
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

無人機飛行載具之智慧計數-報告說明文件

目錄

壹、	環境.....	2
貳、	演算方法與模型架構.....	3
參、	創新性.....	4
肆、	資料處理.....	6
伍、	訓練方式.....	7
陸、	分析與結論.....	9
柒、	程式碼.....	13
捌、	使用的外部資源與參考文獻.....	13
玖、	作者聯絡資料表.....	13

無人機飛行載具之智慧計數-報告說明文件

壹、 環境

- (一) 作業系統：Ubuntu 20.04.3 LTS、Windows11
- (二) 顯示卡：RTX2080 Ti、Tesla T4、Nvidia A100
- (三) VRAM：10 GB
- (四) 語言：Python3.8
- (五) 套件：
 - 1. Glob==3.8
 - 2. matplotlib==3.5.0
 - 3. numpy==1.19.4
 - 4. opencv-python==4.1.1
 - 5. os==3.8
 - 6. Pandas==1.5.1
 - 7. Pillow==8.4.0
 - 8. protobuf==3.17.3
 - 9. PyYAML==5.4.1
 - 10. requests==2.26.0
 - 11. scipy==1.4.1
 - 12. shutil==3.8
 - 13. time==3.8
 - 14. torch==1.10.0+cu113
 - 15. torchvision==0.11.1+cu113
 - 16. tqdm==4.62.3
- (六) 預訓練模型
 - 1. YOLOv7-X

無人機飛行載具之智慧計數-報告說明文件

貳、 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

Model	#Param.	FLOPs	Size	AP ^{val}	AP ^{val} ₅₀	AP ^{val} ₇₅	AP ^{val} _S	AP ^{val} _M	AP ^{val} _L
YOLOv4 [3]	64.4M	142.8G	640	49.7%	68.2%	54.3%	32.9%	54.8%	63.7%
YOLOv4-u5 (r6.1) [81]	46.5M	109.1G	640	50.2%	68.7%	54.6%	33.2%	55.5%	63.7%
YOLOv4-CSP [79]	52.9M	120.4G	640	50.3%	68.6%	54.9%	34.2%	55.6%	65.1%
YOLOv4-CSP [81]	52.9M	120.4G	640	50.8%	69.5%	55.3%	33.7%	56.0%	65.4%
YOLOv7	36.9M	104.7G	640	51.2%	69.7%	55.5%	35.2%	56.0%	66.7%
improvement	-43%	-15%	-	+0.4	+0.2	+0.2	+1.5	=	+1.3
YOLOv7-X [81]	96.9M	226.8G	640	52.7%	71.3%	57.4%	36.3%	57.5%	68.3%
YOLOv7-X	71.3M	189.9G	640	52.9%	71.1%	57.5%	36.9%	57.7%	68.6%
improvement	-36%	-19%	-	+0.2	-0.2	+0.1	+0.6	+0.2	+0.3

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。

無人機飛行載具之智慧計數-報告說明文件

參、 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 閾值大於 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。

無人機飛行載具之智慧計數-報告說明文件

未經資料後處理前成績

Advanced_Public_Recall_TI oU	Advanced_Public_Precision_TI oU	Advanced_Public_Score_ dis
0.658215	0.669438	0.799332

經過內縮處理的成績

Advanced_Public_Recall_TI oU	Advanced_Public_Precision_TI oU	Advanced_Public_Score_ dis
0.652213	0.681768	0.793506

經完整資料後處理後的成績

Advanced_Public_Recall_TI oU	Advanced_Public_Precision_TI oU	Advanced_Public_Score_ dis
0.647016	0.69472	0.78985

Figure2：經不同資料處理後的成績比較

無人機飛行載具之智慧計數-報告說明文件

肆、 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：

第一步：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。

第二步：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：

1. 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。

2. 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)

2. 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 閾值大於 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。

無人機飛行載具之智慧計數-報告說明文件

伍、 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：

第一步：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。

第二步：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。

第三步：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。

第四步：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。

無人機飛行載具之智慧計數-報告說明文件



Figure 3：寬為 640 pixels 圖檔的訓練流程

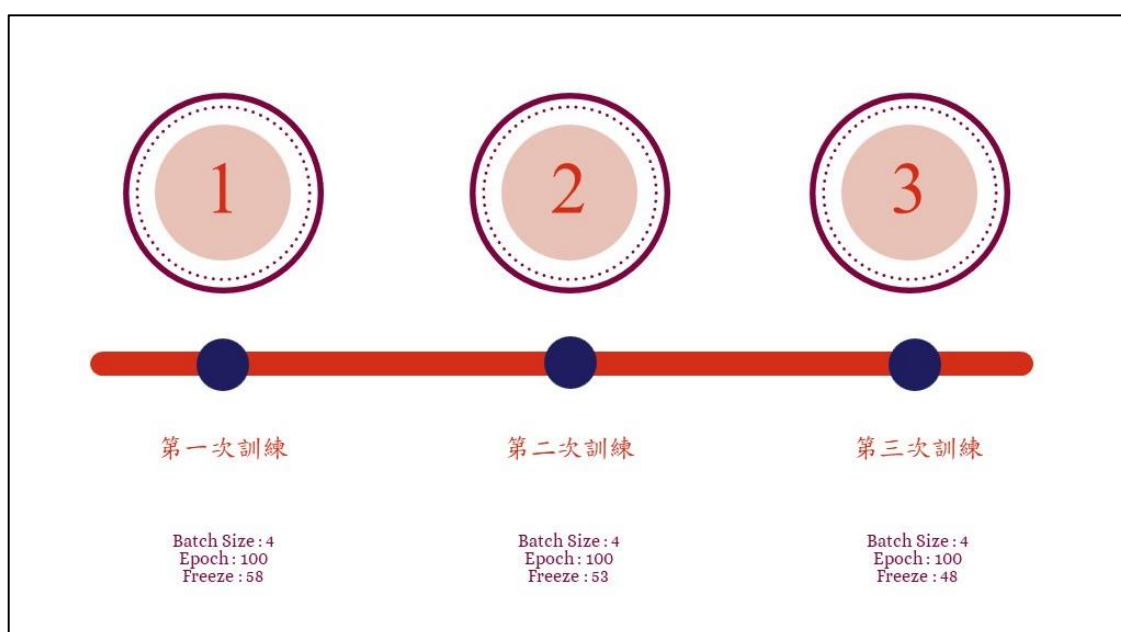


Figure 4：寬為 1088 pixels 圖檔的訓練流程

無人機飛行載具之智慧計數-報告說明文件

陸、 分析與結論

Confusion Matrix 分析:

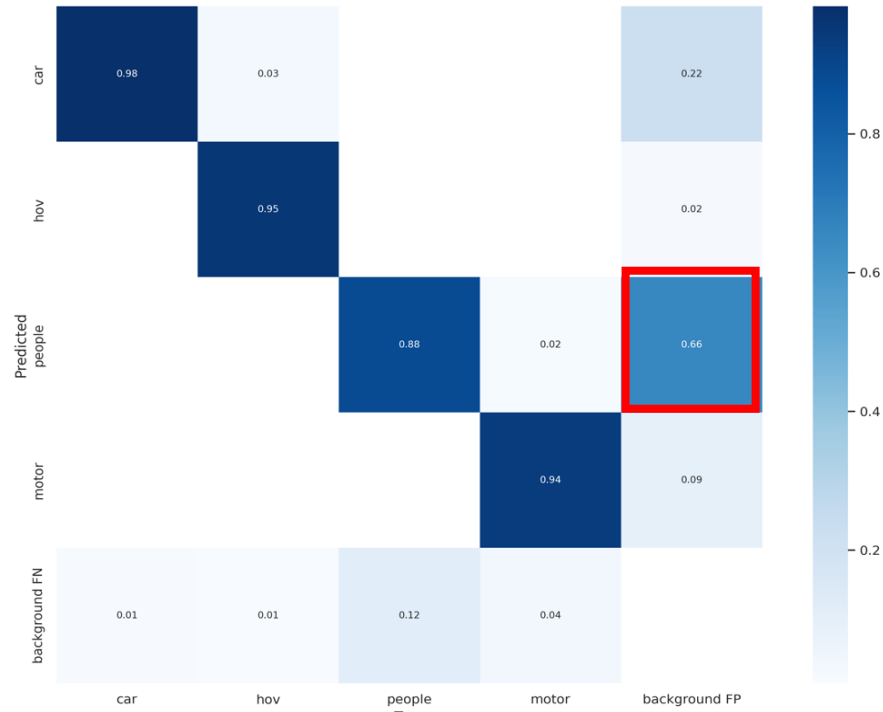


Figure 5 : 以 640*640 為裁切大小訓練模型之 confusion matrix, 紅框的部分得知模型把背景預測為人的信心指數達 0.66

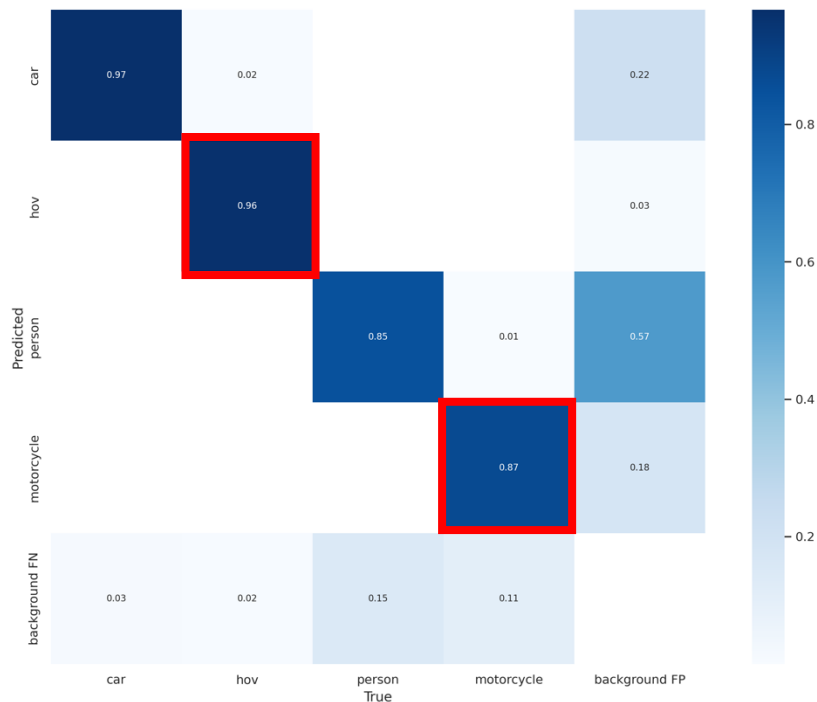


Figure 6: 以寬度為 1088 裁切訓練模型之 confusion matrix, 紅框標誌為比較 640 訓練模型信心指數高的部分

無人機飛行載具之智慧計數-報告說明文件

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。

F1-score 分析:

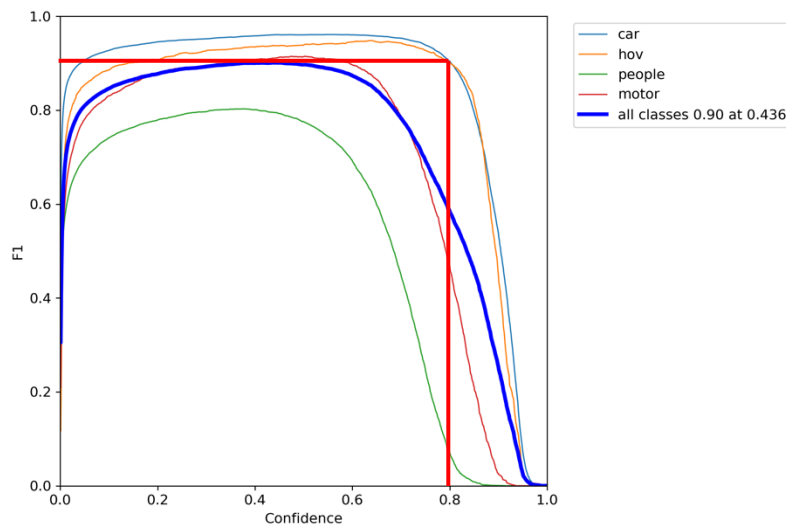


Figure 7: 640 模型的 F1 curve, 大部分 class 可以在信心指數大於 0.5 的時候, F1-score 還能保持高於 0.8

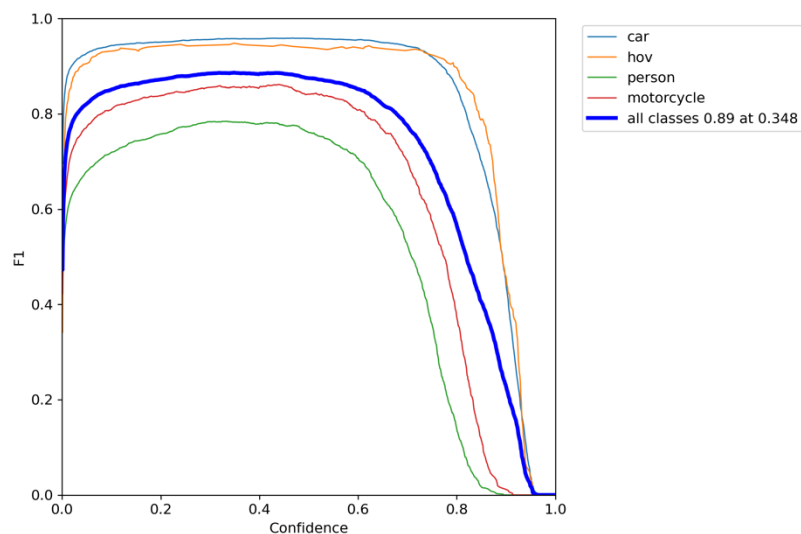


Figure 8: 1088 模型的 F1 curve, 車及大型車在的 F1-score 在信心指數 0.1 至 0.7 左右都非常穩定且高

無人機飛行載具之智慧計數-報告說明文件

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

	Car	HOV	People	Motor	Size Total
Small	6799	300	11964	4293	23356.0
Medium	17142	1413	473	1722	20750.0
Large	932	150	3	0	1085.0
Total	24873	1863	12440	6015	45191.0
Mean	2466	3769	302	804	NaN
Std	3455	6803	421	785	NaN
Max	62006	87655	17550	5670	NaN
Min	36	154	0	21	NaN

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。

	Car	HOV	People	Motor	Size Total
Small	9322	407	27371	5733	42833.0
Medium	21715	1912	608	2124	26359.0
Large	1220	187	4	0	1411.0
Total	32257	2506	27983	7857	70603.0
Mean	2427	3553	246	773	NaN
Std	3383	6230	305	752	NaN
Max	64368	98088	16168	6256	NaN
Min	54	143	32	54	NaN

Figure 10: 640 模型預測

	Car	HOV	People	Motor	Size Total
Small	7700	363	12504	4872	25439.0
Medium	16732	1420	396	1474	20022.0
Large	832	128	5	0	965.0
Total	25264	1911	12905	6346	46426.0
Mean	2286	3050	259	699	NaN
Std	3146	4046	381	714	NaN
Max	52250	42864	15960	5934	NaN
Min	48	96	18	24	NaN

Figure 11: 1088 模型預測

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。

無人機飛行載具之智慧計數-報告說明文件

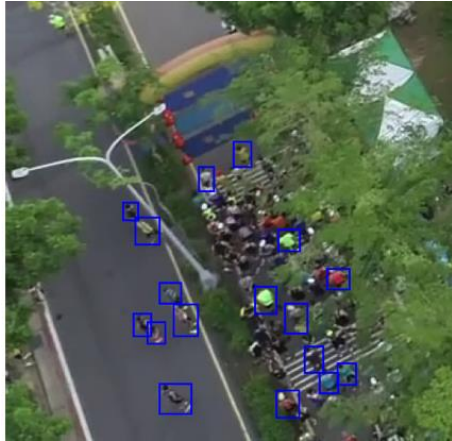


Figure 12: 1088 模型預測

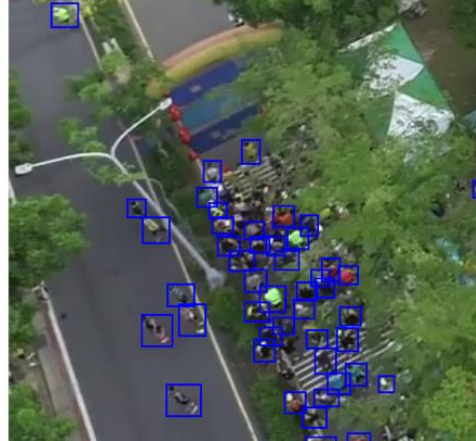


Figure 13: 640 模型預測

(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。

無人機飛行載具之智慧計數-報告說明文件

柒、 程式碼

<https://drive.google.com/drive/folders/1IAkyG32tMoVhVMXcXJCKrHpuuUSjK6eB?usp=sharing>

捌、 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>

玖、 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