DOI: 10.31276/VJST.2025.3727 ***Khoa học tự nhiên*** | *Khoa học máy tính và thông tin*

Phát hiện bệnh trên lá cây trồng phổ biến ở Việt Nam bằng kiến trúc YOLO có khả năng giải thích

Nguyễn Trung Kiên, Lê Đình Hoàng Vũ, Võ Hoài Việt*

Khoa Công nghệ Thông tin, Trường Đại học Khoa học Tự nhiên, Đại học Quốc gia TP. Hồ Chí Minh, 227 Nguyễn Văn Cừ, phường Chợ Quán, TP. Hồ Chí Minh, Việt Nam

Ngày nhận bài 6/11/2025; ngày chuyển phản biện 13/11/2025;
ngày nhận phản biện 19/12/2025; ngày chấp nhận đăng 25/12/2025

Tóm tắt:

Nông nghiệp là ngành kinh tế trọng điểm của Việt Nam, nhưng đang phải đối mặt với thách thức nghiêm trọng từ sâu bệnh - nguyên nhân gây ra những tổn thất về năng suất nặng nề, ảnh hưởng đến GDP và xuất khẩu. Việc phát hiện và chẩn đoán bệnh sớm là yếu tố sống còn để triển khai các biện pháp can thiệp kịp thời, giảm thiểu thiệt hại. Tuy nhiên, các phương pháp truyền thống đang phụ thuộc nhiều vào con người, điều đó gây tốn thời gian, hiệu suất thấp và chi phí tốn kém, đặc biệt khi áp dụng trên quy mô lớn. Để giải quyết thách thức này, nghiên cứu của chúng tôi đề xuất một giải pháp tự động dựa trên học sâu, sử dụng kiến trúc phát hiện đối tượng thời gian thực YOLO. Mô hình được huấn luyện và đánh giá trên một bộ dữ liệu lớn chứa bệnh của các cây trồng chiến lược của Việt Nam là lúa, cà phê và chè. Kết quả trên tập đánh giá rất cao, cụ thể độ chính xác là 0,987 và độ chính xác trung bình mAP50 là 0,984. Qua đó, khẳng định hiệu suất vượt trội và tiềm năng to lớn trong việc xây dựng các hệ thống giám sát cây trồng thông minh, có khả năng giải thích và mở rộng ở Việt Nam.

Từ khóa: **nông nghiệp thông minh, phát hiện bệnh trên lá, xAI, yolo.**

Chỉ số phân loại: **1.2**

Detection of diseases on common crop leaves in Vietnam using explainable YOLO architectures

Trung Kien Nguyen, Dinh Hoang Vu Le, Hoai Viet Vo*

Faculty of Information Technology, University of Science, Vietnam National University - Ho Chi Minh, 227 Nguyen Van Cu Street, Cho Quan Ward, Ho Chi Minh, Vietnam

Received 6 November 2025; revised 19 December 2025; accepted 25 December 2025

Abstract:

Agriculture is a key economic sector in Vietnam but is increasingly confronted with significant challenges from pests and diseases, which lead to substantial yield losses and negatively impact both GDP and exports. Reports indicate that prevalent

* *Tác giả liên hệ: Email: vhviet@fit.hcmus.edu.vn*

diseases, such as rice blast, can result in yield reductions of 50-100%, while coffee dieback can decrease productivity by up to 75%. Early detection and diagnosis are critical for timely intervention to mitigate such damages. Nevertheless, traditional methods are predominantly manual, making them time-consuming, inefficient, and costly, particularly for large-scale applications. To address these challenges, this study proposes an automated solution based on deep learning, utilizing the YOLO real-time object detection architecture. The model was trained and evaluated on an extensive dataset of Vietnam's strategic crops, including rice, coffee, and tea. The experimental results demonstrated high performance, with an accuracy of 0.987 and an mAP50 of 0.984. These findings confirm the model's superior performance and its potential for developing intelligent, explainable, and scalable crop monitoring systems in Vietnam.

Keywords: **leaf disease detection, smart agriculture, xAI, yolo.**

Classification number: **1.2**

1. Đặt vấn đề

Nông nghiệp đóng một vai trò không thể thiếu trong việc cung cấp lương thực và nguyên liệu trên toàn thế giới, đặc biệt là Việt Nam. Các báo cáo từ những năm gần đây cho thấy tình hình dịch bệnh trên cây trồng ở Việt Nam vẫn diễn biến phức tạp, gây hại cho cây trồng và ảnh hưởng đến sản lượng cũng như kinh tế của người nông dân [1]. Trong đó, những bệnh phổ biến như đạo ôn lúa có thể gây thất thoát tới 50-100% sản lượng [2], hay bệnh khô cành, khô quả trên cây cà phê có thể làm giảm năng suất tới 75% [3]. Trong bối cảnh đó, việc phát hiện sớm và chính xác các mầm bệnh là yếu tố then chốt để có biện pháp can thiệp kịp thời, đảm bảo chất lượng nông sản, đồng thời kiểm soát được lượng thuốc trừ sâu sử dụng cho cây bệnh, tránh tình trạng phun thuốc đồng loạt, từ đó giảm thiểu thiệt hại cho môi trường đất. Bên cạnh đó, việc ứng dụng công nghệ và trí tuệ nhân tạo trong việc chẩn đoán dịch bệnh trên cây trồng sẽ giúp tạo ra nông nghiệp thông minh, hỗ trợ việc tạo ra năng suất cao.

Trong nhiều năm, phương pháp thủ công chủ yếu dựa vào quan sát và kinh nghiệm của các chuyên gia hoặc người nông dân. Mặc dù có giá trị, phương pháp thủ công này vẫn còn nhiều hạn chế: (1) tốn thời gian và chi phí nhân công cao; (2) phụ thuộc vào kinh nghiệm chủ quan, dẫn đến khả năng chẩn đoán không nhất quán; (3) có độ trễ thông tin lớn, khó có khả năng mở rộng trên các vùng canh tác quy mô lớn [4]. Mặc dù các phương pháp thị giác máy tính truyền thống đã được đề xuất, chúng thường yêu cầu các bước trích xuất đặc trưng thủ công phức tạp [5] và tỏ ra kém hiệu quả trong các điều kiện thực địa không được kiểm soát [6].

Sự trỗi dậy của các mô hình học sâu (Deep Learning), đặc biệt là Mạng nơ-ron tích chập (CNN), đã tạo ra một cuộc cách mạng trong lĩnh vực nhận dạng hình ảnh và mở ra

hướng đi đột phá cho việc chẩn đoán bệnh cây trồng tự động [7]. Nhiều công trình đã chứng minh khả năng của CNN trong việc phân loại bệnh với độ chính xác cao. Tuy nhiên, việc triển khai các mô hình này trong điều kiện thực tế tại Việt Nam vấp phải nhiều thách thức đáng kể [8]. Thứ nhất, dữ liệu hình ảnh thu thập từ người dùng cuối rất đa dạng về thiết bị, góc chụp và điều kiện ánh sáng, đòi hỏi mô hình phải có khả năng khái quát hóa cao để hoạt động cho môi trường thay đổi [9]. Thứ hai, các triệu chứng hình thái của một số loại bệnh ở giai đoạn đầu rất giống nhau, dễ gây nhầm lẫn [10]. Thứ ba, kích thước của các đốm bệnh biến đổi lớn, từ vài pixel đến chiếm trọn cả lá, yêu cầu mô hình phải có khả năng phát hiện đối tượng đa thang đo [11]. Thứ tư, bối cảnh phức tạp với lá cây chồng chéo và nền nhiễu đòi hỏi khả năng định vị đối tượng chính xác [12]. Thứ năm, yêu cầu về hiệu quả tính toán là một rào cản thực tiễn, mô hình cần có khả năng suy luận thời gian thực trên các thiết bị có tài nguyên hạn chế như điện thoại di động [13]. Cuối cùng, bản chất "hộp đen" của các mô hình học sâu là một thách thức lớn, vì thiếu tính diễn giải khiến người dùng khó tin tưởng vào các chuẩn đoán của mô hình, làm giảm khả năng chấp nhận và triển khai công nghệ [14].

Trong các bài toán về nhận dạng đối tượng được chứng minh qua các tập dữ liệu COCO, các mô hình có mAP cao cũng bị sụt giảm hiệu suất nghiêm trọng (lên tới 55,7%) [15]. Điều này cho thấy mô hình cần phải có một khả năng chống chịu tốt với thách thức về sự đa dạng dữ liệu. Nghiên cứu này đề xuất một hệ thống phát hiện bệnh tự động dựa trên kiến trúc YOLOv8 [16], một trong những mô hình phát hiện đối tượng một pha (one-stage) tiên tiến nhất hiện nay. YOLOv8 được lựa chọn vì khả năng cân bằng vượt trội giữa tốc độ và độ chính xác, cũng như hiệu quả trong việc phát hiện các đối tượng có kích thước đa dạng. Đóng góp chính của chúng tôi bao gồm:

- Đánh giá mô hình YOLO và ứng dụng để phát hiện đa bệnh trên ba loại cây trồng chiến lược của Việt Nam: lúa, cà phê và chè. Kết quả đạt được tỷ lệ mAP50 cao và nhất quán: 0.984 trên cây cà phê, 0.983 trên cây lúa và 0.991 trên cây chè.

- Áp dụng và đánh giá các phương pháp giải thích Grad-CAM và HiResCAM cho mô hình YOLOv8 trong bài toán này, làm tăng tính minh bạch và tin cậy của mô hình.

- Giải thích và làm rõ các độ đo được sử dụng trong việc giải thích Grad-CAM và HiResCAM của mô hình.

Tiếp theo, trong phần 2, chúng tôi sẽ trình bày các công trình nghiên cứu liên quan tập trung vào các thách thức mà chúng tôi đang gặp phải. Phần 3, chúng tôi sẽ trình bày về kiến trúc của mô hình YOLOv8 được sử dụng, các kĩ thuật tăng cường dữ liệu và mô hình xAI được sử dụng trong đề xuất của chúng tôi. Phần 4 và 5 sẽ trình bày thực nghiệm, thảo luận cũng như phân tích và kết luận về nghiên cứu này.

2. Các công trình nghiên cứu liên quan

2.1. Cải tiến kiến trúc mô hình cho môi trường phức tạp

Để giải quyết các thách thức trong việc phát hiện bệnh trên cây trồng trong môi trường nông nghiệp phức tạp, nghiên cứu của Miao và cộng sự (2025) [12] đã đề xuất một kiến trúc YOLOv8 cải tiến có tên là SerpensGate YOLOv8. Các môi trường thực tế thường có nhiều yếu tố gây nhiễu như cành, lá, và quả che khuất vùng bệnh, đòi hỏi mô hình phải có khả năng nhận diện các đặc trưng phức tạp. Nhận thấy những hạn chế này, các tác giả đã thực hiện ba cải tiến quan trọng cho kiến trúc YOLOv8 gốc:

- Tích hợp Dynamic Snake Convolution (DySnake Conv) vào mô-đun C2f: Cải tiến này được thực hiện để nâng cao khả năng phát hiện các đặc trưng tinh vi trong các cấu trúc phức tạp và kéo dài, chẳng hạn như các vết bệnh có hình dạng uốn lượn, không đều. Khác với các lớp tích chập tiêu chuẩn có vùng tiếp nhận cố định, DySnakeConv có thể tự động điều chỉnh hình dạng để bám sát các đường viền phức tạp của vết bệnh, giúp trích xuất đặc trưng hình thái chính xác hơn.

- Thay thế mô - đun SPPF bằng SPPELAN: Mô - đun SPPELAN là sự kết hợp giữa Spatial Pyramid Pooling (SPP) và Efficient Local Aggregation Network (ELAN). Việc thay thế này nhằm mục đích tăng cường khả năng trích xuất và hợp nhất đặc trưng ở nhiều quy mô khác nhau. Điều này rất quan trọng để phát hiện các đốm bệnh có kích thước đa dạng, từ những đốm nhỏ đến các vùng bị hại lớn, giúp mô hình có cái nhìn toàn diện hơn về bối cảnh.

- Giới thiệu cơ chế Super Token Attention (STA): Cơ chế chú ý (attention) này được thêm vào để tăng cường khả năng mô hình hóa các đặc trưng toàn cục ngay từ các giai đoạn đầu của mạng. STA giúp mô hình tập trung vào các vùng quan trọng bị ảnh hưởng bởi bệnh tật và tăng cường khả năng chống chịu trước các nền ảnh phức tạp và nhiễu.

Nhờ những cải tiến này, mô hình SerpensGate YOLOv8 đã giải quyết được các vấn đề về nhận dạng đối tượng có hình dạng bất thường và trích xuất đặc trưng đa quy mô, giúp tăng chỉ số mAP50 lên 3,3% so với mô hình YOLOv8 gốc trên tập dữ liệu PlantDoc. Tuy nhiên, việc thêm những cải tiến này đã làm giảm đáng kể thời gian xử lý so với phiên bản gốc, SerpensGate YOLOv8 đạt tốc độ suy luận 5,9 ms, trong khi YOLOv8 bản chuẩn của chúng tôi có tốc độ chỉ 2,33 ms. Ngoài ra, việc thiết kế các cải tiến chuyên biệt này theo tập dữ liệu PlantDoc với mAP chỉ đạt 64,9%, kết quả này không đảm bảo mô hình đạt được kết quả ổn định khi thực nghiệm trên nhiều tập dữ liệu khác nhau. Nghiên cứu hướng đến mô hình có khả năng triển khai trên nhiều thiết bị và cần thực nghiệm trên nhiều tập dữ liệu, một mô hình đảm bảo độ chính xác trên nhiều tập dữ liệu cũng như đảm bảo tốc độ là điều cần thiết. Chính vì vậy, mô hình YOLOv8 gốc là một kiến trúc khả thi cần nghiên cứu.

2.2. Thách thức về thu thập và xử lý dữ liệu chuyên biệt

Một trong những thách thức cơ bản nhất trong việc phát hiện bệnh cây trồng là xây dựng một bộ dữ liệu chất lượng cao và phù hợp với môi trường cụ thể. Nghiên cứu của Trinh Cong Dong và cộng sự (2024) [8] về phát hiện bệnh trên lá lúa tại Việt Nam là một ví dụ điển hình cho việc giải quyết vấn đề này.

Để xây dựng một mô hình hiệu quả cho bối cảnh địa phương, nhóm tác giả đã tiếp cận theo hướng thu thập dữ liệu hỗn hợp. Đầu tiên, họ đã chụp trực tiếp 1,634 ảnh tại các cánh đồng thực nghiệm của Học viện Nông nghiệp Việt Nam. Quá trình này đảm bảo rằng 2 dữ liệu phản ánh đúng điều kiện thực tế về giống cây, loại bệnh và môi trường canh tác tại Việt Nam. Tuy nhiên, nhóm nghiên cứu phải đối mặt với nhiều khó khăn như số lượng ảnh thực địa còn hạn chế và điều kiện chụp không đồng nhất (thay đổi giữa trời nắng và có mây, ảnh bị mờ do chụp ngược sáng).

Để khắc phục những hạn chế này và tăng cường tính đa dạng cho dữ liệu, nghiên cứu đã bổ sung thêm ảnh từ các nguồn trên Internet để đạt tổng cộng 3,175 ảnh. Quan trọng hơn, họ đã áp dụng một loạt các kỹ thuật tăng cường dữ liệu (data augmentation) mạnh mẽ như thay đổi không gian màu HSV, dịch chuyển, thay đổi tỷ lệ, lật ảnh và đặc biệt là kỹ thuật mosaic. Kỹ thuật mosaic [8] ghép bốn ảnh ngẫu nhiên thành một, tạo ra bối cảnh mới và phức tạp hơn, buộc mô hình phải học cách nhận diện các đối tượng trong nhiều ngữ cảnh khác nhau. Cách tiếp cận này cho thấy, việc kết hợp dữ liệu thực địa được xác thực bởi chuyên gia với các phương pháp tăng cường dữ liệu hiệu quả là một chiến lược khả thi để giải quyết bài toán thiếu dữ liệu và huấn luyện các mô hình có khả năng nhận dạng tốt trong một môi trường canh tác cụ thể.

2.3. Sử dụng và tối ưu hóa bộ dữ liệu công khai quy mô lớn

Trái ngược với việc xây dựng bộ dữ liệu chuyên biệt, một hướng tiếp cận khác là sử dụng các bộ dữ liệu công khai có quy mô lớn, như trong nghiên cứu của Abid và cộng sự (2024) [17] về việc phát hiện bệnh trên năm loại cây trồng chủ lực của Bangladesh. Nhóm tác giả đã sử dụng bộ dữ liệu công khai Plant Village, một tập dữ liệu rất lớn với hơn 21,000 hình ảnh.

Vấn đề chính khi làm việc với một bộ dữ liệu lớn như vậy là nguy cơ mất cân bằng giữa các lớp bệnh và chi phí tính toán cao khi huấn luyện trên toàn bộ dữ liệu. Để giải quyết vấn đề này, các nhà nghiên cứu đã áp dụng một chiến lược xử lý thông minh:

Tạo một tập dữ liệu con cân bằng: Thay vì sử dụng toàn bộ hơn 21,000 ảnh, họ đã tuyển chọn và tạo ra một bộ dữ liệu con có tổ chức và cân bằng hơn. Cụ thể, họ đã chọn ra đúng 150 hình ảnh cho mỗi một trong 19 lớp bệnh, tạo ra một tập dữ liệu ban đầu gồm 2,850 ảnh. Cách tiếp cận này giúp ngăn chặn sự thiên vị (bias) của mô hình trong quá trình huấn luyện, đảm bảo rằng mỗi loại bệnh đều được thể hiện một cách công bằng, đồng thời giúp quá trình huấn luyện trở nên khả thi và dễ quản lý hơn.

Tăng cường dữ liệu để bù đắp: Sau khi tạo ra một bộ dữ liệu con nhỏ hơn, vấn đề tiếp theo là nguy cơ mô hình bị quá khớp (overfitting) do số lượng ảnh hạn chế. Để giải quyết điều này, họ đã sử dụng nền tảng Roboflow để áp dụng các kỹ thuật tăng cường dữ liệu như xoay và lật ảnh. Quá trình này đã tăng gần gấp đôi số lượng ảnh trong bộ dữ liệu từ 2,850 lên 5,750 ảnh, giúp tăng tính đa dạng và giảm thiểu nguy cơ quá khớp.

Nghiên cứu [17] nhấn mạnh một phương pháp hiệu quả để làm việc với các bộ dữ liệu công khai quy mô lớn: đó là không nhất thiết phải sử dụng toàn bộ dữ liệu, mà thay vào đó, có thể tạo ra một tập dữ liệu con cân bằng và dễ quản lý, sau đó sử dụng các kỹ thuật tăng cường dữ liệu để nâng cao chất lượng và số lượng, đảm bảo mô hình được huấn luyện một cách mạnh mẽ.

2.4. Quá trình phát triển của các phương pháp phát hiện vật thể trước YOLO

Mô hình thích ứng vị trí bộ phận (Deformable Part Models) [18] xem mỗi vật thể như một tập hợp các phần (part) liên kết với phần gốc. Các phần có vị trí linh hoạt (deformable) để thích nghi với sự biến dạng vật thể. Ý tưởng là trượt cửa sổ trên ảnh ở nhiều tỷ lệ để tìm vật thể.

Tìm kiếm có chọn lọc (Selective Search) [19] giảm chi phí của việc quét cửa sổ trượt trên toàn ảnh bằng cách tạo ra các vùng có khả năng chứa vật thể dựa trên phân đoạn ảnh và gộp các vùng tương đồng.

R-CNN [20] sử dụng tìm kiếm có chọn lọc là bước đầu tiên để tạo các vùng đề xuất sau đó chạy một mô hình phân loại trên các vùng đề xuất này. Sau khi phân loại, bước hậu xử lý sẽ tinh chỉnh lại hộp giới hạn, loại bỏ các dự đoán trùng và tính điểm lại cho các dự đoán dựa trên các dự đoán khác.

Fast R-CNN [21] vẫn sử dụng các thành phần giống R-CNN, tuy nhiên việc sinh vùng trước dẫn đến phải chạy mạng CNN qua nhiều vùng tốn chi phí. Fast R CNN cải tiến bằng cách chạy mạng CNN trước để trích xuất đặc trưng, sau đó mới sinh vùng. Tiếp theo, R-CNN cắt đặc trưng theo các vùng này rồi cho qua một đầu phân loại và tinh chỉnh như R-CNN.

Faster R-CNN [22] cải tiến Fast R-CNN bằng cách dùng một mạng CNN con học trực tiếp cách sinh các vùng đề xuất ngay trong mô hình, dẫn đến giảm chi phí.

2.5. Quá trình phát triển của YOLO

YOLOv1 [23] xem phát hiện vật thể là một nhiệm vụ hồi quy, từ ảnh đầu vào, nó dự đoán xác suất và tọa độ hộp giới hạn chỉ trong một lần lan truyền tiến. Nó sử dụng lưới 7x7, mỗi ô dự đoán hai vật thể. Hạn chế là khó dự đoán vật thể nhỏ và vật thể gần nhau vì kích thước ô lưới lớn, mỗi ô chỉ dự đoán hai vật thể.

YOLOv2-7 [24] cải tiến về thiết kế hộp neo, các phép tăng cường dữ liệu, mạng xương sống mạnh mẽ hơn, kiến trúc cổ kết hợp đặc trưng đa tỷ lệ.

YOLOv8 [16] có 3 cải tiến. Thứ nhất, nó cải tiến mạng xương sống bằng việc thay CSPDarknet bằng C2f để giữ nguyên khả năng trích xuất đặc trưng nhưng lại giảm số lượng tham số, sử dụng hàm kích hoạt SiLU giúp giảm triệt tiêu gradient. Thứ hai, nó dự đoán bằng điểm neo (mỗi ô trong bản đồ đặc trưng cho một dự đoán) thay cho việc dùng hộp neo định sẵn kém linh hoạt. Cuối cùng, nó sử dụng kiến trúc đầu (Head) tách riêng hai đầu cho phân loại và dự đoán hộp giới hạn, tuy được giới thiệu lần đầu tiên ở YOLOX (giữa YOLOv5 và YOLOv6) nhưng được YOLOv8 tinh chỉnh kiến trúc và tích hợp dự đoán không hộp neo. Việc tích hợp dự đoán không hộp neo vào kiến trúc đầu đã làm cho YOLOv8 trở nên mạnh mẽ, minh chứng rõ ràng nhất ở việc ultralytics tạo ra phiên bản YOLOv3u và YOLOv5u tích hợp kiến trúc đầu này để tăng hiệu suất. Hơn nữa, các phiên bản sau đó cũng sử dụng kiến trúc đầu này.

YOLOv9-12 [25] cải tiến kiến trúc mạng để trích xuất đặc trưng tốt hơn cho vật thể nhỏ, tăng cường tổng hợp đặc trưng đa tỷ lệ, sử dụng cơ chế chú ý cục bộ để học tốt hơn trong điều kiện phức tạp.

YOLOv13 [26] có 3 cải tiến. Thứ nhất, nó cải tiến cơ chế chú ý 1-1 thành cơ chế chú ý nhiều-nhiều, từ chú ý trong bản đồ đặc trưng thành chú ý giữa các bản đồ đặc trưng giữa các lớp khác nhau nên kết hợp được thông tin đa tỷ lệ. Thứ hai, nó phân phối toàn bộ đặc trưng đã được tăng cường tương quan (bởi cơ chế chú ý) đến toàn bộ mạng. Thứ ba, nó thay thế lớp tích chập thông thường thành tích chập sau tách biệt (lấy cảm hứng từ MobileNet) để giảm đáng kể độ phức tạp tính toán.

YOLOv8 tuy là phiên bản cũ nhưng nó ổn định và được tối ưu tốt. Các phiên bản mới hơn được báo cáo là tốt hơn (trên bộ dữ liệu chuẩn COCO) nhưng lại không tốt hơn nhiều và chưa được tối ưu tốt, bằng chứng là kết quả YOLOv13 kém hơn YOLOv8 trên các bộ dữ liệu mà chúng tôi thực nghiệm. Một lí do khác là độ trễ của YOLOv8 thấp hơn (tốt hơn) các phiên bản sau đó (chúng tôi so sánh YOLOv8-13 phiên bản s - small), tuy không nhiều nhưng ảnh hưởng đến các thiết bị có tài nguyên hạn chế. Cụ thể độ trễ của YOLOv8s là 2.33 ms, trong khi độ trễ của YOLOv9s-13s dao động từ 2.53 đến 3.44 ms. Đó là lí do chúng tôi sử dụng YOLOv8 làm kiến trúc chính cho bài toán này.

2.6. Giải thích mô hình quyết định

Hiệu suất của các mô hình học sâu thường cao nhưng lại đánh đổi khả năng giải thích. Chúng thường hoạt động như một hộp đen, chỉ cho ra kết quả mà không giải thích được lí do mô hình ra quyết định. AI khả diễn (xAI) ra đời vì lí do này và xAI trong thị giác máy tính tập trung vào 4 hướng giải thích chính [27]. trong bài báo này, chúng tôi trình bày trình bày bản đồ nhiệt để cho người đọc thấy được trực tiếp vùng mà mô hình chú ý khi đưa ra quyết định. Cụ thể, chúng tôi sử dụng HiResCAM [28] và so sánh với GradCAM [29] để thấy được ưu điểm của HiResCAM, các độ đo và kết quả được trình bày trong phần thực nghiệm.

3. Phương pháp nghiên cứu

YOLO là viết tắt của You Only Look Once, cái tên đã thể hiện ý tưởng của phương pháp: chỉ nhìn một lần duy nhất. Cái tên này xuất hiện bởi vì chuỗi các phương pháp dùng R-CNN (R-CNN [20], Fast R-CNN [21], Faster R-CNN [22]) để phát hiện vật thể trước đó nhìn vào ảnh nhiều lần với hai bước là: sinh vùng đề xuất và phân loại. R-CNN phải chạy phân loại trên nhiều vùng nên tốn thời gian, chi phí tính toán. YOLO khắc phục vấn đề này bằng cách sử dụng một kiến trúc duy nhất (hình 1): nhìn vào ảnh một lần và cho ra kết quả. Bảng 1 mô tả các thuật ngữ và từ viết tắt chính được sử dụng trong phương pháp đề xuất của chúng tôi.

Bảng 1. Bảng tra cứu thuật ngữ quan trọng.

Từ viết tắt	Mô tả
batch	Một lô ảnh, trong huấn luyện và kiểm thử, máy tính tính toán với nhiều ảnh cùng lúc
Momentum	Hệ số điều chỉnh để cập nhật trung bình, phương sai trong lớp BatchNorm2d
BottleNeck	Khối cổ chai, có tên gọi này vì số kênh đặc trưng nó xử lý ít hơn số kênh đầu vào, số kênh đầu ra
CSP	Là viết tắt của Cross Stage Partial. Khối này thực hiện phân tách bản đồ đặc trưng ở một giai đoạn (stage) của mạng để xử lý một phần (partial) và kết hợp với phần còn lại, tạo ra một luồng thông tin chéo (cross) qua các giai đoạn
C2f	Ý tưởng giống CSP nhưng kết hợp tất cả đầu ra trung gian (full) và thực hiện song sor nhanh hơn (faster). Nó sử dụng 2 khối tích chập trong mỗi BottleNeck, 2 khối tích chập r BottleNeck
SPP	Là viết tắt của Spatial Pyramid Pooling (tầng gộp kim tự tháp không gian). Nó thực hiện gộp bản đồ đặc trưng với kích thước tăng dần làm kích thước đầu ra nhỏ dần, khi xếp chồng sẽ có hình kim tự tháp
SPPF	Phiên bản cải tiến của SPP để xử lý nhanh hơn (faster)

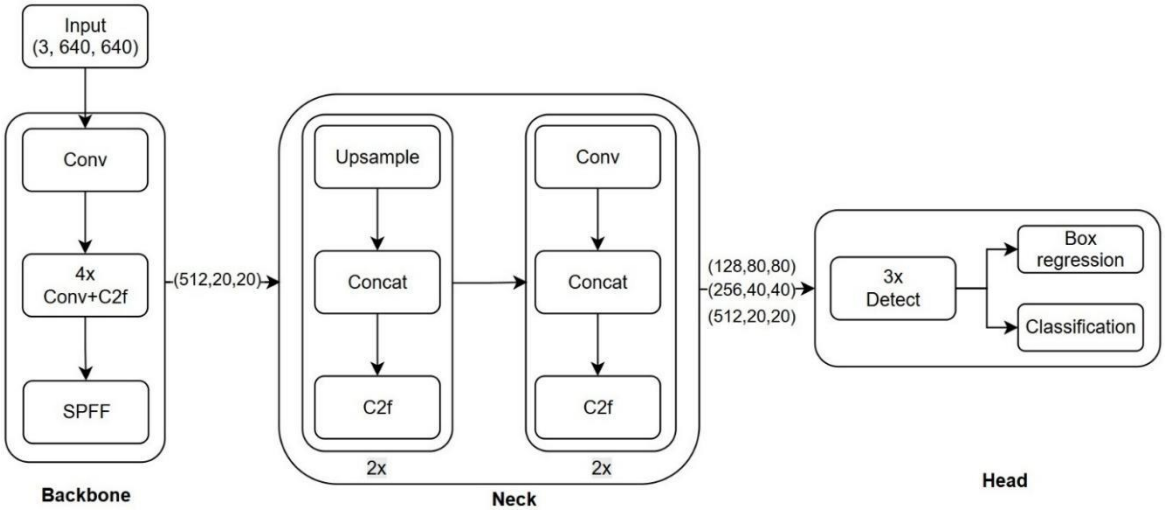
Bài toán được đặt ra trong nghiên cứu này là xây dựng một mô hình học sâu có khả năng tự động phát hiện và phân loại các loại bệnh trên lá cây nông nghiệp từ hình ảnh kỹ thuật số. Mô hình cần giải quyết các yêu cầu và ràng buộc cụ thể như sau:

Đầu vào: Một ảnh kỹ thuật số (RGB) chụp lá cây nông nghiệp. Ảnh có thể được chụp trong các điều kiện môi trường đa dạng, không được kiểm soát chặt chẽ, bao gồm nhiều góc chụp, khoảng cách và điều kiện ánh sáng khác nhau (nắng gắt, trong bóng râm, ngược sáng). Ảnh đầu vào có thể chứa một hoặc nhiều lá, và các lá có thể bị che khuất một phần.

Đầu ra: Đối với mỗi đốm bệnh được phát hiện trong ảnh đầu vào, mô hình phải cung cấp:

1) Hộp giới hạn (Bounding box): Tọa độ của hộp chữ nhật bao quanh đốm bệnh.
2) Tên bệnh (Class label): Nhãn của một trong bốn lớp bệnh đã được xác định.

3) Điểm tin cậy (Confidence score): Một giá trị số thể hiện mức độ chắc chắn của mô hình đối với dự đoán đã đưa ra.



Hình 1: Kiến trúc YOLOv8. Mạng xương sống học đặc trưng bằng cách gấp đôi số kênh, giảm một nửa kích bước bản đồ đặc trưng kết hợp SPPF để tăng vùng nhìn. Cổ (Neck) học thông tin vị trí trong giai đoạn đầu, học thông tin ngữ nghĩa trong giai đoạn sau. Đầu (Head) có 3 đầu để dự đoán vật thể với các kích thước khác nhau, tách biệt đầu để tối ưu cho dự đoán lớp riêng, tối ưu cho dự đoán hộp giới hạn riêng.

3.1. Giải pháp vấn đề dữ liệu

Như đã đề cập trong phần 1 và 2, các bộ dữ liệu có sẵn thường không phản ánh chính xác thực tế. Do đó, chúng tôi áp dụng các phép biến đổi để tăng tính tổng quát của dữ liệu, buộc mô hình học các đặc trưng có tính bất biến với ánh sáng, kích thước đầu vào, góc chụp. Các phép biến đổi mà nhóm sử dụng được mô tả trong bảng 2. Phép biến đổi quan trọng nhất là mosaic, vừa giúp mô hình học được đặc trưng của vật thể nhỏ, vừa giúp mô hình học được bối cảnh phức tạp (được minh họa trong hình 2). Thuật toán tăng cường dữ liệu được trình bày trong giải thuật 1 - tạo ảnh mosaic từ 4 ảnh đầu vào.

Bảng 2. Các phép biến đổi được sử dụng.

Phép biến đổi	**Mô tả**
Mosaic	Ghép 4 ảnh thành một ảnh
Fliplr	Lật ảnh theo chiều ngang
Hsv_h	Thay đổi màu sắc
Hsv_s	Thay đổi độ đậm nhạt màu sắc
Hsv_v	Thay đổi độ sáng
Scale	Thay đổi kích thước ảnh
Translate	Dịch ảnh theo phương đứng hoặc phương ngang

Giải thuật 1: Tạo ảnh mosaic từ 4 ảnh đầu vào

Đầu vào: Kích thước đầu vào s, 4 ảnh đầu vào img đã chuẩn hóa kích thước s (giữ tỷ lệ khung hình) kèm nhãn $label$ function MOSAIC(img, s, label)

Khởi tạo ảnh img4 màu đen, kích thước 2s x 2s, nhãn label4 = ∅

xc, yc = random(s/2, 3s/2), random(s/2, 3s/2)

for i=0 to 4 do

h,w = img[i].size

(x1, y1) → (x2, y2): vùng ảnh nhỏ được cắt

(x41, y41) → (x42, y42): vùng ảnh lớn được thêm

if i=0 then

x42, y42 ← xc, yc

x41, y41 ← max(xc-w, 0), max(yc-h, 0)

x2, y2 ← w, h

x1, y1 ← w – (x42-x41), h – (y42-y41)

else if i=1 then

x41, y42 ← 0, yc

x42, y41 ← min(xc+w, 2s), max(yc-h, 0)

x2, y2 ← 0, h

x1, y1 ← x42-x41, h – (y42-y41)

else if i=2 then

x42, y42 ← xc, 0

x41, y41 ← max(xc-w, 0), min(y+h, 2s)

x2, y2 ← w, 0

x1, y1 ← w – (x42-x41), y42-y41

else

x42, y42 ← xc, yc

x41, y41 ← min(xc+w, 2s), min(yc+h, 2s)

x2, y2 ← 0, 0

x1, y1 ← x42-x41, y42-y41

end if

Cập nhật nhãn label4 cho ảnh img4

end for

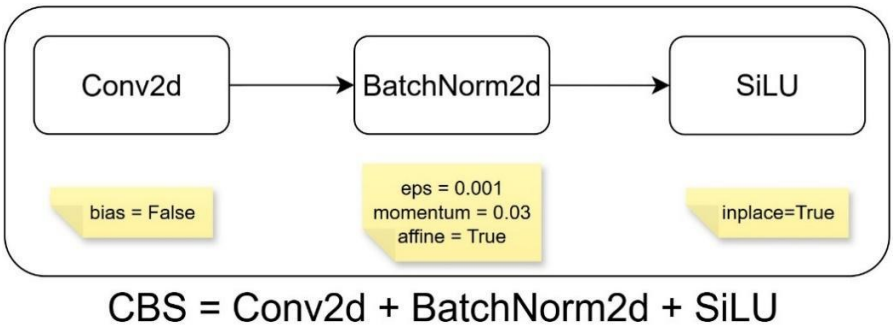
return img4, label4

end function

Hình 2. Mosaic ghép 4 ảnh thành một ảnh để học các vật thể nhỏ, học trong điều kiện phức tạp.

3.2. Khối tích chập

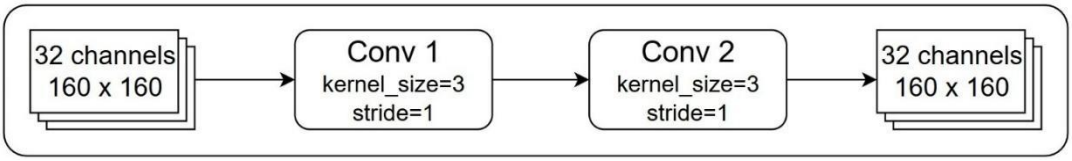
Khối tích chập có nhiệm vụ học đặc trưng, là sự kết hợp của 3 lớp: Conv2d, BatchNorm2d và SiLU. Do đó, 1 số tài liệu viết tắt là CBS (hình 3). Lớp Conv2d học đặc trưng, BatchNorm2d chuẩn hóa các kênh đặc trưng, hàm kích hoạt được sử dụng là SiLU để khắc phục vấn đề triệt tiêu gradient.



Hình 3. Khối tích chập là sự kết hợp của 3 lớp: lớp tích chập, lớp chuẩn hóa theo lô, lớp kích hoạt. Nó là cấu trúc chuẩn của các mạng học sâu hiện nay.

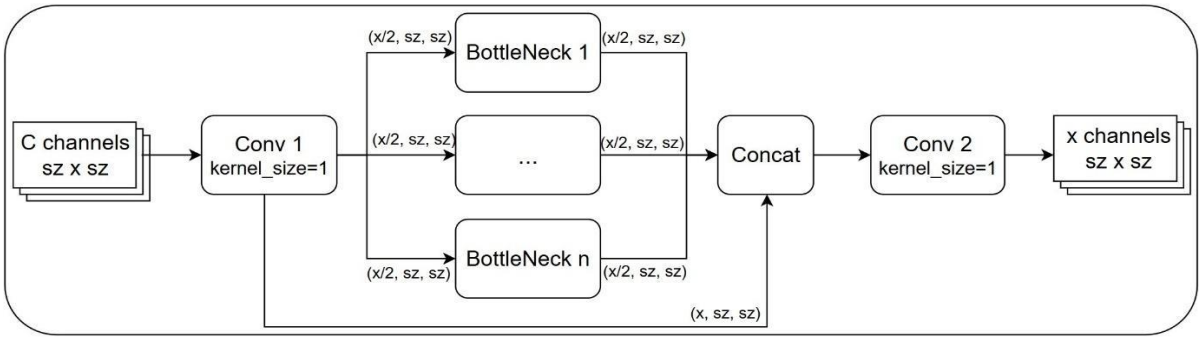
3.3. Khối C2f

Khối BottleNeck (hình 4) có nhiệm vụ kết hợp đặc trưng, gồm 2 khối tích chập liên tiếp học đặc trưng không thay đổi số kênh, kích thước bản đồ đặc trưng.



Hình 4. BottleNeck học cách kết hợp các kênh.

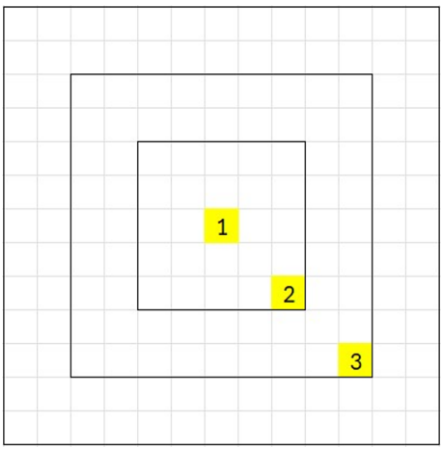
Khối C2f (hình 5) có nhiệm vụ học đặc trưng, gồm nhiều khối BottleNeck song song và 2 khối tích chập. Nhiệm vụ của chuỗi BottleNeck, Concat và Conv là học lại đặc trưng từ đầu ra của khối tích chập đầu tiên. Nếu bỏ chuỗi này đi, kích thước đầu ra không đổi nhưng mạng sẽ học kém hơn. BottleNeck được thiết kế song song nhằm 2 mục đích: giảm hiện tượng triệt tiêu gradient và tận dụng khả năng xử lý song song của GPU.



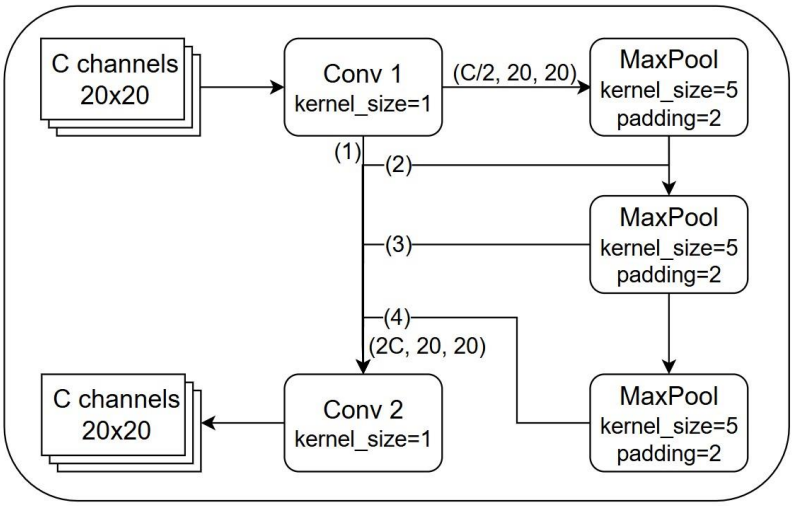
Hình 5. C2f sử dụng nhiều nhánh BottleNeck để kết hợp đặc trưng hiệu quả.

3.4. Mô - đun SPPF

Mô - đun SPPF có nhiệm vụ tăng vùng nhìn, gồm 3 lớp MaxPool xen giữa 2 khối tích chập. Mỗi điểm ảnh sau MaxPool 1 nhìn được một vùng 5x5, sau MaxPool 2 nhìn được một vùng 9x9, sau MaxPool 3 nhìn được một vùng 13x13 trên bản đồ đặc trưng 20x20 (hình 6). Mục tiêu là mỗi điểm ảnh chứa ngữ cảnh của một vùng lớn. Tuy nhiên, việc này làm mất thông tin vị trí. Do đó, ta tiến hành nối các kết quả từ Conv 1, MaxPool 1, MaxPool 2, MaxPool 3 để vừa có thông tin vị trí, vừa có thông tin tổng quát. Sau đó, đưa kết quả này qua lớp tích chập thứ hai để giảm số kênh. Luồng hoạt động được mô tả trong hình 7.



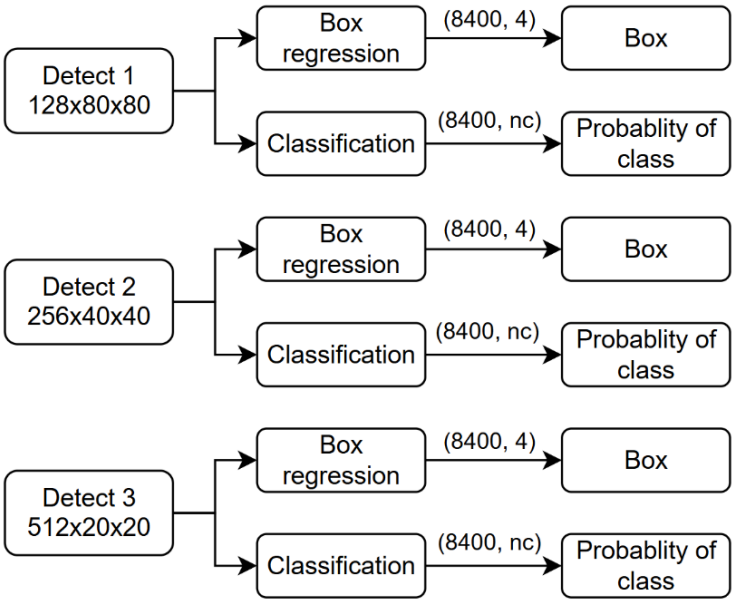
Hình 6. Vùng nhìn tăng mạnh sau 3 lớp MaxPool.



Hình 7. SPPF học cách kết hợp các kênh có vùng nhìn rộng từ 3 lớp MaxPool và các kênh ban đầu chứa thông tin vị trí.

3.5. Mô - đun phát hiện

Mô - đun phát hiện có nhiệm vụ dự đoán lớp và dự đoán hộp giới hạn, gồm 2 nhánh, mỗi nhánh thực hiện một nhiệm vụ. Mỗi ô trong bản đồ đặc trưng dự đoán 1 đầu ra gồm: đầu ra phân loại và đầu ra hộp giới hạn. Đầu ra phân loại là xác suất mỗi lớp, đầu ra hộp giới hạn là tọa độ hộp giới hạn. YOLOv8 còn thêm 1 lớp dự đoán phân phối tọa độ hộp giới hạn. Kiến trúc đầu được thể hiện trong hình 8.



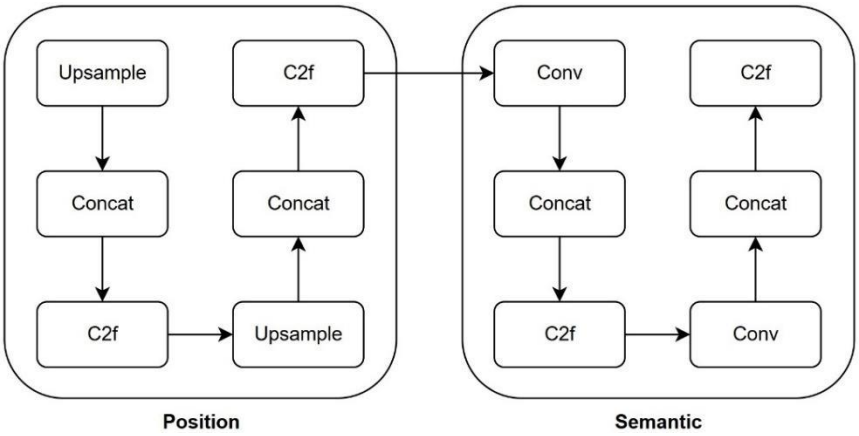
Hình 8. Phần đầu (Head) có 3 đầu dự đoán vật thể với các kích thước khác nhau, trong đó tách biệt đầu để tối ưu cho từng nhiệm vụ.

3.6. Vai trò của ba phần trong kiến trúc mạng

Mạng xương sống liên tục thực hiện Conv+C2f để tăng số kênh (học thêm đặc trưng) và giảm kích thước bản đồ đặc trưng để giảm độ phức tạp tính toán. Lớp SPPF ở cuối để tăng vùng nhìn của mỗi ô trong bản đồ đặc trưng.

Cổ (Neck) nhận đầu vào từ mạng xương sống rất giàu thông tin ngữ nghĩa. Tuy nhiên, kích thước bản đồ đặc trưng liên tục bị giảm ở mạng xương sống dẫn đến mất thông tin vị trí. Do đó, ở nửa đầu cổ (neck), UpSample+Concat+C2f được thực hiện 2 lần (hình 9). UpSample để phóng to gấp đôi các bản đồ đặc trưng ở cổ (neck). Concat để kết hợp các bản đồ đặc trưng vừa phóng to ở cổ (neck) với các bản đồ đặc trưng có cùng kích thước ở mạng xương sống để lấy lại thông tin vị trí. Ngoài ra, C2f được sử dụng để học cách kết hợp các kênh mới. Sau 2 lần UpSample+Concat+C2f ta thu được đầu vào thứ nhất (P3) của Đầu (Head) vừa có thông tin ngữ nghĩa (từ đầu ra mạng xương sống), vừa có thông tin vị trí. Cuối cùng, Conv+Concat+C2f được thực hiện 2 lần để học lại thông tin ngữ nghĩa, tạo 2 đầu ra khác (P4, P5) cho Đầu (Head) với kích thước nhỏ hơn. P3 tức là kích thước bản đồ đặc trưng bị thu nhỏ 23=8 lần so với kích thước thước ảnh đầu vào. Ví dụ với ảnh đầu vào kích thước 640x640 (3 kênh) thì kích thước bản đồ đặc trưng P3 là 80x80.

Đầu (Head) sử dụng 3 đầu ra có kích thước khác nhau từ cổ (neck) (P3, P4, P5) để dự đoán các vật thể nhỏ, vừa lớn. Mỗi ô trong P3 tương ứng với một vùng nhỏ hơn mỗi ô trong P5. Lại có mỗi ô trong bản đồ đặc trưng cho 1 dự đoán nên P3 dự đoán vật thể nhỏ, P5 dự đoán vật thể lớn, P4 dự đoán vật thể vừa. Thông tin về đầu ra, kiến trúc đã được mô tả ở mô-đun phát hiện.



Hình 9. Cổ (Neck) học thông tin vị trí trong giai đoạn đầu, học thông tin ngữ nghĩa trong giai đoạn sau.

3.7. Gán nhiệm vụ trong huấn luyện

Gán nhãn động Task Align Assigner được dùng để xác định 1 hộp giới hạn thật được dự đoán bởi các điểm neo nào. Quy trình được thể hiện trong hình 10.

Hàm mất mát IoU tổng thể có công thức:

$$CIoU = IoU\left(b, \hat{b}\right) - \frac{\rho^2\left(b, \hat{b}\right)}{c^2} - \alpha v \quad (1)$$

$$v = \frac{4}{\pi^2}\left(arctan\frac{w}{h} - arctan\frac{\hat{w}}{\hat{h}}\right) \quad (2)$$

$$\alpha = \frac{v}{1 - IoU\left(b, \hat{b}\right) + v} \quad (3)$$

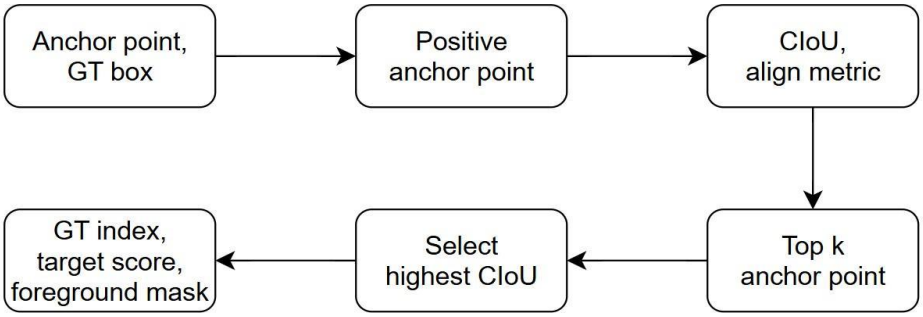
trong đó $b, \hat{b}$ là hộp giới hạn thật và hộp giới hạn dự đoán, $\rho^2(b, \hat{b})$ là khoảng cách Euclid giữa tâm 2 hộp, c là độ dài đường chéo của hộp bao phủ cả hộp dự đoán và hộp thật, v đo sự sai lệch tỷ lệ khung hình, α là hệ số tỷ lệ.

Sau đây là phần diễn giải cho hình 10. Mỗi điểm neo (mỗi ô trong bản đồ đặc trưng từ đầu vào của đầu (Head)) được xét với một hộp giới hạn thật. Điểm neo nào sinh ra hộp có tâm nằm trong hộp giới hạn thật được xác định là điểm neo dương. Từ các điểm neo dương, tính chỉ số $CIoU$ (công thức 1), $align$ (công thức 4) với hộp giới hạn thật để chọn ra k điểm neo có chỉ số $align$ cao nhất cho mỗi hộp giới hạn thật. Mỗi điểm neo chỉ nên gán cho 1 hộp giới hạn thật, do đó với điểm neo được gán cho nhiều hộp giới hạn thật, ta chọn hộp thật có $CIoU$ với hộp neo cao nhất. Kết quả cuối cùng là chỉ số của hộp giới hạn thật (truy xuất được nhãn và tọa độ hộp giới hạn), điểm dự đoán (sau này sẽ so sánh với xác suất lớp từ dự đoán của mô hình), mặt nạ vật thể dạng nhị phân (công thức 5, mặt nạ có giá trị 0 khi là nền, 1 khi là vật thể).

$$align_{ij} = clsprob_{ij}^{\alpha}.CIoU\left(b_i, \widehat{b_j}\right)^{\beta} \quad (4)$$

$$gt_{ij} = mask_j.align_{ij}.\frac{CIoU\left(b_i, \widehat{b_j}\right)^{\beta}}{align_{ij}} \quad (5)$$

trong đó $clsprob_{ij}$ là xác suất điểm neo i được chọn có lớp giống lớp của hộp giới hạn thật j, $mask_j$ là mặt nạ nhị phân để chỉ chọn ra các dự đoán được gán với hộp giới hạn thật j (sau bước lọc cuối cùng – chọn $CIoU$ cao nhất trong hình 10), gt_{ij} là target score trong hình 10.



Hình 10. Gán nhiệm vụ trong huấn luyện dựa trên cả độ khớp về lớp và độ khớp về hộp giới hạn. Từ mỗi điểm neo và hộp giới hạn thật chọn các điểm neo dương sau đó tính độ khớp. Chọn ra các điểm neo có độ khớp tốt với mỗi hợp giới hạn thật, xử lý trường hợp một điểm neo được gán nhiệm vụ dự đoán hai hộp giới hạn thật.

3.8. Hàm mất mát

Hàm mất mát lớp có công thức:

$$L_{cls} = BCE(\sigma(pred), gt).scale \quad (6)$$

$$scale = \begin{cases} \alpha[\sigma(pred)]^{\gamma} \; khi \; label = 0 \\ gt \; khi \; label = 1 \end{cases} \quad (7)$$

trong đó BCE là hàm binary cross-entropy để tính mất mát giữa dự đoán của mô hình và nhãn thật. Tham số $label = 0$ khi mô hình dự đoán đối tượng là nền, là 1 khi mô hình dự đoán đối tượng là vật thể. Trong một ảnh, số lượng vật thể thường ít, nền thường nhiều. Do đó, tham số $scale$ được sử dụng để giảm ảnh hưởng của L_{cls} khi mô hình dự đoán đối tượng là nền, nếu không mô hình chỉ cần học tốt cách xác định nền để tối thiểu hóa L_{cls}. Việc giảm L_{cls} ở nền được thực hiện dễ dàng bằng việc nhân thêm một số $\alpha \in (0,1)$ hoặc lũy thừa một số $\gamma > 1$ vì $\sigma(pred) \in (0,1]$. Trong mã nguồn, tác giả chọn $\alpha = 1, \beta = 6$. Tuy $\alpha = 1$ không giúp giảm trọng số nhưng $\beta = 6$ đã giảm trọng số của L_{cls} đi rất nhiều.

Hàm mất mát phân phối tọa độ hộp giới hạn có công thức:

$$L_{DFL} = (t_r - t)CE(p, t_l) + (1 - w_L)CE(p, t_r) \quad (8)$$

trong đó p là phân phối tọa độ hộp giới hạn dự đoán, t là khoảng cách từ điểm neo đến cạnh của hộp giới hạn thật, t_l, t_r lần lượt là bin trái, bin phải của hộp giới hạn thật (số nguyên), CE là hàm cross-entropy.

Hàm mất mát tọa độ hộp giới hạn có công thức:

$$L_{box} = \frac{1}{\sum_i align_{ij}} \sum_{i \in FG} \left(1 - CIoU\left(b_i, \widehat{b_j}\right) s_i\right) \quad (9)$$

trong đó b, $\hat{b}$ là hộp giới hạn thật và hộp giới hạn dự đoán, s_i là tổng trọng số giữa điểm khớp giữa hộp thật j và tất cả các hộp i được gán nhiệm vụ dự đoán nó (công thức (8)), FG (foreground) là tập hợp các hộp chứa vật thể.

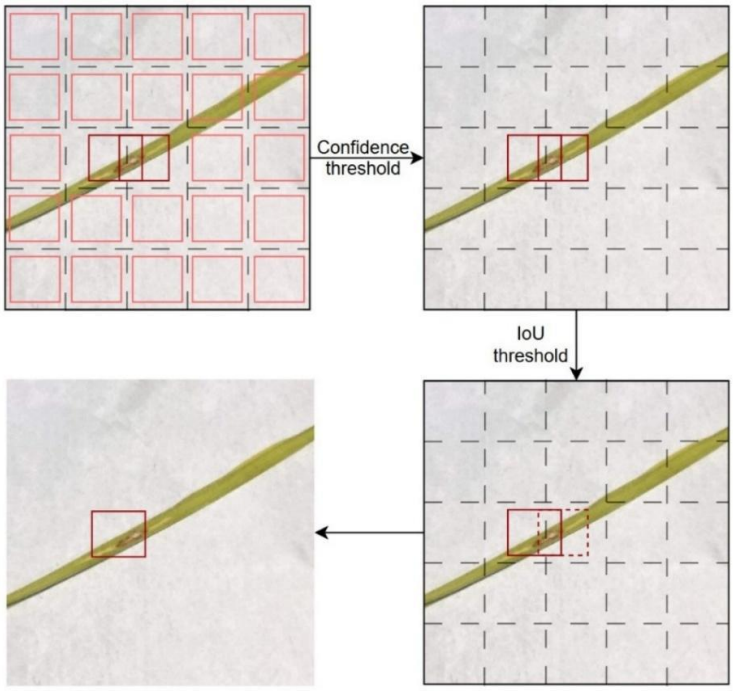
Hàm mất mát tổng thể:

$$L = w_{cls} L_{cls} + w_{box} L_{box} + w_{DFL} L_{DFL} \quad (10)$$

trong đó trọng số mặc định là $w_{cls} = 0.5, w_{box} = 7.5$ và $w_{DFL} = 1.5$ để điều chỉnh mức độ ảnh hưởng của lớp và hộp giới hạn.

3.9. Hậu xử lý

Dự đoán thô của mô hình được lọc với 2 ngưỡng: điểm tin cậy và IoU. Đầu tiên, các dự đoán có điểm tin cậy thấp hơn ngưỡng bị loại, ngưỡng mặc định là 0,25. Tiếp theo, các dự đoán trùng nhau sẽ bị loại theo quy trình sau: lấy ra dự đoán có điểm tin cậy cao nhất, loại các dự đoán có chỉ số IoU với hộp được chọn lớn hơn ngưỡng (mặc định là 0,7). Việc này được thực hiện cho đến khi không còn dự đoán nào. Quy trình được minh họa trong hình 11.



Hình 11. Hậu xử lý thực hiện hai bước: loại dự đoán có điểm tin cậy thấp và loại dự đoán trùng.

3.10. Luồng hoạt động

Luồng hoạt động của quá trình huấn luyện, suy luận được thể hiện trong các hình 12 và 13. Phần này chỉ giải thích quá trình suy luận để hiểu một phần cách mô hình quyết định (phần còn lại được giải thích bằng bản đồ nhiệt).

Đầu vào: Ảnh RGB.

Tiền xử lý: Điều chỉnh kích thước về 640x640 trong đó thêm phần đen (padding) để giữ tỷ lệ khung hình (không làm méo ảnh).

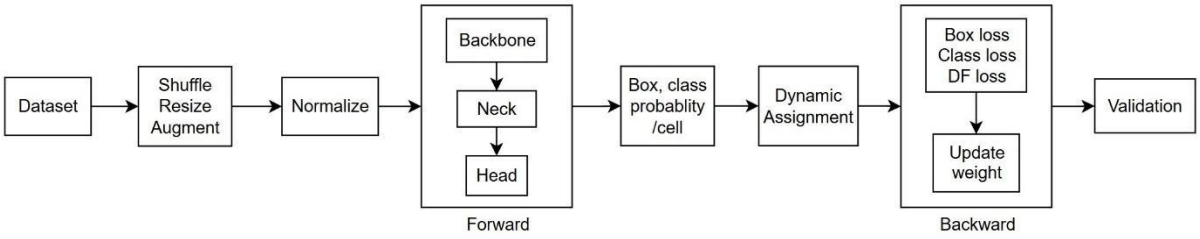
Mạng xương sống: Trích xuất đặc trưng và giảm kích thước bản đồ đặc trưng, SPPF tăng vùng nhìn. Kích thước bản đồ đặc trưng đầu ra: 20x20.

Cổ (Neck): Giai đoạn đầu lấy lại thông tin vị trí, giai đoạn sau lấy thông tin ngữ nghĩa. Ba đầu ra: 80x80, 40x40, 20x20 để dự đoán vật thể nhỏ, vừa, lớn.

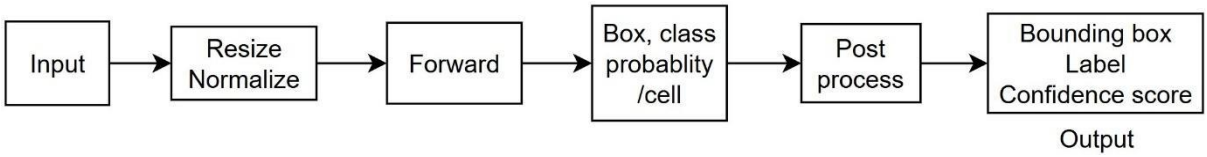
Đầu (Head): Dùng 3 đầu ra từ Cổ (Neck), mỗi ô từ 8400 ô (80.80+40.40+20.20=8400) cho 1 dự đoán gồm xác suất mỗi lớp và tọa độ hộp giới hạn. Điểm tin cậy chính là xác suất lớp có giá trị lớn nhất.

Hậu xử lý: Loại dự đoán có điểm tin cậy thấp và loại dự đoán trùng.

Đầu ra: Hộp giới hạn, lớp, điểm tin cậy.



Hình 12. Quá trình huấn luyện thực hiện: chuẩn bị dữ liệu, áp dụng các phép chuẩn hóa kèm tăng cường dữ liệu và tráo dữ liệu, đưa dữ liệu qua mô hình để mỗi ô trong bản đồ đặc trưng cho một dự đoán, so khớp với hộp giới hạn thật để gán nhiệm vụ, tính giá trị các hàm mất mát để cập nhật trọng số và kiểm thử.



Hình 13. Quá trình suy luận thực hiện: chuẩn hóa, đưa dữ liệu vào mô hình để mỗi ô trong bản đồ đặc trưng cho một dự đoán, hậu xử lý để loại dự đoán có độ tin cậy thấp và dự đoán trùng.

3.11. Giải thích mô hình quyết định

GradCAM là bản đồ kích hoạt lớp dựa trên gradient. Nó tính mức độ đóng góp của từng điểm trong ảnh đầu vào bằng cách lan truyền ngược từ lớp dự đoán đến lớp tích chập.

Với mỗi kênh A^k trong lớp tích chập, GradCAM tính độ quan trọng của từng kênh bằng việc lan truyền ngược gradient từ đầu ra lớp quan tâm Y_c về từng điểm ảnh trong kênh A^k rồi lấy giá trị trung bình:

$$\alpha_k = \frac{1}{H * W} \Sigma_i \Sigma_j \frac{\partial Y_c}{\partial A_{ij}^k} \quad (11)$$

trong đó H, W là chiều cao, chiều rộng của bản đồ đặc trưng, A_{ij}^k là giá trị tại vị trí (i, j) của kênh đặc trưng thứ k, α_k là mức độ quan trọng của kênh thứ k đối với quyết định dự đoán đầu ra Y_c.

Bản đồ nhiệt GradCAM ban đầu là trung bình có trọng số của các kênh trong lớp tích chập:

$$GradCAM_0 = \Sigma_k \alpha_k A^k \quad (12)$$

Bản đồ nhiệt GradCAM cuối cùng chỉ giữ lại những điểm có ảnh hưởng tích cực đến dự đoán của mô hình:

$$GradCAM = ReLU(\Sigma_k \alpha_k A^k) \quad (13)$$

HiResCAM là một phương pháp giải thích khác để khắc phục 2 nhược điểm của GradCAM. Thứ nhất, lớp tích chập cuối cùng tuy giàu thông tin ngữ nghĩa nhưng lại nghèo thông tin vị trí vì kích thước bản đồ đặc trưng quá nhỏ. Do đó, HiResCAM cải tiến bằng cách tính bản đồ nhiệt cho nhiều lớp tích chập rồi thay đổi về kích thước ảnh đầu vào và lấy trung bình. Thứ hai, việc nén độ quan trọng của một kênh thành một số thực α_k khiến mọi thông tin trong kênh A^k quan trọng như nhau. Do đó, HiResCAM cải tiến bằng cách giữ nguyên độ quan trọng của kênh A^k là một mảng 2 chiều:

$$gradient_k = \frac{\partial Y_c}{\partial A^k} \quad (14)$$

$$HiResCAM = ReLU(\Sigma_k gradient_k \odot A^k) \quad (15)$$

4. Thực nghiệm

4.1. Độ đo

Độ chính xác được là tỷ lệ chính xác của mô hình trên tất cả dự đoán. Độ phủ là tỷ lệ dương tính mà mô hình tìm ra trên tất cả các mẫu dương tính thực tế.

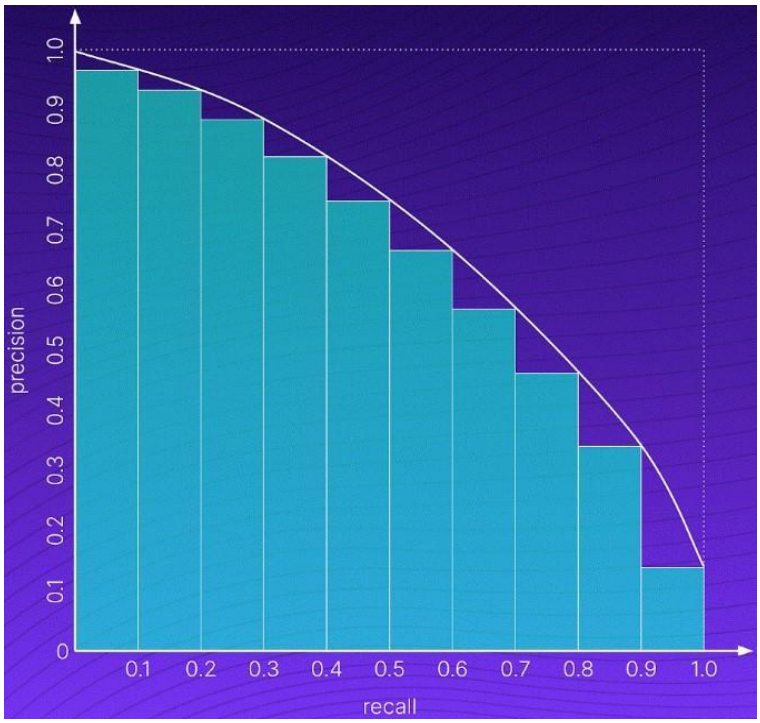
$$P = \frac{TP}{TP + FN} \quad (16), \qquad R = \frac{TP}{TP + FP} \quad (17)$$

trong đó kí tự đầu tiên là T (True) hoặc F (False) thể hiện mô hình dự đoán đúng hoặc sai, kí tự thứ hai là P (Positive) hoặc N (Negative) thể hiện đối tượng mà mô hình dự đoán trong thực tế là vật thể hoặc nền.

AP (Average precision) là trung bình độ chính xác, tính độ chính xác trung bình trên toàn ngưỡng tin cậy (từ 0 đến 1) trong biểu đồ độ chính xác - độ phủ (minh họa trong hình 14). Độ đo này được dùng với một lớp c, một ngưỡng IoU cụ thể. $AP^c@K$ ràng buộc một dự đoán là đúng chỉ khi hộp dự đoán có chỉ số IoU với hộp giới hạn thật lớn hơn ngưỡng K. Công thức:

$$AP^c@K = \frac{\Sigma_{conf=0}^{1} P_{conf}^{c}}{Z} \tag{18}$$

trong đó $conf$ là ngưỡng tin cậy P_{conf}^{c} là độ chính xác của lớp c tại ngưỡng tin cậy $conf$, Z là số ngưỡng tin cậy được tính (ví dụ trong hình 14 số ngưỡng tin cậy là 11).



Hình 14: Độ chính xác trung bình được tính dựa trên biểu đồ độ chính xác - độ phủ.

Trung bình độ chính xác trên toàn bộ lớp là:

$$mAP@K = \frac{\Sigma_c AP^c@K}{nc} \tag{19}$$

$$mAP@K_1 - K_2 = \frac{\Sigma_{k=K_1}^{K_2} mAP@K}{Z} \tag{20}$$

trong đó $mAP@K$ trong công thức (19) là trung bình độ chính xác $AP^c@K$ trên toàn bộ lớp, $mAP@K_1 - K_2$ trong công thức (20) là trung bình độ chính xác trên các ngưỡng IoU từ K_1 đến K_2, nc là số lớp, Z là số ngưỡng IoU được tính.

Trong bài toán phát hiện vật thể nói chung, các ngưỡng tin cậy được quan tâm là 0.5 và 0.95 ứng với hai độ đo là $mAP@0.5$ và $mAP@0.5 - 0.95$. Trong các phần tiếp theo, chúng tôi sử dụng một kí hiệu khác của hai độ đo trên là $mAP50$ và $mAP50 - 95$.

Ngoài ra, người ta còn dùng ma trận nhầm lẫn được sử dụng để trực quan hóa mối quan hệ giữa kết quả dự đoán và kết quả đúng, chỉ số $fitness$ để đánh giá hiệu suất của mô hình (chọn ra file trong số tốt nhất dựa trên chỉ số này trên tập kiểm định):

$$fitness = 0.9mAP_{50-95} + 0.1mAP_{50} \quad (21)$$

trong đó mAP_{50-95}, mAP_{50} được viết với ngưỡng IoU ở dưới để công thức dễ nhìn hơn.

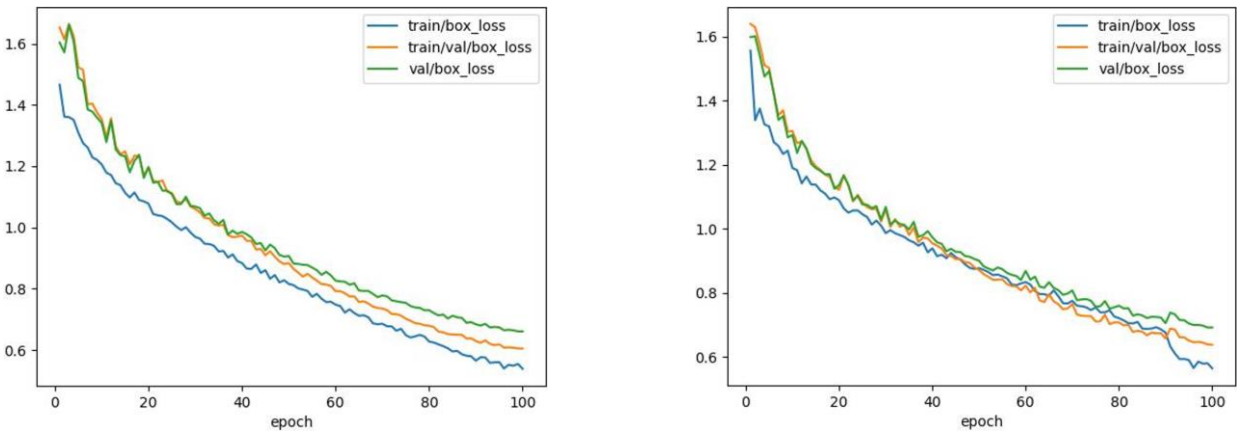
4.2. Chiến lược huấn luyện

Chúng tôi thực nghiệm trên bộ dữ liệu cây cà phê để đánh giá ảnh hưởng của kích thước đầu vào, phép biến đổi mosaic lên độ đo. Với kích thước đầu vào, nhóm thử nghiệm huấn luyện với các kích thước khác nhau của ảnh đầu vào trong cùng thời gian huấn luyện và đo các chỉ số trên tập kiểm thử. Độ đo tốt nhất đạt tại kích thước 640x640, kích thước này sẽ được sử dụng trong huấn luyện và kiểm thử. Chi tiết kết quả trong bảng 3.

Bảng 3. Ảnh hưởng của kích thước đầu vào đến kết quả phát hiện đốm bệnh.

Kích thước	**320**	**640**	**960**
Thời gian (giây)	7557	7488	7526
Số epoch	100	100	56
Độ chính xác	0,935	0,984	0,978
Độ phủ	0,876	0,964	0,973
mAP50	0,926	0,985	0,989
mAP50-95	0,758	0,875	0,856
Fitness	0,775	0,886	0,869

(A) **(B)**



Hình 15. Chỉ số box_loss. (A) Không áp dụng mosaic; **(B)** Áp dụng mosaic.

Với phép biến đổi mosaic, nhóm thực nghiệm và so sánh kết quả khi áp dụng mosaic cho 90% quá trình huấn luyện và không áp dụng mosaic trong huấn luyện. Cụ thể, nhóm so sánh chỉ số box_loss trên 3 tập: tập huấn luyện có tăng cường dữ liệu, tập huấn luyện

gốc, tập kiểm định. Để đơn giản, chúng tôi thực hiện với kích thước ảnh đầu vào là 320x320. Kết quả được trực quan hóa ở hình 15.

Điểm chung là mất mát trên tập huấn luyện có tăng cường dữ liệu (train/box_loss) thấp nhất vì mô hình học trên dữ liệu này, mất mát trên tập huấn luyện gốc thấp thứ hai vì đây là biến thể của tập dữ liệu được học, mất mát trên tập kiểm định thấp nhất vì mô hình chưa từng học qua.

Mất mát trên tập huấn luyện có tăng cường dữ liệu giảm đột ngột ở epoch 90 chúng tôi tắt mosaic ở 10 epoch cuối, khiến mô hình đang học các ảnh khó của mosaic đột ngột trở về học các ảnh dễ của tập huấn luyện gốc. Tuy chỉ số box loss khi mosaic cao hơn (tệ hơn), nhưng các chỉ số độ đo trên tập đánh giá khi mosaic lại tốt hơn vì mô hình học được vật thể nhỏ, cũng như điều kiện phức tạp, xem bảng 4.

Bảng 4: Ảnh hưởng của mosaic lên độ đo: huấn luyện với mosaic cho kết quả tốt hơn.

	Độ chính xác	**Độ phủ**	**mAP50**	**mAP50-95**
Có mosaic	0,935	0,876	0,926	0,758
Không mosaic	0,922	0,871	0,925	0,742

Tổng kết, chiến lược thực nghiệm là áp dụng các phép biến đổi đã nêu trong phần đầu của Phương pháp với kích thước đầu vào 640x640, trong đó áp dụng mosaic trong 90% quá trình huấn luyện.

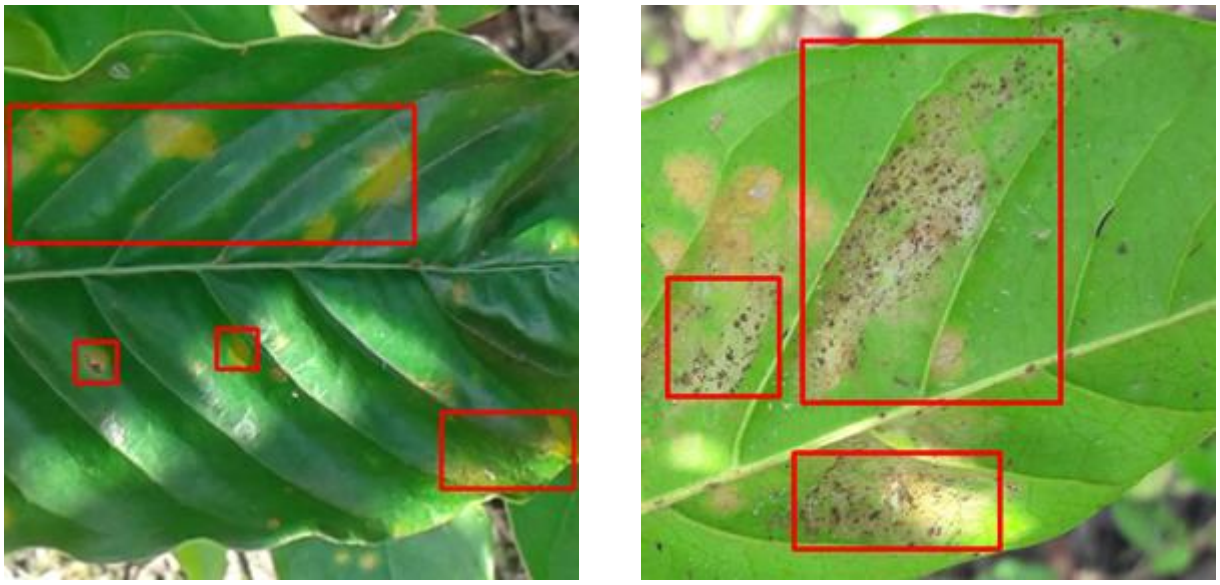
4.3. Cây cà phê

Bộ dữ liệu gồm 7024 ảnh lá cày phê từ Roboflow với 4 bệnh: đốm mắt nâu, sâu vẽ bùa, gỉ sắt, rệp nhện đỏ. Tỷ lệ huấn luyện : kiểm định : đánh giá là 81:13:6. Triệu ứng của 4 bệnh được mô tả trong bảng 5 kèm minh họa tương ứng trong hình 16. Dữ liệu được lấy trong điều kiện ánh sáng đa dạng.

Kích thước đốm bệnh được chúng tôi định nghĩa là căn bậc hai của diện tích hộp giới hạn bao quanh đốm bệnh (đơn vị pixel), tính trong ảnh sau resize 640x640 (giữ tỷ lệ ảnh, padding kích thước ngắn). Số lượng và kích thước đốm bệnh được thống kê trong bảng 6, bảng 7. Mặc dù có sự chênh lệch về số lượng đốm bệnh theo lớp nhưng điều này phản ánh đúng tính chất của từng lớp: lớp có nhiều đốm bệnh thì kích thước đốm bệnh nhỏ, khó học; lớp có ít đốm bệnh thì ngược lại.

(A) **(B)**

(C) **(D)**



Hình 16. Triệu chứng các bệnh lá cà phê. (A) Đốm mắt nâu, (B) Sâu vẽ bùa, (C) Gỉ sắt, (D) Rệp nhện đỏ.

Bảng 5. Triệu chứng các bệnh lá cà phê.

Bệnh	Triệu chứng
Đốm mắt nâu	Các đốm nâu: đậm ở ngoài, nhạt ở trong, hình dạng giống con mắt
Sâu vẽ bùa	Các mảng lớn màu nâu trên lá
Gỉ sắt	Các vết màu vàng trên lá
Rệp nhện đỏ	Tơ nhện mỏng kèm nhiều chấm đen

Bảng 6. Số lượng đốm bệnh trên tập dữ liệu lá cà phê.

Tập dữ liệu	Huấn luyện	Kiểm định	Đánh giá
Đốm mắt nâu	10271	1030	502
Sâu vẽ bùa	7977	1389	763
Gỉ sắt	13570	1638	785
Rệp nhện đỏ	3391	375	369

Bảng 7. Kích thước đốm bệnh trên tập dữ liệu lá cà phê (pixel).

Tập dữ liệu	Huấn luyện	Kiểm định	Đánh giá
Đốm mắt nâu	20,13	18,98	18,88
Sâu vẽ bùa	96,32	88,05	82,82
Gỉ sắt	26,51	31,32	26,25
Rệp nhện đỏ	98,10	96,85	70,82

Bảng 8. Các độ đo trên tập dữ liệu lá cà phê của YOLOv8.

Lớp	Độ chính xác	Độ phủ	mAP50	mAP50-95
Đốm mắt nâu	0,978	0,906	0,957	0,795
Sâu vẽ bùa	0,997	0,999	0,995	0,98
Gỉ sắt	0,975	0,936	0,988	0,869
Rệp nhện đỏ	0,998	1	0,995	0,982
Trung bình	0,987	0,96	0,984	0,906

Bảng 9. Các độ đo trên tập dữ liệu lá cà phê của YOLOv13.

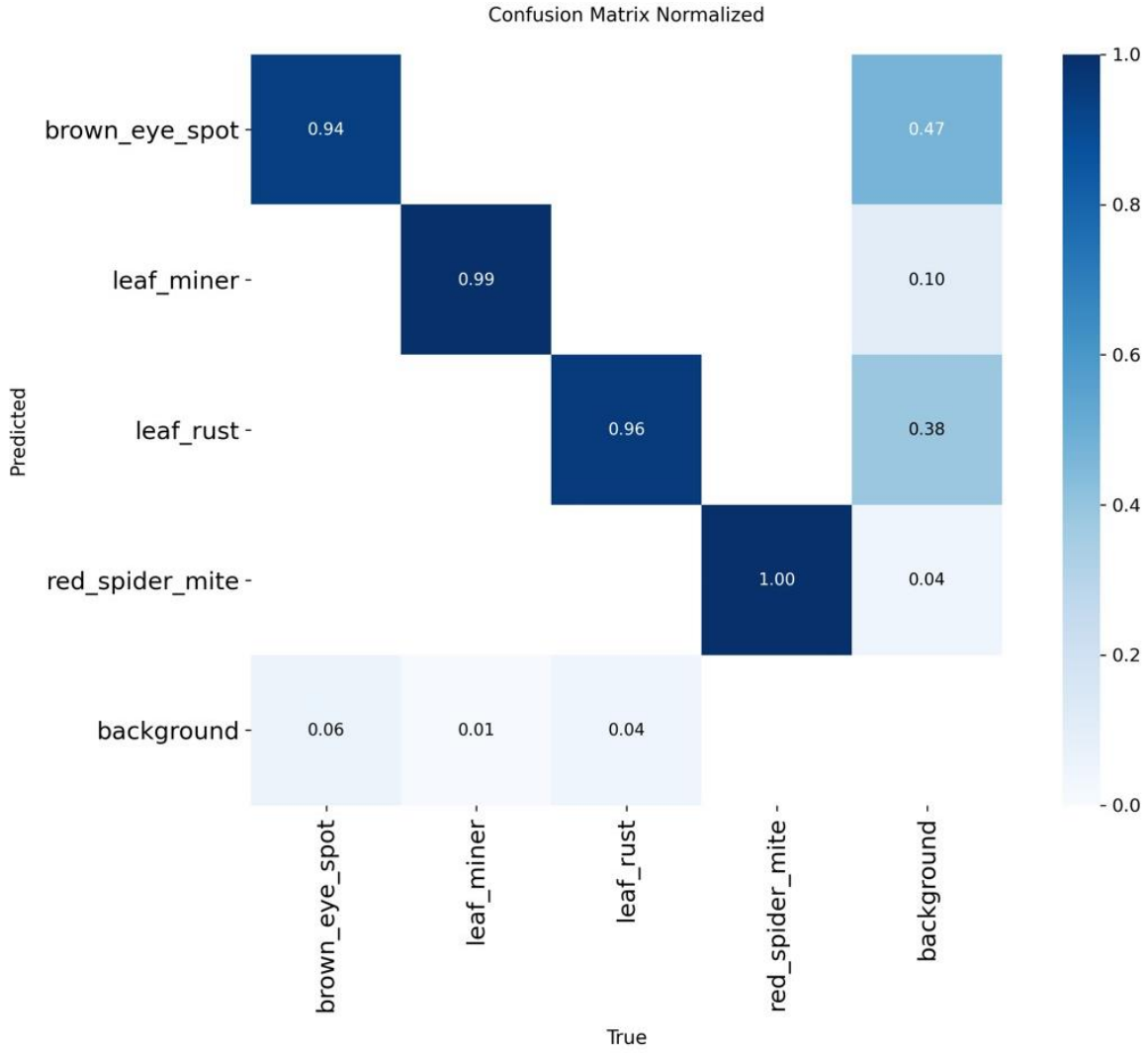
Lớp	Độ chính xác	Độ phủ	mAP50	mAP50-95
Đốm mắt nâu	0,93	0,828	0,908	0,632
Sâu vẽ bùa	0,984	0,983	0,993	0,891
Gỉ sắt	0,912	0,86	0,95	0,629
Rệp nhện đỏ	0,976	0,991	0,995	0,93
Trung bình	0,95	0,916	0,962	0,77

Kết quả các độ đo khi sử dụng YOLOv8 và YOLOv13 được mô tả trong bảng 8 và 9. Để so sánh 2 phương pháp trên bộ dữ liệu thực nghiệm, chúng tôi dùng 2 độ đo là mAP50 và mAP50-95. Kết quả cho thấy YOLOv8 tốt hơn ở cả 2 chỉ số này, đặc biệt là chỉ số mAP50-95 phản ánh mức độ định vị chính xác.

Với YOLOv8, hầu hết các chỉ số độ đo đều trên 0.9, ngoại trừ mAP50-95 của bệnh đốm mắt nâu và bệnh gỉ sắt, không có sự chênh lệch đáng kể về độ đo giữa các lớp. Ma

trận nhầm lẫn được thể hiện trong hình 17. Trung bình mô hình dự đoán đúng 97.25% bệnh, số lượng dương tính giả trung bình là 24.75%, số lượng âm tính giả trung bình là 2.75%. Đây là kết quả tốt, tuy nhiên số lượng dương tính giả còn nhiều.

FPS đo được trên kích thước đầu vào 640x640, trên máy cấu hình RTX 3050 với 4GB VRAM là 54.3, thích hợp để chạy thời gian thực, có độ trễ nhẹ với video 60 FPS.



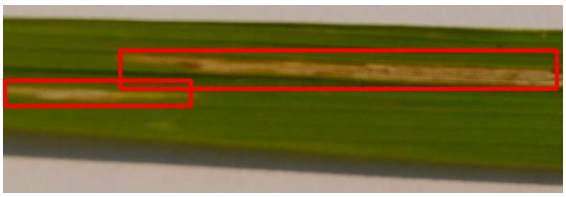
Hình 17. Ma trận nhầm lẫn thể hiện mô hình dự đoán với độ chính xác cao, có một phần nhỏ nhầm lẫn bệnh với nền.

4.4. Cây lúa

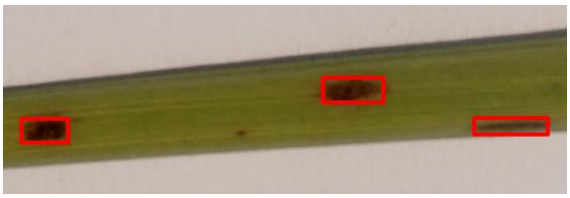
Bộ dữ liệu gồm 6715 ảnh lá lúa từ Roboflow với 3 bệnh: bạc lá, đốm nâu, muội than. Tỷ lệ huấn luyện: kiểm định: đánh giá là 90:6:4. Triệu ứng của 3 bệnh được mô tả trong bảng 10 kèm minh họa tương ứng trong hình 18. Dữ liệu được lấy trong điều kiện ánh sáng đa dạng.

Mặc dù có sự chênh lệch về số lượng đốm bệnh theo lớp (bảng 11) nhưng kích thước đốm bệnh đều lớn (bảng 12) dẫn đến mô hình dễ học, độ đo không chênh lệch đáng kể.

(A)



(B)



(C)

Hình 18. Triệu chứng các bệnh lá lúa. (A) Bạc lá; **(B)** Đốm nâu; **(C)** Muội than.

Bảng 10. Triệu chứng các bệnh lá lúa.

Bệnh	Triệu chứng
Bạc lá	Các vệt dài màu nâu nhạt hoặc trắng
Đốm nâu	Các vết màu nâu đậm hoặc đen
Muội than	Các vết màu trắng dẹt ở hai đầu, rìa màu vàng

Bảng 11. Số lượng đốm bệnh trên tập dữ liệu lá lúa.

Tập	Huấn luyện	Kiểm định	Đánh giá
Bạc lá	2907	185	125
Đốm nâu	12667	719	606
Muội than	6342	394	252

Bảng 12. Kích thước đốm bệnh trên tập dữ liệu lá lúa (pixel).

Tập	Huấn luyện	Kiểm định	Đánh giá
Bạc lá	314,63	274,32	255,01
Đốm nâu	47,67	49,96	46,12
Muội than	110,41	103,34	108,72

Kết quả các độ đo khi sử dụng YOLOv8 và YOLOv13 trong bảng 13 và 14. YOLOv8 tốt hơn ở cả 2 chỉ số mAP50 và mAP50-95, đặc biệt là chỉ số mAP50-95 phản ánh mức độ định vị chính xác.

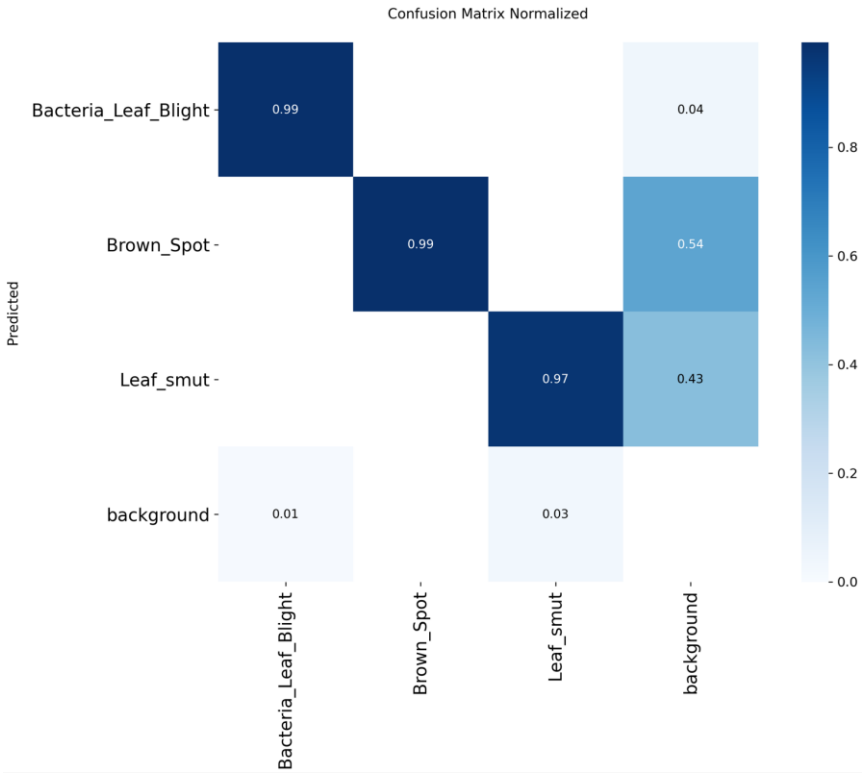
Với YOLOv8, hầu hết các chỉ số độ đo đều trên 0.9, ngoại trừ mAP50-95 của bệnh đốm nâu, không có sự chênh lệch đáng kể về độ đo giữa các lớp. Ma trận nhầm lẫn được thể hiện trong hình 19. Trung bình mô hình dự đoán đúng 98.33% bệnh, số lượng dương tính giả trung bình là 33.67%, số lượng âm tính giả trung bình là 1.33%. Tương tự với bộ dữ liệu cây cà phê, đây là kết quả tốt, tuy nhiên số lượng dương tính giả còn nhiều.

Bảng 13. Các độ đo trên tập dữ liệu lá lúa của YOLOv8.

Lớp	Độ chính xác	Độ phủ	mAP50	mAP50-95
Bạc lá	0,989	0,984	0,983	0,941
Đốm nâu	0,992	0,974	0,993	0,868
Muội than	0,924	0,948	0,973	0,9
Trung bình	0,968	0,969	0,983	0,903

Bảng 14. Các độ đo trên tập dữ liệu lá lúa của YOLOv13.

Lớp	Độ chính xác	Độ phủ	mAP50	mAP50-95
Bạc lá	0,976	0,984	0,979	0,861
Đốm nâu	0,927	0,883	0,955	0,686
Muội than	0,871	0,96	0,948	0,784
Trung bình	0,925	0,942	0,961	0,777

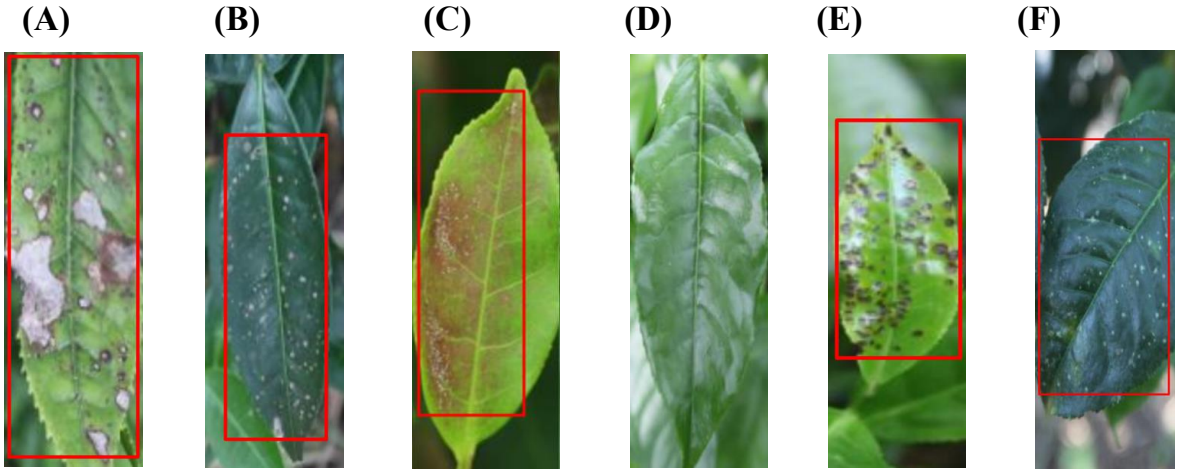


Hình 19. Ma trận nhầm lẫn cho thấy độ chính xác cao trong dự đoán kèm một phần nhầm lẫn giữa bệnh và nền.

4.5. Cây chè

Bộ dữ liệu lá chè gốc lấy từ Roboflow, được chúng tôi chỉnh sửa lại (loại 2 lớp ít mẫu, hoán đổi tập kiểm định và tập đánh giá để đảm bảo số ảnh tập kiểm định nhiều hơn). Bộ dữ liệu sau chỉnh sửa gồm 9579 ảnh lá chè với một lớp khỏe mạnh và 5 lớp bệnh: thối đen, gỉ sắt, nhện đỏ, bọ muỗi, đốm trắng. Tỷ lệ huấn luyện : kiểm định : đánh giá là 76:16:8. Triệu ứng của 5 bệnh được mô tả trong bảng 15 kèm minh họa tương ứng trong hình 20. Dữ liệu được lấy ở ngoài trời trong điều kiện sáng vừa phải.

Mặc dù có sự chênh lệch về số lượng đốm bệnh theo lớp (bảng 16) nhưng kích thước đốm bệnh đều lớn (bảng 17) dẫn đến mô hình dễ học, độ đo không chênh lệch đáng kể.



Hình 20. Ảnh minh họa 5 lá chè bệnh và một lá chè khỏe mạnh. (A) Thối đen; **(B)** Gỉ sắt; **(C)** Nhện đỏ; **(D)** Khỏe mạnh; **(E)** Bọ muỗi; **(F)** Đốm trắng.

Bảng 15. Triệu chứng các bệnh lá chè.

Bệnh	Triệu chứng
Thối đen	Các vùng xám, có màu giống giấy bị cháy
Gỉ sắt	Các chấm nhỏ màu trắng có lớp rìa ngoài vàng nhạt, thường gồm nhiều chấm nhỏ trên lá
Nhện đỏ	Tơ nhện kèm theo vệt đỏ (khác với chấm đen của cà phê)
Bọ muỗi	Các đốm tròn màu đen
Đốm trắng	Các chấm nhỏ màu trắng, thường gồm nhiều chấm nhỏ trên lá

Bảng 16. Số lượng đốm bệnh trên tập dữ liệu lá chè.

Tập dữ liệu	Huấn luyện	Kiểm định	Đánh giá
Thối đen	92	30	6
Gỉ sắt	2749	616	20
Nhện đỏ	735	212	75
Khỏe mạnh	5501	1168	585
Bọ muỗi	299	71	57
Đốm trắng	240	45	39

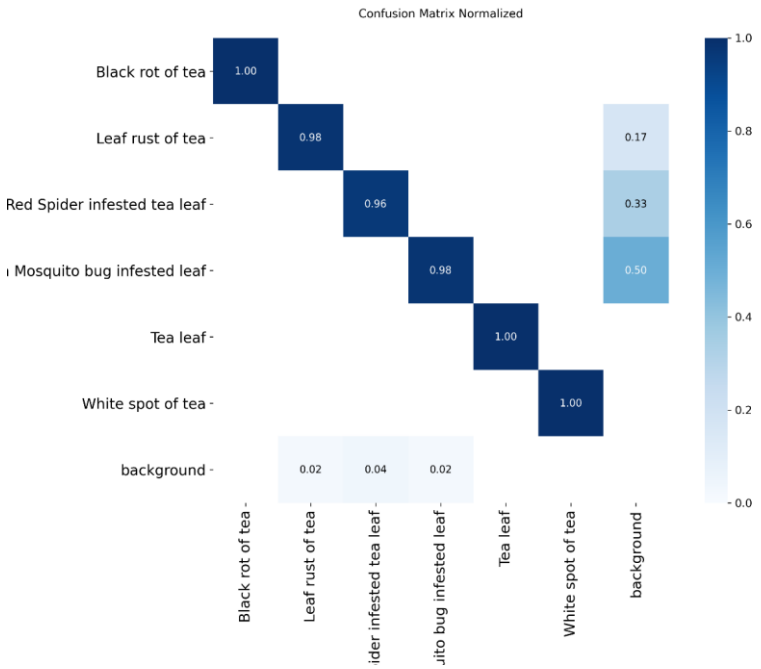
Bảng 17. Kích thước đốm bệnh trên tập dữ liệu lá chè (pixel).

Tập dữ liệu	Huấn luyện	Kiểm định	Đánh giá
Thối đen	262,69	263,40	266,63
Gỉ sắt	239,95	238,57	241,70
Nhện đỏ	271,33	278,64	275,44
Khỏe mạnh	190,82	187,33	188,76
Bọ muỗi	311,41	319,37	315,36
Đốm trắng	378,77	389,70	386,73

Kết quả các độ đo khi sử dụng YOLOv8 và YOLOv13 được mô tả trong bảng 18 và 19. Kết quả cho thấy YOLOv8 tốt hơn ở 2 chỉ số mAP50 và mAP50-95, ngoại trừ mAP50-95 của 2 lớp bệnh thối đen và đốm trắng.

So với 2 tập lá lúa và lá chè, YOLOv8 cho thấy sự sụt giảm hiệu suất còn YOLOv13 vẫn học khá tốt, khoảng cách giữa YOLOv8 và YOLOv13 đã được thu hẹp đáng kể. Với YOLOv8, ngoại trừ chỉ số mAP50-95 đánh giá độ chính xác khi độ trùng khớp cao thì các chỉ số còn lại đều trên 0.9. Không có sự chênh lệch đáng kể về độ đo giữa các lớp, mô hình học tốt.

Ma trận nhầm lẫn được thể hiện trong hình 21. Trung bình mô hình dự đoán đúng 98.67% bệnh, số lượng dương tính giả trung bình là 16.67%, số lượng âm tính giả trung bình là 1.33%. Khác với bộ dữ liệu lá cà phê và bộ dữ liệu lá lúa, đây là kết quả tốt, số lượng dương tính giả rất ít.



Hình 21.. Ma trận nhầm lẫn cho thấy dự đoán của mô hình có độ chính xác gần như tuyệt đối.

Bảng 18. Các độ đo trên tập dữ liệu lá chè của YOLOv8.

Lớp	Độ chính xác	Độ phủ	mAP50	mAP50-95
Thối đen	0,983	1	0,995	0,677
Gỉ sắt	0,988	0,975	0,992	0,717
Nhện đỏ	0,977	0,906	0,986	0,794
Bọ muỗi	0,973	0,96	0,985	0,698
Khỏe mạnh	0,994	1	0,995	0,906
Đốm trắng	0,993	1	0,995	0,728
Trung bình	0,985	0,973	0,991	0,753

Bảng 19. Các độ đo trên tập dữ liệu lá chè của YOLOv13.

Lớp	Độ chính xác	Độ phủ	mAP50	mAP50-95
Thối đen	0,929	1	0,995	0,758
Gỉ sắt	0,919	0,979	0,979	0,716
Nhện đỏ	0,916	0,96	0,968	0,745
Bọ muỗi	0,989	0,969	0,979	0,672
Khỏe mạnh	0,993	1	0,995	0,74
Đốm trắng	0,997	0,923	0,945	0,762
Trung bình	0,957	0,972	0,977	0,732

4.6. Diễn giải kết quả bằng GradCAM và HiResCAM

Trong phần này, chúng tôi tạo heatmap cho tất cả các dự đoán (không tạo riêng theo từng bệnh) và đánh giá chất lượng 2 phương pháp giải thích theo 2 tiêu chí: định tính và định lượng. Qua thực nghiệm, việc tạo heatmap cho các lớp (trong mô hình, phân biệt với lớp bệnh) khác nhau cho kết quả rất khác nhau. Do đó, với mỗi phương pháp giải thích, chúng tôi chọn ra những lớp cho heatmap tốt nhất (vùng nóng trên heatmap nằm trong vùng ground truth nhiều nhất) để đánh giá. Lớp tối ưu cho định lượng là lớp cho heatmap tốt nhất (đo bằng energy sẽ giới thiệu sau), 3 lớp tối ưu cho định tính vừa phải cho heatmap tốt, vừa phải bao gồm lớp nông và lớp sâu để thể hiện quá trình thay đổi vùng chú ý của mô hình, thông số cụ thể trong bảng 20.

Bảng 20. Lớp tối ưu cho định tính và định lượng của 2 phương pháp diễn giải.

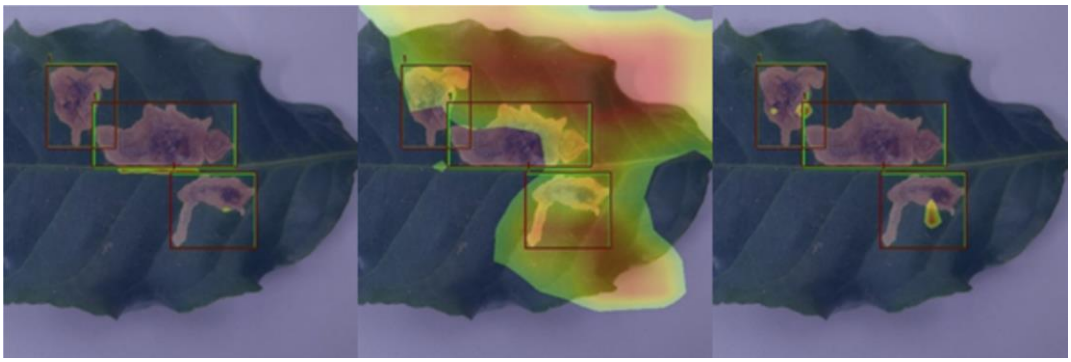
Phương pháp	Định tính	Định lượng
GradCAM	3,9,18	18
HiResCAM	4,9,18	15

Ngoài ra, để heatmap không bị chú ý phân tán, chúng tôi xóa heatmap ở những vùng có độ chú ý thấp (giá trị độ xám trong heatmap nhỏ hơn 128). Cách làm trên được áp dụng cho cả đánh giá định tính và định lượng.

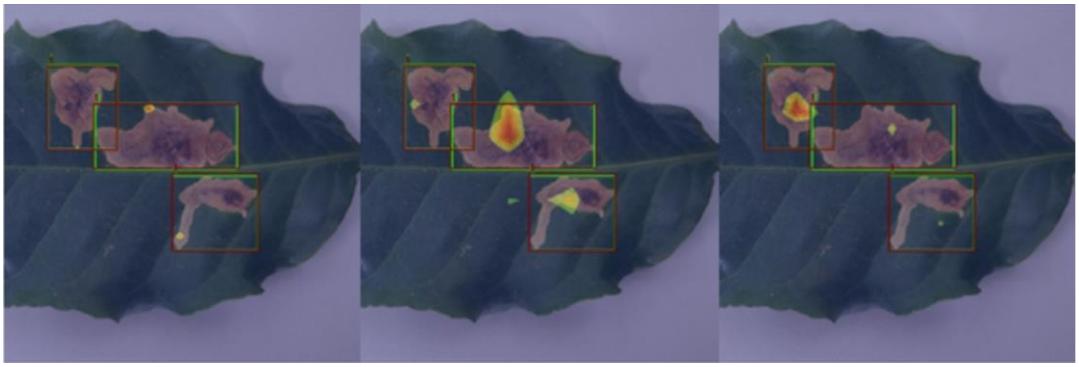
Kết quả định tính được trực quan hóa trong các hình 22, 23 và 24 (kết quả của GradCAM là 3 ảnh trên, HiResCAM là 3 ảnh dưới trong mỗi hình) trong đó mô hình chú ý nhiều nhất vào vùng màu đỏ, tiếp theo đến vàng, đến xanh. Các khung màu xanh là vùng mô hình dự đoán, các khung màu nâu là vùng đốm bệnh được gán nhãn, con số trên đầu mỗi khung là lớp bệnh. Mỗi hình thể hiện quá trình thay đổi vùng chú ý của mô hình khi diễn giải bằng GradCAM và HiResCAM. Điểm chung của 2 phương pháp diễn giải là vùng chú ý có tỷ lệ giao trên hợp với vùng dự đoán lớn hơn 0. Càng về lớp cuối thì tỷ lệ này thường lớn hơn. Với các lớp đầu, mô hình chú ý nhiều vào một số điểm biên cạnh. Với các lớp giữa, mô hình mở rộng khu vực chú ý. Với các lớp cuối, vùng chú ý được co lại, tập trung vào các vùng dự đoán. Điều này chứng minh mô hình biết nhìn vào đúng vùng, độ tin cậy được đảm bảo.

Điểm khác là vùng chú ý của mô hình khi diễn giải bằng GradCAM kém tập trung vào vùng dự đoán hơn so với HiResCAM. Khi diễn giải bằng GradCAM, mô hình chú ý nhiều hơn nhưng không chất lượng. Do đó, về mặt định tính, HiResCAM diễn giải tốt hơn so với GradCAM.

(A)

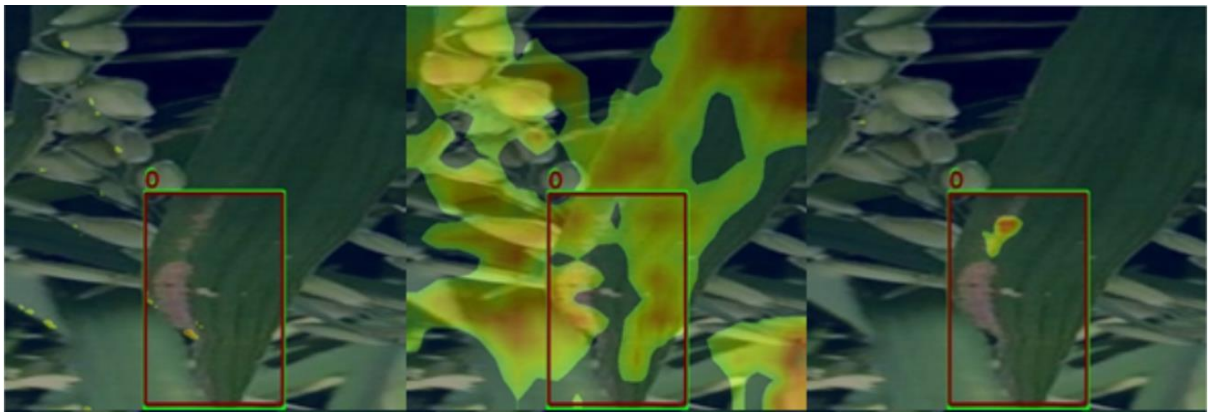


(B)



Hình 22. Heatmap trên ảnh lá cà phê: các khung màu nâu là ground truth, các khung màu xanh là dự đoán, số trên mỗi khung là lớp bệnh (số 1 trong ảnh tương ứng với bệnh sâu vẽ bùa). (A) GradCAM; **(B)** HiResCAM.

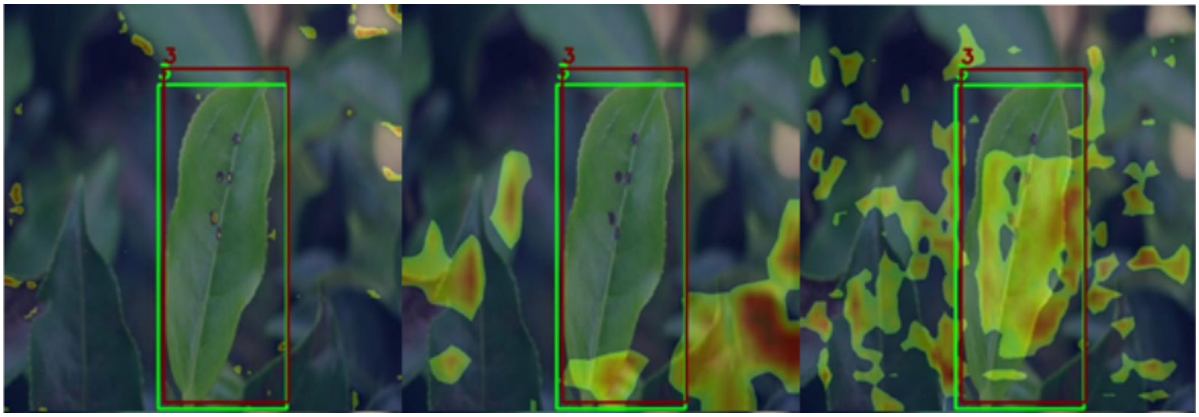
(A)



(B)

Hình 23. Heatmap trên ảnh lá lúa dùng: các khung màu nâu là ground truth, các khung màu xanh là dự đoán, số trên mỗi khung là lớp bệnh (số 0 trong ảnh tương ứng với bệnh bạc lá). (A) GradCAM; **(B)** HiResCAM.

(A)



(B)

Hình 24. Heatmap trên ảnh lá chè: các khung màu nâu là ground truth, các khung màu xanh là dự đoán, số trên mỗi khung là lớp bệnh (số 3 trong ảnh tương ứng với bệnh bọ muỗi). (A) GradCAM; **(B)** HiResCAM.

Tóm lại, các phân tích trên cho thấy mô hình biết nhìn vùng đúng vùng (mặc dù còn phân tán) nhưng đã giúp người dùng cuối và chuyên gia nông nghiệp biết được mô hình chú ý vào đâu trong quá trình đưa ra quyết định, qua đó độ tin cậy về mặt định tính được đảm bảo. Về mặt định lượng, chúng tôi sử dụng 2 độ đo là pointing game [29] và energy [30].

Độ đo pointing game xác định vùng nóng nhất trên heatmap có thuộc đối tượng quan tâm không, nếu thuộc thì tính là Hit, ngược lại là Miss. Tuy nhiên, trong bài này, chúng tôi sử dụng HiResCAM sinh ra cho toàn bộ lớp (không chỉ cho 1 lớp cụ thể). Do đó, pointing game cũng được tính cho toàn bộ lớp. Cụ thể, chúng tôi tính pointing game (PG) trên toàn bộ ảnh trong tập đánh giá:

$$PG = \frac{Hit}{Hit + Miss} \tag{22}$$

Độ đo energy đo năng lượng heatmap trong vùng đối tượng (ground truth) trên tổng năng lượng heatmap:

$$E = \frac{E(box_{gt})}{E(heatmap)} \tag{23}$$

Tương tự pointing game, độ đo này được tính trên toàn bộ lớp. Kết quả là trung bình chỉ số energy trên toàn bộ ảnh trong tập đánh giá.

Bảng 21 thể hiện độ đo pointing game và energy cho 3 bộ dữ liệu: lá cà phê, lá lúa, lá chè. Cả hai chỉ số khi diễn giải bằng GradCAM đều thấp hơn HiResCAM chứng tỏ heatmap tạo bằng HiResCAM cho vùng chú ý tốt hơn, tập trung hơn như đã phân tích ở phần định tính.

Bảng 21. Độ đo pointing game, energy cho 2 phương pháp giải thích với từng bộ dữ liệu: HiResCAM cho kết quả tốt hơn.

Độ đo	Phương pháp	Cà phê	Lúa	Chè	Trung bình
Pointing game (PG)	GradCAM	0,28	0,29	0,23	0,27
	HiResCAM	0,89	0,94	0,83	0,89
Energy (E)	GradCAM	0,17	0,22	0,20	0,20
	HiResCAM	0,82	0,93	0,82	0,86

Kết quả 2 độ đo pointing game, energy khi diễn giải bằng HiResCAM đạt 0.89 và 0.86 (rất cao) đã chứng minh độ tin cậy khi diễn giải mô hình bằng phương pháp này. Ngoài ra, các chỉ số độ đo khi diễn giải (pointing game, energy) cũng phản ánh đúng chỉ số độ đo mAP50-95 khi phát hiện đốm bệnh (tỷ lệ thuận), cụ thể là mô hình học tốt nhất trên tập lá lúa, lá cà phê, cuối cùng là tập lá chè.

5. Kết luận

Nghiên cứu này đã giải quyết bài toán phát hiện bệnh tự động trên lá cây trồng chiến lược tại Việt Nam. Để đối phó với các thách thức thực tiễn như sự đa dạng về đốm bệnh, bối cảnh phức tạp và điều kiện chụp ảnh không đồng nhất. Chúng tôi đã đề xuất, huấn luyện và đánh giá một mô hình dựa trên kiến trúc YOLOv8. Mô hình đã đạt được hiệu suất vượt trội trên ba bộ dữ liệu cây trồng, các chỉ số trung bình mAP50 cao và nhất quán: 0,984 trên cây cà phê, 0,983 trên cây lúa và 0,991 trên cây chè. Đồng thời, chúng tôi đã thử nghiệm và khám phá ra kỹ thuật tăng cường mosaic giúp mô hình học được các đặc trưng mạnh mẽ hơn. Bên cạnh đó, tồn tại khoảng cách hiệu suất giữa khả năng nhận diện (mAP50) và khả năng định vị chính xác (mAP50-95) trên lá cà phê với các bệnh "đốm mắt nâu" và "gỉ sắt" lần lượt là (0,978-0,795) và (0,988-0,869). Thêm vào đó, phân tích bằng Grad-CAM, HiResCAM cho thấy, mô hình có khả năng diễn giải và tập trung vào đúng vùng bệnh, sự chú ý đôi khi còn phân tán, lý giải cho sự sụt giảm hiệu suất ở các độ đo yêu cầu độ chính xác khoanh vùng cao. Kết quả thực nghiệm cho thấy rằng YOLOv8 cho kết quả tốt hơn YOLOv13 trên các bộ dữ liệu thực nghiệm. YOLOv8 là phiên bản thường được sử dụng vì tính ổn định, được tối ưu hóa tốt. Chúng tôi chọn kiến trúc này vì các chỉ số độ đo trên bộ dữ liệu thực nghiệm tốt và khả năng triển khai trên nhiều thiết bị.

Tuy nhiên, nghiên cứu vẫn còn một số giới hạn, chủ yếu là khả năng định vị chính xác các vết bệnh rất nhỏ và bản chất "hộp đen" chưa được giải quyết rõ ràng. Các hướng phát triển trong tương lai tập trung vào việc tối ưu hóa kiến trúc để cải thiện khả năng phát hiện đối tượng nhỏ, đồng thời nghiên cứu các phương pháp giảm nhẹ mô hình, cho phép triển khai hiệu quả trên các thiết bị di động có tài nguyên hạn chế.

TÀI LIỆU THAM KHẢO

[1] Department of Agriculture and Rural Development of Thua Thien Hue Province (2025), "Situation of crop pests (from July 9, 2025 to July 15, 2025)", https://snnmt.hue.gov.vn/thong-bao-tinh-hinh-sinh-vat-gay-hai-cay-trong/tinh-hinh-sinh-vat-gay-hai-cay-trong-tu-ngay-09-7-2025-den-ngay-15-7-2025.html, accessed 7 January 2026 (in Vietnamese).

[2] BMFE Corp, "Rice blast disease: Causes, damage mechanisms and effective prevention", https://baominhagri.com/benh-dao-on-hai-lua-nguyen-nhan-co-che-gay-hai-va-cach-phong-tru-hieu-qua, accessed 7 January 2026 (in Vietnamese).

[3] Bio-materials (2024), "Dry branch and dry fruit disease on coffee trees", https://nguyenlieusinhhoc.com/benh-kho-canh-kho-qua-tren-cay-ca-phe/, accessed 7 January 2026 (in Vietnamese).

[4] M. Arsenovic, M. Karanovic, S. Sladojevic, et al. (2019), "Solving Current Limitations of Deep Learning Based Approaches for Plant Disease Detection", *Symmetry*, **11(7)**, 939, DOI: 10.3390/sym11070939.

[5] L. Li, S. Zhang, B. Wang (2021), "Plant Disease Detection and Classification by Deep Learning - A Review", *IEEE Access*, **9**, pp.56683-56698, DOI: 10.1109/ACCESS.2021.3069646.

[6] J.G.A. Barbedo (2019), "Plant disease identification from individual lesions and spots using deep learning", *Biosystems Engineering*, **180**, pp.96-107, DOI: 10.1016/j.biosystemseng.2019.02.002.

[7] S.P. Mohanty, D.P. Hughes, M. Salathé (2016), "Using Deep Learning for Image-Based Plant Disease Detection", *Frontiers in Plant Science*, **7**, DOI: 10.3389/fpls.2016.01419.

[8] C.D. Trinh, T.A. Mac, V.K. Giap, et al. (2024), "Rice Leaf Diseases Detection Using YOLOv8", *JST: Engineering and Technology for Sustainable Development*, **34(2)**, DOI: 10.51316/jst.173.etsd.2024.34.2.6.

[9] D. Singh, N. Jain, P. Jain, et al. (2020), "PlantDoc: A Dataset for Visual Plant Disease Detection", *CoDS COMAD 2020: Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, pp.249-253, DOI: 10.1145/3371158.3371196.

[10] M.A. Jahin, S. Shahriar, M.F. Mridha, et al. (2025), "Soybean Disease Detection via Interpretable Hybrid CNN-GNN: Integrating MobileNetV2 and GraphSAGE with Cross-Modal Attention", *arXiv*, DOI: 10.48550/arXiv.2503.01284.

[11] H. Sun, X. Li, X. Li, et al. (2025), "A multi-scale detection model for tomato leaf diseases with small target detection head", *Frontiers in Plant Science*, **16**, 1598534, DOI: 10.3389/fpls.2025.1598534.

[12] Y. Miao, W. Meng, X. Zhou (2025), "SerpensGate-YOLOv8: an enhanced YOLOv8 model for accurate plant disease detection", *Frontiers in Plant Science*, **15**, 1514832, DOI: 10.3389/fpls.2024.1514832.

[13] A. Kumar, H.P. Monga, T. Brahma, et al. (2025), "Mobile-Friendly Deep Learning for Plant Disease Detection: A Lightweight CNN Benchmark Across 101 Classes of 33 Crops", *arXiv*, DOI: 10.48550/arXiv.2508.10817.

[14] N. Nigar, H.M. Faisal, M. Umer, et al. (2024), "Improving Plant Disease Classification With Deep-Learning-Based Prediction Model Using Explainable Artificial Intelligence", *IEEE Access*, **12**, pp.100005-100014, DOI: 10.1109/ACCESS.2024.3428553.

[15] X. Mao, Y. Chen, Y. Zhu, et al. (2023), "COCO-O: A Benchmark for Object Detectors under Natural Distribution Shifts", *arXiv*, DOI: 10.48550/arXiv.2307.12730.

[16] G. Jocher, A. Chaurasia, T. Qiu, et al. (2023), "Ultralytics YOLOv8", [Online], Available: https://github.com/ultralytics/ultralytics.

[17] M.S.Z. Abid, B. Jahan, A.A. Mamun, et al. (2024), "Bangladeshi crops leaf disease detection using YOLOv8", *Heliyon*, **10(18)**, e36694, DOI: 10.1016/j.heliyon.2024.e36694.

[18] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, et al. (2010), "Object Detection with Discriminatively Trained Part-Based Models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32(9)**, pp.1627-1645, DOI: 10.1109/TPAMI.2009.167.

[19] J.R. Uijlings, K.E. van de Sande, T. Gevers, et al. (2013), "Selective Search for Object Recognition", *International Journal of Computer Vision*, **104**, pp. 154-171, DOI: 10.1007/s11263-013-0620-5.

[20] R. Girshick, J. Donahue, T. Darrell, et al. (2014), "Rich feature hierarchies for accurate object detection and semantic segmentation", *arXiv*, DOI: 10.48550/arXiv.1311.2524.

[21] R. Girshick (2015), "Fast R-CNN", *arXiv*, DOI: 10.48550/arXiv.1504.08083.

[22] S. Ren, K. He, R. Girshick, et al. (2016), "Faster R-CNN: Towards RealTime Object Detection with Region Proposal Networks", *arXiv*, DOI: 10.48550/arXiv.1506.01497.

[23] J. Redmon, S. Divvala, R. Girshick, et al. (2016), "You Only Look Once: Unified, Real-Time Object Detection", *arXiv*, DOI: 10.48550/arXiv.1506.02640.

[24] M. Kotthapalli, D. Ravipati, R. Bhatia (2025), "YOLOv1 to YOLOv11: A Comprehensive Survey of Real-Time Object Detection Innovations and Challenges", *arXiv*, DOI: 10.48550/arXiv.2508.02067.

[25] B.A. Kyem, J.K. Asamoah, A. Dontoh, et al. (2025), "PaveSync: A Unified and Comprehensive Dataset for Pavement Distress Analysis and Classification", *arXiv*, DOI: 10.48550/arXiv.2512.20011.

[26] M. Lei, S. Li, Y. Wu, et al. (2025), "YOLOv13: Real-Time Object Detection with Hypergraph-Enhanced Adaptive Visual Perception", *arXiv*, DOI: 10.48550/arXiv.2506.17733.

[27] N.V. Tu, P.N.H. Long, V.H. Viet (2025), "xAI-CV: An Overview of Explainable Artificial Intelligence in Computer Vision", *arXiv*, DOI: 10.48550/arXiv.2509.18913.

[28] R.L. Draelos, L. Carin (2021), "Use HiResCAM instead of Grad-CAM for faithful explanations of convolutional neural networks", *arXiv*, DOI: 10.48550/arXiv.2011.08891.

[29] R.R. Selvaraju, M. Cogswell, A. Das, et al. (2019), "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization", *International Journal of Computer Vision (IJCV)*, **128**, pp.336-359, DOI: 10.1007/s11263-019-01228-7.

[30] Q.K. Nguyen, T.T.H. Nguyen, V.T.K. Nguyen, et al. (2024), "Efficient and Concise Explanations for Object Detection with Gaussian-Class Activation Mapping Explainer", *arXiv*, DOI: 10.48550/arXiv.2404.13417.