

《AI 影像辨識專案》

【檢測學生午餐是否營養均衡】

專案製作人：

田祐安

目錄

研究動機	3
研究過程	4
結果分析	5
Confusion Matrix-Normalized	5
P、R、PR、F1-Curve	6
Loss Function	10
Labels. jpg	13
Labels correlogram. jpg	14
val_batch0	15
train_batch	15
研究總結和心得	16
參考資料	17

研究動機

此專案的研究動機起源於我自身在成長過程中常因身材因素被同儕霸凌，但高中時期的課業繁忙導致我沒有多餘的時間運動，因此只能用控制飲食攝取來達成減肥目標。此專案的目的是利用影像識別技術來檢測餐點食材進而判定該餐點是否營養均衡，能應用在學校教育端提供學生和師長快速判別所攝取的學餐或便當是否營養均衡。

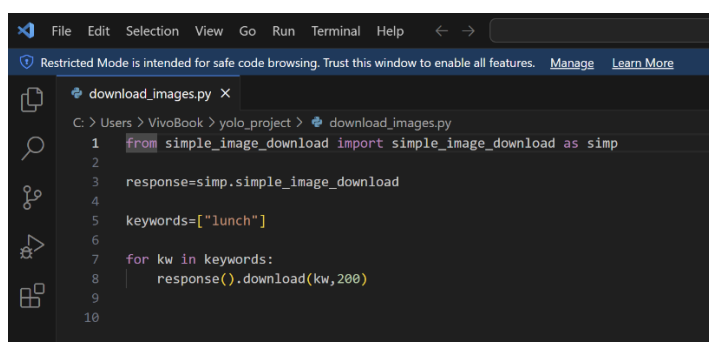
除了上述動機外，我也希望藉由此次專案製作更加瞭解除除了競程外，程式設計和資訊領域在實作方面的操作過程和成果分析。

研究過程

【建立虛擬環境】

在 anaconda prompt 中建立新環境，並下載圖片爬蟲程式(simple image download)

【爬蟲程式】



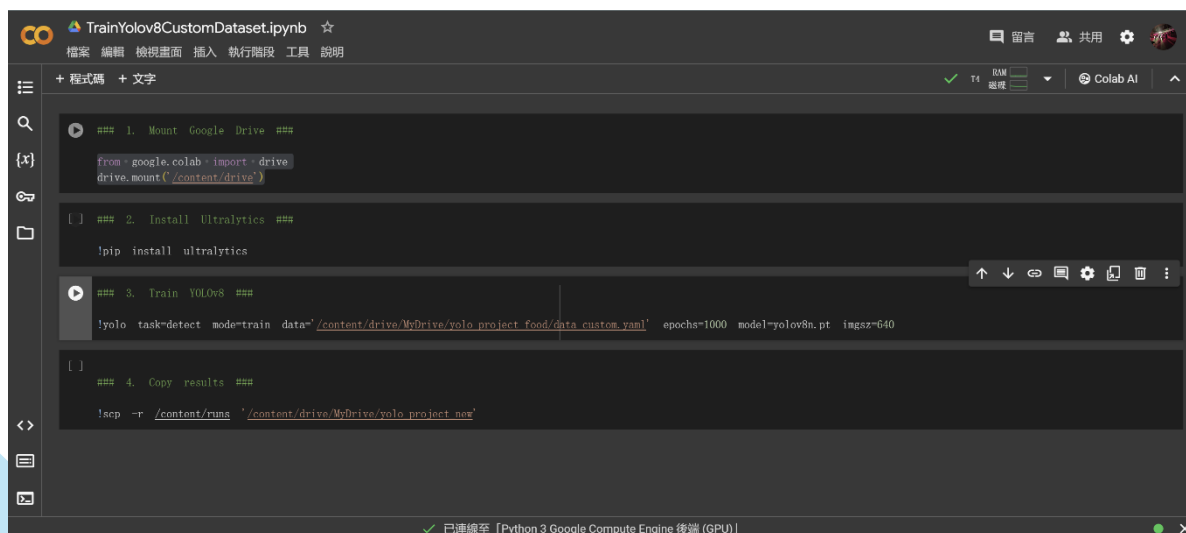
```
1 from simple_image_download import simple_image_download as simp
2
3 response=simp.simple_image_download
4
5 keywords=["lunch"]
6
7 for kw in keywords:
8     response().download(kw,200)
9
10
```

【標框】

使用 labelImg 將剛剛抓到的圖片一個個手動標框

【訓練和匯出結果】

由於我的筆電只有內顯，所以借用 googlecolab 的 T4 GPU 來訓練模型。



```
1. Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

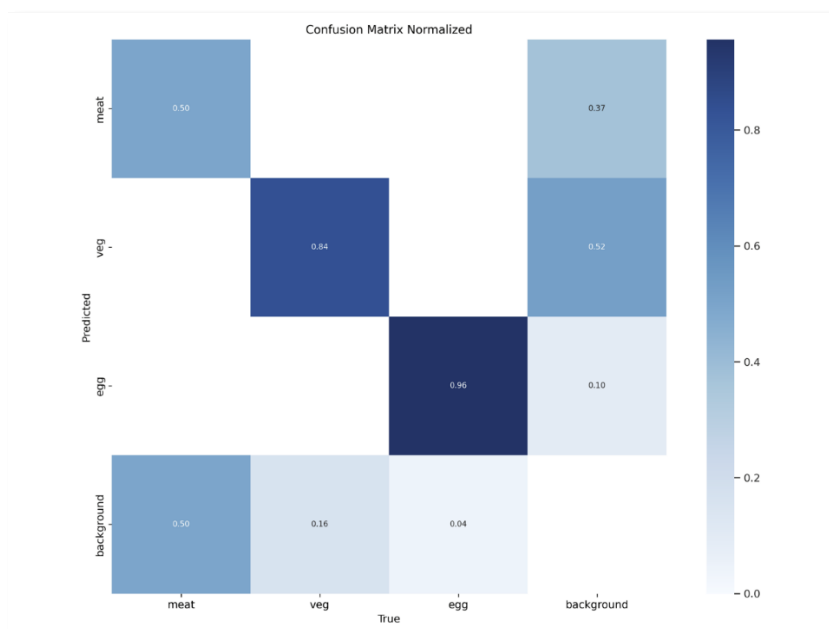
2. Install Ultralytics
!pip install ultralytics

3. Train YOLOv8
!yolo task=detect mode=train data='/content/drive/MyDrive/yolo_project_food/data_custom.yaml' epochs=1000 model=yolov8n.pt imgsz=640

4. Copy results
!scp -r /content/runs '/content/drive/MyDrive/yolo_project_new'
```

結果分析

Confusion Matrix-Normalized



Confusion Matrix-Normalized

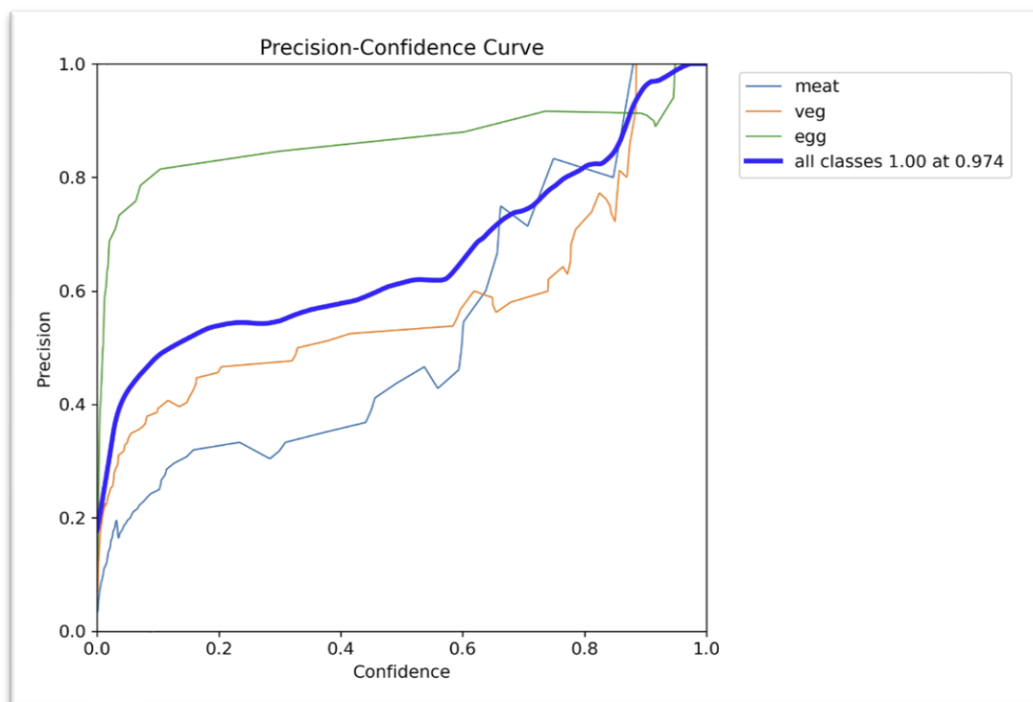
這個圖表是混淆矩陣，也就是模型預測正確數量所佔整體的比例，使用 TP(True Positives)、TN(True Negatives)、FP(False Positives)、FN(False Negatives)來繪製。

$$\text{公式: Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

由圖可知雞蛋(egg)的準確率最高，為 0.96；其次為蔬菜(veg)，為 0.84；肉類(meat)最低，為 0.50。此外肉類最常被誤認為背景(background)，為 0.50；而背景最常被模型誤認為蔬菜。

此計算方式對於資料比例懸殊的情況並不具參考價值，因此需要下列指標來完成。

P、R、PR、F1-Curve

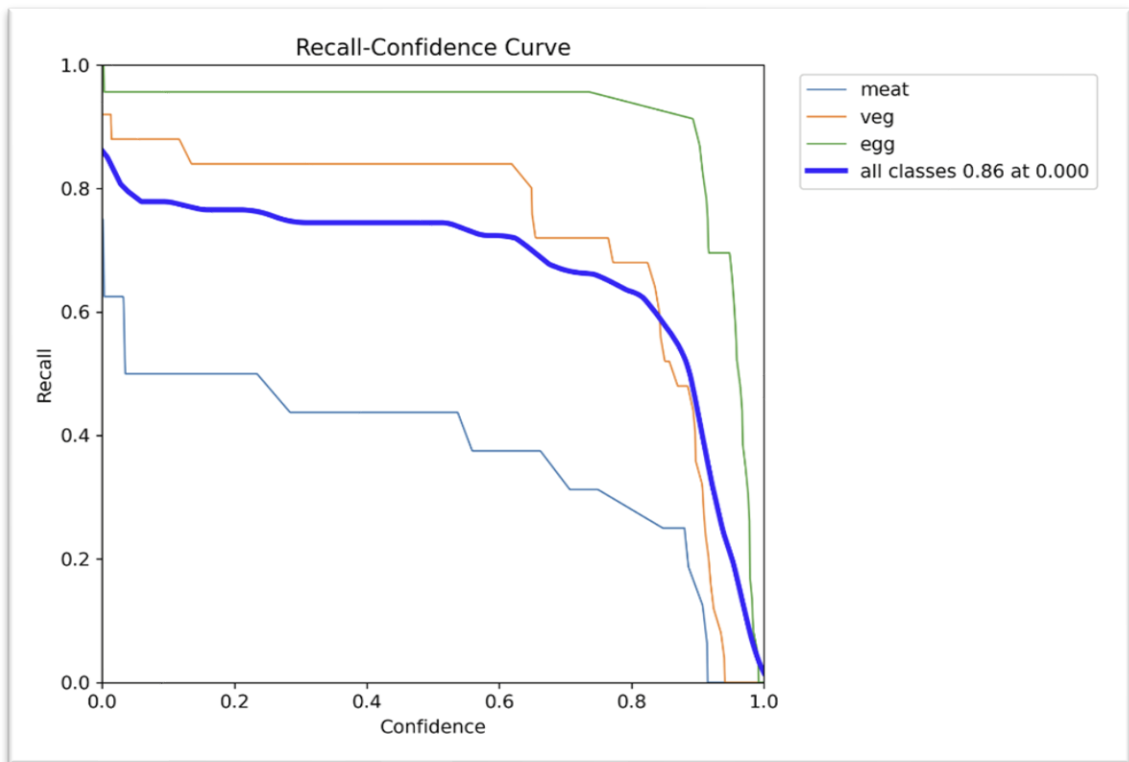


P Curve

Precision-Confidence Curve 為被預測為 Positive 的資料中，有多少是真的 Positive，並結合模型信心程度作為 X 軸來繪製。

$$\text{公式: Precision} = \frac{TP}{TP+FP}$$

依照定義，我們會希望模型在越低的信心有越高的精確率，即圖形往左上形成開口向下的遞增曲線(完全沒信心時歸零)，而在此指標下蛋的精確率在信心程度 0.2 下有著最高的精確率 0.8；蔬菜和肉類則在信心程度 0.8 時才有精確率 0.8。

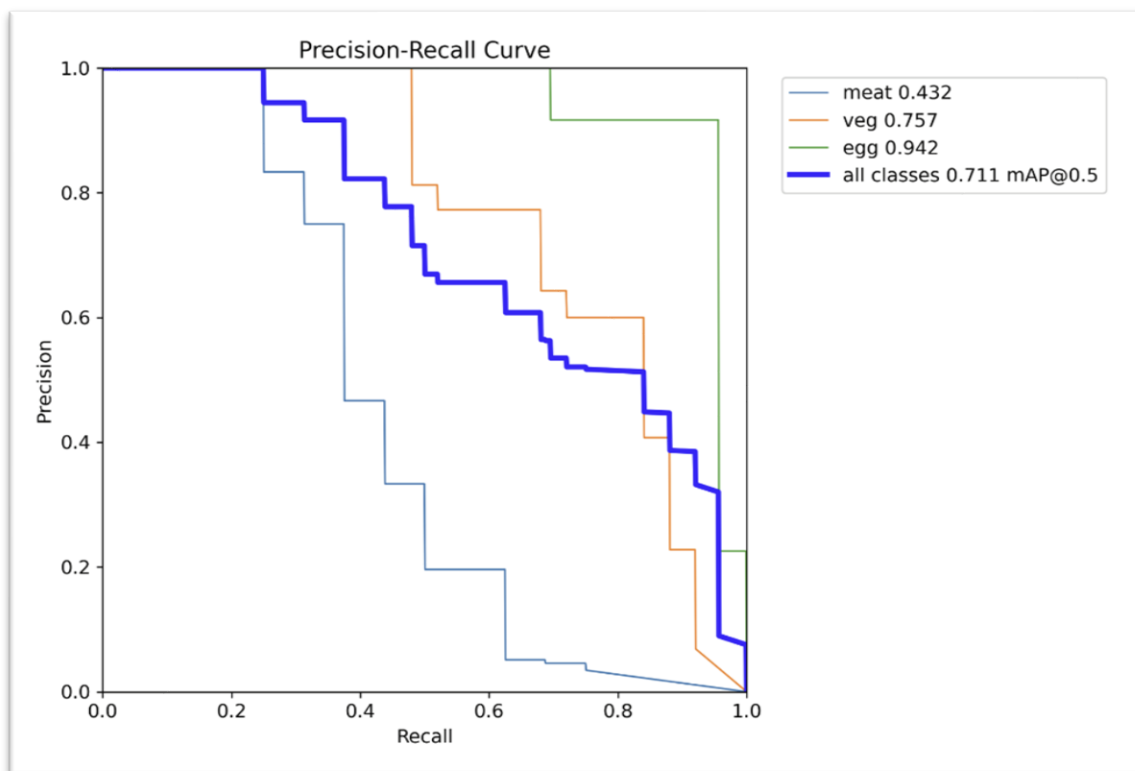


R Curve

Recall-Confidence Curve 為在原本 Positive 的資料中被預測出多少，並結合模型信心程度作為 X 軸來繪製。

$$\text{公式: Recall} = \frac{TP}{TP+FN}$$

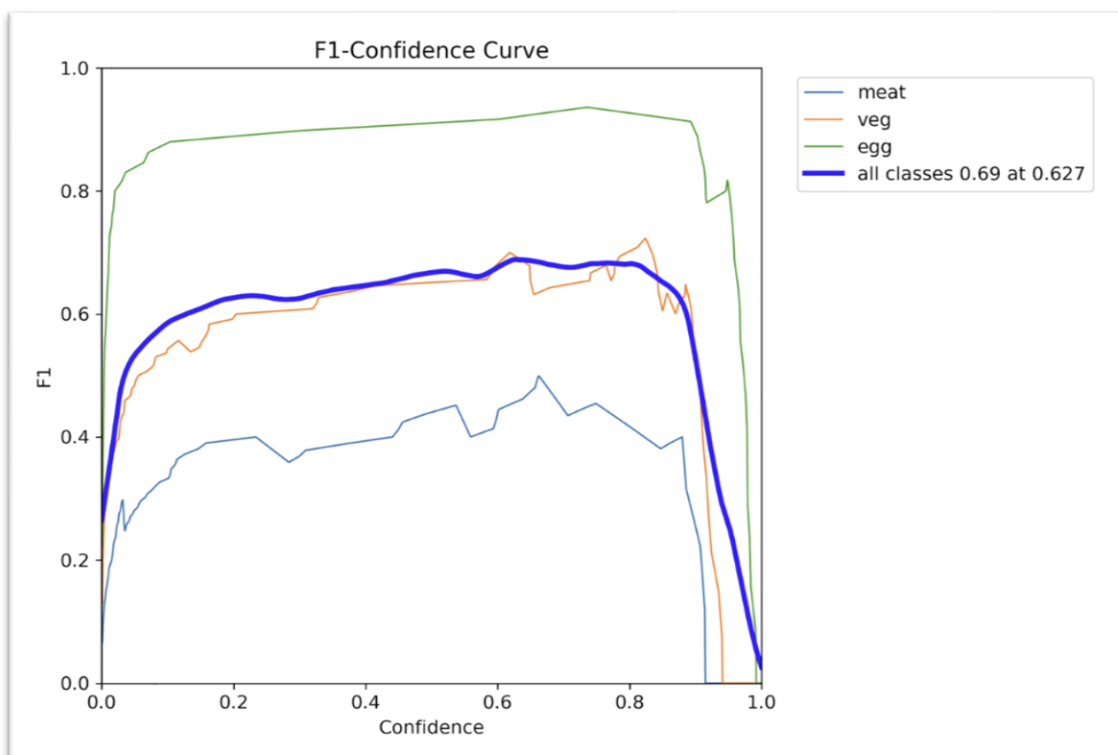
按照定義，在信心程度低時，召回率高，但誤判率也會相對高，蛋的表現最好(圖形往右上最好)，蔬菜的表現接近平均，肉類的表現最差，一開始就掉到 0.6、0.5。



PR Curve

Precision-Recall Curve 反映精確度和召回率的關係，而 mAP 代表 Mean Average Precision 平均精確度，也就是曲線下面積，若 PR 曲線往右上凸出，表示 (P, R) 越接近 $(1, 1)$ ，曲線下面積趨近於 1，也就是完美預測，反之則否。而 PR Curve 由於沒有考慮 TN (True Negatives)，因此適用於樣本類別不平均的資料集。

我們發現蛋的 PR 曲線最接近完美預測，AP 為 0.942；表現最差的則為肉類，AP 為 0.432；平均面積為 0.711。



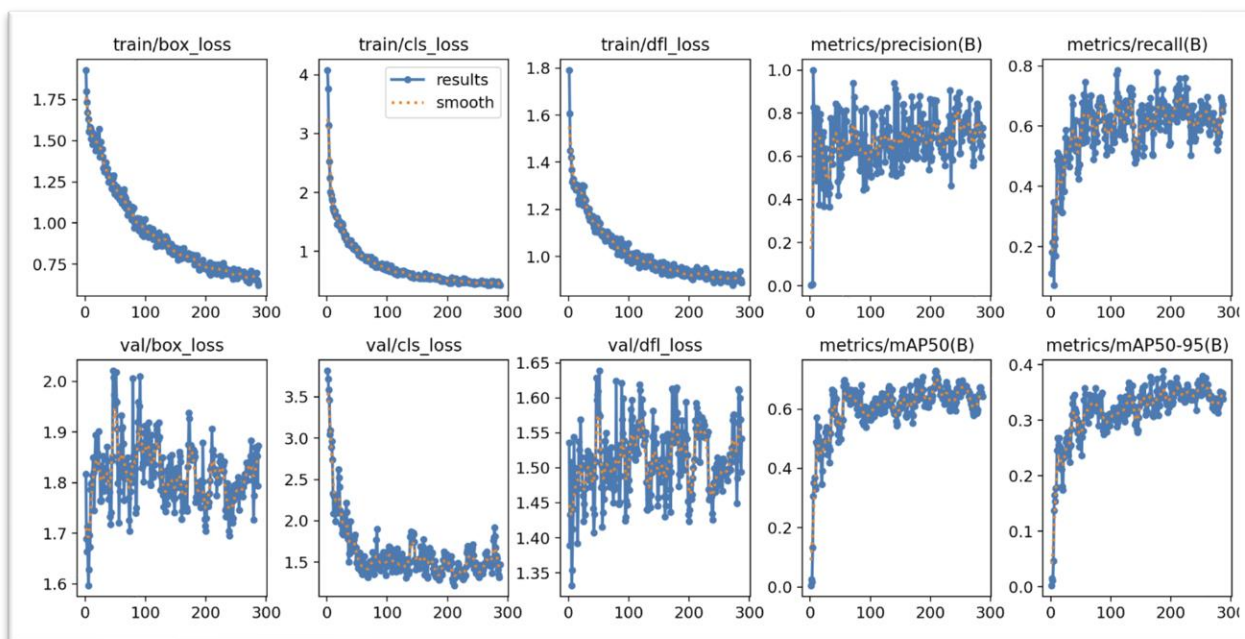
F1-Curve

由上述兩指標可得知不同情況下所關注的指標不同，也可能要同時考慮到兩者，這時候將精確率和召回率做調和平均做為新的指標，就是 F1-Score。在樣本不平衡的資料集上，F1-Score 通常更能反映模型的表現。

$$\text{公式: } F1 = \frac{2(PR)}{P+R}$$

從圖中可知當 Confidence 為 0.627 時最能優化 F1，此時的 F1 為 0.69。

Loss Function



Loss Function

損失函數為衡量深度學習模型的預測值與實際值之間的差距(或稱殘差)，以下個別介紹各圖表的意義。

【box loss】

這是用於預測框位置的損失函數。Box loss 的目標是使模型的位置預測更接近真實目標框的位置，這有助於提高目標定位的準確性。

在我原本使用的 train 圖片集中，Box loss 呈現平滑曲線，但在 validation 圖片集中的表現不太好，有可能是蔬菜和肉類通常為不規則圖形，而 YOLO v8 的框選只有矩形，因此造成模型無法準確預測框的座標。

【cls loss】

Cls loss 用於分類目標的類別，檢測目標分類的準確性。Cls loss 的作用是確保模型的類別預測盡可能準確。

由於我的 YOLO 模型只有三個 class(蛋、蔬菜、肉)，因此曲線是這幾項 loss 中最接近收斂的曲線。我最一開始設定的 class 高達 16 項，但經由 googlecolab 跑完後的數據差強人意，因此將類別減少，增加訓練圖片集，曲線才能勉強收斂。

【dfl loss】

焦點損失 (Focal Loss) 的主要目標是應對樣本不平衡問題，就是正負樣本比例嚴重失衡。dfl_loss 不僅考慮到少量類別，還注意到模型的預測如何與實際類別分佈相符。

我的驗證模型的 dfl loss 之所以如此脫序的原因有可能是訓練數不夠多，val 的圖片也不多，在混淆矩陣中可以看出肉類最常被誤認為背景，而背景最常被誤認為蔬菜，導致難以分類的正樣本太多，使數據更不平衡。

【precision(B)】

【recall(B)】

precision(B)是模型在預測中有多少是正確的，相對於所有預測中的正確和錯誤；recall(B) 是模型正確預測的目標數與實際目標數之比，而當中的(B)代表正樣本，因為公式中沒有考慮到 TN。

而精確率的曲線較不穩定的原因也應該歸咎於肉類的訓練數量不夠和常被誤認為背景。

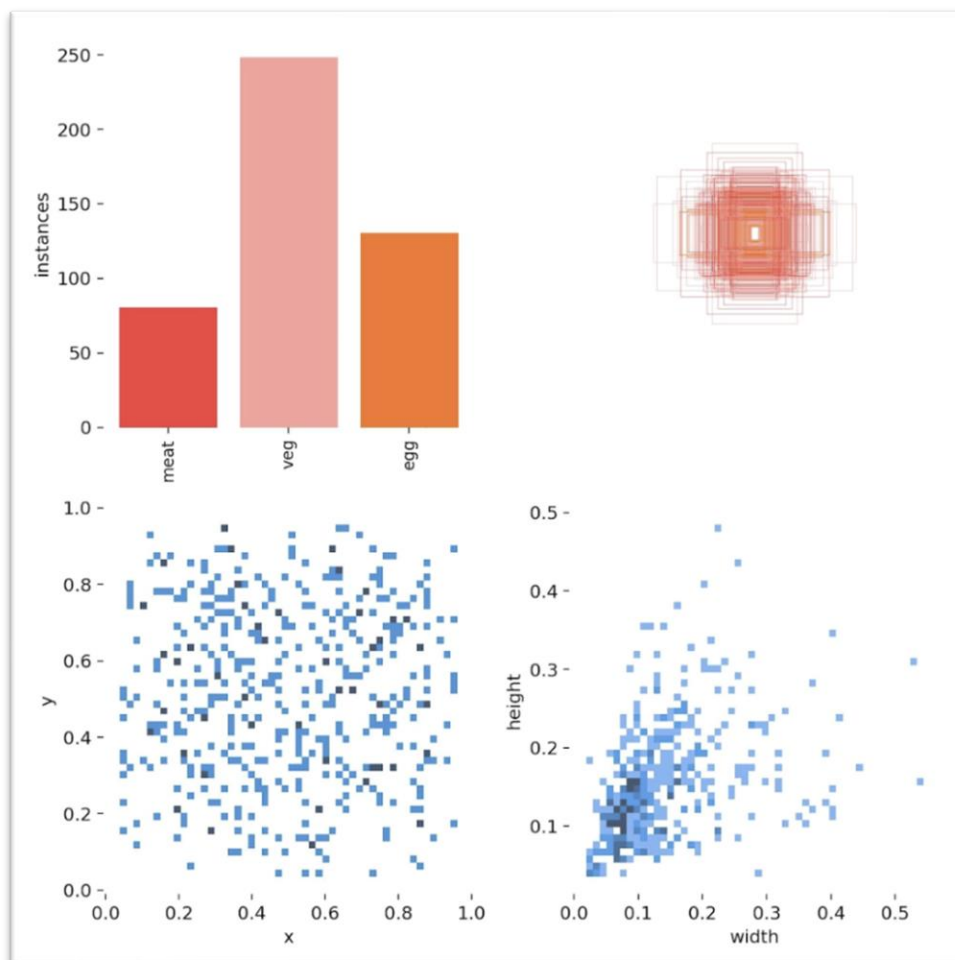
【mAP50(B)】

IOU (Intersection over Union) 主要用於衡量預測的目標與實際目標之間的重疊程度。公式為 $IOU = \frac{AREA\ OF\ OVERLAP}{AREA\ OF\ UNION}$ 。而 mAP50(B) 意思是在 IOU 為 0.50 的情況下，針對較大目標計算的 mAP。在模型圖表中可以發現圖形在 0.6 收斂。

【mAP50-95】

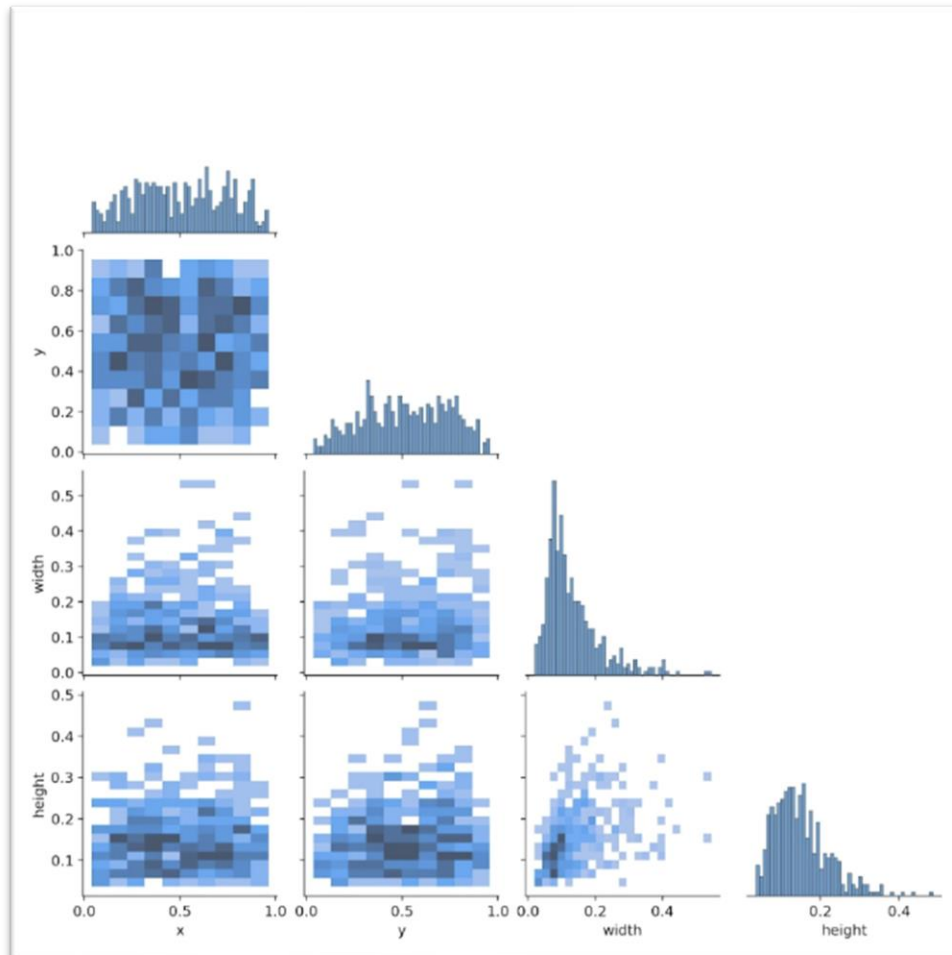
mAP50-95 表示在 50% 到 95% 的 IOU 閾值範圍內的平均精確率，這個指標更全面地評估了模型在不同重疊度下的性能。在圖表當中收斂於 0.35 左右。

Labels.jpg



各類別有幾個	框的尺寸和數量
中心點相對於圖的位置	圖中目標相對於整幅圖的高寬比

Labels correlogram.jpg



中心點座標和框的高寬比例關係

val_batch0



train_batch



研究總結和心得

從上述的數據圖表中我們得知，蛋的表現一直都不錯，原因可能是蛋在料理中的形狀和顏色都十分相似(頗面)，也不太會被誤認成其他食材和背景，因此辨識度高。蔬菜的部分雖然也具有辨識度高的綠色，但因為品種太多，色度太深的部分會與背景搞混，表現適中。肉類的表現一直是最差的，除了樣本個數較少外，肉類的形狀和顏色有太多不同的類型(烹煮方式、品種、切法都會影響)。

其實 YOLO V8 的操作過程不難，但結果的解釋和數據背後的意義花了我很多時間理解。在自學 C++、python、YOLOv8 的過程中我利用我的英文專長查找了 Stack Overflow、github、Ultralytics YOLOv8 官網、Medium 等論壇和網站來獲取更多訊息，我在做完此專案後對於 YOLOv8 更感興趣，也正在學習 YOLOv8-Pose，希望未來能做出動態捕捉等更進階的專案內容。

參考資料

https://www.youtube.com/watch?v=gRAyOPjQ9_s&t=194s

-Complete YOLO v8 Custom Object Detection Tutorial | Windows & Linux

<https://github.com/ultralytics/ultralytics>

<https://claire->

[chang.com/2023/08/16/yolov8%E6%A8%A1%E5%9E%8B%E8%A8%93%E7%B7%B4%E5%8F%8A%E5%85%B6%E6%8C%87%E6%A8%99%E6%84%8F%E7%BE%A9/](https://claire-chang.com/2023/08/16/yolov8%E6%A8%A1%E5%9E%8B%E8%A8%93%E7%B7%B4%E5%8F%8A%E5%85%B6%E6%8C%87%E6%A8%99%E6%84%8F%E7%BE%A9/)

-YOLOv8 模型訓練及其指標意義

<https://stackoverflow.com/questions/75950283/yolov8-dfl-loss-metric>

-yolov8-dfl-loss-metric

<https://jianjiesun.medium.com/dl-ml%E7%AD%86%E8%A8%98-21-yolo-v4-reference-2-data-imbalance-bounding-box-regression-e97c252d24a1>

-DL、ML 筆記(21): YOLO v4 Reference(2): Data Imbalance、Bounding Box Regression