



ĐẠI HỌC CÔNG NGHỆ

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

-----800000-----

Khoa: Điện Tử - Điện Viễn Thông

Ngành: Kỹ Thuật Robot

-----o0o-----

BÁO CÁO HỌC PHẦN HỌC MÁY

Đề Tài

PHÂN LOẠI BỘ TRÊN BẦY VÀNG

Người thực hiện:

Vũ Văn Lộc – QH-2021-I/CQ-R – 21020928

GIẢNG VIÊN HƯỚNG DẪN: PGS.TS Phạm Minh Triển

HÀ NỘI - 2025

Contents

I. Tổng quan về đề tài phân loại côn trùng trên bẫy vàng	2
II. Xây dựng, phân tích và mô tả bộ dữ liệu	3
2.1 Gán nhãn bộ dữ liệu	3
2.2 Khám phá dữ liệu.....	3
2.3 Phân bố lớp	4
III. Tiền xử lý dữ liệu	5
3.1 Tiền xử lý (Preprocessing)	5
3.2 Tăng cường dữ liệu (Augmentations).....	5
3.3 Phân bố bộ dữ liệu	7
3.4 Mục tiêu của tiền xử lý	7
IV. Train mô hình.....	7
V. Kết quả và đánh giá.....	8
5.1 Yolov8n	8
5.2 yolov11n	12
5.3 Faster R-CNN với ResNet-50 FPN	13
VI. Kết luận và hướng phát triển.....	15
6.1 Kết luận.....	15
6.2 Hướng phát triển	15
1. Tối ưu hóa mô hình cho môi trường triển khai thực tế	15
2. Nâng cao độ chính xác thông qua cải tiến dữ liệu và mô hình.....	16
3. Kết hợp mô hình (Ensemble hoặc Two-Stage Detection).....	16
4. Tích hợp hệ thống và đánh giá toàn diện	16
5. Hướng nghiên cứu mở rộng	17
Tổng kết.....	17

I. Tổng quan về đề tài phân loại côn trùng trên bẫy vàng

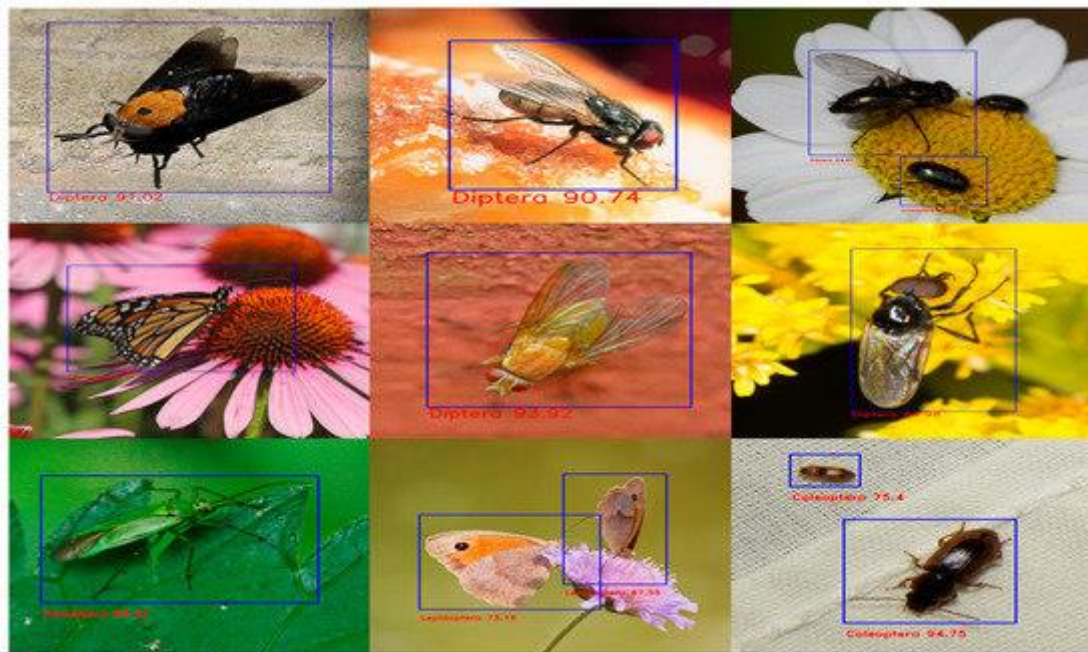
Trong nông nghiệp hiện đại, việc theo dõi và kiểm soát dịch hại là một yếu tố then chốt nhằm đảm bảo năng suất và chất lượng cây trồng. Một trong những phương pháp phổ biến để giám sát côn trùng gây hại là sử dụng bẫy vàng dính (yellow sticky traps) – công cụ đơn giản nhưng hiệu quả giúp thu hút và bắt giữ các loài côn trùng bay.



Hình 1. Bẫy vàng ở cây trồng

Tuy nhiên, việc phân loại và đếm thủ công số lượng côn trùng trên các bẫy dính không chỉ tốn thời gian, công sức mà còn dễ xảy ra sai sót. Với sự phát triển nhanh chóng của trí tuệ nhân tạo (AI) và học máy (Machine Learning), việc tự động hóa quy trình nhận diện,

phân loại và thống kê côn trùng từ ảnh chụp bẫy vàng đang trở thành một hướng tiếp cận hiệu quả, thông minh và tiết kiệm chi phí.



Hình 1. Kết quả phát hiện các loại côn trùng khác nhau bằng yolovx

II. Xây dựng, phân tích và mô tả bộ dữ liệu

2.1 Gán nhãn bộ dữ liệu

Bộ dữ liệu được xây dựng từ các ảnh chụp thực tế những tấm bẫy vàng dính côn trùng trong môi trường nhà kính. Sau khi thu thập, dữ liệu được đưa lên nền tảng Roboflow để tiến hành gán nhãn thủ công. Mỗi cá thể côn trùng trên ảnh được gán nhãn dưới dạng bounding box với thông tin về loài côn trùng tương ứng. Các nhãn trong bộ dữ liệu bao gồm:

- *Macrolophus pygmaeus*: Là loài bọ có lợi (MR)
- *Nesidiocoris tenuis*: Thiên địch nhưng đôi khi gây hại (NC)
- *Trialeurodes vaporariorum*: Bọ phấn trắng – loài gây hại phổ biến (WF)

Mỗi nhãn đại diện cho một loài côn trùng thường gặp trong nhà kính và có ảnh hưởng khác nhau đến cây trồng. Việc gán nhãn chính xác là bước rất quan trọng để mô hình học máy có thể học được đặc điểm hình thái của từng loài

2.2 Khám phá dữ liệu

- Tổng số ảnh là 284 ảnh
- Kích thước ảnh gốc: Đồng nhất
- Định dạng nhãn: Pascal VOC (có thể chuyển đổi sang YOLO hoặc COCO dễ dàng)

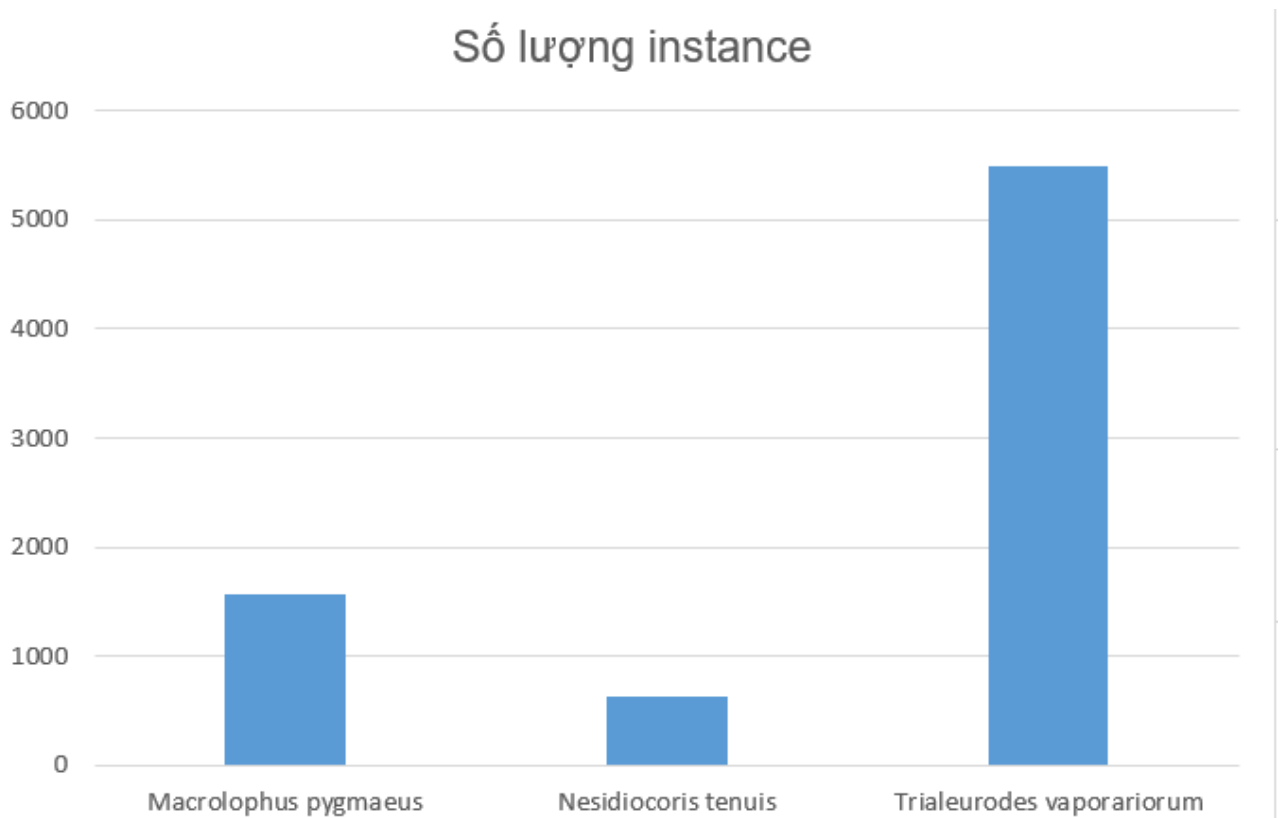
- Kiểu dữ liệu: ảnh màu định dạng JPG hoặc PNG, kết hợp với file XML hoặc TXT chứa thông tin nhãn.
- Ảnh được chụp trong điều kiện ánh sáng tự nhiên.
- Bầy vàng thường có nền vàng đậm, nhiều côn trùng chồng lấn.
- Độ phân giải cao giúp dễ dàng phát hiện vật thể nhỏ.

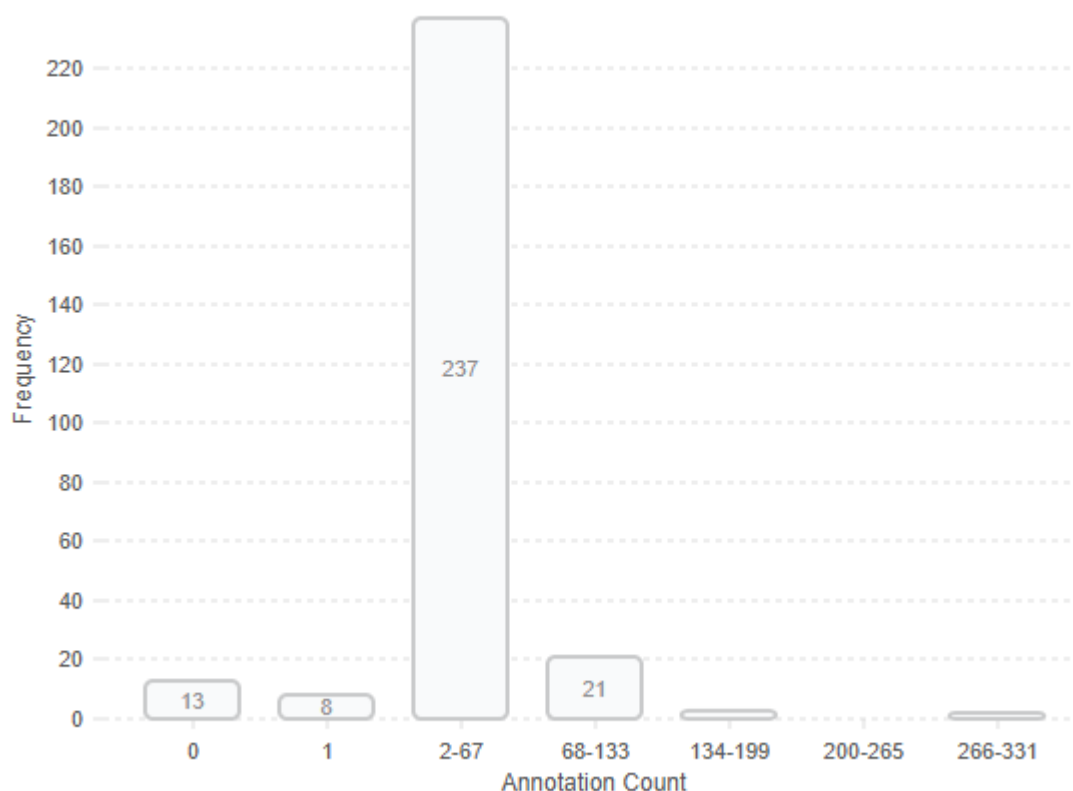
2.3 Phân bố lớp

Phân bố số lượng đối tượng theo từng loài như sau:

Loài côn trùng	Số lượng instance
<i>Macrolophus pygmaeus</i>	1561
<i>Nesidiocoris tenuis</i>	638
<i>Trialeurodes vaporariorum</i>	5487

Biểu đồ phân bố của các label





Biểu đồ phân bố số lượng box trên các ảnh

III. Tiền xử lý dữ liệu

Sau khi xây dựng và gán nhãn bộ dữ liệu côn trùng bầy vàng, bước tiếp theo là tiền xử lý dữ liệu để chuẩn bị đầu vào chất lượng cho các mô hình học máy. Việc tiền xử lý này được thực hiện trực tiếp trên nền tảng Roboflow với các bước cụ thể như sau:

3.1 Tiền xử lý (Preprocessing)

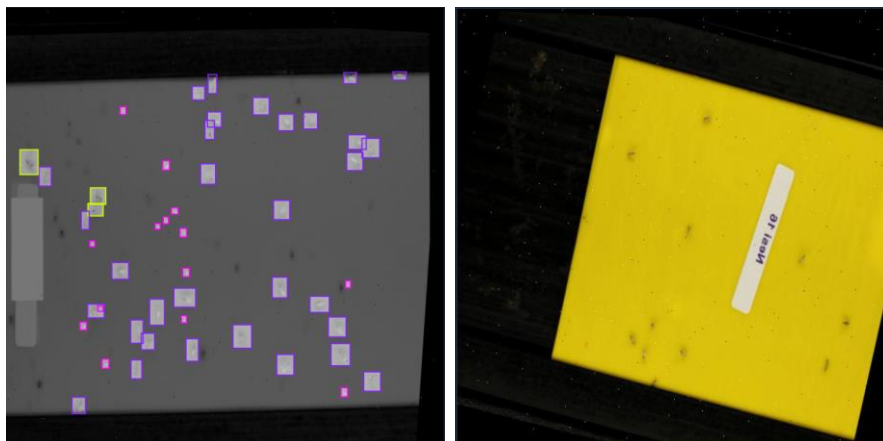
Auto-Orient: Applied - Các ảnh được tự động xoay đúng hướng, đảm bảo thống nhất về góc nhìn ban đầu, tránh tình trạng ảnh bị lệch do camera chụp sai chiều.

Resize: Stretch to 640x640 pixels - Tất cả ảnh trong tập dữ liệu được co giãn tỷ lệ và điều chỉnh kích thước về đúng khung 640x640 pixel, đảm bảo đồng nhất kích thước đầu vào cho mô hình YOLO và các mô hình CNN khác.

3.2 Tăng cường dữ liệu (Augmentations)

Để cải thiện khả năng khái quát hóa của mô hình và tăng độ đa dạng cho dữ liệu huấn luyện, mỗi ảnh trong tập huấn luyện được tự động tạo thêm 3 biến thể với các kỹ thuật tăng cường như sau:

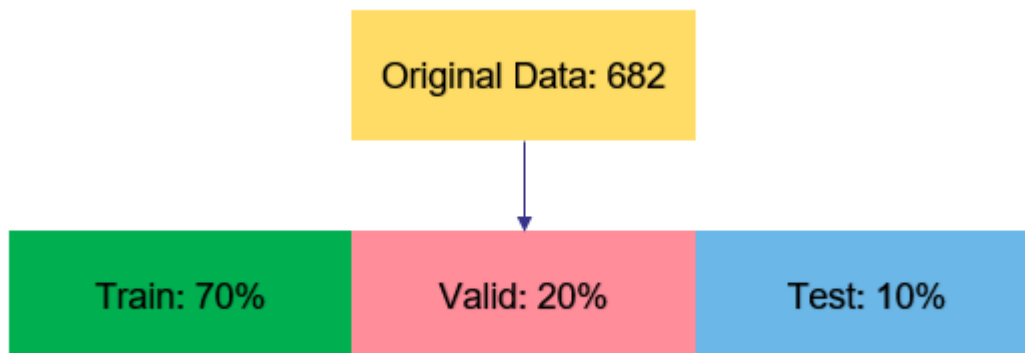
- Lật ảnh (Flip): Lật ngang và dọc giúp mô hình nhận diện côn trùng ở các góc đối xứng.
- Xoay 90 độ (90° Rotate): Bao gồm xoay theo chiều kim đồng hồ, ngược chiều kim đồng hồ và lật ngược, giúp mô hình học được các hướng phổ biến của côn trùng trên bầy.
- Zoom ảnh (Crop): Phóng to ngẫu nhiên từ 0% đến 20% giúp mô hình làm quen với các tỷ lệ kích thước khác nhau của vật thể.
- Xoay tự do (Rotation): Góc xoay từ -15° đến $+15^\circ$, tăng khả năng nhận diện khi ảnh bị lệch nhẹ.
- Nghiêng ảnh (Shear): Góc shear ngang và dọc $\pm 10^\circ$, mô phỏng các biến dạng hình học thực tế.
- Chuyển sang ảnh xám (Grayscale): Áp dụng cho 15% ảnh, giúp mô hình học được đặc trưng kết cấu thay vì chỉ phụ thuộc vào màu sắc.
- Biến đổi màu sắc:
 - Hue: Thay đổi tông màu từ -15° đến $+15^\circ$.
 - Saturation: Giảm hoặc tăng độ bão hòa từ -25% đến +25%.
 - Brightness: Điều chỉnh độ sáng từ -15% đến +15%.
 - Exposure: Thay đổi mức phơi sáng từ -10% đến +10%.
- Làm mờ (Blur): Làm mờ ảnh với độ mờ tối đa là 2.5px để mô phỏng tình huống ảnh mất nét.
- Thêm nhiễu (Noise): Chèn nhiễu ngẫu nhiên lên đến 0.1% tổng số pixel, giúp mô hình chống chịu tốt hơn trong môi trường thực.



Xoay và tăng cường ảnh

3.3 Phân bố dữ liệu

Sau khi tăng cường ảnh thì từ 284 ảnh thành → 682 ảnh và được phân bố như sau :



Tập dữ liệu	Số ảnh	Tỷ lệ (%)
Train	478	70%
Validation	136	20%
Test	68	10%

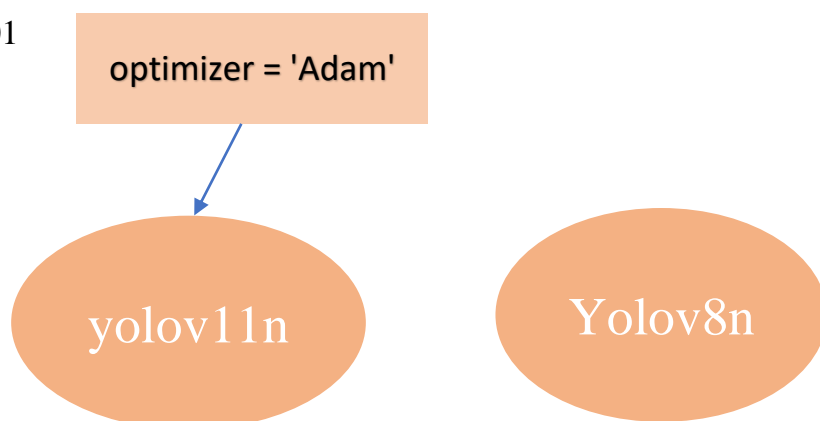
Việc chia dữ liệu được thực hiện ngẫu nhiên nhưng đảm bảo mỗi lớp xuất hiện tương đối đồng đều trong cả ba tập.

3.4 Mục tiêu của tiền xử lý

- Tăng độ đa dạng mẫu huấn luyện, giúp mô hình học được các biến thể khác nhau của côn trùng.
- Cải thiện tính tổng quát (generalization) để mô hình hoạt động tốt hơn khi gặp ảnh mới.
- Giảm overfitting do mô hình không còn học thuộc từng ảnh cụ thể mà học được khái quát về đặc trưng.

IV. Train mô hình

- Learning rate = 0.01
- Batch = 16
- Epochs = 1000

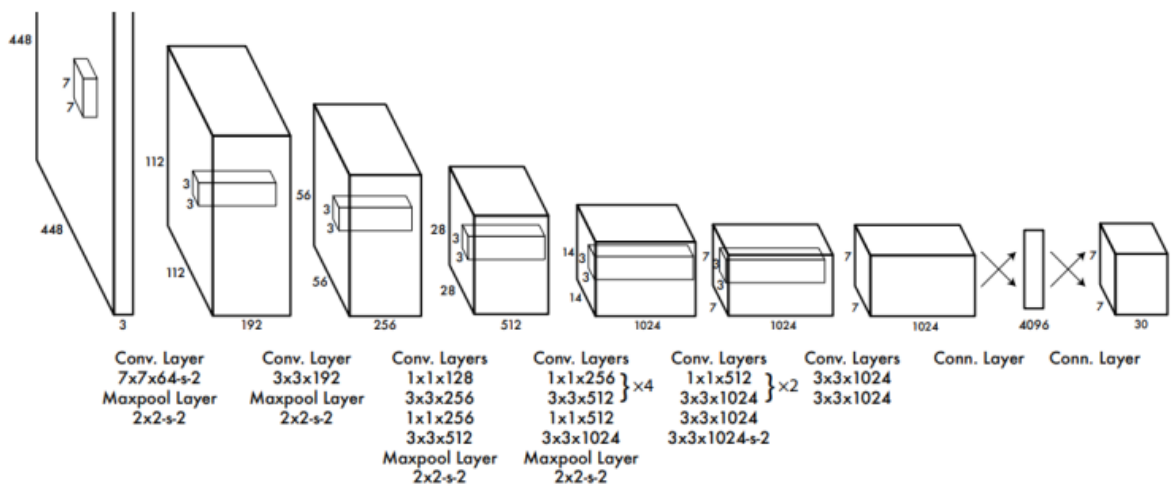


Kết quả hội tụ ở epochs và thời gian dưới bảng sau:

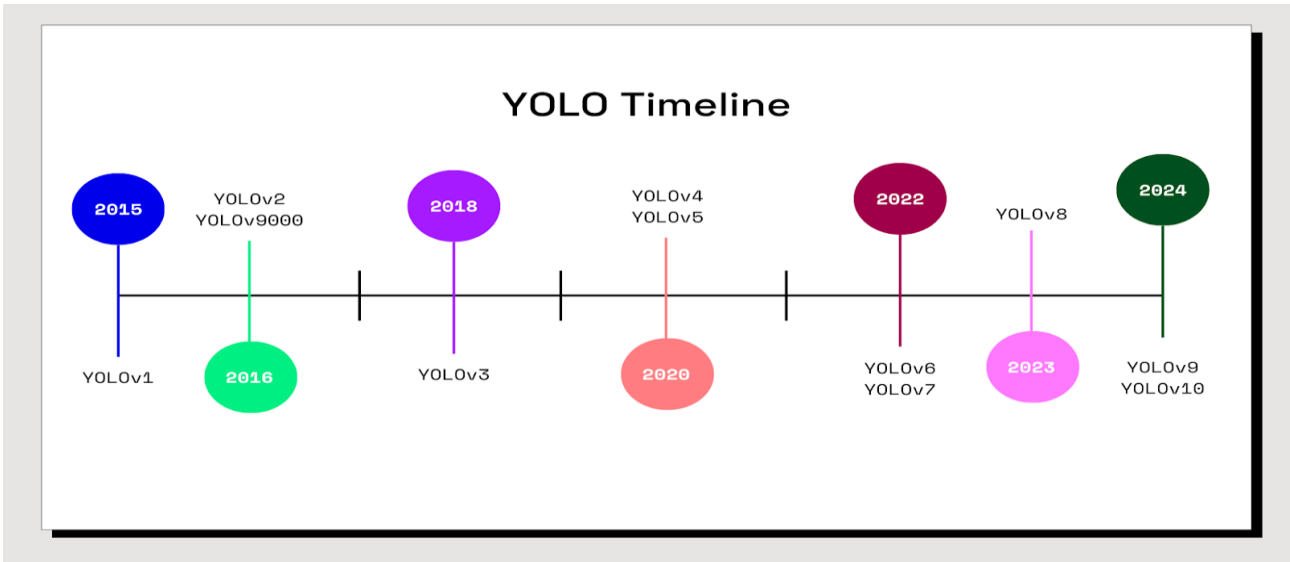
model	Epoch	time
yolov8n	387	0.8722 (h)
yolov11n	349	1.687 (h)

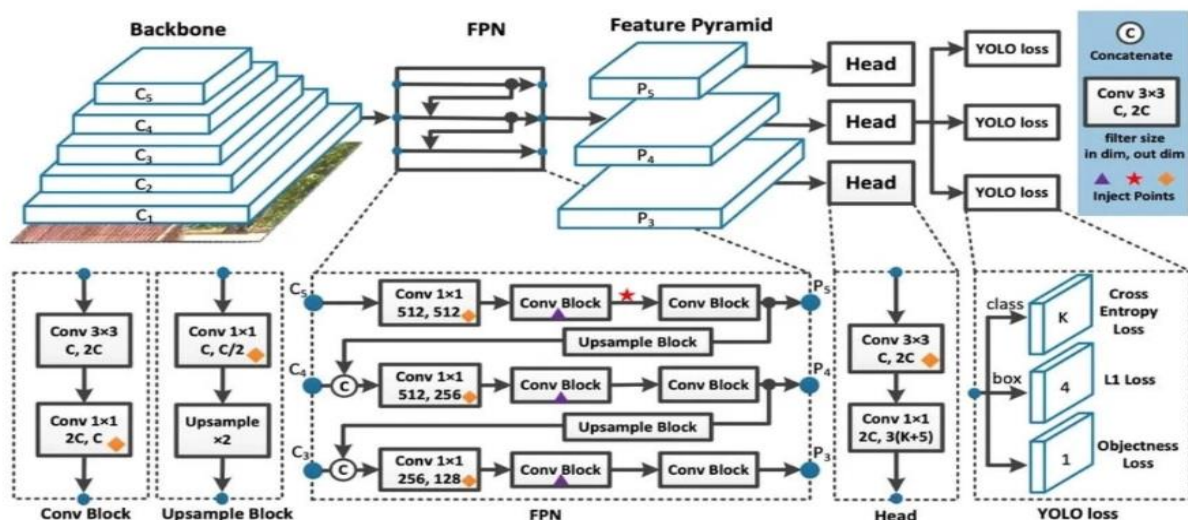
V. Kết quả và đánh giá

5.1 Yolov8n

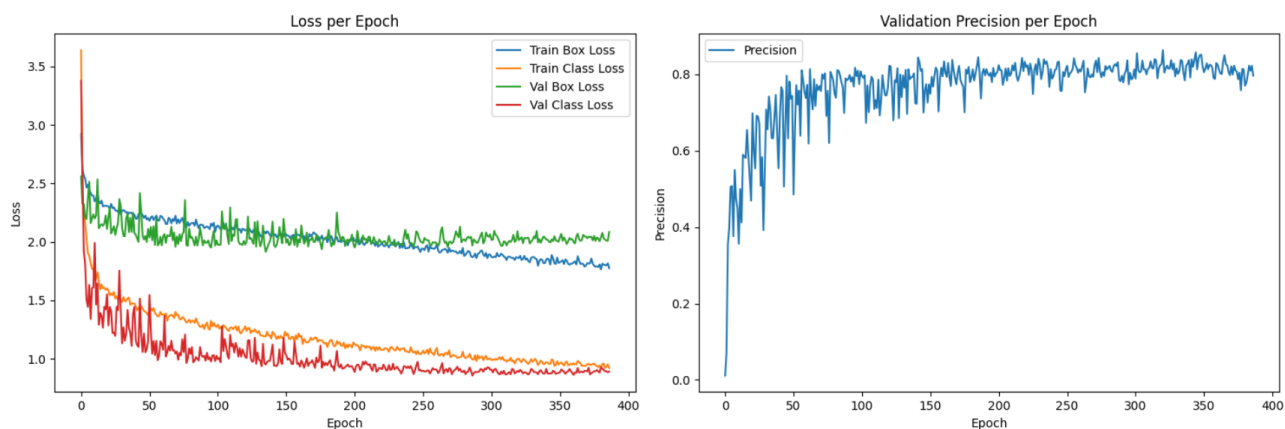


Cấu trúc tích chập của yolo





Cấu trúc của yolov8n



Train Box Loss & Train Class Loss: giảm đều qua các epoch, đặc biệt là trong 100 epoch đầu, chứng tỏ mô hình học tốt từ dữ liệu huấn luyện.

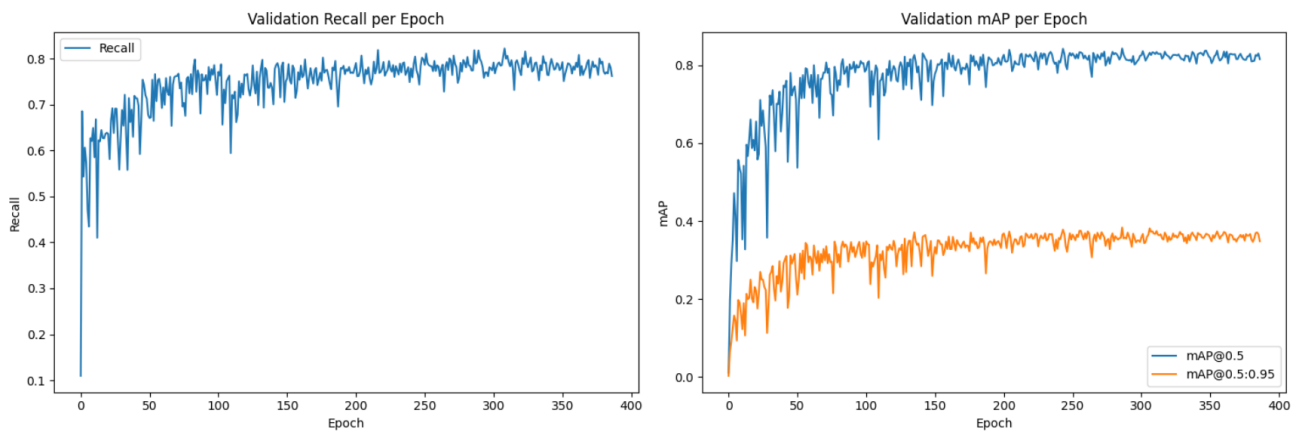
Val Box Loss & Val Class Loss: cũng giảm dần và ổn định, không có dấu hiệu overfitting (loss validation không tăng).

➔ Nhận xét: Mô hình học ổn định và không bị overfitting, dữ liệu huấn luyện và validation khá đồng nhất.

Precision tăng nhanh trong 50 epoch đầu và dần tiến tới ngưỡng ~0.82.

Dao động nhẹ, nhưng nhìn chung ổn định từ epoch ~100 trở đi.

➔ Nhận xét: Mô hình rất chính xác trong việc phát hiện đúng đối tượng (ít phát hiện nhầm).



Recall bắt đầu thấp nhưng tăng mạnh, đạt ~ 0.80 từ khoảng epoch 150 trở đi.

➔ Nhận xét: Mô hình bắt đầu phát hiện được gần hết các đối tượng mục tiêu, tức là ít bỏ sót.

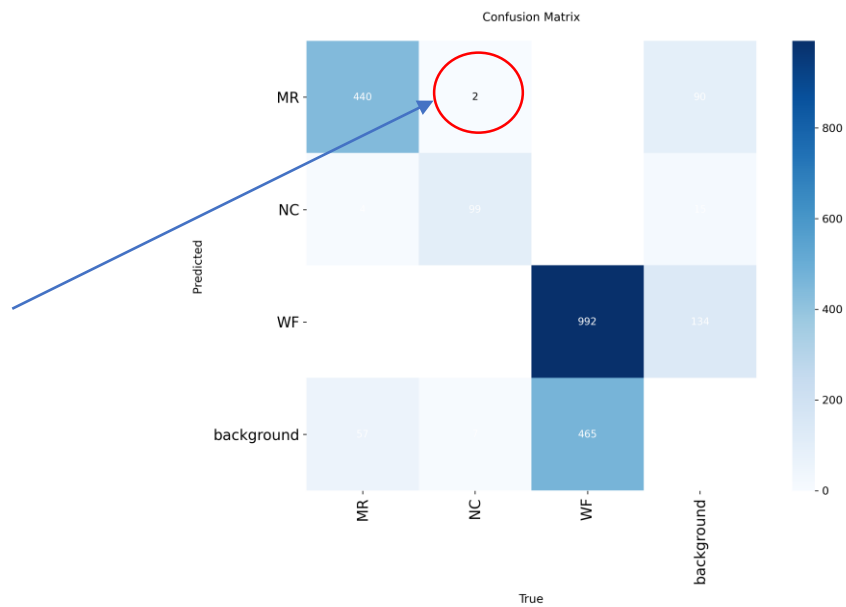
mAP@0.5: đạt ~ 0.85 , rất tốt.

mAP@0.5:0.95: dao động quanh 0.33, hợp lý với YOLOv8 dùng default anchor-based detection.

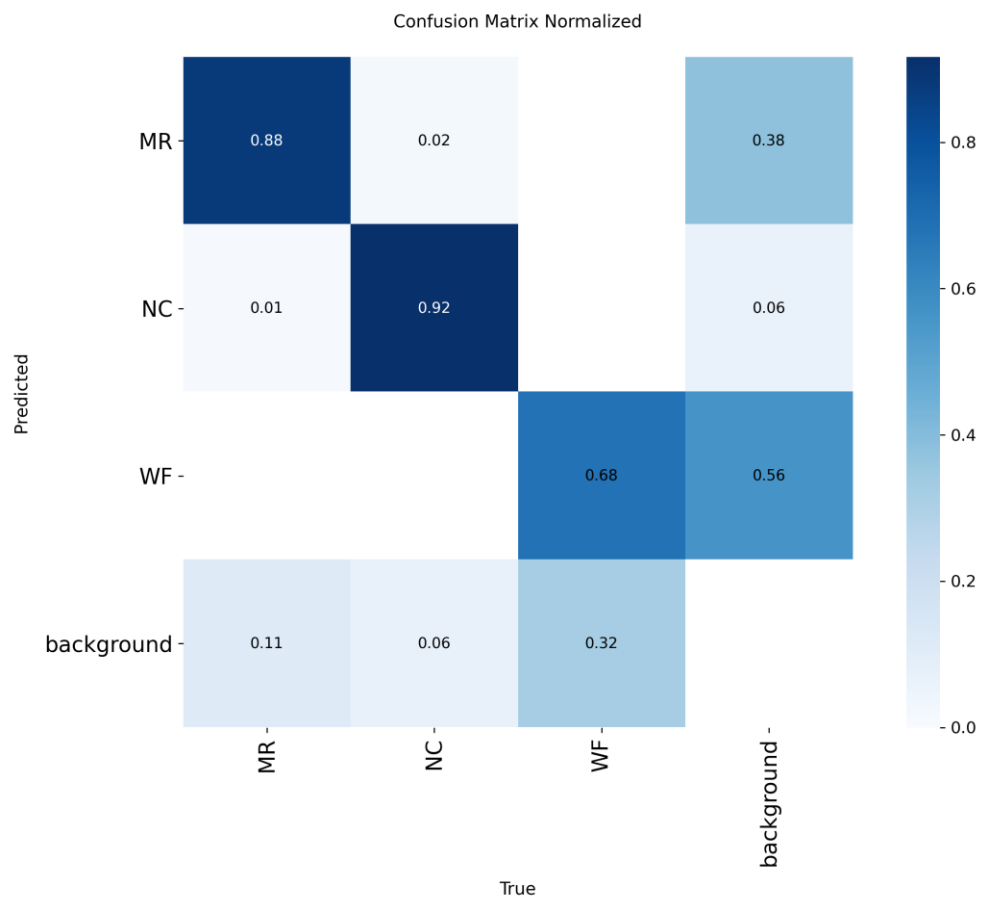
➔ Nhận xét:

mAP@0.5 cao \rightarrow mô hình có khả năng định vị tốt đối tượng.

mAP@0.5:0.95 tương đối \rightarrow mô hình chưa hoàn hảo về localization ở mức IoU cao (có thể do kích thước nhỏ/phức tạp của đối tượng bỏ trong ảnh bẫy vàng).



Ma trận nhầm lẫn



Chuẩn hóa lại

5.2 yolov11n

YOLOv11

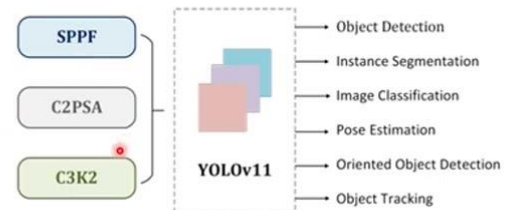
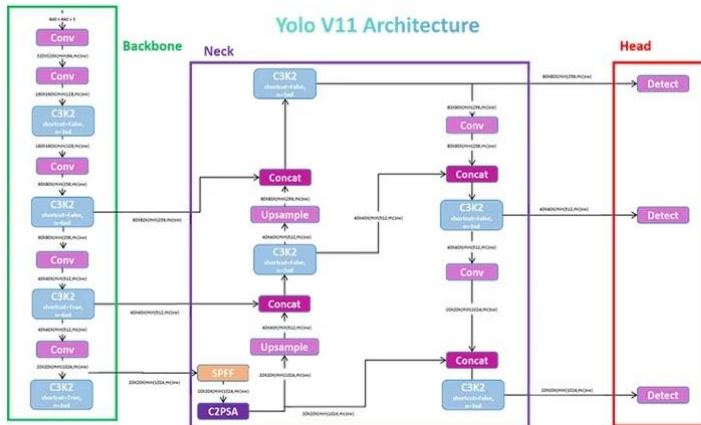
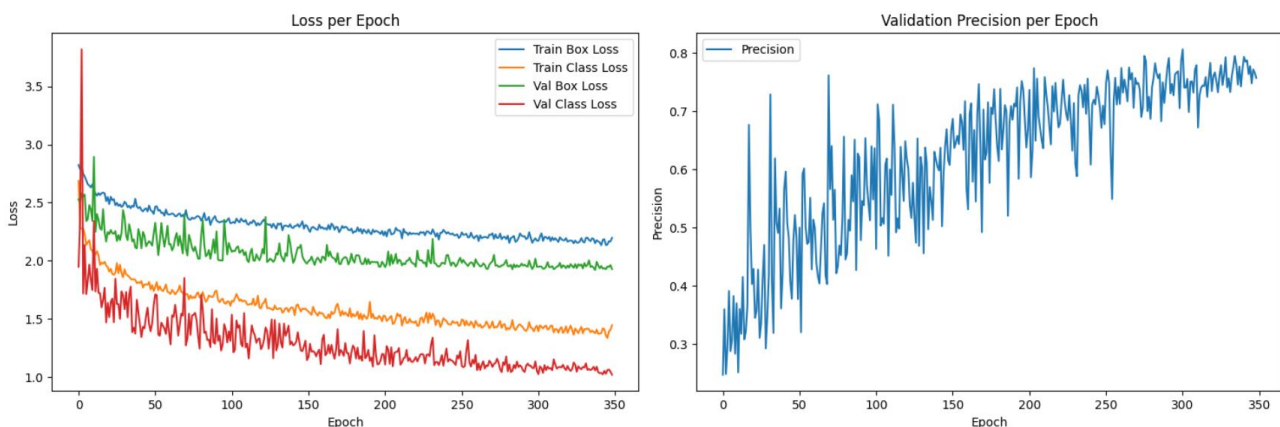


Figure 1: Key architectural modules in YOLOv11



Train Box/Class Loss và Validation Box/Class Loss đều giảm đều và ổn định. Không có hiện tượng loss tăng lại hoặc dao động mạnh → mô hình không bị overfitting.

Tuy nhiên, loss dừng lại ở mức cao hơn so với YOLOv8 ở biểu đồ trước → do mô hình YOLOv11n nhẹ hơn, ít tham số hơn.

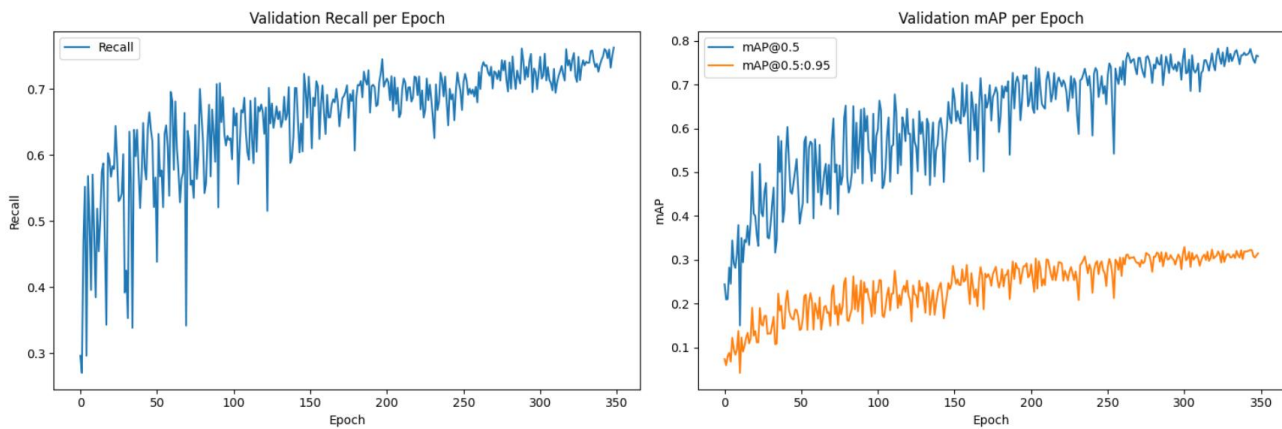
➔ Nhận xét: Quá trình học ổn định, nhưng mức loss cuối vẫn hơi cao, phản ánh khả năng học kém hơn mô hình lớn hơn.

Precision bắt đầu từ mức thấp (~0.3), tăng dần đều.

Đến khoảng epoch 300 đạt khoảng ~0.75-0.78, tuy chưa bằng YOLOv8 nhưng vẫn là mức tốt.

Dao động ban đầu lớn, sau dần ổn định.

➔ Nhận xét: Mô hình ngày càng phát hiện đúng nhiều đối tượng, mặc dù tốc độ tăng chậm hơn YOLOv8.



Bắt đầu từ ~ 0.25 , tăng đều và đạt gần ~ 0.75 ở cuối.

Dao động giảm dần cho thấy mô hình học ổn định hơn theo thời gian.

➔ Nhận xét: Mô hình đang bắt đầu phát hiện được đa số đối tượng cần thiết.

mAP@0.5 dao động nhưng xu hướng tăng đều, đạt ~ 0.75 ở cuối.

mAP@0.5:0.95 thấp hơn, đạt $\sim 0.3-0.32$ → phản ánh rằng mô hình chưa định vị thật chính xác ở nhiều mức IoU.

➔ Nhận xét:

mAP@0.5 đạt mức khá tốt.

mAP@0.5:0.95 vẫn hơi thấp, phù hợp với mô hình nhẹ YOLOv11n (ưu tiên tốc độ hơn độ chính xác chi tiết).

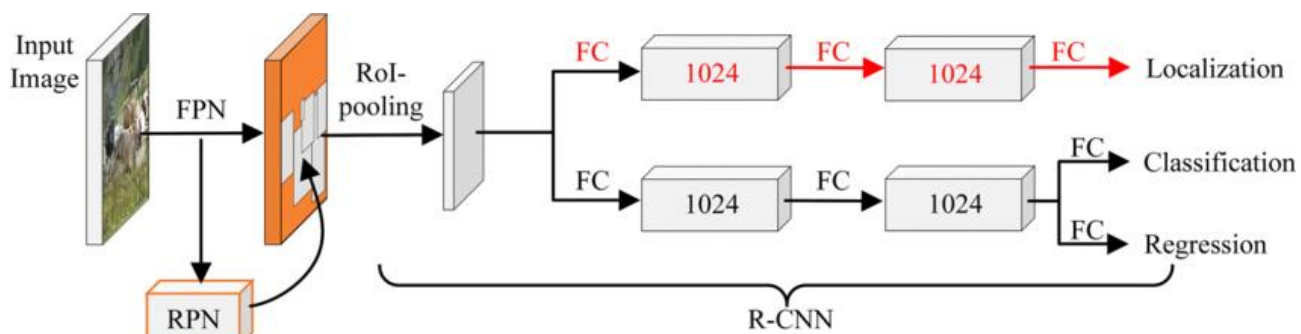
5.3 Faster R-CNN với ResNet-50 FPN

- Sử dụng ResNet-50 để trích xuất đặc trưng từ ảnh đầu vào.
- Kết hợp với FPN (Feature Pyramid Network)
- Bộ tối ưu hóa (Optimizer)

Learning rate	0.005	Tốc độ học (learning rate) ban đầu
---------------	-------	---------------------------------------

momentum	0.9	Giảm dao động, giúp tối ưu ổn định hơn
weight_decay	0.0005	Regularization để tránh overfitting

→ Num_epochs = 50



🕒 Inference time: 5012.25 ms/image (0.20 FPS)

💾 Model size: 158.09 MB



Kết quả test

VI. Kết luận và hướng phát triển

6.1 Kết luận

Tiêu chí	YOLOv8	YOLOv11n	Faster R-CNN (ResNet50-FPN)
Precision	≈ 0.83	≈ 0.77	≈ 0.85
Recall	≈ 0.81	≈ 0.75	≈ 0.88
mAP@0.5	≈ 0.85	≈ 0.75	≈ 0.86
mAP@0.5:0.95	≈ 0.33	≈ 0.32	≈ 0.38
Tốc độ suy luận (FPS)	Rất nhanh	Nhanh nhất	Chậm
Kích thước mô hình	Trung bình	Nhẹ nhất	Nặng

◆ YOLOv8:

Cho hiệu quả cao, tốc độ suy luận tốt.

Mô hình cân bằng giữa độ chính xác và hiệu năng, phù hợp dùng thực tế.

◆ YOLOv11n:

Mô hình rất nhẹ, thích hợp chạy trên thiết bị nhúng/di động.

Độ chính xác chấp nhận được ($\sim 75\%$) nhưng chưa đủ tốt cho các ứng dụng yêu cầu độ chính xác cao.

◆ Faster R-CNN:

Cho kết quả tốt nhất về mAP@0.5:0.95 và Recall.

Tuy nhiên, tốc độ suy luận chậm, khó triển khai thời gian thực trên thiết bị yếu.

6.2 Hướng phát triển

Dựa trên kết quả thực nghiệm và phân tích hiệu năng của ba mô hình phát hiện đối tượng là YOLOv8, YOLOv11n và Faster R-CNN (ResNet-50 FPN) trên tập dữ liệu bẫy vàng, đề tài đã đạt được những kết quả khả quan ban đầu về độ chính xác và khả năng tổng quát hóa của mô hình. Tuy nhiên, để nâng cao hiệu quả ứng dụng trong thực tế cũng như mở rộng phạm vi nghiên cứu, một số hướng phát triển tiếp theo có thể được đề xuất như sau:

1. Tối ưu hóa mô hình cho môi trường triển khai thực tế

Trong các ứng dụng như giám sát sâu bệnh trên đồng ruộng thông qua hình ảnh từ bẫy vàng, thời gian xử lý và độ chính xác đều đóng vai trò quan trọng. Do đó, việc lựa chọn và điều chỉnh mô hình cần cân nhắc đến điều kiện phần cứng thực tế như:

thiết bị biên (edge device), máy tính nhúng (Jetson Nano, Raspberry Pi), hoặc triển khai trên đám mây.

Một hướng đi tiềm năng là sử dụng các mô hình gọn nhẹ như YOLOv11n hoặc tối ưu YOLOv8 bằng các kỹ thuật như:

- Pruning (cắt tỉa trọng số)
- Quantization (lượng tử hóa mô hình)
- TensorRT/TFLite (chuyển mô hình sang định dạng tăng tốc)

Điều này có thể giúp giảm kích thước mô hình và thời gian suy luận mà vẫn giữ được độ chính xác ở mức chấp nhận được.

2. Nâng cao độ chính xác thông qua cải tiến dữ liệu và mô hình

Mặc dù kết quả mAP@0.5 và mAP@0.5:0.95 của các mô hình hiện tại là tương đối ổn định, vẫn còn nhiều không gian để cải thiện, đặc biệt là trong việc phát hiện các đối tượng nhỏ hoặc bị che khuất một phần. Một số cải tiến có thể thực hiện bao gồm:

- Tăng cường dữ liệu (Data Augmentation) nâng cao: sử dụng các kỹ thuật như Mosaic, CutMix, MixUp hoặc ảnh chụp từ nhiều góc độ khác nhau.
- Cải tiến nhãn dữ liệu: đánh giá và tinh chỉnh lại annotation, giảm lỗi gán nhãn sai giúp mô hình học chính xác hơn.
- Fine-tuning với backbone tốt hơn: ví dụ thay thế ResNet-50 bằng ResNeXt, EfficientNet hoặc ConvNeXt trong Faster R-CNN.
- Thử nghiệm các mô hình tiên tiến hơn như YOLOv9, YOLO-World, RT-DETR, Grounding DINO,... nhằm cải thiện khả năng biểu diễn đặc trưng.

3. Kết hợp mô hình (Ensemble hoặc Two-Stage Detection)

Một hướng tiếp cận khác là kết hợp các mô hình nhằm tận dụng ưu điểm của từng mô hình:

- YOLOv8 để phát hiện nhanh các vùng nghi ngờ (stage 1)
- Faster R-CNN để phân tích kỹ hơn các vùng đã phát hiện (stage 2)

Chiến lược này có thể giúp cân bằng giữa tốc độ suy luận và độ chính xác, đồng thời giảm false positives trong các ứng dụng yêu cầu độ tin cậy cao.

4. Tích hợp hệ thống và đánh giá toàn diện

Hướng phát triển tiếp theo nên hướng đến xây dựng một hệ thống hoàn chỉnh từ thu thập ảnh bẫy vàng tự động → tiền xử lý ảnh → phát hiện đối tượng → lưu kết quả hoặc cảnh báo. Hệ thống có thể tích hợp các thành phần như:

- Camera hoặc drone tự động chụp ảnh
- Bộ xử lý nhúng chạy mô hình

- Cơ sở dữ liệu lưu trữ kết quả và thống kê số lượng côn trùng
- Dashboard trực quan hóa dữ liệu và cảnh báo

Ngoài ra, cần mở rộng bộ dữ liệu và tiến hành đánh giá mô hình trên dữ liệu thực tế trong các điều kiện ánh sáng, môi trường khác nhau để kiểm tra tính ổn định và khả năng tổng quát của mô hình.

5. Hướng nghiên cứu mở rộng

Cuối cùng, hướng nghiên cứu có thể được mở rộng sang các bài toán liên quan như:

- Phân loại loại côn trùng theo mức độ nguy hiểm
- Dự đoán mật độ quần thể theo thời gian
- Ứng dụng học sâu trong phân tích diễn biến dịch hại và hỗ trợ ra quyết định nông nghiệp

Tổng kết

Việc ứng dụng học sâu trong nhận diện côn trùng bẫy vàng đã cho thấy tiềm năng rõ rệt trong việc tự động hóa giám sát nông nghiệp. Các hướng phát triển được đề xuất không chỉ giúp hoàn thiện mô hình hiện tại mà còn mở ra các khả năng ứng dụng rộng lớn hơn trong thực tiễn.