

方法描述

1 問題定義

本專案旨在解決集約化養殖環境中的豬隻自動偵測問題，在該場域，有許多靠得很近的密集豬仔，所以主要挑戰包括：

- **密集遮擋**：豬隻習慣群聚，導致個體間相互重疊，僅露出頭部或背部特徵，傳統偵測器極易發生漏檢或邊界框合併。
- **多視角變異**：影像來源包含左側與右側架設之攝影機。由於安裝角度不同，豬隻在影像中的幾何投影存在明顯差異。
- **極端光照條件**：資料集涵蓋日間自然光（RGB 頻譜）與夜間紅外線補光（IR 成像）。夜間紅外影像缺乏色彩資訊，紋理特徵與日間截然不同。

2. 資料預處理策略

為應對上述挑戰，本實驗採用了「全數據驅動」與「分佈一致性」的預處理策略。

- **資料集分割**：原始資料集共計 9,700 張影像。考量到 YOLO11x 大型模型對數據量的需求，以及影像資料具有連續序列特性，傳統的 70:30 切分會導致訓練樣本不足以覆蓋所有稀有視角。因此，本實驗提交的最佳權重採用 9200 (Train) : 500 (Val) 的切分比例，與此報告提到的權重不同(此報告為 70:30)。
- **標籤分佈分析**：依據 labels.jpg 之統計結果，顯示豬隻主要分佈於影像中下區域，符合俯視監控視角特性。邊界框尺寸中(Fig 1)，側面視角較扁長、正面視角較方正，證實了左/右視角造成的透視形變，所以訓練時可以進行左右映射，避免模型偏重某一方向。

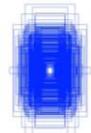


Fig 1 豬仔形狀分布

3. 模型架構設計

本實驗選用 **YOLO11x (Ultralytics v8.3.0+)** 作為核心偵測架構[1]。選擇 Extra Large 版本的動機如下：

- **參數量與特徵容量**：YOLO11x 擁有最大的參數量（56M parameters），具備強大的特徵萃取能力。面對「密集重疊」場景，小型模型易因特徵圖解析度不足而無法區分相鄰目標，而 x 版本能保留更多高頻細節。
- **架構改進**：YOLO11 引入了改進的 C3k2、C2PSA 模組與 SPPF[1]，處理不同大小目標時表現更優異。
- **Head**：採用 Anchor-free 的 decoupled head 設計[2]，將分類與回歸任務分開處理，有效解決了豬隻特徵相似但位置重疊造成的衝突問題。

訓練流程

1. 環境設置與參數配置

硬體配置為 NVIDIA L4 GPU (24GB VRAM)。軟體環境為 PyTorch 2.x 與 Ultralytics 框架。詳細參數設置：輸入解析度=1280、Batch size=4、Optimizer=AdamW[3]、learning rate=從 0.00005 至 0.01、Epoch 設定 40 (實際執行到 17 Epoch 就不讓他跑了)。

2. 數據增強

針對「左右視角」與「日夜光影」，實施了針對性的增強：

- **水平翻轉 (fliplr: 0.5)**：將影像以 50% 機率進行水平鏡像翻轉，消除左右相機視角的偏差。
- **HSV 色彩空間擾動 (hsv_h: 0.015, hsv_s: 0.7, hsv_v: 0.4)**：模擬從日間自然光到夜間 IR 的光譜變化。特別是飽和度與亮度的劇烈擾動，強迫模型不依賴色彩特徵，轉而依賴紋理與輪廓特徵，提升對 IR 灰階影像的穩定性。
- **幾何擾動 (translate: 0.1, scale: 0.5)**：模擬豬隻進出畫面邊緣與遠近變化。

3. 損失函數與評估指標設計

- **損失函數**：**ClOu** 為考慮重疊面積、中心點距離與長寬比，優化邊界框定位；**Cls Loss** 為二元交叉熵損失，用於類別分類（本案僅一類）；**DFL Loss**：解決邊界模糊問題，判定密集遮擋下的邊緣。
- **自定義適應度函數**：修改 train.py 中的 fitness 邏輯，將權重全數集中於 mAP@0.5 (return x[2])。

實驗結果與視覺化

1 訓練與驗證損失的背離

Training AP50 呈現完美的下降趨勢(0.97)，這表明模型具有極強的學習能力，能夠充分擬合訓練數據中的特徵；但 Validation AP50 呈現出上升或停滯 (0.912~0.922) 趨勢，這是典型的 Overfitting。

訓練損失的持續下降與驗證損失的持續上升形成了「剪刀」特徵。特別是 val/dfl_loss 的顯著惡化(Fig 2、3)，代表模型在未見過的測試數據上，對於預測邊界框的「確定性」越來越差。儘管它還能識別出豬，但它越來越無法確定豬的邊緣在哪裡。這問題在密集堆疊的豬圈中非常嚴重，因為模型可能將兩頭緊挨著的豬框在一起，或者框選範圍過大/過小。

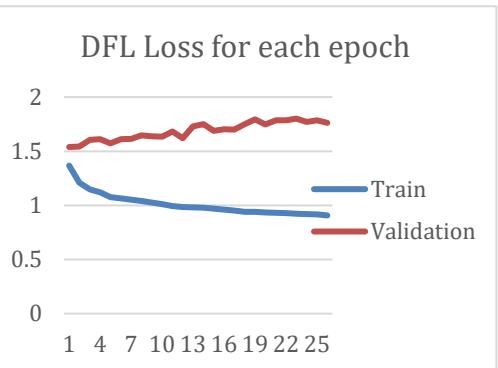


Fig 2 DFLLoss

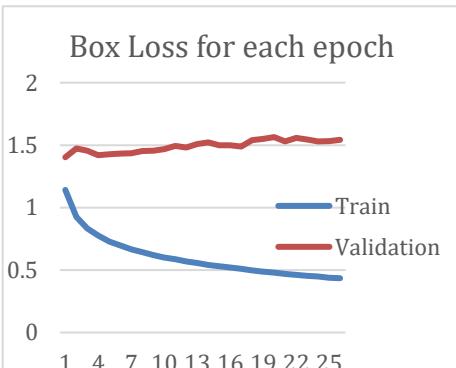


Fig 3 Box Loss

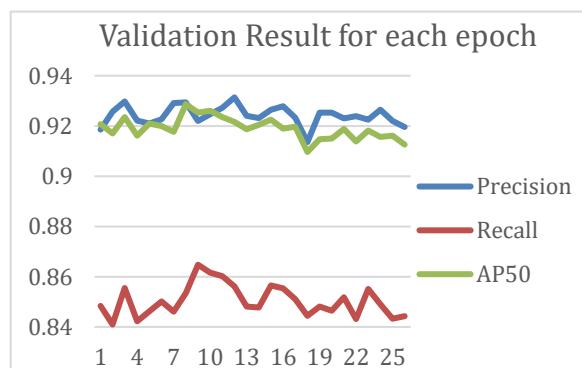


Fig 4 驗證集精確度表

- **Precision:** 穩定維持在 0.92 左右的高位。這顯示模型非常謹慎，極少產生誤報。只要它預測是豬，幾乎就是豬。
- **Recall:** 維持在 0.84 - 0.85。這意味著約有 15% 的豬隻未被檢出。考慮到高密度環境，這些漏檢很可能發生在嚴重遮擋或夜間紅外影像不清的區域。
- **mAP@50:** 高達 0.92。這證明模型在 IoU 門檻為 0.5 時表現優異，基本滿足了計數或存在性檢測的需求。
- **mAP@50-95:** 從初始的 0.575 緩慢下降至 0.545。

整理回報：mAP@50 與 mAP@50-95 之間的巨大落差 (0.92 vs 0.55) 且後者隨訓練下降，直接證實了我對損失函數的分析：模型正在犧牲定位精度來換取分類信心。由於我們在 train.py 中自定義了 Fitness Function 只看 mAP@50，訓練過程實際上是在「獎勵」這種行為——只要框到了 ($\text{IoU} > 0.5$)，框得準不準 ($\text{IoU} > 0.75+$) 並不重要。這導致模型在處理密集目標時，邊界回歸能力退化。

3 視覺化推論結果分析

雖然由於豬隻過於密集(Fig 5)，難以利用推論圖形來判別分類表現，但可以藉由測試 test.py 的 mAP50 最高值 ($\text{conf}=0.25, \text{iou}=0.55$) 及 Confusion Matrix (Fig 6)，可以推斷視覺結果：在豬隻休息堆疊時，模型可能會產生合併框（一個框包含多頭豬）或漂移框（框的邊緣切到了旁邊的豬）。由於紅外影像缺乏色彩對比，豬隻與地面的溫差若不明顯，邊緣特徵將極其微弱。模型可能依賴「斑點」特徵而非「輪廓」特徵進行檢測，導致邊界框並不貼合豬隻身體，這可能是 dfl_loss 在驗證集上會升高的原因。另外，跟 mAP50 計算相關的 Precision-Recall curve 如圖(Fig.7)。

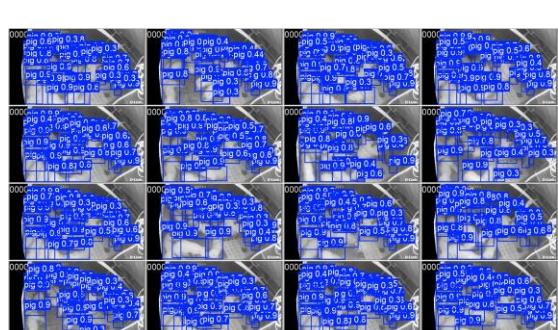


Fig 5 Inference (Visualized)

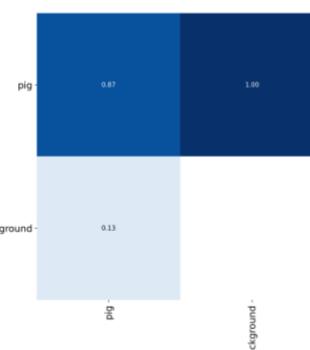


Fig 6 Confusion Matrix

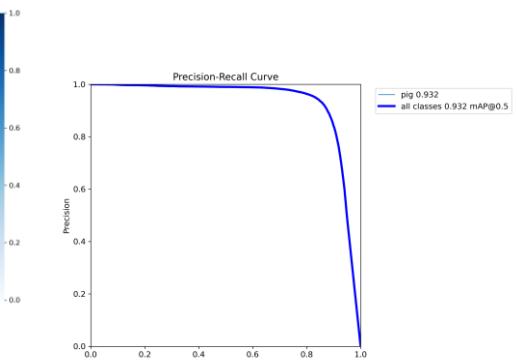


Fig 7 Precision-Recall Curve

1. 過擬合的成因與解構

本實驗中嚴重的過擬合現象並非偶然，而是多重因素疊加的結果：

- **數據增強的缺失**：關閉 mosaic 和 mixup 是最主要的失誤。YOLOv11-X 擁有 5600 萬個參數，具備極強的記憶能力。Mosaic 增強本應通過拼接圖片來創造虛擬的「邊緣截斷」和「小目標」場景，強迫模型學習局部特徵。缺乏此機制，模型傾向於記憶特定視角下的豬隻整體形態，一旦測試集中的豬隻姿態或堆疊方式稍有不同，模型便無法適應。
- **領域偏移**：訓練集涵蓋了日夜兩種模式。若訓練數據中日間圖片佔主導，模型會學習依賴可見光紋理。當遇到夜間 IR 圖片時，紋理特徵消失，模型在源域學到的特徵在目標域失效，導致泛化誤差上升。
- **優化目標偏差**：強制模型只優化 mAP@50，使得模型在訓練後期不再微調邊界框的精準度，而是滿足於「差不多框住」的狀態。

2. 高密度遮擋的挑戰

在豬舍環境中，遮擋不僅是物體間的重疊，還包括欄杆、飼料槽等設施的遮擋。YOLOv11 的 Attention 機制 (C2PSA) 理論上能緩解此問題，但若缺乏足夠多樣化的遮擋樣本訓練（如 Mixup 模擬的半透明遮擋），Attention 機制可能將遮擋物誤判為背景特徵。目前的 Recall 停滯在 0.85，說明約 15% 的嚴重遮擋案例仍是盲點

優化策略

1. 資料集重組與全域隨機分配

- **策略**：將原始訓練集與測試集完全合併後，進行全域隨機打亂，再依比例重新劃分。
- **目的**：解決原始資料集可能存在的場景分佈不均問題，消除特定視角或背景的偏差。**(最終採納)**

2. Mosaic 增強策略重啟

- **策略**：在訓練階段重新啟用 Mosaic 數據增強技術。
- **目的**：增加訓練樣本的多樣性與複雜度，有效抑制模型 Overfitting，提升對密集與遮擋目標的泛化能力。**(最終採納)**

3. 模型架構更換：RT-DETR [4]

- **過程**：嘗試引入 Transformer-based 的 RT-DETR 架構進行比較實驗。
- **結果**：實驗顯示其收斂成效不如預期，mAP@50 僅達 0.90，未能超越現有 YOLO11x 模型，故最終未採用。

4. 切片推論技術評估：SAHI [5]

- 過程：嘗試導入 SAHI (Slicing Aided Hyper Inference) 技術，將高解析度影像切片以優化微小目標偵測。
- 結果：由於 SAHI 的輸出格式與 Ultralytics 標準評測指標整合困難，無法進行統一且標準化的量化評估，故忍痛捨棄此方案

最終修正

採用了 資料集重組分配、與 Mosaic 增強策略($mosaic=1, mixup=0.1$)，並將總 epoch 設為 20 (前 19 設置 mosaic, 後 1 epoch 關閉)。根據結果，開啟 mosaic 階段，模型的 AP@0.5 與 Box Loss 呈下降趨勢，但關閉後，呈現準確率、框率下降，足以展現馬賽克的重要性，所以我們採用 Epoch=19 的權重。

我分別使用原始最佳 與 修改後最佳 的權重做推論比較 (資料為修改後的測試集)，測試結果於 Table 1.，顯示新的訓練方式在 Precision、Recall 與 AP@0.5 皆有些許的提升，且新的權重尚未看過測試集資料，所以我最終選擇新的權重上傳。

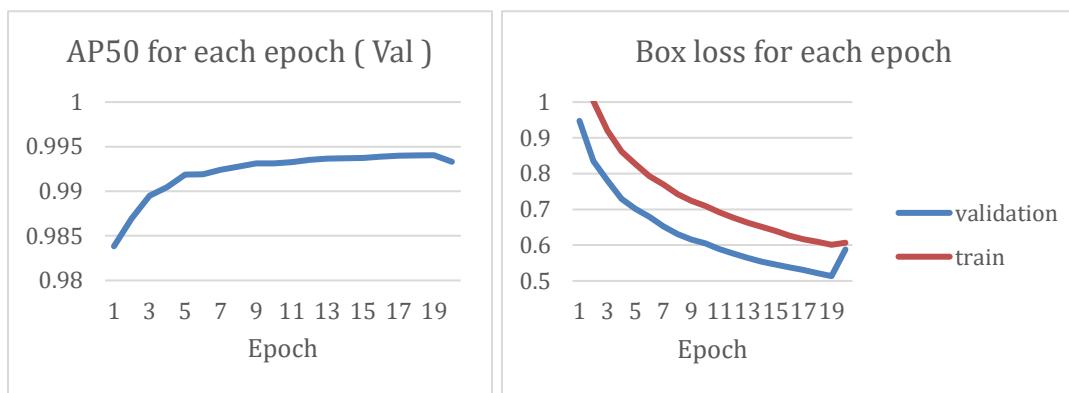


Fig. 8 AP50 for new training

Fig. 9 Box loss for new training

| Best | Old | New |
|-----------|--------|--------|
| Best IOU | 0.6 | 0.65 |
| Precision | 0.9869 | 0.9869 |
| Recall | 0.9808 | 0.9835 |
| AP50 | 0.9940 | 0.9941 |
| AP50-95 | 0.9026 | 0.9048 |

Table. 1 Comparison with the best weight via old and new way

Reference

- [1] G. Jocher et al., "Ultralytics YOLO," Ultralytics, 2024. [Online].
- [2] Z. Ge et al., "YOLOX: Exceeding YOLO Series in 2021,".
- [3] Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *Proc. ICLR*, New Orleans, LA, USA, May 2019.
- [4] Yian Zhao et al., "DETRs Beat YOLOs on Real-time Object Detection," in *Proc. IEEE/CVF Conf. CVPR*, Seattle, WA, USA, June 2024
- [5] F. C. Akyon, S. O. Altinuc, and A. Temizel, "Slicing Aided Hyper Inference and Fine-Tuning for Small Object Detection," in *Proc. ICIP*, Bordeaux, France