**無人機飛行載具之智慧計數**

報告說明文件

團隊編號:

Team2086

作者

卓子揚 輔仁大學 數學系

詹茹萍 輔仁大學 數學系

紀昌賢 輔仁大學 數學所

林峻宇 輔仁大學 數學系

楊鎧丞 輔仁大學 數學系

指導老師:

陳泓勳老師

**目錄**

[壹、 環境 2](#_Toc121865780)

[貳、 演算方法與模型架構 3](#_Toc121865781)

[參、 創新性 4](#_Toc121865782)

[肆、 資料處理 6](#_Toc121865783)

[伍、 訓練方式 7](#_Toc121865784)

[陸、 分析與結論 9](#_Toc121865785)

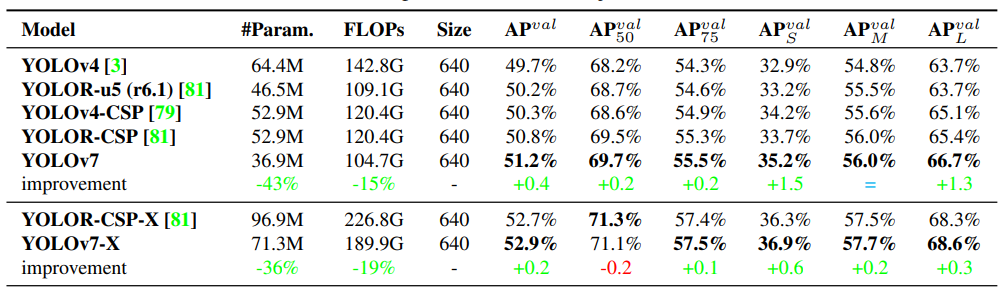
[柒、 程式碼 13](#_Toc121865786)

[捌、 使用的外部資源與參考文獻 13](#_Toc121865787)

[玖、 作者聯絡資料表 13](#_Toc121865788)

1. 環境
2. 作業系統：Ubuntu 20.04.3 LTS、Windows11
3. 顯示卡 : RTX2080 Ti、Tesla T4、Nvidia A100
4. VRAM : 10 GB
5. 語言：Python3.8
6. 套件：
7. Glob==3.8
8. matplotlib==3.5.0
9. numpy==1.19.4
10. opencv-python==4.1.1
11. os==3.8
12. Pandas==1.5.1
13. Pillow==8.4.0
14. protobuf==3.17.3
15. PyYAML==5.4.1
16. requests==2.26.0
17. scipy==1.4.1
18. shutil==3.8
19. time==3.8
20. torch==1.10.0+cu113
21. torchvision==0.11.1+cu113
22. tqdm==4.62.3
23. 預訓練模型
24. YOLOv7-X
25. 演算方法與模型架構

本團隊採用YOLOv7(以下簡稱為v7)幫助我們訓練模型。由於本次資料物件最大值絕大多數並無大於YOLO定義的小物件，其物件大小大於32\*32 pixels，故本團隊從v7的模型架構下，在顧及有限的時間運算與可運用的資源下，選擇 YOLOv7-X 作為本次的主要模型架構，以下為其架構之比較圖



由上圖可知，v7的版本模型相較於其YOLOv4 (以下簡稱為v4) 的版本，

在總參數上，v7以較少之參數(#Param.)及較低耗能下(FLOPs)，以較高的準確率優於v4，故在團隊商榷後決定使用v7。

相較於其餘兩個模型，YOLOv7-X雖然擁有較多的參數導致訓練時的時間延長，但在小物件的判別上明顯優於二者，在此數據的參考下，本團隊利用YOLOv7-X作為偵測模型。

在進行資料前處理後，將照片裁切成640\*640 pixels的圖片，共1986張放進YOLOv7-X中，並以Batch Size為9與Epoch為143的參數調整下 Learning Rate同步調整至0.001作為第一次訓練，完成後以凍結87層，Epoch 為100的參數其Learning Rate設置0.001進行模型再訓練。

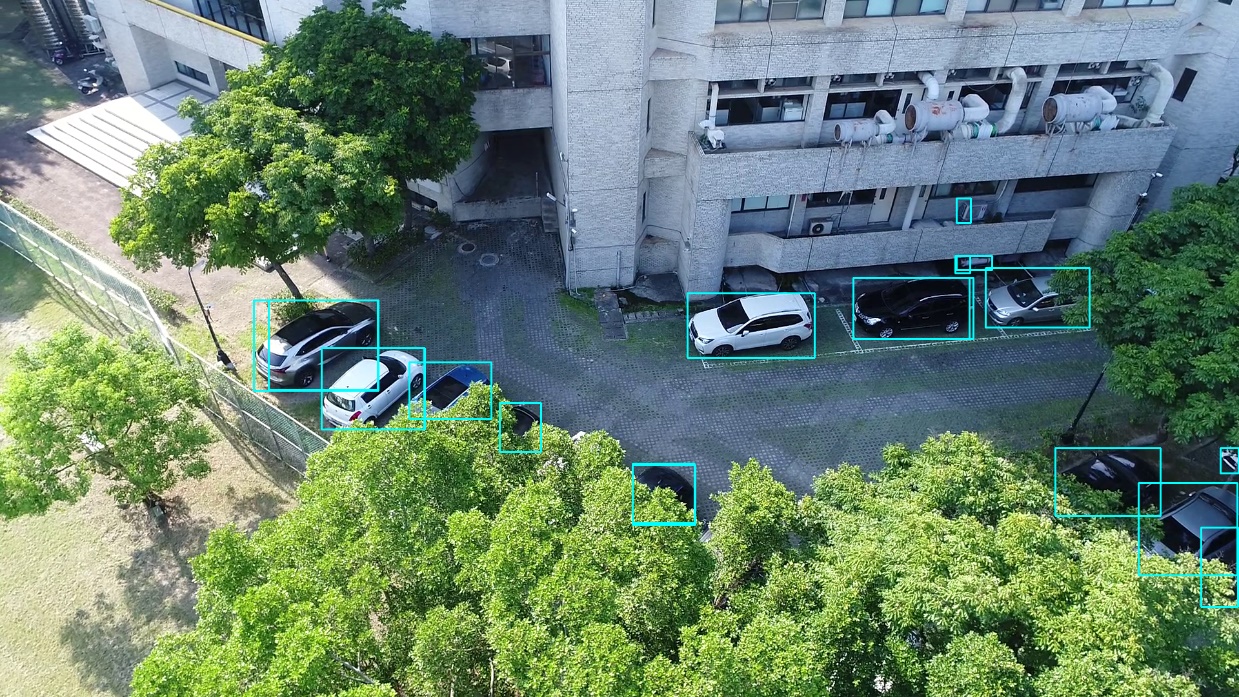
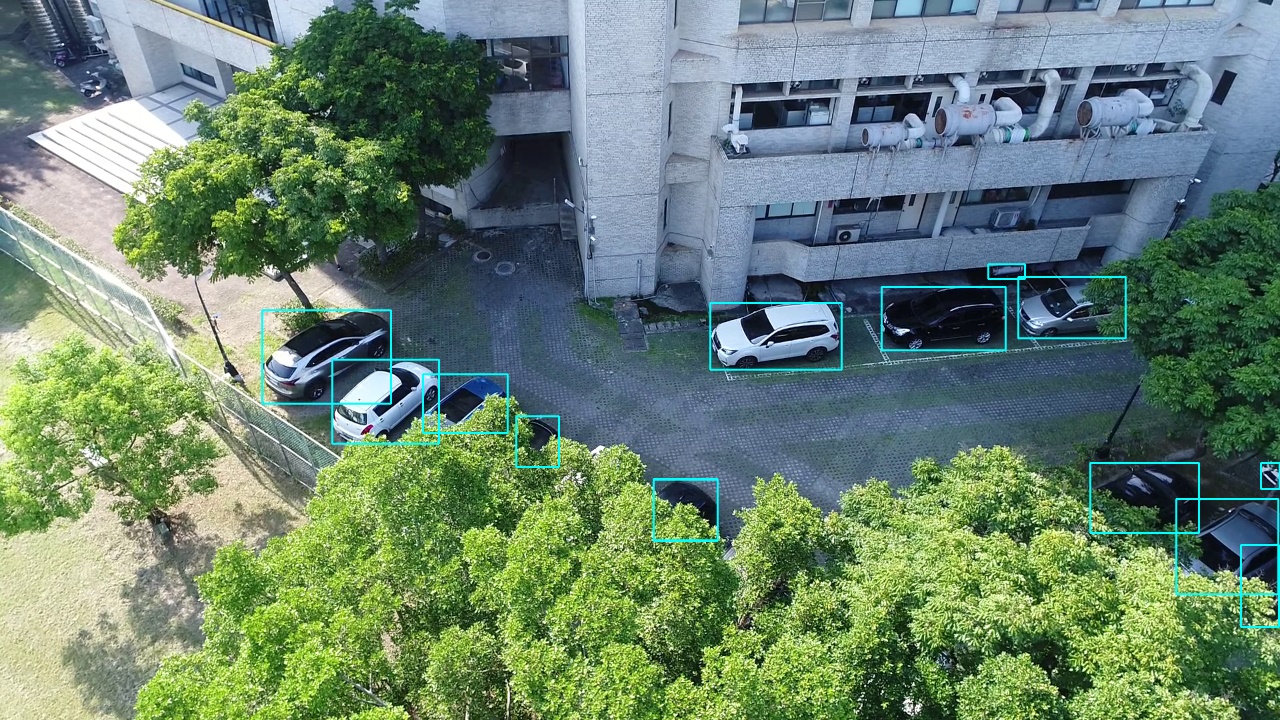
同時間，本團隊將照片大小不裁切僅縮放成寬為1088 pixels的圖片，共900張放進YOLOv7-X，Batch為4與Epoch為100，在不調整Learning Rate的參數下，以凍結58層、53層以及48層，依序對模型進行再訓練。

1. 創新性

本團隊在本次競賽中發現了主辦方給予的圖檔之框有不盡完美之處，導致在訓練模型時，會發生訓練錯誤資訊等狀況，為了彌補這樣的缺憾，本團隊有對訓練之資料進行修正，其作法如下:

將主演算法分為兩個模型進行訓練，皆使用YOLOv7-X模型進行訓練，分為縮放成寬為1088\*1088 pixels 圖檔與裁切成640\*640 pixels圖檔。前者其訓練資料首次由官方提供之標準答案進行訓練，之後的每一次皆由其自己所預測出來Train圖檔之預測框進行訓練，共訓練3次，後者所使用的資料集皆來自於官方所提供之標準答案進行訓練，共訓練兩次，此作法既可避免過多來自官方不正確的框放至模型訓練，亦可同時彌補在調整Train圖檔的框時被刪除之正確框而導致的資料不完整。

為了使Private圖檔能更好的被預測，我們將圖檔分別裁切至寬為1088 pixels與640\*640 pixels圖檔(後稱1088圖檔和640圖檔)，以供模型預測，在合併1088圖檔至原本照片大小時，將其圖像重疊像素邊界內縮128 pixels，以防止其因裁圖而導致預測不完全僅有框到車頭或車尾之狀況(**Figure1**)，同理在合併640圖檔至原本照片大小時，也將其圖像重疊像素邊界內縮50 pixels。

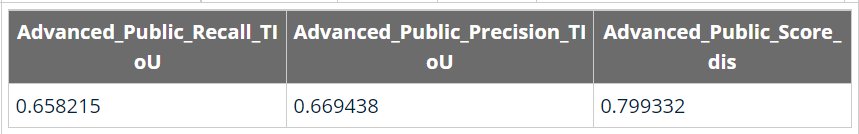
 

**Figure1 :** 未內縮處理前(左)及內縮處理後(右)

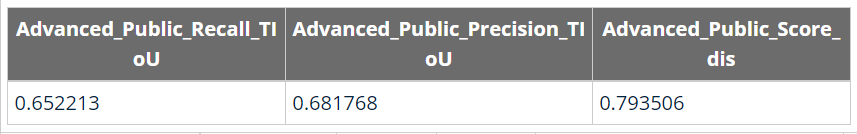
最後將兩個模型對Private圖檔進行預測之預測框進行合併，通過Non- Maximum Suppression (NMS)的IOU閥值大於0.5的部分再進行預測框之刪除，以消除因為切圖而導致同物件被多個預測框預測而導致Precision分數下降的風險(**Figure 2**)。

為了佐證本團隊的資料處理有成效，**Figure 2**為此數據的資料佐證。在不經任何資料處理前的數據儘管有較高的Recall成績，但在Precision項目的成績是最低的。內縮處理後，模型數據因刪除多餘的不完整框，Precision 提升了約1%的成績，但經過完整的處理後Precision跟原本未經任何處理前的對比，可大幅提升約3%的Precision成績。

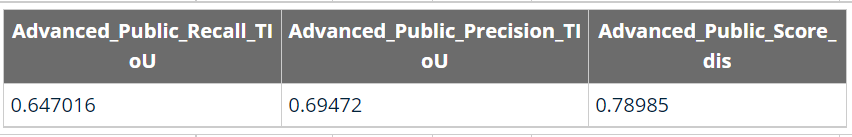
**未經資料後處理前成績**



**經過內縮處理的成績**



**經完整資料後處理後的成績**



**Figure2 : 經不同資料處理後的成績比較**

1. 資料處理

在本次競賽中，本團隊對於資料的處理主要分成了訓練前的資料預處理， 以及訓練後資料整合兩大部分。

在資料預處理的部分，本團隊進行的是對圖片切割。訓練的過程中我們 發現官方給的資料集中，大部分圖片是有雜訊的亦或是物件過小以及不清楚， 因此我們根據圖片的大小將圖片進行了分切，希望透過此方式來放大那些不 夠清楚的特徵，以便讓本團隊達到更好的訓練效果，而切割的方式則如下說 明:

1. : 分類資料

* 先將官方的訓練資料集分類出兩種大小，1920\*1080 pixels以及1344\*720 pixels兩種大小。

1. : 切割縮放圖片(對本競賽中的所有dataset進行相同裁切)

* 對於Train資料集進行兩種裁切:

裁成640\*640 pixels圖檔 (1920切8份、1344切6份)

將圖片切成8和6等份讓彼此有重疊，以避免一個物體被裁一半，並將因切圖造成的不完整框進行刪除，以避免資訊不完整的訓練。

將兩種大小的圖片縮放成1088\*1088 pixels

* 對於Public 和Private圖檔進行兩種裁切:

1. 裁成640\*640 pixels圖檔 (1920切8份、1344切6份)

(方法和原因同Train資料集中寬為640 pixels圖檔)

1. 寬為1080 pixels圖檔 (將兩種大小各切成3份)

(原因同Train資料集中640\*640 pixels圖檔)

在訓練後資料整合的部分，本團隊則是利用了在創新性的部分便有提到 過的圖像內縮、訓練結果的合併以及Non- Maximum Suppression (NMS)演 算 法的處理。

本團隊先將，分別對兩種不同方式裁切的Train資料集進行訓練，產生兩種模型並將Public 和Private圖檔根據裁切方式使用模型進行 detect(640\*640 pixels圖檔使用640\*640 pixels圖檔產生的模型，寬為1080 pixels圖檔使用1088\*1088 pixels產生的模型)，在detect後，為了防止其因裁圖而導致的預測錯誤(例如:僅有框到車頭或車尾之狀況)，因此我們進行圖像內縮(方法如創新性的說明)，並利用程式將源於同一圖片的裁切圖進行合併。合併結束後我們會利用Non- Maximum Suppression (NMS)演算法，以IOU閥值大於0.5條件，來進行預測框的刪減，以消除因為切圖而導致同物件被多個預測框預測而導致Precision分數下降的風險。

1. 訓練方式

本團隊採用的訓練方法為遷移學習，並在訓練中使用凍結訓練法。使用遷移學習的原因在於其精確性高，及耗時短的特性；凍結訓練的使用則是為了加快訓練效率及防止權值被破壞。而訓練的過程採用的是YOLOv7-X(以下簡稱為v7-X)幫助我們訓練模型，訓練的過程和方法如下 :

1. : 將資料導入v7-X的模型中訓練

* 在此訓練的資料為經過處理的資料，其中分別將640\*640 pixels的裁切資料和縮放成1088\*1088 pixels的資料放入v7-X中訓練。

1. : 選擇權重

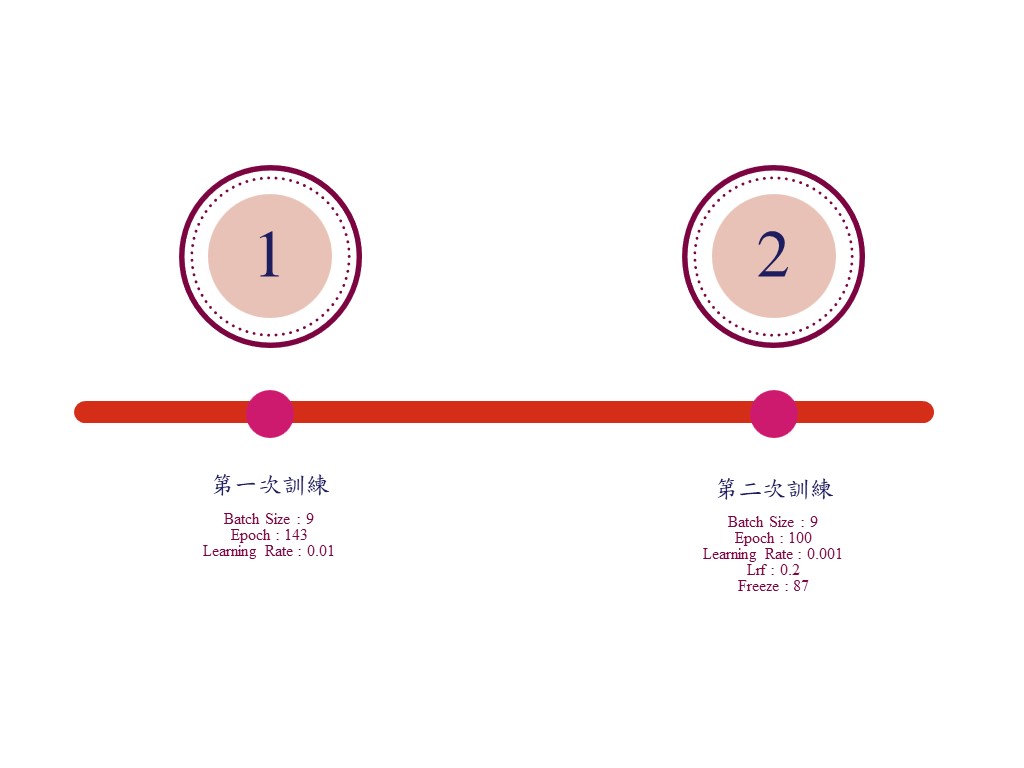
* 在該步驟中，本團隊在最開始的那一次所選擇v7-X中所提供的預訓練權重作為初始的模型參數，而往後則是以訓練後所產生的模型作為訓練權重。

1. : 調整參數(對兩種資料分別進行訓練)

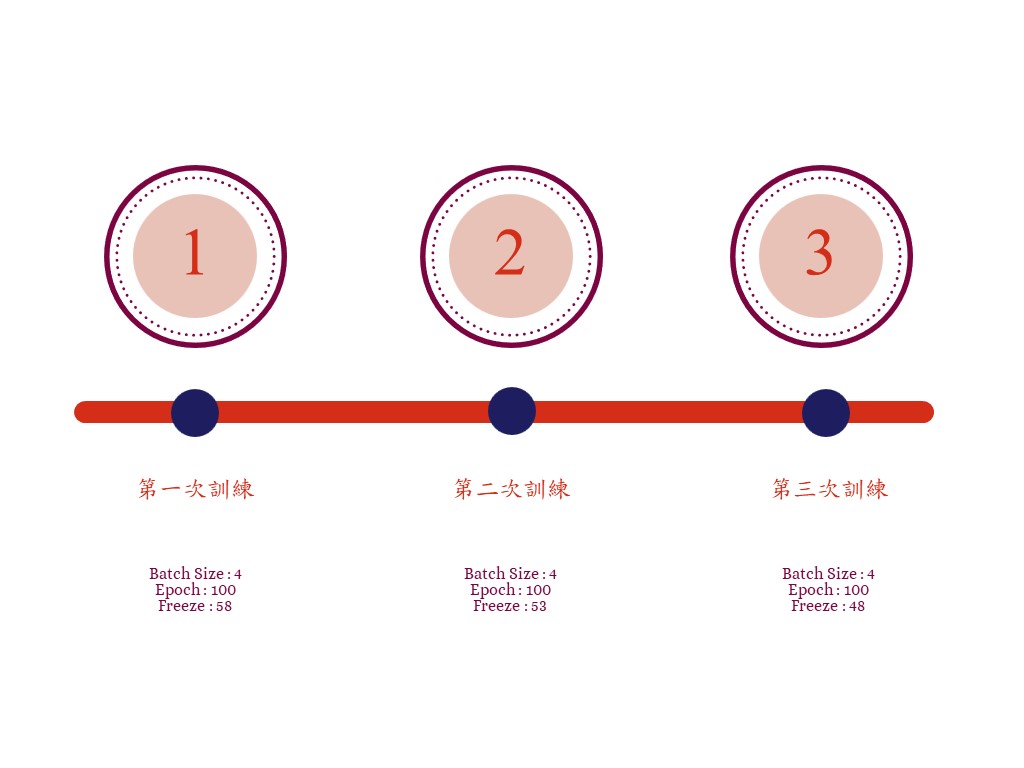
* 對於640\*640 pixels的資料，我們以Batch Size為9與Epoch為143的參數調整下 Learning Rate同步調整至0.001作為第一次訓練。
* 對於寬為1088 pixels的資料，我們以Batch為4與Epoch為100，在不調整Learning Rate的參數下，以凍結58層做為第一次訓練。

1. : 凍結訓練

* 在此步驟中，本團隊對凍結的層數進行調整並對切割後的資料進行再訓練，其中權重的部分則是以每次訓練後產生的模型作為下一次訓練的訓練權重。
* 對於640\*640 pixels的資料，我們在進行過第一次訓練後，以第一次訓練的結果為權重，凍結87層、lrf : 0.2 、Epoch 為100的參數其Learning Rate設置0.001進行一次模型在訓練，在此調整Learning Rate和lrf 是為了收斂到新的最小值。
* 對於寬為1088 pixels的資料，我們在第一次訓練中便已凍結58層，在不改變其他參數的條件下，調整凍結層數來進行模型的再訓練，其中再訓練的權重如上所說皆是上一次訓練後所產生的模型。



**Figure 3** : 寬為640 pixels圖檔的訓練流程



**Figure 4** : 寬為1088 pixels圖檔的訓練流程

1. 分析與結論

Chart, waterfall chart

Description automatically generated **Confusion Matrix 分析:**

**Figure 5** ：以640\*640為裁切大小訓練模型之confusion matrix，

Chart, waterfall chart

Description automatically generated 紅框的部分得知模型把背景預測為人的信心指數達0.66

**Figure 6**：以寬度為1088裁切訓練模型之confusion matrix，紅框標誌 為比較640訓練模型信心指數高的部分

**Figure 5**為原本的Train圖以640\*640pixel裁切後再以原圖訓練的模型(簡稱640模型)進行預測後，將新標記出來的label作為新的模型訓練label所訓練出來的模型confusion matrix。在640模型的confusion matrix中我們得知模型對於validation的預測信心指數都高達0.85以上。此模型的預測準確率算是優秀的，但從紅框的部分得知此模型在預測時有0.66的信心指數將背景預測為人。雖然本團隊將訓練用的圖做裁切以便凸顯人的特徵，但人的label還是太小，在卷積過程中會被處理得更小導致資訊不足，從而導致這種情況發生。用**Figure 6**比較640模型，可見以1088為寬度裁切訓練的模型（簡稱1088模型）的大型車及機車的信心指數高於640模型，將背景判斷為人的confidence也下降至0.57。

* Chart

  Description automatically generated**F1-score 分析:**

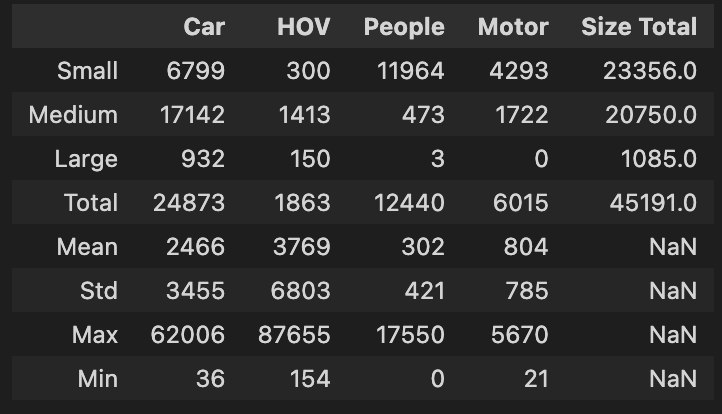
Chart

Description automatically generated **Figure 7**：640模型的F1 curve，大部分class可以在信心指數大於0.5 的時候，F1-score還能保持高於0.8

**Figure 8：**1088模型的F1 curve，車及大型車在的F1-score在信心指數 0.1至0.7左右都非常穩定且高

從640模型及1088模型的F1 curve分析，由於1088模型框的數量比640模型少，所以大部分的F1-score 的曲線比起640模型都相對穩定，代表準確度也會相對正確，比較少框錯物品的情況。兩個表格的中，人的F1-score比起其他三個物件都相對低，兩者都沒有達到0.8 的F1-score。這種情況可能在於人物的特徵原本就很小，在訓練時卷積的過程又將圖像卷得更小，導致特徵變得更加不明顯，模型沒辦法很好的訓練。

* **標籤統計分析：**

在得到官方最新的Train資料集時，將Label標記到圖片中後，透過觀察照片圖檔發現官方的Label 標籤不盡完善，因此本團隊以官方Label

**Figure 9**：Train Label統計數據

為標準對640模型及1088模型預測出的Train Label做統計分析。

在官方的數據中，人物Label的最小面積出現為0，這是資料集中的錯誤資料，再加上上述所提到的問題，本團隊決定用自己模型預測出的Label作為新的訓練資料集。

|  |  |
| --- | --- |
| **Figure 10：**640模型預測A picture containing text, black, scoreboard  Description automatically generated | **Figure 11：**1088模型預測A picture containing text, scoreboard, black  Description automatically generated |

本團隊用640模型及1088模型去預測Train圖所得出的Label做分析。640模型的人物預測比1088模型多了1倍左右，再將Label標記到Public的圖片做比較後發現640模型的小型物件的準確率比1088模型高。

|  |  |
| --- | --- |
| **Map  Description automatically generated Figure 12**：1088模型預測 | **Map  Description automatically generatedFigure 13：**640模型預測 |
| (如圖所示640模型預測的人比1088模型準確率高) | |

綜上所述，最後本團隊決定在模型預測的Public資料及Private資料中將1088模型的車輛和大型車與640模型的人及機車Label做合併，並用NMS處理合併文件，作為最後的Result檔案。

1. 程式碼(如附件)
2. 使用的外部資源與參考文獻

* Kin-Yiu, Wong(2022,December 2) yolov7 README.md. GitHub

<https://github.com/WongKinYiu/yolov7/blob/main/README.md>

1. 作者聯絡資料表

* **隊伍**

|  |  |  |
| --- | --- | --- |
| 隊伍名稱 | Private Leaderboard成績 | Private Leaderboard 名次 |
| TEAM\_2086 | 0.730121 | 26 |

* **隊員(隊長請填第一位)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 姓名 | 學校名稱 | 系所 | 電話 | E-mail |
| 卓子揚  (Tzu-Yang Cho) | 輔仁大學 | 數學系 | 0909-308-660 | henry2.900109@gamil.com |
| 詹茹萍  (Ju-Ping Chan) | 輔仁大學 | 數學系 | 0970-174-105 | feliciachan572@gmail.con |
| 紀昌賢  (Chiong-Hean Kee) | 輔仁大學 | 數學系 | 0937-788-128 | keechionghean@gmail.com |
| 林峻宇  (Chun-Yu Lin) | 輔仁大學 | 數學系 | 0958-387-818 | f130803417@gmail.com |
| 楊鎧丞  (Yang-Kai Cheng) | 輔仁大學 | 數學系 | 0920-596-085 | pa901020@gmail.com |

* **指導教授/指導業師**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 教授/業師姓名 | 課程名稱 | 課程代碼 | 學校名稱 | 系所 | 電話 | E-mail |
| 陳泓勳 |  |  | 輔仁大學 | 數學系 |  | 152228@mail.fju.edu.tw |