ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA (HCMUT)
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

ĐỒ ÁN TỐT NGHIỆP (KHOA HỌC MÁY TÍNH) (CO4337)

PHÁT TRIỂN ỨNG DỤNG AI PHÁT HIỆN VÀ NHẬN DIỆN SÂU BỆNH GÂY HẠI CÂY TRỒNG

NGÀNH: KHOA HỌC MÁY TÍNH

Hội đồng:

Khoa học máy tính - Hội đồng 2

Giảng viên hướng dẫn:

TS. Nguyễn Tiến Thịnh - HCMUT

ThS. Băng Ngọc Bảo Tâm - HCMUT

Giảng viên phản biện:

ThS. Mai Xuân Toàn - HCMUT

Sinh viên thực hiện:

Phạm Ngọc Tân - 1915071

Thành phố Hồ Chí Minh, tháng 12/2023

Ngày ký

Thầy Nguyễn Tiến Thịnh
Tiến sĩ
Khoa Khoa học và Kỹ thuật Máy tính
Giảng viên hướng dẫn

Thầy Băng Ngọc Bảo Tâm
Thạc sĩ
Khoa Khoa học và Kỹ thuật Máy tính
Giảng viên hướng dẫn

ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC BÁCH KHOA

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

KHOA: KH & KT Máy tính
BỘ MÔN: KHMT

NHIỆM VỤ LUẬN VĂN/ ĐỒ ÁN TỐT NGHIỆP

Chú ý: Sinh viên phải dán tờ này vào trang nhất của bản thuyết trình

HỌ VÀ TÊN: Phạm Ngọc Tân MSSV: 1915071
NGÀNH: Khoa học Máy tính LỚP:

1. Đầu đề luận văn/ đồ án tốt nghiệp:
Phát triển ứng dụng AI phát hiện và nhận diện sâu bệnh gây hại cây trồng

2. Nhiệm vụ (yêu cầu về nội dung và số liệu ban đầu):
a. Tìm hiểu kiến thức nền tảng AI và kiến thức nền tảng xây dựng ứng dụng web.
b. Phân tích bài toán phát hiện và nhận diện sâu bệnh gây hại cây trồng và phát triển mô hình AI giải quyết bài toán.
c. Tìm hiểu các độ đo đánh giá hiệu quả mô hình AI.
d. Thử nghiệm và đánh giá mô hình AI đã xây dựng.
e. Phát triển ứng dụng web tích hợp mô hình AI và đánh giá hiệu quả của ứng dụng.
f. Viết báo cáo và hoàn thiện đề tài.

3. Ngày giao nhiệm vụ: 28/08/2023
4. Ngày hoàn thành nhiệm vụ: 04/01/2024
5. Họ tên giảng viên hướng dẫn: **Phần hướng dẫn:**
1) Nguyễn Tiến Thịnh hướng dẫn các nội dung từ (a) đến (f).
2) Băng Ngọc Bảo Tâm hướng dẫn các nội dung từ (a) đến (f).

Nội dung và yêu cầu LVTN/ ĐATN đã được thông qua Bộ môn.

Ngày ..... tháng ..... năm .....
CHỦ NHIỆM BỘ MÔN **GIẢNG VIÊN HƯỚNG DẪN CHÍNH**

PHẦN DÀNH CHO KHOA, BỘ MÔN:
Người duyệt (chấm sơ bộ):
Đơn vị:
Ngày bảo vệ:
Điểm tổng kết:
Nơi lưu trữ LVTN/ĐATN:

356

TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KH & KT MÁY TÍNH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

Ngày tháng năm

PHIẾU ĐÁNH GIÁ LUẬN VĂN/ ĐỒ ÁN TỐT NGHIỆP

(Dành cho người hướng dẫn)

1. Họ và tên SV: Phạm Ngọc Tân
MSSV: 1915071 Ngành (chuyên ngành): Khoa học Máy tính
2. Đề tài: Phát triển ứng dụng AI phát hiện và nhận diện sâu bệnh gây hại cây trồng

3. Họ tên người hướng dẫn: Nguyễn Tiến Thịnh

4. Tổng quát về bản thuyết minh:
Số trang: 98 Số chương: 07
Số bảng số liệu: 16 Số hình vẽ: 37
Số tài liệu tham khảo: 60 Phần mềm tính toán:
Hiện vật (sản phẩm)
5. Những ưu điểm chính của LV/ ĐATN:

- Xây dựng được một ứng dụng web cơ bản với chức năng chính là phát hiện và nhận diện sâu bệnh gây hại cây trồng (gồm tám loại cây và bốn mươi loại sâu bệnh) theo hướng tiếp cận Deep Learning.
- Biết vận dụng và kết hợp các mô hình Deep Learning sử dụng trong các bài toán khác nhau để tăng độ chính xác của việc phát hiện và nhận diện sâu bệnh gây hại cây trồng như Unet cho bài toán Leaf Segmentation và YOLOv8 cho bài toán Classification.

6. Những thiếu sót chính của LV/ĐATN:
- Bộ dữ liệu chưa đa dạng, không chứa yếu tố ngoại cảnh và không chứa các bộ phận khác của cây trồng.
- Các đánh giá phân tích chưa kỹ lưỡng, còn mang tính chủ quan và các thử nghiệm chưa được thực hiện nhiều cần bổ sung thêm.
- Chưa đánh giá được việc triển khai ứng dụng vào trong thực tiễn.

7. Đề nghị: Được bảo vệ □ Bổ sung thêm để bảo vệ □ Không được bảo vệ □
8. Các câu hỏi SV phải trả lời trước Hội đồng:
a. Hãy trình bày khả năng mở rộng của đề tài bằng cách nêu một vài hướng phát triển trong tương lai và ý tưởng hoặc giải pháp cho hướng phát triển đó.

9. Đánh giá chung (bằng chữ: Xuất sắc, Giỏi, Khá, TB): Điểm : 9 /10

Ký tên

Nguyễn Tiến Thịnh

356

TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KH & KT MÁY TÍNH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

Ngày 28 tháng 12 năm 2023

PHIẾU ĐÁNH GIÁ LUẬN VĂN/ ĐỒ ÁN TỐT NGHIỆP

(Dành cho người hướng dẫn/phản biện)

1. Họ và tên SV: Phạm Ngọc Tân
   MSSV: 1915071 Ngành (chuyên ngành): Khoa học máy tính
2. Đề tài: Phát triển ứng dụng AI phát hiện và nhận diện sâu bệnh gây hại cây trồng
3. Họ tên người phản biện: Mai Xuân Toàn
4. Tổng quát về bản thuyết minh:
   Số trang: 98 Số chương: 7
   Số bảng số liệu: 16 Số hình vẽ: 37
   Số tài liệu tham khảo: 60 Phần mềm tính toán:
   Hiện vật (sản phẩm):
5. Những ưu điểm chính của LVTN:
   - Luận văn có bố cục rõ ràng, dễ hiểu và mạch lạc.
   - Tác giả có định hướng nghiên cứu rõ ràng.
   - Sinh viên biết tìm hiểu, nghiên cứu về kiến thức xử lý ảnh và phát triển ứng dụng web.

6. Những thiếu sót chính của LV/ĐATN:
   - Báo cáo còn một số lỗi chính tả, thuật ngữ đưa ra tạo sự nhầm lẫn.

7. Đề nghị: Được bảo vệ ☒ Bổ sung thêm để bảo vệ ☐ Không được bảo vệ ☐
8. Các câu hỏi SV phải trả lời trước Hội đồng:

9. Đánh giá chung (bằng chữ: giỏi, khá, TB): Điểm : 9.0 /10

Ký tên (ghi rõ họ tên)

Mai Xuân Toàn

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi, được thực hiện dưới sự hướng dẫn của TS. Nguyễn Tiến Thịnh và ThS. Băng Ngọc Bảo Tâm. Các kết quả nghiên cứu của tôi là hợp pháp và chưa được công bố dưới bất kỳ hình thức nào trước đó.

Ngoài ra, trong quá trình nghiên cứu này, tôi đã trích dẫn đầy đủ các nguồn thông tin, tư liệu, kết quả của các tác giả, tổ chức khác theo đúng Nguyên tắc tuân thủ các nguyên tắc đạo đức khi xây dựng luận văn tốt nghiệp.

Trong mọi trường hợp đạo văn, tôi đứng trước hành động của mình và sẽ chịu trách nhiệm về chúng. Do đó, Trường Đại học Bách Khoa - Đại học Quốc gia Thành phố Hồ Chí Minh không chịu trách nhiệm về bất kỳ hành vi vi phạm bản quyền nào trong quá trình nghiên cứu của tôi.

TP. Hồ Chí Minh, Tháng 12/2023
Tác giả
Phạm Ngọc Tân

LỜI CẢM ƠN

Trước hết, tôi xin gửi lời cảm ơn sâu sắc đến TS. Nguyễn Tiến Thịnh đã giao cho tôi đề tài này và cho tôi cơ hội duy nhất được làm việc dưới sự hướng dẫn của thầy. Đồ án tốt nghiệp này cho đến nay là trải nghiệm học thuật tuyệt vời nhất mà tôi từng có bởi niềm đam mê, tinh thần trách nhiệm và sự cống hiến, theo đuổi sự hoàn hảo, những lời khuyên cũng như sự hỗ trợ vô cùng tận tâm mà thầy Nguyễn Tiến Thịnh đã dành cho tôi.

Tôi cũng xin cảm ơn ThS. Băng Ngọc Bảo Tâm đã cung cấp cho tôi cái nhìn sâu sắc, lời khuyên và hỗ trợ kỹ thuật khi tôi cần nhất. Thầy là một nguồn động viên, cảm hứng và thông tin to lớn trong suốt thời gian của toàn bộ Đồ án.

Xin chân thành cảm ơn Trường Đại học Bách Khoa - Đại học Quốc gia Thành phố Hồ Chí Minh đã tạo điều kiện học tập và nghiên cứu rất tốt và chuyên nghiệp cho tôi. Đây là nơi tôi gặp được những cộng sự tốt nhất để hoàn thành Đồ án này và nhận được rất nhiều sự hỗ trợ từ các thầy cô tâm huyết, yêu nghề trong 4 năm qua.

Tôi xin bày tỏ lòng biết ơn sâu sắc nhất tới gia đình, những người luôn sát cánh bên tôi, đã không ngừng yêu thương, ủng hộ, động viên tôi học tập và trưởng thành để hoàn thiện hơn mỗi ngày.

TỔNG QUAN

Trong nông nghiệp, việc chẩn đoán sớm và nhận diện các vấn đề liên quan đến sức khỏe cây trồng đóng vai trò quan trọng trong việc duy trì năng suất và chất lượng sản phẩm. Tuy nhiên, thực hiện việc này thông qua các phương pháp truyền thống đang đối mặt với những thách thức, bao gồm sự tốn kém về thời gian, công sức và độ chính xác không cao do phụ thuộc vào con người.

Điều này đã thúc đẩy sự gia tăng về sự hứa hẹn của việc áp dụng AI và **Computer Vision (Thị giác máy tính)** vào lĩnh vực nông nghiệp để tạo ra những giải pháp thông minh và tự động hóa. Trong bối cảnh đó, dự án này nhằm mục tiêu xây dựng một ứng dụng thông minh cho phép người dùng nhận diện và chẩn đoán sớm các vấn đề liên quan đến bệnh tật của cây trồng thông qua hình ảnh.

Để đạt được mục tiêu này, một số các thử nghiệm đã được thực hiện nhằm đánh giá hiệu suất của các mô hình và chọn ra mô hình phù hợp, bắt đầu từ việc tìm kiếm, thu thập nhiều nguồn dữ liệu khác nhau về các loại cây cũng như các loại sâu bệnh gây hại, cho đến việc huấn luyện, đánh giá và cải thiện hiệu suất mô hình. Với kết quả về hiệu suất vượt trội, YOLOv8 đã chứng minh khả năng mạnh mẽ trong việc phát hiện và nhận diện sâu bệnh trên cây trồng.

Với cốt lõi là mô hình YOLOv8 phân loại sâu bệnh, tôi đã xây dựng một ứng dụng với chức năng chính là cho phép người dùng phát hiện sâu bệnh trên cây trồng thông qua ảnh chụp lá cây. Tuy nhiên, việc triển khai và đánh giá trong môi trường thực tế cần được tiếp tục nghiên cứu để đảm bảo ứng dụng thực tế của hệ thống.

Thông qua Đồ án này, tôi hi vọng sẽ phát triển được một ứng dụng hiệu quả có khả năng giải quyết các thách thức đặt ra với vấn dề phát hiện và nhận diện sâu bệnh trên cây trồng và góp phần rút ngắn khoảng cách giữa nông nghiệp và công nghệ.

MỤC LỤC

Lời cam đoan **v**

Lời cảm ơn **vii**

Tổng quan **ix**

Danh sách hình vẽ **xv**

Danh sách bảng **xvii**

1 Giới thiệu **1**
1.1 Trình bày vấn đề 1
1.2 Động lực 2
1.3 Mục tiêu 2
1.4 Cấu trúc báo cáo. 3

2 Cơ sở lý thuyết **5**
2.1 Kiến thức về các loại cây và các loại sâu bệnh gây hại. 5
2.1.1 Cà phê 5
2.1.2 Ngô (Bắp) 6
2.1.3 Nho 8
2.1.4 Mít 9
2.1.5 Xoài 10
2.1.6 Lúa 12
2.1.7 Mía 14
2.1.8 Trà. 15
2.2 Phân loại hình ảnh 17
2.2.1 Phân loại hình ảnh là gì? 17
2.2.2 Các phương pháp phân loại hình ảnh. 17
2.2.3 Các bước phân loại hình ảnh 18
2.2.4 Những thách thức đối với bài toán phân loại hình ảnh 20
2.3 Phân đoạn hình ảnh 21
2.3.1 Phân đoạn hình ảnh là gì?. 21
2.3.2 Các loại tác vụ phân đoạn hình ảnh 21

3 Những công trình liên quan 23
3.1 Các kỹ thuật truyền thống 23
3.1.1 SVM 23
3.1.2 KNN. 25
3.2 Các kỹ thuật dựa trên Deep Learning 27
3.2.1 CNN 27
3.2.2 YOLO 31

4 Thử nghiệm 37
4.1 Một số mô hình được sử dụng 37
4.1.1 YOLOv8 37
4.2 Các tập dữ liệu được sử dụng 39
4.2.1 Giới thiệu tập dữ liệu PlantVillage 39
4.2.2 Tập dữ liệu chính cho tác vụ phân loại sâu bệnh trên nhiều loại cây 39
4.2.3 Tập dữ liệu kiểm tra overfitting cho tác vụ phân loại sâu bệnh trên nhiều loại cây 41
4.2.4 Tập dữ liệu cho tác vụ phân đoạn lá nho 42
4.2.5 Tập dữ liệu cho tác vụ phân loại sâu bệnh trên cây nho 43
4.3 Một số thử nghiệm đã được thực hiện 44
4.3.1 Lựa chọn mô hình phân loại sâu bệnh 44
4.3.2 Kiểm tra overfitting của mô hình phân loại sâu bệnh 44
4.3.3 Lựa chọn ngưỡng tin cậy cho ứng dụng. 44
4.3.4 Phân loại sâu bệnh kết hợp tiền xử lý phân đoạn lá cây 45

5 Phát triển ứng dụng 47
5.1 Phân tích yêu cầu 47
5.1.1 Đối tượng người dùng 47
5.1.2 Phạm vi và loại trừ 48
5.1.3 Yêu cầu 49
5.2 Kiến trúc hệ thống 53
5.3 Công nghệ 54
5.3.1 Front-end: Vue.js 54
5.3.2 Back-end: FastAPI 56
5.3.3 Cơ sở dữ liệu: Tortoise ORM và PostgreSQL 57

5.4 Giao diện và luồng hoạt động của ứng dụng 60
5.4.1 Trang chủ 60
5.4.2 Trang chức năng dự đoán trên một ảnh 60
5.4.3 Trang chức năng dự đoán trên nhiều ảnh 63
5.4.4 Trang thông tin chi tiết cho mỗi ảnh 65
5.5 Cơ sở dữ liệu 65
5.5.1 Thu thập dữ liệu 65
5.5.2 Tổ chức cơ sở dữ liệu với Tortoise ORM 66

6 Đánh giá 67
6.1 Các thông số và tiêu chí đánh giá 67
6.1.1 Đánh giá mô hình phân loại sâu bệnh 67
6.1.2 Đánh giá toàn bộ ứng dụng 68
6.2 Kết quả đánh giá mô hình phân loại sâu bệnh 70

7 Tổng kết và hướng phát triển 71
7.1 Kết quả đạt được 71
7.1.1 Đối với cơ sở lý thuyết và công nghệ 71
7.1.2 Đối với mục tiêu đề tài 72
7.2 Những thuận lợi và khó khăn trong quá trình phát triển ứng dụng 72
7.2.1 Thuận lợi 72
7.2.2 Khó khăn 72
7.3 Hướng phát triển trong tương lai 73

Tài liệu tham khảo 75

DANH SÁCH HÌNH VẼ

2.1 Lá cây cà phê khỏe mạnh 5
2.2 Lá cây ngô (bắp) khỏe mạnh 6
2.3 Lá cây nho khỏe mạnh 8
2.4 Lá cây mít khỏe mạnh 9
2.5 Lá cây xoài khỏe mạnh 10
2.6 Lá cây lúa khỏe mạnh 12
2.7 Lá cây mía khỏe mạnh 14
2.8 Lá cây trà khỏe mạnh 15
2.9 Một số thách thức trong bài toán phân loại hình ảnh 20
2.10 Input (bên trái) và Output (bên phải) của mô hình phân đoạn hình ảnh đối với bài toán một đối tượng. Mỗi một nhãn sẽ được thể hiện bởi một màu sắc khác nhau. Màu xám là nền, màu vàng là đường viền của ảnh và màu tím là nằm bên trong vật thể. 21
2.11 Các loại tác vụ phân đoạn hình ảnh 22

3.1 Support Vector Machine trong không gian hai chiều và ba chiều 24
3.2 Ví dụ minh họa cho giải thuật K-Nearest Neighbor 26
3.3 Phân loại bệnh ở thực vật sử dụng Convolutional Neural Network 29
3.4 Ảnh được chia thành ma trận ô vuông 7x7 32
3.5 Mỗi ô vuông chịu trách nhiệm dự đoán 2 boundary box của đối tượng 33
3.6 Vector dự đoán cho mỗi ô vuông 33

4.1 Kiến trúc mạng YOLOv8 38
4.2 Một vài mẫu trong tập dữ liệu PlantVillage 39
4.3 Phân bố dữ liệu ở một số class trong tập dữ liệu chính . . . 41
4.4 Một vài mẫu trong tập dữ liệu dùng để kiểm tra overfitting của mô hình phân loại sâu bệnh 42
4.5 Một vài mẫu trong tập dữ liệu dùng để phân đoạn lá nho từ Roboflow 43

4.6 Một vài mẫu trong tập dữ liệu dùng để phân loại sâu bệnh trên cây nho 43
4.7 Kiến trúc mô hình thử nghiệm áp dụng tiền xử lý phân đoạn lá cây 45

5.1 Activity diagram cho chức năng dự đoán trên một ảnh . . . 49
5.2 Activity diagram cho chức năng dự đoán trên nhiều ảnh . 51
5.3 Kiến trúc tổng quát của hệ thống 54
5.4 Trang chủ 60
5.5 Trang chức năng dự đoán trên một ảnh 61
5.6 Giao diện sau khi ảnh được tải lên thành công 61
5.7 Giao diện kết quả dự đoán trên một ảnh (trường hợp cây khỏe mạnh) 62
5.8 Giao diện kết quả dự đoán trên một ảnh (trường hợp cây bị sâu bệnh gây hại) 62
5.9 Giao diện kết quả dự đoán trên một ảnh (trường hợp ứng dụng không thể dự đoán loại cây và tình trạng của cây) . . 63
5.10 Giao diện sau khi các ảnh được tải lên thành công 63
5.11 Giao diện kết quả dự đoán trên nhiều ảnh 64
5.12 File CSV chứa kết quả dự đoán trên nhiều ảnh 64
5.13 Trang thông tin chi tiết cho mỗi ảnh 65

DANH SÁCH BẢNG

2.1 Một số loại sâu bệnh trên cây cà phê 6
2.2 Một số loại sâu bệnh trên cây ngô (bắp) 7
2.3 Một số loại sâu bệnh trên cây nho 8
2.4 Một số loại sâu bệnh trên cây mít 9
2.5 Một số loại sâu bệnh trên cây xoài 10
2.6 Một số loại sâu bệnh trên cây lúa 12
2.7 Một số loại sâu bệnh trên cây mía 14
2.8 Một số loại sâu bệnh trên cây trà 15

3.1 Những nghiên cứu đã được thực hiện bằng phương pháp sử dụng SVM 24
3.2 Những nghiên cứu đã được thực hiện bằng phương pháp sử dụng KNN 26
3.3 Những nghiên cứu đã được thực hiện bằng phương pháp sử dụng CNN 30
3.4 Những nghiên cứu đã được thực hiện bằng phương pháp sử dụng YOLO 34

4.1 Số lượng ảnh của các class trong tập dữ liệu huấn luyện mô hình phân loại 40
4.2 Kết quả phân loại sâu bệnh trên 8 loại cây với YOLOv5 và YOLOv8 44
4.3 Kết quả phân loại sâu bệnh trên 2 tập dữ liệu 44
4.4 Tỉ lệ dự đoán đúng của mô hình phân loại sâu bệnh với các ngưỡng khác nhau 45
4.5 Hiệu suất của mô hình phân loại sâu bệnh trên cây nho trong 2 trường hợp 46

5.1 Use case scenario cho chức năng dự đoán trên một ảnh . 50
5.2 Use case scenario cho chức năng dự đoán trên nhiều ảnh 52

1

GIỚI THIỆU

Chương đầu tiên tập trung làm rõ về ngữ cảnh và động lực cũng như mục tiêu chính để thực hiện Đồ án này, đồng thời trình bày tóm tắt về cấu trúc của báo cáo.

1.1. TRÌNH BÀY VẤN ĐỀ

Ngành nông nghiệp đóng một vai trò không thể phủ nhận trong việc cung cấp lương thực đáp ứng nhu cầu ngày càng tăng của dân số toàn cầu. Tuy nhiên, những người làm nông gặp phải những thách thức lớn trong việc duy trì và bảo vệ sức khỏe của các loại cây trồng, đặc biệt là trước nguy cơ từ các sâu bệnh. Đây không chỉ là vấn đề về năng suất mà còn đặt ra rủi ro về an ninh lương thực.

Cách tiếp cận truyền thống trong việc chẩn đoán sâu bệnh dựa vào quan sát trực quan của những người có kinh nghiệm, điều này không chỉ tốn nhiều thời gian và lao động mà còn dễ gây ra sai sót. Điều này trở thành một thách thức đáng kể trong bối cảnh nhu cầu nông nghiệp bền vững và quản lý thông minh hơn về nguồn lực.

Tuy nhiên, sự phát triển nhanh chóng của AI và Computer Vision mở ra cơ hội mới để cải thiện tình hình. Chúng ta có thể phát triển các hệ thống tự động có khả năng phát hiện và phân loại các bệnh thực vật với độ chính xác cao hơn. Sự tiến bộ này không chỉ giúp nâng cao năng suất mà còn mang lại khả năng dự đoán và phòng ngừa sớm hơn cho nguy cơ từ các bệnh trên cây trồng, giúp nông dân có thể áp dụng các biện pháp phòng tránh và can thiệp kịp thời.

Sự kết hợp giữa công nghệ và nông nghiệp đang mở ra một tiềm năng lớn cho sự phát triển bền vững và tăng cường khả năng đối phó với thách thức từ các sâu bệnh trong nông nghiệp.

1.2. ĐỘNG LỰC

Động lực cho việc thực hiện Đồ án này bắt nguồn từ nhu cầu thu hẹp khoảng cách giữa các phương pháp chẩn đoán bệnh thực vật truyền thống và khả năng do công nghệ AI mang lại. Bằng cách phát triển một ứng dụng dựa trên AI để phân loại bệnh cây trồng, chúng tôi mong muốn cung cấp cho nông dân và các chuyên gia nông nghiệp một công cụ thân thiện với người dùng có thể xác định nhanh chóng và chính xác các loại bệnh trên cây trồng của họ, cho phép can thiệp sớm, điều trị chính xác và quản lý sâu bệnh một cách chủ động, giúp cải thiện sức khỏe cây trồng, tăng năng suất và thực hiện nông nghiệp bền vững. Hơn nữa, sự phát triển của một ứng dụng như vậy phù hợp với các mục tiêu rộng lớn hơn về nông nghiệp kỹ thuật số. Bằng cách khai thác tiềm năng của AI, chúng ta có thể trao quyền cho nông dân đưa ra quyết định dựa trên dữ liệu, nâng cao năng suất và lợi nhuận của họ, đồng thời đóng góp vào nỗ lực toàn cầu nhằm đảm bảo an ninh lương thực.

1.3. MỤC TIÊU

Mục tiêu của Đồ án này là phát triển một ứng dụng dựa trên AI để phát hiện và phân loại sâu bệnh trên cây trồng. Mục tiêu chính là tận dụng các kỹ thuật Machine Learning và Computer Vision để xác định và chẩn đoán chính xác các vấn đề về sức khỏe của cây trồng một cách kịp thời và hiệu quả. Ứng dụng này nhằm cung cấp cho nông dân và các chuyên gia nông nghiệp một công cụ đáng tin cậy để phát hiện sớm, chủ động quản lý và điều trị các loại sâu bệnh hại cây trồng.

1.4. CẤU TRÚC BÁO CÁO

Bản báo cáo này bao gồm 7 chương:

- **Chương 1 - Giới thiệu**: Tổng quan về Ngữ cảnh, động lực thực hiện đề tài và mục tiêu của Đồ án.
- **Chương 2 - Cơ sở lý thuyết**: Cơ sở kiến thức liên quan đến một số loại cây cũng như bài toán phát hiện và nhận diện sâu bệnh gây hại cây trồng.
- **Chương 3 - Những công trình liên quan**: Những nghiên cứu về nhận diện sâu bệnh gây hại cây trồng bằng nhiều phương pháp khác nhau đã được thực hiện.
- **Chương 4 - Thử nghiệm**: Một số thử nghiệm để xây dựng mô hình cho bài toán phát hiện và nhận diện sâu bệnh gây hại cây trồng.
- **Chương 5 - Phát triển ứng dụng**: Quá trình phát triển ứng dụng bao gồm: phân tích yêu cầu, tìm hiểu các công nghệ, thiết kế giao diện, thu thập dữ liệu và xây dựng ứng dụng hòa chỉnh.
- **Chương 6 - Đánh giá**: Các thông số và tiêu chí đánh giá mô hình cho bài toán phát hiện và nhận diện sâu bệnh cũng như toàn bộ ứng dụng.
- **Chương 7 - Tổng kết và hướng phát triển**: Tổng kết ngắn gọn về những kết quả đã đạt được, những thuận lợi, khó khăn trong quá trình thực hiện Đồ án và hướng phát triển trong tương lai.

2

CƠ SỞ LÝ THUYẾT

Chương này trước tiên cung cấp kiến thức cơ bản về những loại cây cũng như những loại sâu bệnh gây hại tương ứng nằm trong phạm vi của ứng dụng. Sau đó, nội dung của chương tập trung vào hai bài toán quan trọng trong thị giác máy tính và cũng là nền tảng lý thuyết quan trọng của Đồ án này: phân loại hình ảnh và phân đoạn hình ảnh.

2.1. KIẾN THỨC VỀ CÁC LOẠI CÂY VÀ CÁC LOẠI SÂU BỆNH GÂY HẠI

Trong phạm vi Đồ án này, các phân tích, dự đoán xoay quanh dữ liệu về 8 loại cây, được chia thành 40 class, bao gồm những class của cây khỏe mạnh và cả những cây bị sâu bệnh gây hại. Dưới đây là chi tiết về từng loại cây và các loại sâu bệnh tương ứng cùng với triệu chứng, nguyên nhân của loại sâu bệnh đó.

2.1.1. CÀ PHÊ

Hình 2.1: Lá cây cà phê khỏe mạnh

Bảng 2.1: Một số loại sâu bệnh trên cây cà phê

Sâu bệnh	Triệu chứng	Nguyên nhân
Sâu ăn lá (Miner)	Lá cây cà phê có các vết mỏng và rõ, như một hệ thống các lỗ nhỏ. Hệ thống mạch lá bị tổn thương, gây mất nước và chất dinh dưỡng. Lá mất màu, héo và có thể rụng.	Bệnh do sâu bệnh gọi là "miner"(một loại sâu nhỏ) gặm vào lá cây. Sâu miner tạo ra các đường mòn trên lá, tiêu thụ mô lá cây, gây ra sự suy yếu cho cây.
Bệnh nấm Phoma	Gây ra các vết đen hoặc nâu trên lá, thân cây và quả cà phê. Làm hỏng cấu trúc lá, gây ra sự suy yếu và rụng lá.	Bệnh do nấm Phoma gây ra thường phát triển trong điều kiện ẩm ướt và nhiệt độ môi trường cao.
Bệnh gỉ sắt	Các vết sắt gỉ xuất hiện trên lá cây, có thể là màu vàng, cam hoặc nâu đậm. Lá bị thối và rụng từ cây. Khi bệnh nặng, có thể làm giảm năng suất và chất lượng của quả cà phê.	Do nấm biến chủng thuộc họ Puccinia gây ra. Phát triển mạnh mẽ trong điều kiện ẩm ướt, đặc biệt là khi có sương mù hoặc độ ẩm cao.

2.1.2. NGÔ (BẮP)

Hình 2.2: Lá cây ngô (bắp) khỏe mạnh

Bảng 2.2: Một số loại sâu bệnh trên cây ngô (bắp)

Sâu bệnh	**Triệu chứng**	**Nguyên nhân**
Bệnh đốm lá Cercospora (Bệnh bạc lá xám)	Gây ra các đốm màu nâu hoặc xám trên lá cây ngô. Những vết này sau đó mở rộng và trở thành các vết lớn hơn có thể che phủ một phần lớn lá. Lá bị héo và có thể rụng, gây suy yếu cho cây ngô.	Bệnh do nấm Cercospora zeae-maydis gây ra. Phát triển mạnh trong điều kiện ẩm ướt và nhiệt độ ấm.
Bệnh gỉ sắt	Gây ra các vết sắt gỉ màu vàng hoặc cam trên lá ngô. Vết bệnh thường xuất hiện dưới dạng các đốm nhỏ ban đầu, sau đó mở rộng và phủ kín lá. Làm mất màu lá, làm ảnh hưởng đến sự phát triển của cây ngô.	Do nấm Puccinia sorghi gây ra. Phát triển và lan rộng nhanh chóng trong điều kiện ẩm ướt và nhiệt độ ấm áp.
Bệnh bạc lá miền Bắc (Bệnh cháy bìa lá)	Gây ra các vết thối mục nâu dài trên lá ngô. Những vết này thường xuất hiện song song theo các đường vân trên lá, từ gốc đến ngọn.	Do nấm Setosphaeria turcica (hay Bipolaris maydis) gây ra. Phát triển và lan rộng nhanh chóng trong điều kiện ẩm ướt, đặc biệt là khi có sương mù hoặc độ ẩm cao.

2.1.3. Nho

Hình 2.3: Lá cây nho khỏe mạnh

Bảng 2.3: Một số loại sâu bệnh trên cây nho

Sâu bệnh	Triệu chứng	Nguyên nhân
Bệnh thối đen	Gây ra các vết thối mục màu đen hoặc nâu trên lá. Các vết này sau đó mở rộng và lan ra các cành, quả và thân cây. Quả bị thối và có thể rụng.	Do nấm Guignardia bidwellii gây ra. Phát triển mạnh trong điều kiện ẩm ướt và nhiệt độ ấm áp.
Bệnh Esca (Bệnh sởi đen)	Gây ra các vết thối mục trên lá, thường là các đốm màu nâu hoặc đen. Có thể xuất hiện các đường thối trên cành và thân cây. Lá và cành có thể héo và rụng.	Do sự kết hợp giữa một số loại nấm, bao gồm Phaeomoniella chlamydospora và Phaeoacremonium spp., và một số yếu tố môi trường khác nhau. Thường phát triển trong điều kiện ẩm ướt hoặc khi cây bị tổn thương.

Bệnh bạc lá (Bệnh đốm lá Isariopsis)	Gây ra các vết nhỏ màu nâu hoặc đen trên lá nho. Các vết này sau đó mở rộng và trở thành các vết lớn hơn, thường là hình tròn hoặc không đều.	Do nấm Isariopsis spp. gây ra. Phát triển và lan rộng trong điều kiện ẩm ướt và nhiệt độ ấm áp.

2.1.4. MÍT

Hình 2.4: Lá cây mít khỏe mạnh

Bảng 2.4: Một số loại sâu bệnh trên cây mít

Sâu bệnh	Triệu chứng	Nguyên nhân
Bệnh đốm lá tảo	Gây ra các vết nhỏ màu xanh hoặc nâu trên lá mít. Các vết này thường xuất hiện dưới dạng các đốm đều và có thể mở rộng trên toàn bộ bề mặt lá. Lá có thể bị héo và rụng.	Do sự phát triển quá mức của tảo xanh trên bề mặt lá, thường xuất hiện trong điều kiện ẩm ướt và nhiệt độ ấm áp.

Bệnh xơ đen	Gây ra các vết đen hoặc nâu trên bề mặt quả hoặc lá mít. Những vết này có thể xuất hiện như đốm nhỏ và sau đó mở rộng, phủ kín phần lớn bề mặt.	Do sự phát triển của nấm hoặc vi khuẩn, thường trong môi trường ẩm ướt và ấm.

2.1.5. Xoài

Hình 2.5: Lá cây xoài khỏe mạnh

Bảng 2.5: Một số loại sâu bệnh trên cây xoài

Sâu bệnh	Triệu chứng	Nguyên nhân
Bệnh thán thư	Gây ra các vết đốm màu đen, nâu hoặc đỏ trên quả, lá, thân cây hoặc cả nhánh non. Các vết này sau đó mở rộng và lan rộng trên bề mặt của cây xoài. Quả xoài bị thối, héo và có thể rụng.	Do nấm Colletotrichum spp. gây ra, phát triển mạnh trong điều kiện ẩm ướt và nhiệt độ ấm áp.
Bọ cắt lá	Bọ cắt lá gây ra những vết ăn trên cành và lá, có thể làm suy yếu cây xoài.	Bọ cắt lá gây hại bằng cách cắt và gặm các phần cây xoài, đặc biệt là cành non.

Bệnh thối loét do vi khuẩn	Gây ra các vết thối mục nước, có thể có màu xám hoặc nâu, xuất hiện trên quả, lá và thân cây. Các vết này thường bắt đầu nhỏ và sau đó mở rộng, phủ kín phần lớn bề mặt.	Do các loại vi khuẩn, thường là Xanthomonas spp. gây ra. Phát triển mạnh trong điều kiện ẩm ướt và thời tiết ấm áp.
Bệnh chết ngọn	Cây xoài bắt đầu chết từ các cành, lá hoặc ngọn, thường bắt đầu từ phía đỉnh của cây và lan dần xuống. Lá và cành có thể chuyển sang màu vàng, nâu hoặc đỏ trước khi héo rụng.	Có thể do nhiều nguyên nhân, bao gồm áp lực môi trường như thiếu nước, thất thoát dinh dưỡng, hoặc tổn thương từ côn trùng gây hại. Bệnh tật hoặc nấm gây hại cũng có thể góp phần vào gây ra chết ngọn.
Bệnh ghẻ	Các đốm hoặc khối u trên lá, cành và thậm chí trên quả xoài. Các khối u có thể xuất hiện như vùng phình to, có thể màu xanh, đỏ hoặc vàng, tùy thuộc vào giai đoạn phát triển của ghẻ.	Do ghẻ gây hại và làm tổ trên cây xoài. Côn trùng này gây tổn thương khi đẻ trứng và phát triển thành ấu trùng trên lá và cành.
Bệnh phấn trắng	Tạo ra một lớp phủ màu trắng bột nhẹ trên bề mặt của lá và cành, có thể bắt đầu từ phía dưới lá hoặc trên cả hai mặt của lá. Lá có thể bị héo, bị cong và dễ rụng.	Do sự phát triển của nấm làm tổ trên lá cây, thường xảy ra trong điều kiện ẩm ướt và khí hậu ấm áp.

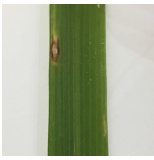
Bệnh bồ hóng	Tạo ra một lớp phủ màu đen, gần như than hoặc nấm mốc trên bề mặt của lá cây xoài. Lớp phủ này thường dày đặc và có thể che khuất bề mặt lá, làm giảm khả năng quang hợp.	Bồ hóng thường xuất hiện khi có một lượng lớn sâu hoặc các loài côn trùng tiết ra mật, phát triển nhựa cây hoặc thải ra phân trên lá. Nấm mốc sau đó phát triển trên chất dư thừa này, tạo ra lớp phủ màu đen.

2.1.6. Lúa

Hình 2.6: Lá cây lúa khỏe mạnh

Bảng 2.6: Một số loại sâu bệnh trên cây lúa

Sâu bệnh	Triệu chứng	Nguyên nhân
Bệnh bạc lá	Gây ra các vết cháy màu nâu trên lá, có thể mở rộng và có hình dạng không đều. Lá có thể bị héo và chết, tạo thành các vùng màu nâu hoặc vàng.	Do vi khuẩn gây hại, thường là Xanthomonas oryzae. Sự lây lan thường xảy ra qua nước, gió hoặc qua các cành cây bị tổn thương.

Bệnh đốm nâu	Gây ra các vết nhỏ màu nâu trên lá lúa, với các vùng màu nâu có thể phát triển thành các vết lớn hơn. Có thể xuất hiện các vết có hình dạng không đều, làm giảm diện tích mặt lá xanh.	Do nấm Bipolaris oryzae gây ra, thường phát triển trong điều kiện ẩm ướt và nhiệt độ ấm áp. Lây lan thông qua hạt giống, dư lượng nước trên lá và qua gió.
Bệnh đạo ôn (Bệnh cháy lá) 	Gây ra các vết sần sùi màu nâu, thường bắt đầu từ các vùng màu xanh lá và sau đó chuyển sang màu nâu đậm. Các vết thường có hình dạng tròn, và có thể phát triển thành các vùng lớn hơn, làm chết và rụng lá.	Do nấm Pyricularia oryzae gây ra, thường phát triển trong môi trường ẩm ướt và nhiệt độ ấm áp. Nấm có thể lây lan qua gió, nước và các vật chủ mang bệnh trên lúa.
Bệnh bỏng lá	Gây ra các vùng lá màu vàng hoặc nâu, thường bắt đầu từ phía mép lá và lan dần vào bên trong. Các vết màu vàng hoặc nâu có thể mở rộng và trở thành những vùng lớn hơn trên bề mặt lá lúa.	Do vi khuẩn Xanthomonas oryzae gây ra, thường phát triển trong môi trường ẩm ướt và nhiệt độ ấm áp. Lây lan qua nước và các phương tiện khác như công cụ trồng.
Bệnh gạch nâu	Gây ra các vết màu nâu nhỏ trên lá lúa, thường có hình dạng hẹp và dài. Các vết thường xuất hiện song song với các gân lá và có thể lan rộng trong điều kiện thích hợp.	Do nấm Bipolaris oryzae gây ra, phát triển trong điều kiện ẩm ướt và nhiệt độ ấm áp. Nấm này thường lan truyền qua hạt giống hoặc từ môi trường lúa bị nhiễm bệnh.

2.1.7. MÍA

Hình 2.7: Lá cây mía khỏe mạnh

Bảng 2.7: Một số loại sâu bệnh trên cây mía

Sâu bệnh	Triệu chứng	Nguyên nhân
Bệnh khảm lá	Gây ra các vùng màu sáng và tối xen kẽ trên lá mía. Lá mía có thể có các vết màu vàng, xanh sáng hoặc màu sẫm tùy thuộc vào loại vi-rút gây bệnh.	Do vi-rút gây bệnh tấn công và lây lan qua côn trùng, vật nuôi hoặc thông qua môi trường trồng trọt. Có thể lây nhiễm từ cây mía bị nhiễm hoặc từ các loại cây khác.
Bệnh thối đỏ	Gây ra sự thối đỏ ở cuống, thân và rễ của cây mía. Các vùng bị nhiễm bệnh thường có màu đỏ hoặc hồng, và khi phát triển có thể lan ra khắp cây.	Do nấm gây bệnh Colletotrichum falcatum hoặc Colletotrichum sublineolum. Nấm này phát triển trong điều kiện ẩm ướt và ấm áp, thường lây lan qua môi trường nước.
Bệnh gỉ sắt	Gây ra các vết sần sùi, có màu nâu hoặc đỏ trên lá mía. Các vết thường xuất hiện ở phía dưới lá và có thể lan rộng khi điều kiện thích hợp.	Do nấm Puccinia melanocephala gây ra, phát triển trong điều kiện ẩm ướt và nhiệt độ ấm áp. Nấm có thể lây lan qua gió, nước và các vật chủ mang bệnh trên mía.

Bệnh vàng lá	Lá cây mía chuyển sang màu vàng từ màu xanh bình thường. Có thể xuất hiện vùng vàng ở cả lá trẻ và lá già, thậm chí lan ra khắp cây mía.	Điều kiện môi trường không lý tưởng, bao gồm thiếu hụt dinh dưỡng hoặc nước, áp lực từ môi trường, nhưng cũng có thể do nhiễm vi khuẩn hoặc vi-rút. Lây nhiễm thông qua đất, nước, hoặc côn trùng.

2.1.8. TRÀ

Hình 2.8: Lá cây trà khỏe mạnh

Bảng 2.8: Một số loại sâu bệnh trên cây trà

Sâu bệnh	**Triệu chứng**	**Nguyên nhân**
Bệnh đốm tảo	Gây ra các vùng nhỏ màu xanh đậm hoặc đen trên lá trà. Có thể xuất hiện những vết nhỏ hoặc lớn, thường tập trung ở phía trên và dưới của lá.	Do vi khuẩn, nấm hoặc tảo phát triển trên bề mặt lá do độ ẩm và ánh sáng phù hợp. Môi trường ẩm ướt và nhiệt độ môi trường lý tưởng cho sự phát triển của các loại vi sinh vật này.

Bệnh bạc lá nâu	Gây ra sự chuyển màu của lá trà từ màu xanh sang màu nâu hoặc đỏ. Các vết nâu có thể xuất hiện ở phía trên và dưới của lá, dẫn đến sự suy yếu và rụng lá.	Thường do vi khuẩn hoặc nấm gây bệnh phát triển trong điều kiện ẩm ướt và ấm áp. Môi trường mưa nhiều, đất ẩm hoặc quá nhiều độ ẩm trong không khí tạo điều kiện lý tưởng cho sự phát triển của bệnh.
Bệnh bạc lá xám	Gây ra sự xuất hiện của các vùng màu xám hoặc xám nhạt trên lá trà. Các vết xám có thể lan rộng trên lá và có thể kèm theo sự suy yếu và rụng lá.	Thường do nấm gây bệnh phát triển trong điều kiện ẩm ướt và ấm áp. Môi trường ẩm, mưa nhiều và ánh sáng yếu tạo điều kiện lý tưởng cho sự phát triển của bệnh.
Bọ xít muỗi gây hại (Helopeltis)	Helopeltis gây ra các vết thương trên lá cây trà bằng cách hút nước mô từ lá, dẫn đến việc lá bị vàng, mất màu xanh tự nhiên, và thậm chí có thể rụng.	Loài côn trùng này hút chất lượng mô từ lá trà để nuôi sống và phát triển. Môi trường ẩm ướt và nhiệt độ cao thường làm tăng sự phát triển và hoạt động của Helopeltis.
Bệnh đốm đỏ	Xuất hiện các đốm màu đỏ hoặc nâu trên lá trà, thường là ở phía dưới lá. Các đốm này có thể là những vết nhỏ hoặc lớn, có thể lan rộng và phát triển thành vùng lớn hơn trên lá.	Thường do vi khuẩn, nấm hoặc ký sinh trùng phát triển trong môi trường ẩm ướt và ấm áp. Môi trường này tạo điều kiện lý tưởng cho sự phát triển của bệnh.

2.2. PHÂN LOẠI HÌNH ẢNH

2.2.1. PHÂN LOẠI HÌNH ẢNH LÀ GÌ?

Phân loại hình ảnh đề cập đến nhiệm vụ gắn một **label (nhãn)** hoặc **category (danh mục)** cho một hình ảnh đầu vào dựa trên nội dung của nó. Đây là một bài toán cơ bản trong Computer Vision nhằm mục đích tự động phân loại hình ảnh thành các **class (lớp)** hoặc category được xác định trước. Mục tiêu của phân loại hình ảnh là phát triển các mô hình hoặc giải thuật có thể nhận dạng và phân biệt chính xác giữa các đối tượng, cảnh vật,... khác nhau được mô tả trong ảnh.

2.2.2. CÁC PHƯƠNG PHÁP PHÂN LOẠI HÌNH ẢNH

Hai cách tiếp cận đáng chú ý nhất cho bài toán phân loại hình ảnh đó là **Supervised Learning (Học có giám sát)** và **Unsupervised Learning (Học không có giám sát)**:

- **Supervised Learning**:

  Supervised Learning dựa trên dữ liệu huấn luyện được gắn nhãn, trong đó mỗi hình ảnh được liên kết với một class hoặc category đã biết. Các nhãn đóng vai trò là thông tin cơ bản để huấn luyện mô hình phân loại.

  Trong giai đoạn huấn luyện, mô hình học cách ánh xạ các đặc trưng của hình ảnh đầu vào tới các nhãn của class tương ứng. Nó cố gắng tối ưu hóa các tham số để giảm thiểu sự khác biệt giữa nhãn dự đoán và nhãn thực tế.

  Các phương pháp phân loại hình ảnh supervised thường sử dụng nhiều loại giải thuật, gồm các giải thuật Machine Learning truyền thống (ví dụ: Support Vector Machines, Random Forests) và các kiến trúc Deep Learning (ví dụ: Convolutional Neural Networks).

  Mô hình được đánh giá trên một tập thử nghiệm riêng biệt, bao gồm các hình ảnh có nhãn đã biết không được sử dụng trong quá trình huấn luyện. Các số liệu như accuracy, precision, recall và F1-score thường được sử dụng để đánh giá hiệu suất của các mô hình phân loại hình ảnh supervised.

- **Unsupervised Learning**:

  Các phương pháp Unsupervised Learning nhằm mục đích khám phá các **pattern (mẫu)**, cấu trúc hoặc mối quan hệ trong dữ liệu mà không cần sử dụng các nhãn class tường minh.

  Trong phân loại hình ảnh unsupervised, giải thuật phân tích cấu trúc vốn có của dữ liệu và tự động nhóm các hình ảnh tương tự thành các cụm dựa trên những đặc điểm chung.

  Các giải thuật Unsupervised Learning phổ biến để phân loại hình ảnh bao gồm các kỹ thuật phân cụm như phân cụm K-means, phân cụm theo cấp bậc và Gaussian Mixture Models - GMM.

  Không giống như Supervised Learning, không có khái niệm định nghĩa trước về nhãn class trong phân loại ảnh unsupervised. Các cụm do giải thuật tạo ra có thể được phân tích hoặc gắn nhãn bổ sung dựa trên thông tin bổ sung.

Cả 2 phương pháp supervised và unsupervised đều có những ưu điểm và hạn chế riêng. Supervised Learning cung cấp kết quả phân loại chính xác khi có sẵn dữ liệu huấn luyện được gắn nhãn. Mặt khác, Unsupervised Learning có thể hữu ích trong các tình huống mà dữ liệu được gắn nhãn khan hiếm hoặc khó lấy, cung cấp thông tin chi tiết về cấu trúc dữ liệu và hỗ trợ trong các nhiệm vụ phân tích tiếp theo.

Trong thực tế, có thể sử dụng kết hợp các phương pháp supervised và unsupervised để tăng cường hỗ trợ các tác vụ phân loại hình ảnh. Ví dụ: huấn luyện trước unsupervised có thể được sử dụng để khởi tạo các **weight (trọng số)** của **Deep Neural Network (Mạng nơ-ron sâu)**, sau đó là điều chỉnh bằng cách sử dụng Supervised Learning với dữ liệu được gắn nhãn để cải thiện độ chính xác của phân loại.

2.2.3. CÁC BƯỚC PHÂN LOẠI HÌNH ẢNH

Bài toán phân loại hình ảnh thường bao gồm các bước như sau:

- **Thu thập dữ liệu**: Thu thập tập dữ liệu gồm các hình ảnh được gắn nhãn trong trường hợp Supervised Learning và các hình ảnh không được gắn nhãn trong trường hợp Unsupervised Learning. Ví dụ: ta có thể thu thập một tập dữ liệu gồm hàng nghìn hình ảnh về động vật, mỗi hình ảnh được gắn nhãn loài động vật tương ứng hoặc không có nhãn nào cả.

- **Tiền xử lý dữ liệu**: Đây là một bước quan trọng trong quy trình phân tích dữ liệu. Nó bao gồm các hoạt động như làm sạch dữ liệu, biến đổi dữ liệu, trích xuất thông tin hữu ích từ dữ liệu và cung cấp dữ liệu đó cho các phân tích tiếp theo. Trong trường hợp phân loại hình ảnh, nó có thể liên quan đến việc thay đổi kích thước hình ảnh, chuyển đổi hình ảnh sang **grayscale (thước xám)** và chuẩn hóa các giá trị pixel.

- **Trích xuất đặc trưng**: Liên quan đến việc trích xuất các thông tin liên quan từ tập dữ liệu. Thông tin được trích xuất được gọi là đặc trưng và nó thường được sử dụng làm cơ sở để phân loại, dự đoán hoặc phân cụm. Nó có thể liên quan đến các kỹ thuật như áp dụng các **convolution (tích chập)** cho hình ảnh hoặc **histograms of oriented gradient - HOGs (các biểu đồ của độ dốc định hướng)**.

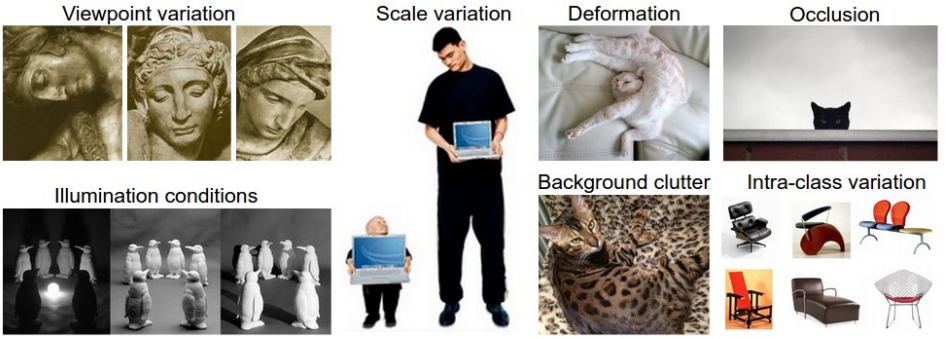
- **Huấn luyện mô hình**: Liên quan đến việc huấn luyện một giải thuật Machine Learning với một danh sách các mẫu (điểm dữ liệu) đã biết. Đó là một quá trình lặp đi lặp lại trong đó một giải thuật học từ một tập hợp dữ liệu được gắn nhãn và sau đó sử dụng kiến thức đó để dự đoán nhãn chính xác cho dữ liệu mới chưa từng thấy trước đây. Nó có thể liên quan đến việc sử dụng giải thuật Supervised Learning, chẳng hạn như Deep Neural Network hoặc Support Vector Machines.

- **Đánh giá mô hình**: Đây là một phần của quy trình Machine Learning để chúng ta đo lường mức độ hoạt động của mô hình. Nó đánh giá hiệu suất của mô hình trên một tập hợp các hình ảnh thử nghiệm không được sử dụng trong quá trình huấn luyện, có thể liên quan đến các chỉ số như accuracy, precision, recall,...

- **Triển khai**: Đây là giai đoạn cuối cùng của quy trình để mô hình được huấn luyện có thể được đưa vào sử dụng. Ở đó, mô hình sẽ được huấn luyện liên tục trên dữ liệu trực tiếp mới và liên tục chạy các dự đoán trên dữ liệu trong thế giới thực.

2.2.4. NHỮNG THÁCH THỨC ĐỐI VỚI BÀI TOÁN PHÂN LOẠI HÌNH ẢNH

Dưới đây là những thách thức mà chúng ta có thể gặp phải khi giải quyết bài toán phân loại hình ảnh:



Hình 2.9: Một số thách thức trong bài toán phân loại hình ảnh

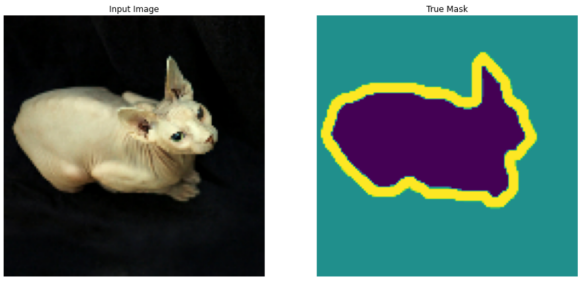
- **Viewpoint variation (Đa dạng về góc nhìn)**: Đối tượng cần phân loại khác nhau khi có những góc nhìn khác nhau.
- **Scale variation (Đa dạng về tỉ lệ/kích thước)**: Cùng một đối tượng cần phân loại tuy nhiên có thể có kích thước khác nhau.
- **Deformation (Biến dạng)**: Sự đa dạng hình ảnh của cùng một đối tượng khi bị biến đổi theo những điều kiện khác nhau.
- **Occlusion (Bị che khuất)**: Đối tượng cần phân loại bị che khuất.
- **Illumination conditions (Điều kiện chiếu sáng)**: Cùng một ảnh chụp một vật nhưng ở những điều kiện ánh sáng khác nhau có thể gây sai sót trong quá trình phân loại.
- **Background clutter (Ảnh hưởng bởi khung cảnh)**: Đối tượng cần phân loại bị nhầm lẫn với môi trường xung quanh.
- **Intra-class variation (Đa dạng về biến thể trong một nhãn)**: Nhãn cần phân loại có rất nhiều loại ví dụ mèo thì có mèo tam thể, mèo mướp,...

2.3. PHÂN ĐOẠN HÌNH ẢNH

2.3.1. PHÂN ĐOẠN HÌNH ẢNH LÀ GÌ?

Phân đoạn hình ảnh là một trong những lĩnh vực chính của thị giác máy tính, được hỗ trợ bởi một lượng lớn nghiên cứu liên quan đến xử lý ảnh và các giải thuật học. Phân đoạn hình ảnh đứng đằng sau các ứng dụng nổi bật như robotics, hình ảnh y tế, xe tự lái và phân tích video thông minh. Lĩnh vực này cũng được biết đến bởi lịch sử nghiên cứu, phát triển lâu dài, với các công trình đầu tiên ra mắt vào những năm 1970-1972.

Mục đích của phân đoạn hình ảnh là nhóm các vùng hoặc phân đoạn một hình ảnh theo các class tương ứng. Tác vụ này tương đương với việc nhóm các pixel. Bên cạnh phân loại, phân đoạn hình ảnh cũng yêu cầu khoanh vùng (xác định vị trí chính xác của một đối tượng bằng cách xác định ranh giới của chúng). Do đó, có thể coi đây là bài toán mở rộng của phân loại hình ảnh.



Hình 2.10: Input (bên trái) và Output (bên phải) của mô hình phân đoạn hình ảnh đối với bài toán một đối tượng. Mỗi một nhãn sẽ được thể hiện bởi một màu sắc khác nhau. Màu xám là nền, màu vàng là đường viền của ảnh và màu tím là nằm bên trong vật thể.

2.3.2. CÁC LOẠI TÁC VỤ PHÂN ĐOẠN HÌNH ẢNH

Phân đoạn hình ảnh có thể được chia thành 3 nhóm tác vụ dựa trên số lượng và loại thông tin mà chúng truyền tải.

SEMANTIC SEGMENTATION (PHÂN ĐOẠN NGỮ NGHĨA)

Semantic Segmentation đề cập đến việc phân loại các pixel trong một hình ảnh thành các class ngữ nghĩa. Các pixel thuộc về một lớp cụ thể được phân loại một cách đơn giản vào class đó mà không cần xem xét thông tin hoặc bối cảnh nào khác.

Tác vụ này không phù hợp khi có nhiều đối tượng được nhóm chặt chẽ trên cùng một class trong hình ảnh. Ví dụ: Hình ảnh đám đông trên đường phố sẽ có mô hình Semantic Segmentation dự đoán toàn bộ khu vực đám đông thuộc về class "người đi bộ", do đó cung cấp rất ít chi tiết hoặc thông tin chuyên sâu về hình ảnh.

INSTANCE SEGMENTATION (PHÂN ĐOẠN CÁ THỂ)

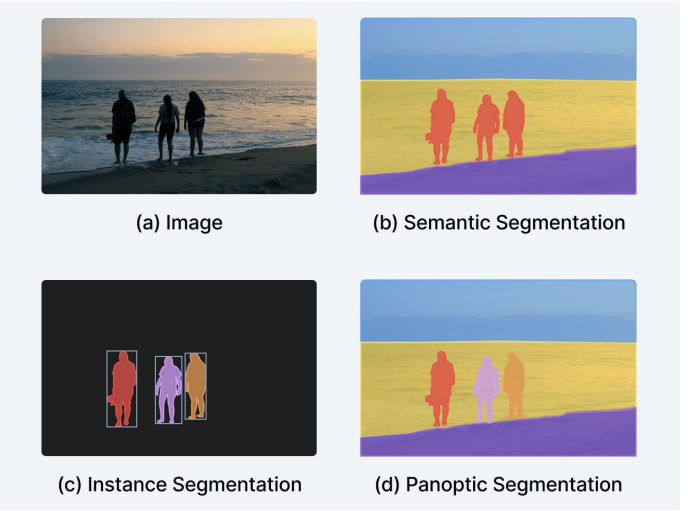
Các mô hình Instance Segmentation phân loại các pixel thành các danh mục trên cơ sở các đối tượng với nhãn cụ thể.

Ví dụ: Nếu một hình ảnh về đám đông được đưa vào mô hình Instance Segmentation, thì mô hình đó sẽ có thể tách biệt từng người khỏi đám đông cũng như các đối tượng xung quanh (lý tưởng nhất).

PANOPTIC SEGMENTATION (PHÂN ĐOẠN TOÀN CẢNH)

Panoptic Segmentation, tác vụ phân đoạn được phát triển gần đây nhất, là sự kết hợp giữa Semantic Segmentation và Instance Segmentation, trong đó ranh giới của mỗi đối tượng trong ảnh được tách biệt và danh tính của đối tượng được dự đoán.

Các giải thuật toán Panoptic Segmentation có khả năng ứng dụng quy mô lớn trong các tác vụ phổ biến như ô tô tự lái, khi một lượng lớn thông tin về môi trường xung quanh phải được ghi lại ngay lập tức với sự trợ giúp của một luồng hình ảnh.



Hình 2.11: Các loại tác vụ phân đoạn hình ảnh

3

NHỮNG CÔNG TRÌNH LIÊN QUAN

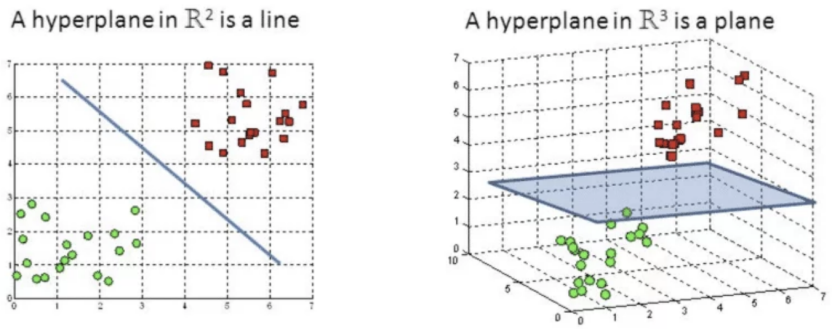
Nội dung của chương này xoay quanh việc tìm hiểu các kỹ thuật cũng như những tài liệu, công trình nghiên cứu đã ứng dụng các kỹ thuật đó để nhận diện sâu bệnh trên cây trồng với cốt lõi là bài toán phân loại hình ảnh, bao gồm 2 nhóm chính là nhóm các kỹ thuật truyền thống và nhóm các kỹ thuật dựa trên Deep Learning.

3.1. CÁC KỸ THUẬT TRUYỀN THỐNG

3.1.1. SVM

Support Vector Machine - SVM là một mô hình phân loại hoạt động bằng việc xây dựng một **hyperplane (siêu phẳng)** có $(n-1)$ chiều trong không gian n chiều của dữ liệu sao cho hyperplane này phân loại các lớp một cách tối ưu nhất. Nói cách khác, cho một tập dữ liệu có nhãn (Supervised Learning), giải thuật sẽ dựa trên dữ liệu học để xây dựng một hyperlane tối ưu được sử dụng để phân loại dữ liệu mới.

Một cách trực quan, trong một mặt phẳng 2 chiều thì hyperlane là một đường thẳng, trong một không gian 3 chiều thì hyperlane là một mặt phẳng.



Hình 3.1: Support Vector Machine trong không gian hai chiều và ba chiều

Để phân chia các class dữ liệu, rõ ràng là có rất nhiều hyperlane có thể làm được điều này. Mặc dù vậy, mục tiêu của chúng ta là tìm ra hyperlane có lề rộng nhất tức là có khoảng cách tới các điểm của các class là lớn nhất, thông qua đó SVM có thể được huấn luyện một cách hiệu quả.

Những nghiên cứu được thực hiện bằng cách sử dụng SVM để phân loại bệnh ở thực vật được tóm lược dưới đây:

Bảng 3.1: Những nghiên cứu đã được thực hiện bằng phương pháp sử dụng SVM

Tiêu đề	**Loại cây**	**Kết quả**
SVM Classifier Based Grape Leaf Disease Detection [1]	Nho	Hệ thống cung cấp accuracy trung bình là 88,89% đối với cả bệnh lá nho Downey và bệnh phấn trắng.
Support vector machine classifier based detection of fungal rust disease in Pea Plant (Pisa sativam) [2]	Đậu	Phương pháp đề xuất có thể chẩn đoán và phân tích bệnh thành công với accuracy 89,60%.
Diseases Detection of Cotton Leaf Spot using Image Processing and SVM Classifier [3]	Bông	Hệ thống đề xuất phát hiện bệnh trên cây bông với accuracy 98,46%.

Plant leaf disease identification using exponential spider monkey optimization [4]	Khoai tây và Táo	Accuracy 92,12% đã đạt được bằng phương pháp này.
Recognition and Detection of Tea Leaf's Diseases Using Support Vector Machine [5]	Trà	Hệ thống đạt được accuracy tổng thể là 93,33%.
Detection and Classification of Diseases of Banana Plant Using Local Binary Pattern and Support Vector Machine [6]	Chuối	Accuracy tối đa mà công trình đề xuất đạt được là 89,1% và 90,9%.

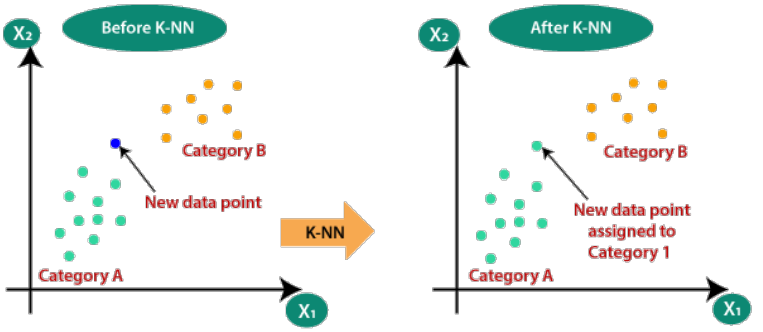
3.1.2. KNN

K-Nearest Neighbor - KNN là một trong những giải thuật Supervised Learning đơn giản nhất (mà hiệu quả trong một vài trường hợp) trong Machine Learning. Khi huấn luyện, giải thuật này không học một điều gì từ dữ liệu huấn luyện (đây cũng là lý do giải thuật này được xếp vào loại **Lazy Learning**), mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới. KNN có thể áp dụng được vào cả hai loại của bài toán Supervised Learning là phân loại và hồi quy.

Với KNN, trong bài toán phân loại, nhãn của một điểm dữ liệu mới được suy ra trực tiếp từ K điểm dữ liệu gần nhất trong tập huấn luyện, có thể được quyết định bằng **major voting (bầu chọn theo số phiếu)** giữa các điểm gần nhất, hoặc có thể đánh trọng số khác nhau cho mỗi điểm trong các điểm gần nhất đó rồi suy ra nhãn.

Một cách ngắn gọn, KNN là giải thuật đi tìm đầu ra của một điểm dữ liệu mới bằng cách chỉ dựa trên thông tin của K điểm dữ liệu trong tập huấn luyện gần nó nhất, không quan tâm đến việc có một vài điểm dữ liệu trong những điểm gần nhất này là nhiễu.

Kỹ thuật này được sử dụng rộng rãi trong các lĩnh vực như khai thác văn bản, dự báo xu hướng trên thị trường chứng khoán và cả trong nông nghiệp để phân loại các bệnh khác nhau ở thực vật, với những nghiên cứu sau:



Hình 3.2: Ví dụ minh họa cho giải thuật K-Nearest Neighbor

Bảng 3.2: Những nghiên cứu đã được thực hiện bằng phương pháp sử dụng KNN

Tiêu đề	Loại cây	Kết quả
Disease Detection and Severity Estimation in Cotton Plant from Unconstrained Images [7]	Bông	Accuracy 82,5% đã đạt được khi sử dụng hệ thống này.
Deep learning for image-based cassava disease detection [8]	Sắn	Accuracy thu được là 73% bằng cách áp dụng KNN và bằng cách sử dụng SVM, accuracy đạt được là 91%.
Recognition of Diseases in Paddy Leaves Using kNN Classifier [9]	Lúa	Với phương pháp này, các tác giả đã đạt được accuracy tổng thể là 76,59%.
A Color and Texture Based Approach for the Detection and Classification of Plant Leaf Disease Using KNN Classifier [10]	Nhiều loại cây	Hiệu suất phân loại của KNN trên bệnh lá cây cho accuracy 96,76%. Cách tiếp cận này cung cấp kết quả tốt hơn so với một số phương pháp hiện có.
A remote sensing technique for detecting laurel wilt disease in avocado in presence of other biotic and abiotic stresses [11]	Bơ	Trong mọi trường hợp, phương pháp MLP thu được giá trị phân loại cao hơn KNN và đôi khi đạt tới 98%.

3.2. CÁC KỸ THUẬT DỰA TRÊN DEEP LEARNING

3.2.1. CNN

Convolutional Neural Network - CNN là một loại Deep Feed Forward Neural Network có khả năng xử lý dữ liệu đa chiều. Mục đích của CNN là đưa hình ảnh về một dạng dễ xử lý hơn mà không ảnh hưởng đến các đặc trưng cần thiết để có thể đưa ra dự đoán chính xác.

Khi xây dựng một mô hình mới, CNN bao gồm một số lớp. Một CNN chứa các lớp **convolutional**, **batch normalization**, **activation** và **pooling**, được sử dụng để khám phá các đặc trưng của ảnh. Nó cũng chứa các lớp **dense** và **dropout**, được sử dụng để phân loại hình ảnh dựa trên các đặc trưng được tạo ra. Sau đó CNN được huấn luyện bằng cách xác định các trọng số tối ưu của Neural Network để giảm sai số giữa đầu ra kỳ vọng và đầu vào thực tế của tập dữ liệu được sử dụng. Các **loss function** và **optimization function (hàm tối ưu hóa)** đóng một vai trò quan trọng trong quá trình **backpropagation (lan truyền ngược)**, đây là một cách tiếp cận điển hình để huấn luyện các Neural Network. Người ta đã chứng minh rằng CNN vượt trội so với các phương pháp trích xuất đặc trưng truyền thống trong việc xác định các bệnh thực vật [12]. Một kiến trúc CNN điển hình được mô tả như sau:

- **Các lớp convolutional**: Các lớp này chứa một tập hợp các **filter** hoặc **kernel** phổ biến được sử dụng để trích xuất các đặc trưng từ ảnh bằng cách di chuyển kernel qua ảnh và tìm kiếm các đặc trưng để tạo ra các **feature map (bản đồ đặc trưng)**. Trong quá trình huấn luyện mạng sử dụng kỹ thuật backpropagation, các filter được thiết lập từ các ví dụ và được coi là trọng số. Các lớp này nhằm mục đích giảm số lượng tham số nhưng vẫn duy trì hiệu quả của mạng.

- **Các lớp batch normalization**: Các lớp này tăng tốc độ huấn luyện Deep Neural Network, vì chúng chia nhỏ dữ liệu thành các batch nhỏ trong quá trình chuẩn hóa để giảm độ phức tạp của việc huấn luyện hệ thống. Mỗi đặc trưng z được tính lại theo phương trình:

$$z_{norm}^{(i)} = \frac{z^{(i)} - \mu_B}{\sqrt{\sigma_B^2 - \varepsilon}}$$

  trong đó ε là một số dương rất nhỏ để tránh chia cho 0 và μ_B là giá trị trung bình của giá trị trung bình của mini-batch và σ_B^2 biểu thị phương sai của mini-batch, vì giá trị của z trong lớp đầu tiên được

tính theo phương trình:

$$z = \omega x + b$$

trong đó x là giá trị của các đặc trưng đầu vào, b là bias.
Để chia tỷ lệ và dịch chuyển các đầu vào đã chuẩn hóa γ và β được thêm vào và học bằng cách sử dụng tham số mạng thông qua phương trình:

$$\tilde{z}^{(i)} = \gamma z_{norm}^{(i)} + \beta$$

- **Các lớp activation**: **Activation function (Hàm kích hoạt)** được sử dụng để tăng tốc các phép toán số học mà không trải qua các vấn đề về vanishing hoặc exploiting. **Rectified Linear Unit - ReLU** được sử dụng làm activation function cơ bản, đây là một trong những activation function được sử dụng rộng rãi nhất. Nó được đặc trưng bởi sự dễ dàng và đơn giản của phương trình và nó biến đổi $\tilde{z}$ trước đó bằng mô hình toán học:

$$a = \max(0; \tilde{z})$$

Việc sử dụng activation function cuối cùng khác ở chỗ nó được sử dụng làm đầu ra của lớp dense cuối cùng. Nó hiển thị xác suất dự đoán là 0 hoặc 1 trong trường hợp phân loại nhị phân hoặc dưới dạng phần trăm thập phân trong trường hợp phân loại đa phân. **SoftMax function (Hàm SoftMax)** được sử dụng để gán xác suất thập phân cho mỗi lớp. Nó được tính từ đơn vị đầu ra i theo phương trình:

$$a^{(i)} = \frac{e^{z^i}}{\sum_i^m e^{z^i}} \text{for} i = 1, 2, 3, ..., m$$

trong đó $z^(i)$ là đầu ra của chiều thứ i, $a^(i)$ là xác suất liên quan đến class thứ i và m là số chiều tương ứng với số class. Một mẫu được gán cho class có xác suất cao nhất khi dự đoán bằng phương pháp được mô tả dưới đây:

$$\hat{y}_i = \max_{i \in [1,m]} a^{(i)})$$

- **Các lớp pooling**: Giảm kích thước của các feature map, vì nó chọn các đặc trưng quan trọng nhất từ các đặc trưng được tạo bởi các lớp convolutional. Có 2 loại lớp pooling: **average pooling** và **max pooling**. Average pooling lấy giá trị trung bình của mỗi feature map, trong khi đó max pooling lấy giá trị lớn nhất.

- **Các lớp fully connected**: Bao gồm các **neuron (nơ-ron)** và tạo thành các lớp cuối cùng của Neural Network, nơi nó lấy đầu vào từ đầu ra cuối cùng của các lớp pooling hoặc các lớp convolutional và biến nó thành một vector duy nhất bằng một lớp **flatten**. Sau đó, nó đưa ra dự đoán bằng cách sử dụng trọng số và xác suất cuối cùng được đưa ra bởi một lớp dense. Số lượng cell trong lớp dense cuối cùng tương ứng với số lượng class.

- **Các lớp dropout**: Các lớp này là một kỹ thuật chính quy hóa chủ yếu được sử dụng trong quá trình huấn luyện mạng để tránh bị overfitting bằng cách xóa một phần của các neuron đến và các kết nối của chúng.

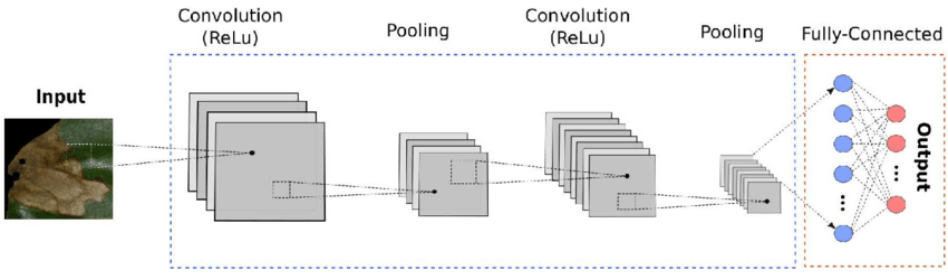
- **Một loss function**: Tên gọi khác của **cost function (hàm chi phí)**. Nó được tính toán bằng **cross-entropy** để tìm độ chênh lệch giữa đầu ra thực tế y và đầu ra kỳ vọng $\hat{y}$. Cross-entropy trong bài toán phân loại nhiều class được tính theo phương trình:

$$L_{cross-entroy(\hat{y},y)} = -\sum_{j=1}^{K} y_j \log(\hat{y}_j)$$

  trong đó K là số class.

- **Một optimization function**: Nó được sử dụng để làm giảm giá trị của loss function.

Có nhiều kiến trúc sẵn có khác nhau dành cho CNN như AlexNet, GoogLeNet, VGG,... Sự phát triển của nó đã thu hút rất nhiều sự quan tâm của các nhà nghiên cứu trong các lĩnh vực khác nhau của khoa học máy tính. Trong nông nghiệp, nó đã được sử dụng để phân loại các bệnh ở thực vật.



Hình 3.3: Phân loại bệnh ở thực vật sử dụng Convolutional Neural Network

Như thể hiện trong hình 3.3, mô hình CNN bao gồm các lớp như đã được đề cập. Để phân loại bệnh ở thực vật một cách chính xác, các hình ảnh được cung cấp làm đầu vào. Lớp convolutional được sử dụng để trích xuất các đặc trưng từ hình ảnh. Lớp pooling tính toán các giá trị đặc trưng từ các đặc trưng được trích xuất. Tùy thuộc vào độ phức tạp của hình ảnh, số lớp convolutional và pooling có thể được tăng thêm để thu được nhiều chi tiết hơn. Lớp fully connected sử dụng đầu ra của các lớp trước đó và biến chúng thành một vector duy nhất có thể được sử dụng làm đầu vào cho lớp tiếp theo. Lớp đầu ra cuối cùng phân loại bệnh cây trồng.

Những nghiên cứu sau đây đã được thực hiện trong việc phát hiện bệnh ở thực vật bằng cách sử dụng phương pháp CNN:

Bảng 3.3: Những nghiên cứu đã được thực hiện bằng phương pháp sử dụng CNN

Tiêu đề	**Loại cây**	**Kết quả**
Using Deep Learning for Image-Based Plant Disease Detection [13]	Việt quất, Cherry, Táo, Ngô, Nho, Cam, Đào, Khoai tây, Cà chua, Dâu tây	Accuracy tổng thể thu được trong khoảng từ 85,53% đến 99,34%. Người ta đã quan sát thấy rằng GoogLeNet hoạt động tốt hơn AlexNet với accuracy là 98,21%.
Identification of rice diseases using deep convolutional neural networks [14]	Lúa	Accuracy tối đa đạt được bằng cách sử dụng stochastic pooling và kích thước filter 16x16 lần lượt là 95,48% và 93,29%.
Plant Disease Detection Using CNNs and GANs as an Augmentative Approach [15]	Nhiều loại cây	Trong đó, mô hình Inception v3 đạt được accuracy 88,6% và MobileNets đạt được accuracy 92%.
Deep learning models for plant disease detection and diagnosis [16]	Táo, Việt quất, Ngô, Bắp cải, Sắn, Cherry, Đào, Dâu, Cà chua	Mô hình đề xuất có accuracy dao động từ 97,26% - 99,47%. Accuracy cao nhất đạt được nhờ CNN Visual Geometry Group - VGG.

Seasonal crops disease prediction and classification using deep convolutional encoder network [17]	Khoai tây, Cà chua, Ngô	Công trình này đạt accuracy 97,50% với kích thước filter 2x2 của convolution trong 100 epoch, trong khi accuracy 100% tương ứng với kích thước filter 3x3.
Depthwise separable convolution architectures for plant disease classification [18]	Những loại cây phân biệt bao gồm Cà chua	Accuracy đạt được bởi Modified MobileNet, Reduced MobileNet và MobileNet lần lượt là 97,65%, 98,34% và 98,65%.
Identification of plant leaf diseases using a nine-layer deep convolutional neural network [19]	Táo, Cà chua, Khoai tây, Việt quất, Nho, Cherry, Đào	Phương pháp đạt được accuracy 96,46%, tốt hơn so với các mô hình khác như SVM, Decision tree, Hồi quy logistic và KNN.
Attention embedded residual CNN for disease detection in tomato leaves [20]	Cà chua	Kết quả thí nghiệm của công trình đề xuất đã có thể phát hiện bệnh với accuracy 98%.

3.2.2. YOLO

YOLO LÀ GÌ?

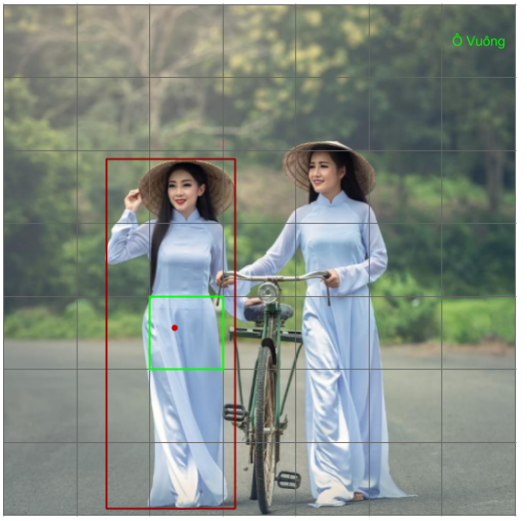
YOLO là một mô hình mạng CNN cho việc phát hiện, nhận dạng, phân loại đối tượng. YOLO được tạo ra từ việc kết hợp giữa các lớp convolutional và các lớp fully connected.Trong đóp các lớp convolutional sẽ trích xuất ra các đặc trưng của ảnh, còn các lớp fully connected sẽ dự đoán ra xác suất đó và tọa độ của đối tượng.

YOLO chỉ sử dụng một Neural Network duy nhất để dự đoán trực tiếp các **bounding box (hộp giới hạn)** và các **class probability (xác suất lớp)** từ toàn bộ bức ảnh bằng một lần đánh giá duy nhất. Vì vậy, ta chỉ cần nhìn một lần duy nhất vào bức ảnh để có thể dự đoán vật thể trong ảnh và vị trí của nó. Đó cũng chính sự giải thích cho cái tên YOLO - **You Only Look Once (bạn chỉ cần nhìn một lần duy nhất)**. Chính vì lý do đó mà tốc độ xử lý của YOLO cực kì nhanh. Toàn bộ quá trình phát hiện đối tượng là một mạng duy nhất nên nó có thể được tối ưu **end-to-end (đầu cuối)**.

CÁCH YOLO HOẠT ĐỘNG

Ảnh được chia thành ma trận ô vuông 7x7, mỗi ô vuông bao gồm một tập các thông tin mà mô hình phải dự đoán.

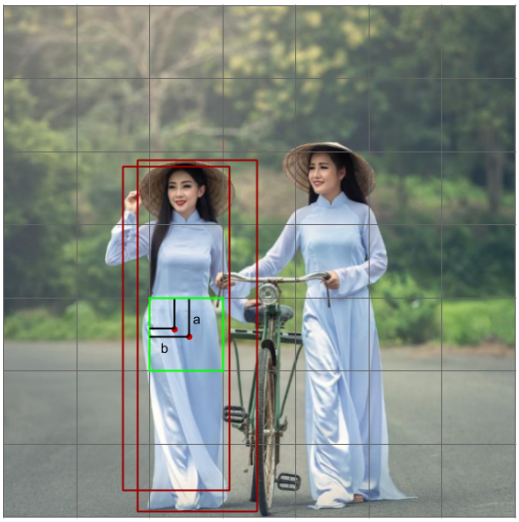
- Đối tượng duy nhất mà ô vuông đó chứa. Tâm của đối tượng cần xác định nằm trong ô vuông nào thì ô vuông đó chứa đối tượng đó. Ví dụ tâm của cô gái nằm trong ô vuông màu xanh, do đó mô hình phải dự đoán được nhãn của ô vuông đó là cô gái. Lưu ý, cho dù phần ảnh cô gái có nằm ở ô vuông khác mà tâm không thuộc ô vuông đó thì vẫn không tính là chứa cô gái, ngoài ra, nếu có nhiều tâm nằm trong một ô vuông thì chúng ta vẫn chỉ gán một nhãn cho ô vuông đó thôi. Chính ràng buộc mỗi ô vuông chỉ chứa một đối tượng là nhược điểm của mô hình này. Nó làm cho ta không thể phát hiện những đối tượng có tâm nằm cùng một ô vuông. Tuy nhiên chúng ta có thể tăng kích thước **grid (lưới)** từ 7x7 lên kích thước lớn hơn để có thể phát hiện được nhiều đối tượng hơn. Ngoài ra, kích thước của ảnh đầu vào phải là bội số của kích thước grid.



Hình 3.4: Ảnh được chia thành ma trận ô vuông 7x7

- Mỗi ô vuông chịu trách nhiệm dự đoán 2 boundary box của đối tượng. Mỗi boundary box dự đoán có chứa đối tượng hay không và thông tin vị trí của boundary box gồm tâm boundary box của đối tượng và chiều dài, rộng của boundary box đó. Ví dụ ô vuông màu xanh cần dự đoán 2 boundary box chứa cô gái như hình minh họa

ở dưới. Một điều cần lưu ý, chúng ta không dự đoán giá trị pixel mà cần phải chuẩn hóa kích thước ảnh về đoạn từ `[0-1]` và dự đoán độ lệch của tâm đối tượng đến box chứa đối tượng đó. Ví dụ, chúng ta thay vì dự đoán vị trí pixel của điểm màu đỏ, thì cần dự đoán độ lệch `a`, `b` trong ô vuông chứa tâm đối tượng.



Hình 3.5: Mỗi ô vuông chịu trách nhiệm dự đoán 2 boundary box của đối tượng

Tổng hợp lại, với mỗi ô vuông chúng ta cần dự đoán các thông tin sau:

- Ô vuông có chứa đối tượng nào hay không?
- Dự đoán độ lệch 2 box chứa đối tượng so với ô vuông hiện tại.
- Lớp của đối tượng đó.

Như vậy với mỗi ô vuông chúng ta cần dự đoán một vector có (`nbox + 4 * nbox + nclass`) chiều. Ví dụ, chúng ta cần dự đoán 2 box, và 3 lớp đối với mỗi ô vuông thì chúng sẽ có một ma trận 3 chiều `7x7x13` chứa toàn bộ thông tin cần thiết.

object 1?	object 2?	offset x1	offset y1	width 1	height 1	offset x2	offset y2	width 2	height 2	0	0	1

Hình 3.6: Vector dự đoán cho mỗi ô vuông

ƯU ĐIỂM CỦA YOLO SO VỚI CÁC MẠNG CNN TRUYỀN THỐNG

- **Có tốc độ xử lý cực kì nhanh**: Vì YOLO coi phát hiện đối tượng là một bài toán hồi quy nên ta không cần dùng đến một cơ chế **pipeline (đường ống)** phức tạp. Ta chỉ đơn giản chạy Neural Network trên một bức ảnh mới tại thời điểm kiểm thử để dự đoán. Điều này giúp ta có thể xử lý video **real-time (thời gian thực)** với độ trễ nhỏ. YOLO đạt được độ chính xác cao hơn nhiều lần so với các hệ thống real-time khác.

- Lập luận một cách toàn cục về bức ảnh khi đưa ra dự đoán. Không giống như các phương pháp khác, YOLO nhìn vào toàn bộ bức ảnh trong thời gian huấn luyện và kiểm thử, vì thế nên nó ngầm hiểu thông tin ngữ cảnh về các class cũng như là ngoại hình của chúng. Vì thế các **background error (lỗi nền)** của YOLO nhỏ hơn đáng kể so với các phương pháp khác, ví dụ như Fast R-CNN.

- Có khả năng học được các biểu diễn có tính tổng quát hóa của vật thể. Khi huấn luyện YOLO với những bức ảnh thực và kiểm thử bằng những tấm ảnh vẽ, YOLO có kết quả vượt trội so với các phương pháp phát hiện đối tượng hàng đầu khác như DPM hay R-CNN. Vì YOLO có tính tổng quát hóa rất cao nên nó ít khi thất bại khi được đưa vào các lĩnh vực mới hay sử dụng các input bất thường.

Bảng 3.4: Những nghiên cứu đã được thực hiện bằng phương pháp sử dụng YOLO

Tiêu đề	**Loại cây**	**Kết quả**
Dragon fruit detection in natural orchard environment by integrating lightweight network and attention mechanism [21]	Thanh long	Với mAP đạt 97.4%, mô hình thể hiện hiệu suất cải thiện so với YOLOv5s gốc. Mô hình có khả năng nhận diện trong các điều kiện ánh sáng khác nhau, đạt AP cao trong nhiều trường hợp.
Tea leaf disease detection and identification based on YOLOv7 (YOLO-T) [22]	Trà	Accuracy phân loại tổng thể là 97,30%, recall và precision lần lượt là 96,4% và 96,7%, vượt trội hơn các mô hình gần đây nhất được báo cáo.

Yolo-Papaya: A Papaya Fruit Disease Detector and Classifier Using CNNs and Convolutional Block Attention Modules [23]	Đu đủ	Mô hình được đề xuất cho thấy sự cải thiện đáng kể so với các mô hình được thử nghiệm khác, đạt mAP trung bình là 86,2% ngay cả ở những class có độ biến thiên trong nội bộ cao.
Tomato Diseases and Pests Detection Based on Improved Yolo V3 Convolutional Neural Network [24]	Cà chua	Kết quả thực nghiệm cho thấy accuracy của giải thuật là 92,39% và thời gian phát hiện chỉ 20,39ms, không chỉ có độ chính xác cao mà còn đáp ứng yêu cầu theo thời gian thực
Rice leaf disease detection based on bidirectional feature attention pyramid network with YOLO v5 model [25]	Lúa	Mô hình được đề xuất có accuracy là 94,87% với efficiency là 91,89% so với các phương pháp hiện có, chi phí tính toán giảm đáng kể.

4

THỬ NGHIỆM

Chương này trước tiên giới thiệu về một số mô hình và tập dữ liệu đã được sử dụng trong các thử nghiệm, sau phân tích và đánh giá kết quả của những thử nghiệm đó.

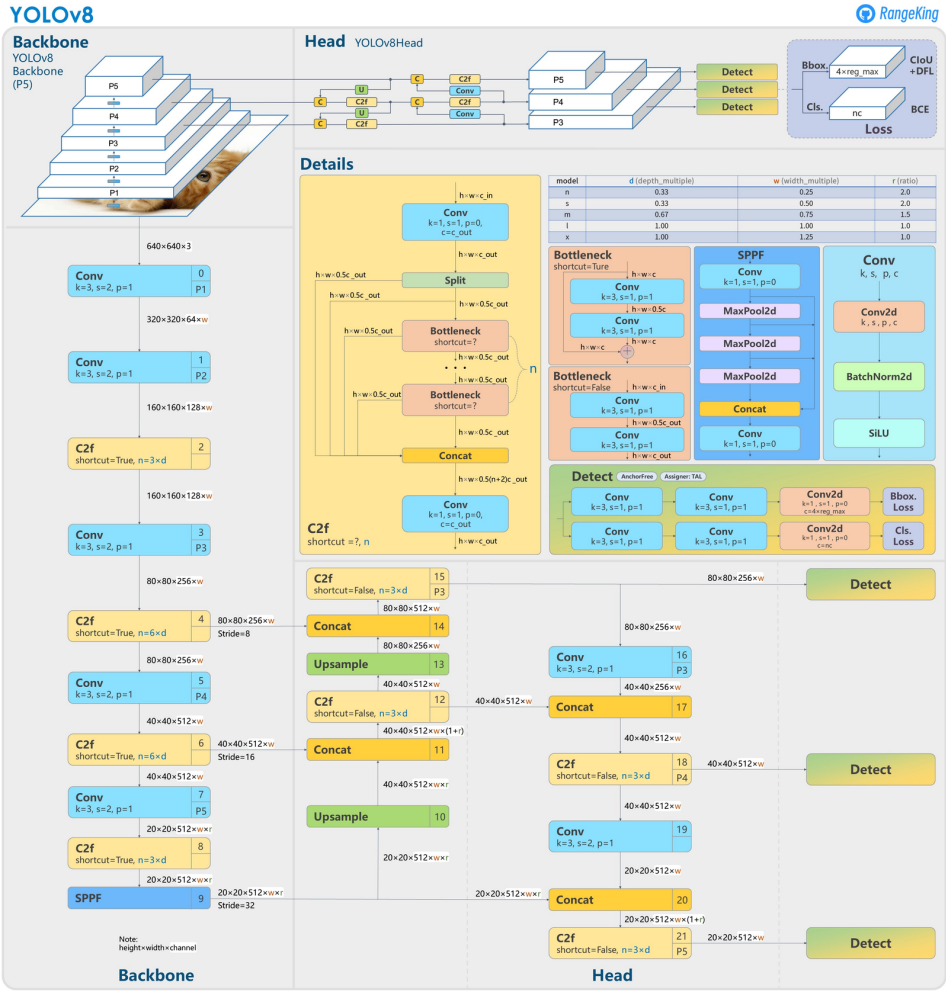
4.1. MỘT SỐ MÔ HÌNH ĐƯỢC SỬ DỤNG

4.1.1. YOLOv8

YOLOv8 là mô hình YOLO tiên tiến nhất có thể được sử dụng cho các tác vụ phát hiện đối tượng, phân loại và phân đoạn hình ảnh. YOLOv8 được phát triển bởi Ultralytics [26], người cũng đã tạo ra mô hình **YOLOv5** có tầm ảnh hưởng và định hướng. YOLOv8 bao gồm nhiều thay đổi và cải tiến về kiến trúc và kinh nghiệm của nhà phát triển so với YOLOv5.

Loạt mô hình YOLO đã trở nên nổi tiếng trong thế giới Computer Vision. Sự nổi tiếng của YOLO là do độ chính xác đáng kể của nó trong khi vẫn duy trì kích thước mô hình nhỏ. Các mô hình YOLO có thể được huấn luyện trên một GPU duy nhất, giúp nhiều nhà phát triển có thể tiếp cận mô hình này. Những người thực hành Machine Learning có thể triển khai nó với chi phí thấp trên thiết bị phần cứng hoặc trên cloud.

YOLOv8 có tỷ lệ chính xác cao được đo bằng COCO và Roboflow 100 [27]. YOLOv8 đi kèm với rất nhiều tính năng thuận tiện cho nhà phát triển, từ CLI dễ sử dụng đến gói Python có cấu trúc tốt. Ảnh dưới đây sẽ giúp ta hình dung chi tiết hơn về cấu trúc mạng YOLOv8:



Hình 4.1: Kiến trúc mạng YOLOv8

4.2. CÁC TẬP DỮ LIỆU ĐƯỢC SỬ DỤNG

4.2.1. GIỚI THIỆU TẬP DỮ LIỆU PLANTVILLAGE

PlantVillage được tạo ra để mang đến các giải pháp hiệu quả nhằm phát hiện 38 bệnh thực vật khác nhau, chứa 54305 hình ảnh lá đơn khỏe mạnh hoặc bị nhiễm bệnh, được tuyển chọn chuyên nghiệp từ 14 loại cây trồng. Các nhà phát triển đã sử dụng 6 kỹ thuật tăng cường dữ liệu để tăng kích thước tập dữ liệu với nhiều điều kiện khác nhau. Các kỹ thuật đó bao gồm: **scaling (thu phóng)**, **rotation (xoay)**, **noise injection (thêm nhiễu)**, **gamma correction (hiệu chỉnh gamma)**, **flipping (lật)**, và **PCA color augmentation (tăng cường màu PCA)**.

PlantVillage là một trong những tập dữ liệu phổ biến nhất, đặc biệt có nhiều ứng dụng trong lĩnh vực chẩn đoán bệnh ở cây trồng, và đã được sử dụng trong nhiều tài liệu nghiên cứu [28].

Hình 4.2: Một vài mẫu trong tập dữ liệu PlantVillage

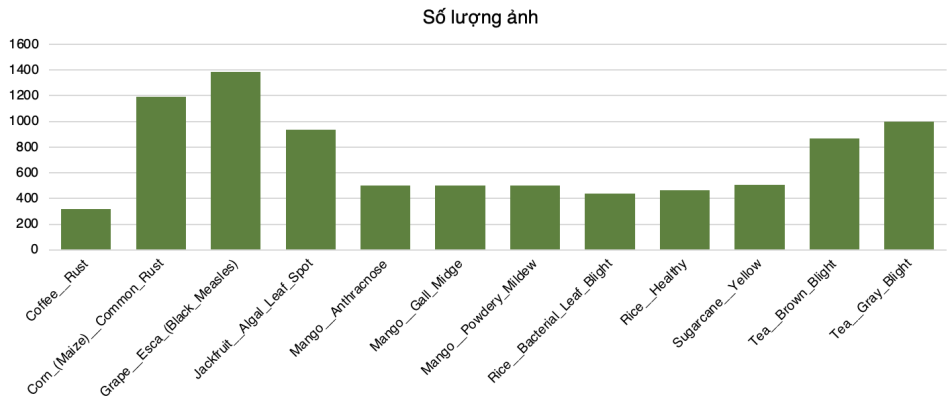
4.2.2. TẬP DỮ LIỆU CHÍNH CHO TÁC VỤ PHÂN LOẠI SÂU BỆNH TRÊN NHIỀU LOẠI CÂY

Tập dữ liệu dùng để huấn luyện mô hình phân loại sâu bệnh chứa tổng cộng 26776 hình ảnh lá cây khỏe mạnh hoặc bị sâu bệnh hại thuộc 40 class từ 8 loại cây khác nhau đã được đề cập trong phần 2.1. Trong đó có một số loại cây và loại sâu bệnh tương ứng được lấy từ tập PlantVillage, số còn lại được tổng hợp từ nhiều nguồn dữ liệu trong các bài báo, bài nghiên cứu có liên quan đã được công bố trên Internet.

Với mỗi class, số lượng ảnh được chia theo tỉ lệ các tập train : validate : test là 8 : 1 : 1. Thông tin chi tiết về số lượng ảnh của mỗi class như trong bảng sau:

Bảng 4.1: Số lượng ảnh của các class trong tập dữ liệu huấn luyện mô hình phân loại

Class	Số lượng
Coffee__Healthy	400
Coffee__Miner	460
Coffee__Phoma	484
Coffee__Rust	320
Corn_(Maize)__Cercospora_Leaf_Spot_(Gray_Leaf_Spot)	513
Corn_(Maize)__Common_Rust	1192
Corn_(Maize)__Healthy	1162
Corn_(Maize)__Northern_Leaf_Blight	985
Grape__Black_Rot	1180
Grape__Esca_(Black_Measles)	1383
Grape__Healthy	423
Grape__Leaf_Blight_(Isariopsis_Leaf_Spot)	1076
Jackfruit__Algal_Leaf_Spot	933
Jackfruit__Black_Spot	667
Jackfruit__Healthy	500
Mango__Anthracnose	500
Mango__Bacterial_Canker	500
Mango__Cutting_Weevil	500
Mango__Die_Back	500
Mango__Gall_Midge	500
Mango__Healthy	500
Mango__Powdery_Mildew	500
Mango__Sooty_Mould	500
Rice__Bacterial_Leaf_Blight	438
Rice__Brown_Spot	466
Rice__Healthy	464
Rice__Leaf_Blast	454
Rice__Leaf_Scald	448
Rice__Narrow_Brown_Spot	440
Sugarcane__Healthy	522
Sugarcane__Mosaic	462
Sugarcane__Red_Rot	518
Sugarcane__Rust	514
Sugarcane__Yellow	505
Tea__Algal_Spot	1000
Tea__Brown_Blight	867
Tea__Gray_Blight	1000
Tea__Healthy	1000
Tea__Helopeltis	1000
Tea__Red_Spot	1000



Hình 4.3: Phân bố dữ liệu ở một số class trong tập dữ liệu chính

Phân bố dữ liệu trong tập dữ liệu này cho thấy đây là một tập dữ liệu mất cân bằng, thể hiện ở việc số lượng ảnh không đồng đều giữa các class. Cụ thể, class có số lượng ảnh nhiều nhất (`Grape__Esca_(Black_Measles)`) có đến 1383 ảnh, trong khi đó class có số lượng ảnh ít nhất (`Coffee__Rust`) chỉ có 320 ảnh. Sự mất cân bằng có thể ảnh hưởng không tốt đến kết quả phân loại, tuy nhiên vẫn còn phụ thuộc vào rất nhiều yếu tố khác cần được thử nghiệm.

4.2.3. TẬP DỮ LIỆU KIỂM TRA OVERFITTING CHO TÁC VỤ PHÂN LOẠI SÂU BỆNH TRÊN NHIỀU LOẠI CÂY

Tập dữ liệu dùng để kiểm tra overfitting của mô hình phân loại sâu bệnh chứa tổng cộng 200 hình ảnh lá cây khỏe mạnh hoặc bị sâu bệnh hại thuộc 40 class từ 8 loại cây khác nhau đã được đề cập trong phần 2.1. Khác với ảnh trong tập dữ liệu chính (chỉ bao gồm lá đơn trong ảnh), ảnh trong tập dữ liệu này mang tính thách thức hơn cho mô hình với nhiều góc chụp, mức độ ánh sáng, phông nền và số lượng lá khác nhau.

Số lượng ảnh trong tập dữ liệu này được chia đều cho tất cả các class, cụ thể là 5 ảnh cho mỗi class.

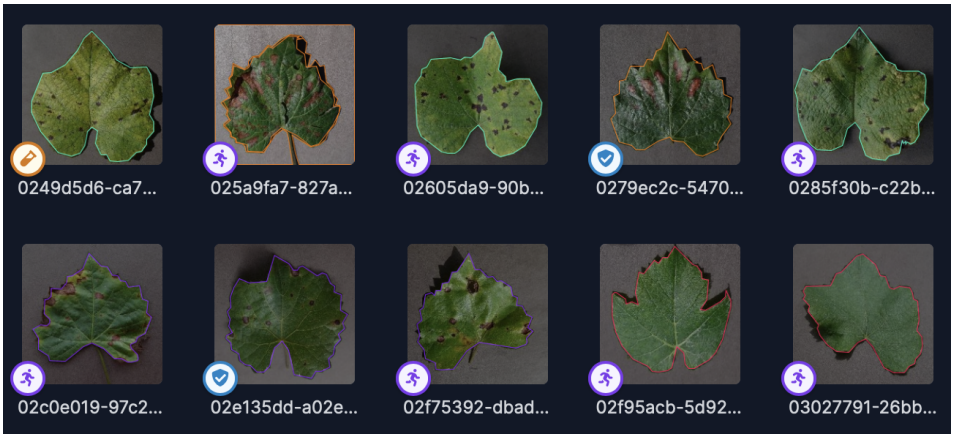
Hình 4.4: Một vài mẫu trong tập dữ liệu dùng để kiểm tra overfitting của mô hình phân loại sâu bệnh

4.2.4. TẬP DỮ LIỆU CHO TÁC VỤ PHÂN ĐOẠN LÁ NHO

Tập dữ liệu được sử dụng để huấn luyện mô hình phân đoạn lá nho là tập dữ liệu "**Grape Segmentation**"có sẵn trên nền tảng của Roboflow [29]. Mục tiêu của tập dữ liệu này là cung cấp một bộ sưu tập toàn diện các hình ảnh có chú thích tập trung cụ thể vào lá nho, cho phép huấn luyện và đánh giá các mô hình phân đoạn lá nho.

Tập dữ liệu được chia theo tỉ lệ các tập train : validate : test là 7 : 2 : 1, bao gồm 2304 ảnh chụp các khía cạnh khác nhau của lá nho, bao gồm các góc chụp, điều kiện ánh sáng, bối cảnh và tình trạng khác nhau. Mỗi hình ảnh trong tập dữ liệu được chú thích tỉ mỉ bằng các **accurate boundary (đường biên chính xác)**, mô tả chính xác vị trí và hình dạng của lá nho trong ảnh.

Với những chú thích này, các nhà nghiên cứu và những người thực hành có thể tận dụng tập dữ liệu này để phát triển các mô hình phân đoạn hình ảnh chính xác và mạnh mẽ, có thể xác định và định vị lá nho trong những hình ảnh chưa được nhìn thấy, một bước quan trọng trong việc tự động phát hiện sâu bệnh ảnh hưởng đến cây nho.



Hình 4.5: Một vài mẫu trong tập dữ liệu dùng để phân đoạn lá nho từ Roboflow

4.2.5. TẬP DỮ LIỆU CHO TÁC VỤ PHÂN LOẠI SÂU BỆNH TRÊN CÂY NHO

Tập dữ liệu được sử dụng để huấn luyện mô hình phân loại sâu bệnh trên cây nho là tập con bao gồm 4 class về lá nho của tập `segmented` được trích từ PlantVillage. Tập dữ liệu này bao gồm 4062 ảnh lá nho đã được phân đoạn, có chứa các thông tin về vị trí và đặc điểm chi tiết của lá nho, với nhiều tình trạng khác nhau, từ lá khỏe mạnh, cho đến những lá bị sâu bệnh gây hại như bệnh thối đen, bệnh Esca (sởi đen), bệnh bạc lá (đốm lá Isariopsis). Với mỗi class, số lượng ảnh được chia theo tỉ lệ các tập train : validate : test là 8 : 1 : 1.

Mục tiêu của tập dữ liệu này là cung cấp một bộ sưu tập các hình ảnh lá nho với nhiều tình trạng sức khỏe khác nhau, cho phép huấn luyện và đánh giá các mô hình phân loại sâu bệnh trên cây nho, với đầu vào là ảnh lá nho đã được phân đoạn.

Hình 4.6: Một vài mẫu trong tập dữ liệu dùng để phân loại sâu bệnh trên cây nho

4.3. MỘT SỐ THỬ NGHIỆM ĐÃ ĐƯỢC THỰC HIỆN

4.3.1. LỰA CHỌN MÔ HÌNH PHÂN LOẠI SÂU BỆNH

Thử nghiệm này đã được thực hiện để lựa chọn mô hình phân loại sâu bệnh, cũng chính là thành phần quan trọng nhất của ứng dụng, bằng cách so sánh hiệu suất của 2 mô hình là YOLOv5 Classification và YOLOv8 Classification trên tập dữ liệu chính, với kết quả như sau:

Bảng 4.2: Kết quả phân loại sâu bệnh trên 8 loại cây với YOLOv5 và YOLOv8

Mô hình	Precision	Recall	F1-Score
YOLOv5 Classification	96.56%	95.89%	96.22%
YOLOv8 Classification	98.40%	98.30%	98.38%

Kết quả cho thấy YOLOv8 đạt được các giá trị về độ chính xác cao hơn so với YOLOv5, cho thấy hiệu quả phân loại tốt hơn trên tập dữ liệu này. Do đó, YOLOv8 Classification sẽ được chọn làm mô hình phân loại sâu bệnh cho ứng dụng.

4.3.2. KIỂM TRA OVERFITTING CỦA MÔ HÌNH PHÂN LOẠI SÂU BỆNH

Thử nghiệm này được thực hiện trên cả tập dữ liệu chính và tập dữ liệu kiểm tra overfitting, với kết quả như sau:

Bảng 4.3: Kết quả phân loại sâu bệnh trên 2 tập dữ liệu

Tập dữ liệu	Precision	Recall	F1-Score
Tập dữ liệu chính	98.40%	98.30%	98.38%
Tập dữ liệu kiểm tra overfitting	96.18%	94.50%	94.25%

Két quả trên tập dữ liệu mới cho thấy mô hình phân loại sâu bệnh vẫn duy trì độ chính xác ở mức cao, chứng tỏ được khả năng tổng quát hóa của mô hình và do đó chứng minh được nó không bị overfitting.

4.3.3. LỰA CHỌN NGƯỠNG TIN CẬY CHO ỨNG DỤNG

Trên thực tế, để có thể mang đến cho người dùng một ứng dụng với độ chính xác cao, cần phải thiết lập một ngưỡng tin cậy cho các dự đoán. Thử nghiệm này được thực hiện với nhiều ngưỡng khác nhau trên cả tập dữ liệu chính và tập dữ liệu kiểm tra overfitting, với kết quả như sau:

Bảng 4.4: Tỉ lệ dự đoán đúng của mô hình phân loại sâu bệnh với các ngưỡng khác nhau

Ngưỡng tin cậy	Tỉ lệ dự đoán đúng	
	Tập dữ liệu chính	Tập dữ liệu kiểm tra overfitting
0.75	99.02%	93.87%
0.80	99.13%	94.27%
0.85	99.28%	93.88%
0.90	99.39%	94.74%
0.95	99.42%	96.64%

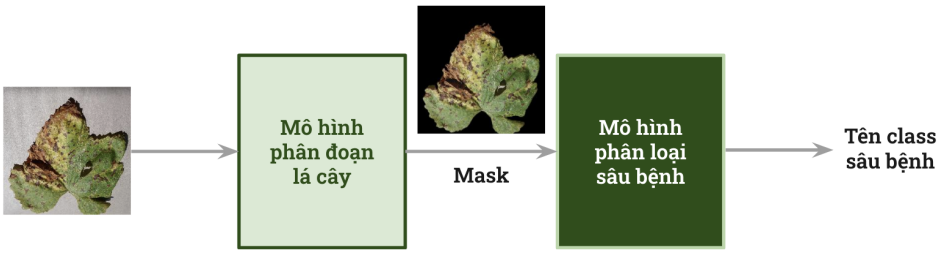
Thử nghiệm cho thấy ngưỡng tin cậy 0.95 cho ra kết quả tốt nhất nên sẽ được chọn làm ngưỡng tin cậy cho ứng dụng.

4.3.4. PHÂN LOẠI SÂU BỆNH KẾT HỢP TIỀN XỬ LÝ PHÂN ĐOẠN LÁ CÂY

Kết quả đánh giá mô hình ở thử nghiệm trước trên tập dữ liệu test cho thấy mô hình có hiệu suất rất cao. Tuy nhiên, hiệu suất trên tập dữ liệu test không nhất thiết phản ánh hoàn toàn hiệu suất của mô hình khi triển khai trong môi trường thực tế.

Dữ liệu trong thực tế có thể có nhiều biến thể, điều kiện ánh sáng, độ ẩm, góc chụp, và nhiều yếu tố môi trường khác. Điều này có thể ảnh hưởng đến khả năng của mô hình khi phân loại. Đôi khi, mô hình có thể gặp khó khăn trong việc nhận diện chính xác khi đối mặt với dữ liệu mới hoặc điều kiện môi trường khác nhau.

Do đó, việc kiểm tra và đánh giá hiệu suất của mô hình trên dữ liệu thực tế, trong môi trường thực tế là cần thiết để xác định được độ chính xác và độ tin cậy của mô hình khi được triển khai thực tế. Vì vậy, thử nghiệm tiền xử lý phân đoạn lá cây trước khi đưa vào mô hình phân loại sâu bệnh đã được tiến hành.



Hình 4.7: Kiến trúc mô hình thử nghiệm áp dụng tiền xử lý phân đoạn lá cây

Thử nghiệm được thực hiện với 2 mô hình:

- **Mô hình YOLOv8 Classification**: Tập dữ liệu huấn luyện là tập gồm 4 class về sâu bệnh trên cây nho đã được đề cập ở mục 4.2.5, mô hình được sử dụng cho tác vụ phân loại sâu bệnh trên cây nho, với đầu vào là ảnh lá nho đã được phân đoạn và đầu ra là tên class sâu bệnh tương ứng.
- **Mô hình YOLOv8 Segmentation**: Tập dữ liệu huấn luyện là tập lá nho đã được đề cập ở ở mục 4.2.4, mô hình được sử dụng cho tác vụ phân đoạn lá nho, với đầu vào là ảnh lá nho bất kỳ và đầu ra là ảnh lá nho đã được phân đoạn để đưa qua mô hình phân loại sâu bệnh (mask).

Dưới đây là kết quả so sánh hiệu suất của việc phân loại sâu bệnh trên cây nho trong 2 trường hợp: có và không có áp dụng tiền xử lý phân đoạn lá nho:

Bảng 4.5: Hiệu suất của mô hình phân loại sâu bệnh trên cây nho trong 2 trường hợp

Mô hình	**Precision**	**Recall**	**F1-Score**
YOLOv8 Classification	98.59%	98.19%	98.38%
YOLOv8 Classification + Segmentation	99.79%	99.82%	99.80%

Kết quả của cả 2 trường hợp đều cho thấy độ chính xác ở mức rất cao. Trong trường hợp có áp dụng phân đoạn, kết quả có cải thiện hơn so với trường hợp không áp dụng phân đoạn. Mặc dù sự cải thiện không lớn, nhưng việc áp dụng tiền xử lý phân đoạn lá nho đã giúp mô hình phân loại sâu bệnh tập trung vào các phần quan trọng của lá, có thể giúp tăng độ chính xác trong việc phân loại các loại sâu bệnh trên cây nho.

Điều này cho thấy rằng việc áp dụng phân đoạn trước quá trình phân loại có thể cải thiện hiệu suất của mô hình phân loại, có thể được xem xét để áp dụng với lượng dữ liệu lớn và phức tạp hơn.

5

PHÁT TRIỂN ỨNG DỤNG

Nội dung chương này liên quan đến quá trình phát triển ứng dụng, từ bước phân tích yêu cầu, tìm hiểu các công nghệ, ngôn ngữ, framework,... cần thiết cho đến việc thu thập dữ liệu, thiết kế giao diện và tổ chức luồng hoạt động để xây dựng ứng dụng hoàn chỉnh.

5.1. PHÂN TÍCH YÊU CẦU

5.1.1. ĐỐI TƯỢNG NGƯỜI DÙNG

- **Nông dân, người làm vườn**: Họ có thể sử dụng ứng dụng này để theo dõi tình trạng cây trồng, phát hiện sớm các vấn đề trong khu vườn, nông trại, cánh đồng,... của họ và thực hiện các hành động thích hợp để thúc đẩy cây trồng khỏe mạnh hơn và năng suất tốt hơn giúp bảo vệ sản lượng.

- **Chính phủ, cơ quan khuyến nông**: Họ có thể sử dụng ứng dụng này trong các chuyến đi thực địa. Ứng dụng hỗ trợ họ đưa ra lời khuyên và khuyến nghị kịp thời cho nông dân, cải thiện các biện pháp thực hành nông nghiệp và giải quyết các vấn đề sức khỏe cây trồng một cách hiệu quả.

- **Các chuyên gia tư vấn nông nghiệp**: Họ có thể sử dụng ứng dụng này để hỗ trợ khách hàng của họ trong việc chẩn đoán các vấn đề cây trồng, đưa ra hướng dẫn chuyên môn và các quyết định dựa trên dữ liệu liên quan đến sức khỏe và quản lý cây trồng.

- **Các nhà nghiên cứu về nông nghiệp**: Họ có thể sử dụng ứng dụng này trong việc thu thập và phân tích dữ liệu. Ứng dụng hỗ trợ thu thập thông tin chuyên sâu về mức độ phổ biến của các loại bệnh cây trồng và hỗ trợ trong giáo dục bằng cách chứng minh các ứng dụng thực tế của trí tuệ nhân tạo và thị giác máy tính trong nông nghiệp.

- **Công ty công nghệ về nông nghiệp**: Họ có thể tích hợp ứng dụng này vào các giải pháp của họ và tận dụng khả năng của mình để nâng cao chất lượng sản phẩm và dịch vụ cung cấp các giải pháp toàn diện hơn cho nông dân.

5.1.2. PHẠM VI VÀ LOẠI TRỪ

PHẠM VI

- Ứng dụng sử dụng ngôn ngữ là tiếng Việt.
- Ứng dụng nhận dữ liệu đầu vào là ảnh lá cây (được người dùng tải lên).
- Ứng dụng cho phép phân tích một hoặc nhiều hình ảnh được người dùng tải lên. Người dùng nhận được phản hồi ngay lập tức.
- Ứng dụng có thể nhận diện được 8 loại cây và dự đoán một số loại bệnh tương ứng đã được đề cập ở mục 2.1.

LOẠI TRỪ

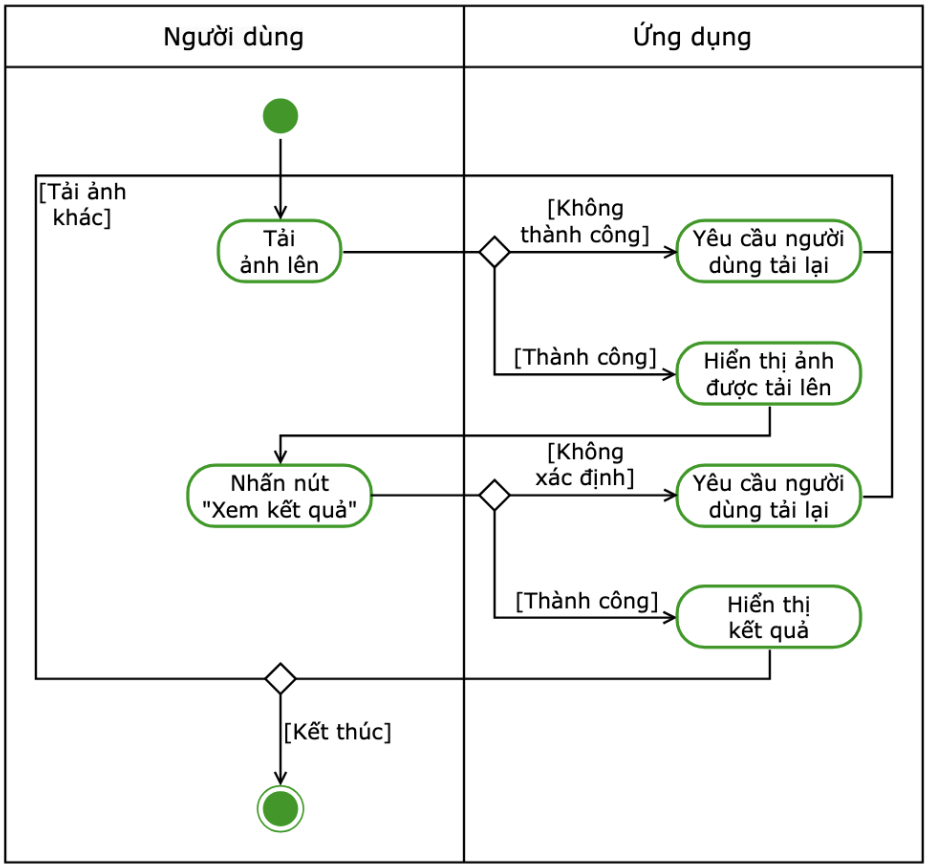
- Ứng dụng không yêu cầu người dùng đăng ký tài khoản và sử dụng tài khoản để đăng nhập vào ứng dụng.
- Ứng dụng không lưu thông tin người dùng.
- Ứng dụng không lưu lịch sử sử dụng của người dùng.
- Ứng dụng không nhận đầu vào là dữ liệu với những định dạng khác như văn bản, âm thanh, video,...
- Ứng dụng không nhận diện loại cây và chẩn đoán tình trạng sức khỏe của cây dựa trên những bộ phận khác ngoài lá cây như thân, quả, hoa,...
- Ứng dụng không áp dụng được trên những loại cây, loại sâu bệnh không có trong mục 2.1.

5.1.3. YÊU CẦU

YÊU CẦU CHỨC NĂNG

Yêu cầu chức năng của ứng dụng được thể hiện qua use case scenario và activity diagram tương ứng với mỗi chức năng:

- Chức năng dự đoán loại cây và tình trạng của cây từ một bức ảnh lá cây được người dùng tải lên:

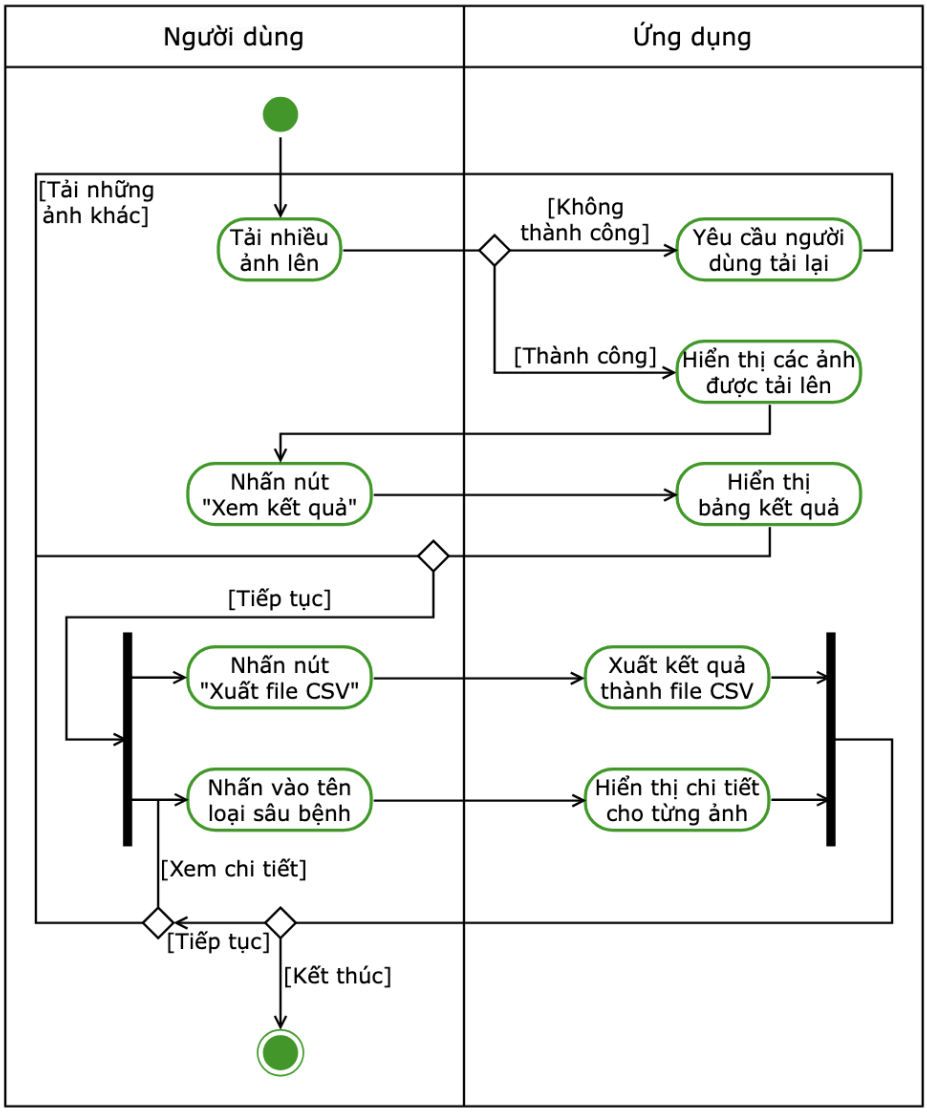


Hình 5.1: Activity diagram cho chức năng dự đoán trên một ảnh

Bảng 5.1: Use case scenario cho chức năng dự đoán trên một ảnh

Use Case Name	Dự đoán trên một ảnh
Brief Description	Từ ảnh lá cây người dùng tải lên, ứng dụng cho biết tên loại cây và tình trạng sức khỏe của cây cùng với độ tin cậy. Trong trường hợp cây bị sâu bệnh gây hại, ứng dụng cung cấp thông tin về nguyên nhân, triệu chứng cũng như phương pháp phòng trừ loại sâu bệnh đó.
Actor	Người dùng
Condition	Người dùng đã ấn nút chọn chức năng phân tích 1 ảnh ở trang chủ.
Basic Flow	1. Người dùng ấn nút "Tải ảnh lên"và chọn ảnh lá cây cần phân tích để tải lên ứng dụng. 2. Ứng dụng hiển thị ảnh được người dùng tải lên. 3. Người dùng ấn nút "Xem kết quả". 4. Ứng dụng hiển thị kết quả loại cây nhận diện được, tình trạng của cây cùng với độ tin cậ và nguyên nhân, triệu chứng cũng như phương pháp phòng trừ loại sâu bệnh (nếu có).
Alternative Flow	Tác vụ có thể kết thúc ở bước 4 nếu người dùng không có nhu cầu xem thông tin chi tiết của loại sâu bệnh ứng với từng ảnh và tải file csv về máy.
Exceptional Flow	Ở bước 1, nếu người dùng tải file lên không thành công (do file không đúng định dạng hình ảnh), ứng dụng hiển thị thông báo yêu cầu người dùng chọn file đúng định dạng và người dùng thực hiện lại bước 1. Ở bước 3, nếu ứng dụng không xác định được loại cây hoặc tình trạng của cây, ứng dụng hiển thị thông báo yêu cầu người dùng chọn ảnh khác và người dùng thực hiện lại bước 1.

- Chức năng dự đoán loại cây và tình trạng của cây từ nhiều bức ảnh lá cây được người dùng tải lên:



Hình 5.2: Activity diagram cho chức năng dự đoán trên nhiều ảnh

Bảng 5.2: Use case scenario cho chức năng dự đoán trên nhiều ảnh

Use Case Name	Dự đoán trên nhiều ảnh
Brief Description	Từ những ảnh lá cây người dùng tải lên, ứng dụng hiển thị 1 bảng tóm tắt thông tin cho từng ảnh về tên loại cây và tình trạng sức khỏe của cây cùng với độ tin cậy. Người dùng có thể xem thông tin chi tiết của loại sâu bệnh ứng với từng ảnh và có thể xuất bảng kết quả thành file CSV để tải về máy.
Actor	Người dùng
Condition	Người dùng đã ấn nút chọn chức năng phân tích nhiều ảnh ở trang chủ.
Basic Flow	1. Người dùng ấn nút "Tải ảnh lên"và chọn những ảnh lá cây cần phân tích để tải lên ứng dụng. 2. Ứng dụng hiển thị những ảnh được người dùng tải lên. 3. Người dùng ấn nút "Xem kết quả". 4. Ứng dụng hiển thị kết bảng kết quả thông tin cho từng ảnh về tên loại cây và tình trạng sức khỏe của cây cùng với độ tin cậy. 5. Người dùng có thể xem thông tin chi tiết của loại sâu bệnh ứng với từng ảnh bằng cách ấn vào tên loại sâu bệnh và/hoặc xuất bảng kết quả thành file CSV để tải về máy.
Alternative Flow	Tác vụ có thể kết thúc ở bước 4 nếu người dùng không có nhu cầu xem thông tin chi tiết của loại sâu bệnh ứng với từng ảnh và tải file CSV về máy.
Exceptional Flow	Ở bước 1, nếu người dùng tải các file lên không thành công (do có ít nhất 1 file không đúng định dạng hình ảnh), ứng dụng hiển thị thông báo yêu cầu người dùng chọn các file đúng định dạng và người dùng thực hiện lại bước 1.

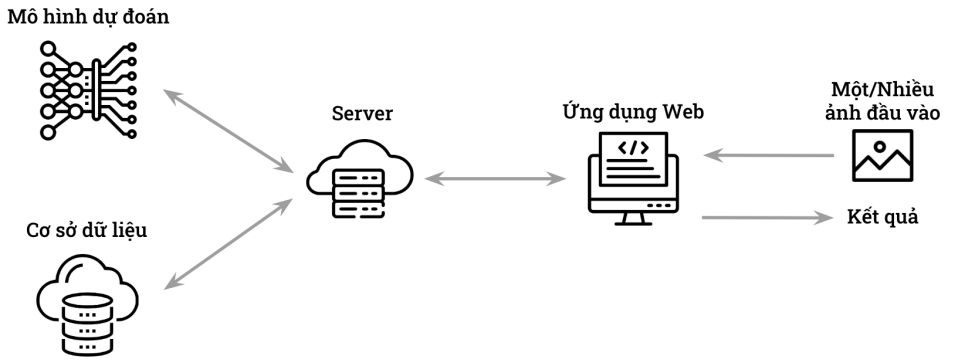
YÊU CẦU PHI CHỨC NĂNG

- Ứng dụng được thiết kế là ứng dụng Web có thể sử dụng được trên nhiều thiết bị như điện thoại di động, laptop, máy tính bảng,...
- Ảnh được người dùng tải lên phải được chụp trong điều kiện ánh sáng vừa phải, không quá mờ, lá cây cần phân tích không bị che khuất.
- Ứng dụng thực hiện các tác vụ phản hồi yêu cầu của người dùng trong thời gian không quá 3 giây.
- Ứng dụng thực hiện thực hiện các chức năng nhận diện loại cây và chẩn đoán tình trạng sức khỏe của cây với độ tin cậy tối thiểu là 95%.

5.2. KIẾN TRÚC HỆ THỐNG

Hệ thống được xây dựng với 4 thành phần chính:

- **Front-end**: Đây là giao diện người dùng (UI/UX), chính là trang Web để người dùng có thể tương tác với hệ thống, cung cấp trải nghiệm thân thiện và trực quan, cho phép người dùng thực hiện các chức năng và làm việc với dữ liệu của ứng dụng.
- **Back-end**: Đây là nơi xử lý logic của ứng dụng, quản lý dữ liệu và thực hiện các chức năng, bao gồm các API, **server (máy chủ)**, nơi các yêu cầu từ phía Front-end được xử lý, và các thành phần quản lý các tác vụ như xác thực người dùng, quản lý phiên làm việc và điều phối dữ liệu.
- **Cơ sở dữ liệu**: Đây là nơi lưu trữ và quản lý dữ liệu của hệ thống. Chức năng của nó là lưu trữ và cung cấp dữ liệu cho Back-end.
- **Mô hình dự đoán**: Đây là thành phần chịu trách nhiệm xử lý việc phân loại sâu bệnh từ ảnh lá cây, với cốt lõi là mô hình phân loại để dự đoán tình trạng của cây trồng dựa vào ảnh được cung cấp từ Front-end, ngoài ra còn có thể có thêm các mô hình giúp hỗ trợ cải thiện hiệu suất phân loại sâu bệnh, chẳng hạn như mô hình phân đoạn lá cây.



Hình 5.3: Kiến trúc tổng quát của hệ thống

5.3. CÔNG NGHỆ

5

5.3.1. FRONT-END: VUE.JS

VUE.JS LÀ GÌ?

Vue.js là một framework Javascript tiên tiến trong xây dựng giao diện người dùng, không giống như các framework khác, Vue được xây dựng từ những dòng code cơ bản nhất nhằm tối ưu tốc độ. Thư viện của Vue chỉ tập trung vào lớp hiển thị, rất đơn giản để tiếp cận và dễ dàng tích hợp vào các hệ thống khác. Vue cũng có khả năng cung cấp các **Single Page Application - SPA (Ứng dụng Web đơn trang)** cho phép kết hợp với nhiều các công cụ hiện đại, như Laravel chẳng hạn.

Vue.js có thể được sử dụng để xây dựng giao diện người dùng giống như React (sử dụng bởi Facebook), Angular (được hậu thuẫn bởi Google), Ember,... Tuy nhiên, Vue.js có tốc độ **render (tạo trang)** rất nhanh và chiếm khá ít bộ nhớ. Chúng ta có thể xem bảng benchmark các framework Javascript nổi tiếng nhất hiện nay, Vue có một thứ hạng không tồi chút nào.

Vue.js mới chỉ ra mắt năm 2015, nhưng Vue.js đã sớm khẳng định mình và sớm trở thành người thay thế Angular và React, đây cũng chính là lý do Laravel giới thiệu Vue.js trong thiết lập mặc định.

LÝ DO SỬ DỤNG VUE.JS

- **Kích thước nhỏ**: Tệp zip được tải xuống của framework này chỉ nặng 18KB. Điều này khiến nó không chỉ cài đặt nhanh mà còn tác động tích cực đến SEO và UX của bạn.

- **Kết xuất và hiệu suất DOM ảo**: Mô hình đối tượng tài liệu (DOM) là thứ bạn có thể gặp phải khi kết xuất các trang web. DOM đại diện cho một trang HTML với các kiểu, thành phần và nội dung dưới dạng cấu trúc cây của các đối tượng (nút). Các đối tượng cây DOM lưu trữ dưới dạng cây và được tạo bởi trình duyệt khi tải trang.
  Khi người dùng tương tác với trang, các đối tượng sẽ thay đổi trạng thái của chúng, do đó trình duyệt sẽ phải cập nhật thông tin và hiển thị trên màn hình. Tuy nhiên, việc cập nhật toàn bộ DOM rất phức tạp. Ưu tiên tốc độ load, Vue.js sử dụng DOM ảo. Hãy coi đây là một bản sao của DOM gốc giúp tìm ra những phần tử cần cập nhật mà không cần kết xuất lại toàn bộ cây nút. Cách tiếp cận này giúp hiển thị trang khá nhanh và cải thiện hiệu suất ứng dụng.

- **Hệ thống phản ứng và các tùy chọn ràng buộc dữ liệu**: Liên kết dữ liệu là kết nối giữa mô hình dữ liệu (nguồn dữ liệu) và mẫu DOM hoặc HTML của chế độ xem. Liên kết dữ liệu một chiều cho phép thông tin truyền theo một hướng, từ mô hình sang chế độ xem hoặc ngược lại. Trong trường hợp đầu tiên, các thay đổi đối với nguồn sẽ tự động cập nhật DOM, nhưng nó không hoạt động ngược lại vì DOM có quyền truy cập chỉ đọc vào mô hình.
  Liên kết dữ liệu hai chiều cho phép trao đổi dữ liệu giữa mô hình và chế độ xem theo cả hai hướng. Nói cách khác, mô hình cũng lắng nghe các sự kiện trên DOM và bất kỳ cập nhật nào ở một bên sẽ phản ánh ngay lập tức ở bên kia. Cách tiếp cận này loại bỏ mã soạn sẵn và đơn giản hóa việc phát triển ứng dụng. Tuy nhiên, việc khó Debug và dễ xảy ra lỗi khiến luồng hai chiều không phù hợp cho các dự án lớn.

- **Tính thuận tiện**: Thông thường, các loại công cụ khác sẽ mất thời gian cho quá nhiều bước xây dựng. Khi sử dụng Vue, người dùng gần như có thể đi thẳng vào vấn đề cần thực hiện mà không mất quá nhiều thời gian cho các bước xây dựng ứng dụng cơ bản.
  Điều này là vì Vue.js không cần đến các build tool phức tạp để hỗ trợ cho việc tạo ra một ứng dụng. Nó mang đến sự thuận tiện tuyệt vời cho người dùng. Bạn chỉ việc khai báo script là có thể bắt đầu phát triển phần mềm qua Vue.

- **Tạo cấu trúc project và tài liệu đa dạng**: Thông qua Vue, người dùng hoàn toàn có thể tạo ra được cấu trúc project rất nhanh chóng nhờ có command line interface. Bên cạnh đó, việc tìm kiếm tài liệu về Vue.js cũng vô cùng đơn giản nên bạn rất dễ dàng để nắm rõ và thành thạo với công cụ này.

- **Hệ sinh thái**: Hiện nay, Vue đang có hệ sinh thái được đánh giá rất cao và thực sự vững chắc, nó có thể cung cấp được add-ons điều này giúp ích khá nhiều cho việc lập trình ứng dụng frontend tiêu chuẩn. Vue.js có thể là: Vue-dev-tools, vue-test-utils, vue-router, vue-cli,...

- **Đáp ứng đầy đủ tính năng**: Core Vue đang sở hữu đầy đủ các tính năng đáp ứng được nhu cầu người dùng. Điều này là vì nó tập trung vào việc render giao diện phù hợp với các tương tác chính của Vue.js và người sử dụng.
  Công cụ này sẽ được trang bị tất tần tật các tính năng hữu ích và cần thiết cho quá trình thiết kế giao diện hay xây dựng các kiến trúc phần mềm cụ thể. Nó lược bỏ hoàn toàn những tính năng không cần thiết nhằm giúp cho framework nhẹ nhàng hơn.

5.3.2. BACK-END: FASTAPI

FASTAPI LÀ GÌ?

FastAPI là một framework Python hiện đại và mạnh mẽ dành cho việc xây dựng các ứng dụng API web. Với việc tận dụng các tiêu chuẩn của Python, FastAPI không chỉ giúp tiết kiệm thời gian viết code mà còn đẩy nhanh quá trình triển khai ứng dụng. Điểm đặc biệt của nó là khả năng tạo ra tài liệu API tự động và chi tiết, giúp bạn dễ dàng hiểu rõ và sử dụng các endpoint một cách hiệu quả. Nếu bạn đang tìm kiếm một công cụ linh hoạt, mạnh mẽ và có thể tin cậy để triển khai dự án của mình trong thời gian ngắn mà vẫn đảm bảo chất lượng, FastAPI chính là sự lựa chọn hàng đầu.

Không chỉ dừng lại ở đó, FastAPI còn hỗ trợ kiến thức đa dạng về bảo mật, cho phép triển khai các chiến lược bảo mật một cách linh hoạt và an toàn. Sự kết hợp với các công nghệ như OAuth2, JWT (JSON Web Tokens), và HTTPBearer giúp bảo vệ API của bạn trước các cuộc tấn công và đảm bảo an toàn cho dữ liệu.

Với FastAPI, bạn có thể tận dụng các tính năng kiên trúc dữ liệu tuyệt vời, khả năng xử lý bất đồng bộ và cách mà nó tạo ra tài liệu API tự động để xây dựng ứng dụng của mình một cách nhanh chóng và đáng tin cậy.

Sự kết hợp hoàn hảo giữa hiệu suất cao, tính linh hoạt và bảo mật mạnh mẽ khiến FastAPI trở thành lựa chọn hàng đầu cho bất kỳ dự án nào đòi hỏi tính chất tiên tiến và chất lượng cao.

LÝ DO SỬ DỤNG FASTAPI

- **Nhanh**: Hiệu năng rất cao khi so sánh với NodeJS và Go, và là một trong những Python framework nhanh nhất.
- **Triển khai nhanh**: Tăng tốc độ triển khai các tính năng từ 200% đến 300% (ước tính được dựa trên những kiểm chứng trong nhóm phát triển nội bộ, xây dựng các ứng dụng sản phẩm).
- **Ít lỗi hơn**: Giảm khoảng 40% những lỗi phát sinh bởi con người (nhà phát triển).
- **Trực quan và tiện dụng**: Được nhiều trình chỉnh sửa hỗ trợ, có thể chạy trên nhiều nền tảng khác nhau. Tốn ít thời gian gỡ lỗi hơn (hưởng lợi từ Python là một ngôn ngữ thông dịch).
- **Dễ dàng tiếp cận**: Thiết kế dễ sử dụng và giảm thời gian đọc tài liệu nhờ có docs.
- **Ngắn gọn**: Code ngắn gọn, bao gồm nhiều tính năng đã được triển khai sẵn và giúp giảm thiểu lỗi.
- **Mạnh mẽ**: Code sẵn sàng cho môi trường production và API docs được sinh tự động.
- **Quy chuẩn**: Thiết kế hoàn toàn dựa trên (và hoàn toàn tương thích với) các tiêu chuẩn mở cho APIs : OpenAPI (trước đó được biết đến là Swagger) và JSON Schema.

5.3.3. CƠ SỞ DỮ LIỆU: TORTOISE ORM VÀ POSTGRESQL

TORTOISE ORM LÀ GÌ?

Tortoise ORM là một công cụ ORM (Object Relational Mapper) dành cho **asyncio** rất dễ sử dụng, lấy cảm hứng từ Django - một trong những ORM xuất sắc và phổ biến nhất. Điểm đặc biệt của Tortoise ORM là nó được xây dựng với sự quan tâm đến các mối quan hệ giữa dữ liệu và kế thừa những điểm mạnh của Django ORM. Với Tortoise ORM, bạn không chỉ làm việc với các bảng dữ liệu mà còn làm việc với các quan hệ dữ liệu.

Tortoise ORM hỗ trợ trên CPython >= 3.8 cho SQLite, MySQL và PostgreSQL. Python có nhiều ORM hiện đại và mạnh mẽ, tuy nhiên, hầu hết

chúng được thiết kế theo một mô hình I/O đối lập với asyncio. asyncio là một công nghệ tương đối mới với mô hình đồng thời khác, và thay đổi lớn nhất là về cách xử lý I/O.

Tortoise ORM mang đến sự linh hoạt, dễ sử dụng và tận dụng tốt các khía cạnh của asyncio, là lựa chọn lý tưởng cho những dự án cần kết nối với cơ sở dữ liệu một cách hiệu quả trong môi trường asyncio.

LÝ DO SỬ DỤNG TORTOISE ORM

- **Dễ sử dụng**: Tortoise ORM được thiết kế để dễ tiếp cận và sử dụng. Với cú pháp rõ ràng và gần gũi, việc làm việc với các đối tượng và dữ liệu trong cơ sở dữ liệu trở nên đơn giản hơn.

- **Hỗ trợ mô hình quan hệ**: ORM được tạo ra với mục tiêu chính là làm việc với dữ liệu có mối quan hệ. Điều này giúp bạn xử lý dữ liệu phức tạp hơn, như các mối quan hệ giữa các bảng dữ liệu một cách dễ dàng và hiệu quả.

- **Hỗ trợ asyncio**: Tortoise ORM được tích hợp tốt với asyncio, một công nghệ quan trọng cho xử lý bất đồng bộ trong Python. Điều này giúp tận dụng tối đa hiệu suất của ứng dụng khi xử lý các hoạt động I/O mà không bị chặn.

- **Hỗ trợ đa nền tảng**: Tortoise ORM hỗ trợ các cơ sở dữ liệu phổ biến như SQLite, MySQL và PostgreSQL. Điều này cho phép bạn lựa chọn cơ sở dữ liệu phù hợp với nhu cầu cụ thể của dự án.

- **Tiếp cận theo phong cách Django ORM**: Nếu bạn đã quen thuộc với Django ORM, việc sử dụng Tortoise ORM sẽ trở nên dễ dàng hơn, vì nó được lấy cảm hứng từ và có cú pháp tương tự, giúp việc chuyển đổi và sử dụng trở nên thuận lợi.

- **Sự phát triển và cải tiến liên tục**: Cộng đồng phát triển Tortoise ORM liên tục cải thiện và bổ sung tính năng mới, đồng thời sửa các lỗi phát sinh, giúp nó trở thành một công cụ ngày càng hoàn thiện và tin cậy.

POSTGRESQL LÀ GÌ?

PostgreSQL là một hệ thống quản trị cơ sở dữ liệu quan hệ và đối tượng (object-relational database management system) miễn phí và nguồn mở (RDBMS) tiên tiến nhất hiện nay. khả năng mở rộng cao và tuân thủ các tiêu chuẩn kỹ thuật. Nó được thiết kế để xử lý một loạt các khối lượng công việc lớn, từ các máy tính cá nhân đến kho dữ liệu hoặc dịch vụ Web có nhiều người dùng đồng thời.

LÝ DO SỬ DỤNG POSTGRESQL

- **Cộng đồng lớn và chuyên nghiệp**: PostgreSQL được phát triển và bảo trì bởi một cộng đồng lớn và đầy nhiệt huyết. Những người tham gia trong cộng đồng này đều là những chuyên gia về cơ sở dữ liệu và có kinh nghiệm thực tiễn.

- **Hiệu suất cao**: Có thể xử lý được tập dữ liệu lớn và đa dạng, đồng thời cung cấp các tính năng như phân vùng bảng, lập danh mục nâng cao và truy vấn đọc song song giúp tăng hiệu suất truy vấn.

- **Bảo mật cao**: Cung cấp nhiều tính năng bảo mật như hệ thống kiểm soát truy cập mạnh mẽ, bảo mật cấp độ cột và hàng, xác thực đa dạng và khả năng mã hóa dữ liệu.

- **Tính linh hoạt**: Cho phép định nghĩa các kiểu dữ liệu tùy chỉnh, xây dựng các hàm tùy chỉnh và viết mã từ các ngôn ngữ lập trình khác nhau. Điều này giúp cho các nhà phát triển có thể tùy chỉnh cơ sở dữ liệu cho phù hợp với nhu cầu của ứng dụng.

- **Hỗ trợ trên nhiều hệ điều hành**: PostgreSQL có thể chạy trên nhiều hệ điều hành khác nhau như Linux, Windows và macOS.

- **Cập nhật thường xuyên**: Được phát triển và cập nhật thường xuyên, với các phiên bản mới được phát hành để cải thiện tính năng, tăng hiệu suất và sửa các lỗi bảo mật.

5.4. GIAO DIỆN VÀ LUỒNG HOẠT ĐỘNG CỦA ỨNG DỤNG

5.4.1. TRANG CHỦ

Trang chủ của ứng dụng là trang đầu tiên hiển thị với người dùng khi truy cập vào ứng dụng. Trang này cung cấp các tùy chọn điều hướng rõ ràng để người dùng truy cập các chức năng có sẵn, bao gồm hai nút bấm. Khi người dùng nhấn vào, ứng dụng sẽ điều hướng tới các chức năng tương ứng phù hợp với nhu cầu của người dùng:

- Nút "Một ảnh": Chuyển đến trang chức năng dự đoán loại cây và tình trạng của cây trên một bức ảnh lá cây được người dùng tải lên (gọi tắt là "Trang chức năng dự đoán trên một ảnh").
- Nút "Nhiều ảnh": Chuyển đến trang chức năng dự đoán loại cây và tình trạng của cây trên nhiều bức ảnh lá cây được người dùng tải lên (gọi tắt là "Trang chức năng dự đoán trên nhiều ảnh").

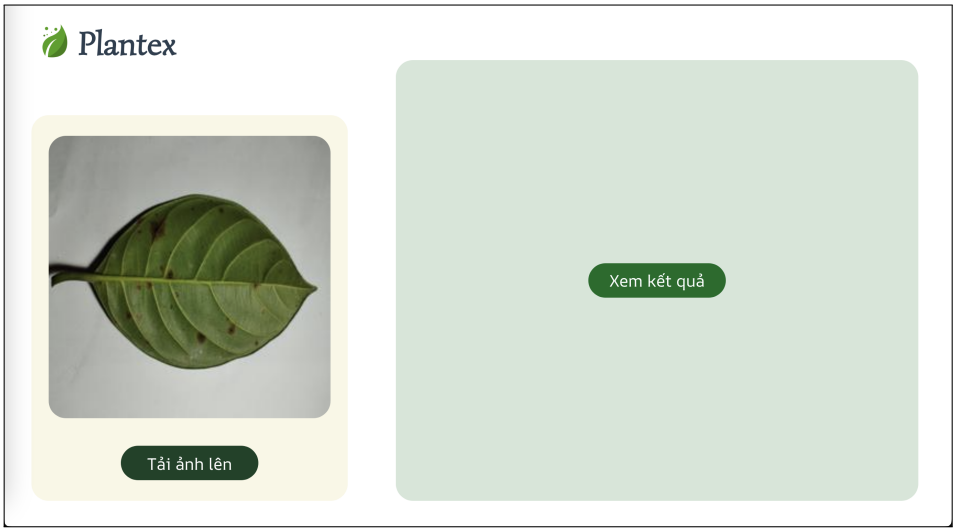
Hình 5.4: Trang chủ

5.4.2. TRANG CHỨC NĂNG DỰ ĐOÁN TRÊN MỘT ẢNH

Ở trang này, người dùng có thể tải lên một file ảnh lá cây bằng cách nhấn nút "Tải ảnh lên"và chọn ảnh có sẵn trên thiết bị.

Hình 5.5: Trang chức năng dự đoán trên một ảnh

Sau khi ảnh được tải lên thành công, nút "Xem kết quả"sẽ xuất hiện như hình dưới đây:

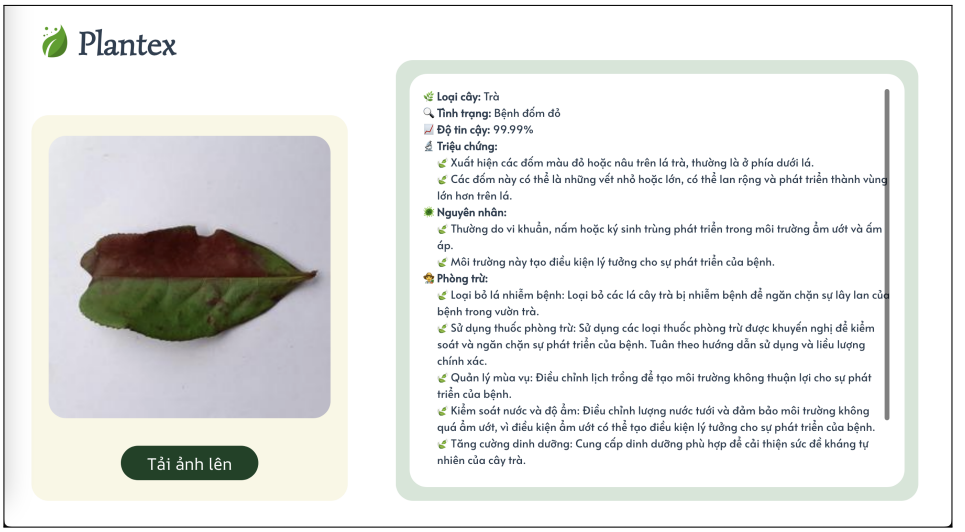


Hình 5.6: Giao diện sau khi ảnh được tải lên thành công

Để xem kết quả dự đoán, người dùng nhấn nút "Xem kết quả". Thông tin về loại cây và tình trạng sức khỏe của cây sẽ được hiển thị như hình dưới đây:

Hình 5.7: Giao diện kết quả dự đoán trên một ảnh (trường hợp cây khỏe mạnh)

Trong trường hợp cây bị sâu bệnh gây hại, ngoài thông tin về tên loại sâu bệnh còn có thêm các thông tin về triệu chứng, nguyên nhân cũng như cách phòng trừ loại sâu bệnh đó.



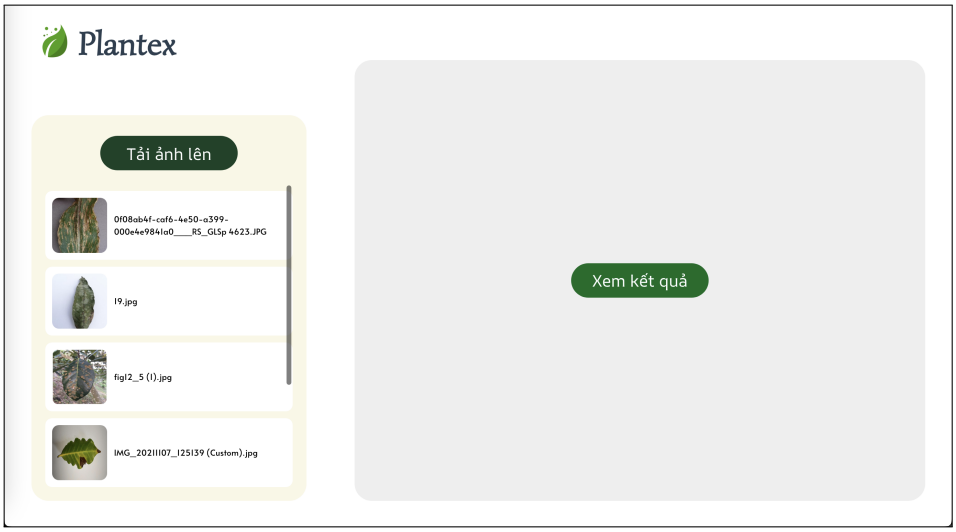
Hình 5.8: Giao diện kết quả dự đoán trên một ảnh (trường hợp cây bị sâu bệnh gây hại)

Nếu ứng dụng không thể nhận diện loại cây cũng như không thể dự đoán tình trạng của cây thì ứng dụng sẽ hiển thị kết quả như hình sau:

Hình 5.9: Giao diện kết quả dự đoán trên một ảnh (trường hợp ứng dụng không thể dự đoán loại cây và tình trạng của cây)

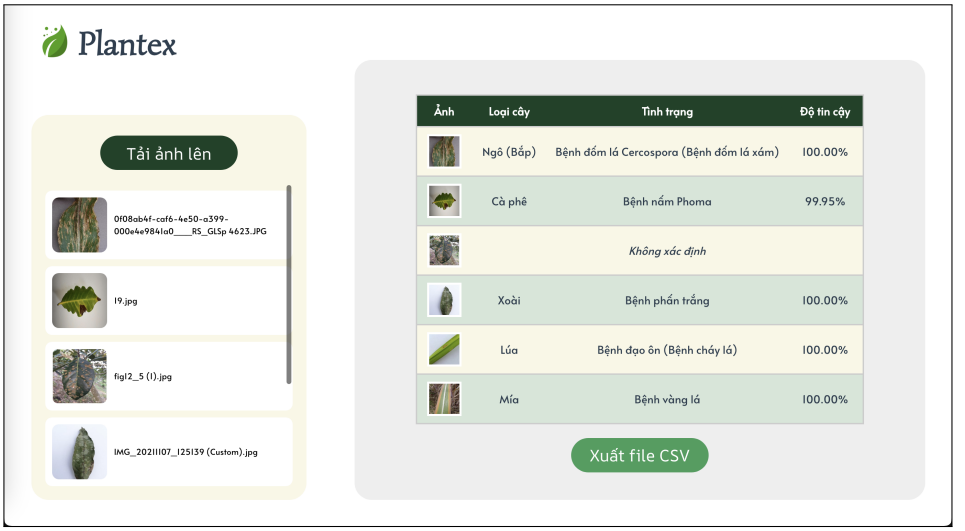
5.4.3. TRANG CHỨC NĂNG DỰ ĐOÁN TRÊN NHIỀU ẢNH

Tương tự với trang chức năng dự đoán trên một ảnh, nhưng ở trang này, người dùng có thể tải lên nhiều file ảnh lá cây bằng cách nhấn nút "Tải ảnh lên"và chọn những ảnh có sẵn trên thiết bị. Sau khi các ảnh được tải lên thành công, nút "Xem kết quả"sẽ xuất hiện như hình dưới đây:



Hình 5.10: Giao diện sau khi các ảnh được tải lên thành công

Để xem kết quả dự đoán, người dùng nhấn nút "Xem kết quả". Thông tin về các loại cây và tình trạng sức khỏe của cây sẽ được hiển thị dưới dạng bảng như hình dưới đây:



Hình 5.11: Giao diện kết quả dự đoán trên nhiều ảnh

Người dùng còn có thể xuất kết quả thành file CSV và tải về thiết bị. File CSV sau khi tải về như sau:

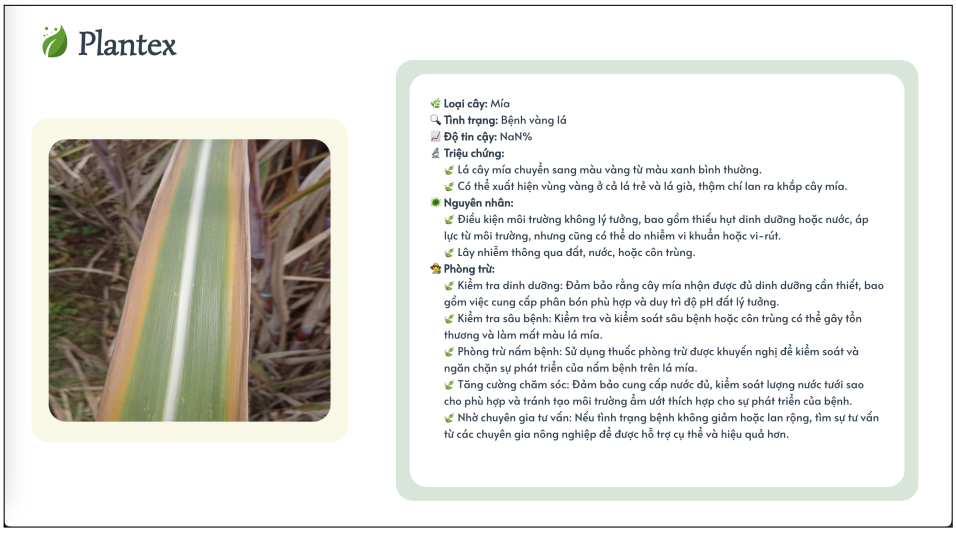
Ảnh	Loại cây	Tình trạng	Độ tin cậy
0f08ab4f-caf6-4e50-a399-000e4e9841a0___RS_GLSp 4623.JPG	Ngô (Bắp)	Bệnh đốm lá Cercospora (Bệnh đốm lá xám)	100.00%
19.jpg	Cà phê	Bệnh nấm Phoma	99.95%
fig12_5 (1).jpg	Không xác định	-	-
IMG_20211107_125139 (Custom).jpg	Xoài	Bệnh phấn trắng	100.00%
leaf_blast (16).jpg	Lúa	Bệnh đạo ôn (Bệnh cháy lá)	100.00%
yellow (105).jpeg	Mía	Bệnh vàng lá	100.00%

Hình 5.12: File CSV chứa kết quả dự đoán trên nhiều ảnh

Người dùng có thể xem chi tiết về dự đoán của từng ảnh bằng cách nhấp vào hàng của ảnh tương ứng để chuyển đến trang thông tin chi tiết.

5.4.4. TRANG THÔNG TIN CHI TIẾT CHO MỖI ẢNH

Trang này có cấu trúc và các thông tin được hiển thị gần giống với trang chức năng dự đoán trên một ảnh, nhưng không có nút "Tải ảnh lên".



Hình 5.13: Trang thông tin chi tiết cho mỗi ảnh

5.5. CƠ SỞ DỮ LIỆU

5.5.1. THU THẬP DỮ LIỆU

HÌNH ẢNH

Việc thu thập dữ liệu hình ảnh cho ứng dụng nhằm mục đích huấn luyện các mô hình phân đoạn lá cây, phân loại sâu bệnh. Nguồn dữ liệu như đã được đề cập trong phần .... Những nguồn này đã góp phần tạo nên kho hình ảnh phong phú và đa dạng, cho phép các mô hình học hỏi từ nhiều loại cây, nhiều loại sâu bệnh gây hại và điều kiện môi trường khác nhau.

VĂN BẢN

Dữ liệu văn bản trong cơ sở dữ liệu liên quan đến các loại sâu bệnh trên cây trồng, bao gồm thông tin chi tiết về tên gọi, các triệu chứng, nguyên nhân và phương pháp phòng trừ, được trích xuất từ nhiều bài báo, nghiên cứu và tài liệu trong lĩnh vực nông nghiệp. Thông tin này đã được tuyển chọn một cách tỉ mỉ và kỹ lưỡng, cho phép ứng dụng cung cấp thông tin và hiểu biết toàn diện cho người dùng.

5.5.2. TỔ CHỨC CƠ SỞ DỮ LIỆU VỚI TORTOISE ORM

Để lưu trữ thông tin trong cơ sở dữ liệu, tôi định nghĩa một `class` có tên là `Plant_Diseases` như một cách biểu diễn có cấu trúc về các loại sâu bệnh trong phạm vi của ứng dụng. `class Plant_Diseases` bao gồm các thuộc tính như sau:

- `id`: Trường này là một số nguyên được sử dụng làm khóa chính cho mỗi class của mô hình, duy nhất và tự động tăng lên cho mỗi mục mới được thêm vào cơ sở dữ liệu.
- `name`: Trường ký tự có độ dài tối đa 255 ký tự, lưu trữ tên class (nhãn) của một loại sâu bệnh cụ thể.
- `plant`: Trường ký tự có độ dài tối đa 255 ký tự, lưu trữ tên loại cây thuộc class tương ứng.
- `disease`: Trường ký tự có độ dài tối đa 255 ký tự, lưu trữ tên loại sâu bệnh thuộc class tương ứng.
- `symptoms`: Trường JSON được sử dụng để lưu trữ thông tin về những triệu chứng liên quan đến loại sâu bệnh đó, có cấu trúc dạng mảng, là một tập hợp các triệu chứng ở dạng văn bản.
- `causes`: Trường JSON được sử dụng để lưu trữ thông tin về những nguyên nhân gây ra loại sâu bệnh đó, có cấu trúc dạng mảng, là một tập hợp các nguyên nhân ở dạng văn bản.
- `measures`: Trường JSON được sử dụng để lưu trữ thông tin về những phưpng pháp được đề xuất để phòng trừ loại sâu bệnh đó, có cấu trúc dạng mảng, là một tập hợp các phương pháp phòng trừ ở dạng văn bản.

6

ĐÁNH GIÁ

Trong chương này, các thông số và tiêu chí dùng để đánh giá ứng dụng sẽ được đề cập, bao gồm các thông số đánh giá mô hình phân loại sâu bệnh và các tiêu chí đánh giá toàn bộ ứng dụng một cách tổng quan.

6.1. CÁC THÔNG SỐ VÀ TIÊU CHÍ ĐÁNH GIÁ

6.1.1. ĐÁNH GIÁ MÔ HÌNH PHÂN LOẠI SÂU BỆNH

LOSS FUNCTION

Trong bài toán phân loại nhiều class (ở đây là phân loại sâu bệnh trên cây trồng), YOLOv8 sử dụng loss function là Categorical Cross-entropy. Categorical Cross-entropy loss phù hợp với các bài toán trong đó mỗi mẫu đầu vào chỉ thuộc về một class. Nó đo sự sai khác giữa xác suất class được dự đoán và nhãn thực tế của class. Gọi p_i đại diện cho xác suất dự đoán và g_i đại diện cho nhãn class thực tế được mã hóa one-hot cho mỗi class i trong C. Công thức Categorical Cross-entropy loss là:

$$L_{categorical\ CE} = -\sum_{i}^{C} g_i \log(p_i)$$

Ngoài ra, để nâng cao hơn nữa hiệu suất của mô hình và giảm thiểu tác động của sự mất cân bằng giữa các class (nếu có trong tập dữ liệu), có thể xem xét kết hợp các kỹ thuật như Loss-weighting. Loss-weighting chỉ định trọng số cao hơn cho các class thiểu số để giảm bớt tác động của việc phân phối dữ liệu không cân bằng.

CÁC THÔNG SỐ ĐÁNH GIÁ

Khi đánh giá hiệu suất của module thứ hai - phân loại tình trạng của cây cà chua bằng mô hình CNN riêng biệt, một số chỉ số dưới đây thường được sử dụng để đánh giá hiệu quả và độ chính xác của mô hình:

- **Accuracy**: Accuracy đo mức độ chính xác tổng thể các dự đoán của mô hình bằng cách tính tỷ lệ các mẫu được phân loại chính xác trên tổng số mẫu. Nó cung cấp một thước đo chung về mức độ hiệu quả của mô hình trong việc phân loại tình trạng của cây cà chua.

- **Precision**: Precision đo tỷ lệ các mẫu positive được phân loại chính xác trong số tất cả các mẫu được dự đoán là positive.

- **Recall**: Recall hay còn được gọi là **sensitivity (độ nhạy)** đo tỷ lệ các mẫu positive được phân loại chính xác trong số tất cả các mẫu positive thực tế.

- **F1-score**: F1-score là trung bình điều hòa giữa precision và recall, cung cấp thước đo cân bằng về hiệu suất phân loại.

- **Confusion matrix**: Confusion matrix cung cấp phân tích toàn diện các dự đoán của mô hình trên các class khác nhau. Nó cho biết số lượng mẫu true positive, true negative, false positive và false negetive cho mỗi class. Từ confusion matrix, các số liệu như accuracy, precision, recall và F1-score có thể được tính cho các class riêng lẻ.

- **Tốc độ và hiệu quả**: Tương tự như module đầu tiên, tốc độ và hiệu quả của mô hình trong quá trình inferrence cũng rất quan trọng. Các số liệu như thời gian inferrence hoặc **latency (độ trễ)** có thể được sử dụng để đánh giá hiệu quả tính toán của mô hình.

6.1.2. ĐÁNH GIÁ TOÀN BỘ ỨNG DỤNG

GIAO DIỆN

- Thiết kế màu sắc đẹp mắt, hình ảnh đặc sắc thu hút người truy cập.

- Bố cục giao diện hợp lý, sắp xếp logic, khoa học, sử dụng các font chữ đồng nhất, theo chuẩn kích cỡ, làm nổi bật nội dung.

- Giao diện web hiện đại, áp dụng những công nghệ tiên tiến, bắt kịp sự thay đổi của thời đại.

TÍNH DỄ SỬ DỤNG

- Các thao tác đơn giản, dễ thực hiện, các chú thích, hướng dẫn dễ hiểu.

- Các nút chuyển hướng trước sau, thanh điều hướng có sắp xếp, bố cục trong tầm mắt người dùng, các liên kết, các đường dẫn, các nút bấm được ấn hoặc chạm một các đơn giản trên mọi thiết bị cũng như việc chuyển qua lại giữa các trang tương tác nhanh chóng.

TỐC ĐỘ PHẢN HỒI

Tốc độ phản hồi của ứng dụng là một yếu tố quan trọng, quyết định đến trải nghiệm người dùng. Ứng dụng phải được tối ưu hóa để cung cấp phản hồi nhanh chóng và hiệu quả. Khi người dùng tải ảnh lên hoặc yêu cầu xem kết quả dự đoán, hệ thống phản hồi ngay lập tức mà không gây chậm trễ đáng kể. Thời gian chờ đợi ngắn giúp người dùng cảm thấy hài lòng với trải nghiệm của họ và tăng tính tương tác.

XỬ LÝ LỖI

Khả năng xử lý lỗi và ngoại lệ của ứng dụng phải được chú trọng để đảm bảo trải nghiệm người dùng không bị gián đoạn bởi các vấn đề kỹ thuật. Các lỗi phải được xử lý một cách tự động, giúp người dùng tránh được những trở ngại không mong muốn. Mọi tình huống lỗi đều được đưa ra thông báo rõ ràng và cung cấp hướng dẫn cụ thể để người dùng có thể giải quyết hoặc báo cáo lỗi.

BẢO MẬT

Ứng dụng phải được tối ưu với mục tiêu bảo mật thông tin cá nhân của người dùng. Dù không yêu cầu đăng nhập cũng như không lưu trữ thông tin và lịch sử sử dụng, nhưng mọi dữ liệu được tải lên phải được xử lý một cách riêng tư và an toàn, không ghi nhận thông tin cá nhân của bất kỳ ai.

TÍNH ỔN ĐỊNH

Tính ổn định của ứng dụng được coi là một trong những yếu tố quan trọng nhất. Mọi tính năng đều phải được kiểm tra và thử nghiệm một cách kỹ lưỡng để đảm bảo chúng hoạt động một cách mượt mà và không gây ra bất kỳ trục trặc nào trong quá trình sử dụng. Các phiên bản cập nhật luôn được kiểm tra kỹ lưỡng trước khi triển khai để đảm bảo rằng ứng dụng luôn ổn định và hiệu quả cho người dùng.

KHẢ NĂNG TƯƠNG THÍCH

Để đảm bảo rằng ứng dụng hoạt động một cách linh hoạt và tương thích trên nhiều thiết bị và môi trường khác nhau, việc kiểm tra và đảm bảo khả năng tương thích là rất quan trọng.

- Ứng dụng phải được xây dựng với tiêu chuẩn chất lượng cao, phát triển sao cho có thể hoạt động một cách tối ưu trên các nền tảng khác nhau bao gồm cả trên các trình duyệt web và thiết bị di động.
- Khả năng tương thích của ứng dụng phải được kiểm tra kỹ lưỡng trên các trình duyệt phổ biến như Chrome, Firefox, Safari, Edge,... cũng như trên các thiết bị di động chạy hệ điều hành iOS và Android.

Qua đó, người dùng có thể trải nghiệm một cách trơn tru mà không gặp phải vấn đề tương thích khi sử dụng ứng dụng trên các thiết bị khác nhau.

KHẢ NĂNG MỞ RỘNG

Để đáp ứng được nhu cầu mở rộng và cải tiến trong tương lai, ứng dụng phải được xây dựng với kiến trúc linh hoạt, cho phép dễ dàng thêm các tính năng mới và mở rộng khả năng hoạt động.

- Cấu trúc code của ứng dụng phải được thiết kế một cách có tổ chức, module hóa giúp việc thêm, chỉnh sửa và nâng cấp các tính năng trở nên dễ dàng hơn.
- Các giao diện và logic trong ứng dụng phải được tách rời, đảm bảo tính riêng biệt giữa các thành phần khác nhau, từ đó tạo điều kiện thuận lợi cho việc mở rộng chức năng và phát triển ứng dụng trong tương lai.

Qua đó, khả năng mở rộng của ứng dụng được tối ưu hóa, giúp dễ dàng thích nghi với các yêu cầu và xu hướng mới mà không gặp phải các rắc rối về cấu trúc hoặc hiệu suất.

6.2. KẾT QUẢ ĐÁNH GIÁ MÔ HÌNH PHÂN LOẠI SÂU BỆNH

Hiệu quả của mô hình phân loại sâu bệnh đã được thể hiện ở bảng 4.3. Với kết quả đều cao trên cả tập dữ liệu chính và tập dữ liệu kiểm tra overfitting, mô hình đã cho thấy tiềm năng mở rộng để có thể triển khai ứng dụng trong thực tế, giúp tạo ra một công cụ đắc lực hỗ trợ con người trong việc phát hiện và nhận diện sâu bệnh trên cây trồng.

7

TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN

Chương này tổng kết một cách ngắn gọn về những kết quả đã đạt được, đồng thời nêu lên những thuận lợi cũng như những khó khăn trong quá trình thực hiện Đồ án và qua đó đề ra hướng phát triển tiếp theo trong tương lai.

7.1. KẾT QUẢ ĐẠT ĐƯỢC

7.1.1. ĐỐI VỚI CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ

Tìm hiểu, nghiên cứu, bổ sung được các kiến thức, kỹ năng liên quan đến xử lý ảnh và thị giác máy tính cũng như phát triển ứng dụng Web:

VỀ LĨNH VỰC XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH

Nắm được lý thuyết cũng như cách hiện thực một số bài toán trong thị giác máy tính như phân loại hình ảnh, phân đoạn hình ảnh, phát hiện đối tượng,...

VỀ LĨNH VỰC PHÁT TRIỂN ỨNG DỤNG WEB

- Học được thêm về cách sử dụng một số ngôn ngữ, thư viện, framework mới: Vue.js, FastAPI, PostgreSQL,...
- Biết cách sử dụng Postman để kiểm tra API.
- Biết cách sử dụng Docker để xây dựng ứng dụng.

7.1.2. ĐỐI VỚI MỤC TIÊU ĐỀ TÀI

- Phân tích được các yêu cầu chức năng và phi chức năng, vẽ các use case, activity diagram, phân tích được phạm vi và ngoại trừ của đề tài.
- Huấn luyện được một mô hình có thể phân loại 40 loại sâu bệnh từ 8 loại cây khác nhau.
- Phát triển được một ứng dụng Web với 2 chức năng chính cho phép người dùng tải lên một hoặc nhiều ảnh lá cây để xem các dự đoán về loại cây và tình trạng sức khỏe của cây.

7.2. NHỮNG THUẬN LỢI VÀ KHÓ KHĂN TRONG QUÁ TRÌNH PHÁT TRIỂN ỨNG DỤNG

7.2.1. THUẬN LỢI

- **Sử dụng công nghệ mới**: Sự phát triển nhanh chóng của công nghệ AI, Machine Learning và Computer Vision đã cung cấp các công cụ mạnh mẽ để xây dựng mô hình phát hiện và nhận diện sâu bệnh trên cây trồng. Điều này đã tạo điều kiện thuận lợi cho quá trình nghiên cứu và phát triển ứng dụng.
- **Sự đa dạng của dữ liệu**: Sự phong phú và đa dạng của dữ liệu về các loại cây trồng và các loại sâu bệnh đã giúp cải thiện độ chính xác của mô hình. Việc có nhiều dữ liệu đa dạng giúp mô hình học được các triệu chứng và đặc điểm của từng loại bệnh một cách chính xác hơn.
- **Tính linh hoạt của công nghệ**: Sử dụng các framework và thư viện mã nguồn mở, cùng với sự linh hoạt trong việc phát triển, đã giúp cho các thử nghiệmn trở nên dễ dàng và cải thiện hiệu suất mô hình một cách nhanh chóng.

7.2.2. KHÓ KHĂN

- **Dữ liệu hạn chế và không cân đối**: Mặc dù có sự đa dạng, nhưng dữ liệu về một số loại sâu bệnh trên cây trồng vẫn hạn chế. Điều này có thể làm giảm độ chính xác của mô hình khi đối mặt với những bệnh hiếm hoặc ít gặp.
- **Chú thích dữ liệu**: Việc chú thích cho dữ liệu (đặc biệt là dữ liệu dùng cho việc phân đoạn lá cây) một cách chính xác và chi tiết đòi

hỏi phải mất nhiều thời gian và công sức. Do đó việc chú thích dữ liệu một cách thủ công đối với lượng dữ liệu lớn hầu như không khả thi.

- **Thách thức trong việc tối ưu hóa mô hình**: Việc tối ưu hóa mô hình để đạt được độ chính xác cao đồng thời giảm thiểu khả năng **overfitting** và **underfitting** để phù hợp với dữ liệu trong thực tế cũng là một thách thức đáng kể.

- **Yêu cầu phức tạp từ người dùng**: Người dùng cần một giao diện thân thiện và dễ sử dụng. Điều này đặt ra một thách thức lớn trong việc thiết kế giao diện và trải nghiệm người dùng tốt.

- **Vấn đề về hiệu suất**: Việc xử lý dữ liệu lớn và phân loại nhanh chóng đôi khi gặp khó khăn với một số tài nguyên hạn chế, đặc biệt khi triển khai trên các thiết bị có cấu hình thấp.

7.3. HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI

- **Tối ưu hóa hiệu suất mô hình**: Nghiên cứu và thử nghiệm các kỹ thuật mới để tối ưu hóa mô hình phân loại và phát hiện sâu bệnh trên cây trồng. Các chiến lược bao gồm tinh chỉnh siêu tham số, mở rộng cơ sở dữ liệu, sử dụng **Transfer Learning (Học chuyển tiếp)** và fine-tuning, cải thiện kiến trúc mạng và tối ưu hóa thời gian chạy. Những nỗ lực này nhằm đạt được một mô hình mạnh mẽ, có độ chính xác cao và có khả năng triển khai linh hoạt, giúp ứng dụng hiệu quả hơn trong việc hỗ trợ nông nghiệp phát triển và quản lý sức khỏe cây trồng.

- **Mở rộng dữ liệu**: Tăng cường và mở rộng cơ sở dữ liệu với nhiều loại cây trồng và sâu bệnh khác nhau. Việc thu thập thêm dữ liệu từ nhiều nguồn khác nhau như PlantVillage và các nguồn dữ liệu trực tuyến khác cung cấp độ đa dạng và phong phú cho tập dữ liệu huấn luyện. Có thể kỹ thuật **Data Augmentation (Tăng cường dữ liệu)** sẽ được áp dụng để tạo ra các biến thể mới từ dữ liệu hiện có, bao gồm việc xoay, phóng to, thu nhỏ, cắt, lật ảnh, hay thay đổi độ sáng tạo nên các điều kiện ánh sáng khác nhau, giúp mô hình học được các biến thể của hình ảnh, làm phong phú thêm tập dữ liệu huấn luyện mà không cần tốn thời gian và chi phí để thu thập dữ liệu mới, góp phần đảm bảo tính đa dạng, biểu diễn và phản ánh chính xác nhất về các điều kiện trong thực tế.

- **Phát triển tính năng mới**: Xây dựng thêm các tính năng hỗ trợ như gợi ý điều trị, hướng dẫn chăm sóc cây trồng, hoặc tích hợp với dịch vụ phân tích dữ liệu thời tiết để cung cấp dự báo về các điều kiện nông nghiệp.

- **Kiểm thử và đánh giá**: Tiến hành các bài kiểm tra thực tế để đánh giá hiệu suất và khả năng áp dụng của ứng dụng trong điều kiện thực tế. Thu thập phản hồi từ người dùng để cải thiện trải nghiệm sử dụng.

- **Tăng cường bảo mật**: Để đảm bảo tính bảo mật cho dữ liệu nhạy cảm về cây trồng và sâu bệnh, việc sử dụng các phương pháp mã hóa, bảo vệ dữ liệu và quản lý truy cập là cần thiết. Các biện pháp an ninh như mã hóa dữ liệu trong quá trình truyền và lưu trữ, cơ chế xác thực người dùng, và giám sát an ninh hệ thống đều đóng vai trò quan trọng trong việc đảm bảo rằng thông tin được bảo vệ một cách an toàn nhất.

- **Quản lý dữ liệu**: Việc quản lý dữ liệu một cách có hiệu quả cũng cần được chú trọng. Điều này bao gồm việc phân loại, đánh chỉ mục, và lưu trữ dữ liệu một cách có tổ chức, dễ dàng truy xuất và xử lý khi cần thiết. Các chính sách quản lý dữ liệu phải được thiết lập rõ ràng để đảm bảo tính chính xác và đáng tin cậy của dữ liệu.

- **Phát triển ứng dụng di động**: Xây dựng một phiên bản ứng dụng di động để cung cấp sự tiện lợi và tiếp cận rộng rãi hơn đối với người dùng.

- **Hợp tác và triển khai**: Tìm kiếm cơ hội hợp tác với các tổ chức nghiên cứu, doanh nghiệp nông nghiệp hoặc cơ quan chính phủ để triển khai và áp dụng ứng dụng vào thực tế.

TÀI LIỆU THAM KHẢO

[1] P. B. Padol and A. A. Yadav, "SVM Classifier Based Grape Leaf Disease Detection," in *2016 Conference on Advances in Signal Processing (CASP)*, pp. 175–179, 2016.

[2] K. Singh, S. Kumar, and P. Kaur, "Support Vector Machine Classifier Based Detection of Fungal Rust Disease in Pea Plant (*Pisam sativam*)," *Int. J. Inf. Tecnol.*, vol. 11, pp. 485–492, 2019.

[3] N. R. Bhimte and V. R. Thool, "Diseases Detection of Cotton Leaf Spot Using Image Processing and SVM Classifier," in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, (Madurai, India), pp. 340–344, 2018.

[4] S. Kumar, B. Sharma, V. Sharma, H. Sharma, and J. Bansal, "Plant Leaf Disease Ddentification Using Exponential Spider Monkey Optimization," *Sustainable Computing: Informatics and Systems*, 2018.

[5] S. Hossain, R. M. Mou, M. M. Hasan, S. Chakraborty, and M. A. Razzak, "Recognition and Detection of Tea Leaf's Diseases Using Support Vector Machine," in *2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA)*, (Batu Feringghi), pp. 150–154, 2018.

[6] A. Aruraj, A. Alex, M. S. P. Subathra, N. J. Sairamya, S. T. George, and S. E. V. Ewards, "Detection and Classification of Diseases of Banana Plant Using Local Binary Pattern and Support Vector Machine," in *2019 2nd International Conference on Signal Processing and Communication (ICSPC)*, (Coimbatore, India), pp. 231–235, 2019.

[7] A. Parikh, M. S. Raval, C. Parmar, and S. Chaudhary, "Disease Detection and Severity Estimation in Cotton Plant from Unconstrained Images," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, (Montreal, QC), pp. 594–601, 2016.

[8] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes, "Deep Learning for Image-based Cassava Disease Detection," *Frontiers in Plant Science*, vol. 8, p. 1852, 2017.

[9] M. Suresha, K. N. Shreekanth, and B. V. Thirumalesh, "Recognition of Diseases in Paddy Leaves Using KNN Classifier," in *2017 2nd International Conference for Convergence in Technology (I2CT)*, (Mumbai), pp. 663–666, 2017.

[10] E. Hossain, M. F. Hossain, and M. A. Rahaman, "A Color and Texture Based Approach for the Detection and Classification of Plant Leaf Disease Using KNN Classifier," in *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, (Cox'sBazar, Bangladesh), pp. 1–6, 2019.

[11] J. Abdulridha, R. Ehsani, A. Abd-Elrahman, and Y. Ampatzidis, "A remote sensing technique for detecting laurel wilt disease in avocado in presence of other biotic and abiotic stresses," *Computers and Electronics in Agriculture*, vol. 156, pp. 549–557, 2019.

[12] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," *Computers and Electronics in Agriculture*, vol. 173, p. 105393, 2020.

[13] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science*, vol. 7, p. 1419, 2016.

[14] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of Rice Diseases Using Deep Convolutional Neural Networks," *Neurocomputing*, vol. 267, pp. 378–384, 2017.

[15] R. Gandhi, S. Nimbalkar, N. Yelamanchili, and S. Ponkshe, "Plant Disease Detection Using CNNs and GANs as An Augmentative Approach," in *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, (Bangkok), pp. 1–5, 2018.

[16] K. P. Ferentinos, "Deep Learning Models for Plant Disease Detection and Diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.

[17] A. Khamparia, G. Saini, D. Gupta, A. Khanna, S. Tiwari, and V. H. C. Albuquerque, "Seasonal Crops Disease Prediction and Classification Using Deep Convolutional Encoder Network," *Circuits, Systems, and Signal Processing*, vol. 39, pp. 818–836, 2019.

[18] K. Kamal, Z. Yin, M. Wu, and Z. Wu, "Depthwise Separable Convolution Architectures for Plant Disease Classification," *Computers and Electronics in Agriculture*, 2019.

[19] G. Geetharamani and A. Pandian, "Identification of Plant Leaf Diseases Using A Nine-layer Deep Convolutional Neural Network," *Computers & Electrical Engineering*, vol. 76, pp. 323–338, 2019.

[20] R. Karthik, M. Hariharan, S. Anand, P. Mathikshara, A. Johnson, and R. Menaka, "Attention Embedded Residual CNN for Disease Detection in Tomato Leaves," *Applied Soft Computing*, vol. 86, 2020.

[21] B. Zhang, R. Wang, H. Zhang, C. Yin, Y. Xia, M. Fu, and W. Fu, "Dragon fruit detection in natural orchard environment by integrating lightweight network and attention mechanism," *Frontiers in Plant Science*, vol. 13, 2022.

[22] M. Soeb, M. Jubayer, and T. e. a. Tarin, "Tea leaf disease detection and identification based on YOLOv7 (YOLO-T)," *Scientific Reports*, vol. 13, p. 6078, 2023.

[23] J. L. de Moraes, J. de Oliveira Neto, C. Badue, T. Oliveira-Santos, and A. F. de Souza, "Yolo-Papaya: A Papaya Fruit Disease Detector and Classifier Using CNNs and Convolutional Block Attention Modules," *Electronics*, vol. 12, no. 10, 2023.

[24] J. Liu and X. Wang, "Tomato Diseases and Pests Detection Based on Improved Yolo V3 Convolutional Neural Network," *Frontiers in Plant Science*, vol. 11, 2020.

[25] V. S. Kumar, M. Jaganathan, A. Viswanathan, M. Umamaheswari, and J. Vignesh, "Rice leaf disease detection based on bidirectional feature attention pyramid network with YOLO v5 model," *Environmental Research Communications*, vol. 5, p. 065014, jun 2023.

[26] S. Rath, "YOLOv8 Ultralytics: State-of-the-Art YOLO Models." `http://rb.gy/rjq09`, Jan 2023.

[27] F. Jacob Solawetz, "What is YOLOv8? The Ultimate Guide.." `http://rb.gy/o1rc1`, Jan 2023.

[28] Hughes, "PlantVillage Dataset | Papers With Code." `http://rb.gy/z1qle`.

[29] university, " grape segmentataion Dataset ." https://rb.gy/srny6l, Nov 2023.

[30] M. Ibrahim, "A Gentle Introduction to Image Classification | ml-articles – Weights & Biases." http://rb.gy/gae6u, Mar 2023.

[31] "What is image classification? Basics you need to know | SuperAnnotate." http://rb.gy/u3cus, Nov 2021.

[32] A. Gotarne and J. Bhanushali, "Challenges in Image Classification – Diycam." http://rb.gy/bm9xy.

[33] N. H. Nam, "Giới thiệu về Support Vector Machine trong Machine Learning." http://rb.gy/tf4eu.

[34] V. H. Tiệp, "K-nearest neighbors - Machine Learning cơ bản." http://rb.gy/v5slx, Jan 2017.

[35] A. Saeed, A. A. Abdel-Aziz, A. Mossad, M. A. Abdelhamid, A. Y. Alkhaled, and M. Mayhoub, "Smart Detection of Tomato Leaf Diseases Using Transfer Learning-Based Convolutional Neural Networks," *Agriculture*, vol. 13, p. 139, Jan 2023.

[36] D. Li, F. Ahmed, N. Wu, and A. I. Sethi, "YOLO-JD: A Deep Learning Network for Jute Diseases and Pests Detection from Images," *Plants*, vol. 11, p. 937, Mar 2022.

[37] khánh hoàng, "Giới thiệu framework Vue.js." https://rb.gy/c2ge5j, 2020 Feb.

[38] H. Nguyễn, "VueJS là gì? Vì sao nên sử dụng Vue.js?." https://rb.gy/byfbaa, Apr 2023.

[39] M. Herman, "Developing a Single Page App with FastAPI and Vue.js | TestDriven.io." https://rb.gy/xq10kw, Dec 2022.

[40] M. Quang, "VueJS là gì? Ứng dụng và một vài tính năng nổi trội của VueJS." https://rb.gy/354xlu, Dec 2022.

[41] John, "FastAPI (Phần 1): Giới thiệu và setup môi trường." https://rb.gy/afnoca, May 2023.

[42] "FastAPI." https://fastapi.tiangolo.com/vi/.

[43] N. V. Thịnh, "Tìm hiểu hệ quản trị cơ sở dữ liệu PostgreSQL." https://rb.gy/8iq07c, Aug 2021.

[44] "PostgreSQL là gì? 6 lý do nên sử dụng PostgreSQL? - Tin tức tên miền hosting." https://rb.gy/wz05ot, Apr 2023.

[45] V. Hoàng, "Tìm hiểu về YOLO trong bài toán real-time object detection." https://rb.gy/84mi86, Nov 2019.

[46] H. Ng, "YOLOv1 - Sự khởi nguồn mới trong Object Detection." https://rb.gy/j4h5ne, Sep 2022.

[47] "Phân đoạn hình ảnh: Các kỹ thuật truyền thống & phương pháp học sâu - VinBigData." https://rb.gy/27taws, Apr 2023.

[48] P. Đình Khánh, "Khoa học dữ liệu." https://rb.gy/xai08w, Jun 2020.

[49] M. L. cơ bản, "Bài 33: Các phương pháp đánh giá một hệ thống phân lớp." https://rb.gy/ocsu05, Jan 2018.

[50] "Nằm lòng 7 tiêu chí đánh giá khả năng sử dụng trang web - MẮT BÃO WS - Dịch Vụ Thiết Kế Website Chuyên Nghiệp." https://rb.gy/gfvmcv, Aug 2018.

[51] D. NGUYÊN, "Các tiêu chí đánh giá website chuyên nghiệp - WEBMINI." https://rb.gy/0jrgn2, May 2022.

[52] P. Kulkarni1 and A. Karwande1, "Plant Disease Detection Using Image Processing and Machine Learning," 2023.

[53] P. Sajitha, A. John, V. L. Devika, G. S. V, and N. Sakhir, "Leaf Disease Detection & Correction using YOLO V7 with GPT3 integrated," *International Journal of Engineering Research & Technology (IJERT)*, vol. 12, June 2023. Issue 06.

[54] M. Shoaib, T. Hussain, B. Shah, I. Ullah, S. Shah, F. Ali, and S. Park, "Deep learning-based segmentation and classification of leaf images for detection of tomato plant disease," *Frontiers in Plant Science*, vol. 13, p. 1031748, 2022.

[55] A. J., J. Eunice, D. E. Popescu, M. K. Chowdary, and J. Hemanth, "Deep Learning-Based Leaf Disease Detection in Crops Using Images for Agricultural Applications," *Agronomy*, vol. 12, no. 10, 2022.

[56] S. Datta and N. Gupta, "A Novel Approach For the Detection of Tea Leaf Disease Using Deep Neural Network,"

[57] N. A. S. Muhammad Juman Jhatial, Riaz Ahmed Shaikh, "Deep Learning-Based Rice Leaf Diseases Detection Using Yolov5,"

[58] S. A. Qadri, N.-F. Huang, T. Wani, and S. Ph.D., "Plant Disease Detection and Segmentation using End-to-End YOLOv8: A Comprehensive Approach," pp. 155–160, Aug 2023.

[59] H. Orchi, M. Sadik, M. Khaldoun, and E. Sabir, "Real-Time Detection of Crop Leaf Diseases Using Enhanced YOLOv8 algorithm," in *2023 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 1690–1696, 2023.

[60] VinBigData, "YOLOv8 có gì nâng cấp so với các phiên bản trước?." `https://rb.gy/8vbfrg`, Aug 2023.

[61] M. AI, "Thử tìm hiểu về mAP - đo lường Object Detection model." `https://rb.gy/q2a6ha`, Oct 2021.

[62] M. . documentation, "Algorithm principles and implementation with YOLOv8." `https://rb.gy/rvwcmo`.

[63] Javatpoint, "Overfitting in Machine Learning." `https://rb.gy/mu1poj`.

[64] E. D. Science, "Overfitting in Machine Learning: What It Is and How to Prevent It." `https://rb.gy/b2q1mz`, Jul 2022.