，減少人為錯誤的可能性。同時，本研究比較不同物件偵測模型在此應用場景下的表現。以下是本研究的主要動機：

1. 提升辨識效率：自動化系統能快速處理大量貨櫃圖像，縮短辨識時間。
2. 降低人力成本：減少對人工檢查、紀錄的依賴，進而節省人力資源並降低成本。
3. 提高準確性：物件偵測和 OCR 技術的結合，能夠減少人工操作中的錯誤，提高辨識的準確度。
4. 高適應性：Yolo 物件偵測模型具有較強的適應性，能夠處理各種不同角度和光線條件下的貨櫃圖像，增加系統的可靠性。
5. 技術可行性：現有的 Yolo 物件偵測模型和 OCR 技術已經相對成熟，具備較高的應用價值，能夠實現較為穩定的貨櫃號碼辨識效果。

因此，本研究期望透過 Yolo 物件偵測模型與 OCR 文字辨識的結合，改善

貨櫃管理，提升整體的辨識效率，提供一種更佳解決方案。

## 1.2 目的

本研究旨在透過 Yolo 系列模型建立物件偵測系統，並搭配 OCR 進行貨櫃號碼識別，評估和比較不同物件偵測模型在貨櫃號碼資料集上的性能，以識別出最適合此資料集的模型。以下為本研究的主要目標：

1. 比較不同 Yolo 物件偵測模型的性能：評估不同 Yolo 物件偵測模型在貨櫃號碼偵測中的準確性和效率，選出最適合的模型。
2. 模型性能評估：評估各模型在貨櫃影像偵測中的準確率、召回率、F1-score 以及 mAP 等指標，並分析其在實際應用中的優劣。
3. 結合 OCR 技術進行辨識：利用 OCR 技術，從 Yolo 物件偵測模型預測出的區域中精確識別貨櫃號碼，並分析整合後的整體系統性能。
4. 對比分析：比較不同物件偵測模型的性能，識別出在準確性、效率和穩定性方面表現最優的模型。

透過達成上述目標，本研究希望能夠為物流業提供高效率、準確的貨櫃號碼自動識別模型，進一步推動物流業自動化發展，提升行業整體運營效率。

## 二、 方法

本研究使用 YoloV5 和 YoloV8 作為物件偵測模型、選用 Tesseract 作為文字辨識模型，對有貨櫃號碼的影像資料集做偵測以及文字提取，程式架構主要包含影像資料集的讀取、影像解析度的處理、真實框標籤轉換、模型導入、模型訓練和驗證、模型績效評估以及文字辨識等步驟。

### 針對 YoloV5 模型：

首先，將資料集導入後透過圖像處理技術進行預處理，將圖片大小調整至 416\*416 大小，並將標籤檔轉換為符合 Yolo 格式，並建立相關的 yaml 檔案，定義物件類別數量 nc，以及物件名稱 name，以供模型建立使用。模型選用了 Ultralytics 開發的 YoloV5 模型架構進行訓練，YoloV5 模型總層數為 25 層，包含了骨幹網路(backbone)和頭部(head)部份。在模型訓練和預測部份，保存預測框的座標以計算 IoU，且針對重複出現的預測框進行處理，只保留最先出現的預測框。在模型績效評估部份，使用 mAP、Recall、Precision 和 F1-score 指標。在文字辨識部份，先進行裁切預測框區域後，接著透過二值化技術將裁切圖像轉換成利於辨識的形式，最後將圖像中的文字識別為可理解的文本形式。在文字辨識的過程中，透過限制貨櫃號碼長度為 15 字，以及過濾非英文和數字符號部份，提高辨識準確率。

### 針對 YoloV8 模型：

本研究首先將貨櫃號碼影像以及影像資料集導入，並將解析度調整為 416x416 以便模型處理。接著，將 XML 格式的真实標籤轉換為 YoloV8 能使

用的格式。

模型建置方面，使用 YoloV8n，其包含 225 層和 3,011,043 個可訓練參數。進行模型訓練和預測時，保存預測框的座標以計算 IoU。預測框處理的部分，使用 Non-Maximum Suppression (NMS) 移除重複的預測框，保留分數最高的框，若無預測框，則補值(0)。模型績效評估使用 Precision、Recall、F1 Score 和 mAP。

文字辨識部分，裁切預測框區域，轉換為灰階後用大津法進行二值化，並使用 Tesseract 進行文字辨識，透過限制貨櫃號碼長度為 15 字和過濾非英文和數字符號，提高辨識準確率。

### 三、實驗

#### 3.1 資料集

本研究使用貨櫃資料集、影片資料集、圖片準確率資料集。在貨櫃資料集中分為六個檔案，分別為用於訓練、驗證、測試的圖檔和標記檔，訓練集數量為 2125 筆，驗證集資料為 536 筆，測試集資料為 755 筆，如表 1 所示。在影片資料集包含 10 個 avi 檔，用於物件偵測影像測試，如表 2 所示。在圖片準確率資料集包含 35 張圖檔，用於計算 OCR 文字辨識準確率，如表 3 所示。

表 1、貨櫃資料集


資料集名稱	資料定義	資料型態	數量	資料 1
訓練集	訓練集圖檔	jpg	2125	
驗證集	驗證集圖檔	jpg	536	
測試集	測試集圖檔	jpg	755	
訓練集_xml	訓練目標標籤	xml	2125	<pre>&lt;?xml version='1.0' encoding='utf-8'&gt; &lt;annotation&gt;   &lt;folder&gt;訓練集&lt;/folder&gt;   &lt;filename&gt;image_0001.jpg&lt;/filename&gt;   &lt;path&gt;E:\貨櫃訓練資料\Yolov4(1)_資料\訓練集\image_0001.jpg&lt;/path&gt;   &lt;source&gt;     &lt;database&gt;Unknown&lt;/database&gt;   &lt;/source&gt;   &lt;size&gt;     &lt;width&gt;1920&lt;/width&gt;     &lt;height&gt;1080&lt;/height&gt;     &lt;depth&gt;3&lt;/depth&gt;   &lt;/size&gt;   &lt;segmented&gt;0&lt;/segmented&gt;   &lt;object&gt;     &lt;name&gt;container&lt;/name&gt;     &lt;pose&gt;Unspecified&lt;/pose&gt;     &lt;truncated&gt;0&lt;/truncated&gt;     &lt;difficult&gt;0&lt;/difficult&gt;     &lt;bndbox&gt;       &lt;xmin&gt;814&lt;/xmin&gt;       &lt;ymin&gt;33&lt;/ymin&gt;       &lt;xmax&gt;908&lt;/xmax&gt;       &lt;ymax&gt;122&lt;/ymax&gt;     &lt;/bndbox&gt;   &lt;/object&gt; &lt;/annotation&gt;</pre>
驗證集_xml	驗證目標標籤	xml	536	
測試集_xml	測試目標標籤	xml	755	

表 2、影片資料集

資料集名稱	資料型態	數量	資料 1
影片資料集	avi	10	

表 3、圖片準確率資料集

資料集名稱	資料型態	數量	資料 1
圖片準確率資料集	jpg	35	

### 3.2 前置處理

本研究針對兩種不同的 Yolo 物件偵測模型進行圖像和標籤的前置處理。

#### 針對 YoloV5 模型：

首先，如圖 1，使用了 `resize()` 方法將圖像調整為 416\*416 大小，並另存至 `images/train`、`images/val`、`images/test` 路徑。接著，讀取訓練集 `_xml`、驗證集 `_xml`、測試集 `_xml`，透過 `ElementTree` 方法解析 xml 文件，讀取 xml 文件中物件的寬度、高度、類別、邊界框等資訊，並使用 `convert` 函數將物件的邊界框座標轉換成 Yolo 格式，並將轉換結果另存至 `labels/train`、`labels/val`、`labels/test`。xml 轉換成 Yolo 格式的結果如圖 2 所示。

#### 針對 YoloV8 模型：

導入貨櫃號碼影像資料集後，如圖 1，使用 `img.resize()` 方法將影像資料集解析度調整為 416\*416 以方便模型處理。接著，使用 `xml.etree.ElementTree` 套件將真實標籤的 XML 格式檔案轉換成 YoloV8 可用格式，以便後續計算 IoU（影像名稱 類別 座標 1 座標 2 座標 3 座標 4），如圖 2。



圖 1、影像解析度轉換

```
<?xml version="1.0" encoding="UTF-8"?>
<annotation>
  <folder>訓練集</folder>
  <filename>image_0001.jpg</filename>
  <path>E:\貨櫃訓練資料\yolov4(1)_資料\訓練集\image_0001.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>814</xmin>
      <ymin>33</ymin>
      <xmax>998</xmax>
      <ymax>122</ymax>
    </bndbox>
  </object>
</annotation>
```

0 0.471875 0.07175925925925926 0.09583333333333333 0.08240740740740742

圖 2、xml 檔轉換成 Yolo 格式

### 3.3 實驗設計

本研究採用 Ultralytics 公司所開發的 Yolo 物件偵測模型進行訓練。在模型訓練的過程中，使用了貨櫃資料集的訓練集，同時使用驗證集進行驗證，以確保模型的泛化能力和效能。在文字辨識部分，則使用了圖片準確率資料集，並利用已訓練好的 Yolo 物件偵測模型搭配 OCR 技術進行實作。根據預測框的座標將圖像進行裁切，然後利用二值化技術將裁切圖像轉換成利於辨識的形式，最後將圖像中的文字識別為可理解的文本形式。最後，以 mAP、Recall、Precision 和 F1-score 指標評估物件偵測模型的績效，並以準確率評估 OCR 文字辨識的績效。為了優化模型性能，進行了一系列實驗，調整超參數組合，例如 batch size、epoch 等，以最大程度優化模型。

#### 針對 YoloV5 模型：

本研究選用 Ultralytics 公司的 YoloV5 模型進行實作，在模型設定檔部分，可得知採用了深度為 6 層的卷積層來提取特徵，然後進行一系列的 C3 卷積操作，模型在訓練時的預測框表現如圖 3 所示。而在模型超參數設定方面，本研究使用模型自行尋找超參數組合的方式，皆使用 Adam 作為優化器，學習率部分則選用 0.002 來進行訓練。在批次大小和訓練週期的調整部份，批次大小的預設數值為 16，後續本研究以 2 的倍數進行調整，而訓練週期則是從預設的 20 調降為 5、8、10 這幾個參數進行實驗，最終選定批次大小與訓練週期皆為 8 的



參數組合，如圖 4 所示，該組合在模型訓練以及驗證的過程中表現出良好績效。在文字辨識部份，表 4 為 OCR 文字辨識的整體流程，首先根據預測框的區域進行圖像裁切，接著將裁切結果進行二值化處理，讓裁切圖像轉換成利於辨識的形式，接著將圖像中的文字識別為可理解的文本形式，為了提升辨識準確率，使用了 `re.sub()` 方法過濾非英文和數字符號，以及將貨櫃號碼長度限制為 15 字。

### 針對 YoloV8 模型：

如圖 5 所示，本研究在預測過程中，發現有多個預測框，因此在 YoloV8 模型使用 Non-Maximum Suppression (NMS) 處理預測框，去除重複的預測框，僅保留與真實框重疊 IoU 最高者，若無預測框，則補值為 (0,0,0,0)。且使用 Tesseract 模型進行 OCR 文字辨識，如表 5 所示，首先，將具有預測框的區域進行裁切，轉換成灰階模式並進行大津法 (Otsu) 二值化處理，並限制貨櫃號碼長度為 15 字，並使用 `re.sub()` 方法過濾非英文和數字符號，以提高準確率。

在模型建置過程，本研究發現若未設置超參數，則 YoloV8 模型會自動找尋超參數組合，因此本研究模型皆使用模型自身找尋的最佳優化器 Adam (動量=0.9) 以及學習率 0.002 進行訓練，僅調整批次大小和訓練週期。且基於資料集龐大的考量，一開始本研究將批次大小設置為 16，接著根據需要以 2 的倍數進行調整；而訓練週期則是採用以 5 的倍數進行調整，初始值為 20。但隨著實驗進行，本研究發現訓練時間過長，因此進行了批次大小為 4、8；訓練週期為 10 和 15 的測試。以這些超參數為例的訓練和驗證結果，以批次大小為 8、訓練週期為 10 為例，如圖 6 所示，在模型訓練以及驗證的過程顯示出良好的表現。

綜合上述，本實驗設計旨在結合物件偵測和文字辨識技術，通過對模型性能的綜合評估和超參數調整，提高模型的準確性和效能。

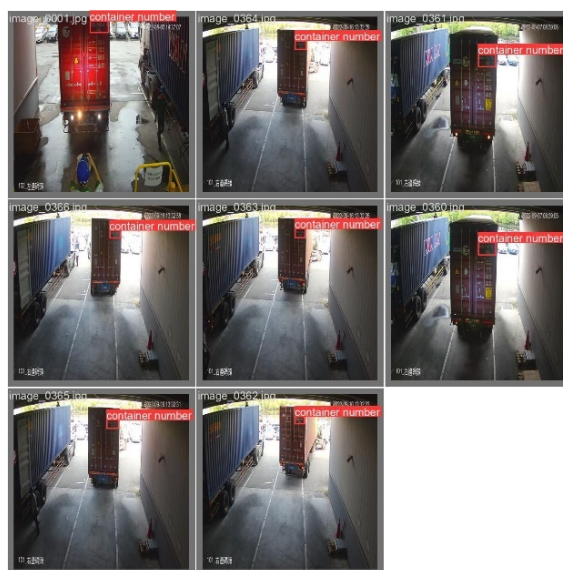


圖 3、YoloV5 模型訓練預測框表現

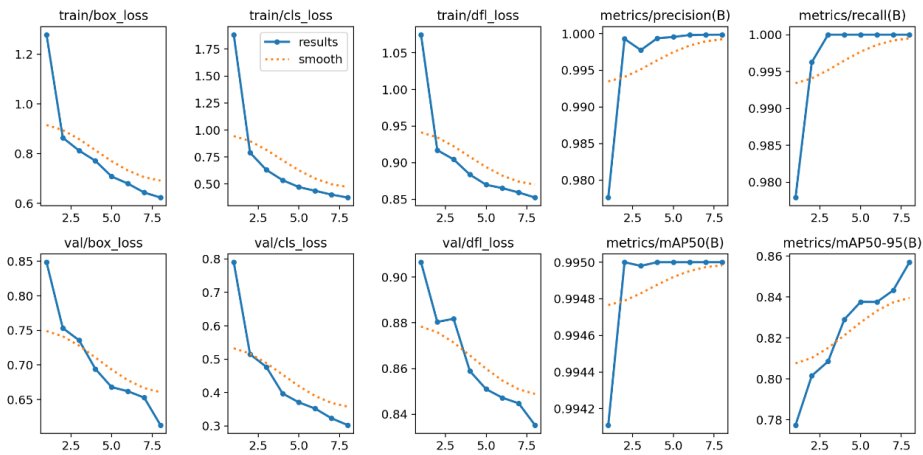


圖 4、YoloV5 模型訓練和驗證績效表現

表 4、YoloV5 OCR 辨識過程



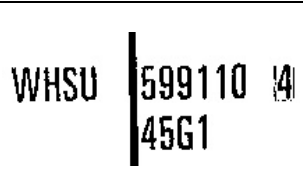
模型預測結果	裁切預測結果	影像二值化	辨識結果	標準答案
			WHSU59911044561	WHSU599110445G1



圖 5、YoloV8 模型之兩個預測框

表 5、YoloV8 OCR 辨識過程

模型預測結果	裁切預測結果	影像二值化	辨識結果	標準答案
			WHSU5368199AG	WHSU5368199 45G1



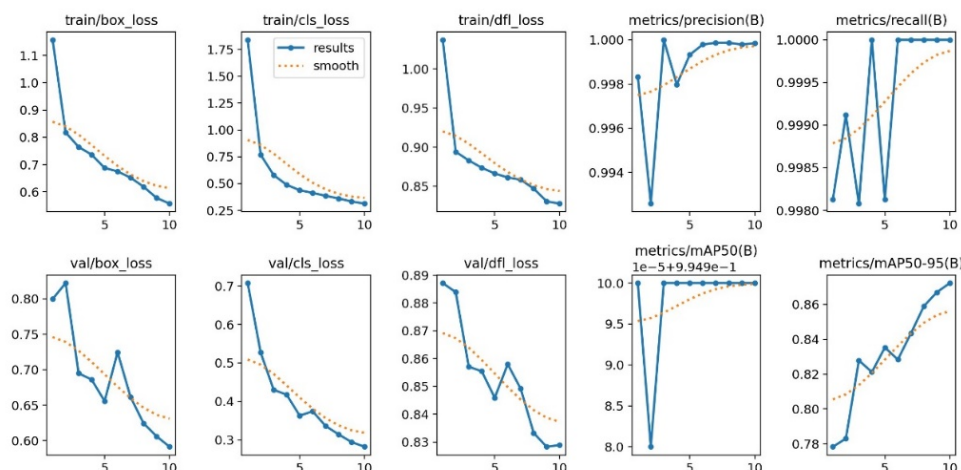


圖 6、YoloV8 模型訓練和驗證績效表現

### 3.4 實驗結果

#### 針對 YoloV5 模型：

如表 6 所示，可發現貨櫃號碼偵測的整體績效表現良好，其中第二組參數組合表現優秀，雖然在 mAP 績效部分低於第一組參數，但在 Precision、Recall、F1-score 指標上的表現優異，整體綜合表現最佳。而文字辨識的結果如表 8 所示，最高的準確率為 0.3143，該數值也是來自表 6 的第二組參數組合，因此在 YoloV5 模型的貨櫃號碼辨識實驗當中，批次大小為 8、訓練週期為 8 的參數組合的績效表現最好。

綜合結果來說，本研究認為 YoloV5 在物件偵測表現上良好，但在 mAP 績效部分仍然有可以進步的空間。而在文字辨識部分，因最高文字辨識準確率只有 0.3143，認為需要進行進一步的調整與優化，以實現更好的績效表現。

#### 針對 YoloV8 模型：

貨櫃號碼偵測績效，如表 7 所示，可以發現整體模型績效表現良好，除了 mAP 的部分還可以待改進，其中為第三組超參數設置為最佳，其批次大小為 8、訓練週期為 15 在整體表現上最好，但 mAP 未達 80%，因此本研究認為還有調整的空間。

使用 Tesseract 模型的辨識績效，如表 8 所示，總樣本數為 35 張影像，整體準確率偏低，本研究認為可以再調整影像處理的方式，如強化影像的對比度，或是更換 OCR 文字辨識模型，以達到更好的績效。

表 6、YoloV5 績效表現

YoloV5	Batch size	Epochs	Precision	Recall	F1-score	mAP
1	16	8	0.9918	0.9563	0.9737	0.7957
<b>2</b>	<b>8</b>	<b>8</b>	<b>1</b>	<b>0.9815</b>	<b>0.9906</b>	<b>0.7918</b>
3	8	5	0.9837	0.9590	0.9712	0.7911
4	8	10	0.9973	0.9669	0.9818	0.7879

表 7、YoloV8 績效表現

YoloV8	Batch size	Epochs	Precision	Recall	F1-score	mAP
1	16	20	0.9858	0.9302	0.9572	0.7378
2	8	20	1.0000	0.9033	0.9492	0.7279
<b>3</b>	<b>8</b>	<b>15</b>	<b>0.9987</b>	<b>0.9934</b>	<b>0.9960</b>	<b>0.7876</b>
4	8	10	0.9959	0.9668	0.9811	0.7534
5	4	15	0.9890	0.9652	0.9770	0.7668
6	4	10	0.9873	0.9383	0.9622	0.7233

表 8、Tesseract OCR 文字辨識結果

YoloV5		YoloV8	
正確樣本	準確率	正確樣本	準確率
10	0.2857	<b>11</b>	<b>0.3143</b>
11	<b>0.3143</b>	7	0.2000
10	0.2857	10	0.2857
7	0.2	8	0.2286
X	X	10	0.2857
X	X	10	0.2857

#### 四、 結論

本研究針對貨櫃資料集運用不同物件偵測模型進行了深入的實驗。對於 YoloV5 模型，整體績效良好，成功偵測貨櫃號碼位置，但在 OCR 文字辨識部分表現不如預期，最佳的準確率只有 0.3143。針對此問題，需要更進一步的處理以提升整體表現。未來，本研究計劃採取一些措施來改善整體表現，尤其是文字辨識部分，透過模型微調和引入語言模型來校正辨識出的文字，以提升文字辨識準確率。此外，除了圖片偵測，也可以進行影片的物件偵測任務，實現對影片中的貨櫃號碼自動辨識和追蹤，擴展應用範圍，增加多樣性和實用性。

另外，本研究也對 YoloV8 模型的績效進行了詳細評估，結果顯示其整體表現與 YoloV5 相當，但在 mAP 方面略有不足(未達到 80%)，這可能是由於處理預測框的方式不佳或是混淆矩陣的閾值設置不當。本研究還發現模型預測時會出現兩個預測框甚至沒有預測框的情況，未來將更嚴謹地選擇評估方法，以提高模型性能；在 OCR（光學字符識別）部分的表現也不盡理想，需要進行更深入的研究，計劃採用更先進的 OCR 技術和算法，進行全面的預處理和後處理優化，以提高字符識別的準確性和效率，改進預測框處理方法，優化非極大值抑制（NMS）演算法，調整預測框的閾值設定，確保在不同場景中獲得最佳的預測結果。同時，增強評估方法，使用更大、更全面的測試數據集來全面評估模型的性能，並嘗試加入影片的即時辨識功能。

## 參考文獻

迪菲赫尔曼. (2022, November 22). YoloV5 + Tesseract-OCR 实现车牌号文本识别

\_yolo ocr-CSDN 博客.

[https://blog.csdn.net/weixin\\_43694096/article/details/127815800](https://blog.csdn.net/weixin_43694096/article/details/127815800)

喝咖啡的 CV. (2023, December 9). YOLO 模型结果中的混淆矩阵怎么看\_混淆矩

阵图怎么分析结果-CSDN 博客.

[https://blog.csdn.net/Cr\\_NanMao/article/details/134895073](https://blog.csdn.net/Cr_NanMao/article/details/134895073)

智脑未来. (2023, May 30). Labelimg 标注的 VOC 格式标签 xml 文件和 Yolo 格式

标签 txt 文件相互转换. 知乎. <https://zhuanlan.zhihu.com/p/633361433>

AI 追随者. (2023, January 11). YOLOV8 训练自己的数据集(超详细)-CSDN 博客.

Retrieved May 10, 2024, from

[https://blog.csdn.net/qq\\_40716944/article/details/128648001](https://blog.csdn.net/qq_40716944/article/details/128648001)

Chang, C. (2023, August 16). YOLOV8 模型訓練及其指標意義 – Claire's blog.

<https://claire->

[chang.com/2023/08/16/yolov8%E6%A8%A1%E5%9E%8B%E8%A8%93%E7](https://claire-chang.com/2023/08/16/yolov8%E6%A8%A1%E5%9E%8B%E8%A8%93%E7)

[F%E7%BE%A9/](https://claire-chang.com/2023/08/16/yolov8%E6%A8%A1%E5%9E%8B%E8%A8%93%E7%B7%B4%E5%8F%8A%E5%85%B6%E6%8C%87%E6%A8%99%E6%84%8)

[F%E7%BE%A9/](https://claire-chang.com/2023/08/16/yolov8%E6%A8%A1%E5%9E%8B%E8%A8%93%E7%B7%B4%E5%8F%8A%E5%85%B6%E6%8C%87%E6%A8%99%E6%84%8F%E7%BE%A9/)

Chou, E. (2023, January 20). YOLOv8 完整詳細介紹以及教你如何手把手訓練自訂

義模型 | Medium. Medium. <https://medium.com/@EricChou711/yolov8->

<https://medium.com/@EricChou711/yolov8-%E4%BB%8B%E7%B4%B9%E5%92%8C%E6%89%8B%E6%8A%8A%E6%89%8B%E8%A8%93%E7%B7%B4%E8%87%AA%E8%A8%82%E7%BE%A9%E6%A8%A1%E5%9E%8B-752d8d32cb73>

GGWithRabbitLIFE. (2019, January 14). [機器學習 ML NOTE]YOLO!!!如何簡單使

用 YOLO 訓練出自己的物件偵測!!! (Windows+Anaconda). Medium.

<https://medium.com/%E9%9B%9E%E9%9B%9E%E8%88%87%E5%85%94%E5%85%94%E7%9A%84%E5%B7%A5%E7%A8%8B%E4%B8%96%E7%95%8C/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-ml-note-yolo-%E5%88%A9%E7%94%A8%E5%BD%B1%E5%83%8F%E8%BE%A8%E8%AD%98%E5%81%9A%E7%89%A9%E4%BB%B6%E5%81%B5%E6%B8%AC-object-detection-%E7%9A%84%E6%8A%80%E8%A1%93-3ad34a4cac70>

iThome. (2021, September 12). 標籤圖片的方法與實作 - Day 12 - IT 邦幫忙::一

起幫忙解決難題,拯救 IT 人的一天. iT 邦幫忙::一起幫忙解決難題,拯救

IT 人的一天. <https://ithelp.ithome.com.tw/articles/10263004>

iThome. (n.d.). IT 邦幫忙::一起幫忙解決難題,拯救 IT 人的一天. iT 邦幫忙::一起幫忙解決難題，拯救 IT 人的一天.

<https://ithelp.ithome.com.tw/m/articles/10271665>

iThome. (n.d.). YOLO v8 實測 - iT 邦幫忙::一起幫忙解決難題，拯救 IT 人的一天. iT 邦幫忙::一起幫忙解決難題，拯救 IT 人的一天.


<https://ithelp.ithome.com.tw/articles/10311114>

Midnightla. (2022, October 10). 《第 25 天》YOLO 訓練流程與資料集 YOLO Txt 格式. IT 邦幫忙. <https://ithelp.ithome.com.tw/articles/10307097>


Office 指南. (2022, March 23). PyTorch 自行訓練 YOLOV5 物件偵測模型教學與範例. <https://officeguide.cc/pytorch-yolo-v5-object-egg-detection-models-tutorial-examples/>

Thomas. (2023, September 23). YOLOV8 車牌辨識 - MahalJSP. [https://mahaljsp.ddns.net/yolov8\\_car/](https://mahaljsp.ddns.net/yolov8_car/)

Ultralytics. (2024, May 18). YOLOV5. Ultralytics YoloV8 Docs. <https://docs.ultralytics.com/models/yolov5/>

Ultralytics. (n.d.). ultralytics/yolov5: YoloV5  in PyTorch > ONNX > CoreML > TFLite. GitHub. Retrieved May 24, 2024, from <https://github.com/ultralytics/yolov5>



Ultralytics. (n.d.). Ultralytics: NEW - YOLOV8  in PyTorch > ONNX >

OpenVINO > CoreML > TFLITE. GitHub.

<https://github.com/ultralytics/ultralytics>