BÁO CÁO CUỐI KÌ

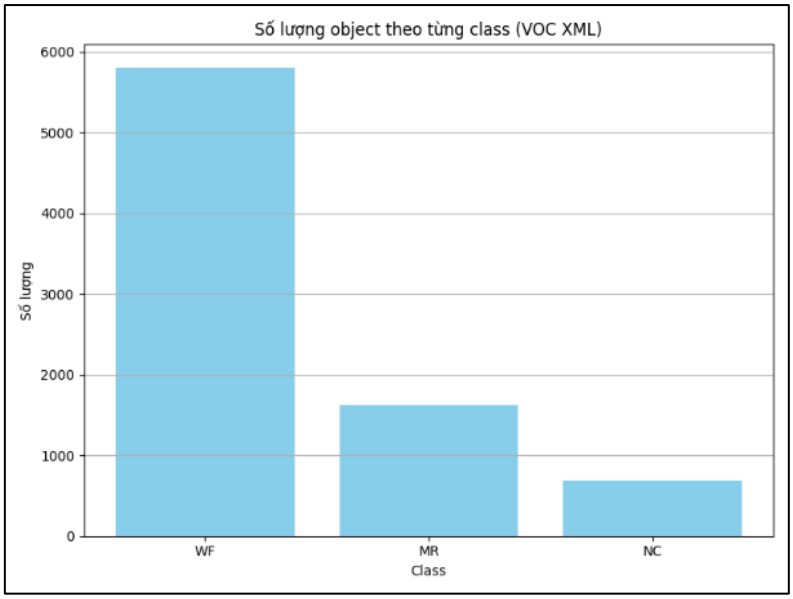
Môn: Học máy

Họ và tên: Trần Đăng Minh Tâm – MSV: 21020565

I. Tổng quan

1. Yêu cầu bài toán

- Đề bài:
  - Phân loại bọ trên bẫy vàng (Nội dung 1 - Chủ đề 3).
  - Phân vùng cải bó xôi (Nội dung 2 – Chủ đề 1).
- Mục tiêu đề ra:
  - Xây dựng mô hình phát hiện và phân loại các loại côn trùng (bọ) xuất hiện trên các tấm bẫy vàng được chụp ảnh.
  - Nhận diện được lá của cải bó xôi theo từng chậu.
- Nguồn dữ liệu:
  - Dataset yellow-sticky-traps-dataset (link dataset: https://github.com/md-121/yellow-sticky-traps-dataset/).
  - Subset 5 (link dataset: https://onedrive.live.com/?id=2BB72BC374F6A715%21sc7cca088d6044c93863bdda980871eb0&cid=2BB72BC374F6A715&sb=name&sd=1)
- Thống kê dữ liệu:
  - Data bẫy vàng:



Ảnh 1: Thống kê số lượng các label

Như có thể thấy trong biểu đồ, số lượng label cho whiteflies (WF) nhiều hơn vượt trội so với 2 nhãn còn lại là Macrolophus (MR) và Nesidiocoris (NC), với số lượng lần lượt là 5807, 1619 và 688 nhãn. WF chiếm áp đảo dữ liệu huấn luyện, điều này có thể gây ra hiện tượng mô hình sẽ dự đoán lệch hơn về class WF. Dữ liệu NC thì lại quá ít, có thể dễ bị bỏ qua hoặc nhầm với các lớp khác.

- Data cải bó xôi:

Gồm 180 ảnh chụp các chậu cải bó xôi từ trên xuống, qua nhiều giai đoạn phát triển, từ khi còn là cây con cho đến khi lớn. Cần được label thủ công.

II. Tiền xử lý dữ liệu

1. Bẫy vàng

- Kiểm tra định dạng, loại bỏ ảnh lỗi.
- Kiểm tra qua annotation cho trước xem label có lỗi gì không.
- Chia tập dữ liệu gồm ảnh và label theo format dùng cho YOLO, với tỉ lệ 70 train – 20 valid – 10 test.
- Tự động resize lại kích thước (nếu cần) cho đầu vào huấn luyện mô hình.
- Chuyển đổi nhãn từ YOLO sang định dạng (xmin, ymin, xmax, ymax) để phục vụ huấn luyện Faster R-CNN.

- Thêm augmentation.

2. Cải bó xôi

- Kiểm tra định dạng, loại bỏ ảnh lỗi.
- Gán nhãn dữ liệu sử dụng CVAT (link: cvat.ai). Vẽ đường biên của các lá sử dụng công cụ vẽ polygon, sau đó sử dụng groupshape nhóm các lá của cùng 1 chậu với nhau.
- Kiểm tra qua annotation cho trước xem label có lỗi gì không.
- Chuyển format label từ CVAT1.1 sang YOLO, gộp các lá có chung group_id vào thành 1 khối.
- Chia tập dữ liệu gồm ảnh và label theo format dùng cho YOLO, với tỉ lệ 70 train – 20 valid – 10 test.
- Tự động resize lại kích thước (nếu cần) cho đầu vào huấn luyện mô hình.
- Thêm augmentation.

III. Trích xuất đặc trưng

- Với YOLOv8: đặc trưng được trích xuất tự động thông qua backbone CSPDarknet
- Với Faster R-CNN: sử dụng ResNet-50 FPN backbone để trích xuất multi-scale features

IV. Xây dựng và mô hình

1. YOLOv8

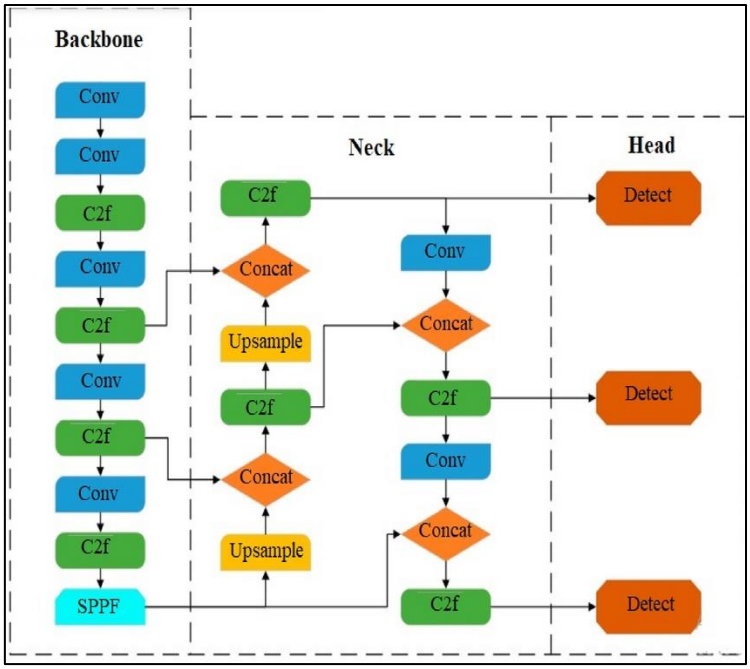
YOLOv8 là một mô hình nhận dạng đối tượng dựa trên mạng convolutional neural network (CNN) được phát triển bởi Joseph Redmon và nhóm nghiên cứu của ông tại Đại học Washington.

YOLOv8 là phiên bản nâng cấp của YOLOv7, với khả năng nhận diện đối tượng nhanh hơn và chính xác hơn. Điều này được đạt được thông qua một số cải tiến, bao

gồm mạng kim tự tháp đặc trưng, các mô-đun chú ý không gian và các kỹ thuật tăng cường dữ liệu tiên tiến.

Mô hình YOLOv8 sử dụng một mạng neural kiến trúc darknet-53 để trích xuất đặc trưng của hình ảnh và áp dụng thuật toán nhận dạng đối tượng YOLOv8 trên các đặc trưng đó.

Kiến trúc YOLO v8 chia thành ba thành phần chính: Backbone, Neck, và Head.



Hình 2: Kiến trúc Yolo V8

- **Backbone**: Đây là phần đầu của mạng, có nhiệm vụ trích xuất các đặc trưng (features) từ ảnh đầu vào.
    - Vai trò: Trích xuất đặc trưng từ ảnh đầu vào.
    - Cấu trúc: Gồm nhiều block C2f (Concatenate-to-Fusion), cải tiến từ C3 trong YOLOv5.
    - C2f giúp giảm tham số, tăng tốc độ suy luận, nhưng vẫn giữ được hiệu quả trích xuất đặc trưng.
- **Neck**: Phần này thực hiện chức năng tổng hợp các đặc trưng từ nhiều mức độ phân giải khác nhau để tăng cường khả năng nhận diện.
    - Vai trò: Kết hợp và khuếch đại đặc trưng từ nhiều tầng.

    - Sử dụng: FPN (Feature Pyramid Network) và PAN (Path Aggregation Network)
        - FPN: Truyền đặc trưng top-down.
        - PAN: Truyền đặc trưng bottom-up.
    - Giúp phát hiện tốt vật thể ở nhiều kích thước khác nhau.
- **Head**: Phần cuối cùng của mạng, có nhiệm vụ dự đoán vị trí và lớp của các vật thể trong ảnh.
    - Vai trò: Dự đoán output cuối cùng: bounding box, class, confidence.
    - Điểm mới: Anchor-Free – thay vì sử dụng anchor boxes thủ công, head dự đoán trực tiếp vị trí trung tâm và kích thước.

Trong bài tập này, phiên bản được sử dụng là YOLOv8n do kích thước nhẹ và tốc độ nhanh.

2. YOLOv11

YOLOv11 (You Only Look Once version 11) là phiên bản mới nhất trong dòng mô hình phát hiện vật thể nổi tiếng YOLO, được phát triển bởi Ultralytics. Được giới thiệu lần đầu vào năm 2024, YOLOv11 đánh dấu một bước đột phá quan trọng về tốc độ, độ chính xác, và khả năng tổng quát hóa trên nhiều tác vụ thị giác máy tính khác nhau.

So với các phiên bản trước, YOLOv11 không chỉ tập trung vào object detection, mà còn hỗ trợ multi-task learning bao gồm:

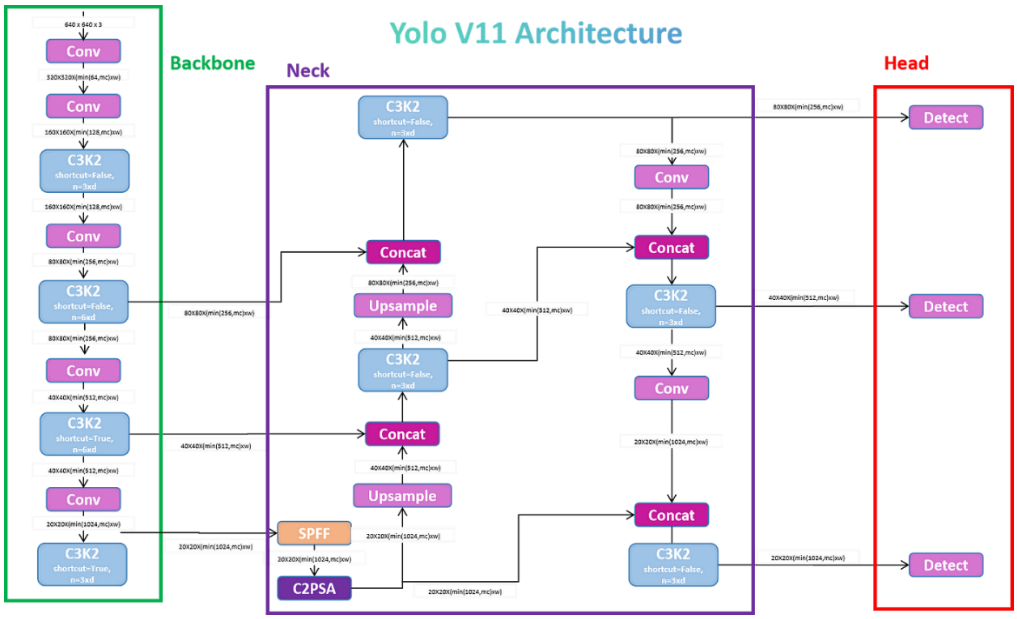
- Phân loại ảnh (classification)
- Phân đoạn ảnh (segmentation)
- Ước lượng tư thế (pose estimation)
- Phát hiện vật thể có hướng (oriented object detection)
- Phát hiện anomaly (anomaly detection)

Kiến trúc của YOLOv11 bao gồm ba thành phần chính: *Backbone, Neck* và *Task Heads*.

- **Backbone hiện đại (RT-DETR Inspired):**

- Kết hợp giữa ConvNeXt Blocks và Transformer Encoder, giúp mô hình học cả đặc trưng cục bộ (local) và ngữ cảnh toàn ảnh (global).

- **Neck: BiFPN cải tiến:**
  - Thay thế PANet bằng Bi-directional Feature Pyramid Network, giúp trộn thông tin từ nhiều tầng một cách hiệu quả và nhẹ hơn.
- **Modular Multi-task Heads** - Thiết kế đầu ra chia nhỏ theo từng tác vụ:
  - Detect Head: phát hiện bounding box + phân loại
  - Segment Head: phân đoạn mask
  - Pose Head: ước lượng tư thế người
  - Oriented Head: phát hiện vật thể có hướng
  - Anomaly Head: phát hiện bất thường



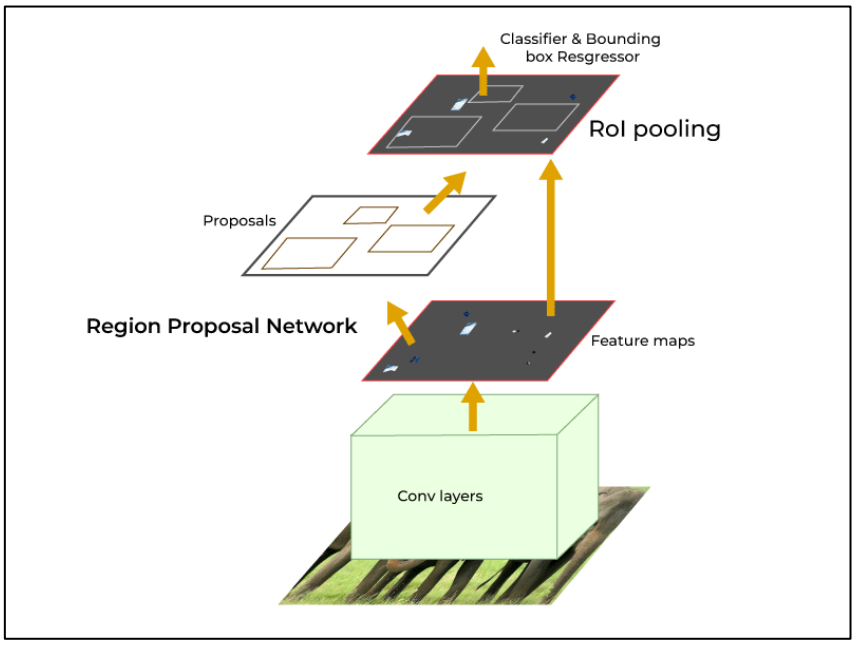
Hình 3: Kiến trúc YOLOv11

3. Faster R-CNN

Faster R-CNN là một mô hình two-stage object detector được giới thiệu vào năm 2015 bởi nhóm nghiên cứu của Shaoqing Ren, Kaiming He và Ross Girshick. Đây là phiên bản cải tiến của R-CNN và Fast R-CNN, nổi bật với:

- Tính chính xác cao.
- Khả năng phát hiện vật thể nhỏ tốt.

- Giải quyết triệt để bài toán Region Proposal bằng cách tích hợp vào mạng chính.



Hình 4: Kiến trúc mạng Faster R-CNN

Tổng quan các thành phần chính:

- **Backbone (CNN)** – trích xuất đặc trưng từ ảnh đầu vào (thường dùng ResNet, VGG...).
    - Dùng mạng CNN như ResNet50 hoặc VGG16.
    - Trích xuất bản đồ đặc trưng (feature map) từ ảnh.
- **Region Proposal Network (RPN)** – tạo ra các "vùng đề xuất" chứa vật thể.
    - Chạy trên feature map để sinh ra Anchor boxes.
    - Mỗi anchor được dự đoán: có phải vật thể không + tọa độ offset.
    - Giữ lại khoảng 2000 proposals tốt nhất → chuyển qua bước sau.
- **RoI Pooling** – chuyển đổi vùng đề xuất thành kích thước cố định.
    - Lấy feature tương ứng với mỗi proposal.
    - Dùng kỹ thuật max pooling để resize về kích thước cố định (thường 7×7).
    - Đảm bảo input cho fully-connected layers có cùng kích thước.
- **Classification & Regression Head** – phân loại và điều chỉnh bounding box.

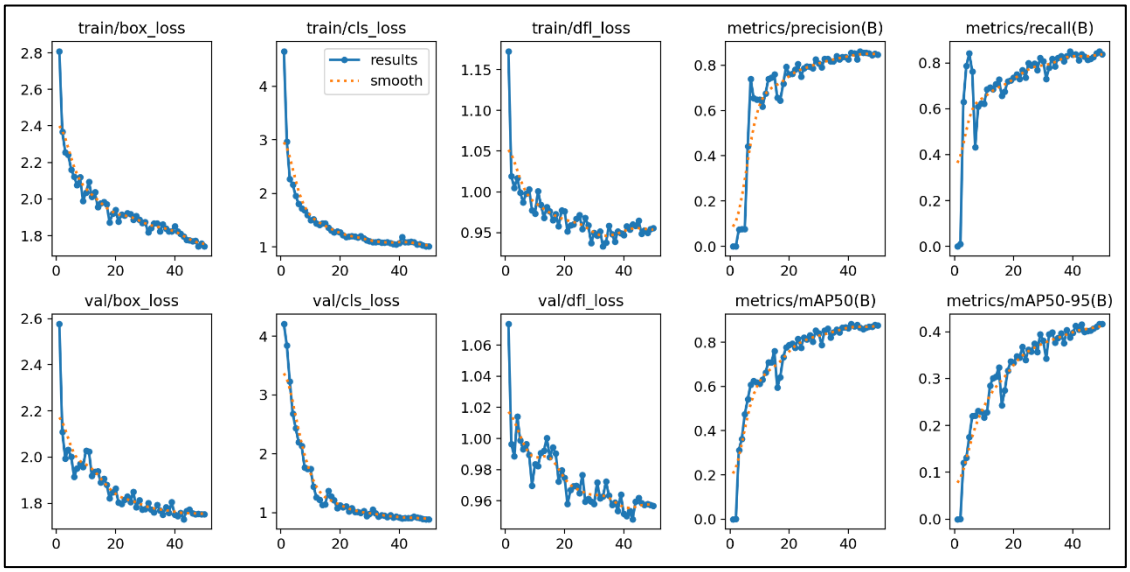
- Classifier: dự đoán nhãn cho mỗi RoI.
- Regressor: tinh chỉnh tọa độ bounding box.

V. Kết quả và đánh giá

1. Bẫy vàng

1.1. YOLOv8

Dưới đây là kết quả đánh giá quá trình huấn luyện sử dụng mô hình YOLOv8n với 50 epoch, batch size = 16:



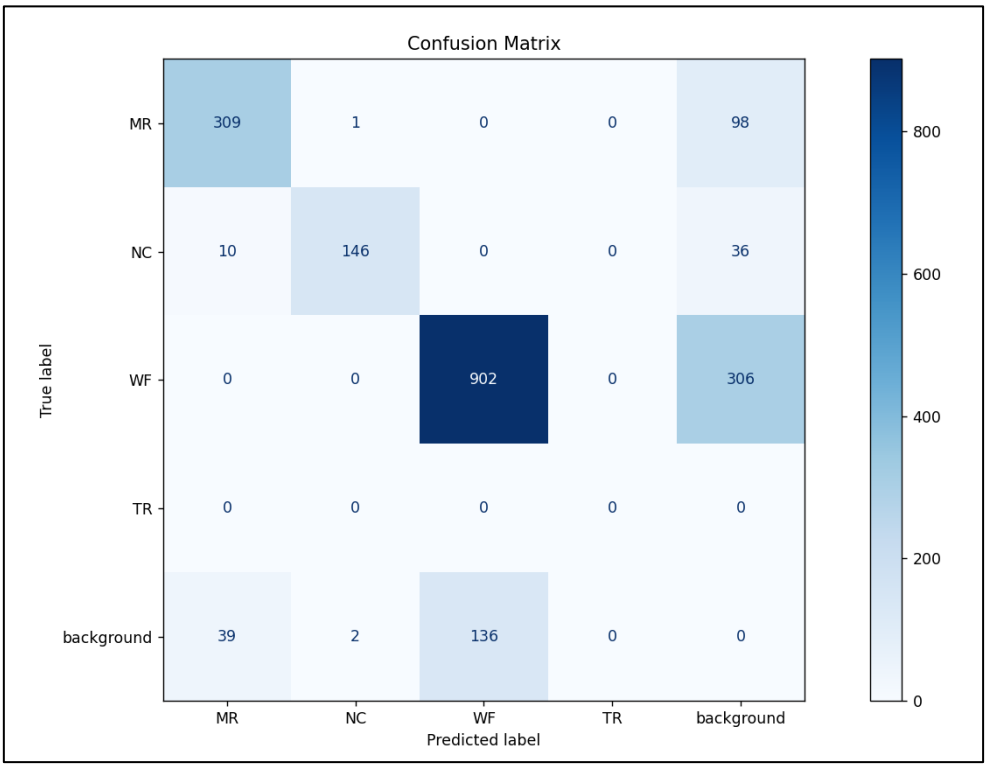
Hình 5: Biểu đồ đánh giá kết quả YOLOv8

Nhận xét:

- **Loss và độ chính xác huấn luyện:**
  - Loss giảm đều qua các epoch, không xảy ra hiện tượng overfitting.
  - Precision (B) và Recall (B) đạt:
    - Precision ≈ 0.85
    - Recall ≈ 0.84
  - mAP@0.5 đạt ≈ 0.85, trong khi mAP@0.5:0.95 đạt ≈ 0.42

Điều này cho thấy YOLOv8 học hiệu quả, đặc biệt tốt với các bounding box có độ chính xác cao (IoU ≥ 0.5), nhưng hiệu suất giảm khi yêu cầu chính xác định vị cao (IoU ≥ 0.95).

- **Ma trận nhầm lẫn:**

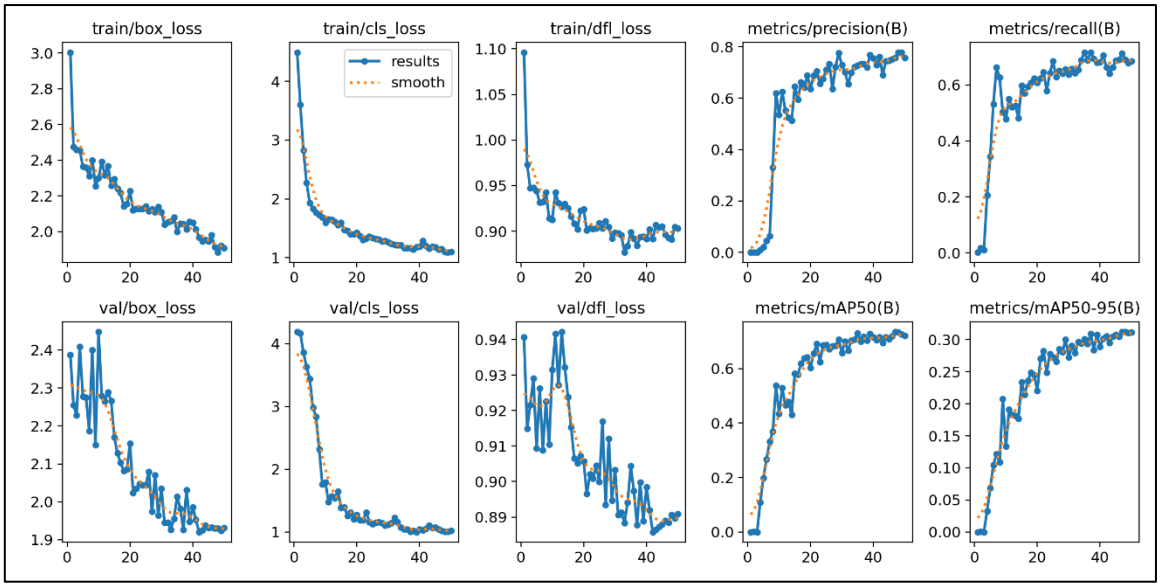


Hình 6: Ma trận nhầm lẫn

- **WF** có độ chính xác rất cao:
  - 902/1208 đúng (≈ 74.6%), nhưng vẫn bị bỏ sót khá nhiều (306 mẫu nhầm thành background). Vấn đề này xảy ra có thể do trên thực tế lớp này có kích thước rất nhỏ, khó nhìn thấy dù là bằng mắt thường.
- **MR** có 309 mẫu đúng / (309 + 98 missed) ≈ 75.9%, tương tự WF.
- **NC** được phân loại đúng 146/192 mẫu, tỷ lệ khá tốt (≈ 76%), chỉ bị bỏ sót nhẹ.

1.2. YOLOv11:

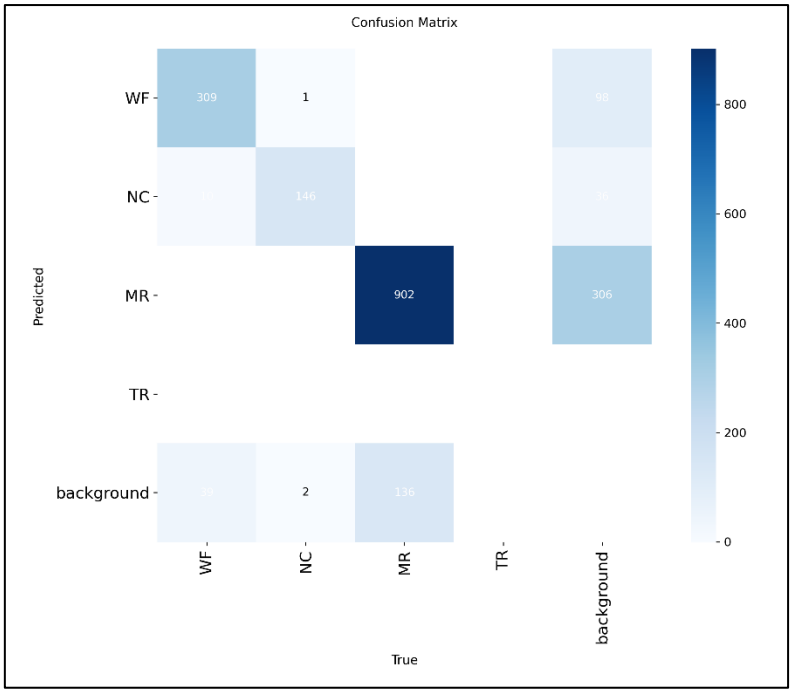
Dưới đây là kết quả đánh giá quá trình huấn luyện sử dụng mô hình YOLOv11n với 50 epoch, batch size = 16:



Hình 7: Biểu đồ đánh giá kết quả YOLOv11

Nhận xét:

- **Loss, độ chính xác và mAP:**
  - Các đường loss giảm đều và ổn định → huấn luyện thành công, không có dấu hiệu overfitting.
  - Precision cao hơn Recall: mô hình có xu hướng cẩn trọng hơn (ít dự đoán sai), nhưng có thể bỏ sót vật thể nhỏ hoặc khó phân biệt.
  - mAP@0.5 khá tốt, tuy nhiên mAP@0.5:0.95 còn thấp → cần cải thiện khả năng định vị chính xác hơn.



Hình 8: Ma trận nhầm lẫn

- **Ma trận nhầm lẫn:**
  - **Lớp WF (Whitefly)**:
    - 578 mẫu được dự đoán đúng.
    - 154 mẫu bị nhầm là background.

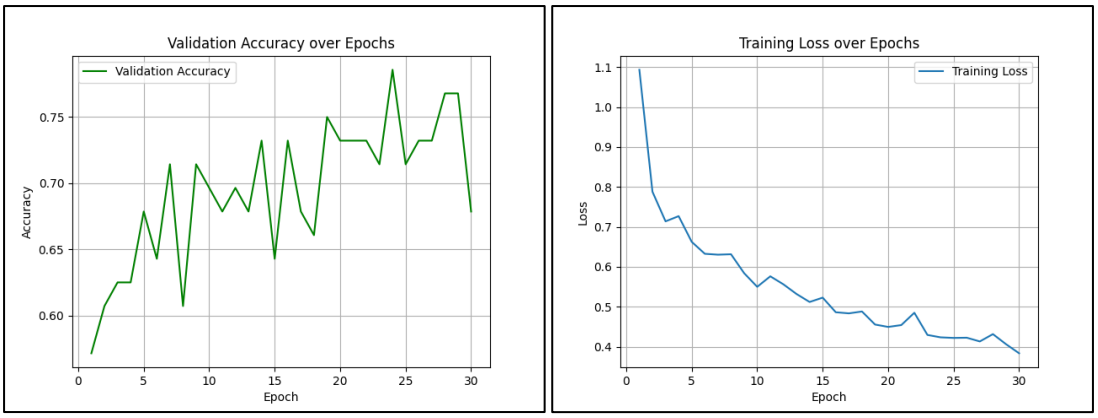
    → Nhận diện tốt nhưng tỷ lệ bỏ sót vẫn còn cao do kích thước nhỏ hoặc bị che khuất.
  - **Lớp MR (Myzus persicae)**:
    - 282 mẫu được dự đoán đúng.
    - 129 mẫu bị nhầm là background.
    - Một số mẫu bị nhầm lẫn sang lớp WF.

    → Cần cải thiện khả năng phân biệt giữa MR và các lớp khác.
  - **Lớp NC (Nesidiocoris tenuis)**:
    - 129 mẫu được phân loại chính xác.
    - 28 mẫu bị nhầm là background.
    - Có 10 mẫu bị nhầm thành MR.

    → Là lớp có tỷ lệ phân loại ổn định nhất.

1.3. Faster R-CNN

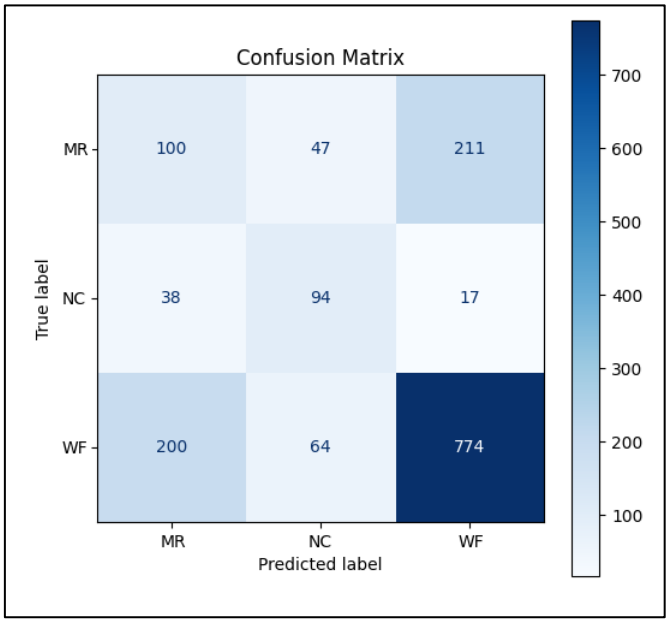
Biểu đồ dưới đây minh họa kết quả đánh giá quá trình huấn luyện mô hình trong 30 epoch, batch size = 4, learning rate = 0.005 trên tập validate:



Hình 9: Kết quả huấn luyện

Nhận xét:

- Loss giảm đều từ ~1.1 về dưới 0.4 cho thấy mô hình đã học ổn định và không bị overfitting.
- Accuracy dao động tăng theo xu hướng chung, đạt ~ 0.78 ở những epoch cao, cho thấy khả năng phân loại vật thể khá tốt.



Hình 10: Ma trận nhầm lẫn

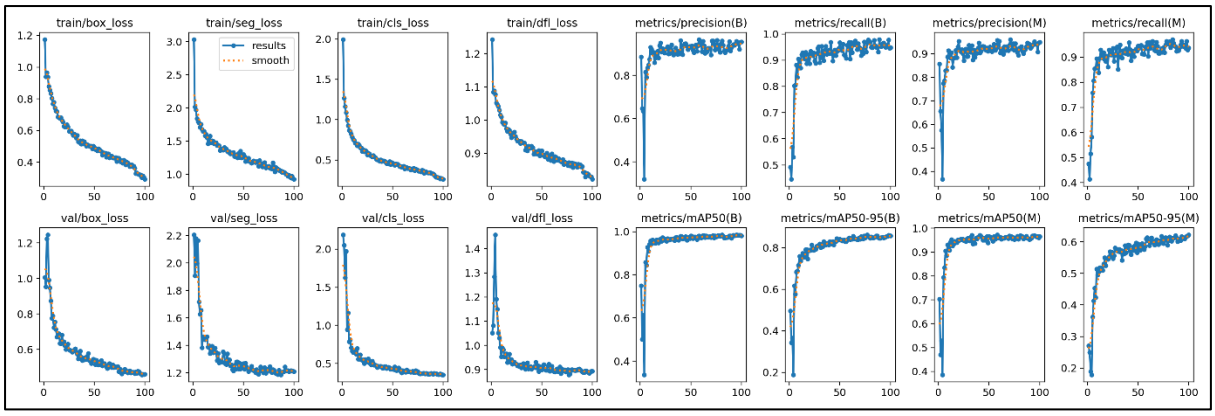
- WF (Thật) được dự đoán đúng tới 774/1038 mẫu, chiếm khoảng 74.6%, cho thấy mô hình hoạt động tốt với lớp côn trùng chính.
- MR (Thật) bị nhầm lẫn nhiều thành WF (211 lần), và ngược lại WF cũng nhầm thành MR 200 lần. Điều này cho thấy mô hình gặp khó khăn khi phân biệt hai lớp có hình dạng nhỏ hoặc dễ nhầm.
- NC (Thật) được phân biệt khá rõ, với 94/149 mẫu đúng, và ít bị nhầm thành các lớp còn lại.

1.4. So sánh các mô hình đã sử dụng

- **YOLOv8:**
  - One-stage detector, anchor-free → nhanh và gọn nhẹ.
  - Precision ~ **0.85**, Recall ~ **0.84**, mAP@0.5 ~ **0.85**, mAP@0.5:0.95 ~ **0.42**.
  - Phát hiện tốt các vật thể nhỏ, chính xác cao.
  - Có nhầm nhẹ với background, nhưng ít hơn YOLOv11.
  - Huấn luyện ổn định, inference nhanh → phù hợp ứng dụng thực tế.
- **Faster R-CNN:**
  - Two-stage detector (RPN → RoI Head), anchor-based.
  - Precision ~ **0.78**, Recall ~ **0.76**, mAP@0.5 ~ **0.80**, mAP@0.5:0.95 ~ **0.39**.
  - Phát hiện tốt các vật thể phức tạp hoặc nhỏ.
  - Thời gian suy luận chậm, không phù hợp ứng dụng realtime.
  - Ít nhầm lẫn với nền, nhưng yêu cầu tài nguyên tính toán cao.
- **YOLOv11**
  - Phiên bản cải tiến từ YOLO, one-stage detector.
  - Precision ~ **0.78**, Recall ~ **0.72**, mAP@0.5 ~ **0.80**, mAP@0.5:0.95 ~ **0.30**.
  - Huấn luyện ổn định, tốc độ tốt.
  - Nhầm lẫn với **background** nhiều (đặc biệt lớp WF, MR).
  - Cần cải thiện hậu xử lý, confidence threshold, hoặc dữ liệu huấn luyện.

2. Cải bó xôi

Dưới đây là kết quả đánh giá quá trình huấn luyện phân vùng cải bó xôi sử dụng mô hình YOLOv8-segment:

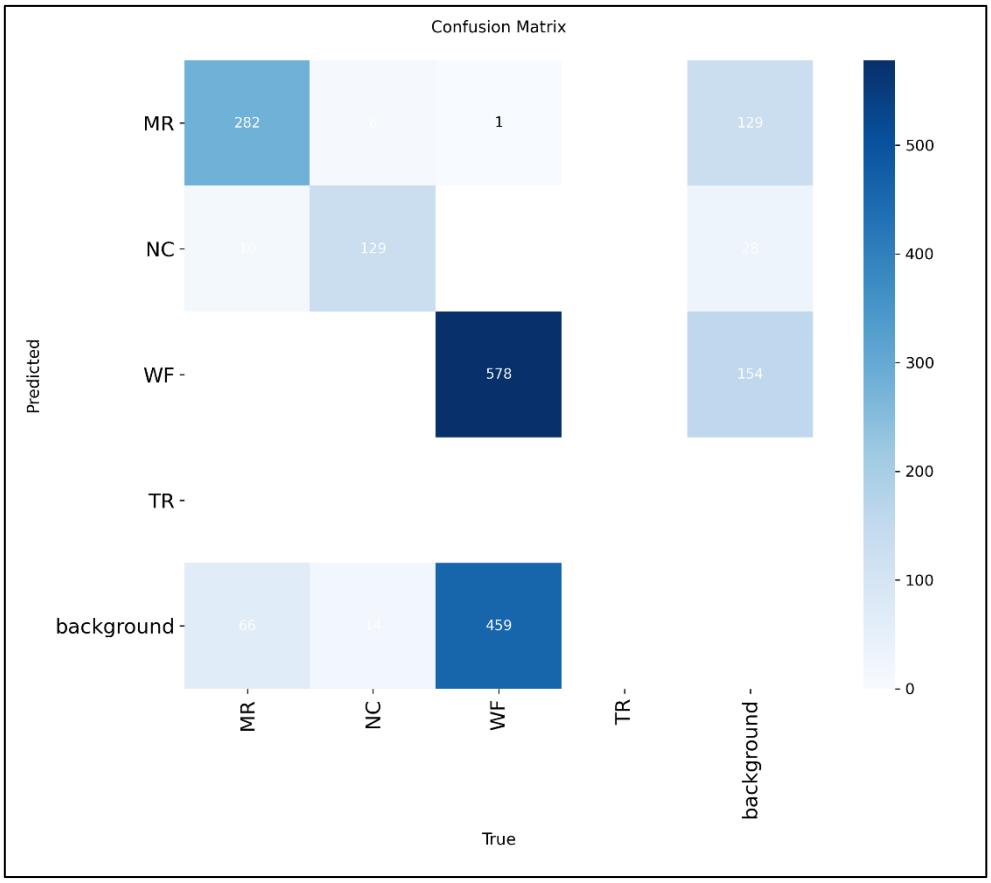


Hình 11: Kết quả segment bằng YOLOv8

Nhận xét:

- **Biểu đồ huấn luyện:**
  - Precision (B): ~0.84
  - Recall (B): ~0.82
  - mAP@0.5 (B): ~0.85
  - mAP@0.5:0.95 (B): ~0.43
  - Loss (box/cls/dfl): giảm đều, ổn định → mô hình học hiệu quả và không bị overfitting
  - Mô hình hội tụ tốt trong ~50 epoch.
  - Cả train và validation loss đều giảm đều → dữ liệu phân bố ổn.
  - mAP cao cho thấy mô hình định vị và phân lớp vật thể hiệu quả.

- **Ma trận nhầm lẫn:**



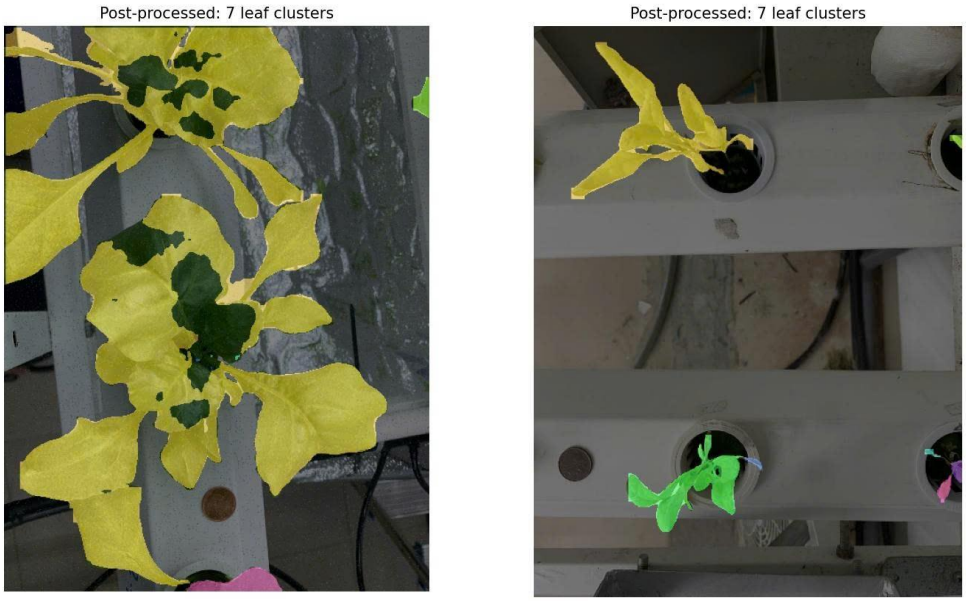
Hình 12: Ma trận nhầm lẫn

  - **Lớp MR:**
    - Dự đoán đúng 902 mẫu.
    - Bỏ sót 306 mẫu (nhầm thành background).

→ Đây là lớp có tỷ lệ chính xác cao nhưng vẫn bị bỏ sót tương đối nhiều.

  - **Lớp NC:**
    - Dự đoán đúng 146 mẫu.
    - Bỏ sót 36 mẫu, có 10 mẫu nhầm sang WF.
    - → Nhận diện ổn định, ít nhầm với các lớp khác.
  - **Lớp WF:**
    - Dự đoán đúng 309 mẫu.
    - Nhầm 98 mẫu thành background.
    - Có 1 mẫu nhầm thành NC → sai sót rất nhỏ, không đáng kể.

Với nội dung này, sau khi kiểm thử mô hình trên tập test, thu được mội vài trường hợp như sau:



Hình 13: Các trường hợp còn chưa tốt

Ở các trường hợp này, mô hình phân vùng chưa hoạt động tốt, các ảnh có các chậu cây lớn lá chồng lên nhau sẽ bị gộp chung vào 1 cụm, còn những lá quá nhỏ hoặc lá nhỏ và cách thưa nhau thì dù chung 1 chậu vẫn bị tính là các cụm khác nhau.

VI. Kết luận

Trong đề tài nhận diện và phân loại côn trùng trên bẫy vàng (bao gồm cả phân vùng cải bó xôi), em đã tiến hành huấn luyện và đánh giá ba mô hình khác nhau: YOLOv8, Faster R-CNN, và YOLOv11.

Các mô hình đều cho thấy khả năng học ổn định và hiệu quả nhất định trong việc nhận diện các lớp côn trùng chính (MR, WF, NC), trong đó:

- YOLOv8 đạt hiệu năng tốt nhất toàn diện: độ chính xác cao, tốc độ nhanh, dễ triển khai.
- Faster R-CNN phù hợp cho các bài toán học thuật, yêu cầu độ chính xác cao, nhưng tốc độ chậm.
- YOLOv11 thể hiện tiềm năng, nhưng vẫn cần cải thiện để giảm nhầm lẫn và nhận diện đầy đủ tất cả các lớp.

Với bài toán phân vùng cải bó xôi, YOLOv8 tỏ ra phù hợp với bài toán, nhưng vẫn cần phải cải thiện.

Kết quả thu được cho thấy các mô hình hiện đại như YOLOv8 hoàn toàn khả thi trong việc ứng dụng thực tế để tự động hóa quá trình giám sát sâu bệnh, giúp giảm sức lao động và tăng hiệu quả quản lý trong nông nghiệp chính xác.

Hướng phát triển tương lai:

- **Cải thiện dữ liệu:**
  - Bổ sung dữ liệu cho lớp TR: hiện tại lớp TR không được mô hình phát hiện → cần:
  - Tăng số lượng ảnh chứa TR.
  - Làm sạch và xác minh annotation để tránh thiếu nhãn.
  - Cân bằng lại tập huấn luyện:
  - Giảm mất cân bằng giữa các lớp (MR chiếm áp đảo).
  - Sử dụng kỹ thuật oversampling/undersampling hoặc data augmentation theo lớp.
- **Tối ưu mô hình:**
  - **Fine-tune threshold & post-processing**:
    - Tăng ngưỡng confidence để giảm false positive với background.
    - Áp dụng Non-Maximum Suppression (NMS) có trọng số để cải thiện phân vùng chồng lấp.
  - **Thử nghiệm thêm mô hình nhẹ hơn**:
    - YOLOv8n, YOLOv7-tiny hoặc EfficientDet → tối ưu cho thiết bị biên, drone giám sát.
  - **Với bài toán phân vùng cải bó xôi:**
    - Có thể huấn luyện sử dụng thêm thông tin về bounding box giới hạn về vị trí chậu để có thể tách biệt các cụm lá to hoặc gộp các cụm lá nhỏ.