

DOI:10.22144/ctujos.2025.019

THIẾT KẾ HỆ THỐNG PHÂN LOẠI TRÁI ỚT SAU THU HOẠCH ỨNG DỤNG THỊ GIÁC MÁY TÍNH

Phan Trung Nguyên¹, Đặng Phát Đạt¹, Nguyễn Kỳ Em¹, Ngô Lữ An Kiên¹, Lâm Hồ Ngọc Hân², Phó Hoàng Linh³ và Nguyễn Văn Mươi^{1*}

¹Khoa Tự động hóa, Trường Bách Khoa, Trường Đại học Cần Thơ, Việt Nam

²Công ty TNHH Phòng xét nghiệm DIAN, quận Tân Bình, Thành phố Hồ Chí Minh, Việt Nam

³Khoa Kỹ thuật Cơ khí, Trường Đại học Kỹ thuật công nghệ Cần Thơ, Việt Nam

*Tác giả liên hệ (Corresponding author): nvmuot@ctu.edu.vn

Thông tin chung (Article Information)

Nhận bài (Received): 28/06/2024

Sửa bài (Revised): 23/07/2024

Duyệt đăng (Accepted): 24/10/2024

Title: Designing a chilli classification system to commercial standards applying image processing techniques

Author(s): Phan Trung Nguyen¹, Dang Phat Dat¹, Nguyen Ky Em¹, Ngo Lu An Kien¹, Lam Ho Ngoc Han², Pho Hoang Linh³ and Nguyen Van Muot^{1*}

Affiliation(s): ¹Can Tho University, Viet Nam; ²DIAN Laboratory Company Limited, Viet Nam; ³Can Tho University of Technology, Viet Nam

TÓM TẮT

Hiện nay, công nghệ xử lý ảnh ngày càng được ứng dụng phổ biến, đa dạng ở các lĩnh vực, kể cả trong nông nghiệp và công nghệ sau thu hoạch. Việc dùng thị giác máy tính để phân loại tình trạng lỗi của trái ớt là một điển hình giúp chọn lọc sản phẩm sau thu hoạch đạt chuẩn thương mại một cách hiệu quả. Việc phân loại ớt thủ công truyền thống tốn kém chi phí và ảnh hưởng đến sức khỏe người dân, đòi hỏi sự cải thiện bằng khoa học công nghệ. Bài báo này trình bày thiết kế phần cứng và thuật toán nhận dạng dùng thị giác máy tính để loại bỏ ớt bị gãy đôi, mất cuống và thán thư. OpenCV và Python được sử dụng phân ngưỡng, áp dụng các mạng thần kinh tích chập như ResNet và MobileNet để phát hiện bệnh thán thư. Thông tin về tình trạng ớt được gửi đến vi điều khiển ESP32 để điều khiển cơ cấu chấp hành phân loại. Thời gian xử lý cho phân ngưỡng và phát hiện thán thư là 0,08 giây, chu kỳ phân loại trung bình là 1 giây. Các thí nghiệm cho thấy hệ thống có tiềm năng áp dụng thực tế ở các hộ nông dân hoặc thương lái.

Từ khóa: Mạng thần kinh tích chập, OpenCV, phân ngưỡng, Python, thị giác máy tính

ABSTRACT

Currently, image processing technology is increasingly being applied in a variety of fields, including agriculture and post-harvest technology. Using computer vision to classify the defect status of chili peppers is a typical example of helping to select post-harvest products that meet commercial standards effectively. Traditional manual sorting of chilli peppers is costly and affects farmers' health, requiring improvement in science and technology. This article presents the hardware design and recognition algorithm using computer vision to remove broken chilli, missing stems, and anthracnose. OpenCV and Python were implemented for thresholding and applies convolutional neural networks like ResNet and MobileNet to detect anthracnose. Microcontroller ESP32 was used to receive data regarding the status of the chillies and to control the sorting actuators. Processing time for thresholding and anthracnose detection was 0.08 seconds, with an average classification cycle of 1 second. Experiments show that the system has the potential for practical application in farmer households or traders.

Keywords: Computer vision, convolutional neural network, OpenCV, Python, thresholding

1. GIỚI THIỆU

Việc phân loại ớt nguyên vẹn không bị gãy đôi, mất cuống, không bị thán thư sau thu hoạch là cần thiết, nhằm giúp người nông dân bán được giá theo chuẩn thương mại. Trong thực tế việc phân loại tình trạng ớt tươi đạt chuẩn phụ thuộc hoàn toàn vào kinh nghiệm của nông dân. Việc ngồi lâu tiếp xúc trực tiếp và thường xuyên trong thời gian dài dễ gây kích ứng nóng rát ở da, mũi và đường hô hấp, ảnh hưởng đến sức khỏe, các mạch máu ngoại vi, chức năng giãn mạch của động mạch chân do công nhân phải ngồi quá lâu để phân loại ớt (Credeur et al., 2019). Bên cạnh đó, chi phí để trả cho nhân công thực hiện việc phân loại là rất lớn đối với quy mô trồng ớt thu hoạch đồng loạt. Vì vậy, việc sử dụng thiết bị máy móc hỗ trợ để phân loại tình trạng ớt thán thư, ớt mất cuống hay ớt bị gãy trong quá trình thu hoạch là một giải pháp tối ưu nhằm tiết kiệm thời gian và chi phí nhân công, mang lại hiệu quả kinh tế cao cho nhà nông và thương lái.

Thị trường ớt ở Việt Nam đang phát triển mạnh mẽ và có tiềm năng lớn để xuất khẩu nếu ớt tươi đạt chuẩn thương mại. Theo nghiên cứu của Nguyen et al. (2017), Đồng bằng sông Cửu Long (ĐBSCL) là một trong những khu vực sản xuất ớt lớn nhất ở Việt Nam. Trong đó, việc tăng cường hiệu quả tài chính cho các nông dân trồng ớt đã được nghiên cứu và đưa ra các giải pháp nhằm nâng cao giá trị sản phẩm ớt. Cụ thể, lợi nhuận của mô hình trồng ớt của nông dân trung bình đạt 158 triệu đồng/hecta và hiệu quả tài chính của nông dân trồng ớt khá cao. Tỷ suất lợi nhuận trên doanh thu đạt 30,8%, và tỷ suất lợi nhuận trên tổng chi phí đạt 135%. Nghiên cứu của Vo et al. (2016) cũng cho thấy sự phát triển và tầm quan trọng của chuỗi giá trị sản phẩm nông nghiệp trong khu vực ĐBSCL, từ sản xuất, chế biến đến tiêu thụ các loại rau quả sau thu hoạch. Tuy nhiên, sản phẩm này đang thừa cung về số lượng nhưng lại thiếu chất lượng để xuất khẩu. Nông sản có chất lượng thấp thì khó thâm nhập được vào các thị trường khắc khe như EU, Mỹ, Nhật Bản. Vì vậy, nghiên cứu về máy phân loại ớt bị thán thư, gãy cuống, mất cuống là một phương án khả thi giúp phân loại sản phẩm ớt tươi nguyên nhằm nâng cao giá trị của chuỗi sản phẩm nông nghiệp tại khu vực ĐBSCL và cả nước.

Tuy nhiên, cũng như nhiều loại cây trồng khác, chất lượng và năng suất ớt bị đe dọa nghiêm trọng bởi các loại dịch bệnh khác nhau như: thán thư, đốm trắng lá, héo xanh, thối đọt non... Nguy hiểm nhất là bệnh thán thư do nấm *Colletotrichum* (Hình 1) gây

ra (*Chilli Anthracnose (Colletotrichum Capsici)*, 2018).



Hình 1. Loại nấm Colletotrichum

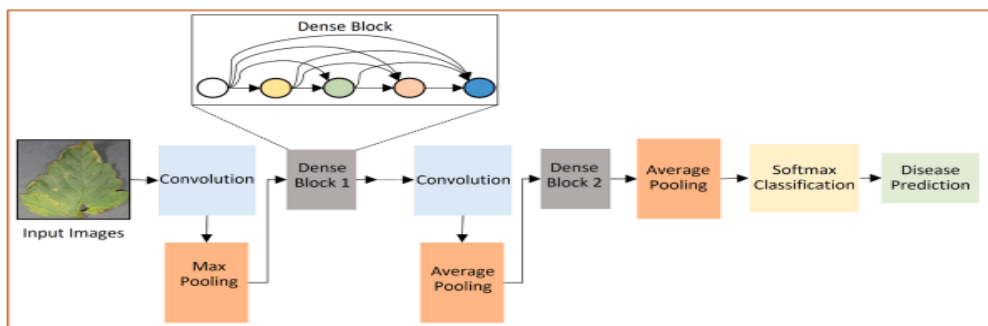
Chủng nấm này rất đa dạng gồm nhiều loài và gây hại trên hầu hết các loại cây trồng, gây thiệt hại rất lớn cho các hộ trồng ớt khiến sản lượng có thể giảm từ 70 đến 80% (Tran et al., 2016). Bên cạnh đó, ớt sau khi đã thu hoạch thường xuất hiện tình trạng gãy đôi hoặc mất cuống do quá trình thu hoạch và vận chuyển còn thô sơ. Hình 2 là hình thái bên ngoài phổ biến của ớt sau khi thu hoạch.

Các loại ớt cần nhận dạng để phân loại			
Ớt thán thư	Ớt gãy đôi	Ớt mất cuống	Ớt đạt chuẩn

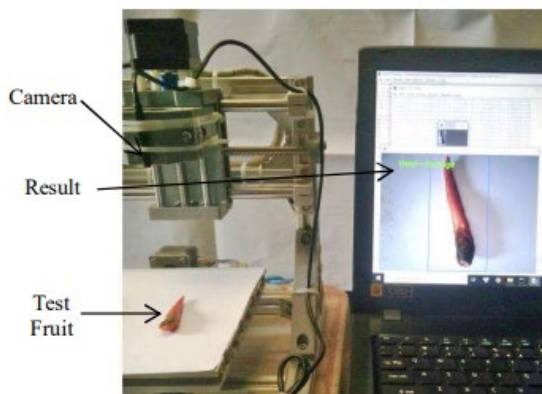
Hình 2. Một số tình trạng trái ớt chỉ thiên

Đã có một số nghiên cứu về việc phát hiện bệnh trên cây trồng. Hình ảnh tổng hợp của Abbas et al. (2021) về phát hiện bệnh cây cà chua bằng phương pháp học chuyên giao với C-GAN dựa trên giải thuật học sâu đã cho thấy kết quả hứa hẹn trong việc phát hiện bệnh cây trồng. Mô hình của nhóm tác giả này đạt độ chính xác 98,16%, 95,08% và 94,34% trên tập dữ liệu gốc PlantVillage lần lượt cho các nhiệm vụ phân loại 5 lớp, 7 lớp và 10 lớp tương ứng. Khi kết hợp với tập dữ liệu gốc và hình ảnh tổng hợp, mô hình đạt độ chính xác lần lượt là 99,51%, 98,65% và 97,11% tương ứng cho các nhiệm vụ phân loại 5 lớp, 7 lớp và 10 lớp. Kiến trúc của mô hình được mô tả ở Hình 3.

Mặt khác, dựa trên một số nghiên cứu của Huynh (2020), việc xác định các tổn hại do bệnh gây ra trên quả ớt tươi đã loại bỏ cuống cho thấy rằng hệ thống sử dụng máy ảnh để chụp hình các trái ớt tươi đã được rút cuống (Hình 4).



Hình 3. Kiến trúc DenseNet (Abbas et al., 2021)



Hình 4. Ảnh chụp và xử lý trên trái ớt

Bên cạnh đó, cũng trong nghiên cứu của Huynh et al. (2021), việc xác định vết nứt trên hệ thống quả ớt tươi (*Capsicum*) đã loại bỏ cuống sẽ có một bộ truyền động tròn gồm bánh răng và băng tải để định vị các trái ớt (Hình 5). Sau đó, cơ cấu này sẽ di chuyển để đưa trái ớt vào vị trí chụp ảnh.

Dựa trên một số tiềm năng thị trường ớt ở Việt Nam như: sự bất tiện của việc phân loại ớt thủ công, ảnh hưởng tiêu cực về sức khỏe của việc lựa ớt, ý tưởng nghiên cứu về việc thiết kế máy phân loại tình trạng trái ớt không bị lỗi, hư được tiến hành. Nghiên cứu này kết hợp áp dụng một số thuật toán thị giác

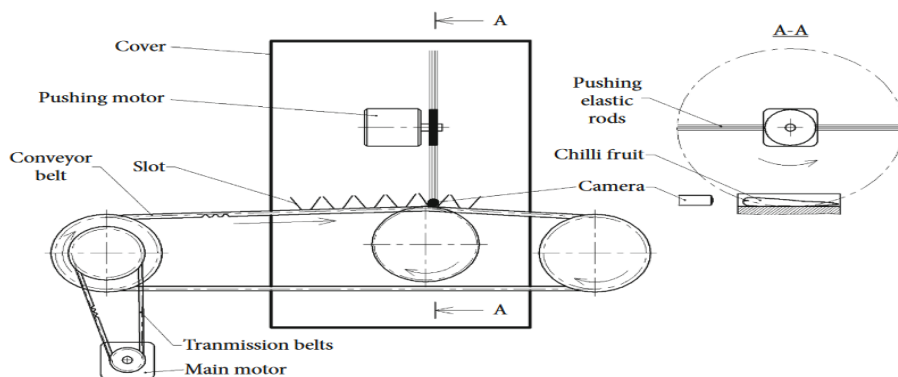
máy tính. Máy phân loại này không chỉ giúp giảm thiểu sự tiếp xúc trực tiếp với ớt, mà còn nâng cao hiệu suất và giảm chi phí lao động. Bằng việc áp dụng công nghệ học sâu và các thuật toán thị giác máy tính, máy phân loại có khả năng nhận diện và phân loại các tình trạng ớt như thán thư, gãy cuống, và mất cuống một cách chính xác và hiệu quả. Điều này giúp cải thiện chất lượng sản phẩm ớt đưa ra thị trường, nâng cao hiệu quả kinh tế, đồng thời bảo vệ sức khỏe người dân.

2. PHƯƠNG PHÁP NGHIÊN CỨU

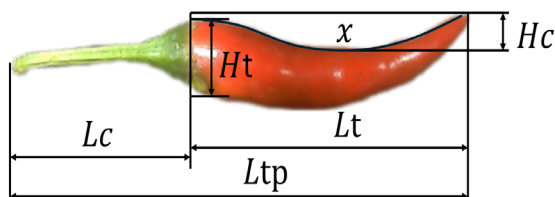
2.1. Nhận dạng từng phần trái ớt

Với mục tiêu phân biệt được 3 loại ớt: thán thư, gãy đôi và gãy cuống, các thông số cơ bản của trái ớt đã được đo đạc thực tế tại cơ sở buôn bán ớt ở xã Tân Hòa, huyện Lai Vung, tỉnh Đồng Tháp. Mặt khác, các số liệu về trái ớt của tác giả Huynh (2019) cũng được tham khảo. Các thông số ớt tiêu chuẩn được thể hiện qua Hình 6.

Trong đó H_c : Khoảng cách giữa phần nhô cao nhất của thân ớt so với đáy của đường cong x , H_t : Khoảng cách phần đầu của thân ớt, L_c : Khoảng cách của cuống ớt, L_t : Khoảng cách của thân ớt, L_{tp} : Khoảng cách của thân ớt và trái ớt cộng lại, x : đường cong tính từ phần đầu của thân ớt đến phần cao nhất của đuôi thân ớt.



Hình 5. Cơ cấu xác định vết nứt trên ớt (Huynh et al., 2021)



Hình 6. Thông số đo đặc của trái ớt

Kết quả khảo sát 500 trái ớt chỉ thiên F1 đạt chuẩn thương mại tại xã Tân Hòa, huyện Lai Vung, tỉnh Đồng Tháp được thể hiện qua Bảng 1.

Bảng 1. Kết quả đo đặc loại ớt chỉ thiên F1

Nội dung	Hc	Ht	Lc	Lt	Ltp
Khoảng cách trung bình (cm)	0,7	1,1	4,3	5	10,5
Khoảng cách tối đa (cm)	0,8	1,3	4,5	7	11
Khoảng cách tối thiểu (cm)	0,3	0,8	2,2	3,5	6

Như vậy, việc thiết kế phần cứng, xây dựng giải thuật được đề cập ở các phần tiếp theo sẽ dựa trên kết quả đo đặc trên Bảng 1.

2.1.1. Xử lý hình ảnh dựa trên OpenCV

OpenCV (*OpenCV: OpenCV-Python Tutorials*, n.d.) là viết tắt của "*Open Source Computer Vision Library*", là một thư viện mã nguồn mở dành cho xử lý ảnh và thị giác máy tính. Trong OpenCV, ảnh được biểu diễn dưới dạng các ma trận, tương tự như MATLAB. Ảnh số là tập hợp hữu hạn các điểm ảnh (pixel) với các giá trị màu hoặc độ xám cụ thể. Mỗi điểm ảnh đại diện cho một phần tử tại tọa độ (x, y) của ma trận ảnh với giá trị màu hoặc mức xám riêng biệt. Khoảng cách và kích thước của các điểm ảnh

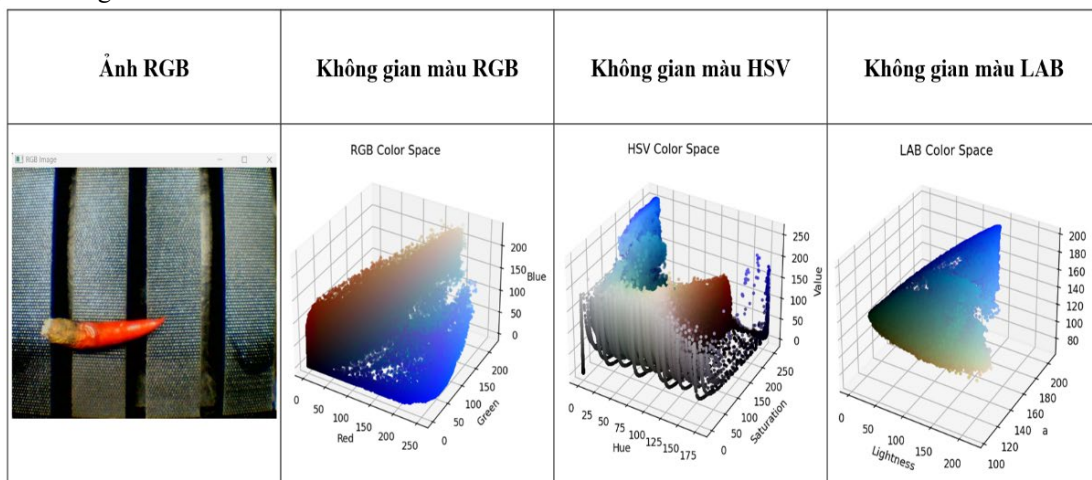
được điều chỉnh để tạo ra cảm giác liên tục về không gian và màu sắc khi nhìn bằng mắt thường, giúp ảnh số trông như ảnh thật. OpenCV cung cấp nhiều phương pháp xử lý ảnh, từ các thao tác cơ bản như thay đổi kích thước, cắt ảnh, đến các thuật toán nâng cao như nhận diện khuôn mặt, phát hiện đối tượng.

2.1.2. Không gian màu

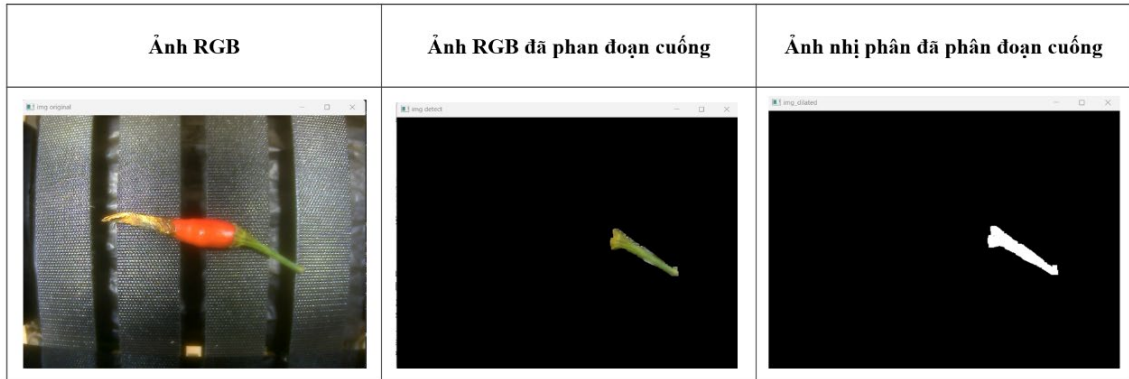
Không gian màu là một khái niệm quan trọng trong lĩnh vực đồ họa và xử lý ảnh, thể hiện cách mà màu sắc được biểu diễn và tương tác trong không gian (Hình 7). Nghiên cứu này chỉ xem xét hai không gian màu được sử dụng là HSV (màu sắc, độ bão hòa, giá trị độ sáng) và LAB (độ sáng, kênh a, kênh b) để phát hiện màu đỏ và màu xanh trên trái ớt, tương tự trong nghiên cứu của Dairath et al. (2023) về việc phân ngưỡng quả xoài dựa trên màu sắc để xử lý.

RGB là không gian màu cơ bản nhất gồm ba màu: R (Màu đỏ), G (xanh lục) và B (xanh lơ) để mô tả tất cả các màu sắc khác. Tuy nhiên, nó không phù hợp cho việc tìm kiếm ảnh được đề cập trong nghiên cứu của Pham et al. (2015). LAB là một không gian màu, gồm Lightness (Độ sáng), A (đỏ - xanh) và B (vàng - lam). HSV là viết tắt của Hue (màu sắc), saturation (độ bão hòa) và value (giá trị hay độ sáng).

Do ưu và nhược điểm của mỗi không gian màu khác nhau nên có tác động khác nhau lên việc phân ngưỡng hình ảnh, do đó trong nghiên cứu này, ảnh từ không gian màu RGB được chuyển sang HSV và LAB với cú pháp `cv2.cvtColor` ("*Ảnh cần chuyển đổi không gian màu*", "*Không gian màu cần chuyển đổi*"), trong đó `cv2` là lệnh gọi thư viện OpenCV. Bên cạnh đó, việc tách từng kênh màu được thực hiện để thuận tiện cho việc phân ngưỡng ảnh.



Hình 7. Biểu diễn các không gian màu cơ bản



Hình 8. Kết quả phân ngưỡng

2.1.3. Phân đoạn ảnh

Kỹ thuật phân đoạn ảnh đóng vai trò quan trọng trong các bài toán xử lý ảnh, với mục tiêu là chia ảnh đầu vào thành các vùng đồng nhất. Mỗi vùng đồng nhất bao gồm các điểm ảnh có chung một số tính chất như màu sắc hoặc kết cấu. Phương pháp nghiên cứu trong bài báo này dựa trên nguyên tắc thử và sai để xác định cường độ tối ưu trên các mặt phẳng đã tách. Phương pháp này đơn giản và hiệu quả cho việc phân đoạn sơ bộ.

Cách thực hiện phân đoạn hình ảnh: Đầu tiên ảnh RGB thu được từ quá trình làm mờ bằng bộ lọc GaussianBlur. Ảnh sẽ chuyển sang một không gian màu khác tùy theo nhu cầu sử dụng. Phân tích các đặc điểm giá trị pixel của tổng thể cường độ, thân ớt và nền trong ảnh cho thấy có sự khác biệt đáng kể về thành phần màu sắc giữa tổng thể cường độ, thân ớt và nền trong ảnh, trong khi sự khác biệt về thành phần độ bão hòa và độ sáng nhỏ hơn. Do đó, bằng cách đặt giá trị tối đa và tối thiểu cho ba kênh, kết hợp với một mặt nạ đánh dấu ban đầu với tất cả các giá trị là 0, ma trận nhị phân các khoảng màu của cường độ và thân ớt được tạo ra. Khoảng màu của cường độ ớt được xác định với $\begin{Bmatrix} Max(188, 120, 188) \\ Min(148, 100, 154) \end{Bmatrix}$. Do màu đỏ có 2 khoảng trên và dưới khác nhau nên ta cần xác định 2 tập màu cho thân ớt với khoảng thứ nhất là $\begin{Bmatrix} Max(11, 255, 255) \\ Min(0, 114, 140) \end{Bmatrix}$, và khoảng thứ hai là $\begin{Bmatrix} Max(180, 255, 255) \\ Min(166, 75, 75) \end{Bmatrix}$. Từ đó từng cặp giá trị màu được duyệt qua dưới dạng cận dưới và cận trên, sử dụng hàm `cv2.inRange` để tạo mặt nạ cho mỗi khoảng màu và kết hợp lại với nhau bằng `cv2.bitwise_or`.

Tiếp theo, tạo một ma trận được gọi là hạt nhân có kích thước 5×5 có tất cả các giá trị pixel bằng một được xác định bằng thư viện `NumPy` trong `Python`. Hàm giãn nở sử dụng ba đối số: Đối số đầu tiên là hình ảnh nhị phân đầu vào, đối số thứ hai là hạt nhân

được xác định ở bước trước và đối số thứ ba là số lần lặp cần thiết. Mục đích của việc giãn nở này là giảm tình trạng mất mát, dứt đoạn từ việc phân đoạn. Cuối cùng là sử dụng mặt nạ đã giãn nở để áp dụng lên ảnh gốc bằng cú pháp `cv.bitwise_and`, kết quả tạo ra ảnh hiển thị các vùng có màu sắc phù hợp với ngưỡng đã xác định. Điều này giúp loại bỏ được những điểm không hoàn hảo và biến dạng liên quan đến nhiễu và kết cấu. Hình 8 minh họa ảnh gốc, ảnh RGB và ảnh nhị phân sau khi được phân đoạn theo ngưỡng tối ưu.

2.1.4. Tính diện tích vật thể

Việc tính diện tích vật thể giúp xác định được số pixel của vật thể được xác định. Từ giá trị pixel đó, ta có thể chuyển sang các đơn vị đo lường khác như milimét vuông (mm^2) hoặc centimet vuông (cm^2), để dùng trong các nghiên cứu và phân tích.

Đầu tiên ảnh được đọc và xử lý, để chuẩn bị cho việc phát hiện viền; ảnh thường được chuyển đổi sang thang màu xám và có thể áp dụng một số phương pháp làm mờ để giảm nhiễu. Sau đó, thuật toán *Canny edge detector* được sử dụng để xác định viền của các đối tượng. Thuật toán này hoạt động qua một loạt các bước để xác định và phát hiện các viền chính xác trong ảnh. Bước đầu tiên của thuật toán là giảm nhiễu bằng cách sử dụng bộ lọc Gaussian. Bước này giúp làm mịn ảnh và loại bỏ nhiễu không mong muốn, tạo điều kiện tốt để phát hiện các biên.

Tiếp theo, thuật toán tính toán gradient cường độ và hướng tại mỗi điểm ảnh trong ảnh được thực hiện. Gradient cường độ là sự thay đổi lớn nhất trong cường độ màu sắc, còn hướng gradient chỉ ra hướng của sự thay đổi đó. Công thức cơ bản cho gradient cường độ G và hướng θ được mô tả lần lượt ở biểu thức (1) và (2).

$$G = \sqrt{G_x^2 + G_y^2} \quad (1)$$

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (2)$$

Các tham số: G_x , G_y là gradient theo trục x và y. Sau đó, thuật toán áp dụng non-maximum suppression để loại bỏ các điểm không phải là cực trị. Điều này đảm bảo rằng chỉ có các điểm cực trị thực sự đại diện cho các biên sắc nét được giữ lại.

Bước tiếp theo là sử dụng ngưỡng kép để xác định các biên mạnh và yếu. Các điểm cường độ cao hơn một ngưỡng được coi là biên mạnh, trong khi các điểm có cường độ thấp hơn một ngưỡng khác được coi là biên yếu. Cuối cùng, thuật toán theo dõi biên bằng cách loại bỏ các biên yếu không liên quan đến biên mạnh. Điều này giúp loại bỏ các biên không mong muốn và chỉ giữ lại các biên quan trọng. Thuật toán này giúp tìm ra các cạnh trong ảnh, là bước quan trọng để xác định các đường viền của vật thể.

Sau khi các cạnh đã được xác định, cú pháp *cv2.findContours* được thực hiện để tìm tất cả các đường bao của đối tượng. Mỗi đường bao được lưu trữ dưới dạng một danh sách các điểm. Với mỗi đường bao tìm được, diện tích bên trong đường bao có thể được tính toán theo cú pháp của hàm *cv2.contourArea*. Hàm này trả về diện tích của vùng bị bao quanh bởi đường bao, dùng để đo diện tích của các đối tượng.



Hình 9. Ảnh được xử lý để tính diện tích vùng

Hình 9 cho thấy ảnh sau khi được làm mờ và các phép toán cần thiết đã xác định: đường bao (đường màu hồng) quanh vật thể, các hình chữ nhật màu vàng (hộp giới hạn của đường bao). Công đoạn này giúp người dùng dễ dàng nhận biết vật thể.

2.2. Nhận dạng bệnh thán thư

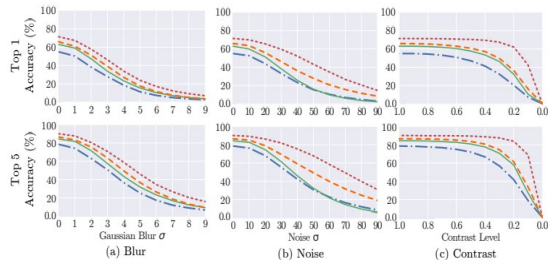
2.2.1. Xây dựng bộ dữ liệu

Nghiên cứu này thiết lập và sử dụng API phát hiện đối tượng của *TensorFlow* (*TensorFlow*, n.d.) để thực hiện, cụ thể là phát hiện đối tượng trong hình ảnh/video. Các công cụ phần mềm được sử dụng trong nghiên cứu gồm: Thư viện *Python* phiên bản

3.9, *TensorFlow* phiên bản 2.5.0, bộ công cụ *CUDA* phiên bản 11.2, *cuDNN* phiên bản 8.1.0, *anaconda*. Thiết bị được dùng để huấn luyện và đánh giá mô hình là máy tính xách tay *Acer Nitro 5*, bộ vi xử lý *Intel Core i5-11800H* thế hệ 11, có 8 nhân và 16 luồng, *RAM: 16 GB Card đồ họa rời NVIDIA GeForce RTX 1650*.

Các mẫu ốt dùng cho nghiên cứu này có nguồn gốc từ hộ dân tại ấp Hòa Định, xã Tân Hòa, huyện Lai Vung, tỉnh Đồng Tháp. Hơn 500 mẫu ốt được lấy ở đợt 1 (lúa ốt với trái to và năng suất cao nhất) và hơn 500 mẫu còn lại ở đợt 2 (lúa ốt với trái nhỏ tình trạng thán thư nhiều) có độ phân giải gốc là 640×480 pixel. Những trái ốt thu thập được đặt trên nền có màu đen, màu trắng và chụp ảnh bằng *Webcam full HD 1080p ACOME AWC11*. Trong quá trình chụp ảnh, có 2 giá trị được kiểm soát chặt chẽ là máy ảnh được duy trì vuông góc với trái ốt để tránh các lỗi góc có thể làm tăng sai số tính toán và khoảng cách chụp được kiểm soát để đảm bảo tất cả các trái ốt đều có kích thước tương tự nhau nhằm giảm thiểu tác động về kích thước khi phân loại.

Kích thước mẫu không đủ trong quá trình huấn luyện mô hình có thể dẫn đến tình trạng quá khớp (*overfitting*), một loạt các biện pháp tăng cường, chuẩn hóa dữ liệu đã được triển khai. Việc này nhằm giúp mở rộng kích thước mẫu và nâng cao tính tổng quát cũng như độ tin cậy của mô hình (Zhu et al., 2024). Nghiên cứu này sử dụng các kết hợp ngẫu nhiên của các kỹ thuật tăng cường dữ liệu như tự động định hướng hình ảnh dựa trên EXIF (*Exchangable image file format*). Đây là chuẩn để lưu giữ các thông tin thông số của ảnh như chế độ *flash*, ngày tạo, tình trạng ống kính, độ sáng. Thuộc tính orientation cần được chú ý (hướng của ảnh) để đưa tất cả các ảnh trong tập dữ liệu về cùng một hướng. Việc thay đổi kích thước tập dữ liệu từ kích thước 640×480 thành 640×640 (kéo dài hình ảnh thô) cũng cần được thực hiện để lấp đầy kích thước đầu ra mong muốn. Bên cạnh đó, kỹ thuật nhiễu lên đến 0,1% lượng pixel, làm mờ lên đến 2,2 giá trị pixel cũng được áp dụng ngẫu nhiên cho hình ảnh. Cụ thể, dựa trên nghiên cứu từ Đại học bang Arizona (Dodge & Karam, 2016), tác giả đã xem xét tác động của các khiếm khuyết hình ảnh khác nhau đối với các mô hình phân loại. Trong số các phương pháp được thử nghiệm, làm mờ và nhiễu có ảnh hưởng bất lợi nhất đến việc phân loại độ chính xác top 1 (đo lường tỷ lệ phần trăm các dự đoán chính xác nhất của mô hình khớp với nhãn đúng của dữ liệu kiểm tra) và top 5 (đo lường tỷ lệ phần trăm các mẫu mà nhãn đúng nằm trong số 5 nhãn được mô hình dự đoán với xác suất cao nhất).



Hình 10. Tác động của các khiếm khuyết hình ảnh khác nhau đối với các mô hình phân loại

Kỹ thuật tiếp theo được áp dụng là thêm khả năng thay đổi độ rực sáng cho hình ảnh để giúp mô hình linh hoạt hơn trước những thay đổi về cài đặt ánh sáng và máy ảnh với mức tăng giảm độ sáng là $\{Max(15\%), Min(-15\%)$, với độ rực là $\{Max(10\%), Min(-10\%)$. Cuối cùng là cân bằng độ tương phản (histogram equalization) dựa trên biểu đồ của hình ảnh để cải thiện khả năng chuẩn hóa và phát hiện đường thẳng trong các điều

kiện ánh sáng khác nhau. Một số mẫu tăng cường dữ liệu được hiển thị trong Hình 12. Kích thước tập dữ liệu đã được mở rộng một cách hiệu quả, cuối cùng là tổng số 9584 hình ảnh đã được chuẩn hóa và tăng cường.

Để tối đa hóa hiệu quả học tập và độ chính xác dự đoán của mô hình, các hình ảnh nâng cao được phân bổ và chia thành 4 tập dữ liệu bao gồm: tập 1198 ảnh được áp dụng EXIF: orientation; 1198 ảnh thay đổi kích thước và EXIF: orientation; 3594 ảnh áp dụng EXIF: orientation, mờ, nhiễu và tăng giảm độ sáng; và 3594 tập ảnh áp dụng EXIF: orientation, mờ, nhiễu và tăng giảm độ sáng, tăng giảm độ rực, thay đổi kích thước và cân bằng độ tương phản.

Các tập huấn luyện và kiểm tra với tỷ lệ 9:1. Tập huấn luyện được sử dụng để học ban đầu và điều chỉnh tham số của mô hình. Bộ kiểm tra được sử dụng để đánh giá hiệu suất của mô hình trên dữ liệu mới.

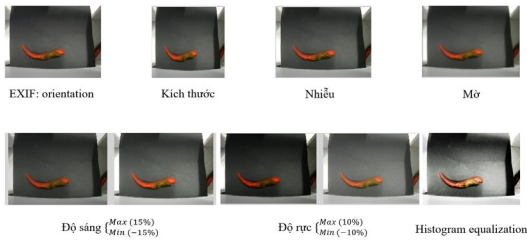


Hình 11. Tập lưu nhãn tương ứng

Bước tiếp theo là việc gán nhãn đối tượng. Trong bài toán phát hiện đối tượng, đây là phần quan trọng để chuẩn bị dữ liệu huấn luyện cho mô hình. Điều này đảm bảo rằng mô hình có thể học được các đặc trưng của các đối tượng một cách chính xác. Gán nhãn đối tượng giúp xác định vị trí và nhãn của các đối tượng trong hình ảnh, bằng cách tạo và hiệu chỉnh các khung bao để xác định ranh giới của đối tượng cũng như nhãn tương ứng và lưu vào tập tin có phần mở rộng *.xml* với mỗi ảnh tương ứng. Nội dung tập tin gồm các thành phần quan trọng: chiều rộng, chiều cao của hình ảnh, tên đối tượng được gán nhãn, tọa độ x của góc trên bên trái hộp giới hạn, tọa độ y của góc trên bên trái hộp giới hạn, tọa độ x

của góc dưới bên phải hộp giới hạn, tọa độ y của góc dưới bên phải hộp giới hạn.

Hình 12 mô tả các thông tin như chiều dài, chiều rộng của nhãn được gán, cũng như vị trí của nhãn được thể hiện. Mỗi nhãn được gán một mã số duy nhất và mỗi dòng trong tệp tương ứng với một nhãn, được định dạng theo cú pháp *Protobuf Text*. Cấu trúc cơ bản của một dòng trong tệp *label_map.pbtxt* bao gồm hai trường chính: một là mã nhận dạng của nhãn và hai là tên của nhãn.



Hình 12. Ảnh được chuẩn hóa và tăng cường

Bước tiếp theo của quá trình huấn luyện mô hình là chuyển đổi tệp *.xml* sang tệp *.record*. Đây là bước cần thiết trong quá trình chuẩn bị dữ liệu huấn luyện cho mô hình nhận diện đối tượng sử dụng *TensorFlow Object Detection API*. Tệp *.record* là định dạng dữ liệu mà *TensorFlow Object Detection API* yêu cầu. Cấu trúc này định dạng lưu trữ dữ liệu nhị phân mà *TensorFlow* sử dụng để tăng hiệu suất đọc và ghi dữ liệu trong quá trình huấn luyện mô hình. Các tệp *TFRecord* có thể chứa dữ liệu dạng văn bản, ảnh, âm thanh và nhiều loại dữ liệu khác.

Để bắt đầu quá trình huấn luyện, mô hình huấn luyện trước trên *TensorFlow 2 Detection Model Zoo* cần được tải về để chỉnh sửa tệp *pipeline.config* phù hợp với nhu cầu huấn luyện. Cụ thể tệp *checkpoint* chứa trọng số của mô hình sau quá trình huấn luyện. Trọng số này bao gồm các thông số của mạng nơ-ron như trọng số của các lớp, các thông số của các bộ lọc và các tham số tối ưu hóa. Khi huấn luyện một mô hình, *TensorFlow* sẽ định kỳ lưu lại các trọng số của mô hình vào tệp *checkpoint*. Điều này cho phép tiếp tục huấn luyện từ nơi đã dừng lại, hoặc sử dụng trọng số đã được huấn luyện để triển khai mô hình trên dữ liệu mới.

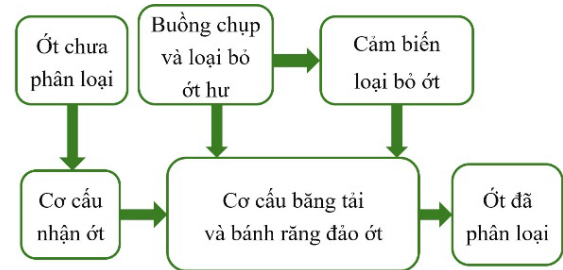
2.2.2. Tiến hành huấn luyện mô hình

Sau khi đã điều chỉnh các trọng số của mô hình huấn luyện, bước tiếp theo là tiến hành huấn luyện mô hình dựa trên thư viện *Tensorflow*. Mô hình đã huấn luyện có thể được chuyển sang định dạng *TensorFlow Lite*. Đây là phiên bản nhẹ của *TensorFlow* được thiết kế đặc biệt cho việc triển khai mô hình trên các thiết bị có tài nguyên hạn chế như điện thoại di động, thiết bị IoT và các thiết bị nhúng khác.

Sau khi đã huấn luyện được mô hình, một ảnh RGB được đưa vào để nhận dạng và xuất ra các thông tin nhận dạng như hộp giới hạn, lớp nhận dạng, độ chính xác và số lượng hộp nhận dạng đối tượng tương ứng.

3. XÂY DỰNG HỆ THỐNG

Sau các bước xây dựng thuật toán phân ngưỡng và huấn luyện các mô hình máy học, việc còn lại là xây dựng hệ thống hoàn chỉnh để tích hợp các mô hình này vào. Hệ thống gồm các thành phần chính được thể hiện ở Hình 13, bao gồm: cơ cấu nhận ốt được sử dụng với chức năng chính là lấy ốt từ khoang chứa và đảm bảo ốt được phân bố đều trên băng tải; cơ cấu băng tải và bánh răng đảo ốt có chức năng chính là di chuyển ốt đi qua từng khâu xử lý và đảo bề mặt của ốt 180 độ; cảm biến loại bỏ ốt có nhiệm vụ phát hiện sự có mặt của ốt để đưa ra tín hiệu điều khiển cho cơ cấu chấp hành (cơ cấu cần gạt, cơ cấu đảo ốt); buồng chụp và loại bỏ ốt hư có chức năng hỗ trợ quá trình chụp ảnh.



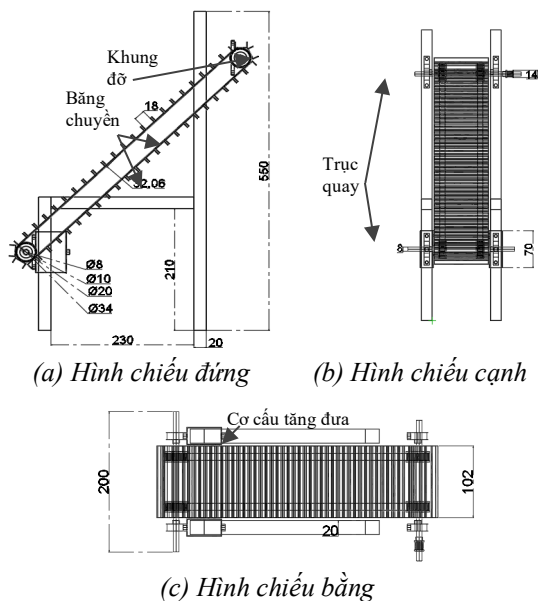
Hình 13. Sơ đồ khối phản cứng

3.1. Cơ cấu nhận ốt

Mục đích: tạo ra một giải pháp giúp lấy ốt từ khoang chứa và đảm bảo ốt được phân bố đều trên bề mặt băng tải khi di chuyển qua các khâu xử lý. Cơ cấu nhận ốt được thiết kế như Hình 14 với các thành phần như sau:

Khung đỡ chịu đựng toàn bộ sức nặng của băng chuyền và phần lớn hệ thống. Trục quay đóng vai trò liên kết giữa khung đỡ với băng chuyền thông qua vòng bi giúp quá trình tải ốt mượt mà.

Băng chuyền là bộ phận nhận ốt và chuyển lên băng tải giúp ốt được nằm đều lên băng tải. Cơ cấu tăng đưa là bộ phận giữ trục quay được cố định vào khung đỡ. Do băng chuyền được thiết kế dùng dây curoa nên sẽ xảy ra co giãn nên cần bộ phận tăng đưa nhằm giảm ảnh hưởng tiêu cực của hiện tượng trên.

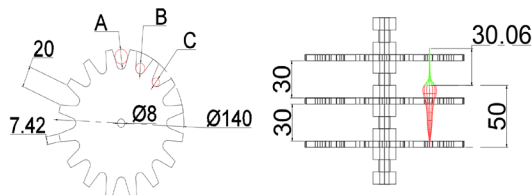


Hình 14. Bản vẽ ba chiều cơ cấu nhận ốt
(mm)

3.2. Cơ cấu băng tải và bánh răng đảo ốt

Mục đích của băng tải và bánh răng đảo ốt là tạo ra một giải pháp giúp lật được ốt 180 độ khi đi qua cơ cấu, tải được ốt theo chiều ngang một cách ổn định. Khi ốt được chuyển lên từ cơ cấu nhận ốt, ốt sẽ được vận chuyển từ từ đi qua 2 buồng chụp ảnh và bánh răng đảo ốt. Ốt được di chuyển đến bánh răng của cơ cấu lật. Ốt sẽ đi vào phần răng lõm của bánh răng và có cảm biến nhận biết tín hiệu đã có ốt để kích hoạt cơ cấu quay đến khoảng răng trống tiếp theo. Ốt được đảo 180 độ khi đi qua bánh răng.

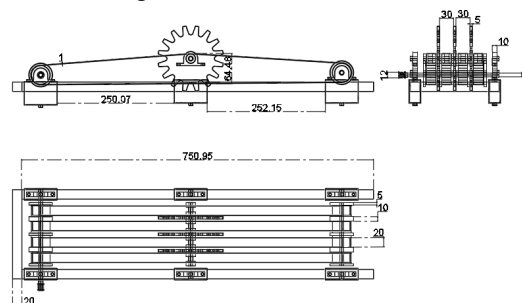
Thiết kế bánh răng dao ớt: Nhiệm vụ chính là dao được trái ớt 180 độ. Trong quá trình dao, ớt cần được đảm bảo định vị trên bánh răng. Thông số của bánh răng được thiết kế để phù hợp với trái ớt chỉ thiên F1 dựa trên Bảng 2. Bánh răng được thiết kế và vẽ trên phần mềm Autocad 2022 (Hình 15).



Hình 15. Hình chiếu cạnh bánh răng đảo ốt

Trong đó: A là khoảng cách *Ht* tối đa, B là khoảng cách *Ht* trung bình, C là khoảng cách *Ht* tối thiểu (Bảng 1). Bảng tải được thiết kế với mục đích chuyên ốt từ đầu ra của cơ cấu nhân ốt rồi di chuyển

qua buồng chụp và xử lý ảnh thứ nhất đến bánh răng đảo ớt, tại đây ớt được lật 180 độ nhờ vào bánh răng đảo ớt rồi di chuyển đến buồng chụp và xử lý ảnh thứ hai. Hình 16 thể hiện mô hình cơ cấu bánh răng đảo ớt và băng tải ớt.



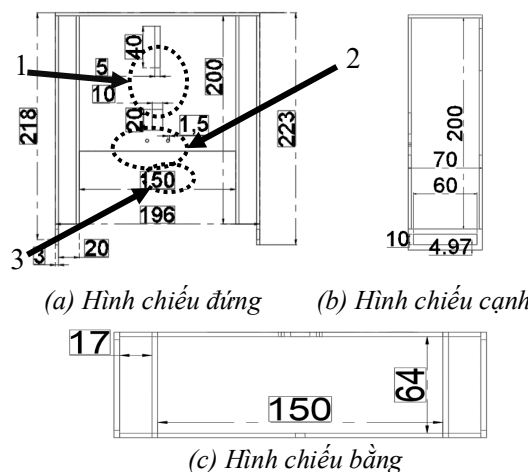
Hình 16. Bản vẽ băng tải và bánh răng đảo ốt

3.3. Cảm biến loại bỏ ốt bị lỗi

Tín hiệu từ cảm biến phát hiện ớt được gửi liên tục đến vi điều khiển để nhận dạng và loại bỏ ớt không đạt chuẩn. Nếu cảm biến phát hiện có ớt đi qua thì quá trình nhận dạng bắt đầu thực hiện, và khi có ớt bị lỗi thì vi điều khiển sẽ xuất tín hiệu điều khiển cần gạt thông qua động cơ servo để loại bỏ ớt không đạt chuẩn ra khỏi băng chuyền.

3.4. Buồng chup và loại bỏ ốt bi lỗi

Quá trình xử lý ảnh và phân loại ốt được diễn ra liên tục và ở mỗi thời điểm trong ngày nên ngoài việc đảm bảo camera chụp ảnh được ổn định, không gian bên trong buồng chụp ảnh không bị nhiễu hoặc thay đổi cường độ ánh sáng do các tác nhân môi trường, thì việc thiết kế buồng chụp ảnh là cần thiết. Hình 17 thể hiện thông số buồng chụp ảnh.



Hình 17. Bảng vẽ buồng chụp ảnh

Nguyên lý hoạt động: Ớt sau khi lên băng tải đi đến vị trí buồng chụp ảnh, được camera gắn tại vị trí 1 để chụp ảnh và gửi về máy tính xử lý. Nếu có tình trạng Ớt kém chất lượng, cảm biến tại vị trí 3 sẽ liên tục đọc tín hiệu khi phát hiện Ớt đi qua sẽ kích hoạt động cơ servo tại vị trí 2 xoay cần gạt để loại bỏ Ớt kém chất lượng.

4. KẾT QUẢ VÀ THẢO LUẬN

4.1. Độ chính xác thuật toán phân ngưỡng

Trong nghiên cứu này, để đảm bảo tính chính xác và độ tin cậy của các thông số trái Ớt chỉ thiên F1 thu được thông qua tính toán OpenCV, các thông số về tính trạng định lượng (Lt , Lc) của trái Ớt đã được chọn làm chỉ số xác nhận để xác minh nghiêm ngặt và phân tích lỗi. Một mẫu ngẫu nhiên gồm 100

Bảng 2. Kết quả sai số giữa các phép đo

	Chiều dài TB đo thủ công	Chiều dài TB đo bằng OpenCV	MAE	Std
Thân	4,80	4,66	0,15	0,15
Cuống	2,55	2,51	0,17	0,20

4.2. Độ chính xác các mô hình máy học

Hệ thống được tiến hành thí nghiệm với 4 tập dữ liệu đã được đề cập ở phần trên nhằm đánh giá hiệu suất của mô hình máy học với các cách tăng cường và chuẩn hóa dữ liệu khác nhau dựa trên ngưỡng Mean Average Precision (mAP). Thí nghiệm kiểm tra mAP gồm 200 hình ảnh mô tả các loại trái Ớt bị bệnh thán thư khác nhau, để kiểm tra độ chính xác thực tế của mô hình.

Biểu đồ ở Hình 18 thể hiện hiệu suất của mô hình SSD-MobilenetV2-FPNLite với các tập hình ảnh khác nhau. Trong đó: Basic là tập hình ảnh 1198 ảnh được áp dụng EXIF: orientation. Resize là 1198 ảnh thay đổi kích thước và EXIF: orientation. Tập V3 gồm 3594 ảnh áp dụng EXIF: orientation, mờ, nhiễu và tăng giảm độ sáng. Tập V4 gồm 3594 ảnh áp dụng EXIF: orientation, mờ, nhiễu và tăng giảm độ sáng, tăng giảm độ rực, thay đổi kích thước và cân bằng độ tương phản.

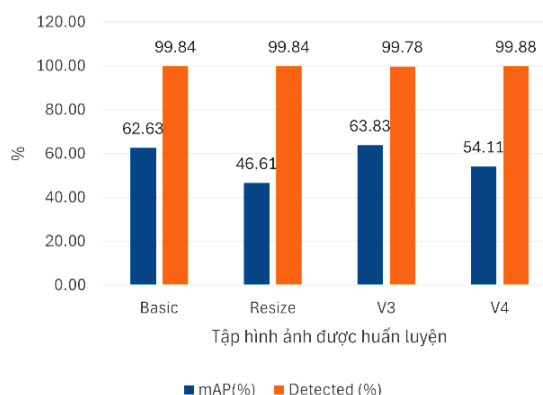
Dựa trên kết quả mAP và kết quả kiểm tra ở Hình 18, hiệu suất của mô hình trên từng bộ ảnh có thể được đánh giá như sau:

Tập Basic, mặc dù ít tiền xử lý, cho thấy một mức mAP tương đối cao. Đối với tập Resize, mặc dù có kích thước tiêu chuẩn, biểu đồ cho thấy sự giảm đáng kể về mAP . Điều này chỉ ra rằng việc thay đổi kích thước có thể ảnh hưởng đến các đặc điểm quan trọng trong hình ảnh, làm suy giảm khả năng của mô hình trong việc dự đoán các đối tượng nhỏ hoặc chi tiết. Tập V3 trên biểu đồ cho thấy việc thêm nhiễu

mẫu trái Ớt đã được lấy và các phép đo thủ công chính xác về Lt , Lc đã được tiến hành, trong đó Lt , Lc được đo bằng thước kẻ 30 cm. Kết quả đo thủ công được sử dụng làm tiêu chuẩn và được so sánh với kết quả tính toán bằng OpenCV (Bảng 2).

Kết quả sai số trung bình (Mean Absolute Error - MAE) của chiều dài thân là 0,145 và đối với cuống là 0,174. Độ lệch chuẩn của sai số (Standard Deviation - Std) của chiều dài thân: 0,146 và chiều dài cuống là 0,208. Kết quả thử nghiệm cho thấy phương pháp sử dụng OpenCV có mức độ chính xác khá cao với sai số trung bình nhỏ và độ lệch chuẩn thấp, thể hiện rằng phương pháp này đáng tin cậy cho việc đo lường chiều dài thân và cuống của các vật thể (trong trường hợp này là trái Ớt chỉ thiên F1).

và điều chỉnh hộp giới hạn một cách nhẹ nhàng (trong một khoảng từ -15% đến +15% độ sáng) có thể cải thiện sự mạnh mẽ của mô hình với phần trăm mAP cao nhất. Tập ảnh V4 mặc dù được tiền xử lý và tăng cường mạnh mẽ, nhưng lại không đạt được mAP cao nhất. Điều này chỉ ra hiện tượng quá mức, hoặc do một số biến thể có thể quá mạnh, do đó làm mờ đi các đặc điểm quan trọng.



Hình 18. Hiệu suất các tập ảnh khác nhau

Bảng 3 so sánh kết quả của các mô hình khác nhau được huấn luyện với tập ảnh V3. Dựa trên các thông số mAP , diện tích trung bình và thời gian xử lý cho từng mô hình cho thấy:

Bảng 3. So sánh các mô hình huấn luyện CNN

<i>Models</i>	<i>mAP</i>	<i>Thời gian (s)</i>
Faster R-CNN	71,24	0,94
SSD_RSN50_V1_FPN	69,45	0,51
Centernet_MB2V2-FPN	83,50	0,19
SSD-MB2V2-FPNLite (chuyển sang tflite)	57,97	0,05
SSD-MB2V2-FPNLite (không chuyển sang tflite)	76,03	0,06

Faster R-CNN: Mô hình này có *mAP* khá cao là 71,24, cho thấy khả năng phát hiện đối tượng tốt. Tuy nhiên, thời gian xử lý là 0,94 giây, cao hơn so với các mô hình khác, điều này có thể làm giảm hiệu quả trong các ứng dụng cần thời gian thực. SSD_RSN50_V1_FPN: Mô hình này có *mAP* là 69,45, thấp hơn một chút so với Faster R-CNN nhưng thời gian xử lý chỉ là 0,51 giây, nhanh hơn đáng kể so với Faster R-CNN. Điều này làm cho nó phù hợp hơn cho các ứng dụng cần đáp ứng nhanh.

Centernet_MB2V2-FPN: Đây là mô hình có *mAP* cao nhất trong số các mô hình được liệt kê là 83,50 và cũng có thời gian xử lý nhanh nhất là 0,19 giây. Điều này cho thấy mô hình này vượt trội cả về độ chính xác lẫn tốc độ, rất phù hợp cho các ứng dụng cần độ chính xác cao và thời gian phản hồi nhanh. SSD-MB2V2-FPNLite (chuyển sang dạng tflite): Mô hình này có *mAP* thấp nhất là 57,97 nhưng lại có thời gian xử lý cực kỳ nhanh là 0,05 giây. Sự đánh đổi giữa độ chính xác và tốc độ có thể làm cho mô hình này phù hợp với các ứng dụng không yêu cầu độ chính xác cao nhưng cần tốc độ xử lý nhanh. SSD-MB2V2-FPNLite (không chuyển sang dạng tflite): Mô hình này có *mAP* là 76,03, cao hơn phiên bản đã chuyển sang tflite với thời gian xử lý chỉ chậm hơn một chút là 0,06 giây. Điều này cho thấy việc không chuyển đổi sang tflite giúp cải thiện độ chính xác mà không ảnh hưởng đáng kể đến tốc độ xử lý.

Từ những phân tích trên, Centernet_MB2V2-FPN nổi bật với sự cân bằng tốt nhất giữa độ chính xác và tốc độ xử lý, làm cho nó trở thành lựa chọn tốt cho hầu hết các ứng dụng. Các mô hình SSD-MB2V2-FPNLite cho thấy sự đánh đổi giữa tốc độ và độ chính xác, trong khi Faster R-CNN và SSD_RSN50_V1_FPN có thể phù hợp cho các tình huống không yêu cầu tốc độ xử lý quá nhanh.

4.3. Kết quả thực nghiệm

Thí nghiệm được tiến hành với 1 kg trái ớt chỉ thiên F1 bao gồm các loại ớt nguyên vẹn, gãy đôi, mất cuống và thân thư. Kết quả của thí nghiệm được ghi nhận ở Bảng 4.

Bảng 4. Kết quả thí nghiệm tổng quát

	Đạt chuẩn	Gãy đôi	Mất cuống	Thân thư
MAE	3,2%	8%	2%	20%
Std	1,14	0,83	0,44	1

Kết quả thí nghiệm cho thấy sai số trung bình của việc phát hiện bệnh thân thư là cao nhất. Điều này được lý giải do ảnh hưởng của các yếu tố khách quan như: số lượng ớt hư do thân thư ít. Mặt khác, nhiều ánh sáng từ bên ngoài cũng gây ảnh hưởng đến kết quả phát hiện bệnh. Hơn nữa, ớt bị đốm sâu hay nhăn do héo cũng ảnh hưởng đến việc nhận diện ớt thân thư bị sai số nhiều hơn các lỗi mất cuống hay gãy đôi. Điểm nổi bật trong kết quả thí nghiệm là việc xác định ớt mất cuống với sai số MAE thấp nhất, chỉ 2%. Điều này cho thấy tính hiệu quả trong việc phân ngưỡng và xác định diện tích vật thể.

Hình 19 thể hiện một số kết quả đo được trong thí nghiệm, khi áp dụng thuật toán phân ngưỡng với OpenCV. Ký hiệu *S red* và *S green* lần lượt là diện tích đo được của đường bao của thân trái ớt và cuống trái ớt (theo đơn vị cm^2). Ký hiệu *length red* và *length green* lần lượt là chiều dài đo được của thân trái ớt và cuống trái ớt (theo đơn vị pixel).



Hình 19. Kết quả đo đạc bằng OpenCV

5. KẾT LUẬN

Nghiên cứu này đã đề xuất một thuật toán phân loại ớt bằng phương pháp xử lý ảnh. Sau quá trình thiết kế hệ thống và thực nghiệm, đề tài đã đạt được kết quả phù hợp với các mục tiêu nghiên cứu: thiết kế phần cứng phù hợp để điều khiển và phân loại ớt trực tiếp tại môi trường thực tế với các bộ phận linh

hoạt dễ lắp ráp và sử dụng; xây dựng được thuật toán phát hiện mẫu và loại bỏ ớt gãy đôi, mất cuống dựa trên mã nguồn mở OpenCV. Các kết quả nhận dạng và phát hiện tình trạng trái ớt bị lỗi khá hiệu quả. Nghiên cứu này có tiềm năng hiện thực hóa giúp người dân giảm bớt chi phí thuê mướn phân loại thủ công, từ đó bảo vệ sức khỏe của người lao động. Tuy nhiên, hệ thống nhận dạng và phân loại ớt thần thư cần được nghiên cứu phát triển thêm để có thể giảm

bớt sai số dưới 20% so với hiện tại. Hơn nữa, hệ thống có thể được mở rộng kho dữ liệu ảnh với nhiều loại bệnh khác nhau trên các loại trái cây khác nhau.

LỜI CẢM Ạ

Nhóm tác giả chân thành cảm ơn Trường Đại học Cần Thơ đã tài trợ kinh phí thực hiện đề tài NCKH Sinh viên cấp Trường, mã số TSV2023-113.

TÀI LIỆU THAM KHẢO (REFERENCES)

- Abbas, A., Jain, S., Gour, M., & Vankudothu, S. (2021). Tomato plant disease detection using transfer learning with C-GAN synthetic images. *Computers and Electronics in Agriculture*, 187, 106279. <https://doi.org/10.1016/j.compag.2021.106279>
- Nguyen, T. T. A., & Vo, T. T. L., (2017). Financial performance analysis of chilli farmers in the Mekong Delta (in Vietnamese). *Can Tho University Journal of Science*, 48, 87 (in Vietnamese). <https://doi.org/10.22144/ctu.jvn.2017.633>
- Chilli anthracnose (Colletotrichum capsici)*. (2018, October 25). IPM Images. <http://www.ipmimages.org/browse/detail.cfm?imgnum=5498970>
- Credeur, D. P., Miller, S. M., Jones, R., Stoner, L., Dolbow, D. R., Fryer, S. M., Stone, K., & McCoy, S. M. (2019). Impact of Prolonged Sitting on Peripheral and Central Vascular Health. *The American Journal of Cardiology*, 123(2), 260–266. <https://doi.org/10.1016/j.amjcard.2018.10.014>
- Dairath, M. H., Akram, M. W., Mehmood, M. A., Sarwar, H. U., Akram, M. Z., Omar, M. M., & Faheem, M. (2023). Computer vision-based prototype robotic picking cum grading system for fruits. *Smart Agricultural Technology*, 4, 100210. <https://doi.org/10.1016/j.atech.2023.100210>
- Dodge, S., & Karam, L. (2016). *Understanding How Image Quality Affects Deep Neural Networks* (arXiv:1604.04004). arXiv. <http://arxiv.org/abs/1604.04004>
- Pham, T. L. H., Tran, T. H., Nguyen, V. K., & Luu, T. H. (2015). Application of digital image processing in tracking the movement of mobile robots. *Can Tho University Journal of Science*, 39 (in Vietnamese).
- Huynh, Q. K., Nguyen, C. N., Vo, N., H. P., Le, H. T., Le, D. K. L., & Nguyen, V. C. (2020). Identification of the Damages Caused by Diseases on Fresh Destemmed Chili Fruits. *2020 12th International Conference on Knowledge and Systems Engineering (KSE)*, 126–130. <https://doi.org/10.1109/KSE50997.2020.9287653>
- Huynh, Q. K., Nguyen, C. N., Vo, N. H. P., Tran, N. P. L., Le, P. H., Le, D. K. L., & Nguyen, V.-C. (2021). Crack Identification on the Fresh Chilli (Capsicum) Fruit Destemmed System. *Journal of Sensors*, 2021, 1–10. <https://doi.org/10.1155/2021/8838247>
- Huynh, Q. K., Nguyen, V. C., Le, H., & Linh, L. (2019). Research on process of chilli processing and propose operating principle of fresh chilli stem classification system. *Can Tho University Journal of Science*, 55, 2-9 (in Vietnamese). <https://doi.org/10.22144/ctu.jvn.2019.032>
- Vo, T. T. L., (2016). *Assessment of agri-product value chains in the Mekong Delta: Problems and solutions*. <https://doi.org/10.22144/ctu.2016.jen.001>
- Tran, N. H., & Nguyen, T. L. T., (2016). Research on the preparation of Trichoderma sp. to control anthracnose caused by Colletotrichum spp. on chilli plants (Capsicum frutescens). *Can Tho University Journal of Science*, 45 (in Vietnamese). <https://doi.org/10.22144/ctu.jvn.2016.529>
- OpenCV: OpenCV-Python Tutorials*. (n.d.). Retrieved May 4, 2024, from https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html
- TensorFlow*. (n.d.). Retrieved May 5, 2024, from <https://www.tensorflow.org/>
- Zhu, Q., Fan, L., & Weng, N. (2024). Advancements in point cloud data augmentation for deep learning: A survey. *Pattern Recognition*, 153, 110532. <https://doi.org/10.1016/j.patcog.2024.110532v>