Long-tail object detection for drone-based counting

班級:電機碩一 、姓名:林郁庭、學號:114318058

1 問題概述

本次作業主要聚焦於利用無人機進行長尾分布物件的偵測，目標物件涵蓋汽車（car）、高空作業車（hov）、機車（motocycle）以及行人（person）等四種類別，且各類物件在資料分布上呈現長尾特性。所使用的訓練集包含 950 張解析度為 1920x1080 的高畫質影像。此外，分數以 mAP50:95 做為評估標準，模型訓練過程不得採用預訓練權重，並顯示記憶體（VRAM）使用需控制在 12GB 以內。

2 模型架構

本次報告中，我採用了影像切片（patch）作為資料前處理方法，並參考了[1]對小物件選用 yolov8s-p2 作為訓練模型，並加入了 class reweighted loss[2]，整體流程採用兩階段訓練策略。

本次實驗選用 YOLOv8s-p2 作為主要模型。YOLOv8 系列根據運算量與模型大小，細分為 n、s、m、l、x 等多種版本，能夠靈活因應不同的硬體資源與應用場景。此外，YOLOv8 具備多種電腦視覺任務的支援能力，包括物件偵測、語意分割、影像分類及目標追蹤等功能。此次專案聚焦於物件偵測任務，考量 VRAM 不得超過 12GB 的限制，故選擇較為輕量化的 s 版本進行實驗，以兼顧效能與資源利用。

YOLO 系列的網路架構主要分為三個核心部分：Backbone、Neck 與 Head：

- Backbone（主幹網路）：負責從輸入影像中萃取多層次特徵，為後續偵測、分類提供基礎資訊。
- Neck（頸部結構）：用於融合多尺度特徵，強化模型對不同大小物件的辨識能力，提升小物件偵測的表現。
- Head（輸出層）：根據前面萃取與融合的特徵，輸出物件的類別與邊界框

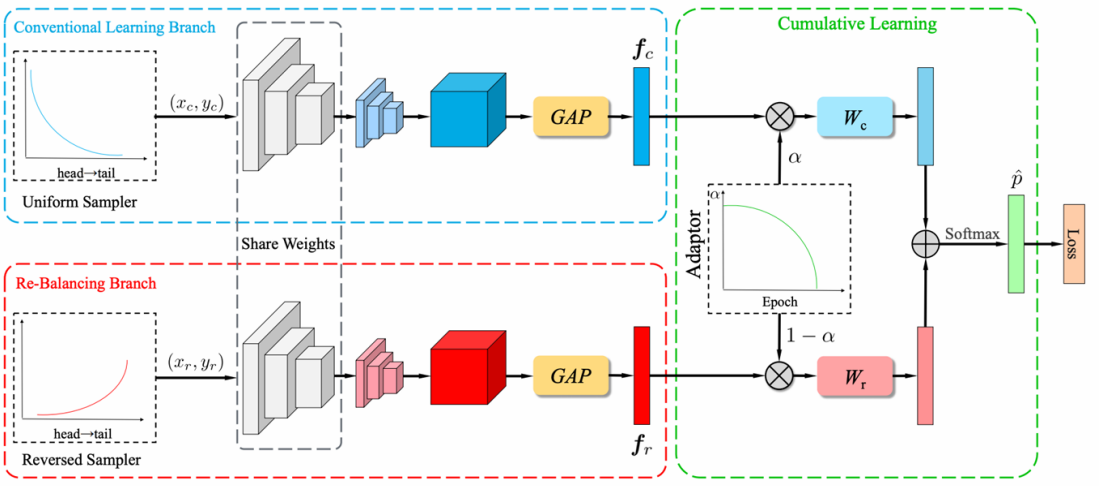
座標，完成最終的偵測任務。

值得一提的是，YOLOv8s-p2 在 特徵萃取器（Backbone） 部分，特別採用較小的卷積步長（stride），這種設計能夠捕捉更多細微特徵，有效提升模型對小型物件的敏感度與偵測精度。

3 實驗細節

2.1 訓練策略

根據圖二所示，本次實驗我參考[3]的實驗架構，並將其簡化為**訓練-微調兩階段**訓練策略，在訓練階段時將會用隨機初始化權重從頭訓練，學習資料的特徵表示。微調階段將會加入 class reweighted loss，並且凍結 backbone 和 neck，讓分類器學習較正確的資料分布，才不會過度偏向數量較多的物件類別。另外也根據小物件使用多尺度學習，以求更穩定的學習表現



圖二、實驗架構

2.2 資料前處理

針對高畫質影像以及小物件偵測的挑戰，如果直接原圖給模型訓練將會占用大量 VRAM。所以這裡的實驗採用 2X3 patch 的方式去將影像分割為 640x540 大小，再將無包含物件框的 patch 捨棄。如此一來便可以省去將近 40%的對學習無太大作用的資料，以達到更快的訓練時間並維持一樣的效能。

2.3 超參數和損失函數

本次實驗第一階段接統一訓練 100 epochs，微調階段訓練設定 30 ~ 50 epochs 不等。優化器採用 **SGD**[4]，並設定初始學習率(lr0)為 0.01。損失函數在第二階段引入 class re-weight loss，讓模型更有效地針對數量較少的分類學習，已達到更好的分類效果。

3 量化分析

本次實驗於 Kaggle 平台上使用 NVIDIA T4 GPU 進行訓練。模型設定中，批次大小（batch size）為 8，輸入影像尺寸（imgsz）為 640。在此配置下，VRAM 使用量約為 5 至 7 GB，顯著低於 12 GB 顯示記憶體上限。

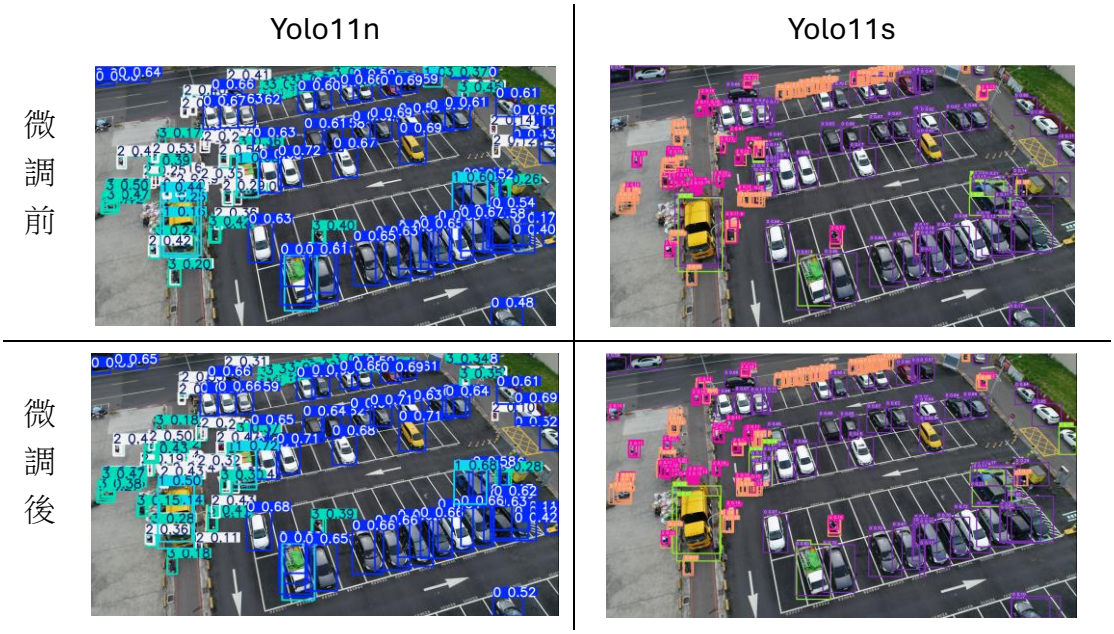
架構	訓練週期	reweighting 前 mAP50:95	reweighting 後 mAP50:95
YOLO11n+AdamW	100+50	0.170	0.180
YOLO11s+AdamW	100+30	0.184	0.180
YOLO11s+patch+AdamW	100+30	0.186	0.197
YOLO8s-p2+patch+SGD	100+30	0.197	**0.202**

表一、各個架構的實際表現

除了主實驗外，本報告亦針對是否採用影像切片（patch）訓練，以及是否引入 re-weighting 機制進行了比較分析。實驗結果顯示，無論是否採用 patch，推論時間皆穩定維持於 100 – 200 毫秒之間。

從表一可觀察到，無論是在 n 或 s 架構下，導入 re-weighting 機制在多數情況下都能為模型表現帶來不同程度的提升。此外，採用 patch 訓練不僅能增強模型在多樣場景下的穩定性與泛化能力，亦能有效降低背景雜訊的干擾，使模型能更專注於目標物件的特徵學習。

最後，實驗亦發現，針對不同任務選擇合適的優化器能帶來更佳的成效。參考[4]指出，SGD 在尋找較平坦極小點時展現出較好的泛化能力，因此在大量參數的訓練情境下，往往優於 Adam。實驗結果也證實，選用 SGD 確實能夠帶來更理想的表現。



圖三、實際測試圖

4 結論

本報告綜合運用了影像切片（patch）、Yolov8s-p2 架構，以及 class reweighted loss 等多項方法，並對各種組合進行了詳細比較分析。透過 patch 機制及 Yolov8s-p2 的特性，顯著提升了對小型物件的偵測能力；同時引入 class reweighted loss，有效緩解了資料長尾分布對模型表現的影響。然而，據悉部分同學採用其他組合後取得了更優的實驗結果，因此本報告也針對目前方法提出幾點待改進之處。

- 影像切片機制方面，建議可加入重疊（overlapping）設計，以適度保留上下文資訊，並針對邊界物件進行更精細的框選調整（本次實驗僅針對左上角座標進行修正）。
- 嘗試導入資料重抽樣（data resampling）方法，有望進一步提升模型對長尾分布資料的偵測與分類效能。

5 引用

[1] Zheng, Jinghan, Jie Guo, and Ming Hao. "Improved YOLOv8 for Small Aircraft Detection in SAR Images." *2024 7th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*. IEEE, 2024.

[2] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In CVPR, pages 9268–

9277, 2019.

[3] Zhou, Boyan, et al. "Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020.

[4] Zhou, Pan, et al. "Towards theoretically understanding why sgd generalizes better than adam in deep learning." *Advances in Neural Information Processing Systems* 33 (2020): 21285-21296.