

Cải tiến thuật toán YOLO v4 ứng dụng cho hệ thống giám sát bệnh phấn trắng và bệnh sương mai trên cây dưa chuột

Improving the YOLO v4 algorithm applied to the powdery mildew and downy mildew monitor system on cucumber plants

Ngô Quang Ước^{1,2,*}, Ngô Trí Dương², Bùi Đăng Thành^{1*}

¹Viện kỹ thuật Điều khiển và Tự động hóa, Viện Điện - Đại học Bách khoa Hà Nội

²Khoa Cơ Điện - Học viện Nông nghiệp Việt Nam

*Corresponding author E-mail: nquoc@vnua.edu.vn, and thanh.buidang@hust.edu.vn.

Abstract

This paper presents the research and design of the powdery mildew and downy mildew monitor system on cucumber plants. The system is designed based on Raspberry PI embedded computer and 5MB PI Camera. We have improved the YOLO v4 algorithm, then integrated it into this system to identify powdery mildew and downy mildew on cucumber plants. The test results of the improved YOLOv4 algorithm with the sample data set gave an accuracy of 83.43%, which is 0.37% higher than the original YOLO v4-tiny algorithm. The monitoring system initially operates stably with Raspberry PI hardware in the greenhouse at Vietnam National University of Agriculture.

Keywords: YOLOv4, Cucumber, Downy mildew, Powdery mildew, Greenhouse

Tóm tắt

Bài báo trình bày về nghiên cứu và thiết kế một hệ thống giám sát bệnh phấn trắng và bệnh sương mai trên cây dưa chuột. Hệ thống được thiết kế dựa trên máy tính nhúng Raspberry PI và Camera PI 5MB. Thuật toán YOLO v4 được chúng tôi cải tiến và được đưa vào trong hệ thống này nhằm nhận dạng bệnh sương mai và bệnh phấn trắng trên cây dưa chuột. Kết quả thử nghiệm thuật toán YOLO v4 cải tiến với bộ dữ liệu mẫu đã cho độ chính xác đạt tới 83.43%, kết quả này cao hơn so với thuật toán YOLO v4-tiny gốc là 0.37%. Hệ thống giám sát bước đầu đã hoạt động ổn định với phần cứng Raspberry PI trong nhà kính tại Học viện Nông nghiệp Việt Nam.

1. Giới thiệu

Bệnh hại cây trồng là một trong những nguyên nhân gây ra những thiệt hại lớn về kinh tế trong sản xuất nông nghiệp. Việc phát hiện và xác định kịp thời các bệnh hại cây trồng là điều cần thiết để chữa trị và kiểm soát chúng. Dưa chuột (*Cucumis sativus* L.) là một trong những loại cây rau phổ biến trên toàn thế giới có giá trị kinh tế cao [1][2][3]. Để sản xuất dưa chuột có năng suất cao thì yếu tố cần chú trọng là phải kiểm soát được sâu bệnh, trong đó bệnh phổ biến trên loại cây này là bệnh phấn trắng và bệnh sương mai [4][5][6][7][8][9]. Phát hiện và cảnh báo các loại bệnh nêu trên để có những giải pháp xử lý sớm nhằm tăng năng suất và giảm lượng thuốc bảo vệ thực vật là điều hết sức cần thiết. Việc ứng dụng công nghệ robot, thuật toán trí tuệ nhân tạo và Internet vạn vật (IoT) cho sản xuất trong nông nghiệp nói chung và cho hệ thống giám sát cây dưa chuột sẽ nâng cao hiệu quả chăm sóc cây trồng, làm giảm bớt sức lao động của người nông dân. Có nhiều công trình khoa học trên thế giới nghiên cứu sử dụng các công nghệ tự động hóa phục vụ sản xuất dưa chuột trong nhà lưới, chẳng hạn trong [10][11] đã phát triển xe để phun thuốc trừ sâu tự

động trong nhà kính trồng dưa chuột. Hệ thống này làm nhiệm vụ phun thuốc, nhưng chức năng nhận diện bệnh trên cây được thực hiện nhờ quan sát của con người. Công trình [12] đã thiết kế một robot để xác định sự thiếu hụt dinh dưỡng Nitơ của cây dưa chuột trong nhà kính từ đó điều khiển cung cấp phân bón cho cây. Một số công trình nghiên cứu khác lại tập trung vào các thuật toán để phân loại bệnh trên cây dưa chuột, Ke Lin [13] đã sử dụng mô hình Unet để phân đoạn bệnh phấn trắng trên lá bệnh dưa chuột, kết quả cho thấy độ chính xác trên 72,11%. Pan Zhang [14], đã sử dụng thuật toán Efficient-Net-B4-Ranger để phân loại các bệnh trên dưa chuột, kết quả cho độ chính xác là 97%, với phương pháp phân loại ảnh này, mỗi ảnh sẽ được phân loại cho một đối tượng, nên khi có nhiều đối tượng trong ảnh thì bài toán sẽ khó thực hiện và khi áp dụng vào thực tế sẽ rất khó. Trong các thuật toán phát hiện các đối tượng trong ảnh, YOLO v4 đang có sự nổi trội về độ chính xác và tốc độ xử lý [15]. Phiên bản YOLO v4-tiny [16] dựa trên YOLO v4 đã được lược bỏ bớt các cấu trúc mạng Neuron tích chập (CNN) để kích thước bé hơn và giảm được thời gian phát hiện đối tượng. YOLO v4-tiny có thể sử dụng được trên các phần cứng cơ bản như Raspberry PI, do đó chúng tôi sẽ cải tiến thuật toán này để ứng dụng cho hệ thống nhận dạng bệnh sương mai và bệnh phấn trắng trên cây dưa chuột, từ đó đưa ra các cảnh báo cho người nông dân.

2. Thiết kế hệ thống

2.1. Mô hình hệ thống

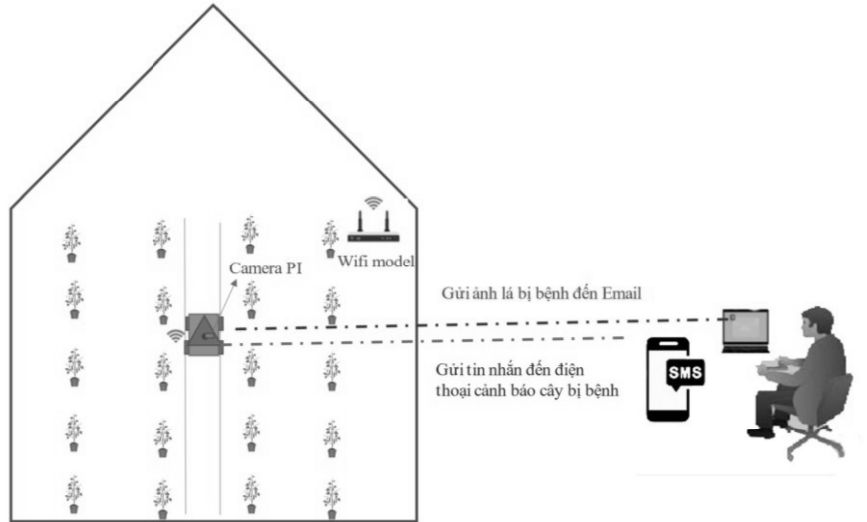
Hệ thống được thiết kế sẽ di chuyển trong nhà lưới trồng dưa chuột, chụp ảnh và xử lý hình ảnh bằng thuật toán trí tuệ nhân tạo (AI) để nhận ra bệnh phấn trắng và bệnh sương mai. Hình ảnh lá bị bệnh sẽ được gửi vào email và nhắn tin cảnh báo vào

điện thoại cho người quản lý. Mô hình cấu trúc hệ thống được thể hiện qua Hình 1.

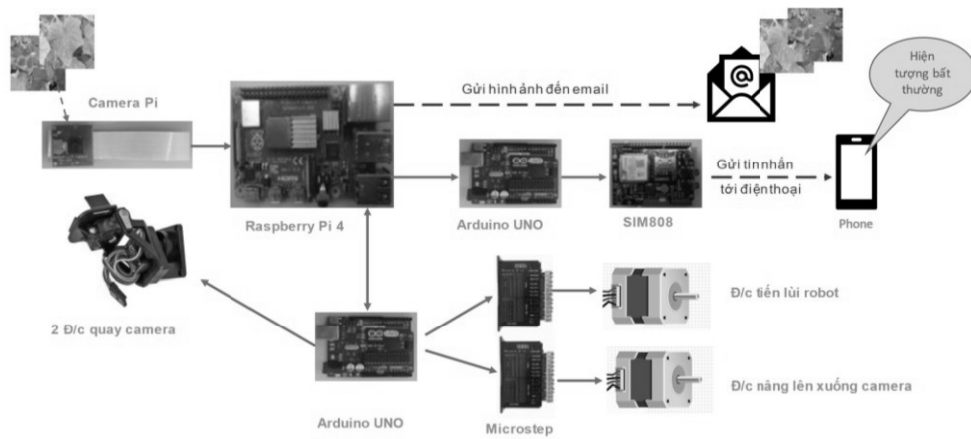
2.2. Cấu trúc phần cứng hệ thống

Trên Hình 2, biểu diễn sơ đồ cấu trúc các thiết bị phần cứng của hệ thống phát hiện và cảnh báo bệnh phần trắng và bệnh

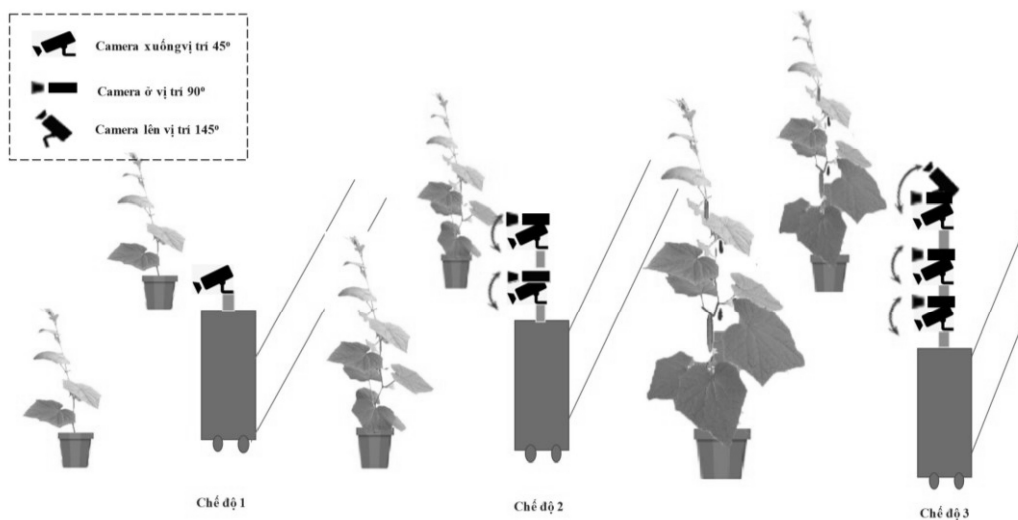
sương mai trên cây dưa chuột, hệ thống gồm có 1 Raspberry Pi 4, kết nối với các arduino để điều khiển động cơ tiến lùi hệ thống, nâng hạ camera; quay camera lên xuống; kết nối với SIM808 để truyền tin nhắn đến người quản lý nhà lưới biết tình trạng bất thường của cây dưa chuột. Đồng thời Raspberry sẽ gửi hình ảnh các lá dưa chuột bị bệnh vào email của người quản lý.



Hình 1: Mô hình tổng thể hệ thống làm việc trong nhà lưới



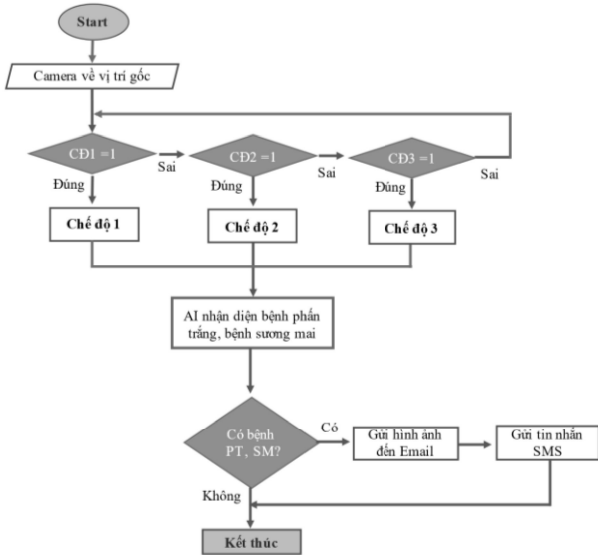
Hình 2: Sơ đồ cấu trúc các thiết bị phần cứng của hệ thống



Hình 3: Sơ đồ biểu diễn các chế độ làm việc của hệ thống

2.3. Thuật toán điều khiển hệ thống

Trên Hình 3, biểu diễn các chế độ làm việc của camera thử nghiệm với mô hình trồng dưa chuột thủy canh trong nhà lưới; vì cây dưa chuột khi phát triển hết có độ cao từ 2 m đến 3 m, do vậy mà cơ cấu của hệ thống được thiết kế để nâng camera lên chụp ảnh khi cây cao; khi cây bé dưới 1 m (từ 1 đến 5 lá thật) hệ thống làm việc ở chế độ 1, khi cây phát triển lên từ khoảng 1 m đến 1,6 m thì camera làm việc ở chế độ 2, và khi cây phát triển trên 1,6 m thì hệ thống làm việc ở chế độ 3. Hệ thống có thể nâng camera lên vị trí cao nhất là 170 cm so với mặt đất.



Hình 4: Lưu đồ thuật toán hệ thống giám sát và cảnh báo bệnh phần trắng và bệnh sương mai trên cây dưa chuột

Chế độ 1: Camera được đưa xuống vị trí thấp nhất là cách mặt đất 120 cm và xoay camera ở vị trí 45° chụp ảnh, để nguyên trạng thái này và di chuyển tiếp 50 cm đến cây tiếp theo chụp ảnh và cứ tiếp tục như vậy cho đến khi hết cây thực nghiệm.
Chế độ 2: Camera chụp ảnh 1 ở vị trí 45°, sau đó quay đến vị trí 90° chụp ảnh tiếp, tiếp đến hệ thống nâng camera lên 25 cm để chụp ảnh tiếp theo, sau đó lại xoay camera xuống 45°

chụp ảnh, tiếp theo hệ thống tiến lên 50 cm chụp ảnh các vị trí như lần đầu, cứ tiếp vậy cho đến khi hết số cây.

Chế độ 3: Cũng giống chế độ 2 nhưng nâng camera lên 25 cm nữa và chụp ở 3 vị trí là ở 45°, 90° và 145°.

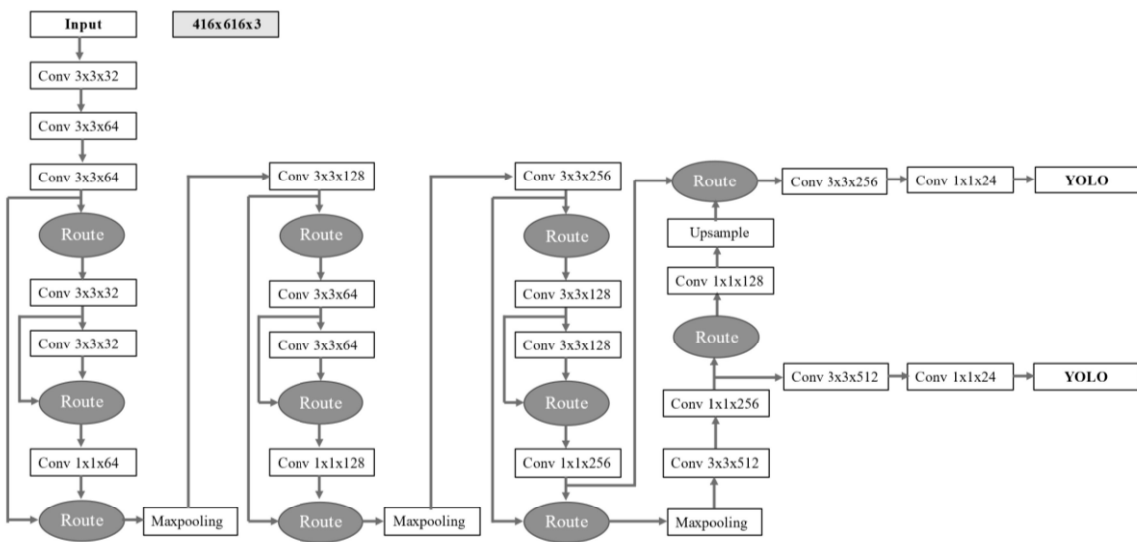
Hình 4, là lưu đồ thuật toán hệ thống giám sát cảnh báo cây dưa chuột bị bệnh phần trắng và bệnh sương mai hoạt động trong nhà lưới; trong đó CD1, CD2, CD3 là các chế độ làm việc của hệ thống camera phụ thuộc vào chiều cao cây trồng. Sau khi hệ thống chụp hết các vị trí của cây, các hình ảnh sẽ được qua thuật toán AI để phát hiện bệnh phần trắng, bệnh sương mai. Nếu ảnh nào bị bệnh sẽ được gửi vào email, tiếp đến là tin nhắn cảnh báo cây dưa chuột bị bệnh sẽ được gửi đến điện thoại của người quản lý và kết thúc. Ngoài ra nếu cây không bị bệnh hệ thống sẽ kết thúc quá trình làm việc.

3. Thuật toán phát hiện bệnh phần trắng và sương mai

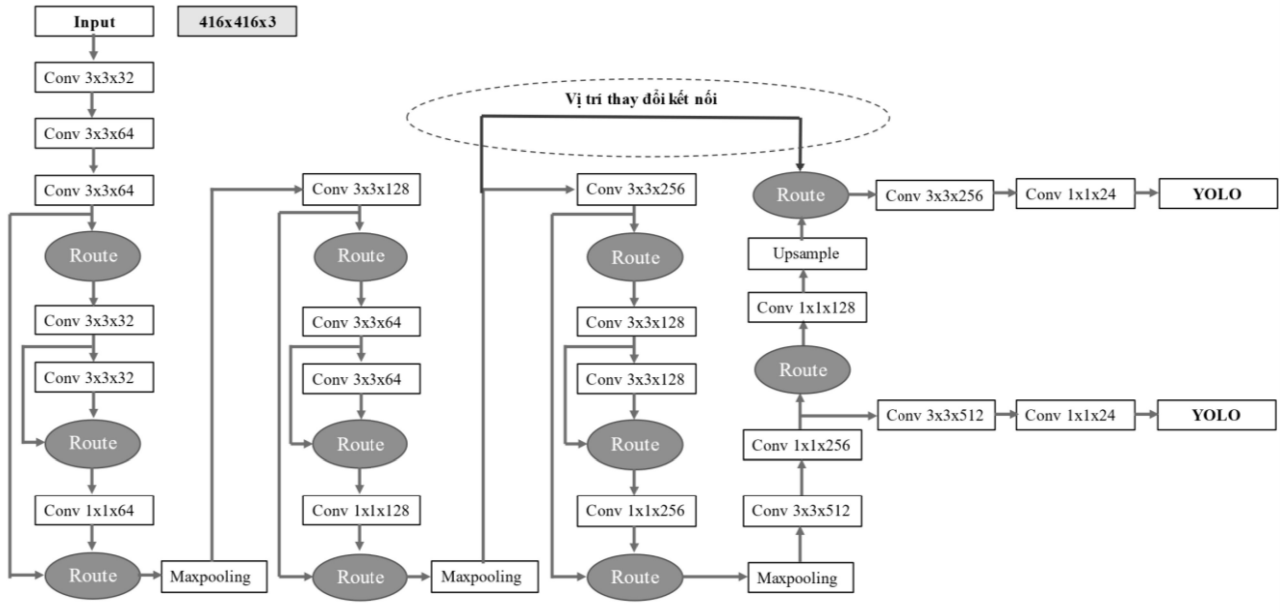
Thuật toán phát hiện bệnh phần trắng và bệnh sương mai được chúng tôi sử dụng là thuật toán YOLO v4-tiny gốc và thuật toán cải tiến của chúng tôi dựa trên YOLO v4-tiny gốc [16]. Trên Hình 5, thể hiện cấu trúc thuật toán YOLO v4-tiny gốc (gọi là YOLO v4-tiny) gồm có 38 lớp là 21 lớp Convolutional (Conv), 11 lớp Route, 3 lớp Maxpooling, 1 lớp Upsample, 2 lớp YOLO. Trên Hình 6, thể hiện cấu trúc thuật toán YOLO v4-tiny cải tiến của chúng tôi (gọi tắt là Improved YOLO v4-tiny), cấu trúc này cũng gồm các lớp giống với YOLO v4-tiny nhưng đã có sự thay đổi một số lớp kết nối với nhau: ở thuật toán YOLO v4-tiny thì lớp 34 là lớp kết nối 33 với lớp 23, trong khi thuật toán Improved YOLO v4-tiny lớp 34 là lớp kết nối 33 với lớp 17.

Độ chính xác trung bình (mAP) của mô hình có mối quan hệ với giá trị learning rate [17], giá trị của learning rate nằm trong khoảng 0.0 ÷ 1, trong model YOLOV4-tiny giá trị learning rate bằng 0.00261. Đối với mô hình Improved YOLO v4-tiny, giá trị learning rate sẽ được chúng tôi thay đổi quanh giá 0.00261 và giới hạn trong khoảng từ 0.002 đến 0.003 để tìm ra giá trị learning-rate phù hợp cho mô hình của chúng tôi.

Các tham số khác cho quá trình đào tạo và kiểm tra của hai mô hình được thể hiện trong Bảng 1.



Hình 5: Thuật toán YOLO v4-tiny gốc để phát hiện bệnh phần trắng và bệnh sương mai trên cây dưa chuột



Hình 6: Thuật toán YOLO v4-tiny cải tiến của chúng tôi để phát hiện bệnh phần trắng và bệnh sương mai trên cây dưa chuột

Bảng 1: Các tham số đào tạo và kiểm tra của 2 mô hình

Tham số	YOLO v4-tiny	Improved YOLO v4-tiny
Momentum	0.9	0.9
Decay	0.0005	0.0005
Batch size	64	64
Classes	3	3
Activation function	Leaky ReLU	Leaky ReLU
Iterations	10000	10000

Các giá trị để xác định trọng số tốt được thực hiện bằng các công thức từ (1) – (6). Trong đó: TP (true positive), FP (fault positive), TN (true negative), and FN (fault negative), R là vùng được phát hiện của hộp giới hạn đối tượng, R' là vùng thực của hộp giới hạn đối tượng, mAP là giá trị trung bình của độ chính xác trung bình (AP) khi một lá dưa chuột được phát hiện, S^2 là số ô lưới trong ảnh đầu vào, B là số bounding box giới hạn trong một lưới, W_{ij}^{obj} là một chức năng của đối tượng: nếu bounding box j của lưới thứ i phát hiện ra đối tượng thì $W_{ij}^{obj} = 1$ nếu không thì $W_{ij}^{obj} = 0$, C_i^j và \hat{C}_i^j lần lượt là điểm tin cậy của predicted box và điểm tin cậy của truth box, λ_{noobj} là một tham số trọng số, $P_i^j(c)$ và $\hat{P}_i^j(c)$ là predicted probability và truth probability mà đối tượng thuộc phân loại c trong bounding box thứ j của lưới thứ i, IOU là intersection over union giữa các hộp predicted bounding box và truth bounding box, w^{gt} và h^{gt} là chiều rộng và chiều cao truth bounding box, w và h là chiều rộng và chiều cao của predicted bounding box, $\rho^2(b, b^{gt})$ biểu thị khoảng cách Euclidean giữa các điểm chính giữa của predicted bounding box và truth bounding box, c là khoảng cách đường chéo nhỏ nhất của box có thể chứa predicted bounding box và truth bounding box.

$$\text{precision} = \frac{TP}{TP + FP} \times 100\% \quad (1)$$

$$\text{recall} = \frac{TP}{TP + FN} \times 100\% \quad (2)$$

$$\text{mAP} = \frac{\sum_{c=1}^C AP(c)}{C} \quad (3)$$

$$\text{IoU}(R, R') = \frac{|R \cap R'|}{|R \cup R'|} \quad (4)$$

$$\text{F1-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (5)$$

$$\text{loss} = - \sum_{i=0}^{S^2} \sum_{j=0}^B W_{ij}^{obj} [\hat{C}_i^j \log(C_i^j) + (1 - \hat{C}_i^j) \log(1 - C_i^j)]$$

$$- \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B (1 - W_{ij}^{obj}) [\hat{C}_i^j \log(C_i^j) + (1 - \hat{C}_i^j) \log(1 - C_i^j)]$$

$$- \sum_{i=0}^{S^2} \sum_{j=0}^B W_{ij}^{obj} \sum_{c=1}^C [\hat{P}_i^j(c) \log(P_i^j(c)) - (1 - \hat{P}_i^j(c)) \log(1 - P_i^j(c))] + 1$$

$$- \text{IOU} + \frac{\rho^2(b, b^{gt})}{c^2}$$

$$+ \frac{16}{\pi^2} \frac{\left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^4}{1 - \text{IOU} + \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2} \quad (6)[18]$$

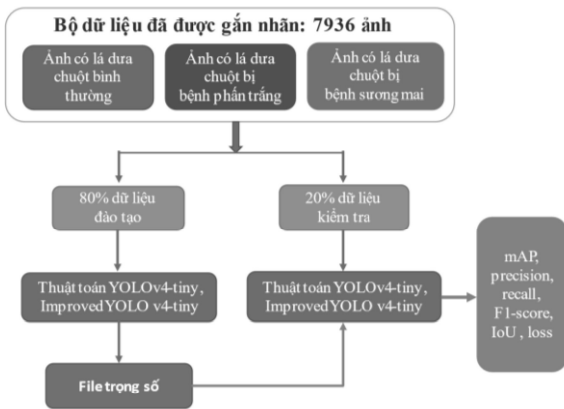
Các mô hình được đào tạo trên cấu hình phần cứng có thông số như Bảng 2. Quá trình đào tạo, tính toán và kiểm tra được thực hiện trên framework Darnet [16].

Bảng 2: Cấu trúc phần cứng đào tạo và kiểm tra 2 mô hình

Computer Configuration	Specific Parameters
CPU	Intel XEON
GPU	GTX 1060
Hệ điều hành	Windows 10
Bộ nhớ GPU	6 GB
Bộ nhớ Ram	64 GB

4. Kết quả và thử nghiệm

4.1. Đào tạo mô hình

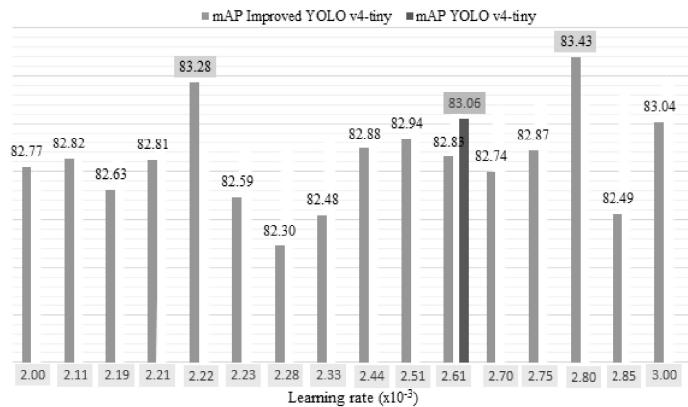


Hình 7: Lưu đồ quá trình đào tạo và kiểm tra hai mô hình

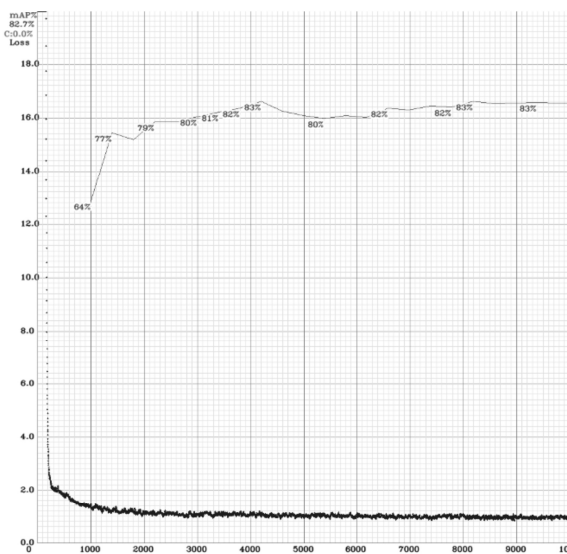
Bộ dữ liệu của chúng tôi gồm 7936 ảnh với 3 lớp là ảnh lá bị bệnh phấn trắng, lá bình thường và lá bị bệnh sương mai được chụp từ các thiết bị điện thoại, máy ảnh có độ phân giải khác

nau; các ảnh được gắn nhãn và chia bộ dữ liệu thành 80% ảnh để đào tạo, còn 20% ảnh để kiểm tra. Trên Hình 7 biểu diễn bộ dữ liệu của quá trình đào tạo và kiểm tra của hai mô hình.

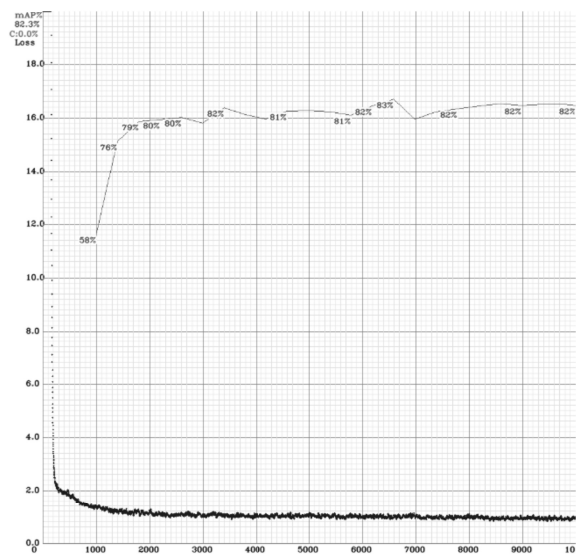
Trên Hình 8 là kết quả mAP khi đào tạo mô hình YOLO v4-tiny với learning rate bằng 0.00261 và mô hình Improved YOLO v4-tiny với giá trị learning rate khác nhau trong khoảng 0.002 đến 0.003, kết quả cho thấy độ chính xác trung bình mAP của mô hình YOLO v4-tiny là 83.06%, mô hình Improved YOLO v4-tiny cho độ chính xác trung bình mAP ở giá trị learning rate bằng 0.00222 và 0.0028 lần lượt là 83.28% và 83.43% cao hơn so với mô hình YOLO v4-tiny là 0.22% và 0.37%. Bảng 3 biểu thị chi tiết các thông số đào tạo của 2 mô hình này, kết quả đã cho thấy thêm kích thước của hai mô hình là không thay đổi bằng 22.4MB, giá trị recall của Improved YOLO v4-tiny cao hơn, trong khi giá trị precision, F1-score và average IoU thấp một ít, điều này không ảnh hưởng nhiều đến bài toán. Hàm loss của 2 mô hình thể hiện trên Hình 9 và Hình 10 là tương đồng nhau. Còn Hình 11, thể hiện một số kết quả kiểm tra của 2 thuật toán, nhận thấy mô hình cải tiến của chúng tôi phát hiện lá dừa chuột bị bệnh tốt hơn so với mô hình YOLO v4-tiny.



Hình 8. Biểu diễn mối quan hệ giữa learning rate với mAP của mô hình Improved YOLO v4-tiny và learning rate = 0.0261 của mô hình YOLO v4-tiny



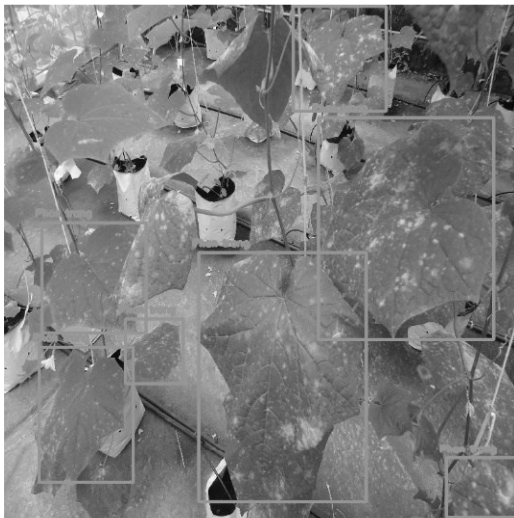
Hình 9: Đồ thị làm loss và mAP của YOLO v4-tiny với learning rate = 0.00261



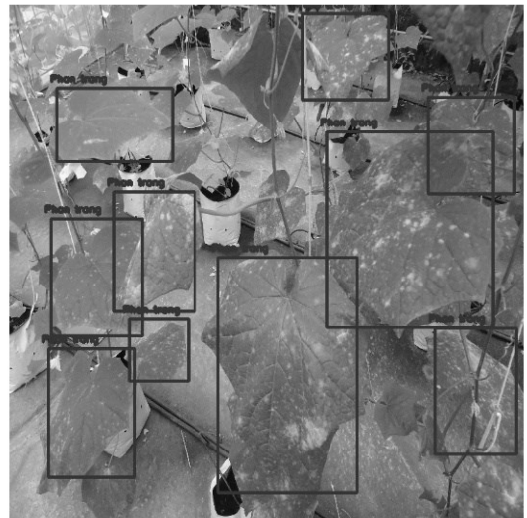
Hình 10: Đồ thị làm loss và mAP của Improved YOLO v4-tiny với learning rate = 0.00280

Bảng 3: Kết của các thông số mAP, precision, recall, F1, IoU của 2 mô hình

Model	Learning rate	mAP	precision	recall	F1-score	average IoU	Model sizes (MB)
YOLO v4-tiny	0.00261	83.06	0.61	0.83	0.70	52.57	22.4
Improved YOLO v4-tiny	0.00280	83.43	0.56	0.84	0.67	46.97	22.4



YOLO v4-tiny



Improved YOLO v4-tiny

Hình 11: Một số kết quả kiểm tra phát hiện bệnh của 2 mô hình

4.2. Mô hình phân cứng và thực nghiệm

Mô hình nhà lưới thực nghiệm trong nghiên cứu này có kích thước 10x5x3.5, được trồng dưa chuột theo phương pháp thủy canh, hệ thống cung cấp dinh dưỡng được thực hiện tự động qua hệ thống tủ điều khiển logic khả trình (PLC). Hình 12,

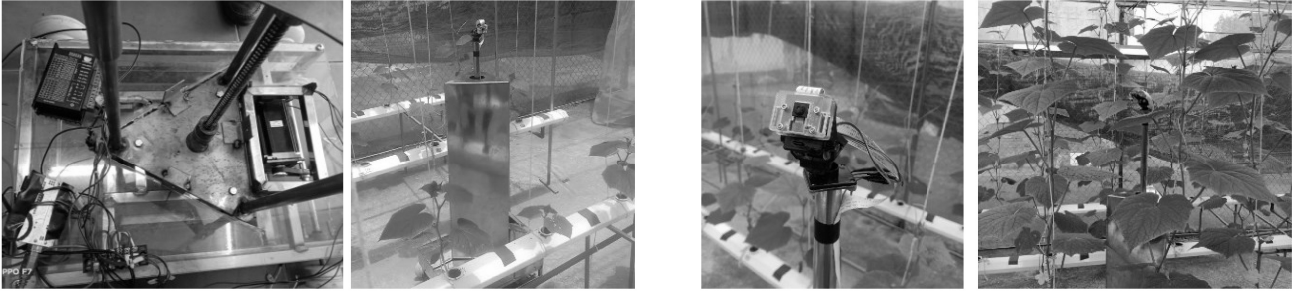
cho một số hình ảnh hệ thống được thiết kế và quá trình thực nghiệm hệ thống giám sát bệnh trên cây dưa chuột trồng trong nhà lưới với các giai đoạn phát triển của cây. Dưa chuột trồng trong nhà lưới có xuất hiện bệnh phân trắng từ giai đoạn 2. Hệ thống đã được đưa vào để thử nghiệm, hình ảnh thu được từ hệ thống với camera PI 5MB đã được cho lần lượt qua 2 thuật toán YOLO v4-tiny và Improved YOLO v4-tiny. Kết quả thu

được từ hệ thống giám sát thực nghiệm với 2 thuật toán cho thấy, thuật toán Improved YOLO v4-tiny đã cho kết quả nhận diện tốt hơn. Trên Hình 13 là một số kết quả, cho thấy các lá bị bệnh phân trắng đối với thuật toán YOLO v4-tiny không phát hiện được trong khi thuật toán Improved YOLO v4-tiny đã phát hiện ra. Bảng 4, kết quả so sánh 2 mô hình trên phần cứng Raspberry Pi 4 có RAM 1GB, cho thấy tốc độ xử lý của Raspberry trên 2 mô hình là tương đương nhau, điều này cũng phù hợp vì kích thước trọng số mô hình là bằng nhau 22.4 MB (Bảng 3). Hệ thống cũng đã hoạt động tốt với việc gửi tin nhắn khi phát hiện ra bệnh phân trắng trên cây dưa chuột cho nhà

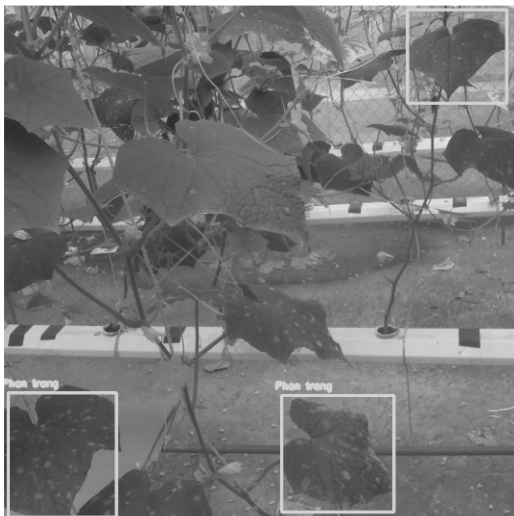
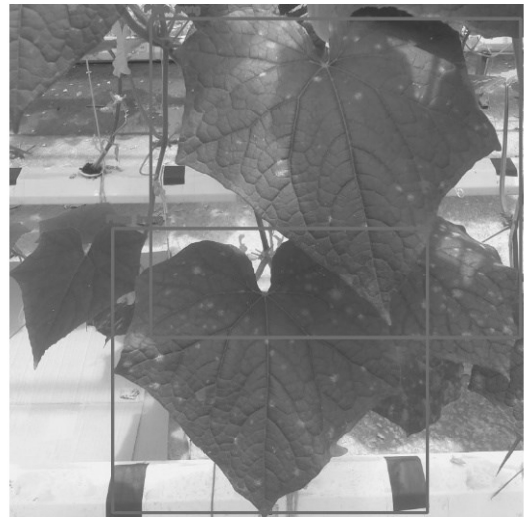
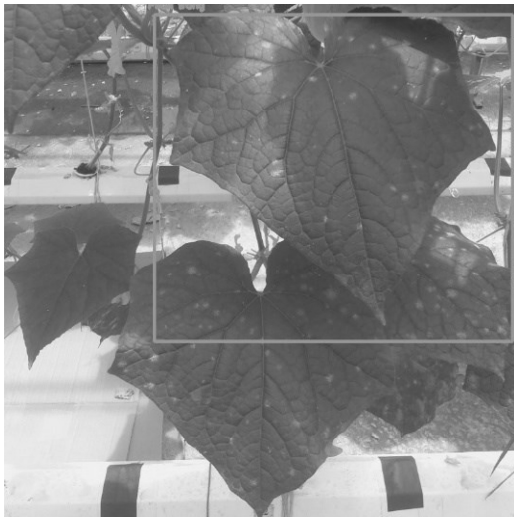
quản lý qua tin nhắn SMS. Đồng thời hình ảnh bị bệnh cũng được gửi tới email của người quản lý để xác thực và theo dõi (Hình 14).

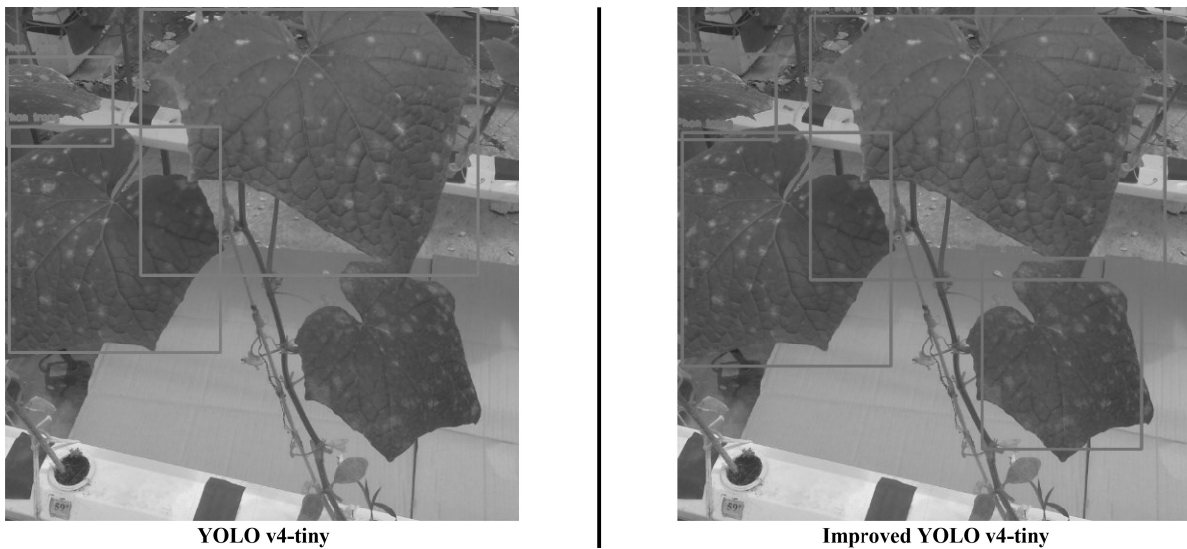
Bảng 4. So sánh khả năng xử lý hai thuật toán trên phần cứng Raspberry Pi 4, 1GB RAM

Mô hình	Thời gian trung bình xử lý 1 ảnh (giây)	Số khung hình hiển thị trong một giây (FPS)
YOLOv4-tiny	0.659	2.510
Improved YOLOv4-tiny	0.661	2.502

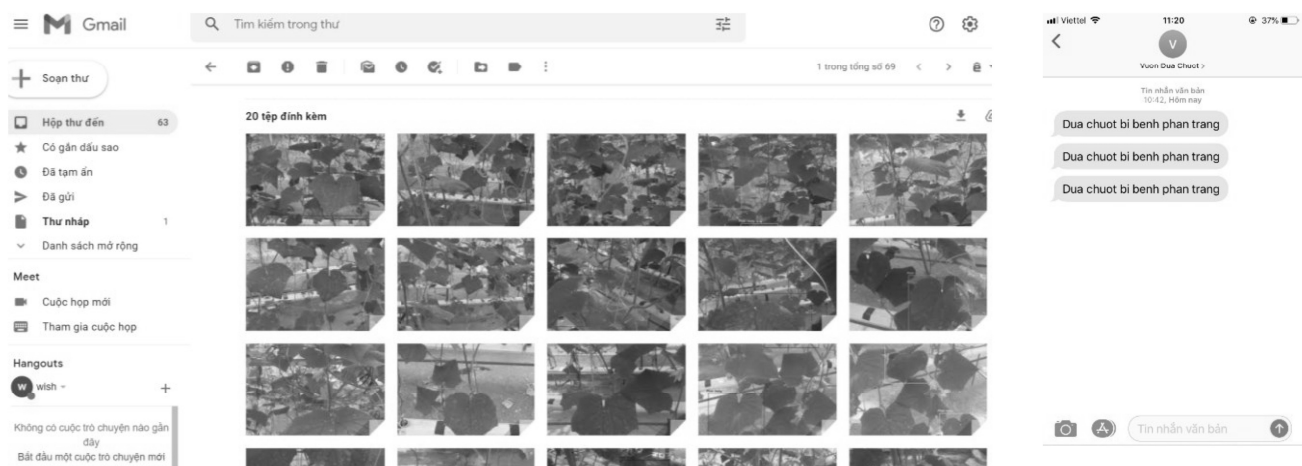


Hình 12: Một số hình ảnh thực nghiệm hệ thống





Hình 13: Kết quả thực nghiệm phát hiện bệnh của hệ thống với 2 mô hình



Hình 14: Kết quả lá bị bệnh được gửi lên email và tin nhắn được gửi cho người quản lý

5. Kết luận

Thuật toán YOLO v4-tiny cải tiến đã được tích hợp thành công vào hệ thống giám sát và cảnh báo bệnh sương mai và bệnh phấn trắng trên cây dưa chuột. Hệ thống đã được thử nghiệm thực tế tại một nhà kính tại Học viện Nông nghiệp Việt Nam. Kết quả bước đầu cho thấy hệ thống đã hoạt động tốt, độ chính xác nhận dạng các loại bệnh trên cây dưa chuột đạt tới 83.43%. Chúng tôi đã bước đầu xây dựng được bộ cơ sở dữ liệu về các bệnh trên cây dưa chuột, các ảnh thu thập được đánh số, ghi thuộc tính để phục vụ các nghiên cứu tiếp theo. Kết quả đạt được trong nghiên cứu này là tích cực, chúng thể hiện khả năng ứng dụng các thuật toán sử dụng trí tuệ nhân tạo (AI) vào các hệ thống giám sát, cảnh báo bệnh trên cây dưa chuột với phần cứng đơn giản. Điều này mở ra cơ hội dễ dàng áp dụng trong điều kiện thực tế ở Việt Nam do chi phí thực hiện hệ thống với giá thành thấp.

6. Tài liệu tham khảo

- [1] Abdullah Alsaeedi, Hassan El-Ramady, Tarek Alshaal, Mohamed El-Garawany, Nevien Elhawawt, Awadh Al-Otaibi (2019). *Silica nanoparticles boost*
- [2] E.V. Maas, G.J. Hoffman (1977). *Crop salt tolerance – current assessment*. J. Irrig. And Drainage Div. ASCE, 103 (IR2), pp. 115-134
- [3] Gulzar Ahmad Nayik, Amir Gull (2020). *Antioxidants in Vegetables and Nuts – Properties and Health Benefits*. ISBN 978-981-15-7469-6 ISBN 978-981-15-7470-2 (eBook) <https://doi.org/10.1007/978-981-15-7470-2>, Springer Nature Singapore Pte Ltd. Pp. 333-340.
- [4] A. Berdugo, R. Zito, S. Paulus and A.-K. Mahlein (2014). *Fusion of sensor data for the detection and differentiation of plant diseases in cucumber*. Plant Pathology (2014) 63, 1344–1356
- [5] Wang H, Jiang YP, Yu HJ et al., (2010). *Light quality affects incidence of powdery mildew, expression of defence-related genes and associated metabolism in cucumber plants*. European Journal of Plant Pathology 127, 125–35.
- [6] Watson A, Napier T (2009). *Disease of cucurbit vegetables*. Primefacts 832:1–6.

growth and productivity of cucumber under water deficit and salinity stresses by balancing nutrients uptake. Plant Physiology and Biochemistry 139, pp. 1–10.

- https://www.dpi.nsw.gov.au/__data/assets/pdf_file/0003/290244/diseases-of-cucurbit-vegetables.pdf.
- [7] Phạm Mỹ Linh, và cộng sự (2015). *Kết quả đánh giá giống dưa chuột trồng trong nhà lưới tại miền Bắc Việt Nam*. Tạp chí Nông nghiệp và phát triển Nông thôn – tháng 2.
- [8] Lê Thị Hường và cộng sự (2019). *Kết quả nghiên cứu khả năng sinh trưởng, phát triển và năng suất của một số giống dưa chuột trong vụ xuân ở huyện Ngọc Lặc, tỉnh Thanh Hóa*. Tạp chí khoa học Trường đại học Hồng Đức – số 44.
- [9] Trần Tô Lâm, Trần Thị Minh Hằng, Phạm Mỹ Linh (2020). *Đánh giá khả năng thích ứng của một số tổ hợp lai dưa chuột có triển vọng tại vùng đồng bằng Sông Hồng*. Tạp chí Khoa học Nông nghiệp Việt Nam 2020, 18(2): 81-87
- [10] S. Singh, T. F. Burks, W. S. Lee. (2005). *Autonomous robotic vehicle development for greenhouse spraying*. American Society of Agricultural Engineers ISSN 0001-2351.
- [11] Sammons, Philip J., Tomonari Furukawa, and Andrew Bulgin (2005). *Autonomous pesticide spraying robot for use in a greenhouse*. Australian Conference on Robotics and Automation. Vol. 1. No. 9.
- [12] Vakilian, K. A., & Massah, J. (2017). *A farmer-assistant robot for nitrogen fertilizing management of greenhouse crops*. Computers and electronics in agriculture, 139, 153-163.
- [13] K. Lin, L. Gong, Y. Huang, C. Liu, and J. Pan (2019). *Deep learning-based segmentation and quantification of cucumber powdery mildew using convolutional neural network*. Front. Plant Sci., vol. 10, no. February, pp. 1–10, doi: 10.3389/fpls.2019.00155.
- [14] P. Zhang, L. Yang, and D. Li (2020). *EfficientNet-B4-Ranger: A novel method for greenhouse cucumber disease recognition under natural complex environment*. Comput. Electron. Agric., vol. 176, no. January, p. 105652, doi: 10.1016/j.compag.2020.105652.
- [15] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao (2020). *Yolov4: Optimal speed and accuracy of object detection*. arXiv preprint arXiv:2004.10934.
- [16] Alexey Bochkovskiy (2020). *Darknet: Open Source Neural Networks in Python*. Available online: <https://github.com/AlexeyAB/darknet> (accessed on 2 November 2020).
- [17] Smith, L. N. (2018). *A disciplined approach to neural network hyper-parameters: Part 1--learning rate, batch size, momentum, and weight decay*. arXiv preprint arXiv:1803.09820.
- [18] Jiang, Z., Zhao, L., Li, S., & Jia, Y. (2020). *Real-time object detection method based on improved YOLOv4-tiny*. arXiv preprint arXiv:2011.04244.