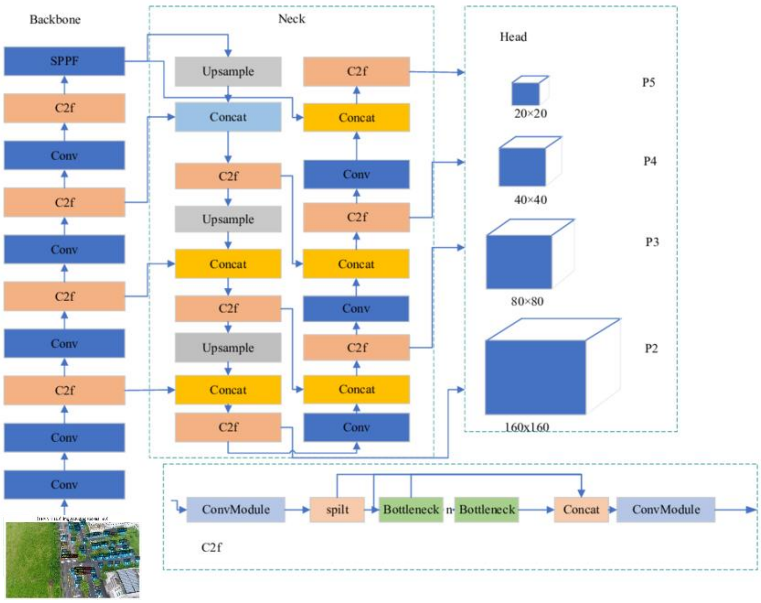
CVPDL HW2: Long Tail Object Detection

113034506 李家欣

一、Model Introduction

這次策略上想要走 Weighted-Boxes-Fusion (WBF)，也就是集合多個模型的輸出去做框的加權融合，來提升整體 AP 表現與穩定性，因此，我採用了兩種異構的模型去從零開始做訓練，比較有互補作用，分別選用傳統 CNN 架構的 yolov8n(增加 p2)，以及 Transformer 架構、具全域 self-attention 能力的 RT-DETR-L。

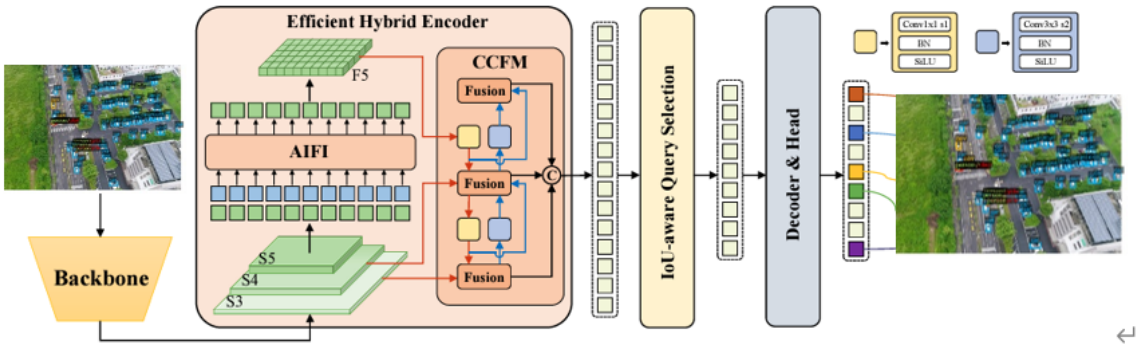
(1) YOLOv8n-P2



<圖一: YOLOv8n-P2 架構圖>

- **類型**：Anchor-free CNN 偵測器
- 核心改動: 新增 P2 輸出層（stride=4）
  相較於官方原版為 **P3/P4/P5，我**新增 P2 層（stride=4） 做四個尺度的輸出，目的是讓模型能在更高解析度下偵測小物，來針對空拍機圖像中偵測物件較小的問題做處理。
- 模型架構
  - Backbone: Conv + C2f × N + SPPF
  - Neck: FPN/PAFPN 上採樣與 Concat (P5→P4→P3→P2)
  - Head: Detect on [P2, P3, P4, P5] (stride=4,8,16,32)
- 模型特色
  - CNN 架構，推論速度快
  - P2 層增加小物件的辨識能力

(2) RT-DETR-L（Ultralytics 版，from-scratch）



<圖二: RT-DETR-L 架構圖>

- 類型: DETR 系列（Real-Time Detection Transformer）
- 模型架構
  - Backbone: 多層語義特徵(S3, S4, S5) 提取
  - Hybrid Encoder
    - **AIFI**（Attention based Intra-scale Feature Interaction）
    - **CCFM**（CNN based Cross-scale Feature Fusion Module）
  - Transformer Decoder: 使用 Object Queries 與 Multi-Head Attention 進行 set-based decoding
- 特色
  - 全域 self-attention、集合式預測（Set Prediction）
  - 對遮擋與密集場景表現佳

二、Implement detail

(1) 資料前處理:

- 資料切分:使用多標籤分層 K-fold 做 5 折
  - 確保每一折都能維持各類別的出現比例，比起隨機切分更能適應長尾分布問題造成的偏誤
- Downsampling: **car 類別**
  - 針對數量最多的 car 類別隨機保留一半數量，來平衡 imbalanced rate 從 17.305 到 3~5 倍的可接受差異(這個倍數的經驗值是課堂上老師提到的)
- Repeat Factor Sampling（RFS）
  - 受 LVIS 與 COCO 等長尾任務啟發，實作 Repeat Factor Sampling，我的策略是對 tail 類含量高的圖，根據重複係數 r(右邊公式)進行再採樣 ，以此確保稀有的類別在訓練過程中能被更多次看到

$$r = \sqrt{\frac{t_c}{\max(freq[c], \epsilon)}}$$

- Per-fold Tail Copy-Paste
  - 在每一折中，根據該折的 tail 類分布動態決定要 Copy-Paste 的對象與數量。

(2) 模型訓練

- 主模型為 **YOLOv8-P2 (小物體友善版本)**，搭配三折交叉訓練，以提升在小尺寸物體上的偵測能力，輔助模型採用 **RT-DETR**，其 Transformer-based 架構能在全域關注物體關係，來補足 yolo 偏向局部特徵的弱點
- 訓練優化器採 **AdamW**，學習率排程為 **Cosine LR decay**，並使用混合精度 (AMP) 加速與節省顯存

(3) 超參數 fine tuning

- 運用 grid search 方式進行超參數組合(imgsz、mosaic、hsv_s、confidence threshold、NMS IoU)尋找，先在 val 上跑出最好的組合，才去做測試集的推論

(4) 推論技巧:

- 異構 WBF 融合
  - 選擇 **YOLOv8-P2** 三折權重加上 **RT-DETR** 單折權重去做融合
  - score calibration: 實際運行上發現這兩種模型的 Confidence score 分布的形狀不同，所以有先針對不同的模型對齊到相同的 Temperature 縮放，避免 DETR 過度影響位置平均。
  - 另外，也有設計多模型都預測到時會做 score 信心加成；座標差異大會作方差的懲罰。
- Test-time augmentation(TTA)
  - 由於空拍視角的物體較小，我的策略是只保留水平翻轉、輕微縮放，避免破壞細節與比例，不去做選轉跟拉伸

三、 Result Analysis

1. 重要策略比較

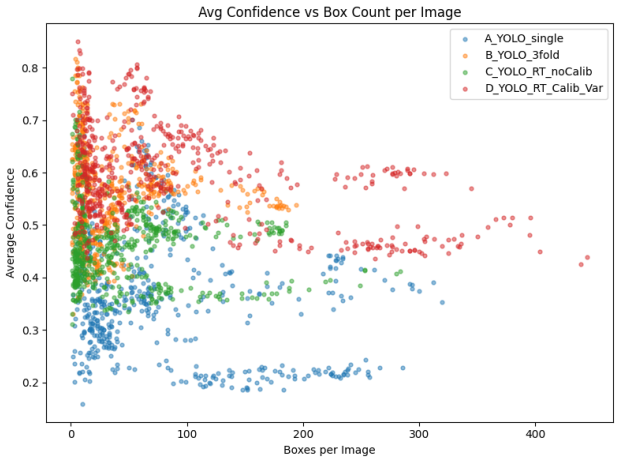
下表是我在策略調整過程的一些重要里程碑，由策略 a,b 可以看出: 使用三折 WBF 融合能平均掉單模型的預測不穩定，提升整體 Recall，而策略 b,d 可以看出增加異構模型對於表現有明顯提升。然而，後續從輸出預測圖與框去觀察時，我卻發現測試集中雖然都是空拍機視角，但是有分車子還是明顯大很多的近距離空拍機，以及非常遠拍攝的超小東西狀況，策略 b 在後者辨識效果特別差，甚至有出現完全零偵測的狀況。為了改善這個問題，我才決定增加異構模型 RT-DETR-L 模型看看。

策略	資料前處理	模型選用				WBF 策略	kaggle score
	data resampling	yolov8-p2 fold0	yolov8-p2 fold1	yolov8-p2 fold2	RT-DETR-L	score calibration	(public)
a	V	V	X	X	X	X	0.20201
b	V	V	V	V	X	X	0.2496
c	V	V	V	V	V	X	0.23352
d	V	V	V	V	V	V	0.28465

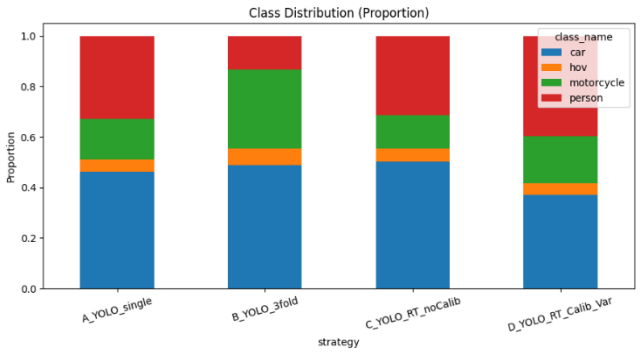
<表一 重要策略比較>

2. Confident score 的水準有異會造成反效果!!

有趣的是，後續的策略 c 和 d 都是使用到四個模型（YOLOv8-P2 三折 + RT-DETR-L 單折）做 wbf 融合，但初期結果卻意外下降至 **0.23352**，甚至比僅使用三種 YOLO 模型的策略還低。後來才發現有 score calibration 的問題(意即即便 confidence score 均為 0.9，在 YOLOv8 中通常代表極高確信的真實物件，而在 RT-DETR-L 中卻可能只表示中等信心水準)，所以造成框的權重受 **RT-DETR-L** 的分數偏低而被低估，連帶相同的信心閾值下，多個重疊框無法收斂為單一預測框，反而拉低整體 mAP，之後才透過新增 Temperature Scaling 跟 Variance Penalty，策略 **D** 的整體表現才有顯著改善。



<表 2 可看出紅色有較多且較高信心的預測框>



<表 3 策略 D 辨識類別的比例中 tail 物件 person 也比較好>

四、 Conclusion

這次任務使用 YOLOv8-P2 + RT-DETR 做異構模型 wbf 融合，並且也針對資料不平衡去做 resampling 跟的設計，切分資料集也需要特別注意了稀少類別的資料量，另外最不一樣的地方是: 第一次使用到 score calibration 與 geometric penalty 的技巧，對於異構模型特別重要。最後是小心得，我覺得這次作業比上次難不少，除了 long tail，我覺得預測物的尺寸也有大挑戰，所以發現需要一步步去觀察資料分布、特性，來去做設計，切分跟資料 resampling 都要應對不同類別去做設計，並且在模型設計上因為 OOM 的限制，所以也變得比較小心，導致我設計的模型訓練時間都超過 6 小時，實在是要注重耐心跟細心。

五、 Reference

https://blog.csdn.net/weixin_45277161/article/details/137775985
https://github.com/orgs/ultralytics/discussions/8227
https://zhuanlan.zhihu.com/p/26076573986
https://docs.ultralytics.com/zh/models/rtdetr/
https://openaccess.thecvf.com/content/ICCV2023/papers/Dong_Boosting_Long-tailed_Object_Detection_via_Step-wise_Learning_on_Smooth-tail_Data_ICCV_2023_paper.pdf
https://rumn.medium.com/the-long-tail-problem-in-object-detection-why-your-model-misses-rare-objects-and-how-to-fix-it-b9a45656b55a
https://github.com/ZFTurbo/Weighted-Boxes-Fusion