

DEVELOPING A FRUIT RIPENESS DETECTION MODULE BASED ON MODERN DEEP LEARNING ARCHITECTURES, APPLIED IN SMART AGRICULTURAL PROCESSING SYSTEMS

PHÁT TRIỂN BỘ PHÁT HIỆN MỨC ĐỘ CHÍN CỦA TRÁI CÂY DỰA TRÊN KIẾN TRÚC MẠNG HỌC SÂU HIỆN ĐẠI, ỨNG DỤNG TRONG CÁC HỆ THỐNG CHẾ BIẾN NÔNG SẢN THÔNG MINH

Nguyễn Duy Linh, Nguyễn Nương Quỳnh
Trường Đại học Quảng Bình

ABSTRACT: The application of artificial intelligence, especially Computer Vision, in smart agriculture has become popular. Human labor is also partly liberated and agricultural output is also improved in both quantity and quality. Fruit classification is a very important step and the final step in the agricultural production and processing chain. This paper focuses on improving the YOLOv8n architecture by replacing the original convolution operations with a new convolution module called the Receptive Field Convolution Block Attention Module to detect fruit ripeness. This module leverages the Group Convolution mechanism and the Convolution Block Attention Module to enhance feature extraction capabilities. The experiments are trained and evaluated on the Fruit Ripening Process Dataset. The experimental results show that the proposed network achieves the best performance at 93.8% mAP@0.5 and proves superiority over other methods under the same experimental conditions.

Keywords: Convolution Block Attention Module (CBAM), convolutional neural network (CNN), fruit classification systems, fruit ripeness detection, YOLOv8.

TÓM TẮT: Việc ứng dụng trí tuệ nhân tạo, đặc biệt là Thị giác máy tính trong nông nghiệp thông minh đã trở nên phổ biến. Sức lao động của con người cũng được giải phóng một phần nào, sản lượng nông sản cũng được nâng cao cả về số lượng và chất lượng. Phân loại trái cây là khâu hết sức quan trọng và là khâu cuối cùng trong chuỗi sản xuất và chế biến nông sản. Bài báo này tập trung vào việc cải thiện kiến trúc YOLOv8n bằng cách thay thế các phép toán tích chập bằng một mô-đun tích chập mới gọi là mô-đun Receptive Field Convolution Block Attention Module để phát hiện độ chín của trái cây. Mô-đun này tận dụng những ưu điểm của cơ chế tích chập nhóm và mô-đun Convolution Block Attention Module để nâng cao khả năng trích xuất đặc trưng. Các thí nghiệm được huấn luyện và đánh giá trên tập dữ liệu Fruit Ripening Process. Kết quả thí nghiệm cho thấy rằng, mạng được đề xuất đạt hiệu suất tốt nhất ở mức 93,8% mAP@0.5 và thể hiện tính ưu việt hơn so với các phương pháp khác trong cùng điều kiện thử nghiệm.

Từ khóa: CBAM, mạng nơ-ron tích chập, hệ thống phân loại trái cây, phát hiện trái cây chín, YOLOv8.

1. ĐẶT VẤN ĐỀ

Theo thống kê của Tổ chức Lương thực và Nông nghiệp của Liên hợp quốc (FAO) năm 2021, sản lượng trái cây Thế giới đạt khoảng 910 triệu tấn và con số này

có thể thay đổi theo từng năm tùy thuộc vào điều kiện tự nhiên [1]. Trong đó, các nước sản xuất hàng đầu là Trung Quốc, Hoa Kỳ, Ấn Độ, Mexico, ... Với sản lượng hàng năm lớn như vậy, việc phân loại và đóng gói

trước khi đưa ra thị trường đòi hỏi phải nhanh chóng và chính xác. Những công việc này rất cần thiết để đảm bảo chất lượng tốt nhất của trái cây. Tuy nhiên, việc phân loại trái cây với các chủng loại và mức độ chín khác nhau không phải là một nhiệm vụ dễ dàng vì sự tương đồng về màu sắc, hình dạng và kích thước [2]. Thông thường, giai đoạn này sẽ được thực hiện bởi một nhóm chuyên gia hoặc người đã được đào tạo, với các yếu tố chính để đánh giá là màu sắc và chất lượng của sản phẩm trái cây. Việc kiểm tra thủ công thường phát sinh nhiều sai số tùy thuộc vào kinh nghiệm và đánh giá của mỗi cá nhân. Do đó, chất lượng phân loại không được đồng đều như mong đợi. Gần đây, nhiều nhà nghiên cứu đã tập trung vào lĩnh vực nông nghiệp và đưa ra nhiều giải pháp và ứng dụng khác nhau cho việc phát hiện và phân loại trái cây. Các kỹ thuật này nhằm mục đích giải quyết một phần những thách thức phức tạp như tính đa dạng, sự không đồng đều và sự không nhất quán về hình dạng, màu sắc và kết cấu [3]. Nổi bật trong số đó là các phương pháp dựa trên Thị giác máy tính (Computer Vision). Các phương pháp này tận dụng các thuật toán học máy, đặc biệt là Mạng Nơ-ron tích chập (Convolutional Neural Network - CNN), để phân biệt các đặc điểm vốn có của trái cây. Theo xu hướng đó, bài báo này đề xuất một kỹ thuật để cải thiện kiến trúc mạng YOLOv8n (Phiên bản nhỏ gọn -nano version) cho phân loại độ chín của trái cây bằng cách phân tích cẩn thận kiến trúc mạng ban đầu và thay thế hoàn toàn các

phép tích chập tiêu chuẩn bằng các mô-đun tích chập Receptive Field Convolution Block Attention Module Convolution (RFCBAMConv) bên trong các mô-đun Xương sống (Backbone) và Cổ (Neck).

Với sự kết hợp của các kiến trúc nhỏ gọn, cơ chế chú ý CBAM (Convolution Block Attention Module) và tối ưu hóa độ phức tạp tính toán, mạng được đề xuất có độ chính xác cao và có tiềm năng được triển khai trong các thiết bị tính toán thấp cho các hệ thống phân loại độ chín tự động của trái cây.

Bài báo cung cấp một số đóng góp chính như sau:

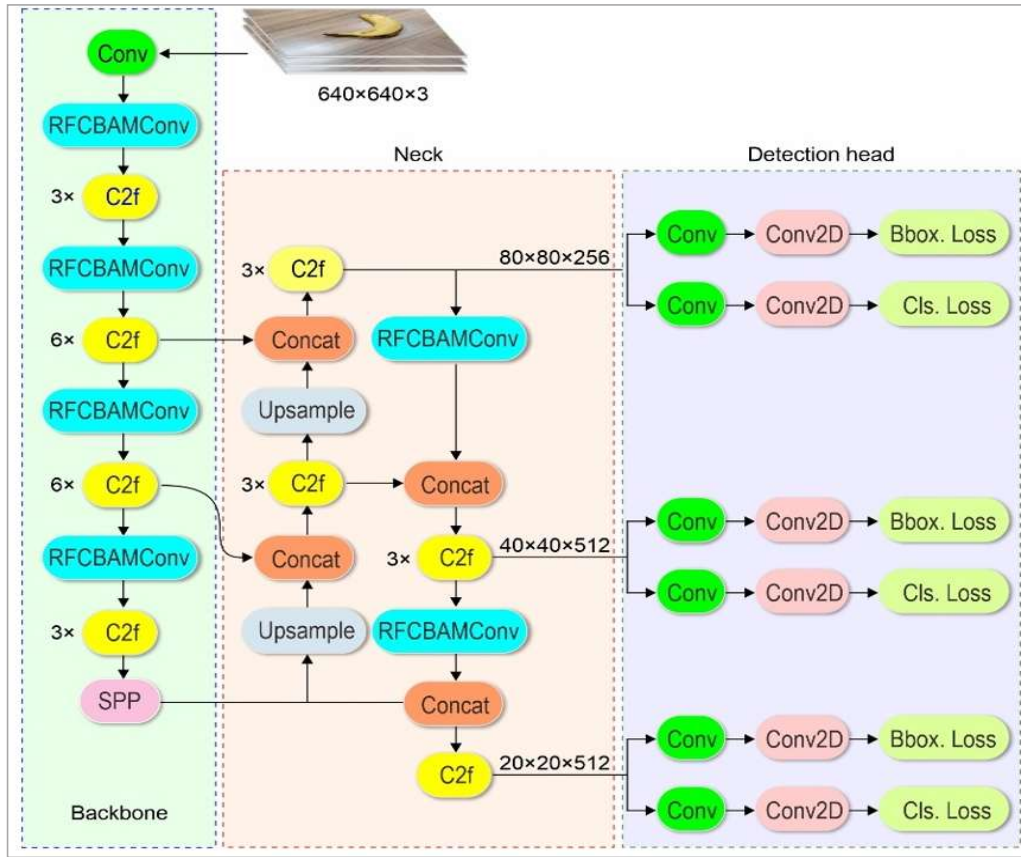
1 - Đề xuất một bộ phát hiện độ chín của trái cây hiệu quả dựa trên kiến trúc YOLOv8n cho các hệ thống phân loại trái cây tự động.

2 - Phương pháp đề xuất đạt hiệu suất tốt hơn so với các phương pháp khác trên tập dữ liệu Fruit Ripening Process.

Phần còn lại của bài báo được sắp xếp như sau: phần 2 trình bày phương pháp đề xuất để cải tiến hệ thống mạng học sâu YOLOv8n cho phát hiện độ chín của trái cây. Phần 3 phân tích và đánh giá các kết quả thực nghiệm. Phần 4 kết luận vấn đề và hướng phát triển trong tương lai.

2. PHƯƠNG PHÁP ĐỀ XUẤT

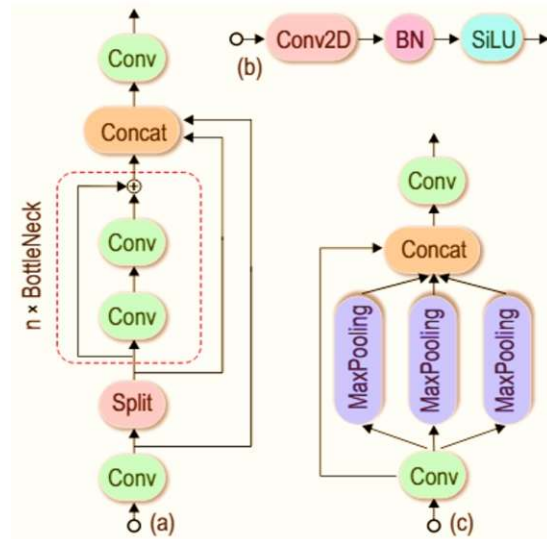
Tổng quan về mạng lưới phát hiện độ chín của trái cây được mô tả chi tiết như trong Hình 1. Đây là kiến trúc YOLOv8 cải tiến, bao gồm ba mô-đun: Xương sống (Backbone), Cổ (Neck) và Đầu phát hiện (Detection head).



Hình 1. Tổng quan về mạng phát hiện độ chín của trái cây được đề xuất

2.1. Kiến trúc mạng đề xuất

Bài báo này phân tích và đánh giá kỹ lưỡng từng thành phần trong kiến trúc YOLOv8 ban đầu [4]. Từ những kết quả phân tích đó, nghiên cứu tập trung vào việc tinh chỉnh một số khối trong các mô-đun Backbone và Neck. Cụ thể, khối Cross Stage Partial Bottleneck với hai phép tích chập (C2f) được tái sử dụng, khối Spatial Pyramid Pooling Fast (SPPF) và Conv được thay thế tương ứng bằng Spatial Pyramid Pooling (SPP) và khối Receptive Field Convolution Block Attention Module Convolution (RFBAMConv) [5]. Chi tiết kiến trúc của các khối Conv, C2f và SPP được thể hiện trong Hình 2.



Hình 2. Kiến trúc của các khối Cross Stage Partial Fast Bottleneck (C2f) (a), Conv (b) và Spatial Pyramid Pooling (SPP) (c).

2.1.1. Mô-đun xương sống (Backbone)

Mô-đun Backbone được thiết kế lại dựa trên một khối Conv, tiếp theo là bốn khối kết hợp giống hệt nhau, và kết thúc bằng một khối SPP. Trong đó, các khối trung gian tương tự được xếp tuần tự với một số khối C2f (với số lần lặp lại là 3, 6, 6 và 3 lần) và một khối RFCBAMConv. Hình 3 thể hiện kiến trúc của RFCBAMConv, là sự kết hợp giữa cơ chế mô-đun chú ý RFCBAM và một mô-đun tích chập tiêu chuẩn (Conv2D). Bên cạnh đó, mô-đun chú ý kênh (CAM) được thay thế bằng mô-đun chú ý Squeeze-and-Excitation (SE). Mô-đun RFCBAMConv được thiết kế để giải quyết vấn đề chia sẻ tham số hạt nhân tích chập và cải thiện khả năng trích xuất đặc trưng cho phép toán tích chập tiêu chuẩn. Khối RFCBAMConv này thực hiện các phép tích chập nhóm, có thể tiết kiệm một lượng lớn tham số. Mặt khác, cơ chế chú ý CBAM hướng dẫn cho mạng tìm tập trung vào các thông tin quan trọng trên mỗi cấp độ bản đồ đặc trưng. Giả sử $F \in R^{W \times H \times C}$ là bản đồ đặc trưng đầu vào và $F' \in R^{W \times H \times C}$ là bản đồ đặc trưng đầu ra. Quy trình vận hành của RFCBAMConv có thể được mô tả như sau:

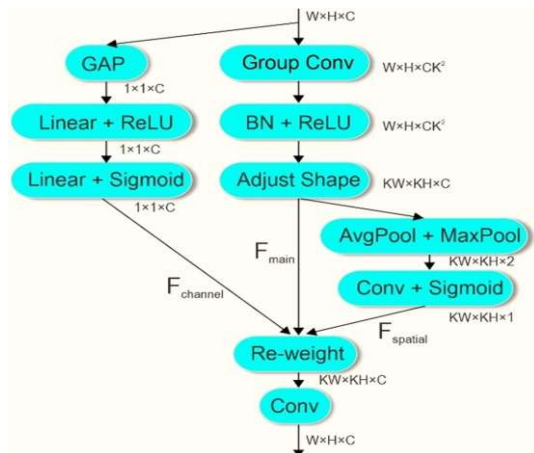
$$F' = f^{3 \times 3} (F_{Channel} \times F_{Main} \times F_{Spatial}),$$

trong đó $f^{3 \times 3}$ là phép toán tích chập tiêu chuẩn với hạt nhân có kích thước 3×3 . $F_{Channel}$, F_{Main} và $F_{Spatial}$ lần lượt là các bản đồ đặc trưng đầu ra của các nhánh chú ý theo kênh, chính và không

gian. $F_{Channel}$, F_{Main} và $F_{Spatial}$ được tính theo các công thức sau:

$$\begin{aligned} F_{Channel} &= \sigma(FC(ReLU(FC(GAP(F))))), \\ F_{Main} &= Reshape(BN(ReLU(f^{3 \times 3}(F))), \\ F_{Spatial} &= \sigma(f^{1 \times 1}([Avg(FM), Max(FM)])), \end{aligned}$$

trong đó, GAP (Global Average Pooling) là lớp gộp trung bình toàn cục. FC (Fully Connected layer) là lớp kết nối đầy đủ. ReLU (Rectified Linear Unit) là hàm kích hoạt đơn vị tuyến tính chỉnh lưu. Avg và Max lần lượt là lớp gộp trung bình (Average Pooling) và gộp tối đa (Max Pooling). Phép toán $[\cdot]$ mô tả lớp ghép nối (Concatenation). $f^{1 \times 1}$ biểu thị phép toán tích chập tiêu chuẩn với kích thước hạt nhân là 1×1 . Ký hiệu σ là hàm kích hoạt Sigmoid. Ở cuối mô-đun Backbone, bài báo này sử dụng khối SPP từ YOLOv5 [6] để thay thế khối SPFF. Kích thước hạt nhân của các lớp gộp tối đa thay đổi từ 3×3 đến 5×5 để đảm bảo các tham số mạng hợp lý.



Hình 3. Kiến trúc mô-đun RFCBAMConv

2.1.2. Mô-đun Cổ (Neck)

Mô-đun Neck tận dụng kiến trúc mạng Path Aggregation Network (PAN) như trong mạng YOLOv8 và cũng thay thế toàn bộ các khối Conv bằng các khối RFCBAMConv. Mô-đun này lấy mẫu lại các bản đồ đặc trưng hiện tại và tổng hợp chúng với các bản đồ đặc trưng cấp thấp trước đó từ mô-đun Backbone bằng các phép toán ghép nối (Concatenation). Ba bản đồ đặc trưng đầu ra tương ứng với ba tỷ lệ của đối tượng (nhỏ, trung bình và lớn) được tạo ra bởi mô-đun Neck. Ba bản đồ đặc trưng này đã được làm phong phú thêm thông tin quan trọng và đi qua mô-đun đầu phát hiện (Detection head).

2.1.3. Mô-đun đầu phát hiện (Detection head)

Mô-đun đầu phát hiện cũng tái sử dụng kiến trúc của ba đầu phát hiện từ YOLOv8 gốc với kỹ thuật tách rời đầu (decouple head) và neo tự do (Free-anchor). Các bản đồ đặc trưng từ đầu ra của mô-đun Neck được chuyển đến hai khối, đó là sự kết hợp giữa khối Conv và tích chập chuẩn để hồi quy hộp giới hạn (với bốn tọa độ của hộp: x, y, h, w) và phân loại (số lớp: c) trên ba tỷ lệ của đối tượng. Khối Conv được mô tả trong Hình 2 (b). Khối này sử dụng một lớp tích chập chuẩn 1×1 (Conv2D), một Batch Normalization (BN) và một hàm kích hoạt ReLU. Các khối Conv chỉ được sử dụng trong mô-đun đầu phát hiện của mạng được đề xuất. Bảng 1 trình bày chi tiết về mô-đun đầu phát hiện.

Bảng 1. Chi tiết của mô-đun đầu phát hiện

Heads	Input	Anchor	Output	Object
1	$80 \times 80 \times 256$	Free	$80 \times 80 \times 4/80 \times 80 \times c$	Small
2	$40 \times 40 \times 512$	Free	$40 \times 40 \times 4/40 \times 40 \times c$	Medium
3	$20 \times 20 \times 512$	Free	$20 \times 20 \times 4/20 \times 20 \times c$	Large

2.2. Hàm tổn thất (Loss function)

Mạng được đề xuất sử dụng hàm mất mát được định nghĩa như sau:

$$L = \lambda_{Box} L_{Box} + \lambda_{DFL} L_{DFL} + \lambda_{Cls} L_{Cls},$$

trong đó, tổn thất hồi quy hộp giới hạn kết hợp L_{Box} với L_{DFL} và áp dụng tương ứng hàm Complete Intersection over Union (CioU) và Distribution Focal Loss (DFL). Tổn thất phân loại L_{Cls} sử dụng hàm Binary Cross-Entropy để tính toán. λ_{Box} , λ_{DFL} , và λ_{Cls} là các tham số cân bằng.

3. KẾT QUẢ THỬ NGHIỆM

3.1. Chuẩn bị tập dữ liệu (Dataset)

Các thí nghiệm tiến hành huấn luyện và đánh giá trên Bộ dữ liệu Fruit Ripening Process [7], được thu thập từ website Roboflow. Các hình ảnh được chú thích bằng nhãn cấp độ đối tượng, nghĩa là mỗi trái cây (chuối) trong hình ảnh được cung cấp một hộp giới hạn và được gán nhãn mức độ chín theo định dạng YOLO. Bộ dữ liệu này bao gồm 6,789 ảnh các loại chuối với các mức độ chín khác nhau như fresh ripe, fresh

unripe, overripe, ripe, rotten, unripe được chọn từ nhiều nguồn khác nhau. Bộ dữ liệu được chia thành hai tập con với 5,264 ảnh dùng để huấn luyện và 1,525 ảnh dùng để đánh giá.

3.2. Cài đặt thử nghiệm

Mạng được đề xuất được viết bằng ngôn ngữ lập trình Python và sử dụng Pytorch framework. Các thí nghiệm được huấn luyện và đánh giá trên GPU GeForce GTX 1080Ti 11GB. Giai đoạn huấn luyện áp dụng phương pháp tối ưu hóa Stochastic Gradient Descent (SGD). Tốc độ học (Learning rate) ban đầu được đặt ở mức 10^{-4} và kết thúc ở mức 10^{-2} . Tham số momentum được đặt ở mức 0.937. Quá trình huấn luyện sử dụng 200 epochs với kích mỗi batch là 16. Các tham số cân bằng được đặt như sau: $\lambda_{\text{Box}}=1.5$, $\lambda_{\text{Cls}}=0.5$, và $\lambda_{\text{DFL}}=1.5$. Để cải thiện tập dữ liệu huấn luyện và tránh các vấn đề quá khớp (Overfitting), một số phương pháp tăng cường dữ liệu (như mosaic, translate, scale và flip) được sử dụng. Trong giai đoạn suy luận, các tham số được thiết lập với kích thước hình ảnh là 640×640 pixels, kích thước batch là 16, ngưỡng confidence threshold = 0.5 và ngưỡng IoU = 0.5. Thời gian suy luận được báo cáo bằng mili giây (ms).

3.3. Kết quả

Để đánh giá hiệu suất của mạng được đề xuất, nghiên cứu này đào tạo lại từ đầu các kiến trúc nhỏ gọn của YOLOv5 và YOLOv8 (phiên bản nano). Mặt khác, nghiên cứu cũng tiến

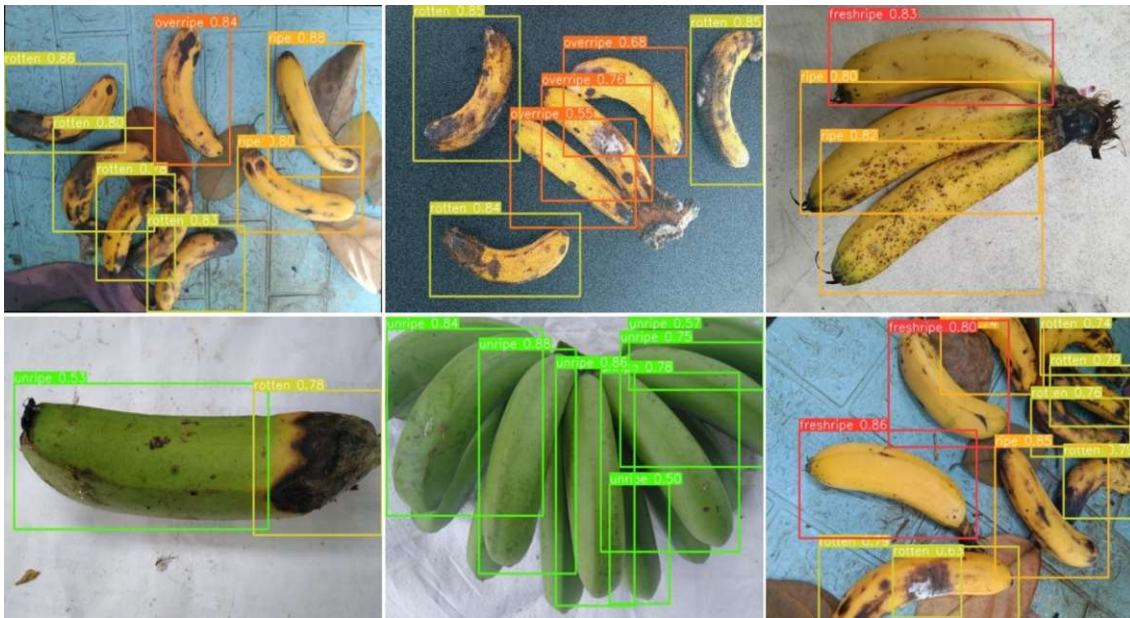
hành tương tự với các phiên bản khác nhau của mạng được đề xuất, từ phiên bản nhỏ gọn nhất đến phiên bản cực lớn (YOLOv8s-RFCBAMConv, YOLOv8m-RFCBAMConv, YOLOv8l-RFCBAMConv và YOLOv8x-RFCBAMConv). Kết quả là, mạng được đề xuất đạt được 93.8% mAP@0.5 và 70.3% mAP@0.5:0.95. Từ kết quả so sánh trong Bảng 2, có thể thấy rằng mạng được đề xuất vượt trội hơn hầu hết các mạng so sánh. Cụ thể hơn, đối với bộ dữ liệu Ripening Process, hiệu suất của mạng được đề xuất tương đương với YOLOv8l-RFCBAMConv phiên bản lớn (0.8%↓) và tốt hơn YOLOv8s-RFCBAMConv (0.4%↑). Trong khi đó, tham số mạng và độ phức tạp tính toán (GFLOPs) tương đương với YOLOv8n và tốc độ suy luận tương đương với YOLOv5n (0.5 ms↓).

Bảng 2. Kết quả so sánh giữa mạng đề xuất và các phương pháp khác trên tập dữ liệu Ripening Process

Method	Parameter	GFLOPs	Weight (MB)	mAP (%)		Inf. (ms)
				0.5	0.5:0.95	
YOLOv5n	1,769,329	4.2	3.8	91.6	64.1	1.4
YOLOv8n	3,006,428	8.1	6.2	92.8	69.9	0.6
YOLOv8s-RFCBAMConv	11,243,112	29.0	22.8	93.4	70.6	2.3
YOLOv8m-RFCBAMConv	34,109,656	86.1	68.1	93.1	70.6	4.1
YOLOv8l-RFCBAMConv	76,910,856	192.9	154.4	93.2	71.1	6.4
YOLOv8x-RFCBAMConv	120,087,934	301.0	240.8	93.1	70.8	9.4
Phương pháp đề xuất	3,064,472	8.3	6.4	93.8	70.3	1.9

Những hiệu suất cao này hứa hẹn mạng đề xuất có thể được triển khai trên các thiết bị tính toán thấp cho các hệ thống phân

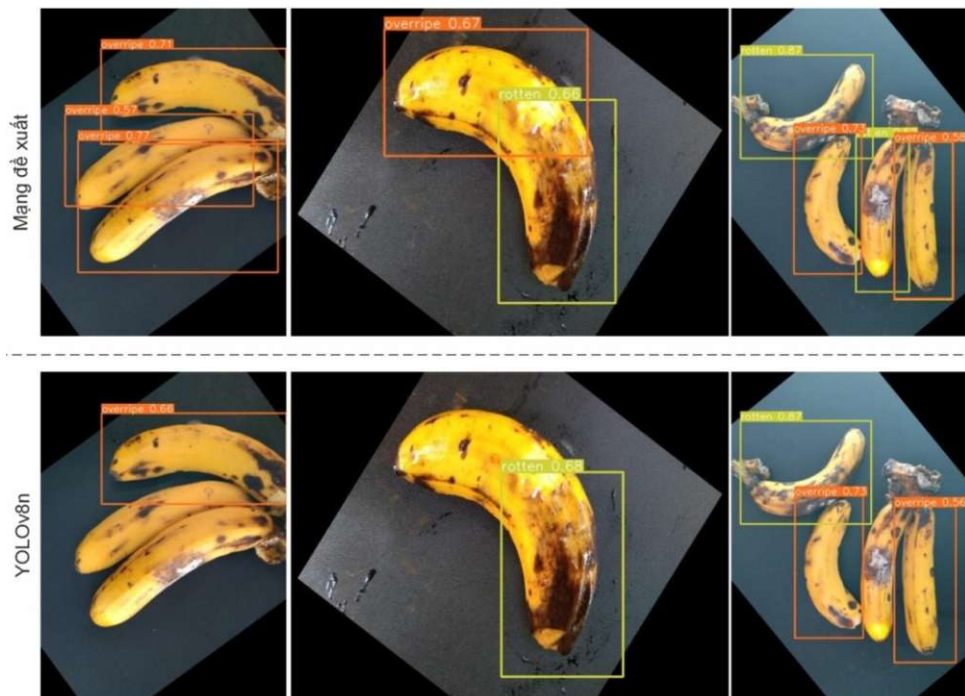
loại độ chín của trái cây theo thời gian thực. Một số kết quả định tính trên hai tập dữ liệu được thể hiện trong Hình 4.



Hình 4. Kết quả định tính của mạng đề xuất trên tập dữ liệu đánh giá của bộ dữ liệu Fruit Ripening Process

Kết quả so sánh giữa mạng đề xuất và YOLOv8n được thể hiện rõ trong Hình 5. Từ kết quả trực quan hóa này, có thể thấy rằng mạng đề xuất có khả năng phát hiện vật thể tốt hơn trong các tình huống bị che khuất, màu sắc tương tự giữa các mức độ chín và các phần bị thối bên trong quả. Tuy nhiên, do một số mức độ chín của quả có màu sắc tương đối gần nhau, việc phát hiện

và phân biệt chúng vẫn là một thách thức lớn. Ngoài ra, kết quả phân loại còn phụ thuộc vào một số tác động của môi trường như độ chiếu sáng, độ phức tạp của nền, vị trí và góc của vật thể,... Do đó, cần phải phát triển các bộ phát hiện vật thể mạnh mẽ và tập dữ liệu phong phú để tối ưu hóa hiệu suất của các hệ thống phân loại độ chín tự động của trái cây.



Hình 5. Kết quả so sánh giữa mạng đề xuất và mạng YOLOv8n trên tập dữ liệu đánh giá của bộ dữ liệu Fruit Ripening Process

4. KẾT LUẬN

Bài báo này đề xuất một kỹ thuật cải tiến kiến trúc YOLOv8n để phát hiện độ chín của trái cây, hỗ trợ các hệ thống phân loại độ chín tự động của trái cây trong các hệ thống chế biến nông sản thông minh. Phương pháp đề xuất bao gồm ba phần: mô-đun xương sống (Backbone), mô-đun Cổ (Neck) và mô-đun Đầu phát hiện (Detection head). Các mô-đun xương sống

và mô-đun cổ được thiết kế lại bằng cách thay thế các khối Conv bằng các khối RFACConv, ngoại trừ khối Conv đầu tiên. Bên cạnh đó, khối SPFF cũng được thay thế bằng khối SPP với các kích thước hạt nhân nhỏ của lớp Max Pooling. Đầu phát hiện tận dụng ý tưởng từ kiến trúc ban đầu trong YOLOv8n. Mạng được đề xuất đạt được mAP tốt nhất ở mức 93,8% và tốt hơn so với một số phương pháp hiện nay. Việc tối ưu

hóa số lượng tham số, độ phức tạp tính toán, thời gian suy luận và độ chính xác phát hiện hứa hẹn nó sẽ được triển khai trên các hệ thống thời gian thực. Trong tương lai, nghiên cứu này sẽ cố gắng triển khai thử nghiệm trên các tập dữ liệu trái cây lớn hơn

với tổng hợp nhiều loại trái cây để tăng tính khái quát hóa của phương pháp đề xuất. Bên cạnh đó, nghiên cứu này cũng sẽ so sánh hiệu suất với các phiên bản mới nhất của mạng YOLO và một số mạng phát hiện vật thể điển hình khác.

TÀI LIỆU THAM KHẢO

- [1] FAO (2022), “Agricultural production statistics 2000-2021,” in FAOSTAT analytical briefs, no. 60.
- [2] J. Steinbrener, K. Posch and R. Leitner (2019), “Hyperspectral fruit and vegetable classification using convolutional neural networks,” Computers and Electronics in Agriculture, vol. 162, pp. 364-372.
- [3] K. Hameed, D. Chai and A. Rassau (2018), “A comprehensive review of fruit and vegetable classification techniques,” Image and Vision Computing, vol. 80, pp. 24-44.
- [4] G. Jocher, A. Chaurasia and J. Qiu (2023), “Ultralytics yolov8”.
- [5] X. Zhang, C. Liu, D. Yang, T. Song, Y. Ye, K. Li and Y. Song (2023), “Rfaconv: Innovating spatial attention and standard convolutional operation”.
- [6] G. Jocher and et al. (2020), “ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements,” Oct.
- [7] F. Ripening (2022), “Fruit ripening process dataset.”
<https://universe.roboflow.com/fruit-ripening/fruit-ripening-process> , oct 2022. visited on 2025-08-26.

Liên hệ:

TS. Nguyễn Duy Linh

Khoa Công nghệ thông tin, Trường Đại học Quảng Bình

Địa chỉ: 18 Nguyễn Văn Linh, Phường Đồng Thuận, Tỉnh Quảng Trị

Email: nguyenduylinhqbu@gmail.com

Ngày nhận bài: 13/10/2025

Ngày gửi phản biện: 4/11/2025

Ngày duyệt đăng: 15/12/2025