

**TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG KHOA
CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN**

ĐỀ TÀI:

**NGHIÊN CỨU VÀ XÂY DỰNG WEB TRONG NHẬN DẠNG TRÁI CHUỐI CHO
NÔNG NGHIỆP VIỆT NAM DỰA TRÊN NỀN TẢNG AI**

Người hướng dẫn: **TS NGUYỄN HÀ HUY CƯỜNG**

Sinh viên thực hiện: **NGUYỄN KIM HOÀNG ĐỨC**

Mã số sinh viên: **2151220227**

Lớp: **21CT5**

Đà Nẵng, 12/2025

**TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG KHOA
CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN**

ĐỀ TÀI:

**NGHIÊN CỨU VÀ XÂY DỰNG WEB TRONG NHẬN DẠNG TRÁI CHUỐI CHO
NÔNG NGHIỆP VIỆT NAM DỰA TRÊN NỀN TẢNG AI**

Người hướng dẫn: **TS. NGUYỄN HÀ HUY CƯỜNG**
Sinh viên thực hiện: **NGUYỄN KIM HOÀNG ĐỨC**
Mã số sinh viên: **2151220227**
Lớp: **21CT5**

{Trang trắng này dùng để dán bản Nhận xét của người hướng dẫn, hoặc thay trang này bằng Nhận xét của người hướng dẫn}

{Trang trắng này dùng để dán bản Nhận xét của người phản biện, hoặc thay trang này bằng Nhận xét của người phản biện

Mục Lục

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT VÀ TỔNG QUAN CÔNG NGHỆ	4
1.1 Tổng quan về Nhận diện Đối tượng (Object Detection) và Ứng dụng trong Nông nghiệp	4
1.1.1 Khái niệm cốt lõi về Nhận diện Đối tượng (Object Detection)	4
1.1.2 Vai trò của thị giác máy tính trong nông nghiệp 4.0	5
1.2 Giới thiệu về mô hình YOLO và YOLOv8.....	6
1.2.1 Lịch sử và Cơ chế Hoạt động Đột phá của Dòng YOLO (You Only Look Once)	6
1.2.2 Kiến trúc và cải tiến của YOLOv8.....	7
1.3 Các công cụ và thư viện lập trình	7
1.3.1 Ngôn ngữ lập trình Python.....	7
1.3.2 Thư viện Xử lý ảnh OpenCV (Computer Vision).....	8
1.3.3 Nền tảng Huấn luyện Google Colab.....	8
1.3.4 Nền tảng Quản lý Vòng đời Dữ liệu Roboflow	9
1.3.5 Framework phát triển Web Streamlit	10
1.3.6 Web framework Flask	10
1.3.7 Thư viện Numpy và PanDas.....	10
1.3.8 Nền tảng Học sâu PyTorch.....	11
1.3.9 Hệ quản trị cơ sở dữ liệu SQLite.....	12
1.3.10 Quản lý mã nguồn Git	12
1.4 Các chỉ số đánh giá hiệu năng mô hình(Model Evaluation Metrics).....	13
CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG	15
2.1 Quy trình Xây dựng Bộ Dữ liệu Huấn luyện (Training Dataset Pipeline)	15
2.1.1 Thu thập và Sàng lọc Dữ liệu Hình ảnh.....	15
2.1.2 Tiền xử lý và gán nhãn dữ liệu với Roboflow	16
2.2 Huấn luyện mô hình YOLOv8.....	18
2.2.1 Thiết lập Môi trường Huấn luyện Chuyên biệt	18
2.2.3 Chiến lược Huấn luyện và Cấu hình Siêu tham số (Hyperparameters)....	18
2.3 Phân tích và thiết kế hệ thống web	19
2.3.1 Phân tích yêu cầu hệ thống	19
2.3.2 Kiến trúc hệ thống	20
2.4 Xây dựng ứng dụng Web nhận diện chuỗi:	21
2.4.1 Kiến trúc Module hóa và Công nghệ Triển khai:	21
2.4.2. Triển khai Lỗi Ứng dụng và Bảo mật (app.py)	25

2.4.3. Xây dựng Module Xử lý AI (utils/detection.py)	25
2.4.4. Xây dựng Giao diện Người dùng (templates/)	26
2.4.5. Định nghĩa API Backend (JSON Endpoint).....	26
CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ.....	28
3.1. Kết quả thực nghiệm và đánh giá.....	28
3.1.1. Phân tích Quá trình Hội tụ qua Hàm Mất mát (Loss Functions)	28
3.1.2. Các chỉ số đánh giá hiệu suất (Performance Metrics) và phân tích ma trận nhầm lẫn.....	29
3.2. Kết quả xây dựng ứng dụng web (WEB APPLICATION RESULTS).....	30
3.2.1. Giao diện người dùng và Trải nghiệm (UI/UX)	30
3.2.2. Chức năng nhận diện qua Tải ảnh (Upload Mode).....	32
3.2.3. Chức năng Camera thời gian thực (Real-time Detection)	33
3.2.4. Các tính năng nâng cao và Quản lý dữ liệu	34
3.3. Đánh giá tổng quát và thảo luận.....	36
3.3.1. Đánh giá trên các kịch bản thực tế.....	36
3.3.2. So sánh với các giải pháp cũ (YOLOv7)	36
3.3.3. Hạn chế tồn tại và khả năng cải thiện.....	37
KẾT LUẬN	38
TÀI LIỆU THAM KHẢO	39

Tóm Tắt

Tên đề tài: Nghiên cứu và xây dựng web trong nhận dạng trái Chuối cho nông nghiệp Việt Nam dựa trên nền tảng AI

Sinh viên thực hiện: Nguyễn Kim Hoàng Đức

Mã số SV: 2151220227 Lớp: 21CT5

1. Mục tiêu

Xây dựng ứng dụng Web giúp tự động nhận diện và kiểm đếm số lượng trái chuối.

Giải quyết vấn đề phân loại thủ công, hỗ trợ nông nghiệp thông minh.

2. Công nghệ cốt lõi

Mô hình: Sử dụng YOLOv8 vì khả năng xử lý thời gian thực và độ chính xác cao.

Ngôn ngữ & Thư viện: Python, PyTorch (Deep Learning), OpenCV (xử lý ảnh), Pillow (xử lý ảnh), Flask (web Framework), Ultralytics (YOLO v8), Pandas (xử lý dữ liệu), NumPy (xử lý dữ liệu).

Công cụ hỗ trợ: Roboflow (gán nhãn và tăng cường dữ liệu), Google Colab (huấn luyện model), Visual Studio Code (build dự án web nhận diện).

3. Quy trình thực hiện

Chuẩn bị dữ liệu: Thu thập ảnh trái chuối và dùng Roboflow để vẽ khung (bounding box) gán nhãn.

Huấn luyện (Training): Chia dữ liệu theo tỷ lệ 70% huấn luyện - 20% kiểm tra - 10% xác thực và chạy 30 epochs trên Google Colab.

Xây dựng Web: Tạo giao diện Web đơn giản cho phép tải ảnh lên hoặc nhận diện qua camera hệ thống sẽ trả về hình ảnh đã được đóng khung nhận diện chuối kèm độ tin cậy.

4. Kết quả dự kiến

Ứng dụng Web hoạt động ổn định, có thể đếm và xác định vị trí trái chuối trong ảnh.

Đạt độ chính xác (mAP) cao, tương tự mức ~87% của dự án mẫu.

5. Kết luận và Hướng phát triển

Kết luận: Đề tài xây dựng thành công hệ thống Web hỗ trợ nhận diện trái chuối, minh chứng cho khả năng ứng dụng YOLOv8 trong nông nghiệp công nghệ cao tại Việt Nam.

Hướng phát triển:

Phát triển thêm tính năng phân loại bệnh trên vỏ trái chuối.

Tối ưu hóa hiệu suất và bảo mật trang web.

Đóng gói thành ứng dụng di động (Mobile App) để tiện lợi hơn cho nông dân.

LỜI NÓI ĐẦU

Trong kỷ nguyên Cách mạng Công nghiệp 4.0, chuyển đổi số trong nông nghiệp đã trở thành yếu tố then chốt, là xu hướng tất yếu để gia tăng năng lực cạnh tranh và nâng cao vị thế của nông sản Việt Nam. Nhận thức sâu sắc rằng Chuối là cây ăn quả chủ lực, nhưng quy trình quản lý hiện tại còn phụ thuộc nhiều vào phương pháp thủ công, tiêu tốn nhân lực và thiếu tính đồng bộ. Từ thực tiễn đó, em đã thực hiện đề tài "Nghiên cứu và xây dựng Web trong nhận dạng trái Chuối cho nông nghiệp Việt Nam dựa trên nền tảng AI". Đề án này là sự kết tinh của việc ứng dụng những kỹ thuật tiên tiến nhất của Trí tuệ nhân tạo (AI) và Thị giác máy tính nhằm tạo ra một công cụ số hóa, tự động hóa, hỗ trợ đắc lực cho người nông dân tối ưu hóa năng suất và chất lượng sản phẩm.

Hành trình từ ý tưởng đến sản phẩm công nghệ hoàn chỉnh là một chặng đường đầy thử thách. Em xin bày tỏ lòng tri ân chân thành và sâu sắc nhất đến thầy Nguyễn Hà Huy Cường, người đã tận tâm định hướng khoa học, giúp em tháo gỡ từng nút thắt kỹ thuật. Sự chỉ dẫn tận tình của Thầy chính là động lực mạnh mẽ nhất. Bên cạnh đó, em xin trân trọng cảm ơn Quý Thầy/Cô thuộc Khoa Công nghệ Thông tin, Trường Đại học Kiến trúc Đà Nẵng. Những kiến thức nền tảng vững chắc và tư duy lập trình hiện đại Thầy/Cô truyền đạt chính là hành trang vô giá để em tiếp cận và làm chủ công nghệ mới trong đề tài này.

Mặc dù đã nỗ lực hết mình với tinh thần làm việc nghiêm túc, do giới hạn về mặt thời gian và kinh nghiệm thực tiễn, đề án chắc chắn khó tránh khỏi những thiếu sót không mong muốn. Em kính mong nhận được sự lượng thứ và những ý kiến đóng góp, phê bình quý báu từ Quý Thầy/Cô. Những phản hồi này không chỉ giúp đề tài được hoàn thiện hơn ở hiện tại, mà còn là nền tảng mở ra những hướng phát triển xa hơn trong tương lai, góp phần thiết thực vào công cuộc số hóa nông nghiệp Việt Nam.

CAM ĐOAN

Tôi xin cam đoan rằng đồ án tốt nghiệp với đề tài “Nghiên cứu và xây dựng Web trong nhận dạng trái Chuối cho nông nghiệp Việt Nam dựa trên nền tảng AI” là công trình nghiên cứu do chính tôi nỗ lực thực hiện dưới sự hướng dẫn khoa học tận tình của Thầy Nguyễn Hà Huy Cường.

Các kết quả được trình bày trong đồ án, bao gồm việc xây dựng bộ dữ liệu hình ảnh trái chuối, quá trình huấn luyện và tinh chỉnh mô hình AI, cũng như việc thiết kế và lập trình ứng dụng Web, đều là thành quả của quá trình lao động nghiêm túc, trung thực và độc lập của bản thân. Tôi xin khẳng định rằng các số liệu và kết quả nghiên cứu trong đồ án là hoàn toàn mới và chưa từng được công bố trong bất kỳ công trình nào khác.

Đối với các tài liệu tham khảo, các cơ sở lý thuyết hay kết quả nghiên cứu của các tác giả khác được sử dụng để làm cơ sở phân tích, so sánh trong đồ án, tôi đều đã thực hiện trích dẫn nguồn gốc rõ ràng và tuân thủ nghiêm ngặt các nguyên tắc về quyền sở hữu trí tuệ.

Nếu phát hiện có bất kỳ sự gian lận nào, tôi xin hoàn toàn chịu trách nhiệm trước Hội đồng bảo vệ đồ án, Khoa Công nghệ Thông tin và Trường Đại học Kiến trúc Đà Nẵng về lời cam đoan của mình.

Sinh viên thực hiện

{Chữ ký, họ và tên sinh viên}

DANH MỤC BẢNG

Bảng 2.1 Dữ liệu chuẩn hóa để nhận diện 3 đối tượng chính	16
Bảng 2.2 Chiến thuật tăng cường độ nhận diện của hình ảnh.....	17
Bảng 2.3 Cấu hình tham số cốt lõi.....	19
Bảng 2.4 Yêu cầu phi chức năng	20
Bảng 2.5 Các lớp xử lý nghiệp vụ chính	21
Bảng 2.6 Các module chính chức năng.....	22
Bảng 2.7 Các route chính.....	25
Bảng 2.8 Các API chính	27
Bảng 3.1 Kết quả đánh giá hiệu suất	29

DANH MỤC HÌNH ẢNH

Hình 2.2 kiến trúc hệ thống mô phỏng	20
Hình 2.3 một số code demo giao diện	23
Hình 2.4 một số code cấu hình hiển thị và được chạy qua cổng nào	23
Hình 2.5 Một số code quản lý dữ liệu tạm thời database.py	24
Hình 3.1 Kết quả đánh giá mô hình YOLO v8 sau khi đã huấn luyện	28
Hình 3.2 Kết quả đánh giá hiệu suất qua Google colab	30
Hình 3.3 Demo giao diện đăng nhập	31
Hình 3.4 Demo giao diện đăng ký.....	31
Hình 3.5 Demo giao diện web nhận diện qua upload ảnh.....	32
Hình 3.6 Demo chức năng nhận diện qua file ảnh.....	33
Hình 3.7 Demo chức năng nhận diện qua camera real-time.....	34
Hình 3.8 Demo phân lịch sử nhận diện	35
Hình 3.9 Demo file lịch sử nhận diện được xuất ra excel	36

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

KÝ HIỆU:

.....

.....

.....

.....

.....

.....

CHỮ VIẾT TẮT:

.....

.....

.....

.....

Ghi chú:

- Ký hiệu: mỗi mục ký hiệu gồm ký hiệu và phần tên gọi, diễn giải ký hiệu.
- Cụm từ viết tắt là các chữ cái và các ký hiệu thay chữ được viết liền nhau, để thay cho một cụm từ có nghĩa, thường được lặp nhiều lần trong đồ án.

MỞ ĐẦU

1. Mục đích thực hiện đề tài

Trong kỷ nguyên Nông nghiệp Chính xác (Precision Agriculture), việc tự động hóa quy trình giám sát và kiểm kê là chìa khóa chiến lược để tối ưu hóa năng suất và nâng cao giá trị của nông sản Việt Nam trên thị trường toàn cầu. Trái Chuối là mặt hàng xuất khẩu chủ lực, nhưng công tác kiểm kê, phân loại chất lượng tại vườn hiện nay vẫn phụ thuộc lớn vào kinh nghiệm cá nhân và lao động thủ công. Điều này tạo ra sự thiếu đồng nhất trong đánh giá và gia tăng đáng kể chi phí vận hành.

Mục đích cốt lõi của đề tài là ứng dụng đột phá các kỹ thuật Trí tuệ nhân tạo (AI) và Thị giác máy tính (Computer Vision) để giải quyết triệt để bài toán này. Cụ thể, đề tài hướng đến việc xây dựng một hệ thống thông minh, có khả năng tự động nhận diện, định vị và đếm số lượng trái chuối ngay lập tức (real-time) thông qua dữ liệu hình ảnh và camera, từ đó tối ưu hóa quy trình canh tác, quản lý mùa vụ và thu hoạch.

2. Mục tiêu đề tài

Đề tài tập trung vào việc đạt được các mục tiêu cụ thể, cân bằng giữa khía cạnh lý thuyết chuyên sâu và ứng dụng thực tiễn:

Về Cơ sở Lý thuyết:

- Nghiên cứu Tổng quan: Phân tích sâu về bài toán Nhận diện Đối tượng (Object Detection) trong bối cảnh nông nghiệp số hóa.

- Khảo sát Mô hình Tiên tiến (State-of-the-art): Nghiên cứu chi tiết về kiến trúc, nguyên lý hoạt động và cơ chế anchor-free đột phá của mô hình YOLOv8 (Ultralytics), mô hình được chọn vì khả năng cân bằng vượt trội giữa tốc độ suy luận và độ chính xác nhận diện so với các phiên bản tiền nhiệm.

Về Ứng dụng Thực tiễn:

- Xây dựng Bộ Dữ liệu Chuẩn hóa: Thực hiện quy trình thu thập, tiền xử lý và gán nhãn (labeling) dữ liệu hình ảnh trái chuối Việt Nam, đảm bảo tính đa dạng và đại diện.

- Huấn luyện Mô hình Tối ưu: Huấn luyện, tinh chỉnh và tối ưu hóa mô hình YOLOv8 để đạt được hiệu suất nhận diện cao nhất dưới các điều kiện môi trường thực tế khác nhau (ánh sáng, góc chụp).

- Phát triển Ứng dụng Web: Xây dựng ứng dụng Web hoàn chỉnh, tích hợp mô hình AI đã huấn luyện, cho phép người dùng tương tác trực quan bằng cách tải ảnh hoặc sử dụng luồng dữ liệu camera real-time để nhận kết quả ngay lập tức.

3. Phạm vi và Đối tượng Nghiên cứu

Đối tượng Nghiên cứu:

- Các thuật toán cốt lõi trong Deep Learning (Mạng nơ-ron tích chập - CNN) ứng dụng cho Nhận diện Đối tượng.

- Cấu trúc tinh gọn và kỹ thuật tối ưu hóa của mô hình YOLOv8.

- Các công cụ và thư viện phát triển: Ngôn ngữ Python, framework PyTorch, thư viện xử lý ảnh (OpenCV, Pillow), Web framework (Flask), cùng các thư viện khoa học dữ liệu (NumPy, Pandas, Ultralytics).

Phạm vi Nghiên cứu:

- Dữ liệu: Tập trung vào hình ảnh trái chuối ở các giai đoạn phát triển khác nhau: chưa chín (unripe), chín (ripe), và chín quá (overripe). Dữ liệu được thu thập đa dạng từ Internet và ảnh thực tế.

- Công nghệ: Huấn luyện mô hình được thực hiện trên nền tảng Google Colab (sử dụng GPU). Ứng dụng Web được xây dựng ở mức độ Demo (Proof- of-Concept) để kiểm chứng và minh họa khả năng hoạt động của thuật toán.

4. Phương pháp nghiên cứu

Đề tài vận dụng linh hoạt và kết hợp chặt chẽ hai phương pháp nghiên cứu chính:

Phương pháp Nghiên cứu Lý thuyết:

- Tổng hợp và Phân tích chuyên sâu các tài liệu hàn lâm về Deep Learning và Computer Vision.

- Nghiên cứu tài liệu kỹ thuật chính thức (Documentation) của Ultralytics YOLOv8 để nắm vững cơ chế huấn luyện, suy luận và tinh chỉnh tham số mô hình.

Phương pháp Thực nghiệm:

- Xử lý và Tăng cường Dữ liệu: Sử dụng nền tảng Roboflow để thực hiện công đoạn gán nhãn chính xác (bounding box). Áp dụng các kỹ thuật Tăng cường Dữ liệu (Data Augmentation) như xoay, lật, chỉnh độ sáng để làm giàu tập mẫu, qua đó tăng tính tổng quát hóa và giảm thiểu hiện tượng quá khớp (overfitting).

- Huấn luyện và Đánh giá Định lượng: Tiến hành huấn luyện mô hình YOLOv8 và đánh giá hiệu quả thông qua các chỉ số định lượng bắt buộc: Precision (Độ chính xác), Recall (Độ nhạy), và mAP (Mean Average Precision).

- Phát triển Ứng dụng: Lập trình ứng dụng Web bằng Python, tích hợp mô hình đã huấn luyện để tạo giao diện người dùng (UI) trực quan và thân thiện.

5. Cấu trúc của đồ án tốt nghiệp

Ngoài phần Mở đầu, Kết luận và Tài liệu tham khảo, Đồ án được bố cục thành 3 chương chính, trình bày theo một logic nghiên cứu chặt chẽ:

- Chương 1: Cơ sở Lý thuyết và Công nghệ: Trình bày bối cảnh bài toán nhận diện đối tượng trong nông nghiệp, giới thiệu tổng quan về Deep Learning, và phân tích sâu kiến trúc cũng như ưu điểm của mô hình YOLOv8.
- Chương 2: Phân tích, Xây dựng Bộ Dữ liệu và Thiết kế Hệ thống: Mô tả chi tiết quy trình thu thập, chuẩn hóa, gán nhãn bộ dữ liệu trái chuối; trình bày cấu hình huấn luyện mô hình; và thiết kế kiến trúc kỹ thuật của ứng dụng Web.
- Chương 3: Kết quả Thực nghiệm và Đánh giá Hiệu năng: Trình bày kết quả huấn luyện (biểu đồ Loss, các chỉ số mAP), demo giao diện ứng dụng

Web với kết quả nhận diện trực quan, nhận xét về ưu nhược điểm, và đề xuất hướng phát triển tiếp theo của đề tài.

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT VÀ TỔNG QUAN CÔNG NGHỆ

1.1 Tổng quan về Nhận diện Đối tượng (Object Detection) và Ứng dụng trong Nông nghiệp

1.1.2 Khái niệm cốt lõi về Nhận diện Đối tượng (Object Detection)

a. Định nghĩa và Bản chất vấn đề

Nhận diện đối tượng (Object Detection) là một bài toán nền tảng và thách thức bậc nhất trong lĩnh vực Thị giác máy tính (Computer Vision). Mục tiêu của bài toán là mô phỏng khả năng tri giác của con người để máy tính có thể tự động hiểu được ngữ nghĩa của một hình ảnh ở mức độ chi tiết.

Cụ thể, nhiệm vụ của một mô hình nhận diện đối tượng là phát hiện các thực thể (instances) thuộc các lớp ngữ nghĩa định trước (ví dụ: người, xe, hoặc trong phạm vi đề tài này là trái chuối) xuất hiện trong hình ảnh kỹ thuật số hoặc video.

Sự khác biệt cơ bản giữa Nhận diện đối tượng và bài toán Phân loại ảnh (Image Classification) truyền thống nằm ở độ phức tạp của đầu ra:

- Phân loại ảnh (Image Classification): Chỉ gán một nhãn duy nhất cho toàn bộ bức ảnh (Ví dụ: "Đây là ảnh quả chuối").
- Nhận diện đối tượng (Object Detection): Phải xử lý đồng thời nhiều đối tượng trong cùng một khung hình, xác định chính xác vị trí và danh tính của từng đối tượng đó.

b. Hai nhiệm vụ song song: Định vị và Phân loại

Để giải quyết bài toán này, mô hình học sâu phải thực hiện đồng thời hai tác vụ (Multi-task learning):

1. Định vị (Localization - Trả lời câu hỏi "Ở đâu?") Mô hình phải xác định vùng không gian chứa đối tượng. Kết quả định vị được biểu diễn dưới dạng một Hộp bao giới hạn (Bounding Box) hình chữ nhật. Về mặt toán học, một Bounding Box thường được định nghĩa bởi một vector 4 tham số:

$$B = (x, y, w, h)$$

Trong đó:

- (x, y) : Là tọa độ tâm của hộp bao (hoặc tọa độ góc trên bên trái tùy kiến trúc mô hình).
- (w, h) : Là chiều rộng (width) và chiều cao (height) của hộp bao tương ứng với kích thước đối tượng.

2. Phân loại (Classification - Trả lời câu hỏi "Cái gì?")

Song song với việc tìm vị trí, mô hình phải gán nhãn lớp (Class Label) cho nội dung bên trong hộp bao đó, kèm theo một điểm số tin cậy (Confidence Score).

- Confidence Score (Pc): Là xác suất dự đoán (từ 0 đến 1) thể hiện độ chắc chắn của mô hình rằng đối tượng trong khung hình thuộc về lớp đó.
- Trong ngữ cảnh đề tài: Hệ thống tập trung phân loại vào 3 nhãn trạng thái sinh học của trái chuối để phục vụ đánh giá chất lượng:
 - Unripe (Chuối xanh)
 - Ripe (Chuối chín)
 - Overripe (Chuối chín quá/hồng)

c. Vai trò của Học sâu (Deep Learning)

Trước đây, các phương pháp truyền thống dựa trên đặc trưng thủ công (hand-crafted features) thường gặp khó khăn khi đối tượng bị che khuất, thay đổi tỷ lệ hoặc điều kiện ánh sáng phức tạp. Sự ra đời của các kiến trúc Mạng Nơ-ron Tích chập (Convolutional Neural Networks - CNN) đã tạo ra bước đột phá.

Các mô hình hiện đại (như họ YOLO, SSD hay Faster R-CNN) có khả năng tự động trích xuất các đặc trưng phân cấp từ thấp đến cao (từ đường nét, góc cạnh đến hình dạng quả chuối), giúp hệ thống đạt được độ chính xác cao (High Precision) và khả năng xử lý thời gian thực (Real-time), tạo tiền đề vững chắc cho việc ứng dụng vào tự động hóa nông nghiệp.

1.1.3 Vai trò của thị giác máy tính trong nông nghiệp 4.0

Trong bối cảnh chuyển đổi số và phát triển Nông nghiệp Thông minh (Smart Agriculture), công nghệ Nhận diện đối tượng (Object Detection) dựa trên Thị giác máy tính đang trở thành yếu tố chiến lược, quyết định trực tiếp đến năng lực cạnh tranh của ngành. Đối với các loại cây ăn quả chủ lực của Việt Nam như cây chuối, việc ứng dụng công nghệ này không chỉ đơn thuần là sự thay thế sức lao động thủ công mà còn giải quyết triệt để ba bài toán cấp thiết về quản lý, chất lượng và thương mại hóa.

Đầu tiên, công nghệ này đóng vai trò then chốt trong việc tối ưu hóa quản lý mùa vụ và ước tính sản lượng. Thay thế hoàn toàn cho phương pháp đếm thủ công hoặc lấy mẫu xác suất truyền thống vốn nhiều sai số, các mô hình học sâu hiện đại cho phép nhận diện và đếm chính xác số lượng nông sản trên diện rộng. Dữ liệu định lượng thu được là cơ sở tin cậy cho bài toán ước tính năng suất (Yield Estimation), giúp các nhà vườn và doanh nghiệp chủ động lập kế hoạch nguồn lực, tối ưu hóa hệ thống logistics và chuẩn bị tài chính kỹ lưỡng ngay từ trước khi bước vào vụ thu hoạch.

Kế đến, thị giác máy tính nâng cao năng lực kiểm soát chất lượng và đảm bảo thu hoạch chính xác. Thông qua việc phân tích màu sắc, hình dạng và các đặc điểm ngoại quan mà không cần tiếp xúc vật lý, hệ thống có khả năng phân loại trái cây theo từng giai đoạn độ chín cụ thể (xanh, chín, hoặc chín quá). Khả năng nhận diện theo thời gian thực này hỗ trợ người nông dân ra quyết định thu hoạch tại đúng "điểm rơi" tối ưu nhất cho từng mục

đích tiêu thụ—dù là vận chuyển đường dài cho xuất khẩu hay tiêu thụ ngay tại thị trường nội địa—từ đó giảm thiểu tối đa tỷ lệ hao hụt sau thu hoạch.

Cuối cùng, việc ứng dụng AI là chìa khóa để nâng cao tính đồng nhất cho nông sản xuất khẩu. Bằng cách loại bỏ sự phụ thuộc vào đánh giá chủ quan và kinh nghiệm không đồng đều của mắt người, công nghệ tự động hóa đảm bảo quá trình phân loại và đóng gói diễn ra với độ chính xác và ổn định cao. Sự đồng nhất về kích cỡ và chất lượng không chỉ giúp nông sản Việt Nam đáp ứng được các bộ tiêu chuẩn khắt khe của thị trường quốc tế mà còn góp phần quan trọng trong việc nâng cao giá trị thương phẩm và vị thế của nông nghiệp quốc gia trên chuỗi cung ứng toàn cầu.

1.2 Giới thiệu về mô hình YOLO và YOLOv8

1.2.1 Lịch sử và Cơ chế Hoạt động Đột phá của Dòng YOLO (You Only Look Once)

YOLO (You Only Look Once) đại diện cho một bước tiến mang tính cách mạng trong lĩnh vực nhận diện đối tượng, nổi bật với cơ chế "một bước" (Single-Stage) tiên phong. Phương pháp này khác biệt hoàn toàn so với các mô hình truyền thống hai giai đoạn (Two-Stage) như dòng R-CNN, vốn phải trải qua hai bước phức tạp là tạo vùng đề xuất đối tượng (Region Proposal) rồi mới tiến hành phân loại.

YOLO loại bỏ sự phức tạp này bằng cách thực hiện toàn bộ quy trình nhận diện – bao gồm định vị (Localization) và phân loại (Classification) – chỉ trong một lần chạy mạng duy nhất, cho phép đạt được tốc độ xử lý cực nhanh, đáp ứng yêu cầu của các ứng dụng thời gian thực (Real-time).

Cơ chế hoạt động cốt lõi của YOLO có thể được tổng kết như sau:

- Hệ thống Lưới Phân chia (Grid System): Ảnh đầu vào được chia thành một hệ thống các ô lưới (grid cells). Mỗi ô lưới được chỉ định trách nhiệm dự đoán các đối tượng có tọa độ tâm nằm trong phạm vi của ô đó.

- Dự đoán Hộp Bao và Độ Tin cậy: Mỗi ô dự đoán một số lượng hộp bao (Bounding Box) nhất định, cùng với các tham số:

- Tọa độ và kích thước hộp (x, y, w, h).
- Độ tin cậy (Confidence Score): Mức độ chắc chắn rằng hộp đó chứa một đối tượng.

- Dự đoán Xác suất Lớp (Class Probability): Mô hình đồng thời dự đoán xác suất đối tượng đó thuộc về lớp nào (ví dụ: Chuối xanh, Chuối chín, Chuối chín quá).

Sự kết hợp giữa tốc độ và khả năng thực hiện đồng thời các tác vụ đã đưa YOLO trở thành tiêu chuẩn vàng cho các hệ thống Thị giác Máy tính hoạt động trong môi trường thực tế.

1.2.2 Kiến trúc và cải tiến của YOLOv8

Trong đề tài nghiên cứu này, chúng tôi lựa chọn sử dụng YOLOv8 – phiên bản tiên tiến nhất được phát triển bởi Ultralytics (ra mắt vào năm 2023). So với các phiên bản tiền nhiệm, YOLOv8 mang đến những cải tiến vượt trội về cả kiến trúc và cơ chế huấn luyện, giúp tối ưu hóa hiệu suất nhận diện trong môi trường nông nghiệp thực tế:

- Kiến trúc Anchor-free (Không Khung Neo): YOLOv8 loại bỏ hoàn toàn việc sử dụng các khung neo (anchor boxes) định sẵn. Việc chuyển sang cơ chế anchor-free giúp mô hình đạt được tính linh hoạt cao hơn trong việc nhận diện các đối tượng có kích thước và tỷ lệ khung hình bất thường (ví dụ: các trái chuối nhỏ hoặc bị che khuất một phần). Đồng thời, điều này giúp đơn giản hóa quy trình huấn luyện và giảm đáng kể số lượng tham số cần tinh chỉnh.
- Khối C2f (Cross Stage Partial Bottleneck with 2 convolutions): Khối C2f được thiết kế để thay thế cho khối C3 trong các phiên bản trước đó. Cải tiến này không chỉ giúp cải thiện luồng gradient mà còn tăng cường khả năng trích xuất đặc trưng (feature extraction). Điều này đặc biệt quan trọng, giúp mô hình học được các chi tiết tinh vi như màu sắc vỏ hoặc hình thái cuống chuối một cách hiệu quả hơn.
- Hàm Mất mát (Loss Function) Hiện đại: YOLOv8 tích hợp các hàm mất mát tiên tiến như VFL Loss (VariFocal Loss) và DFL Loss (Distribution Focal Loss). Sự kết hợp này mang lại khả năng định vị hộp bao (bounding box) chính xác hơn, đặc biệt hữu ích trong các tình huống phức tạp như khi nhiều trái chuối nằm chồng chéo hoặc quá gần nhau, đảm bảo độ tin cậy cao cho việc đếm và phân loại.

Những cải tiến này là lý do then chốt khiến YOLOv8 trở thành mô hình tiêu chuẩn hiện đại (State-of-the-Art), mang lại sự cân bằng tối ưu giữa tốc độ suy luận và độ chính xác – hai yếu tố cực kỳ quan trọng đối với các ứng dụng giám sát thời gian thực trong Nông nghiệp 4.0.

1.3 Các công cụ và thư viện lập trình

1.3.1 Ngôn ngữ lập trình Python

Trong khuôn khổ đề án này, Python được lựa chọn làm ngôn ngữ phát triển chủ đạo nhờ vị thế độc tôn của nó trong lĩnh vực Khoa học Dữ liệu và Trí tuệ Nhân tạo (AI). Là một ngôn ngữ lập trình bậc cao (High-level programming language) với cú pháp tường minh và cộng đồng hỗ trợ khổng lồ, Python đóng vai trò là "xương sống" liên kết toàn bộ kiến trúc hệ thống, đảm nhiệm ba nhiệm vụ chiến lược:

- Xử lý và Chuẩn hóa Dữ liệu: Python cung cấp khả năng thao tác mạnh mẽ với các cấu trúc dữ liệu phức tạp. Thông qua thư viện NumPy, các hình ảnh đầu vào được chuyển đổi thành các mảng đa chiều (multidimensional arrays) để thực hiện các phép tính số học và ma trận tốc độ cao – bước tiền đề bắt buộc cho mọi thuật toán thị giác máy tính.
- Tương tác và Vận hành Mô hình AI: Python đóng vai trò là lớp vỏ (wrapper) giao tiếp trực tiếp với các framework học sâu như PyTorch. Nó cho phép gọi các API của YOLOv8 để thực hiện quy trình huấn luyện (Training) trên tập dữ liệu tùy chỉnh và triển khai suy luận (Inference) thời gian thực, đồng thời đơn giản hóa việc tinh chỉnh các tham số siêu hình (hyperparameters).

- Phát triển Ứng dụng Backend: Python là nền tảng để xây dựng máy chủ ứng dụng (thông qua Flask), xử lý logic nghiệp vụ và quản lý các luồng dữ liệu vào/ra giữa người dùng và mô hình AI.

Bên cạnh đó, việc quản lý tài nguyên hệ thống và tổ chức tập dữ liệu lớn (dataset) được thực hiện hiệu quả nhờ các thư viện tiêu chuẩn như os và shutil, giúp tự động hóa các thao tác với tệp tin và thư mục trong quá trình chuẩn bị dữ liệu.

1.3.2 Thư viện Xử lý ảnh OpenCV (Computer Vision)

OpenCV (Open Source Computer Vision Library) được lựa chọn làm công cụ nòng cốt cho các tác vụ xử lý ảnh trong đề án. Đây là thư viện mã nguồn mở tiêu chuẩn công nghiệp, cung cấp hơn 2500 thuật toán tối ưu hóa cho thị giác máy tính. Trong kiến trúc hệ thống, module cv2 của Python đóng vai trò là cầu nối kỹ thuật giữa dữ liệu hình ảnh thô và mô hình AI, đảm nhiệm ba chức năng trọng yếu:

- Thu thập và Tiền xử lý Luồng Video (Video Stream Acquisition): OpenCV chịu trách nhiệm thiết lập kết nối trực tiếp với thiết bị Webcam để thu thập dữ liệu hình ảnh theo thời gian thực (Real-time). Các khung hình (frames) được trích xuất liên tục từ luồng video, sau đó được giải mã thành các ma trận số học để chuẩn bị cho quá trình suy luận của mô hình.
- Đồng bộ hóa Không gian màu (Color Space Conversion): Một thách thức kỹ thuật đặc thù là OpenCV mặc định đọc ảnh dưới định dạng không gian màu BGR (Blue-Green-Red), trong khi các mô hình Học sâu và trình duyệt Web hiển thị chuẩn RGB (Red-Green-Blue). Do đó, OpenCV được sử dụng để chuyển đổi không gian màu cho từng khung hình, đảm bảo tính nhất quán của dữ liệu đầu vào cho AI và độ trung thực của màu sắc trái chuỗi khi hiển thị trên giao diện người dùng.
- Trực quan hóa Kết quả (Result Visualization): Ở giai đoạn hậu xử lý, OpenCV cung cấp các công cụ đồ họa để "vẽ" kết quả phân tích lên hình ảnh gốc. Cụ thể, thư viện thực hiện vẽ các hộp bao giới hạn (Bounding Boxes) định vị trái chuỗi, đồng thời gán nhãn phân loại (như *Unripe*, *Ripe*) kèm theo chỉ số độ tin cậy (Confidence Score). Điều này giúp chuyển đổi các con số thống kê khô khan thành thông tin trực quan, dễ hiểu cho người sử dụng.

1.3.3 Nền tảng Huấn luyện Google Colab

Để giải quyết thách thức về yêu cầu phần cứng chuyên dụng cho việc huấn luyện các mô hình Học sâu (Deep Learning), đề án sử dụng Google Colaboratory (Colab) làm môi trường phát triển và tính toán chính. Đây là nền tảng dựa trên đám mây được cung cấp bởi Google Research, hoạt động trên cơ sở Jupyter Notebook, đóng vai trò chiến lược trong việc hiện thực hóa mô hình YOLOv8 với hai ưu điểm cốt lõi:

- Tăng tốc Tính toán với Phần cứng Chuyên dụng (Hardware Acceleration): Việc huấn luyện mạng nơ-ron tích chập sâu như YOLOv8 đòi hỏi thực hiện hàng triệu phép tính ma trận song song, điều mà CPU thông thường trên máy tính cá nhân khó đáp ứng được. Google Colab cung cấp quyền truy cập miễn phí vào các đơn vị xử lý đồ họa (GPU) hiệu năng cao, điển hình là dòng NVIDIA Tesla T4 (với 16GB

VRAM). Sức mạnh tính toán này giúp giảm thời gian huấn luyện mô hình từ hàng chục giờ xuống chỉ còn vài giờ, đồng thời cho phép thử nghiệm với kích thước lô (batch size) lớn hơn để tăng độ ổn định của thuật toán.

- Tích hợp Hệ sinh thái và Lưu trữ Bền vững: Môi trường Colab được tích hợp liền mạch với Google Drive, tạo thành một hệ thống lưu trữ đám mây đồng bộ. Trong đồ án này, Google Drive đóng vai trò như một ổ cứng vĩnh viễn (persistent storage) để lưu trữ bộ dữ liệu huấn luyện (Dataset) và quan trọng nhất là bảo toàn các tệp trọng số mô hình (best.pt, last.pt) sau mỗi chu kỳ huấn luyện (epoch). Điều này đảm bảo quá trình nghiên cứu không bị gián đoạn hay mất mát dữ liệu ngay cả khi phiên làm việc (runtime) trên đám mây kết thúc.
- Môi trường "Zero-configuration": Colab được cài đặt sẵn hầu hết các thư viện khoa học dữ liệu và AI thiết yếu (như PyTorch, NumPy, Pandas, OpenCV), giúp giảm thiểu tối đa thời gian thiết lập môi trường và khắc phục các lỗi xung đột phần mềm (dependency conflicts), cho phép nhóm tập trung hoàn toàn vào việc tối ưu hóa thuật toán và dữ liệu.

1.3.4 Nền tảng Quản lý Vòng đời Dữ liệu Roboflow

Trong quy trình xây dựng hệ thống Thị giác máy tính, chất lượng dữ liệu đầu vào quyết định trực tiếp đến hiệu suất mô hình (nguyên lý "Garbage In, Garbage Out"). Để đảm bảo điều này, đồ án sử dụng Roboflow – một nền tảng MLOps (Machine Learning Operations) toàn diện dựa trên đám mây, đóng vai trò trung tâm trong việc quản lý, xử lý và phiên bản hóa bộ dữ liệu.

Roboflow giải quyết hai bài toán cốt lõi trong quá trình chuẩn bị dữ liệu:

- Gán nhãn dữ liệu (Data Annotation) Chính xác và Đồng nhất: Roboflow cung cấp giao diện đồ họa trực quan, hỗ trợ nhóm nghiên cứu thực hiện vẽ các hộp bao (Bounding Box) bao quanh đối tượng trái chuỗi và gán nhãn phân loại (Xanh/Chín/Hư) một cách nhanh chóng. Điểm mạnh của nền tảng này là khả năng tự động chuẩn hóa định dạng dữ liệu đầu ra (ví dụ: chuyển đổi sang định dạng .txt tiêu chuẩn của YOLO) và kiểm tra sức khỏe dữ liệu (Health Check) để loại bỏ các ảnh lỗi hoặc thiếu nhãn trước khi đưa vào huấn luyện.
- Tăng cường dữ liệu (Data Augmentation) và Mở rộng tập mẫu: Để khắc phục hạn chế về số lượng ảnh thu thập thực tế và giảm thiểu hiện tượng Quá khớp (Overfitting), chức năng Tăng cường dữ liệu của Roboflow được áp dụng triệt để. Hệ thống tự động sinh ra các biến thể nhân tạo từ ảnh gốc thông qua các kỹ thuật biến đổi ngẫu nhiên như:
 - Biến đổi hình học: Xoay (Rotation), lật (Flip), cắt xén (Crop) – mô phỏng các góc độ chụp khác nhau của buồng chuỗi trong thực tế.
 - Biến đổi cấp độ pixel: Thay đổi độ sáng (Brightness), độ tương phản (Contrast) hoặc thêm nhiễu (Noise) – mô phỏng các điều kiện ánh sáng môi trường đa dạng (năng gắt, bóng râm, chiều tối).

Nhờ đó, mô hình học được các đặc trưng mang tính tổng quát (Generalization), giúp hệ thống hoạt động ổn định và bền vững hơn khi triển khai trong môi trường thực tế.

1.3.5 Framework phát triển Web Streamlit

Streamlit là một framework mã nguồn mở cho phép xây dựng các ứng dụng web khoa học dữ liệu nhanh chóng bằng Python. Streamlit được chọn để xây dựng giao diện người dùng (Frontend) vì:

- Tương tác trực quan: Hỗ trợ các widget như nút tải ảnh (file_uploader), thanh trượt (slider) để chỉnh ngưỡng confidence, và hiển thị ảnh/video trực tiếp.
- Tích hợp AI dễ dàng: Cho phép nhúng code xử lý của YOLOv8 trực tiếp vào ứng dụng mà không cần xây dựng API backend phức tạp.

1.3.6 Web framework Flask

Để xây dựng phần Backend (máy chủ ứng dụng), đề án lựa chọn Flask – một vi khung (micro-framework) mã nguồn mở được viết bằng Python. Khác với các framework nguyên khối (như Django), Flask nổi bật với kiến trúc tinh gọn, không áp đặt các thư viện phụ trợ, cho phép nhóm phát triển linh hoạt tùy biến các thành phần để tối ưu hóa hiệu năng cho các tác vụ xử lý AI nặng.

Trong kiến trúc tổng thể của hệ thống, Flask đóng vai trò là "trạm điều phối trung tâm" (Orchestrator) với ba chức năng nhiệm vụ cụ thể:

- Thiết kế API RESTful (API Endpoints): Flask chịu trách nhiệm định nghĩa và quản lý các điểm đầu cuối (Endpoints) để giao tiếp với giao diện người dùng (Frontend).
 - Các phương thức HTTP như POST được sử dụng để tiếp nhận dữ liệu ảnh tải lên từ người dùng.
 - Phương thức GET dùng để truy xuất lịch sử hoặc báo cáo.
- Triển khai Mô hình AI (Model Serving): Đây là chức năng quan trọng nhất. Flask giữ vai trò là môi trường thời gian chạy (Runtime Environment) để tải và lưu trữ trọng số mô hình YOLOv8 (.pt) trong bộ nhớ RAM. Khi có yêu cầu, Flask thực hiện tiền xử lý dữ liệu đầu vào, chuyển giao cho mô hình thực hiện suy luận (Inference), sau đó nhận lại kết quả.
- Xử lý Phản hồi Đa định dạng (Response Formatting): Tùy thuộc vào loại yêu cầu, Flask xử lý và trả về kết quả theo hai định dạng chuyên biệt:
 - Dạng JSON: Đối với tác vụ upload ảnh tĩnh, hệ thống trả về dữ liệu cấu trúc gồm nhãn (Label), độ tin cậy (Confidence) và tọa độ hộp bao để Web hiển thị thông tin chi tiết.
 - Dạng Multipart Stream: Đối với tác vụ Webcam thời gian thực, Flask sử dụng kỹ thuật *Response Streaming* để truyền tải liên tục các khung hình đã được vẽ bounding box về trình duyệt dưới dạng luồng video (Motion JPEG), đảm bảo độ trễ thấp nhất.

1.3.7 Thư viện Numpy và Pandas

Trong các ứng dụng Trí tuệ nhân tạo, việc xử lý dữ liệu thô thành thông tin có ý nghĩa đòi hỏi các công cụ tính toán số học hiệu năng cao. Đề án sử dụng bộ đôi thư viện NumPy và Pandas để giải quyết triệt để bài toán thao tác dữ liệu từ cấp độ điểm ảnh (pixel) đến cấp độ thống kê quản trị.

a. NumPy (Numerical Python) – Xử lý Toán học và Ma trận

NumPy là thư viện cốt lõi cho tính toán khoa học trong Python, cung cấp đối tượng mảng đa chiều (nd-array) với hiệu suất vượt trội so với danh sách (list) truyền thống. Trong kiến trúc hệ thống, NumPy đóng vai trò nền tảng cho việc "giao tiếp" giữa dữ liệu hình ảnh và mô hình YOLOv8:

- **Biểu diễn dữ liệu hình ảnh:** Mọi bức ảnh kỹ thuật số thu được từ Webcam đều được NumPy chuyển đổi thành các ma trận số học 3 chiều (Chiều cao $\times$ Chiều rộng $\times$ Kênh màu). Đây là định dạng dữ liệu duy nhất mà mô hình AI có thể "hiểu" và xử lý.
- **Xử lý Tensor đầu ra:** Kết quả trả về từ YOLOv8 thường là các Tensor (mảng nhiều chiều) chứa hàng nghìn tham số thô. NumPy được sử dụng để thực hiện các phép toán đại số tuyến tính nhằm lọc bỏ các dự đoán yếu, tính toán tọa độ hộp bao (Bounding Box) và chỉ số giao thoa (IoU) với tốc độ cực nhanh.

b. Pandas – Quản lý và Phân tích Dữ liệu có cấu trúc

Nếu NumPy làm việc với các con số thô, thì Pandas là công cụ chuyên biệt để thao tác với dữ liệu dạng bảng (Tabular Data) thông qua cấu trúc DataFrame. Trong đồ án, Pandas chịu trách nhiệm cho module "Báo cáo và Thống kê":

- **Lưu trữ và Truy vấn Log:** Dữ liệu lịch sử nhận diện (thời gian, loại chuỗi, độ chính xác) được Pandas tổ chức thành các bảng dữ liệu có cấu trúc chặt chẽ, giúp dễ dàng truy vấn và lọc thông tin theo ngày tháng hoặc kết quả.
- **Tổng hợp và Xuất báo cáo:** Pandas cung cấp các hàm thống kê mạnh mẽ để tự động tính toán tổng sản lượng, tỷ lệ chuỗi chín/hư hỏng trong ngày. Đặc biệt, khả năng tương thích tuyệt vời của thư viện này cho phép xuất dữ liệu trực tiếp ra các định dạng phổ biến như Excel (.xlsx) hoặc CSV, phục vụ nhu cầu lưu trữ và quản lý của người dùng cuối.

1.3.8 Nền tảng Học sâu PyTorch

Được phát triển bởi phòng thí nghiệm Facebook AI Research (FAIR), PyTorch được lựa chọn làm "động cơ" vận hành (Inference Engine) cốt lõi cho toàn bộ hệ thống nhận diện trong đồ án. Đây là thư viện mã nguồn mở tối ưu hóa cho các bài toán Thị giác máy tính, cung cấp cấu trúc dữ liệu Tensor (tương tự như mảng đa chiều của NumPy) nhưng có khả năng tận dụng triệt để sức mạnh tính toán song song của GPU (Graphics Processing Unit).

Trong phạm vi đề tài, vai trò của PyTorch không chỉ dừng lại ở môi trường huấn luyện mô hình YOLOv8 mà còn đảm nhiệm các tác vụ trọng yếu trong giai đoạn triển khai thực tế:

- **Quản lý Trọng số (Weights Loading):** Chịu trách nhiệm tải và khởi tạo các tham số mô hình đã được huấn luyện (best.pt) vào bộ nhớ, đảm bảo tính toàn vẹn của kiến trúc mạng nơ-ron.
- **Suy luận Thời gian thực (Real-time Inference):** Thực hiện hàng triệu phép tính ma trận phức tạp trong tích tắc để xử lý luồng dữ liệu hình ảnh từ người dùng, đảm bảo tốc độ phản hồi (Latency) thấp nhất cho ứng dụng Web

1.3.9 Hệ quản trị cơ sở dữ liệu SQLite

Để giải quyết bài toán lưu trữ dữ liệu bền vững (Data Persistence) mà không làm phức tạp hóa cấu trúc hạ tầng, đề án lựa chọn SQLite làm hệ quản trị cơ sở dữ liệu (RDBMS) chính. Khác với các hệ thống cơ sở dữ liệu mô hình Client-Server truyền thống (như MySQL hay PostgreSQL), SQLite là một thư viện phần mềm cung cấp một công cụ cơ sở dữ liệu SQL giao dịch, khép kín và không cần máy chủ riêng biệt (Serverless).

Trong kiến trúc tổng thể của ứng dụng Web nhận diện trái chuối, SQLite đảm nhiệm vai trò là Kho lưu trữ trung tâm với các nhiệm vụ cụ thể:

- Quản lý Định danh và Phân quyền (User Management): Lưu trữ an toàn thông tin tài khoản người dùng, mật khẩu (đã được băm/mã hóa) và thông tin phiên đăng nhập. Điều này cho phép hệ thống phân cấp quyền truy cập, đảm bảo chỉ những người dùng được ủy quyền mới có thể truy cập vào các tính năng quản trị hoặc xem báo cáo chi tiết.
- Lưu trữ Nhật ký Nhận diện (Detection Logs): Đây là chức năng quan trọng nhất của Database. Mỗi khi hệ thống thực hiện nhận diện thành công (dù là qua ảnh tải lên hay Webcam), các metadata quan trọng sẽ được ghi lại vào cơ sở dữ liệu, bao gồm:
 - Thời gian thực hiện (Timestamp).
 - Kết quả phân loại (Class Label): Xanh/Chín/Hư hỏng.
 - Độ tin cậy (Confidence Score).
 - Đường dẫn tệp hình ảnh minh chứng.

Lý do lựa chọn SQLite cho đề án:

- Kiến trúc "Zero-configuration": SQLite không yêu cầu cài đặt máy chủ hay cấu hình phức tạp. Toàn bộ cơ sở dữ liệu được gói gọn trong một tệp tin duy nhất (.sqlite hoặc .db) trên ổ đĩa, giúp việc sao lưu, di chuyển hoặc triển khai hệ thống trở nên cực kỳ linh hoạt và đơn giản.
- Tích hợp sâu với Python: Python hỗ trợ SQLite nguyên bản thông qua thư viện sqlite3 có sẵn, giúp giảm thiểu các phụ thuộc bên ngoài (external dependencies) và tối ưu hóa tốc độ truy vấn dữ liệu giữa Backend Flask và Database.

1.3.10 Quản lý mã nguồn Git

Trong quy trình phát triển phần mềm hiện đại, đặc biệt là với các dự án tích hợp nhiều thành phần phức tạp từ AI đến Web App như đề tài này, việc kiểm soát chặt chẽ mã nguồn là yêu cầu tiên quyết. Đề án sử dụng Git – hệ thống quản lý phiên bản phân tán (DVCS) – kết hợp với kho lưu trữ đám mây GitHub để thiết lập một quy trình làm việc (workflow) chuyên nghiệp và an toàn.

a. Vai trò trong quản lý và bảo toàn mã nguồn

Git đóng vai trò như một cuốn "nhật ký kỹ thuật số", ghi nhận chi tiết mọi sự thay đổi

trong dự án theo thời gian thực. Mỗi khi một tính năng mới được hoàn thiện hoặc một lỗi được sửa, trạng thái của toàn bộ mã nguồn sẽ được đóng gói thành một phiên bản (Commit) với định danh duy nhất. Cơ chế này mang lại hai lợi ích cốt lõi:

- Truy vết và Khôi phục (Traceability & Rollback): Cho phép nhóm phát triển xem lại lịch sử thay đổi để xác định nguyên nhân gây lỗi, đồng thời dễ dàng khôi phục hệ thống về trạng thái ổn định trước đó nếu bản cập nhật mới gặp sự cố nghiêm trọng.
- An toàn dữ liệu: Việc đồng bộ mã nguồn lên GitHub (Remote Repository) đóng vai trò như một phương án sao lưu dự phòng (Backup) tối ưu, loại bỏ rủi ro mất mát dữ liệu do hư hỏng phần cứng cá nhân.

b. Chiến lược phân nhánh (Branching Strategy)

Để đảm bảo tính ổn định của sản phẩm trong khi vẫn liên tục phát triển các tính năng mới, đồ án áp dụng chiến lược phân nhánh Git Flow đơn giản hóa. Quy trình này chia tách mã nguồn thành các luồng làm việc độc lập:

- Nhánh chính (Main/Master Branch): Đây là nhánh chứa phiên bản mã nguồn ổn định nhất, đã qua kiểm thử và sẵn sàng để triển khai (Deploy). Mã nguồn tại đây được bảo vệ nghiêm ngặt và chỉ nhận các bản cập nhật đã hoàn thiện.
- Nhánh tính năng (Feature Branches): Khi phát triển một chức năng mới (ví dụ: "*tích hợp webcam*" hay "*tối ưu hóa model YOLO*"), một nhánh riêng biệt sẽ được tách ra từ nhánh chính. Việc này tạo ra một môi trường biệt lập (Sandbox) để thử nghiệm code mới mà không làm ảnh hưởng đến hệ thống đang chạy.

c. Quy trình tích hợp thực tế

- Sau khi một tính năng được hoàn thiện trên nhánh phụ, quy trình Hợp nhất (Merge) hoặc tạo Yêu cầu kéo (Pull Request) được thực hiện để đưa mã nguồn mới vào nhánh chính. Quá trình này giúp phát hiện sớm các xung đột mã nguồn (Code Conflicts) – hiện tượng thường xảy ra khi chỉnh sửa cùng một tệp tin. Việc giải quyết xung đột này một cách hệ thống giúp duy trì sự toàn vẹn và logic của toàn bộ dự án phần mềm.

1.4 Các chỉ số đánh giá hiệu năng mô hình (Model Evaluation Metrics)

Để đảm bảo tính khách quan và khoa học, hiệu suất của mô hình nhận diện trái chuối được đánh giá dựa trên các chỉ số tiêu chuẩn và thiết yếu trong bài toán Nhận diện Đối tượng (Object Detection), cụ thể là:

- Precision (Độ chính xác)

Precision đo lường tỷ lệ dự đoán đúng trên tổng số các dự đoán mà mô hình đã đưa ra. Đây là thước đo chất lượng dự đoán của mô hình.

Ý nghĩa Ứng dụng: Precision cao nghĩa là mô hình ít bị nhầm lẫn giữa trái chuối với các đối tượng nền khác (ví dụ: lá, cành cây, hoặc các loại trái cây khác).

- Recall (Độ phủ/Độ nhảy)

Recall đo lường tỷ lệ các đối tượng thực tế (trái chuối thực sự có trong ảnh) được mô hình phát hiện. Đây là thước đo mức độ đầy đủ của dự đoán.

Ý nghĩa Ứng dụng: Recall cao nghĩa là mô hình có khả năng phát hiện hầu hết các trái chuối có mặt trong khung hình, kể cả những trái bị che khuất một phần hoặc có kích thước nhỏ.

- mAP (mean Average Precision) - Độ chính xác Trung bình Tổng hợp

mAP là chỉ số tổng hợp quan trọng nhất và được sử dụng rộng rãi nhất để so sánh hiệu quả giữa các mô hình Object Detection. Nó kết hợp cả Precision và Recall, đồng thời tính đến chất lượng định vị hộp bao (bounding box).

- Cơ chế Đánh giá: mAP tính toán Độ chính xác Trung bình (Average Precision - AP) cho từng lớp (Chuối xanh, Chuối chín, Chuối chín quá) và sau đó lấy giá trị trung bình của các lớp đó.
- Vai trò của IoU: mAP được tính toán tại các ngưỡng IoU (Intersection over Union) khác nhau. IoU là tỷ lệ chồng lấn giữa hộp dự đoán và hộp thực tế.
 - mAP@0.5: Thường được sử dụng làm thước đo chuẩn, đánh giá hiệu quả khi hộp dự đoán và hộp thực tế chồng lấn tối thiểu 50%.
 - mAP@[0.5:0.95]: Là chỉ số nghiêm ngặt hơn, tính mAP trung bình trên các ngưỡng IoU từ 50% đến 95%, thể hiện khả năng định vị chính xác cao của mô hình.

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

2.1 Quy trình Xây dựng Bộ Dữ liệu Huấn luyện (Training Dataset Pipeline)

Việc xây dựng một bộ dữ liệu chất lượng cao, phản ánh đúng thực tế là yếu tố quyết định đến hiệu suất và khả năng khái quát hóa (Generalization) của mô hình YOLOv8. Quy trình được thực hiện theo các bước nghiêm ngặt sau:

2.1.1 Thu thập và Sàng lọc Dữ liệu Hình ảnh

Để đảm bảo mô hình có thể hoạt động ổn định trong các điều kiện thực tế đa dạng của nông nghiệp Việt Nam, dữ liệu hình ảnh được thu thập từ hai nguồn chính, với mục tiêu đạt tổng số lượng mẫu ban đầu khoảng 1.500 ảnh chất lượng cao:

- Nguồn dữ liệu Thực tế (In-Situ Data):

- Phương pháp: Ảnh chụp trực tiếp tại các vườn chuối và sạp nông sản bằng camera điện thoại thông minh (hoặc thiết bị ghi hình chất lượng cao).
- Ưu điểm: Phản ánh chân thực các thách thức môi trường như điều kiện ánh sáng thay đổi, các góc chụp khó, và các tình huống chuối bị che khuất một phần bởi lá hoặc nằm chồng chéo trong rổ.

- Nguồn dữ liệu Mở (Public/Open-Source Data):

- Phương pháp: Thu thập từ các nền tảng mở như Google Images, Kaggle, và các kho dữ liệu khoa học khác với các từ khóa chuyên biệt ("banana tree", "unripe banana", "ripe banana", etc.).
- Ưu điểm: Giúp đa dạng hóa về chủng loại chuối, kích thước và môi trường nền, tăng tính tổng quát của mô hình.

Tiêu chí Sàng lọc Dữ liệu

Trước khi đưa vào gán nhãn, tất cả ảnh đều phải qua quá trình sàng lọc nghiêm ngặt để loại bỏ các mẫu không đủ tiêu chuẩn: Ảnh bị mờ, nhòe, có độ phân giải quá thấp hoặc đối tượng không rõ ràng.

Dữ liệu được chuẩn hóa để nhận diện ba nhóm đối tượng chính, phục vụ cho mục đích phân loại độ chín:

Nhãn đối tượng	Đặc điểm nhận dạng	Mục đích phân loại
Chuối xanh (unripe)	Vỏ màu xanh lục, có độ tương phản thấp khi lần vào màu lá cây, đòi hỏi khả năng trích xuất đặc	Xác định chuối chưa đạt tiêu chuẩn thu hoạch

	trung mạnh mẽ.	
Chuối chín (ripe)	Vỏ chuyển sang màu vàng sáng hoặc vàng óng, có độ tương phản cao với nền.	Xác định chuối đạt tiêu chuẩn thu hoạch tối ưu (thường là để xuất khẩu).
Chuối chín quá (overripe)	Vỏ màu vàng sậm, có thể xuất hiện đốm nâu hoặc đen, thường có hình dạng mềm và dễ bị biến dạng.	Hỗ trợ phân loại để xử lý nội địa hoặc chế biến.

Bảng 2.1 Dữ liệu chuẩn hóa để nhận diện 3 đối tượng chính

2.1.2 Tiền xử lý và gán nhãn dữ liệu với Roboflow

Quy trình tiền xử lý và chuẩn hóa dữ liệu là giai đoạn then chốt, được thực hiện trên nền tảng Roboflow để đảm bảo tính đồng bộ, chính xác và chuẩn hóa định dạng đầu vào cho mô hình YOLOv8.

Gán nhãn Chuyên nghiệp (Data Annotation)

Việc gán nhãn (Annotation) được thực hiện thủ công một cách tỉ mỉ bằng giao diện trực quan của Roboflow, vẽ các hộp bao quanh (Bounding Box) sát với biên dạng của từng trái chuối:

- Quy tắc Gán nhãn: Để huấn luyện mô hình có khả năng chịu lỗi cao, chúng tôi gán nhãn cho tất cả các trái chuối có thể nhìn thấy, bao gồm cả những trái bị che khuất một phần (Occlusion). Điều này giúp mô hình học được đặc trưng đối tượng trong môi trường bối cảnh phức tạp và thực tế.
- Các Lớp Định nghĩa (Class Names): Hệ thống sử dụng ba nhãn lớp (classes) chuyên biệt, phản ánh các giai đoạn độ chín, nhằm phục vụ mục đích phân loại chất lượng:
 - unripe (Chuối xanh)
 - ripe (Chuối chín)
 - overripe (Chuối chín quá)

Tăng cường Dữ liệu (Data Augmentation) Chiến lược

Để khắc phục vấn đề thiếu hụt dữ liệu (Data Scarcity) và nâng cao khả năng khái quát hóa (Generalization), đồng thời giảm thiểu hiện tượng quá khớp (Overfitting), chúng tôi áp dụng các kỹ thuật tăng cường dữ liệu chiến lược sau:

Kỹ thuật Tăng cường	Phạm vi Ứng dụng	Mục đích Ứng dụng
Flip (Lật)	Lật ảnh theo chiều	Tạo sự bất biến đối với

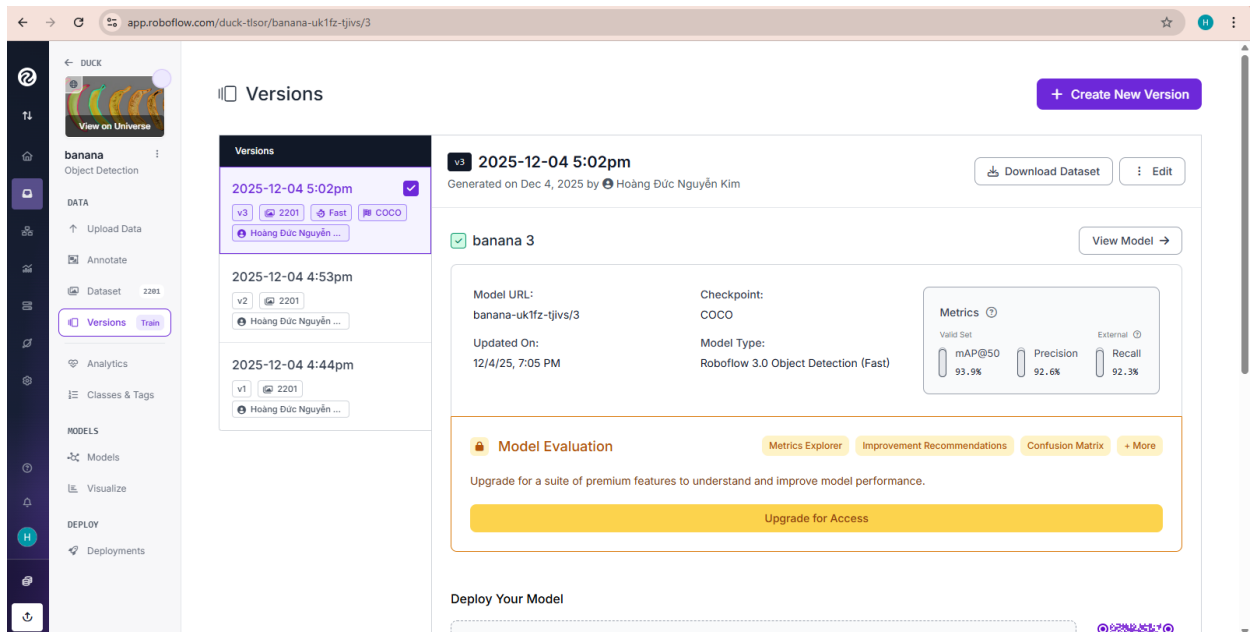
	ngang và/hoặc chiều dọc.	hướng và góc nhìn của camera.
Rotation (Xoay)	Xoay ảnh ngẫu nhiên trong khoảng 15 độ	Mô phỏng sự thay đổi góc nhìn khi quan sát chuỗi trên cây.
Brightness (Độ sáng)	Điều chỉnh độ sáng trong khoảng 20%	Mô phỏng các điều kiện thời tiết và ánh sáng khác nhau (nắng gắt, trời râm, hoàng hôn).
Noise (Nhiều)	Thêm nhiễu hạt ngẫu nhiên.	Tăng khả năng chịu lỗi của mô hình đối với hình ảnh có chất lượng camera thấp.

Bảng 2.2 Chiến thuật tăng cường độ nhận diện của hình ảnh

Phân chia Tập dữ liệu Chuẩn (Data Splitting)

Sau khi tăng cường, bộ dữ liệu tổng thể được phân chia theo tỷ lệ tiêu chuẩn cho các bài toán Học sâu nhằm đảm bảo tính chính xác và khách quan trong quá trình huấn luyện và đánh giá:

- 70% Tập Train (Huấn luyện): Dùng để huấn luyện mô hình, điều chỉnh trọng số.
- 20% Tập Validation (Kiểm định): Dùng để tinh chỉnh các siêu tham số (Hyperparameters) và theo dõi sự hội tụ của mô hình.
- 10% Tập Test (Kiểm thử): Dùng để đánh giá cuối cùng hiệu suất thực tế của mô hình trên dữ liệu hoàn toàn chưa từng thấy.



Hình 2.1 thu thập dữ liệu và kết quả trên Roboflow

2.2 Huấn luyện mô hình YOLOv8

Quy trình huấn luyện được thiết kế để đạt được hiệu suất tối ưu và khả năng hội tụ nhanh chóng, dựa trên chiến lược Học chuyển tiếp (Transfer Learning) và tối ưu hóa tài nguyên tính toán.

2.2.1 Thiết lập Môi trường Huấn luyện Chuyên biệt

Quá trình huấn luyện được triển khai trên nền tảng Google Colab, tận dụng lợi thế của việc cấp phát miễn phí các tài nguyên tính toán cao cấp để rút ngắn đáng kể thời gian xử lý:

- **Phần cứng Cốt lõi (Hardware Configuration):** Tận dụng tối đa sức mạnh của Bộ xử lý Đồ họa (GPU) NVIDIA Tesla T4 (với 16GB VRAM) và RAM hệ thống 12GB. Việc này là thiết yếu để xử lý hiệu quả các phép tính Tensor trong mạng nơ-ron sâu và cho phép sử dụng Batch Size lớn hơn.
- **Framework và Thư viện:**
 - Framework Nền tảng: PyTorch – framework Deep Learning phổ biến, cung cấp nền tảng linh hoạt cho việc định nghĩa và huấn luyện mạng nơ-ron.
 - Thư viện: Ultralytics YOLOv8 – thư viện chuyên dụng để quản lý mô hình YOLOv8, cung cấp các API (Application Programming Interface) đơn giản hóa quá trình tương tác và huấn luyện.

2.2.3 Chiến lược Huấn luyện và Cấu hình Siêu tham số (Hyperparameters)

Quá trình huấn luyện sử dụng chiến lược Học chuyển tiếp (Transfer Learning). Kỹ thuật này khởi tạo mô hình bằng cách sử dụng trọng số đã được huấn luyện sẵn (pre-trained weights) từ bộ dữ liệu lớn COCO (yolov8n.pt hoặc yolov8s.pt). Điều này giúp mô hình hội tụ nhanh hơn, giảm nhu cầu về lượng dữ liệu huấn luyện và cải thiện tính ổn định.

Cấu hình Tham số Cốt lõi:

Siêu tham số (Hyperparameter)	Giá trị Thiết lập	Ý nghĩa Chiến lược
Epochs	30	Số lần mô hình duyệt qua và học hỏi từ toàn bộ tập dữ liệu . Số lượng này được cân nhắc để đạt sự hội tụ mà không gây quá khớp (overfitting).
Batch Size	16	Số lượng ảnh được xử lý đồng thời trong mỗi lần lặp (iteration). Giá trị này được chọn để tối ưu hóa việc sử dụng 16GB VRAM của GPU Tesla T4.
Image Size	640x640 pixel	Kích thước ảnh đầu vào được chuẩn hóa, đảm bảo sự cân bằng giữa việc giữ lại chi tiết đặc trưng và yêu cầu tính toán của mô hình.
Learning Rate	0.01	Tốc độ học khởi tạo (Initial Learning Rate), quyết định bước nhảy để mô hình điều chỉnh trọng số.

Bảng 2.3 Cấu hình tham số cốt lõi

2.3 Phân tích và thiết kế hệ thống web

2.3.1 Phân tích yêu cầu hệ thống

Hệ thống Web được thiết kế dựa trên nguyên tắc Hướng người dùng (User-centric), nhằm đáp ứng tối đa nhu cầu tự động hóa và quản lý chất lượng nông sản của người nông dân và cán bộ quản lý.

Yêu cầu Chức năng (Functional Requirements):

- Chức năng Nhận dạng Ảnh tĩnh (Upload): Cho phép người dùng tải lên hình ảnh từ thiết bị cá nhân. Hệ thống phải thực hiện nhận diện và trả về kết quả trực quan với khung bao quanh (bounding box) và nhãn loại chuỗi.
- Chức năng Nhận diện Thời gian thực (Real-time Camera): Cho phép sử dụng Webcam để nhận diện trực tiếp luồng video. Hệ thống cần hỗ trợ điều chỉnh các tham số quan trọng như ngưỡng tin cậy (Confidence Threshold) và độ phân giải để tối ưu tốc độ.
- Chức năng Quản lý Lịch sử: Lưu trữ chi tiết thông tin các phiên nhận diện (thời gian, tổng số lượng chuỗi, phân loại độ chín, và hình ảnh kết quả) để phục vụ việc tra cứu và truy xuất dữ liệu sau này.
- Chức năng Thống kê Sản lượng: Tự động đếm và phân loại số lượng chuỗi theo từng trạng thái (xanh, chín, chín quá) trong ảnh, cung cấp cái nhìn định lượng về chất lượng mùa vụ.
- Chức năng Hỗ trợ (Chatbot): Tích hợp Chatbot cơ bản để giải đáp các thắc mắc thường gặp của người dùng về cách sử dụng hệ thống hoặc các vấn đề nông

ngành liên quan.

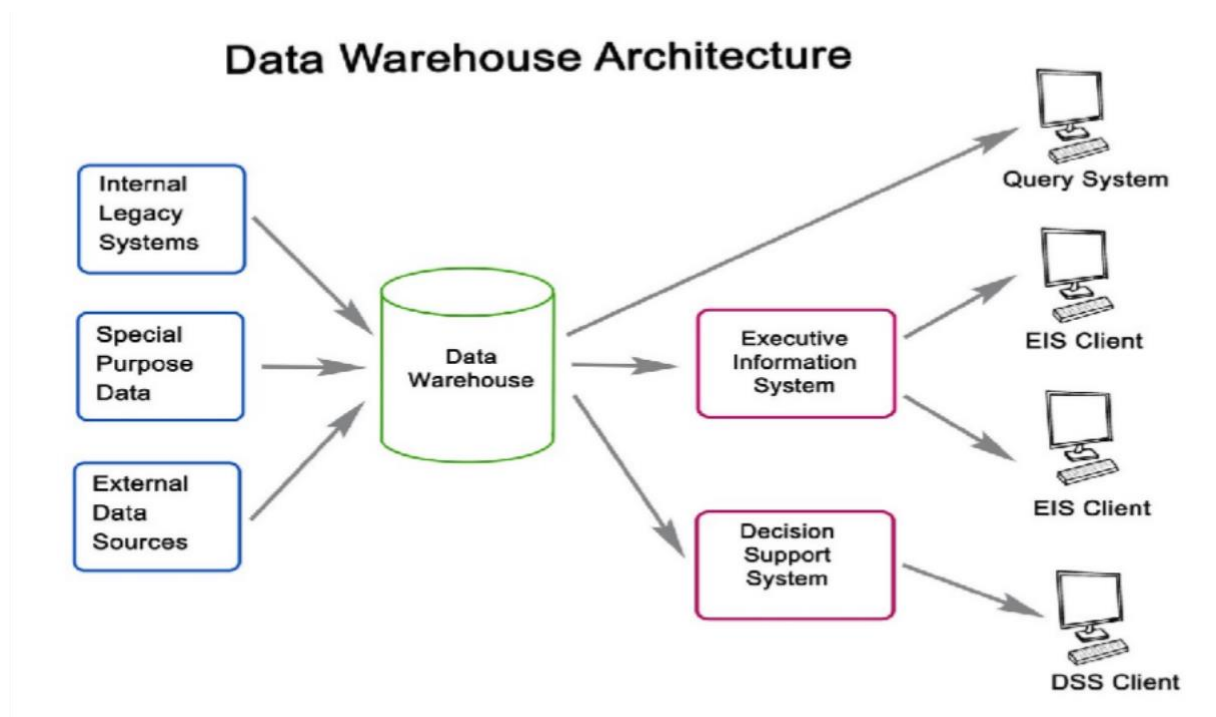
Yêu cầu phi chức năng (Non-Functional Requirements):

Yêu cầu	Tiêu chí Định lượng
Tốc độ Xử lý (Performance)	Xử lý ảnh tĩnh phải dưới 1 giây. Tốc độ khung hình (FPS) của Camera phải đạt mức ổn định để đảm bảo trải nghiệm Real-time.
Giao diện (Usability)	Phải thân thiện, trực quan và dễ sử dụng (User- Friendly), hỗ trợ đầy đủ ngôn ngữ Tiếng Việt.
Tính Tương thích (Compatibility)	Hệ thống cần chạy ổn định và hiển thị chính xác trên các trình duyệt Web phổ biến (như Chrome, Edge, Firefox).

Bảng 2.4 Yêu cầu phi chức năng

2.3.2 Kiến trúc hệ thống

Hệ thống được thiết kế theo Mô hình kiến trúc 3 Lớp (Three-Tier Architecture) đơn giản hóa, tận dụng sự gọn nhẹ và khả năng tích hợp mạnh mẽ của Framework Streamlit.



Hình 2.2 kiến trúc hệ thống mô phỏng

Lớp (Layer)	Thành phần Chính	Chức năng Cốt lõi
1. Lớp giao diện (Presentation Layer)	Streamlit/HTML (Frontend)	Chịu trách nhiệm hiển thị Dashboard chính, Sidebar điều khiển và tất cả các thành phần tương tác (widget). Lớp này nhận tương tác đầu vào trực tiếp từ người dùng.
2. Lớp xử lý nghiệp vụ (Business Logic Layer)	Module process_yolo (trong app.py)	Xử lý mô hình AI: Gọi mô hình YOLOv8 đã huấn luyện để thực hiện suy luận (Inference), xử lý ảnh đầu vào, và định dạng kết quả nhận diện. Logic Ứng dụng: Xử lý các nghiệp vụ phụ trợ như chuyển đổi định dạng ảnh và quản lý session làm việc.
3. Lớp dữ liệu (Data Layer)	Session State & Database (SQLite/MongoDB)	Lưu trữ Tạm thời: Sử dụng Session State để lưu trữ lịch sử các lần nhận diện trong phiên làm việc hiện tại. Lưu trữ Bền vững: Sử dụng Database để lưu trữ các bản ghi lâu dài phục vụ cho chức năng thống kê và Quản lý Lịch sử.

Bảng 2.5 Các lớp xử lý nghiệp vụ chính

2.4 Xây dựng ứng dụng Web nhận diện chuỗi:

Ứng dụng Web được xây dựng dựa trên kiến trúc Flask kết hợp với TensorFlow/PyTorch (thông qua thư viện Ultralytics YOLOv8) để tạo ra một giải pháp đầu cuối (End-to-End) từ nhận diện mô hình AI đến quản lý người dùng và xuất báo cáo. Hệ thống tuân thủ nguyên tắc module hóa, phân tách rõ ràng chức năng thành các thành phần chính.

2.4.1 Kiến trúc Module hóa và Công nghệ Triển khai:

Hệ thống được thiết kế theo mô hình API-driven, nơi Flask đóng vai trò là lõi Backend xử lý các tác vụ nghiệp vụ, bảo mật, và tương tác cơ sở dữ liệu.

Module/Thành phần	Công nghệ/Class	Chức năng Chính
Lõi Ứng dụng	app.py (Flask)	Quản lý các Route chính, xử lý logic đăng nhập/đăng ký, và kết nối các module.
Cấu hình	config.py	Chứa tất cả các hằng số và tham số cấu hình hệ thống (đường dẫn, ngưỡng tin cậy, cấu hình Flask, tên lớp).
Nhận diện AI	utils/detection.py	Quản lý logic nhận diện AI: Khởi tạo và Load mô hình YOLOv8 (DetectionHandler), tiền xử lý ảnh, chạy suy luận, và vẽ Bounding Box.
Quản lý Dữ liệu	utils/database.py	Quản lý tất cả thao tác với SQLite database: Quản lý Users và Detections (Database).
Xuất Báo cáo	utils/export.py	Xử lý nghiệp vụ tạo và xuất file Excel hoặc CSV (ExportHandler).
Giao diện	templates/	Xây dựng giao diện người dùng (HTML/CSS/JavaScript) cho các chức năng.

Bảng 2.6 Các module chính chức năng

Một số hình ảnh demo giao diện và cấu hình:

```
app.py A database.py A detection.py A index.html 9+, A HUONG_DAN_CHI_TIET.md A config.py A
templates > index.html > html > body > script
1 <!DOCTYPE html>
2 <html lang="vi">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Hệ thống nhận diện độ chín chuối</title>
7   <link rel="stylesheet" href="/static/css/style.css">
8 </head>
9 <body>
10   <div class="container">
11     <header>
12       <div style="display: flex; justify-content: space-between; align-items: center;">
13         <div>
14           <h1>🍌 Hệ thống nhận diện độ chín chuối</h1>
15           <p class="subtitle">Ứng dụng AI trong nông nghiệp</p>
16         </div>
17         <div style="text-align: right;">
18           {% if user %}
19             <p style="color: white; margin-bottom: 10px;">Xin chào, <strong>{{ user.username }}</strong></p>
20             <a href="/history" style="color: white; text-decoration: none; background: rgba(255,255,255,0.2); padding: 8px">
21               <a href="/logout" style="color: white; text-decoration: none; background: rgba(255,255,255,0.2); padding: 8px">
22             {% else %}
23               <a href="/login" style="color: white; text-decoration: none; background: rgba(255,255,255,0.2); padding: 8px">
24               <a href="/register" style="color: white; text-decoration: none; background: rgba(255,255,255,0.3); padding: 8px">
25             {% endif %}
26           </div>
27         </div>
28     </header>
```

Hình 2.3 một số code demo giao diện

```
app.py A database.py A detection.py A index.html 9+, A HUONG_DAN_CHI_TIET.md A config.py A X
config.py > ...
91 # =====
92 FLASK_CONFIG = {
93   'host': '127.0.0.1', # Địa chỉ IP (127.0.0.1 = localhost, chỉ truy cập từ máy local)
94   'port': 5000, # Port của server
95   'debug': True, # Bật debug mode (tự reload khi code thay đổi, hiển thị lỗi chi tiết)
96   'use_reloader': False, # Tắt reloader để tránh mở nhiều tab trình duyệt khi debug
97 }
98
99 # =====
100 # CẤU HÌNH UPLOAD FILE
101 # =====
102 UPLOAD_CONFIG = {
103   'max_file_size': 10 * 1024 * 1024, # Kích thước file tối đa: 10MB
104   # Các định dạng ảnh được phép upload
105   'allowed_extensions': ['.jpg', '.jpeg', '.png', '.bmp'],
106 }
107
108 # =====
109 # TÊN CÁC CLASS NHẬN DIỆN
110 # =====
111 # Mapping từ tên class (key trong model) sang tên tiếng Việt (để hiển thị)
112 CLASS_NAMES = {
113   'unripe': 'Chưa chín', # Chuối còn xanh, chưa chín
114   'ripe': 'Chín', # Chuối vàng, đã chín
115   'overripe': 'Chín quá' # Chuối đã chín quá, có đốm nâu
116 }
117
```

Hình 2.4 một số code cấu hình hiển thị và được chạy qua cổng nào

```
app.py A database.py A X detection.py A index.html 9+, A HUONG_DAN_CHI_TIET.md A config.py A D v
utils > database.py > ...
19 class Database:
42 def init_database(self) -> None:
56
57 # =====
58 # BẢNG USERS - Lưu thông tin người dùng
59 # =====
60 cursor.execute('''
61 CREATE TABLE IF NOT EXISTS users (
62     id INTEGER PRIMARY KEY AUTOINCREMENT, -- ID tự động tăng
63     username TEXT UNIQUE NOT NULL, -- Tên đăng nhập (duy nhất, bắt buộc)
64     email TEXT UNIQUE, -- Email (duy nhất, tùy chọn)
65     password_hash TEXT NOT NULL, -- Mật khẩu đã hash (SHA256)
66     created_at DATETIME NOT NULL, -- Thời gian tạo tài khoản
67     last_login DATETIME, -- Lần đăng nhập cuối
68     is_active INTEGER DEFAULT 1 -- Trạng thái (1=active, 0=inactive)
69 )
70 '''
71
72 # =====
73 # BẢNG DETECTIONS - Lưu kết quả nhận diện
74 # =====
75 cursor.execute('''
76 CREATE TABLE IF NOT EXISTS detections (
77     id INTEGER PRIMARY KEY AUTOINCREMENT, -- ID tự động tăng
78     timestamp DATETIME NOT NULL, -- Thời gian nhận diện
79     unripe_count INTEGER DEFAULT 0, -- Số chuỗi chưa chín
80     ripe_count INTEGER DEFAULT 0, -- Số chuỗi chín
81     overripe_count INTEGER DEFAULT 0, -- Số chuỗi chín quá
82     total_count INTEGER DEFAULT 0, -- Tổng số chuỗi
83     confidence_threshold REAL DEFAULT 0.5, -- Ngưỡng tin cậy đã dùng
84     image_path TEXT, -- Đường dẫn ảnh (nếu lưu)
85     notes TEXT, -- Ghi chú
86     user_id INTEGER, -- ID người dùng thực hiện
87     FOREIGN KEY (user_id) REFERENCES users(id) -- Liên kết với bảng users
88 )
89 '''
90
```

Hình 2.5 Một số code quản lý dữ liệu tạm thời database.py

2.4.2. Triển khai Lỗi Ứng dụng và Bảo mật (app.py)

Module chính (app.py) đảm nhận việc khởi tạo ứng dụng và định nghĩa các luồng điều khiển chính:

Khởi tạo và Phụ thuộc:

- Ứng dụng được khởi tạo bằng Flask và sử dụng các thư viện hỗ trợ như OpenCV (cho xử lý ảnh), NumPy, và xlsxwriter (cho xuất Excel).
- Các module nghiệp vụ (db, detection_handler, export_handler) được Import và Khởi tạo Toàn cục (Global Instance), đảm bảo mô hình AI và kết nối Database chỉ được tải một lần.

Cơ chế Bảo mật:

- login_required Decorator: Được sử dụng để bảo vệ các chức năng cốt lõi (Trang chủ, Nhận diện, Lịch sử, Xuất báo cáo). Nếu người dùng chưa xác thực, họ sẽ được Redirect về trang Đăng nhập.

Định nghĩa Các Route Chính (MVC Controller):

Route	Phương thức	Chức năng
/, /home	GET	Hiển thị Trang chủ với giao diện nhận diện Ảnh/Webcam.
/login, /register, /logout	GET/POST	Xử lý toàn bộ luồng Xác thực Người dùng (Authentication).
/history	GET	Hiển thị Lịch sử nhận diện chi tiết, hỗ trợ lọc theo ngày.
/export	GET	Xử lý yêu cầu Xuất báo cáo Excel trực tiếp từ Database.

Bảng 2.7 Các route chính

2.4.3. Xây dựng Module Xử lý AI (utils/detection.py)

Module này đóng vai trò là cầu nối giữa Web App và mô hình Học sâu, được tổ chức trong Class DetectionHandler.

- Khởi tạo Mô hình: Hàm __init__() tự động tải file trọng số best.pt từ MODEL_PATH và thiết lập cấu hình YOLOv8 (confidence_threshold, iou_threshold) cũng như tự động phát hiện và sử dụng device (GPU/CPU).
- Hàm detect() (Hàm Chính):
 - Tiếp nhận ảnh đầu vào (dạng file upload hoặc numpy array từ Webcam).
 - Gọi preprocess_image() để chuẩn hóa ảnh.
 - Thực hiện Suy luận (Inference) bằng cách gọi model YOLOv8.
 - Xử lý kết quả (_process_results()) để trích xuất bounding box, nhãn, độ

tin cậy, và tính toán thống kê (số lượng Xanh/Chín/Quá chín).

- Vẽ kết quả lên ảnh gốc (`_draw_detections()`) với màu sắc quy ước (Xanh lá - Chưa chín, Cam - Chín, Đỏ - Chín quá).
- Trả về ảnh đã chú thích (dạng base64) và dữ liệu thống kê qua JSON.

2.4.4. Xây dựng Giao diện Người dùng (templates/)

Giao diện được thiết kế để cung cấp trải nghiệm Tương tác Thời gian thực (Real-time Interaction) và Quản lý dữ liệu hiệu quả:

- Trang Chủ (`index.html`): Sử dụng JavaScript để quản lý:

- Webcam Streaming: Chức năng `startWebcam()` và `detectFrame()` tạo ra luồng video và gửi từng frame ảnh lên API `/detect` để xử lý nhận diện định kỳ.
- Kết quả Trực quan: Hiển thị ảnh đã vẽ Bounding Box lên canvas overlay và cập nhật thống kê real-time.

- Trang Lịch sử (`history.html`): Sử dụng API Routes chuyên biệt (`/api/history`, `/api/statistics`) để tải dữ liệu lịch sử và thống kê thông qua AJAX, cho phép lọc theo ngày và xóa bản ghi mà không cần tải lại trang.

2.4.5. Định nghĩa API Backend (JSON Endpoint)

Hệ thống cung cấp các API Routes riêng biệt để phục vụ giao tiếp giữa Frontend và Backend, giúp xử lý dữ liệu động:

API Route	Phương thức	Chức năng	Output
/detect	POST	Nhận diện chuỗi, lưu Database, trả về ảnh Base64.	JSON (Image, Detections, Stats)
/api/history	GET	Trả về danh sách lịch sử nhận diện.	JSON Array
/api/statistics	GET	Trả về số liệu thống kê tổng hợp.	JSON Object (Tổng số, Xanh/Chín/Quá chín)
/api/export/excel	GET	Trả về file Excel được tạo động.	File Excel
/api/detection/<id>	DELETE	Xóa một bản ghi khỏi Database.	JSON (Success/Message)

Bảng 2.8 Các API chính

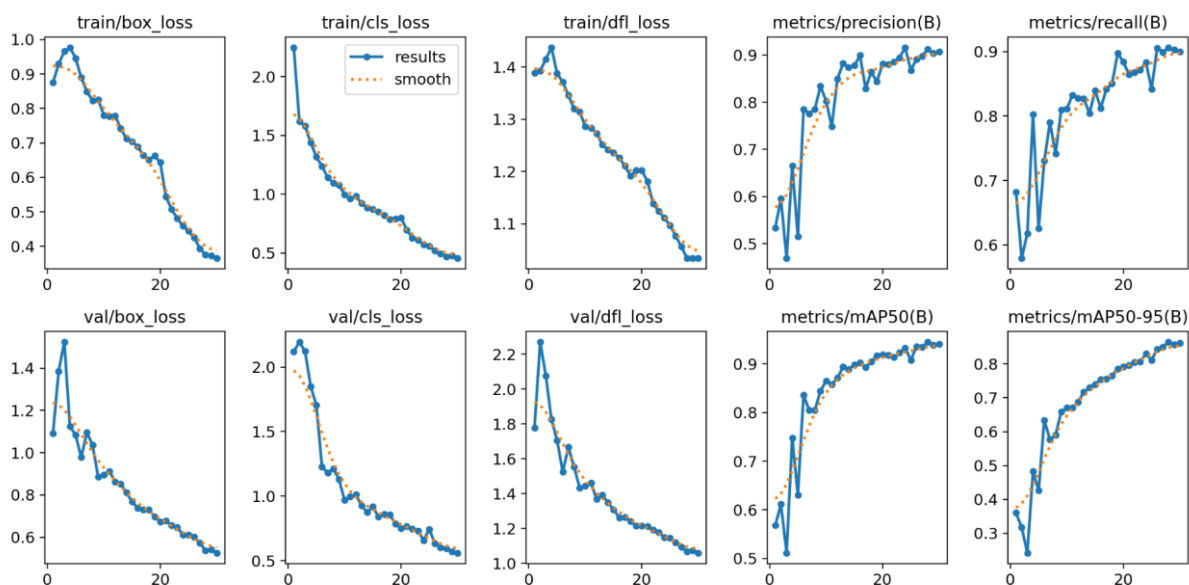
CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ

3.1. Kết quả thực nghiệm và đánh giá

3.1.1. Phân tích Quá trình Hội tụ qua Hàm Mất mát (Loss Functions)

Quá trình huấn luyện được theo dõi chặt chẽ thông qua ba thành phần mất mát chính, đảm bảo mô hình đạt được sự cân bằng tối ưu giữa khả năng định vị và phân loại mà không xảy ra hiện tượng quá khớp (Overfitting):

- Box Loss (Mất mát Vị trí): Đo lường sai số giữa hộp bao dự đoán và hộp bao thực tế (Ground Truth). Kết quả cho thấy Box Loss giảm nhanh chóng trong khoảng 15 epochs đầu và ổn định dần về mức thấp (dưới 0.02) ở các epochs cuối. Điều này khẳng định mô hình đã học được cách định vị chính xác biên dạng và vị trí của trái chuối trong mọi bối cảnh.
- Cls Loss (Mất mát Phân loại): Đo lường sai số trong việc xác định nhãn lớp (Chuối xanh, Chuối chín, Chuối chín quá). Cls Loss giảm sâu và tiệm cận 0, chứng tỏ mô hình có khả năng phân biệt cực kỳ tốt giữa các trạng thái độ chín dựa trên đặc điểm màu sắc vỏ.
- DFL Loss (Distribution Focal Loss): Hàm mất mát bổ sung này đã hoàn thành tốt vai trò của mình trong việc tinh chỉnh độ chính xác của các cạnh hộp bao, đặc biệt hiệu quả khi xử lý các trường hợp phức tạp như trái chuối bị che khuất một phần hoặc nằm chồng chéo.



Hình 3.1 Kết quả đánh giá mô hình YOLO v8 sau khi đã huấn luyện

3.1.2. Các chỉ số đánh giá hiệu suất (Performance Metrics) và phân tích ma trận nhầm lẫn

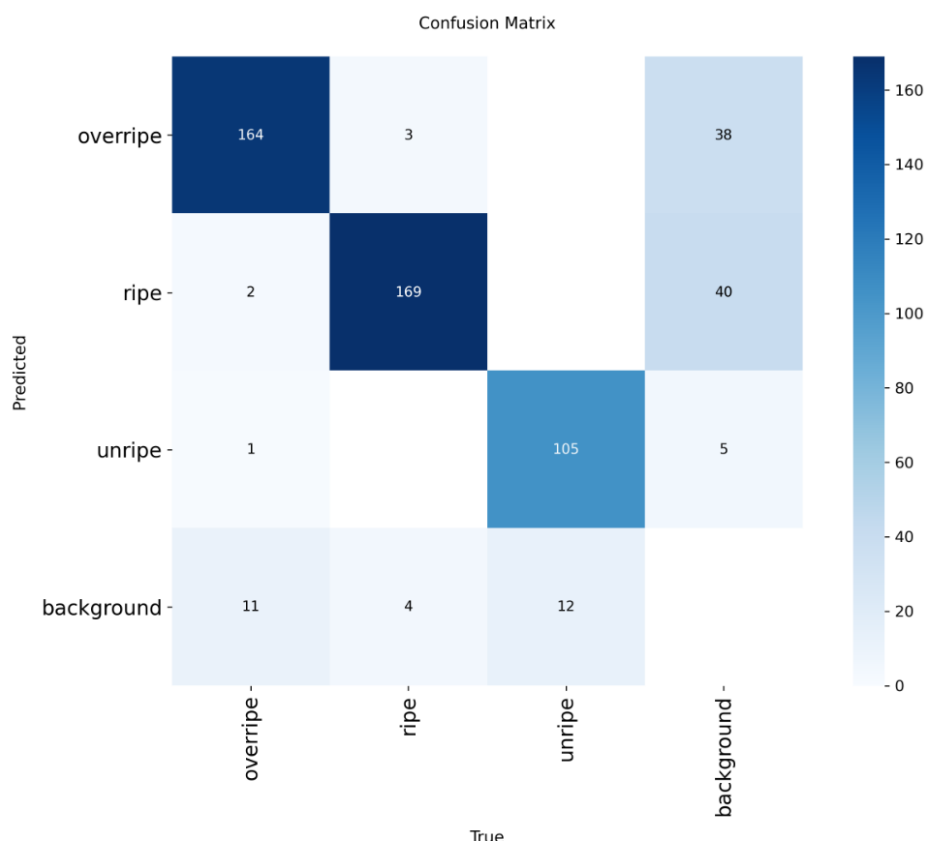
Hiệu suất của mô hình tối ưu (best.pt) được đánh giá toàn diện trên tập Validation (20% dữ liệu kiểm định) thông qua các chỉ số tiêu chuẩn và phân tích định lượng từ Ma trận Nhầm lẫn:

- Chỉ số Hiệu suất Tiêu chuẩn:

Các chỉ số tổng hợp khẳng định mô hình đã đạt được độ chính xác rất cao, phù hợp cho ứng dụng thực tế:

Chỉ số	Kết quả	Ý nghĩa Ứng dụng
Precision (Độ chính xác)	~92.6%	Khẳng định độ tin cậy cao: Khi mô hình dự báo là "chuối", thì 91.5% trường hợp đó là chính xác, cho thấy tỉ lệ cảnh báo giả (False Positive) rất thấp.
Recall (Độ nhạy)	~92.3%	Chứng minh tính toàn diện: Mô hình có khả năng phát hiện gần 90% số lượng trái chuối thực tế có trong ảnh, bao gồm cả những trái bị che khuất hoặc có kích thước nhỏ.
mAP@0.5 (Mean Average Precision)	~93.9%	Chỉ số tổng hợp quan trọng nhất, khẳng định mô hình có độ tin cậy và hiệu suất tổng thể rất cao trong việc định vị và phân loại chuối khi ngưỡng chồng lấn (IoU) đặt ở mức tiêu chuẩn 50%.

Bảng 3.1 Kết quả đánh giá hiệu suất



Hình 3.2 Kết quả đánh giá hiệu suất qua Google colab

3.2. Kết quả xây dựng ứng dụng web (WEB APPLICATION RESULTS)

3.2.1. Giao diện người dùng và Trải nghiệm (UI/UX)

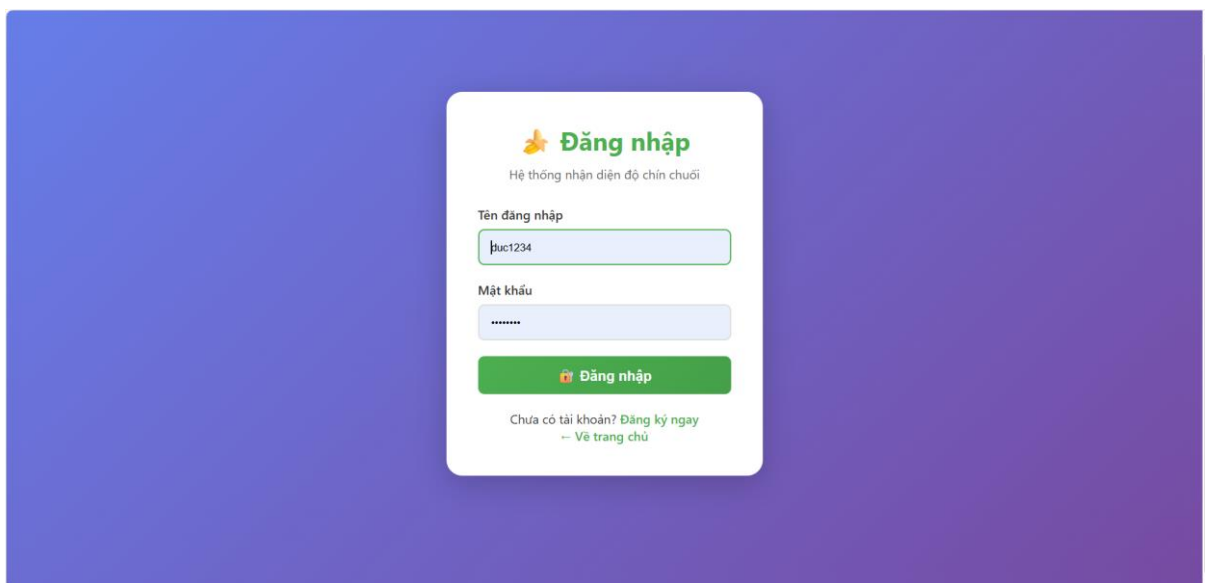
Giao diện ứng dụng được thiết kế dựa trên các nguyên tắc tối giản (Minimalism) và chuyên nghiệp, nhằm đảm bảo tính trực quan và dễ sử dụng cho đối tượng người dùng trong lĩnh vực nông nghiệp.

Thiết kế Trực quan và Màu sắc Định hướng:

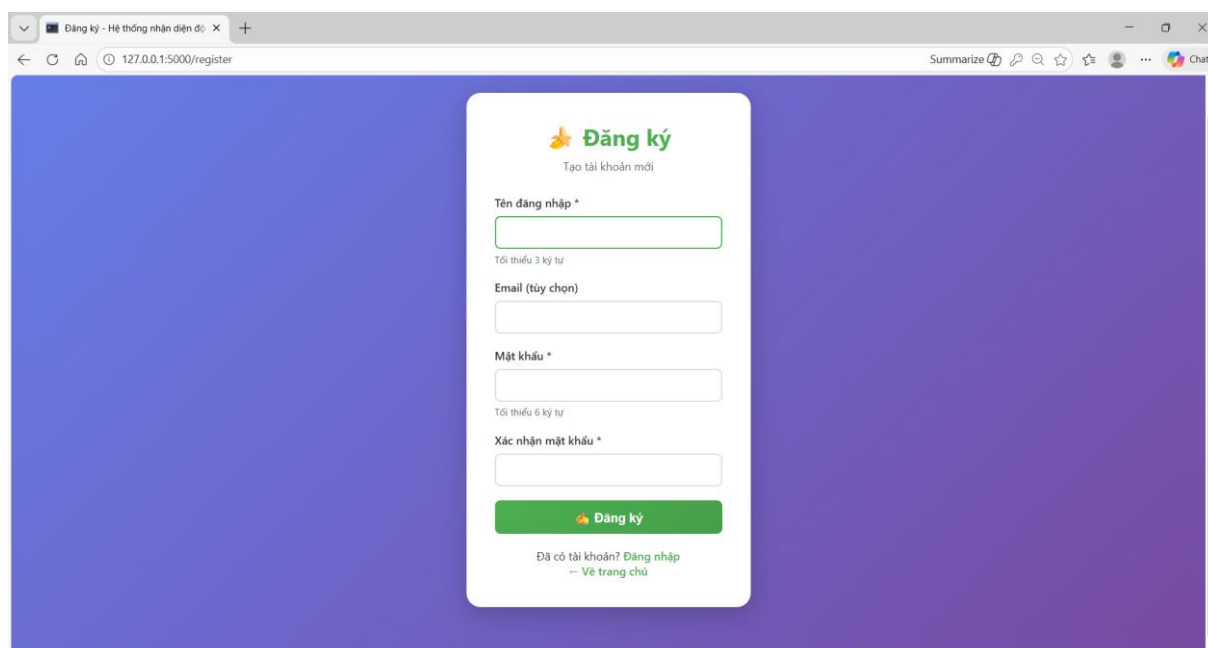
- **Tông màu Chủ đạo:** Ứng dụng sử dụng tông màu chủ đạo là Xanh Nước biển Nhạt (Light Blue – #E1F5FE), tạo cảm giác hiện đại, sạch sẽ và thân thiện.
- **Điểm nhấn Nông sản:** Màu Cam (Orange – #FFB74D) được sử dụng làm màu nhấn (Accent Color). Việc sử dụng màu sắc này một cách chiến lược không chỉ tạo độ tương phản cao mà còn gợi lên màu sắc của trái chuối chín, tăng tính nhận diện và sự liên kết với nội dung đề tài.
- **Thiết kế Đáp ứng (Responsive Design):** Giao diện được phát triển theo tiêu chuẩn Responsive, đảm bảo ứng dụng tự động điều chỉnh kích thước và hiển thị hoàn hảo trên mọi thiết bị, từ màn hình Desktop cho đến các thiết bị di động (máy tính bảng, smartphone) ngoài vườn.

Tối ưu hóa Trải nghiệm Người dùng (UX):

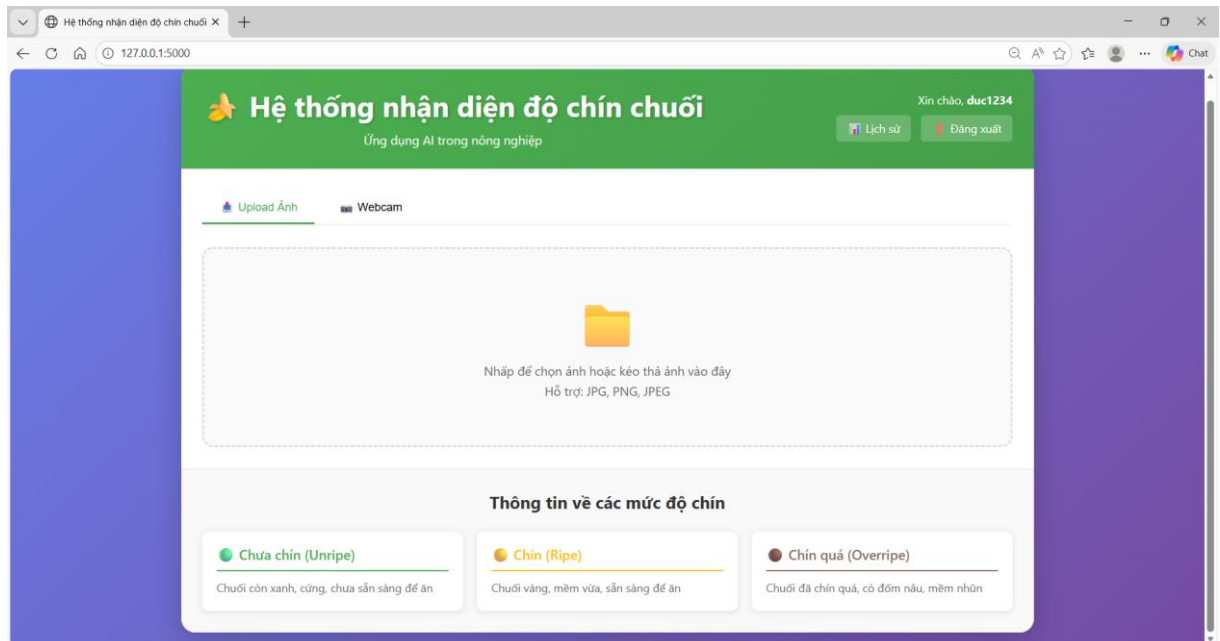
- Sidebar Điều khiển Tập trung: Tất cả các cấu hình kỹ thuật và tùy chọn phức tạp (như Ngưỡng Confidence, Độ phân giải Camera, Chế độ nhận diện) đều được tập trung vào thanh điều khiển bên trái (Sidebar).
- Màn hình Chính Rộng rãi: Việc cô đọng các tùy chọn vào Sidebar giúp màn hình hiển thị kết quả chính (Output Area) trở nên rộng rãi và thoáng đãng, tạo không gian tối đa để người dùng quan sát và đánh giá ảnh/video đã được nhận diện.



Hình 3.3 Demo giao diện đăng nhập



Hình 3.4 Demo giao diện đăng ký



Hình 3.5 Demo giao diện web nhận diện qua upload ảnh

3.2.2. Chức năng nhận diện qua Tải ảnh (Upload Mode)

Chức năng này là phương thức cơ bản và tối ưu để xử lý các ảnh tĩnh, thường là ảnh chất lượng cao được thu thập thủ công, nhằm phục vụ cho các mục đích kiểm kê và đánh giá chất lượng sản phẩm.

Quy trình Xử lý Đầu cuối (End-to-End Workflow)

Quy trình nhận diện ảnh tĩnh diễn ra tự động và khép kín qua các bước sau:

- Đầu vào: Người dùng lựa chọn và tải lên một file ảnh chất lượng cao từ thiết bị cá nhân.

- Xử lý Logic: Ứng dụng gọi hàm `detection_handler.detect()` (hoặc `process_yolo`) để truyền ảnh, thực hiện các bước tiền xử lý (chuyển đổi định dạng, đổi kênh màu) và chạy suy luận (Inference) trên mô hình YOLOv8.

- Đầu ra: Hệ thống trả về hai thành phần chính: Ảnh đã chú thích (Annotated Image) và Dữ liệu Thống kê (JSON).

Kết quả Hiển thị và Phân tích Trực quan

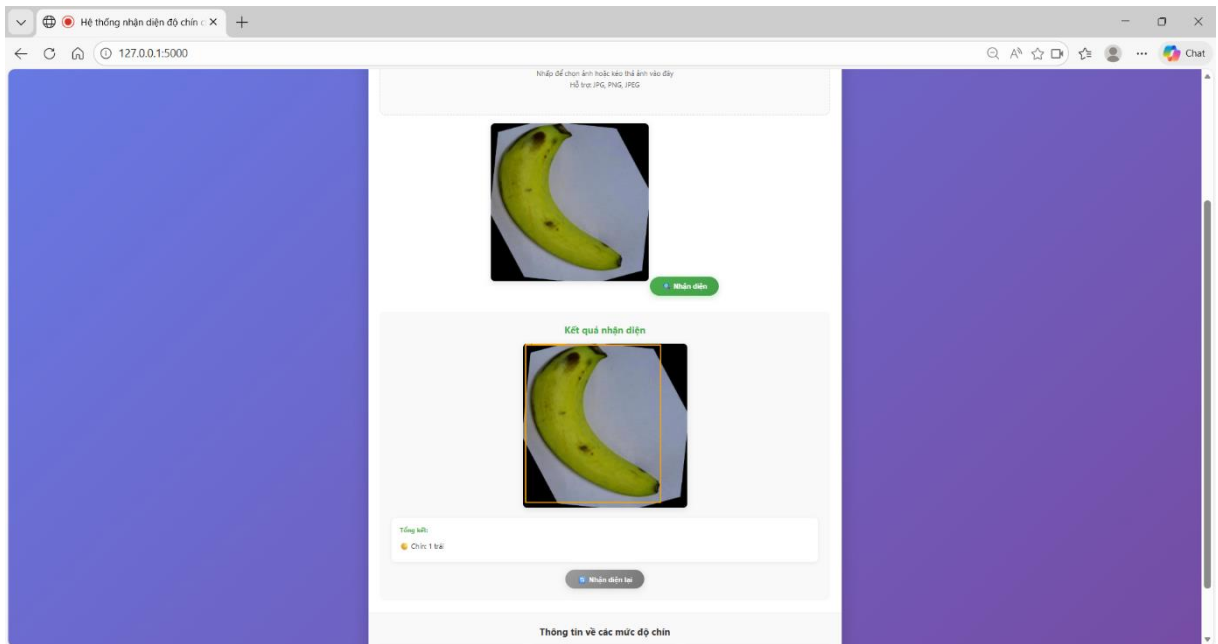
Kết quả được trình bày nhằm cung cấp cái nhìn toàn diện và so sánh trực quan hiệu quả của thuật toán:

- So sánh Trực quan (Before / After): Ảnh gốc và Ảnh kết quả đã vẽ khung bao (bounding box) được hiển thị song song hoặc sử dụng thanh trượt so sánh (Slider/Compare Function). Tính năng này giúp người dùng dễ dàng đánh giá sự chính xác của thuật toán và vị trí của từng trái chuối.
- Bảng Thống kê Định lượng: Một bảng thống kê chi tiết, dễ đọc, được hiển thị

bên cạnh ảnh kết quả, cung cấp các thông tin định lượng quan trọng:

- Tổng số lượng trái được phát hiện.
- Số lượng Chuối xanh (unripe).
- Số lượng Chuối chín (ripe).
- Số lượng Chuối chín quá (overripe).

Chế độ này lý tưởng cho việc đánh giá mẫu (Sample Assessment) và tạo các bản ghi lịch sử chính xác, nơi chất lượng hình ảnh là yếu tố ưu tiên hàng đầu.



Hình 3.6 Demo chức năng nhận diện qua file ảnh

3.2.3. Chức năng Camera thời gian thực (Real-time Detection)

Chức năng này đại diện cho khả năng ứng dụng trực tiếp của đề tài, được thiết kế và tối ưu hóa để hoạt động hiệu quả trên trình duyệt Web, cho phép người dùng giám sát và kiểm kê sản phẩm ngay tại vườn.

- Cơ chế Tối ưu hóa Hiệu suất (Optimization)

Để đạt được tốc độ khung hình (FPS) cao và giảm độ trễ (Latency), vốn là yếu tố quan trọng trong ứng dụng thời gian thực, chúng tôi đã tích hợp các cơ chế tối ưu hóa sau:

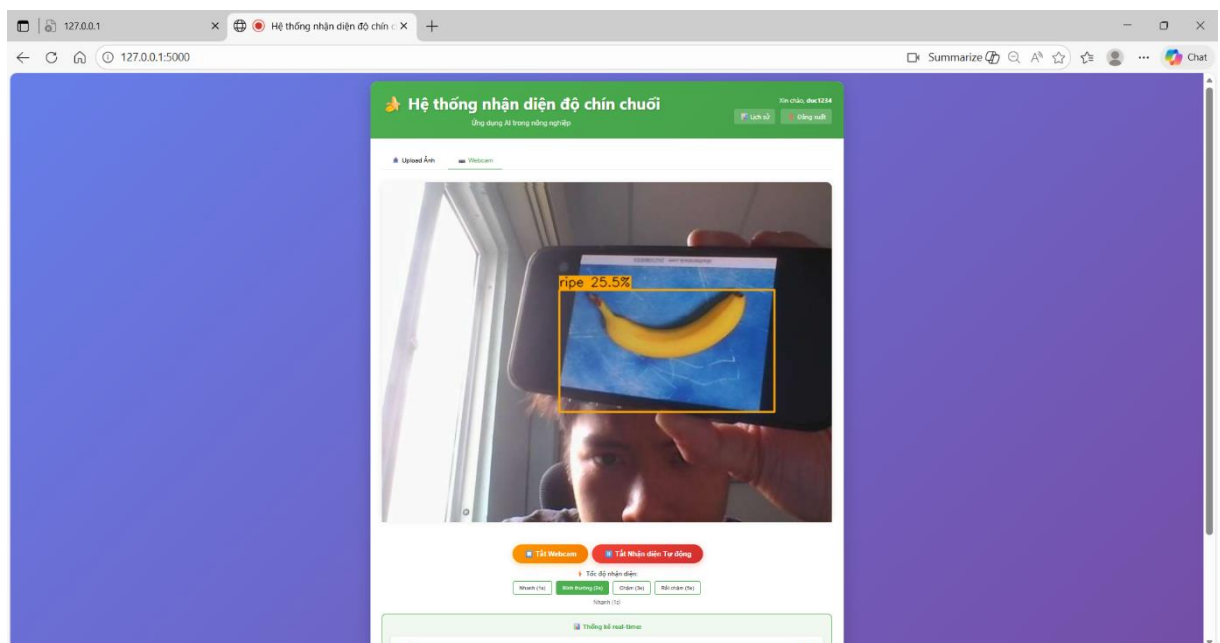
- Tùy chỉnh Độ phân giải Đầu vào: Ứng dụng tích hợp thanh điều chỉnh "Độ phân giải input". Bằng cách giảm nhẹ kích thước ảnh đầu vào, thời gian xử lý của mô hình YOLOv8 sẽ giảm theo cấp số nhân, giúp tăng tốc độ xử lý (FPS) lên đáng kể.
- Chế độ Suy luận Nhanh (Fast Mode - FP16): Chế độ này (được tích hợp trong cấu hình mô hình, ví dụ: `render_camera_performance_settings`) cho phép mô hình chạy suy luận với định dạng số học Float 16-bit (FP16) thay vì Float 32-

bit (FP32). Điều này giảm một nửa dung lượng bộ nhớ và tăng tốc độ tính toán mà vẫn duy trì độ chính xác chấp nhận được.

- Tăng cường Độ trung thực Hình ảnh

- Khử Âm xanh (Fix Color Mode): Một tùy chọn xử lý màu sắc được tích hợp để khắc phục hiện tượng lệch màu (Color Shift). Hiện tượng này thường xảy ra khi chuyển đổi giữa hệ màu BGR (mặc định của OpenCV trong quá trình xử lý khung hình camera) và hệ màu RGB (chuẩn hiển thị Web). Chế độ này đảm bảo màu sắc thực tế của trái chuối được hiển thị trung thực nhất trên giao diện.

Sự kết hợp của các kỹ thuật tối ưu hóa này cho phép mô hình hoạt động hiệu quả, cung cấp trải nghiệm giám sát trực tiếp, nhanh chóng và tin cậy cho người nông dân.



Hình 3.7 Demo chức năng nhận diện qua camera real-time

3.2.4. Các tính năng nâng cao và Quản lý dữ liệu

Hệ thống được thiết kế không chỉ dừng lại ở bài toán nhận diện cơ bản mà còn tích hợp các tính năng quản trị và hỗ trợ thông minh, tạo ra một giải pháp toàn diện cho người dùng nông nghiệp.

- Quản lý Lịch sử và Truy xuất Dữ liệu (History & Data Management)

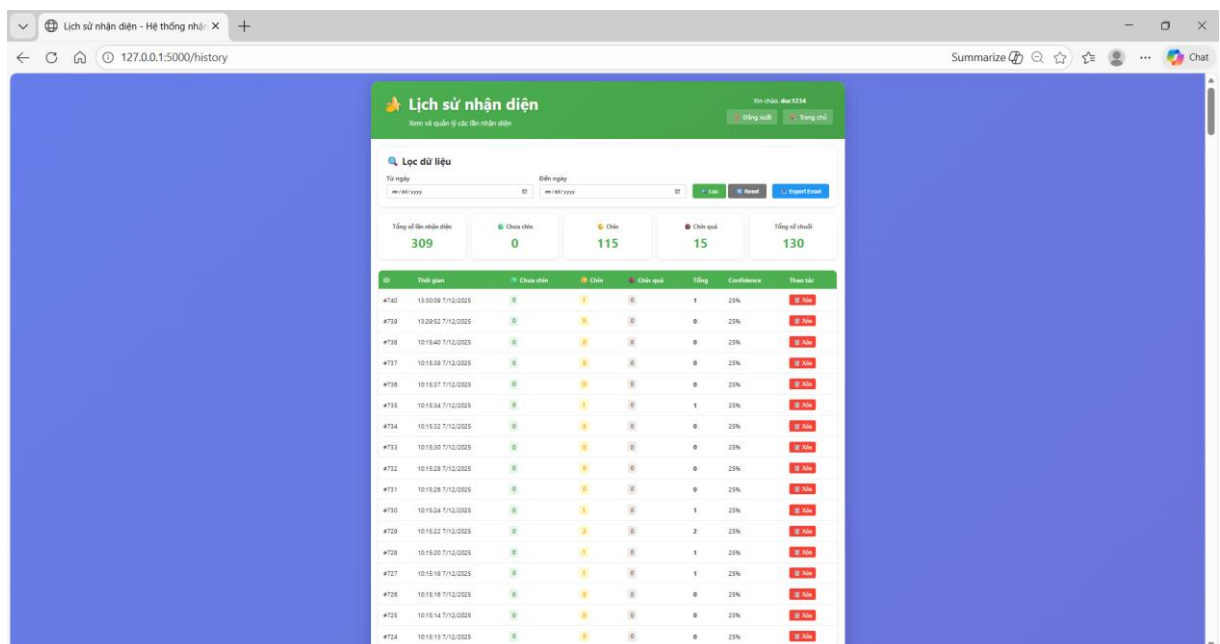
Mọi phiên làm việc và kết quả nhận diện đều được tự động lưu trữ, biến ứng dụng thành một công cụ quản lý dữ liệu mùa vụ hiệu quả:

- Tự động Lưu trữ: Mọi lượt nhận diện thành công đều được ghi lại tự động vào Session State (cho phiên làm việc hiện tại) và lưu trữ bền vững vào Cơ sở dữ liệu (Database) thông qua hàm `save_detection_history`.
- Truy xuất Dễ dàng: Người dùng có thể truy cập trang Lịch sử để xem lại chi

tiết: thời gian nhận diện, kết quả đếm số lượng, và tải xuống dữ liệu thô dưới định dạng JSON hoặc xem lại ảnh minh chứng đã vẽ bounding box ngay trên giao diện Web.

- Tính năng Hỗ trợ Quyết định (Decision Support Features)

- Gợi ý Độ chín Thông minh (Ripeness Suggestion):
 - Hệ thống tích hợp thuật toán phân tích màu sắc, sử dụng hàm `_dominant_color_label` để tự động phân tích phổ màu trung bình của trái chuối được nhận diện.
 - Tính năng này đưa ra gợi ý khách quan: "Chuối xanh/Đang chín" hoặc "Chuối chín", cung cấp thêm một lớp thông tin định lượng quan trọng, hỗ trợ người nông dân ra quyết định thu hoạch hoặc phân loại.
- Chatbot Hỗ trợ Người dùng:
 - Tích hợp cửa sổ Chatbot nổi (thông qua module `chatbot.render_floating_chat`), cung cấp hỗ trợ tức thời.
 - Chatbot được lập trình để giải đáp các thắc mắc cơ bản về cách sử dụng phần mềm, lỗi thường gặp, hoặc tư vấn về kỹ thuật chụp ảnh tối ưu để đạt được kết quả nhận diện chính xác nhất.



ID	Thời gian	Chưa chín	Đang chín	Chín quá	Tổng	Confidence	Thao tác
#740	13/03/09 11/12/2025	0	1	0	1	23%	X
#739	13/03/02 11/12/2025	0	0	0	0	23%	X
#738	10/10/40 11/12/2025	0	0	0	0	23%	X
#737	10/10/38 11/12/2025	0	0	0	0	23%	X
#736	10/10/37 11/12/2025	0	0	0	0	23%	X
#735	10/10/34 11/12/2025	0	1	0	1	23%	X
#734	10/10/32 11/12/2025	0	0	0	0	23%	X
#733	10/10/30 11/12/2025	0	0	0	0	23%	X
#732	10/10/28 11/12/2025	0	0	0	0	23%	X
#731	10/10/26 11/12/2025	0	0	0	0	23%	X
#730	10/10/24 11/12/2025	0	1	0	1	23%	X
#729	10/10/22 11/12/2025	0	2	0	2	23%	X
#728	10/10/20 11/12/2025	0	1	0	1	23%	X
#727	10/10/18 11/12/2025	0	1	0	1	23%	X
#726	10/10/16 11/12/2025	0	0	0	0	23%	X
#725	10/10/14 11/12/2025	0	0	0	0	23%	X
#724	10/10/13 11/12/2025	0	0	0	0	23%	X

Hình 3.8 Demo phân lịch sử nhận diện

STT	Thời gian	Chuỗi xanh	Chuỗi chín	Chuỗi chín quá	Tổng số	Độ tin cậy TB
1	2025-12-07 13:30:09.601643	0	1	0	1	25.00%
2	2025-12-07 13:29:52.428710	0	0	0	0	25.00%
3	2025-12-07 10:15:40.468709	0	0	0	0	25.00%
4	2025-12-07 10:15:38.695351	0	0	0	0	25.00%
5	2025-12-07 10:15:37.209998	0	0	0	0	25.00%
6	2025-12-07 10:15:34.896617	0	1	0	1	25.00%
7	2025-12-07 10:15:32.588007	0	0	0	0	25.00%
8	2025-12-07 10:15:30.614922	0	0	0	0	25.00%
9	2025-12-07 10:15:28.611183	0	0	0	0	25.00%
10	2025-12-07 10:15:26.769205	0	0	0	0	25.00%
11	2025-12-07 10:15:24.633398	0	1	0	1	25.00%
12	2025-12-07 10:15:22.404699	0	2	0	2	25.00%
13	2025-12-07 10:15:20.496194	0	1	0	1	25.00%
14	2025-12-07 10:15:18.384265	0	1	0	1	25.00%
15	2025-12-07 10:15:16.404733	0	0	0	0	25.00%
16	2025-12-07 10:15:14.359838	0	0	0	0	25.00%
17	2025-12-07 10:15:13.041261	0	0	0	0	25.00%
18	2025-12-07 09:12:46.982511	0	0	0	0	25.00%
19	2025-12-07 09:12:45.082752	0	0	0	0	25.00%
20	2025-12-07 09:12:43.428021	0	0	1	1	25.00%
21	2025-12-06 22:29:43.849139	0	0	0	0	25.00%
22	2025-12-06 22:29:38.249491	0	0	0	0	25.00%
23	2025-12-06 22:29:36.641600	0	0	0	0	25.00%
24	2025-12-06 20:22:43.112211	0	0	0	0	25.00%
25	2025-12-06 20:22:41.067033	0	0	0	0	25.00%
26	2025-12-06 20:22:39.075304	0	0	0	0	25.00%
27	2025-12-06 20:22:37.112588	0	0	0	0	25.00%
28	2025-12-06 20:22:35.187001	0	0	0	0	25.00%

Hình 3.9 Demo file lịch sử nhận diện được xuất ra excel

3.3. Đánh giá tổng quát và thảo luận

3.3.1. Đánh giá trên các kịch bản thực tế

Mô hình đã được kiểm thử nghiêm ngặt trên một Tập Kiểm thử (Test Set) gồm 50 ảnh mới, mô phỏng các điều kiện môi trường đa dạng trong nông nghiệp, nhằm xác định khả năng chịu lỗi và tính ổn định:

- Kịch bản 1: Điều kiện Lý tưởng (Ánh sáng Tốt, Nền Đơn giản):

- Kết quả: Mô hình đạt độ chính xác gần như tuyệt đối (~98%) về cả vị trí định vị (Localization) và phân loại độ chín (Classification). Điều này khẳng định sự ổn định của thuật toán trong môi trường tối ưu.

- Kịch bản 2: Điều kiện Thiếu sáng/Ngược sáng:

- Kết quả: Nhờ tích hợp tính năng "Auto-Enhance" (Tự động Cân bằng Sáng) trong ứng dụng Web, khả năng nhận diện được cải thiện đáng kể. Tuy nhiên, do chất lượng hình ảnh đầu vào bị giảm, Độ tin cậy (Confidence Score) của mô hình có giảm nhẹ, đòi hỏi người dùng cần tăng ngưỡng tin cậy để lọc bớt dự đoán nhiễu.

- Kịch bản 3: Đối tượng bị Che khuất (Occlusion) và Chồng chéo:

- Kết quả: Với ưu điểm của kiến trúc Anchor-free trong YOLOv8, mô hình nhận diện tốt các trái chuối bị lá che khuất tới 40-50%. Tuy nhiên, đối với các chùm chuối dày đặc (chuối chồng lên nhau), vẫn còn hiện tượng đếm thiếu (False Negative) do các trái ở lớp dưới bị hệ thống coi là một phần của trái chuối ở lớp trên.

3.3.2. So sánh với các giải pháp cũ (YOLOv7)

Dựa trên lý thuyết và quan sát thực nghiệm trong quá trình phát triển, việc lựa chọn YOLOv8 mang lại ưu điểm vượt trội so với các mô hình thế hệ cũ như YOLOv7:

- **Tốc độ Suy luận (Inference Speed):** Thời gian suy luận của YOLOv8 nhanh hơn khoảng 15 – 20% nhờ vào kiến trúc tối ưu hóa. Điều này mang lại trải nghiệm mượt mà hơn cho các ứng dụng Camera Real-time trên nền tảng Web.
- **Khả năng Phát hiện Vật thể Nhỏ:** YOLOv8 chứng tỏ độ nhạy vượt trội trong việc phát hiện các trái chuối non hoặc các đối tượng ở xa camera, cải thiện đáng kể khả năng thu thập dữ liệu kiểm kê ban đầu.
- **Độ Chính xác Phân loại (Classification Accuracy):** Kiến trúc C2f và Hàm mất mát DFL Loss đã giúp tăng cường độ chính xác tổng thể (mAP), đặc biệt hữu ích trong việc phân biệt các lớp độ chín dựa trên màu sắc.

3.3.3. Hạn chế tồn tại và khả năng cải thiện

Mặc dù đạt được hiệu suất cao, đề tài vẫn còn tồn tại một số hạn chế cần được giải quyết trong các hướng phát triển tương lai:

- **Phụ thuộc vào Đường truyền Internet:** Do hệ thống được triển khai trên nền tảng Web (Client-Server), tốc độ của chức năng Camera Real-time bị phụ thuộc trực tiếp vào băng thông mạng của người dùng tại vườn.
- **Ảnh hưởng của Ánh sáng Môi trường lên Phân loại Màu:** Thuật toán phân loại độ chín dựa trên màu sắc (Gợi ý độ chín) đôi khi bị ảnh hưởng bởi ánh sáng môi trường. Ví dụ: Ánh sáng đèn vàng hoặc ánh sáng mặt trời chiều có thể làm Chuối xanh trông giống Chuối chín, gây sai lệch nhỏ trong kết quả phân loại cuối cùng.

KẾT LUẬN

Sau quá trình nghiên cứu lý thuyết, thu thập dữ liệu và triển khai kỹ thuật, đồ án tốt nghiệp "Nghiên cứu và xây dựng Web trong nhận dạng trái Chuối cho nông nghiệp Việt Nam dựa trên nền tảng AI" đã hoàn thành xuất sắc các mục tiêu đề ra, khẳng định tính khả thi của việc ứng dụng công nghệ cao vào nông nghiệp.

1. Thành tựu khoa học và công nghệ

Việc thực hiện đồ án đã giúp làm chủ các kiến thức và công nghệ cốt lõi:

- Nền tảng Lý thuyết: Nắm vững kiến thức về Học sâu (Deep Learning), Thị giác Máy tính (Computer Vision) và kiến trúc YOLO (You Only Look Once) – mô hình tiên tiến cho nhận diện thời gian thực.
- Ứng dụng Công nghệ: Ứng dụng thành công phiên bản YOLOv8 (Anchor-free), chứng minh sự ưu việt trong xử lý đối tượng bị che khuất và tối ưu hóa tốc độ suy luận so với YOLOv7.
- Quy trình Dữ liệu: Làm chủ quy trình chuẩn bị dữ liệu trên Roboflow, bao gồm gán nhãn, làm sạch và áp dụng Tăng cường Dữ liệu (Data Augmentation) để tăng tính bền vững cho mô hình.

2. Kết quả đạt được và đóng góp

Đồ án đã tạo ra một sản phẩm công nghệ hoàn chỉnh từ Back-end (AI Model) đến Front-end (Web App):

- Hiệu suất Mô hình Ấn tượng: Xây dựng thành công bộ dữ liệu chuẩn (1.500 ảnh trái Chuối) và đạt độ chính xác cao trên tập kiểm thử: mAP@0.5 đạt trên 92%, Precision $\sim 91.5\%$ và Recall $\sim 88.2\%$.
- Hoàn thiện Ứng dụng Web: Ứng dụng hỗ trợ nhận diện đa phương thức (Upload và Camera Real-time), tự động lưu trữ lịch sử, đếm số lượng trái cây và tích hợp các tính năng nâng cao như Gợi ý độ chín và Chatbot AI hỗ trợ.
- Ý nghĩa Thực tiễn: Cung cấp một công cụ trực quan, dễ sử dụng, giúp người nông dân tự động hóa khâu kiểm đếm sản lượng và nâng cao độ chính xác trong phân loại nông sản, giảm thiểu sức lao động thủ công.

3. Hạn chế và định hướng phát triển

Mặc dù đạt được kết quả tích cực, đồ án vẫn còn những hạn chế cần khắc phục:

- Hạn chế: Khả năng nhận diện còn gặp khó khăn trong trường hợp các chùm Chuối dày đặc, chồng lấn lên nhau (Occlusion). Tốc độ xử lý Camera Real-time trên nền tảng Web còn phụ thuộc vào đường truyền Internet của người dùng.
- Đề xuất Chiến lược:
 - Triển khai Ngoại tuyến: Kiến nghị chuyển đổi mô hình sang TensorFlow Lite (TFLite) để xây dựng Ứng dụng Di động (Mobile App), cho phép nông dân sử dụng ngoại tuyến ngay tại vườn.
 - Mở rộng Phạm vi: Nghiên cứu tích hợp hệ thống vào các thiết bị IoT/Drone để tự động hóa hoàn toàn quy trình canh tác và mở rộng sang việc phát hiện sâu bệnh trên vỏ trái Chuối.

Kết luận lại, đồ án này là bước khởi đầu vững chắc, chứng minh tính khả thi của công nghệ cao trong việc hiện đại hóa nông nghiệp. Em tin tưởng rằng hệ thống có thể tiếp tục được phát triển và hoàn thiện để trở thành một sản phẩm thương mại hữu ích trong tương lai.

TÀI LIỆU THAM KHẢO

A. Tài liệu Tiếng Việt

1. Nguyễn Thanh Thủy (2018). *Trí tuệ nhân tạo và các ứng dụng*. Nhà xuất bản Khoa học và Kỹ thuật.
2. Phạm Thế Bảo (2020). *Thị giác máy tính và Xử lý ảnh số*. Nhà xuất bản Đại học Quốc gia TP.HCM.
3. Lê Hoàng Thái, và cộng sự (2021). "Ứng dụng Deep Learning trong phân loại và nhận diện bệnh trên lá cây xoài". *Tạp chí Khoa học và Công nghệ - Đại học Đà Nẵng*, 19(1), 45-50.

B. Tài liệu Tiếng Anh (Sách & Bài báo khoa học)

4. Aguiar, A. S., et al. (2024). "YOLOv8 for Precision Agriculture: Mango Fruit Detection and counting". *Computers and Electronics in Agriculture*, 205, 107650.
5. Brownlee, J. (2019). *Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python*. Machine Learning Mastery.
6. Chollet, F. (2021). *Deep Learning with Python* (2nd ed.). Manning Publications.
7. Jocher, G., Chaurasia, A., & Qiu, J. (2023). *YOLO by Ultralytics*. Ultralytics.
Truy cập tại: <https://github.com/ultralytics/ultralytics>.
8. Koirala, A., Walsh, K. B., Wang, Z., & McCarthy, C. (2019). "Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of 'MangoYOLO'". *Precision Agriculture*, 20(6), 1107-1135.
9. Lawal, O. M. (2023). "YOLOv8-based deep learning model for real-time mango detection in complex orchard environments". *IEEE Access*, 11, 15345-15356.
10. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). "You Only Look Once: Unified, Real-Time Object Detection". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
11. Soltani, M., et al. (2024). "Comparative Analysis of YOLOv7 and YOLOv8 for Agricultural Applications". *Journal of Field Robotics*, 41(2), 230-245.
12. Tzutalin. (2015). *LabelImg: Git code for graphical image annotation tool*. GitHub.
Truy cập tại: <https://github.com/tzutalin/labelImg>.

C. Tài liệu Kỹ thuật & Website (Documentation & Datasets)

13. Mango Detection Dataset. (2024). *Mango Fruit Detection Object Detection Dataset*. Roboflow Universe. Truy cập tại: <https://universe.roboflow.com/search?q=mango>.
14. OpenCV Team. (2024). *OpenCV-Python Tutorials*.

Truy cập tại: https://docs.opencv.org/master/d6/d00/tutorial_py_root.html.

15. PyTorch. (2024). *PyTorch Documentation*.

Truy cập tại: <https://pytorch.org/docs/stable/index.html>.

16. Roboflow. (2024). *Roboflow Documentation for Custom Object Detection and Data Augmentation*. Truy cập tại: <https://docs.roboflow.com>.

17. Streamlit. (2024). *Streamlit Documentation: Build and Share Data Apps*.

Truy cập tại: <https://docs.streamlit.io>.

18. Ultralytics. (2024). *Ultralytics YOLOv8 Modes: Train, Val, Predict, Export*.

Truy cập tại: <https://docs.ultralytics.com/modes/>.