



國立中山大學資訊管理所

Kaggle competition : TensorFlow - Help Protect the Great Barrier Reef

國立中山大學資訊管理所 - 深度學習 課程 - 第十組 Final Report

授課老師: 楊惠芳 博士
學生 : M104020053 莊明輯
M104020029 邱承漢



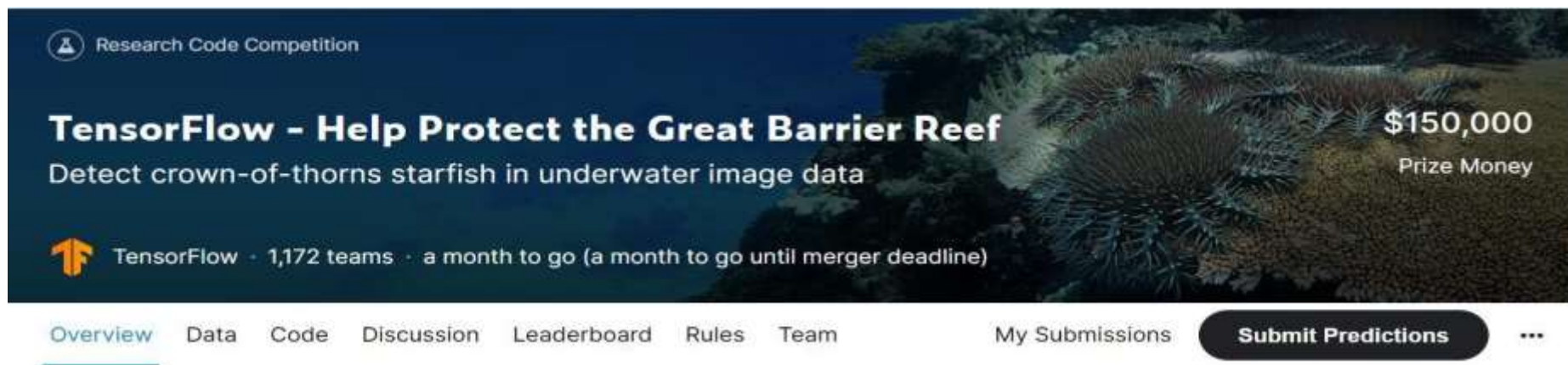
索引

▶一、任務介紹-----	3
▶二、實作-----	4
▶三、最終結果-----	22
▶四、心得-----	23
▶五、附錄-----	24





一、任務介紹



- ▶ 本次競賽的目標是通過珊瑚礁水下影片來訓練物件偵測模型，準確識別海星。我們將在大堡礁周圍不同時間和地點拍攝的水下圖像序列中預測棘冠海星(COTS)的存在和位置。預測採用 bounding box 的形式以及每個識別出的海星的置信度分數。一個圖像可能包含零個或多個海星。這次工作將幫助研究人員識別威脅澳大利亞大堡礁的物種。
- ▶ Kaggle 連結：

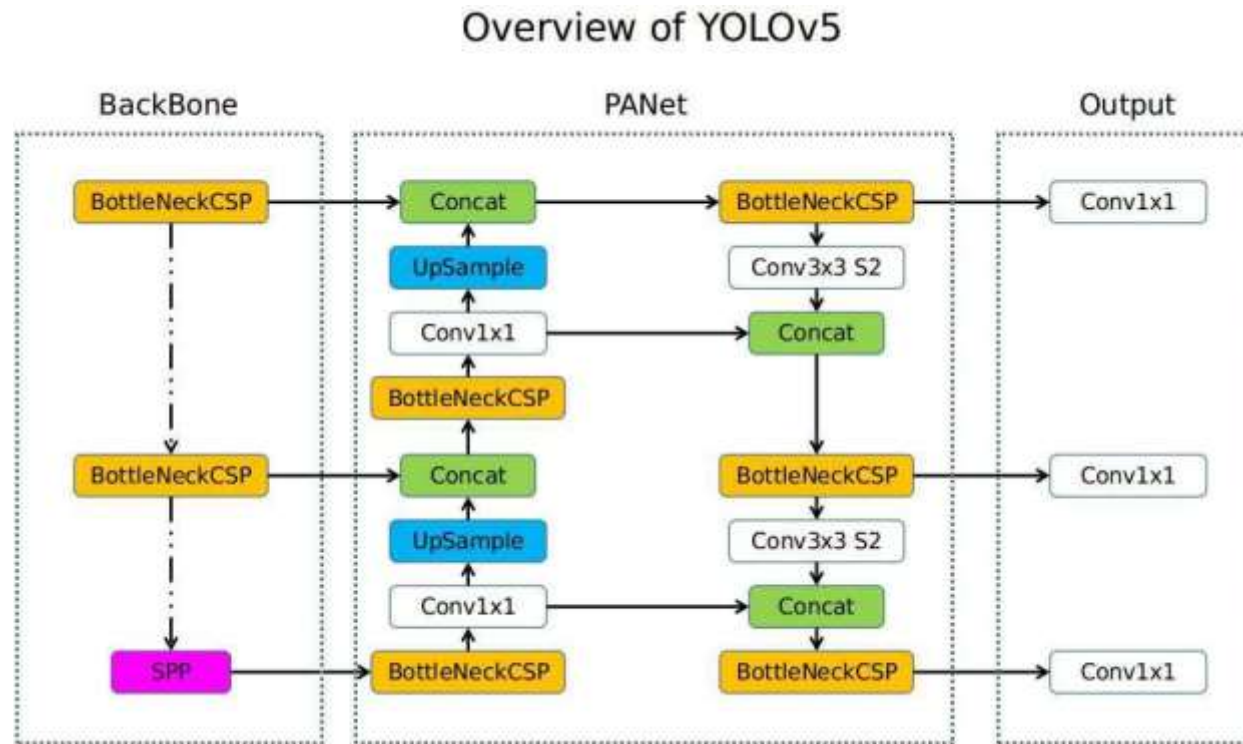
<https://www.kaggle.com/c/tensorflow-great-barrier-reef/leaderboard>He, K., Zhang, X.,





二、實作

- 不同於期中 Progress 我們使用 Two-Stage 的 Faster RCNN，這次我們嘗試使用更先進的 one-stage detector：YOLO 系列物件偵測模型來進行實作。我們選擇的是 YOLOv5 作為本次完成任務的骨幹，即使他在學界上有些許的爭議，但不影響他整體優秀的表現及評價。由於 YOLOv5 並沒有發表官方論文，以下架構圖由論壇網友繪製，供參考：





資料前處理

- ▶ 為了提升訓練效率，所以我們先把 $\text{annotation} = 0$ 的訓練資料剔除。

原始資料集有 23501 張圖片，11898 個 bounding box:

```
1 print('number of BBOX:', df['NumBBox'].sum())
2 print("number of images:", df.shape[0])
```

```
number of BBOX: 11898
number of images: 23501
```

- ▶ 剔除 $\text{annotation} = 0$ 的訓練資料:

```
1 # 去除沒有COTS的圖片
2 df_train=df[df["NumBBox"]>0]
3 df_train.sample(2)
```

	video_id	sequence	video_frame	sequence_frame	image_id	annotations	NumBBox
12642	1	60754	9077	661	1-9077	[{'x': 506, 'y': 568, 'width': 28, 'height': 3...	7
12784	1	60754	9219	803	1-9219	[{'x': 336, 'y': 582, 'width': 65, 'height': 5...	5

```
1 print('number of BBOX:', df_train['NumBBox'].sum())
2 print("number of images:", df_train.shape[0])
```

```
number of BBOX: 11898
number of images: 4919
```

- ▶ 可以發現，竟然只有兩成的資料是有包括 COTS 的標註，我們剔除了約八成的資料。





- ▶ 最後我們將訓練資料分成 train set(0.8)跟 validation set(0.2)

```
1 list1 = os.listdir('COTS/images/train') |  
2 number_files1 = len(list1)  
3 print("Number of images in ./COTS/images/train folder",number_files1)  
4 list2 = os.listdir('COTS/images/valid')  
5 number_files2 = len(list2)  
6 print("Number of images in ./COTS/images/valid folder",number_files2)
```

```
Number of images in ./COTS/images/train folder 3974  
Number of images in ./COTS/images/valid folder 945
```



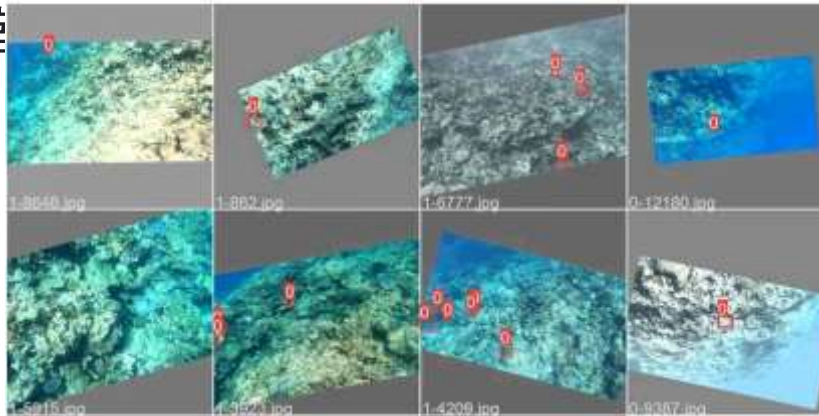


資料增強

- **Data Augument:**由於訓練資料並不多，我們希望能透過資料增強讓訓練後的模型能夠更加泛化。

```
hsv_h: 0.015 # image HSV-Hue augmentation (fraction)
hsv_s: 0.7 # image HSV-Saturation augmentation (fraction)
hsv_v: 0.4 # image HSV-Value augmentation (fraction)
degrees: 30.0 # image rotation (1/- deg)
translate: 0.1 # image translation (+/- fraction)
scale: 0.3 # image scale (1/- gain)
shear: 0.0 # image shear (+/- deg)
perspective: 0.0 # image perspective (+/- fraction), range 0-0.001
flipud: 0.5 # image flip up-down (probability)
fliplr: 0.5 # image flip left-right (probability)
mosaic: 0.0 # image mosaic (probability)
mixup: 0.0 # image mixup (probability)
copy_paste: 0.0 # segment copy-paste (probability)
```

我們讓圖片翻轉正負 30 度且縮小的訓練資料上下或水平翻轉。





► **Run1: IMAGE_SIZE =1280, BATCH_SIZE=14, EPOCHS = 30**

```
!python train.py --img {IMG_SIZE} \  
--batch {BATCH_SIZE} \  
--epochs {EPOCHS} \  
--data data.yaml \  
--weights yolov5s.pt \  
--project kaggle-Reef
```

分數：[Great-Barrier-Reef: YOLOv5 \[infer\]](#)

epoch30 (version 1/1)

21 minutes ago by M104020053

Notebook [Great-Barrier-Reef: YOLOv5 \[infer\]](#) | Version 1

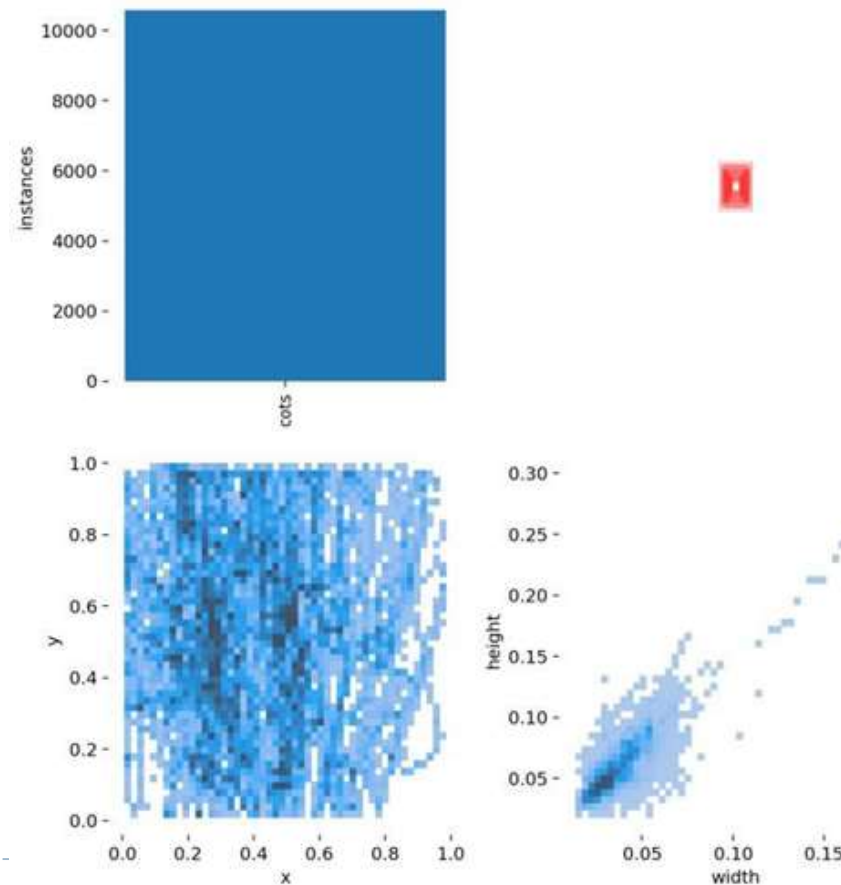
Succeeded

0.291



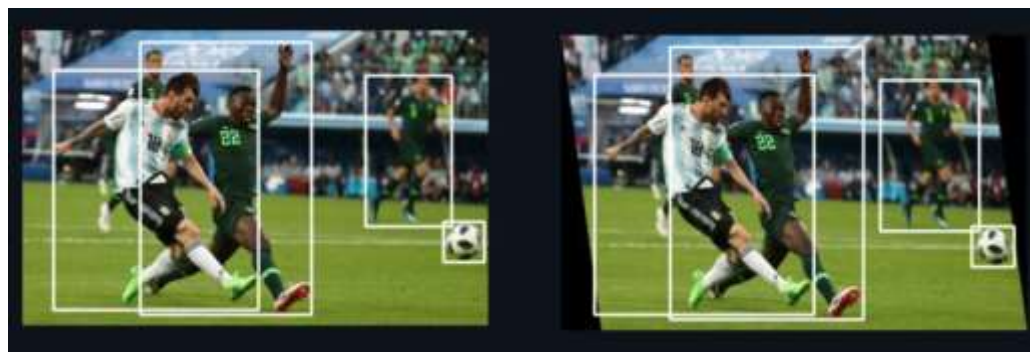


- ▶ 老實說，這個情況完全超乎我們的預期.....比我們想像的表現差很多。我們認為主要原因還是這次任務是屬於小物件偵測，這也是物件偵測一直以來相對弱勢的部份。從下面的圖可以很明顯的瞭解到，訓練資料的 Bounding Box 大小分布大多都是小物件。

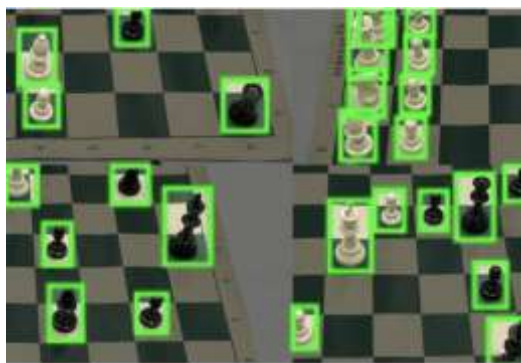




- ▶ 接下來我們將嘗試更多的 Data Augment 方法，在查詢相關資料後，我們發現 YOLOv5 有支援這兩種蠻有趣的資料增強方式
- ▶ **shear**：這種方法會以斜拉的方式變形我們的圖片。

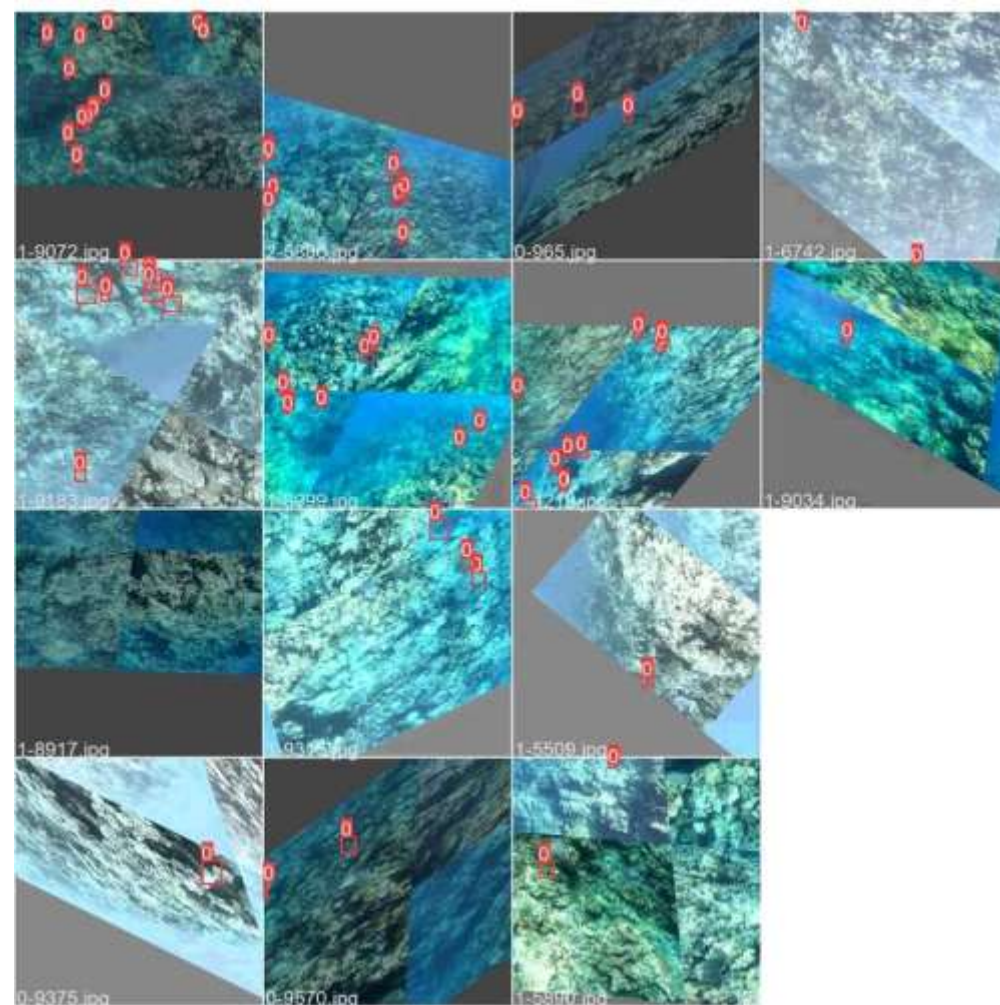


- ▶ **mosaic**：會以拼接的方式將 4 張圖片合在一起，這種方法將有助於模型檢測小物件的能力，且提升模型的泛化能力。





```
hsv_h: 0.015 # image HSV-Hue augmentation (fraction)
hsv_s: 0.7 # image HSV-Saturation augmentation (fraction)
hsv_v: 0.4 # image HSV-Value augmentation (fraction)
degrees: 30.0 # image rotation (+/- deg)
translate: 0.1 # image translation (+/- fraction)
scale: 0.3 # image scale (+/- gain)
shear: 30.0 # image shear (+/- deg)
perspective: 0.0 # image perspective (+/- fraction), range 0-0.001
flipud: 0.5 # image flip up-down (probability)
fliplr: 0.5 # image flip left-right (probability)
mosaic: 1.0 # image mosaic (probability)
mixup: 0.0 # image mixup (probability)
copy_paste: 0.0 # segment copy-paste (probability)
```





訓練方法

- ▶ **增加 Epoch:**在參考網路上許多 YOLO 的訓練案例後，我們嘗試用超大的 EPOCH 數訓練模型我們嘗試訓練 3000 次，用 RTX 2080Ti 訓練了整整一天後，選擇 best_model(YOLOv5 根據內建的評估方式保存的表現最佳的模型參數)進行推論，但卻沒有得到顯著的效果提升。


Great-Barrier-Reef: YOLOv5 [infer] 

epoch3000 (version 8/24)

3 days ago by Chiu Cheng Han

Succeeded

0.370

Notebook Great-Barrier-Reef: YOLOv5 [infer]  | Version 8

- ▶ 而選擇last_model(真的訓練 3000 次後的最後模型參數)表現更慘。


Great-Barrier-Reef: YOLOv5 [infer] 

epoch3000_conf0.15_last_model (version 11/24)

3 days ago by Chiu Cheng Han

Succeeded

0.273

Notebook Great-Barrier-Reef: YOLOv5 [infer]  | Version 11





- ▶ 我們也嘗試從頭開始訓練模型(不使用 YOLOv5 的 pretrained weight), 效果並未有提升,

```
!python train.py --img {IMG_SIZE} \  
                  --batch {BATCH_SIZE} \  
                  --epochs {EPOCHS} \  
                  --data data.yaml \  
                  --weights " " \  
                  --project kaggle-Reef
```

Great-Barrier-Reef: YOLOv5 [infer]

non_pretrained_test (version 17/29)

2 days ago by Chiu Cheng Han

Succeeded

0.228

Notebook Great-Barrier-Reef: YOLOv5 [infer] | Version 17





- **Run2:**在我們失望之時，我們發現在官方的文檔中有說明到，YOLOv5 有分成 4 種版本，複雜度從小排到大分別是 YOLOv5s、YOLOv5m、YOLOv5l、YOLOv5x，在 Run1 中我們使用的是最小、最快的版本 YOLOv5s，而其他版本都是在其網路架構基礎下不斷加深、加寬的模型。



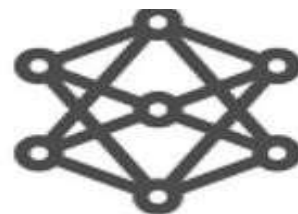
Small
YOLOv5s

14 MB_{FP16}
2.0 ms_{V100}
37.2 mAP_{COCO}



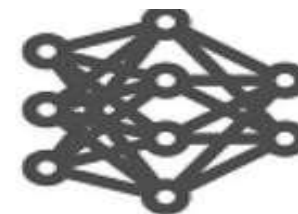
Medium
YOLOv5m

41 MB_{FP16}
2.7 ms_{V100}
44.5 mAP_{COCO}



Large
YOLOv5l

90 MB_{FP16}
3.8 ms_{V100}
48.2 mAP_{COCO}



XLarge
YOLOv5x

168 MB_{FP16}
6.1 ms_{V100}
50.4 mAP_{COCO}





- ▶ 雖然就這次任務的實際面而言，考慮到水下攝影機的硬體計算資源要同時保持準確度以及即時性，使用越小的且越快的模型應該是很大的考量因素。但就目前看來，我們模型的預測精準度遠遠低於預期，故我們嘗試使用最深、最寬的 **YOLOv5x** 進行訓練。
- ▶ 由於硬體資源的關係，我們最大的 Batch 只能最多設定到 4，並訓練 4 次。選用 best_model 進行預測，在相同大小的測試資料集下，YOLOv5s 推論完所有資料只需要 12 分鐘，而這次的 YOLOv5x 則需要花上兩倍以上的時間進行預測。



TensorFlow - Help Protect the Great B...

33 minutes

Competition submission

Executing for 33 minutes: Scoring...





- 我們的任務表現出現了突破性的提升，直接從 20~30%的準確率提升到 **0.525!!!!**

Great-Barrier-Reef: YOLOv5 [infer] epoch-16-mosaic-and-shear_last (version 5/5) 25 minutes ago by M104020053 Notebook Great-Barrier-Reef: YOLOv5 [infer] Version 5	Succeeded	0.298	<input type="checkbox"/>
Great-Barrier-Reef: YOLOv5 [infer] epoch16-mosaic-and-shear (version 4/5) 28 minutes ago by M104020053 Notebook Great-Barrier-Reef: YOLOv5 [infer] Version 4	Succeeded	0.314	<input type="checkbox"/>
Great-Barrier-Reef: YOLOv5 [infer] yolov5x_test (version 3/5) 38 minutes ago by M104020053 Notebook Great-Barrier-Reef: YOLOv5 [infer] Version 3	Succeeded	0.525	<input checked="" type="checkbox"/>
Great-Barrier-Reef: YOLOv5 [infer] epoch30_conf0.4 (version 2/5) 2 hours ago by M104020053 Notebook Great-Barrier-Reef: YOLOv5 [infer] Version 2	Succeeded	0.274	<input type="checkbox"/>
Great-Barrier-Reef: YOLOv5 [infer] epoch30 (version 1/5) 2 hours ago by M104020053	Succeeded	0.291	<input type="checkbox"/>





實驗:Data Augument

- ▶ 原本為(0.506)

Great-Barrier-Reef: YOLOv5 [infer]	Succeeded	0.506
YOLOv5l_batch4_epoch3_best_conf0.25 (version 13/15)		
8 hours ago by M104020053		
Notebook Great-Barrier-Reef: YOLOv5 [infer] Version 13		

- ▶ 嘗試不同種的資料增強方法(shear 30 degree、rotation 30 degree、垂直翻轉 0.5), 皆對最後的測試結果造成負面的影響。

Great-Barrier-Reef: YOLOv5 [infer]	Succeeded	0.406
YOLOv5l_batch4_epoch3_shear30_best_conf0.25 (version 3/4)		
an hour ago by nemo0053		
Notebook Great-Barrier-Reef: YOLOv5 [infer] Version 3		

Great-Barrier-Reef: YOLOv5 [infer]	Succeeded	0.390
YOLOv5l_batch4_epoch3_degrees30_best_conf0.25 (version 2/4)		
2 hours ago by nemo0053		
Notebook Great-Barrier-Reef: YOLOv5 [infer] Version 2		

Great-Barrier-Reef: YOLOv5 [infer]	Succeeded	0.470
YOLOv5l_batch4_epoch3_flipud0.5_best_conf0.25 (version 1/4)		
3 hours ago by nemo0053		
Notebook Great-Barrier-Reef: YOLOv5 [infer] Version 1		





- 為了展示影響，我們一次加入 flipud(上下翻轉)、shear 跟 rotation 後，直接對模型的表現造成毀滅性的打擊。在其他條件不變下(epoch4)測試表現從 0.45 跌到 0.29 分。

Great-Barrier-Reef: YOLOv5 [infer]

YOLOv5l_batch4_epoch4_data_augument_conf0.25 (version 12/12)

39 minutes ago by M104020053

Notebook Great-Barrier-Reef: YOLOv5 [infer] | Version 12

Succeeded

0.290



Great-Barrier-Reef: YOLOv5 [infer]

YOLOv5l_batch4_epoch4_best_conf0.25 (version 11/12)

2 hours ago by M104020053

Notebook Great-Barrier-Reef: YOLOv5 [infer] | Version 11

Succeeded

0.450





實驗: Batch Size

- ▶ 然而，在我們的多次嘗試下，我們發現 Batch Size 對本次任務的泛化有巨大的影響，在所有其他參數設定一樣的情況下，Batch Size = 4 的效果遠優於 Batch Size = 2

- ▶ Batch Size = 2:

Great-Barrier-Reef: YOLOv5 [infer]	Succeeded	0.389
YOLOv5x_epoch _best_model_conf0.25_batch2 (version 7/7)		
33 minutes ago by M104020053		
Notebook Great-Barrier-Reef: YOLOv5 [infer] Version 7		

Great-Barrier-Reef: YOLOv5 [infer]	Succeeded	0.238
YOLOv5x_epoch5_best_model_conf0.25 (version 6/7)		
3 hours ago by M104020053		
Notebook Great-Barrier-Reef: YOLOv5 [infer] Version 6		

- ▶ Batch Size = 4

Great-Barrier-Reef: YOLOv5 [infer]	Succeeded	0.484	<input checked="" type="checkbox"/>
YOLOv5x_epoch5_best_model_conf0.25 (version 28/28)			
40 minutes ago by Chiu Cheng Han			
Notebook Great-Barrier-Reef: YOLOv5 [infer] Version 28			

Great-Barrier-Reef: YOLOv5 [infer]	Succeeded	0.523	<input checked="" type="checkbox"/>
YOLOv5x_best_model_conf15 (version 27/28)			
7 hours ago by Chiu Cheng Han			
Notebook Great-Barrier-Reef: YOLOv5 [infer] Version 27			

- ▶ 但由於硬體資源的不足，我們無法測試 YOLOv5x 在 Batch Size 大於 4 以上的表現。





實驗: Epoch

- ▶ 我們持續增加 epoch 觀察YOLOv5x 在這次任務的隱藏測試資料集中的表現。

- ▶ Epoch 8 : 0.447

Great-Barrier-Reef: YOLOv5 [infer] Succeeded 0.447 ☐
YOLOv5x_batch4_epoch8_best_conf0.25 (version 29/29)
3 hours ago by Chiu Cheng Han
Notebook Great-Barrier-Reef: YOLOv5 [infer] | Version 29

- ▶ Epoch 10 : 0.415

Great-Barrier-Reef: YOLOv5 [infer] Succeeded 0.415 ☐
yolov5x_batch4_epoch10_best_conf0.25 (version 10/10)
an hour ago by M104020053
Notebook Great-Barrier-Reef: YOLOv5 [infer] | Version 10

- ▶ Epoch 16 : 0.355

Great-Barrier-Reef: YOLOv5 [infer] Succeeded 0.355
YOLOv5x_batch4_epoch16_best_conf0.25 (version 30/30)
2 hours ago by Chiu Cheng Han
Notebook Great-Barrier-Reef: YOLOv5 [infer] | Version 30

- ▶ 可以發現表現是逐漸下滑的, 代表模型可能在逐漸的 overfitting。





► 為了更精準的找出最佳的訓練次數，我們做了一系列的測試，並列出以下表格。表現最佳參數：

► Model : YOLOv5x

► Batch Size =4, Epoch = 4

► Data Augument

```
hsv_h: 0.015  
hsv_s: 0.7  
hsv_v: 0.4  
degrees: 0.0  
translate: 0.1  
scale: 0.5  
shear: 0.0  
perspective: 0.0  
flipud: 0.0  
fliplr: 0.5  
mosaic: 1.0  
mixup: 0.0  
copy_paste: 0.0
```

epoch \ model		
	YOLOv5l	YOLOv5x
1	0.243	0.415
2	0.509	0.472
3	0.506	0.443
4	0.45	0.533
8	0.394	0.447
10	x	0.415
16	0.427	0.355





三、最終結果


- ▶ 截至目前為止(1/12), Leaderboard 上的最高分為 0.673, 我們的最高分為 0.533, 排名為 390名, 總共隊伍數為 1172 隊。

1	CALYPSO	  	0.673	231	14h
2	saivanov		0.664	153	11h
3	(-_-) (^ ^)..zzZ (-_-)		0.658	99	1d
4	Sea of Dreams	 	0.655	115	2d
5	Alex Wong		0.650	118	1h
6	YOLOにかける		0.646	59	14h

387	Team Mach	   	0.535	38	1d
388	Costas Voglis		0.535	25	14h
389	furu	  	0.534	44	11h

390

NSYSU MIS M104020029



0.533

44

38m

Your Best Entry ↑

Your submission scored 0.533, which is an improvement of your previous score of 0.523. Great job!

Tweet this

391	-.-		0.532	13	13h
392	Deep Sea Learners	   	0.532	33	2d
393	Pranshu Mittal		0.532	10	3d





四、心得

- ▶ 我們一開始以為這個任務不會造成我們太大的困難，畢竟物件偵測並不算是個太新的領域，且技術也已經相當的成熟，網路上也能輕易地找到許多參考資料，所以便興高采烈的參加了這次的競賽。當我們發現榜單上的大神們最高的準確率竟然只有 0.6%，才意識到這個任務的困難點。我們認為這次任務主要的難點在於要**偵測的目標(COTS)在畫面的比例相當的小**，屬於小物件偵測。
- ▶ 而在我們嘗試使用我們認為的主流先進模型 YOLOv5 後，也是在踩了大大小小的坑，用了千方百計後才好不容易勉強得到及格邊緣的分數，我們不禁感嘆物件偵測這個領域真的是水很深。雖然最後的分數結果不是很好看，但在之前未接觸過任何物件偵測相關實作的狀況下，我們已經相當滿意這次的結果，也從中獲得了許多寶貴的經驗，為這堂充實的課程畫下完美的句點。



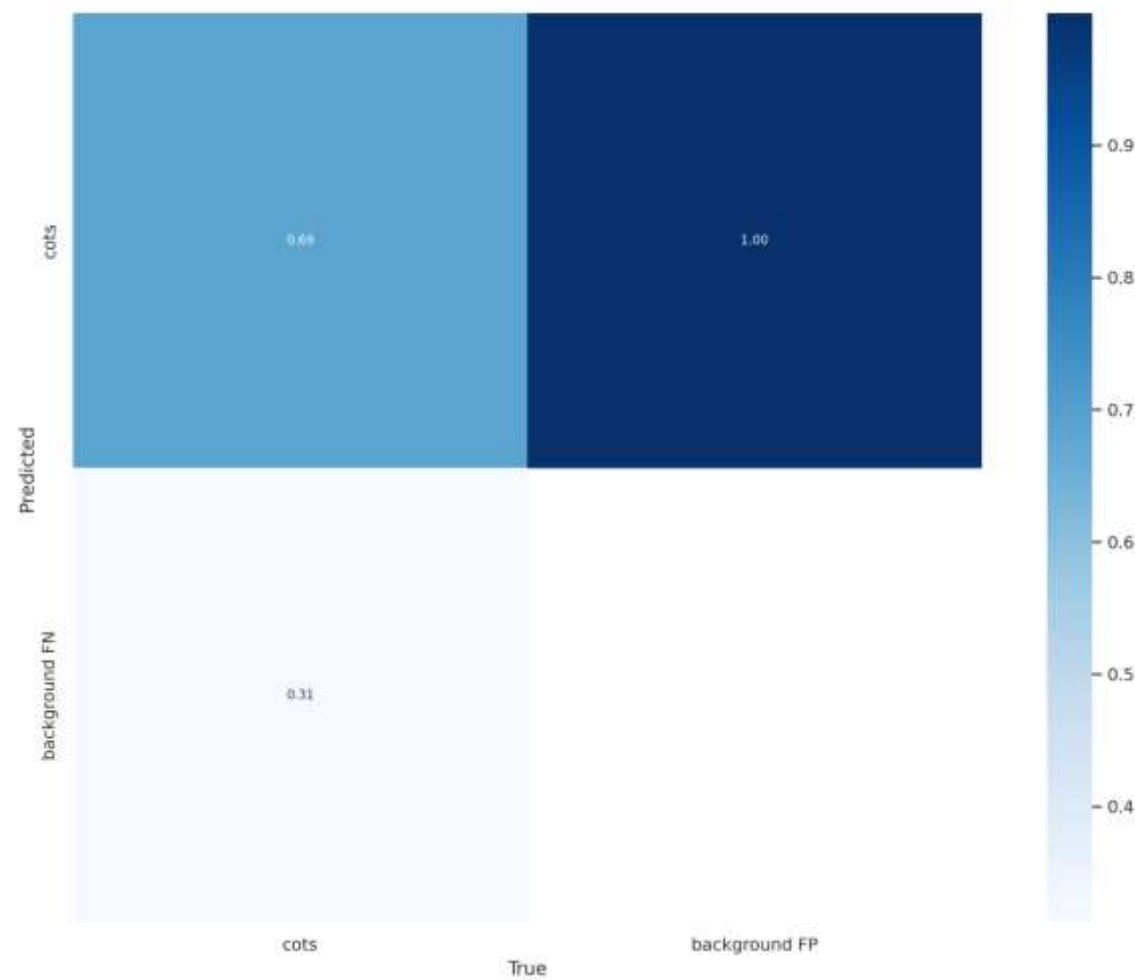


五、附錄: val batch comparison





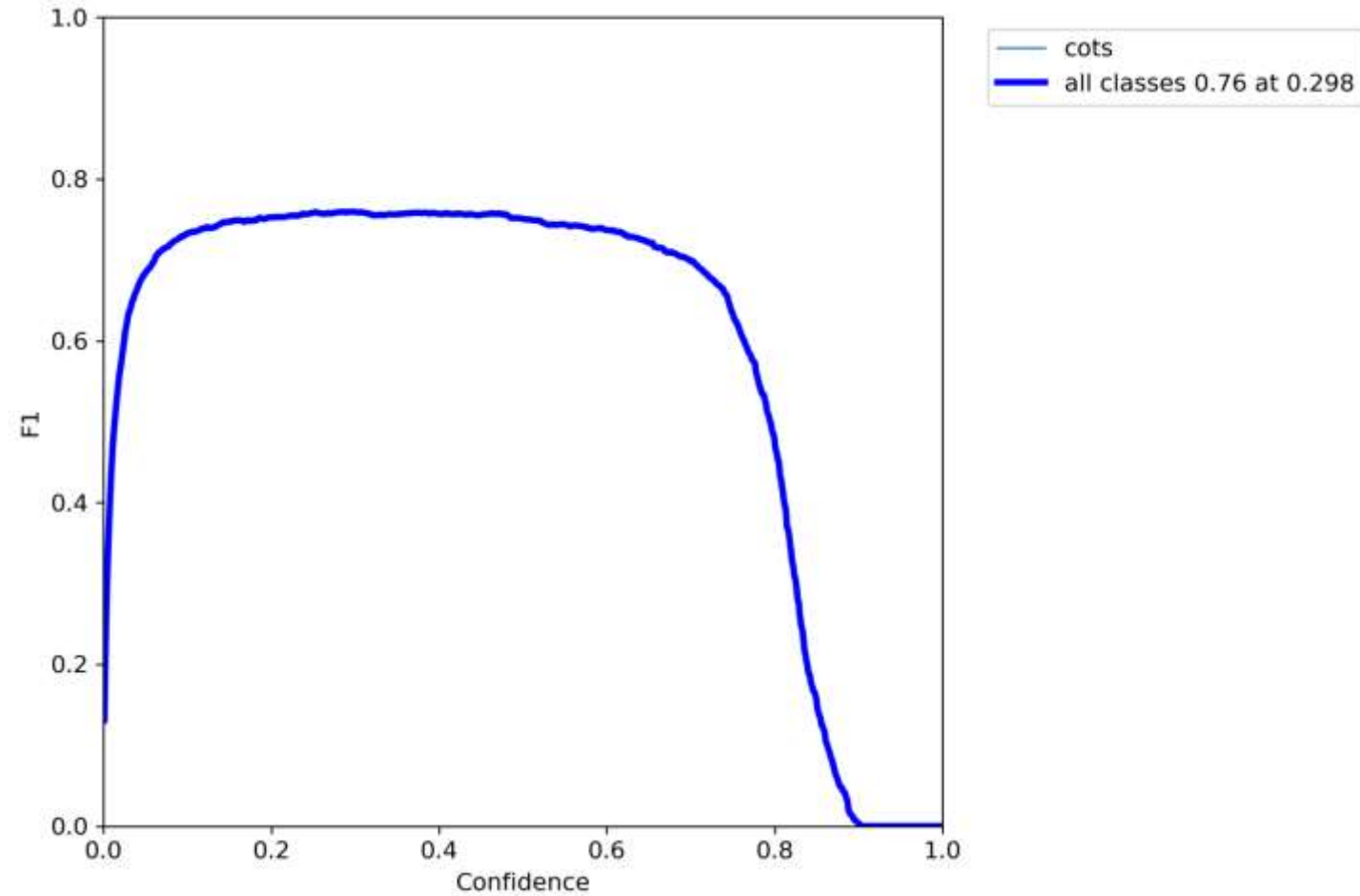
五、附錄： confusion matrix





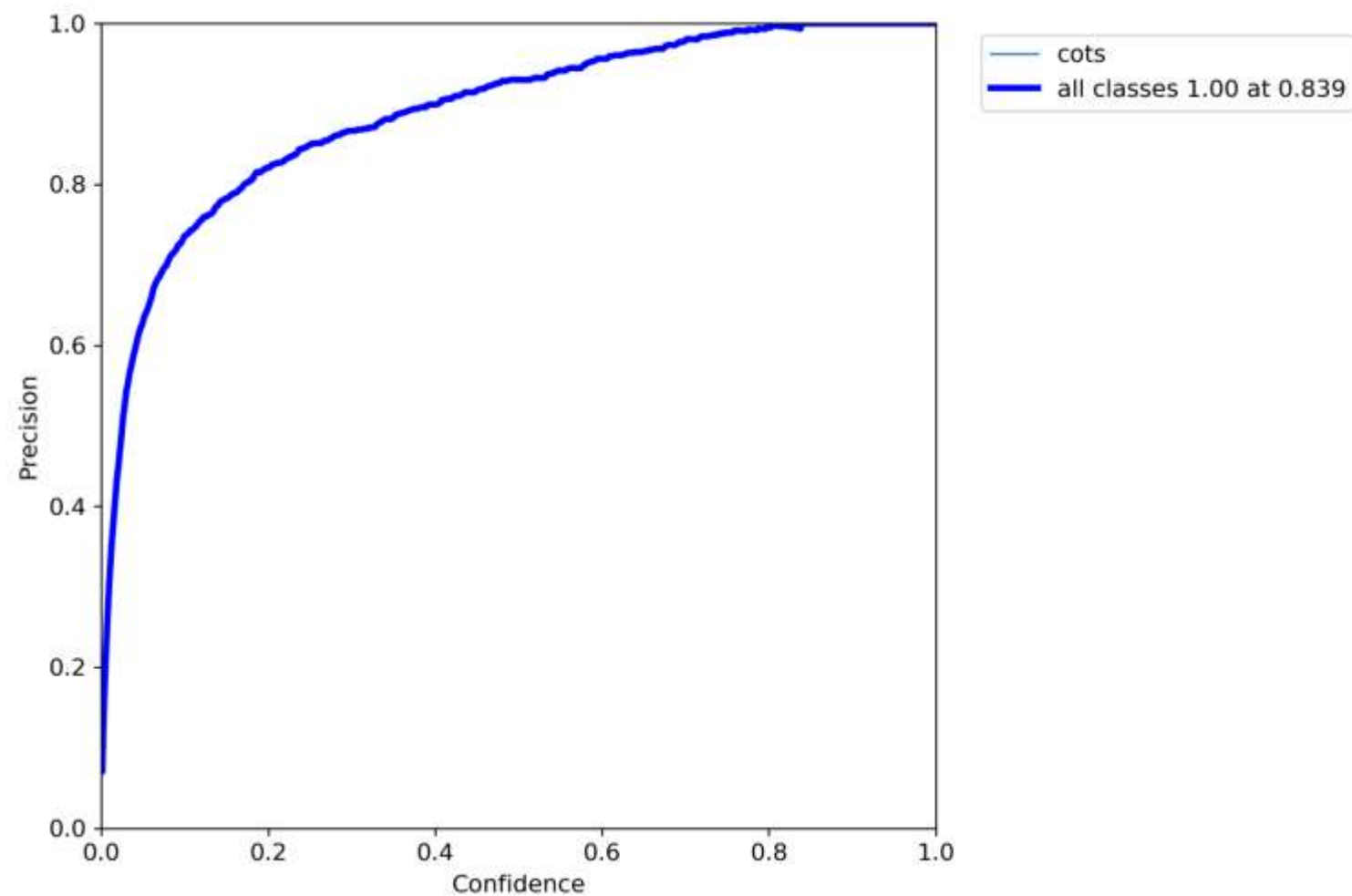
五、附錄

F1_curve



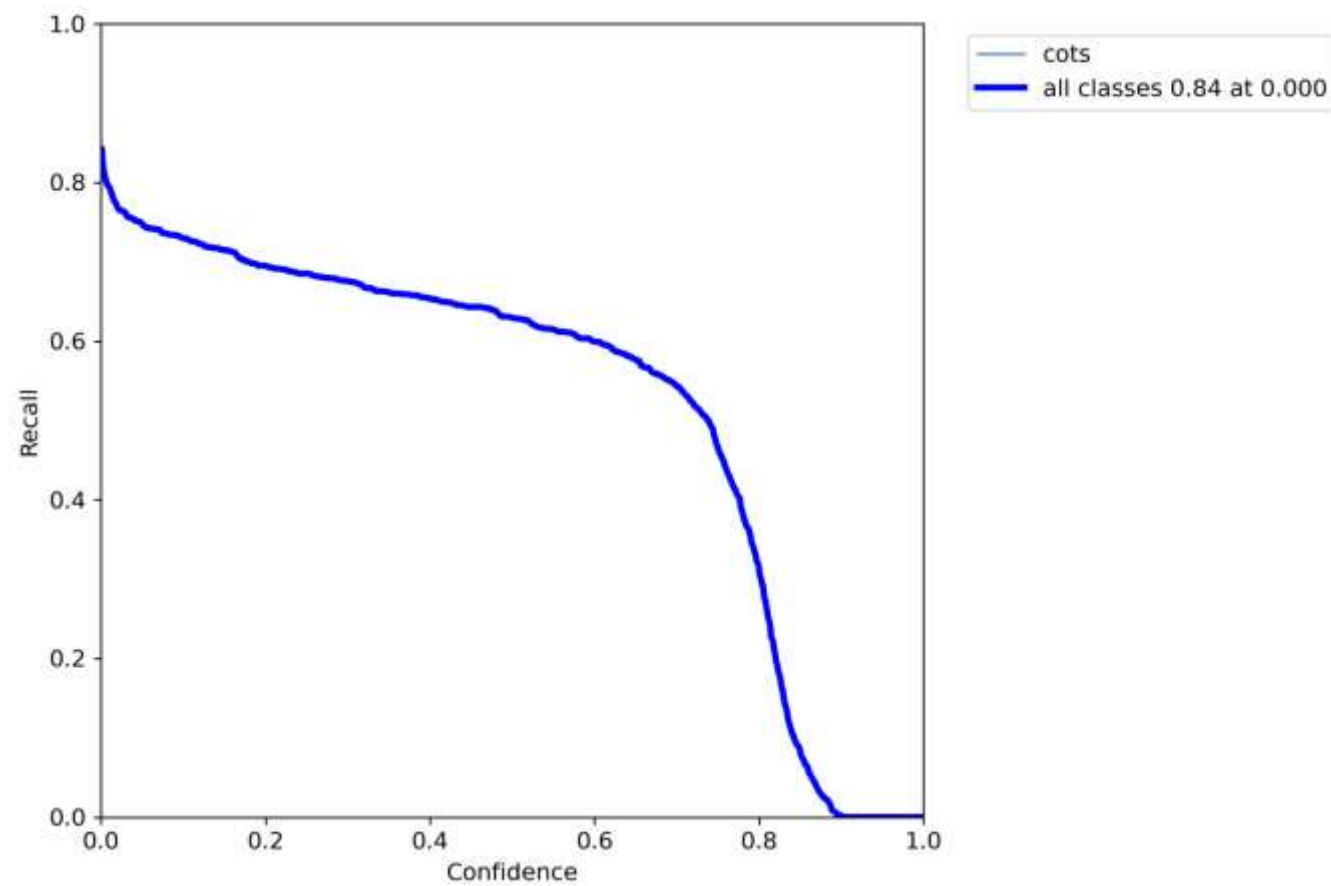


五、附錄 P_curve





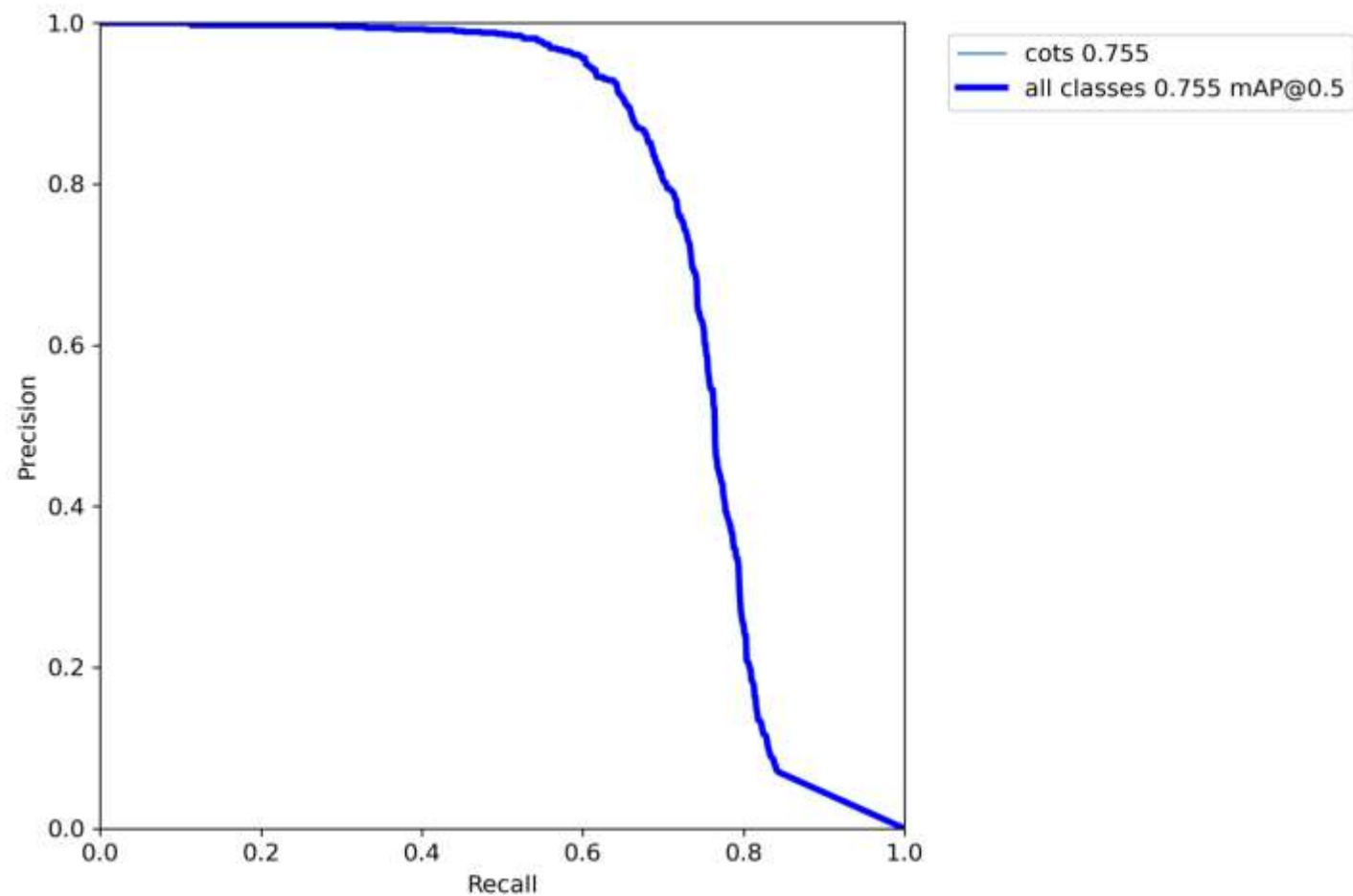
五、附錄 R_curve





五、附錄

PR_curve





► **References :**

- [1] Syed Sahil Abbas Zaidi, Mohammad Samar Ansari, Asra Aslam, Nadia Kanwal, Mamoona Asghar, and Brian Lee. 2021. A Survey of Modern Deep Learning based Object Detection Models. *ArXiv210411892 Cs Eess* (May 2021). Retrieved December 20, 2021 from <http://arxiv.org/abs/2104.11892>

