



Nghiên cứu phát triển hệ thống phần mềm nhận diện số lượng bọ hà trong bẫy phục vụ quản lý sâu bệnh hại trên cây khoai lang

Lê Hoàng Vũ¹, Trần Hà Phương¹, Trần Thị Thu Phương², Hà Quang Hưng³,
Phạm Minh Triển¹, Vũ Minh Trung^{1,*}

¹ Khoa Công nghệ Nông nghiệp, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội

² Khoa Nông học, Học viện Nông nghiệp Việt Nam

³ Khoa Công nghệ Thông tin, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội

*Email: trungvm@vnu.edu.vn

Tóm tắt

Bọ hà khoai lang (*Cylas formicarius*) là loài côn trùng gây hại phổ biến, gây tổn thất lớn về kinh tế do làm giảm năng suất và chất lượng củ. Nghiên cứu phát triển quy trình nhận diện đối tượng nhỏ mật độ cao nhằm cung cấp công cụ dự báo sớm dịch hại, tối ưu hóa sử dụng thuốc bảo vệ thực vật. Tập dữ liệu gồm 334 ảnh thực địa với mật độ từ 10 đến 100 cá thể mỗi ảnh được gán nhãn theo định dạng YOLO. Thử nghiệm so sánh hiệu năng giữa các phiên bản YOLOv8, v9, v10 và v11 được thực hiện trong cùng điều kiện dữ liệu và cấu hình huấn luyện. Kết quả cho thấy YOLOv11n đạt hiệu quả cao nhất với Precision 99,12%, Recall 97,78%, mAP@0.5 đạt 98,77% và mAP@[0.5:0.95] đạt 77,15%. Với cấu trúc tối ưu gồm 2,65 triệu tham số và 6,6 GFLOPs, mô hình này đảm bảo tính gọn nhẹ cho việc triển khai. Một hệ thống phần mềm đã được phát triển để tích hợp mô hình, thực hiện trực quan hóa kết quả và kiểm chứng khả năng ứng dụng thực tế trong giám sát dịch hại nông nghiệp.

Từ khóa: Bọ hà khoai lang, học sâu, phát hiện đối tượng, xử lý ảnh, YOLO.

Abstract

The sweet potato weevil (*Cylas formicarius*) is a common pest insect that causes significant economic losses by reducing both yield and tuber quality. This study develops a recognition pipeline for detecting small, densely distributed objects, aiming to provide an early pest forecasting tool and optimize the use of plant protection chemicals. The dataset consists of 334 field images with densities ranging from 10 to 100 individuals per image, annotated in YOLO format. Experimental comparisons of performance among YOLOv8, YOLOv9, YOLOv10, and YOLOv11 were conducted under the same data conditions and training configurations. The results indicate that YOLOv11n achieves the best performance, with a Precision of 99.12%, Recall of 97.78%, mAP@0.5 of 98.77%, and mAP@[0.5:0.95] of 77.15%. With an optimized architecture comprising 2.65 million parameters and 6.6 GFLOPs, this model ensures compactness for deployment. A software system was developed to integrate the model, visualize the results, and validate its practical applicability in agricultural pest monitoring.

Keywords: Sweet potato weevil; Deep learning; Object detection; Image processing; YOLO



1. ĐẶT VẤN ĐỀ

Bọ hà khoai lang còn khó kiểm soát do vòng đời phát triển chủ yếu diễn ra trong thân và củ, khiến việc phát hiện sớm và can thiệp kịp thời gặp nhiều hạn chế cũ [1]. Do đó, việc ước lượng sớm mật độ dịch hại có ý nghĩa quan trọng trong quản lý dịch hại tổng hợp, góp phần giảm thiểu tổn thất và nâng cao hiệu quả sản xuất nông nghiệp. Việc lựa chọn mô hình được thực hiện thông qua so sánh các phiên bản YOLO khác nhau trong cùng điều kiện dữ liệu và huấn luyện [2].

Trên thực tế, việc theo dõi mật độ bọ hà khoai lang hiện nay chủ yếu dựa vào các phương pháp bẫy thủ công và kiểm tra trực quan, dẫn đến nhiều hạn chế trong khả năng phản hồi sớm và quy mô giám sát [3]. Phương pháp thủ công này phụ thuộc vào kinh nghiệm của người nông dân và chuyên gia, tiêu tốn thời gian và tài nguyên, đồng thời khó áp dụng hiệu quả trong quản lý dịch hại trên diện tích lớn. Các nghiên cứu tổng quan về công nghệ nhận dạng đối tượng trong nông nghiệp chỉ ra rằng việc sử dụng các giải pháp tự động dựa trên thị giác máy tính và học sâu là xu hướng tất yếu để giảm chi phí và tăng độ chính xác trong giám sát dịch hại so với phương pháp thủ công [4]. Trong đó, các mô hình một giai đoạn như YOLO (You Only Look Once) đã được chứng minh là phù hợp cho các nhiệm vụ phát hiện đối tượng trong môi trường thực tế nhờ khả năng xử lý nhanh và hiệu quả với đối tượng nhỏ và nền ảnh phức tạp [5]. Các nghiên cứu gần đây trong lĩnh vực phát hiện sâu bệnh và côn trùng gây hại sử dụng YOLO hay biến thể của nó đã mô tả rõ hiệu quả của kiến trúc này trong môi trường nông nghiệp, nơi đối tượng bị che khuất, kích thước nhỏ và nền ảnh đa dạng là thách thức thường trực. Ví dụ, các mô hình YOLO đã được áp dụng thành công để phát hiện và đếm sâu bệnh trên nhiều loại cây trồng, với độ chính xác cao và chi phí tính toán hiệu quả, đồng thời hỗ trợ triển khai trên thiết bị có tài nguyên tính toán hạn chế [6]. Phát hiện côn trùng là bài toán đối tượng nhỏ đầy thách thức do đặc trưng hình thái dễ bị triệt tiêu khi đi qua các tầng trích xuất sâu. Vì vậy, các biến thể YOLO hiện đại đã tích hợp cơ chế học đặc trưng đa quy mô (multi-scale feature learning) nhằm bảo toàn và khai thác hiệu quả thông tin của đối tượng ở kích thước tối thiểu [7].

Vì vậy, nghiên cứu này hướng tới: (1) Xây dựng tập dữ liệu ảnh bọ hà khoai lang được gán nhãn thống nhất theo định dạng YOLO, hỗ trợ trực tiếp cho việc huấn luyện mô hình phát hiện đối tượng; (2) Huấn luyện và đánh giá các phiên bản YOLO khác nhau trên cùng tập dữ liệu ảnh bọ hà. So sánh này nhằm đánh giá ảnh hưởng của số lượng tham số (params), số phép toán dấu phẩy động (FLOPs) và các chỉ số về độ chính xác của mô hình như độ chính xác (Precision), độ hồi tưởng (Recall), mAP@0.5 (Mean Average Precision tại ngưỡng Intersection



over Union – IoU = 0.5, trong đó IoU đo mức độ chồng lấp giữa hộp dự đoán và hộp nhãn thực) và mAP@0.5:0.95 (giá trị mAP trung bình trên dải ngưỡng IoU từ 0.5 đến 0.95 với bước 0.05), nhằm để xác định cấu hình phù hợp nhất cho bài toán phát hiện và đếm côn trùng nhỏ trong thực địa; (3) Phát triển một hệ thống phần mềm để triển khai mô hình YOLO đã huấn luyện, phục vụ cho việc thử nghiệm và đánh giá khả năng triển khai mô hình trong giám sát dịch hại nông nghiệp. Ứng dụng này có mục tiêu tích hợp mô hình học sâu vào quy trình xử lý ảnh và trực quan hóa kết quả phát hiện, từ đó minh họa khả năng áp dụng thực tế của mô hình cho bài toán nhận diện bọ hà khoai lang.

2. VẬT LIỆU VÀ PHƯƠNG PHÁP

2.1 Xây dựng bộ dữ liệu bọ hà khoai lang

Nhằm giải quyết bài toán tự động hóa việc phát hiện và đếm bọ hà khoai lang ngoài đồng ruộng, một bộ dữ liệu hình ảnh đã được xây dựng theo quy trình kiểm soát chặt chẽ [3]. Đầu tiên, nhóm nghiên cứu thực hiện việc sử dụng bẫy mồi pheromone và bẫy chuyên dụng nhằm thu hút các mẫu bọ hà trên cánh đồng khoai lang tại tỉnh Bắc Ninh (xem Hình 1). Sau đó, các mẫu bọ hà sẽ được chụp bằng điện thoại iPhone XS Max với camera thường, không chỉnh sửa thông số. Các ảnh được chụp trong môi trường có độ sáng đồng đều, giúp loại bỏ nhiễu từ ánh sáng thay đổi trong môi trường thực tế. Ảnh thu được có độ phân giải 1920x2560 pixels, đảm bảo độ chi tiết cần thiết cho quá trình huấn luyện và suy luận của mô hình huấn luyện.



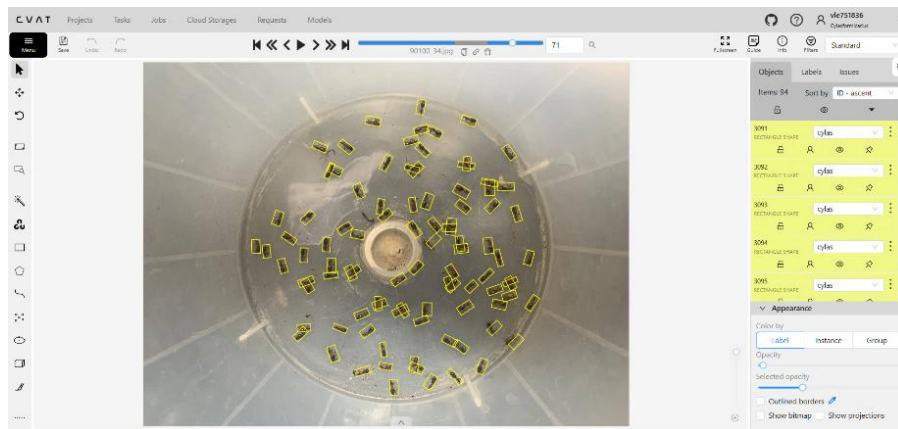
Hình 1. Thu thập bọ hà ngoài thực tế

Do dữ liệu ảnh ban đầu được thu thập với kích thước 1920×2560 pixels, trước khi thực hiện các bước tiền xử lý tiếp theo, ảnh được cắt về kích thước 1920×1920 pixels nhằm chuẩn hóa tỷ lệ khung hình. Bước xử lý này được áp dụng để loại bỏ sự không đồng nhất về hình dạng ảnh đầu vào, đồng thời giảm thiểu sai lệch hình học của đối tượng có thể phát sinh trong quá trình điều chỉnh kích thước ảnh về kích thước đầu vào cố định của mô hình. Việc cắt ảnh về dạng vuông trước khi điều chỉnh kích thước theo cơ chế bảo toàn tỉ lệ khung hình giúp giữ lại



đặc trưng hình thái của đối tượng, đồng thời đảm bảo tính nhất quán của dữ liệu trong toàn bộ quá trình huấn luyện và đánh giá mô hình.

Quá trình gán nhãn dữ liệu được thực hiện thủ công bằng nền tảng CVAT (Computer Vision Annotation Tool) trong Hình 2 [8]. Mỗi cá thể bọ hà trong ảnh được gán nhãn thủ công bằng hộp bao quanh và các nhãn này được tổ chức theo định dạng YOLO phục vụ cho quá trình huấn luyện mô hình [9]. Bài toán trong nghiên cứu được thiết lập dưới dạng phát hiện đối tượng một lớp, trong đó toàn bộ các cá thể bọ hà khoai lang được gán chung một nhãn. Trong quá trình gán nhãn, số lượng cá thể bọ hà trong mỗi ảnh có xu hướng dao động trong khoảng từ 10 đến 100, cho thấy đặc trưng phân bố mật độ cao và mức độ chồng lấn đáng kể giữa các đối tượng trong dữ liệu thực địa. Nhằm đảm bảo chất lượng tập dữ liệu, quá trình gán nhãn được kiểm soát chặt chẽ để các hộp bao quanh đại diện chính xác từng cá thể bọ hà. Các bước này nhằm giảm thiểu sai lệch và tăng độ chính xác trong quá trình huấn luyện mô hình [10, 11].



Hình 2. Gán nhãn dữ liệu trên nền tảng CVAT

2.2 Ứng dụng mô hình học sâu

Nghiên cứu này tập trung so sánh thực nghiệm hiệu năng giữa các phiên bản khác nhau của các biến thể mô hình YOLO trong cùng một bài toán phát hiện bọ hà khoai lang. Cách tiếp cận này nhằm đánh giá mức độ phù hợp của từng phiên bản YOLO đối với bài toán phát hiện và đếm côn trùng kích thước nhỏ, xuất hiện dày đặc trong điều kiện ảnh thực địa. YOLO là dòng mô hình phát hiện đối tượng một giai đoạn nổi bật với khả năng suy luận thời gian thực và kiến trúc gọn nhẹ [2]. Các phiên bản YOLO liên tiếp được giới thiệu với những thay đổi về cấu trúc mạng, cơ chế huấn luyện và chiến lược trích xuất đặc trưng, dẫn đến sự khác biệt đáng kể về độ chính xác, tốc độ suy luận và chi phí tính toán [11].

Để đảm bảo tính công bằng và khách quan trong việc đánh giá hiệu năng, tất cả các mô hình (YOLOv8n, v8s, v9n, v9s, v10n và v11n) đều được huấn luyện và kiểm thử trên cùng một



tập dữ liệu chuẩn duy nhất. Quá trình thực nghiệm sử dụng phương pháp Transfer Learning với các trọng số tiền huấn luyện (Pre-trained weights) từ tập dữ liệu COCO, đồng thời áp dụng cấu hình siêu tham số thống nhất cho mọi phiên bản trong suốt quá trình huấn luyện [11]. Cụ thể, quá trình huấn luyện được thiết lập với 80 epochs, ảnh đầu vào được điều chỉnh kích thước về 1024×1024 pixels theo cơ chế bảo toàn tỷ lệ khung hình (letterbox), kích thước lô huấn luyện (batch size) được cố định ở mức 8 nhằm đảm bảo tính ổn định của quá trình tối ưu và sự nhất quán giữa các cấu hình mô hình. Việc cố định các thông số này giúp hạn chế tối đa ảnh hưởng của các yếu tố ngoại lai, đảm bảo sự khác biệt về kết quả đầu ra chỉ phản ánh đặc trưng kiến trúc của từng phiên bản mô hình.

Quá trình thực nghiệm được triển khai trên môi trường Kaggle Notebook với cấu hình phần cứng sử dụng 02 GPU NVIDIA Tesla T4 (16GB GDDR6 mỗi card), hỗ trợ tăng tốc tính toán song song. Các siêu tham số huấn luyện được thiết lập dựa trên cấu hình mặc định của YOLOv11 để đảm bảo tính khách quan khi so sánh giữa các phiên bản, cụ thể như sau: (1) Bộ tối ưu hoá (Optimizer) Sử dụng thuật toán tối ưu hóa tự động (auto), mặc định là SGD với xung lượng (momentum) đạt 0.937 và hệ số suy giảm trọng số (weight decay) là 0.0005. Quá trình khởi động được duy trì trong 3.0 epoch đầu tiên để ổn định gradient; (2) Tốc độ học (Learning rate) được khởi tạo ở mức 0.01. Hệ thống áp dụng cơ chế giảm cấp tốc độ học với hệ số kết thúc bằng 0.01, giúp mô hình hội tụ sâu hơn ở các giai đoạn cuối của quá trình huấn luyện; (3) Tăng cường dữ liệu (Augmentation) biến đổi không gian màu HSV (H: 0.015, S: 0.7, V: 0.4), dịch chuyển ảnh (0.1), thay đổi tỉ lệ (0.5) và lật ngang ảnh (0.5). Đặc biệt, kỹ thuật Mosaic augmentation được thiết lập ở mức tối đa (1.0).

Tập dữ liệu được chia thành ba tập con bao gồm cả ảnh côn trùng riêng lẻ và chồng lán: tập huấn luyện/ đánh giá/ kiểm thử tương ứng với 8:1:1. Các mô hình sau khi huấn luyện sẽ được đánh giá trên tập kiểm thử để đo lường khả năng tổng quát hóa trên dữ liệu chưa từng xuất hiện trong quá trình huấn luyện trước đó [12]. Hiệu năng phát hiện của mô hình được đánh giá thông qua hai chỉ số cơ bản là Precision và Recall. Trong đó, Precision phản ánh tỷ lệ các dự đoán dương tính đúng (true positives, TP) trên tổng số dự đoán dương, bao gồm cả các dự đoán sai (false positives, FP), còn Recall thể hiện tỷ lệ các đối tượng thực tế được mô hình phát hiện đúng trên tổng số đối tượng cần phát hiện, bao gồm cả các trường hợp bị bỏ sót (false negatives, FN). Hai chỉ số này được xác định theo các công thức chuẩn trong bài toán phát hiện đối tượng, lần lượt được trình bày trong công thức (1) và công thức (2). Bên cạnh Precision và Recall, các chỉ số $mAP@0.5$ và $mAP@[0.5:0.95]$ được sử dụng để đánh giá toàn diện năng lực phát hiện và chất lượng định vị đối tượng. Công thức (3) xác định $mAP@0.5$, tính toán tại ngưỡng chồng lán $IoU = 0,5$, nhằm đánh giá khả năng nhận diện đối tượng ở



mức cơ bản và phân biệt mục tiêu với nền môi trường. Trong khi đó, công thức (4) mô tả $mAP@[0.5:0.95]$, giá trị trung bình của mAP trên dải ngưỡng IoU từ 0,5 đến 0,95. Đây là thước đo khắt khe hơn, phản ánh độ chính xác tinh vi trong việc thiết lập khung bao sát hình dáng cá thể, yếu tố then chốt đối với bài toán đếm đối tượng kích thước nhỏ và mật độ dày đặc.

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$mAP@0.5 = \frac{1}{N} \sum_{i=1}^N AP_i \quad (3)$$

$$mAP@[0.5:0.95] = \frac{1}{10} \sum_{i=1}^{10} mAP_{IoU} \quad (4)$$

Bên cạnh hai chỉ số trên, nghiên cứu sử dụng chỉ số mAP để đánh giá tổng hợp đồng thời khả năng phát hiện và độ chính xác định vị hộp bao quanh của mô hình. Cụ thể, $mAP@0.5$ được tính tại ngưỡng chồng lấn IoU cố định bằng 0,5, phản ánh hiệu năng phát hiện trong điều kiện yêu cầu định vị ở mức cơ bản, trong khi $mAP@[0.5:0.95]$ được xác định bằng cách lấy trung bình giá trị AP trên dải ngưỡng IoU từ 0,5 đến 0,95 với bước nhảy 0,05, cho phép đánh giá khắt khe hơn về chất lượng định vị trong các kịch bản phát hiện thực tế. Việc kết hợp Precision, Recall và các chỉ số mAP giúp phản ánh toàn diện hiệu năng của mô hình trong bài toán phát hiện và đếm bọ hà khoai lang.

Nghiên cứu đồng thời xem xét chi phí tính toán của từng phiên bản mô hình thông qua hai đại lượng: số lượng tham số và số phép toán dấu phẩy động. Hai chỉ số này được sử dụng như thước đo hỗ trợ để đánh giá mức độ “gọn nhẹ” và yêu cầu tính toán của mô hình. Việc kết hợp các chỉ số độ chính xác và chi phí tính toán cho phép so sánh hiệu năng mô hình theo hướng toàn diện, phục vụ mục tiêu lựa chọn cấu hình phù hợp cho bài toán phát hiện và đếm bọ hà [13].

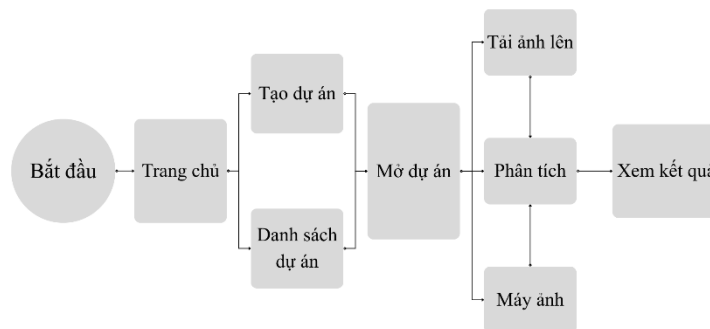
2.3 Hệ thống phần mềm phát hiện và đếm bọ hà

Trong nghiên cứu này, một hệ thống phần mềm hỗ trợ phát hiện và đếm bọ hà khoai lang được xây dựng nhằm phục vụ quá trình thử nghiệm và minh họa khả năng suy luận của các mô hình YOLO đã huấn luyện. Hệ thống đóng vai trò trung gian giữa mô hình học sâu và dữ liệu ảnh đầu vào, cho phép thực hiện toàn bộ quy trình từ thu ảnh, tiền xử lý, suy luận đến hiển thị kết quả phát hiện. Về mặt kiến trúc chức năng, hệ thống được thiết kế theo hướng mô-đun, bao gồm các thành phần chính: quản lý dự án, thu ảnh đầu vào, xử lý cùng suy luận bằng mô hình tiền huấn luyện và hiển thị kết quả phát hiện. Ảnh đầu vào được người dùng cung cấp thông qua chức năng tải ảnh từ thiết bị lưu trữ hoặc thu nhận trực tiếp từ máy ảnh. Mỗi ảnh được gắn



với một dự án cụ thể nhằm phục vụ việc tổ chức và theo dõi quá trình phân tích. Sau khi tiếp nhận, ảnh được đưa qua bước tiền xử lý, bao gồm chuẩn hóa kích thước và định dạng theo yêu cầu của mô hình YOLO đã huấn luyện. Dữ liệu sau tiền xử lý được chuyển tới mô-đun suy luận, tại đó mô hình YOLO thực hiện phát hiện các cá thể bộ hạ và sinh ra các khung bao tương ứng. Kết quả suy luận bao gồm vị trí các khung bao và số lượng đối tượng được phát hiện trong mỗi ảnh [2, 11].

Kết quả đầu ra được trực quan hóa trên giao diện người dùng, trong đó các khung bao được hiển thị chồng lên ảnh gốc nhằm biểu diễn vị trí của từng cá thể bộ hạ. Luồng xử lý của hệ thống được xây dựng thống nhất với quy trình huấn luyện và đánh giá mô hình, đảm bảo dữ liệu đầu vào và đầu ra của mô-đun suy luận tuân theo cùng một chuẩn định dạng. Quy trình hoạt động tổng thể của hệ thống được minh họa trong Hình 3, bao gồm các bước: khởi tạo hoặc mở dự án, tiếp nhận ảnh đầu vào, thực hiện suy luận bằng mô hình YOLO và hiển thị kết quả phát hiện. Thiết kế này cho phép hệ thống hoạt động độc lập với giai đoạn huấn luyện, tập trung vào việc triển khai và kiểm thử mô hình trong các kịch bản sử dụng ảnh khác nhau. Thông qua giao diện này, người dùng có thể nhanh chóng kiểm tra khả năng phát hiện của mô hình trong các kịch bản ảnh khác nhau, bao gồm ảnh có mật độ đối tượng cao và nền ảnh phức tạp.



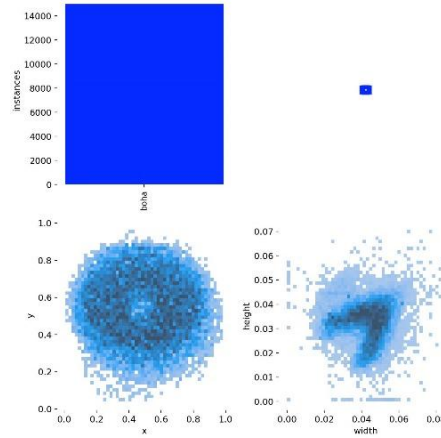
Hình 3. Quy trình hoạt động của ứng dụng

3. KẾT QUẢ VÀ BÀN LUẬN

3.1 Tập dữ liệu bộ hạ khoai lang

Tập dữ liệu bộ hạ khoai lang được sử dụng trong nghiên cứu được chia thành ba tập con bao gồm 267 ảnh huấn luyện, 33 ảnh đánh giá và 34 ảnh kiểm thử, với tổng cộng 334 ảnh. Toàn bộ các ảnh trong ba tập đều có tệp nhãn tương ứng, không tồn tại trường hợp ảnh thiếu nhãn hoặc tệp nhãn rỗng, cho thấy quá trình chuẩn bị và kiểm tra dữ liệu được thực hiện đầy đủ và nhất quán. Tập dữ liệu bao gồm tổng cộng 18.509 hộp bao quanh, trong đó tập huấn luyện chiếm phần lớn với 14.999 nhãn. Phân tích thống kê phân bố nhãn và kích thước hộp bao quanh

trong tập huấn luyện, được minh họa trong Hình 4, cho thấy các hộp bao quanh có kích thước tương đối nhỏ so với kích thước ảnh và phân bố không đồng đều trong không gian ảnh. Đặc điểm này phản ánh rõ bản chất của đối tượng cần phát hiện là các cá thể bọ hà kích thước nhỏ, xuất hiện với mật độ cao và thường chồng lấn trong điều kiện ảnh thực địa.



Hình 4. Phân bố nhãn và kích thước hộp bao quanh trong tập huấn luyện

3.2 Huấn luyện và so sánh mô hình

Kết quả so sánh hiệu năng giữa các phiên bản YOLO được trình bày trong Bảng 1, có thể nhận thấy các phiên bản YOLO được khảo sát đều đạt hiệu năng phát hiện cao trên tập dữ liệu bọ hà khoai lang, cho thấy tính phù hợp của dòng mô hình phát hiện một giai đoạn đối với bài toán phát hiện và đếm côn trùng mật độ cao. Tuy nhiên, giữa các phiên bản vẫn tồn tại sự khác biệt rõ rệt về mức độ cân bằng giữa độ chính xác phát hiện và chi phí tính toán.

Bảng 1. So sánh các phiên bản YOLO

Model	Precision (%)	Recall (%)	mAP@0.5 (%)	mAP@[0.5:0.95] (%)	Params (M)	FLOPs (G)
YOLOv8n	97,85	96,36	97,81	71,34	3,01	8,1
YOLOv8s	98,49	97,56	98,17	74,18	11,13	28,4
YOLOv9n	97,38	96,93	98,01	71,82	1,97	7,6
YOLOv9s	98,66	97,45	98,31	74,4	7,17	26,7
YOLOv10n	94,93	93,27	97,34	70,68	2,69	8,2
YOLOv11n	99,12	97,78	98,77	77,15	2,65	6,6

Xét tổng thể các kết quả trình bày trong Bảng 1, YOLOv11n cho thấy hiệu năng toàn diện tốt nhất trong số các mô hình được đánh giá, khi đạt được sự cân bằng hợp lý giữa độ chính xác

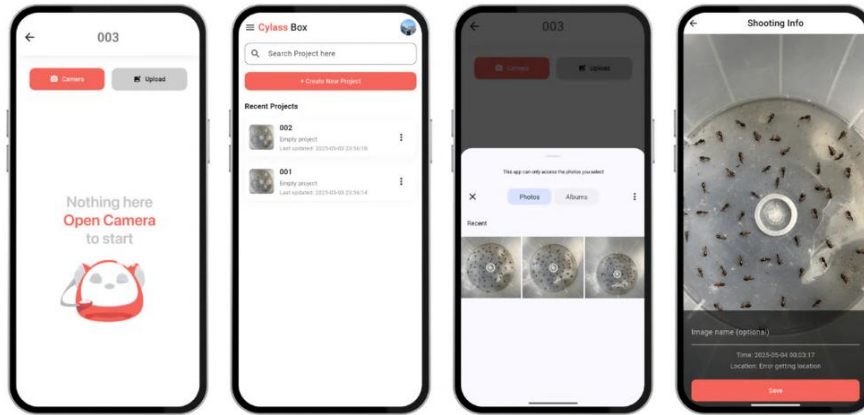


phát hiện và chi phí tính toán. Mô hình này nổi bật so với các phiên bản YOLO còn lại, đặc biệt trong bối cảnh bài toán phát hiện và đếm bọ hà khoai lang với mật độ đối tượng cao và kích thước nhỏ. Về hiệu năng phát hiện, YOLOv11n đạt Precision 99,12% và Recall 97,78%, là các giá trị cao nhất trong nhóm các mô hình so sánh. Đồng thời, mô hình đạt $mAP@0.5 = 98,77\%$, vượt qua các phiên bản YOLOv8n (97,81%), YOLOv9n (98,01%) và YOLOv10n (97,34%). Đáng chú ý, tại chỉ số $mAP@[0.5:0.95]$, YOLOv11n đạt 77,15%, cao hơn rõ rệt so với các mô hình còn lại (từ 70,68% đến 74,40%). Kết quả này cho thấy YOLOv11n có khả năng định vị chính xác hơn trong các điều kiện IoU cao, phù hợp với đặc thù của bài toán phát hiện đối tượng nhỏ và chồng lấn.

Về chi phí tính toán, các phiên bản YOLO cho thấy sự khác biệt rõ rệt giữa mức độ phức tạp mô hình và hiệu năng đạt được. YOLOv11n chỉ sử dụng 2,65 triệu tham số và 6,6 GFLOPs, là một trong những cấu hình nhẹ nhất trong nhóm mô hình được khảo sát, thấp hơn YOLOv8n (3,01 triệu tham số, 8,1 GFLOPs) và YOLOv10n (2,69 triệu tham số, 8,2 GFLOPs), đồng thời giảm đáng kể chi phí so với các phiên bản kích thước lớn như YOLOv8s (11,13 triệu tham số, 28,4 GFLOPs) và YOLOv9s (7,17 triệu tham số, 26,7 GFLOPs). Mặc dù YOLOv9n có số tham số thấp nhất (1,97 triệu), FLOPs của mô hình này vẫn ở mức 7,6G và không mang lại lợi thế rõ rệt về hiệu năng so với YOLOv11n. Như vậy, YOLOv11n đạt được mức tiết kiệm tài nguyên tính toán đáng kể cả về bộ nhớ mô hình lẫn số phép toán suy luận, trong khi vẫn duy trì hiệu năng phát hiện vượt trội, cho thấy cấu trúc mô hình được tối ưu tốt hơn cho bài toán phát hiện và đếm đối tượng mật độ cao trong điều kiện tài nguyên hạn chế tuy nhiên vẫn tồn tại các sai số đặc trưng: (1) Âm tính giả (False Negative) chỉ số Recall đạt 97,78%, tương ứng tỉ lệ bỏ sót khoảng 2,22%. Lỗi này phát sinh chủ yếu do bọ hà nằm chồng lấn lên nhau hoặc bị khuất ở mép ảnh, khiến mô hình khó trích xuất đủ đặc trưng hình dạng; (2) Dương tính giả (False Positive) với Precision đạt 99,12%, tỉ lệ nhận diện nhầm là 0,88%. Nguyên nhân do các mảnh thực vật hoặc vết bẩn trong ảnh có màu sắc và kích thước tương đồng với đối tượng mục tiêu.

3.3 Hệ thống phần mềm phát hiện và đếm bọ hà

Bên cạnh kết quả đánh giá hiệu năng của các mô hình học máy, nghiên cứu đã triển khai một hệ thống phần mềm nhằm tích hợp mô hình YOLO đã huấn luyện vào môi trường ứng dụng thực tế cho bài toán phát hiện và đếm bọ hà khoai lang. Hệ thống đóng vai trò như một công cụ hỗ trợ thử nghiệm và trực quan hóa kết quả suy luận, qua đó kiểm chứng tính nhất quán giữa giai đoạn huấn luyện mô hình và giai đoạn ứng dụng.



Hình 5. Giao diện ứng dụng

Giao diện của hệ thống phần mềm, như minh họa trong Hình 5 cho phép quản lý dữ liệu ảnh theo từng dự án một cách độc lập. Các chức năng chính của hệ thống bao gồm tạo và truy xuất dự án, nhập ảnh đầu vào từ nguồn lưu trữ hoặc thiết bị thu nhận ảnh, cũng như thực hiện suy luận tự động bằng mô hình YOLO đã được huấn luyện. Đối với mỗi ảnh được xử lý, kết quả phát hiện được trực quan hóa thông qua các hộp bao tương ứng với từng cá thể bọ hà, đồng thời cung cấp thông tin về số lượng đối tượng được nhận diện trong ảnh. Việc triển khai hệ thống phần mềm cho phép quan sát trực quan hành vi của mô hình trong các kịch bản ảnh khác nhau, đặc biệt trong điều kiện mật độ đối tượng cao và hiện tượng chồng lấn thường gặp trong dữ liệu thực địa. Thông qua hệ thống này, khả năng chuyển giao mô hình từ môi trường nghiên cứu sang bối cảnh ứng dụng thực tế bước đầu được kiểm chứng ở mức chức năng, tạo tiền đề cho các nghiên cứu tiếp theo hướng tới hoàn thiện hệ thống giám sát dịch hại nông nghiệp dựa trên thị giác máy tính.

4. KẾT LUẬN

Nghiên cứu đã thực hiện so sánh thực nghiệm các phiên bản khác nhau của dòng mô hình YOLO trong bài toán phát hiện và đếm bọ hà khoai lang từ ảnh thực địa, trên cùng một tập dữ liệu và quy trình huấn luyện thống nhất. Kết quả cho thấy các mô hình có sự khác biệt rõ rệt về độ chính xác phát hiện và chi phí tính toán. Trong số các phiên bản được đánh giá, YOLOv11n thể hiện sự cân bằng tốt nhất giữa hiệu năng phát hiện, độ ổn định định vị và mức chi phí tính toán thấp, qua đó được lựa chọn là cấu hình phù hợp cho bài toán nghiên cứu. Bên cạnh việc xác định mô hình phù hợp, nghiên cứu làm rõ ý nghĩa của việc lựa chọn kiến trúc phát hiện đối tượng trong bối cảnh dữ liệu có mật độ đối tượng cao và điều kiện ảnh thực địa phức tạp. Quy trình so sánh được xây dựng theo hướng thống nhất và có khả năng tái lập, cho phép đánh giá khách quan sự đánh đổi giữa độ chính xác và chi phí tính toán của các phiên bản YOLO khác



nhau. Ngoài ra, việc tích hợp mô hình đã huấn luyện vào hệ thống phần mềm phát hiện và đếm bọ hà cho thấy khả năng chuyển giao từ giai đoạn nghiên cứu sang triển khai thử nghiệm trong thực tế.

Tuy nhiên, nghiên cứu vẫn tồn tại một số giới hạn nhất định. Trước hết, tập dữ liệu được xây dựng tập trung vào một loài dịch hại duy nhất và được thu nhận trong điều kiện ghi hình tương đối đồng nhất, do đó chưa phản ánh đầy đủ sự biến thiên của môi trường canh tác ngoài đồng ruộng. Bên cạnh đó, việc đánh giá mô hình trong nghiên cứu chủ yếu dựa trên các chỉ số hiệu năng phát hiện và chi phí tính toán, trong khi các yếu tố liên quan đến tốc độ suy luận thực tế như độ trễ hay số khung hình trên giây trên phần cứng triển khai cụ thể chưa được xem xét. Ngoài ra, bài toán được thiết lập dưới dạng phát hiện một lớp đối tượng nhằm tập trung đánh giá khả năng định vị và đếm đối tượng trong điều kiện mật độ cao, nên chưa mở rộng sang các kịch bản đa loài hoặc phân loại theo mức độ gây hại. Trong các nghiên cứu tiếp theo, hướng phát triển có thể tập trung vào mở rộng tập dữ liệu với nhiều điều kiện môi trường và đối tượng dịch hại khác nhau, đồng thời đánh giá khả năng triển khai mô hình trên các nền tảng phần cứng hạn chế tài nguyên và trong các kịch bản giám sát gần thời gian thực.

TÀI LIỆU THAM KHẢO

- [1] S. M. Hue and M. Y. Low, “An Insight into Sweet Potato Weevils Management: A Review,” 2015, *Hindawi Limited*. doi: 10.1155/2015/849560.
- [2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” May 2016, [Online]. Available: <http://arxiv.org/abs/1506.02640>
- [3] C. M. Schloemer, S. H. Graham, and K. S. Lawrence, “Sweetpotato pest challenges and management options,” *J. Integr. Pest Manag.*, vol. 16, no. 1, 2025, doi: 10.1093/jipm/pmaf003.
- [4] Z. Khan, Y. Shen, and H. Liu, “ObjectDetection in Agriculture: A Comprehensive Review of Methods, Applications, Challenges, and Future Directions,” Jul. 01, 2025, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/agriculture15131351.
- [5] M. L. Ali and Z. Zhang, “The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection,” Dec. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/computers13120336.
- [6] S. Khalid, H. M. Oqaibi, M. Aqib, and Y. Hafeez, “Small Pests Detection in Field Crops Using Deep Learning Object Detection,” *Sustainability (Switzerland)*, vol. 15, no. 8, Apr. 2023, doi: 10.3390/su15086815.
- [7] M. Nikouei *et al.*, “Small object detection: A comprehensive survey on challenges, techniques and real-world applications,” Sep. 01, 2025, *Elsevier B.V.* doi: 10.1016/j.iswa.2025.200561.



- [8] B. Sekachev *et al.*, “opencv/cvat: v1.1.0,” Aug. 2020, *Zenodo*. doi: 10.5281/zenodo.4009388.
- [9] M. Aljabri, M. AlAmir, M. AlGhamdi, M. Abdel-Mottaleb, and F. Collado-Mesa, “Towards a better understanding of annotation tools for medical imaging: a survey,” *Multimed. Tools Appl.*, vol. 81, no. 18, pp. 25877–25911, Jul. 2022, doi: 10.1007/s11042-022-12100-1.
- [10] S. A. Magalhães *et al.*, “Evaluating the single-shot multibox detector and yolo deep learning models for the detection of tomatoes in a greenhouse,” *Sensors*, vol. 21, no. 10, May 2021, doi: 10.3390/s21103569.
- [11] P. Hidayatullah, N. Syakrani, M. R. Sholahuddin, T. Gelar, and R. Tubagus, “YOLOv8 to YOLO11: A Comprehensive Architecture In-depth Comparative Review.”
- [12] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” Apr. 2020, [Online]. Available: <http://arxiv.org/abs/2004.10934>
- [13] Z.-Q. Zhao, P. Zheng, S. Xu, and X. Wu, “Object Detection with Deep Learning: A Review,” Apr. 2019, [Online]. Available: <http://arxiv.org/abs/1807.05511>