**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: XỬ LÝ ẢNH**

**ĐỀ TÀI: NHẬN DIỆN HOA QUẢ**

**Giáo viên hướng dẫn: ThS. Lê Trung Hiếu**

**Sinh viên thực hiện:**  **Nhóm 4**

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Mã sv** | **Họ và tên** | **Lớp** |
| 1 | 1571020088 | Phùng Đăng Hậu | CNTT 15-02 |
| 2 | 1571020057 | Nguyễn Tiến Đạt | CNTT 15-03 |
| 3 | 1571020068 | Nguyễn Minh Đức | CNTT 15-03 |

**Hà Nội, năm 2025**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: XỬ LÝ ẢNH**

**ĐỀ TÀI: NHẬN DIỆN HOA QUẢ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Mã Sinh Viên** | **Họ và Tên** | **Ngày Sinh** | **Điểm** | |
| **Bằng Số** | **Bằng Chữ** |
| **1** | **1571020088** | **Phùng Đăng Hậu** | **07/10/2003** |  |  |
| **2** | **1571020057** | **Nguyễn Tiến Đạt** | **02/02/2003** |  |  |
| **3** | **1571020068** | **Nguyễn Minh Đức** | **09/04/2003** |  |  |

**CÁN BỘ CHẤM THI 1 CÁN BỘ CHẤM THI 2**

**Hà Nội, năm 202**

# **LỜI NÓI ĐẦU**

Trong những năm gần đây, trí tuệ nhân tạo (AI) và thị giác máy tính (Computer Vision) đã trở thành những lĩnh vực cốt lõi thúc đẩy sự đổi mới trong nhiều ngành công nghiệp, đặc biệt là trong nông nghiệp thông minh, tự động hóa và bán lẻ. Nhận diện đối tượng là một trong những bài toán then chốt của thị giác máy, cho phép máy tính có khả năng "nhìn thấy" và "hiểu được" hình ảnh trong thế giới thực.

Trong bối cảnh đó, việc xây dựng một hệ thống nhận diện hoa quả tự động không chỉ mang lại giá trị ứng dụng cao trong kiểm kê, phân loại sản phẩm nông nghiệp mà còn đóng vai trò quan trọng trong việc giảm thiểu chi phí lao động, tăng độ chính xác và nâng cao hiệu quả sản xuất.

Với mục tiêu áp dụng các phương pháp học sâu tiên tiến, đặc biệt là mô hình YOLOv8 – một kiến trúc hiện đại và tối ưu trong bài toán nhận diện thời gian thực – nhóm thực hiện đã tiến hành nghiên cứu, thiết kế và xây dựng một hệ thống có khả năng nhận diện chính xác các loại hoa quả phổ biến như táo, chuối, cam, xoài.

Nội dung báo cáo bao gồm toàn bộ quá trình thực hiện đề tài từ khảo sát công nghệ, xây dựng bộ dữ liệu, huấn luyện mô hình cho đến đánh giá hiệu quả và đề xuất hướng phát triển trong tương lai. Nhóm hy vọng rằng kết quả nghiên cứu này không chỉ là một minh chứng cho khả năng ứng dụng AI trong thực tiễn mà còn là bước đệm để phát triển các giải pháp tự động hóa hiệu quả trong lĩnh vực nông nghiệp và tiêu dùng thông minh.

# **LỜI CẢM ƠN**

Trước tiên, nhóm 4 chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy Lê Trung Hiếu, người đã tận tình hướng dẫn, chỉ bảo chúng em trong suốt quá trình thực hiện báo cáo thực tập này. Thầy không chỉ giúp chúng em hiểu rõ hơn về chuyên môn, cách tiếp cận và giải quyết vấn đề mà còn truyền động lực để chúng em không ngừng nỗ lực, hoàn thiện bản thân.

Nhóm sinh viên thực hiện đề tài “Nhận diện hoa quả bằng mô hình YOLOv8” xin gửi lời cảm ơn chân thành và sâu sắc đến Thầy Lê Trung Hiếu – người đã trực tiếp hướng dẫn, tận tình chỉ bảo và định hướng cho nhóm trong suốt quá trình thực hiện đề tài.

Nhờ sự hướng dẫn tận tâm, kiến thức chuyên môn sâu rộng và tinh thần hỗ trợ nhiệt tình của thầy, nhóm đã có cơ hội tiếp cận với những kiến thức hiện đại về trí tuệ nhân tạo, học sâu và thị giác máy tính, từ đó ứng dụng thành công vào việc xây dựng một hệ thống nhận diện hoa quả hiệu quả.

Một lần nữa, nhóm xin trân trọng cảm ơn thầy và mong rằng sẽ tiếp tục nhận được sự đồng hành, định hướng của thầy trong những chặng đường học tập và nghiên cứu tiếp theo.

# 

# **MỤC LỤC**

[**CHƯƠNG 1: GIỚI THIỆU** 9](#_Toc198122599)

[**1.1.** **Giới thiệu đề tài** 9](#_Toc198122600)

[**1.2.** **Mục tiêu nghiên cứu** 10](#_Toc198122601)

[**1.3.** **Lý do chọn đề tài** 10](#_Toc198122602)

[**1.4.** **Lý do lựa chọn Yolo V8** 10](#_Toc198122603)

[**1.5.** **phạm vi nghiên cứu** 10](#_Toc198122604)

[**1.6.** **Đối tượng và ý nghĩa nghiên cứu** 11](#_Toc198122605)

[*1.6.1.* *Đối tượng:* 11](#_Toc198122606)

[*1.6.2.* *Ý nghĩa thực tiễn:* 11](#_Toc198122607)

[**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT** 12](#_Toc198122608)

[**2.1. Xử lý ảnh và thị giác máy tính** 12](#_Toc198122609)

[**2.2. Nhận diện vật thể** 13](#_Toc198122610)

[**2.3. Công nghệ sử dụng** 14](#_Toc198122611)

[**CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG** 16](#_Toc198122612)

[**3.1. Kiến trúc hệ thống Hệ thống nhận diện hoa quả sử dụng mô hình YOLOv8 được thiết kế với các thành phần chính như sau:** 16](#_Toc198122613)

[**3.2. Quy trình thực hiện** 16](#_Toc198122620)

[**CHƯƠNG 4: THỰC NGHIỆM VÀ KẾT QUẢ** 18](#_Toc198122621)

[**4.1. Dữ liệu huấn luyện** 18](#_Toc198122622)

[**4.2. Thông số huấn luyện** 29](#_Toc198122623)

[**CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 31](#_Toc198122624)

[**5.1. Kết luận** 31](#_Toc198122625)

[**5.2. Hướng phát triển** 31](#_Toc198122626)

[**KẾT LUẬN** 32](#_Toc198122627)

[**DANH MỤC TÀI LIỆU THAM KHẢO** 33](#_Toc198122628)

# **MỤC LỤC HÌNH ẢNH**

# **CHƯƠNG 1: GIỚI THIỆU**

* 1. **Giới thiệu đề tài**

Trong thời đại công nghệ số hiện nay, thị giác máy tính (Computer Vision) đã trở thành một trong những lĩnh vực then chốt của trí tuệ nhân tạo (AI), góp phần thay đổi cách mà máy móc nhận thức và tương tác với thế giới thực. Trong đó, xử lý ảnh đóng vai trò quan trọng trong việc trích xuất và phân tích thông tin hình ảnh nhằm phục vụ cho các ứng dụng như giám sát an ninh, y tế, xe tự lái, nhận diện khuôn mặt, và đặc biệt là nhận diện vật thể.

Nhận diện vật thể là một bài toán nền tảng trong thị giác máy tính, cho phép hệ thống xác định chính xác sự xuất hiện, vị trí và nhãn của các đối tượng trong một hình ảnh hoặc đoạn video. Nhờ đó, máy có thể hiểu được môi trường xung quanh và đưa ra những phản ứng phù hợp. Trong nông nghiệp, công nghệ nhận diện vật thể mang lại giải pháp tự động hóa hiệu quả như phân loại trái cây, giám sát mùa vụ, và phát hiện sâu bệnh. Việc ứng dụng mô hình nhận diện vào nhận dạng hoa quả không chỉ giúp tối ưu hoá quy trình sản xuất mà còn mang lại nhiều giá trị thực tiễn trong phân phối và bán hàng tự động.

Trong đề tài này, nhóm nghiên cứu lựa chọn sử dụng mô hình YOLO (You Only Look Once) – một trong những mô hình phát hiện vật thể tiên tiến và phổ biến nhất hiện nay – để triển khai bài toán nhận diện hoa quả. Với tốc độ nhanh và độ chính xác cao, YOLO đặc biệt phù hợp với các ứng dụng cần xử lý thời gian thực. Qua đề tài, nhóm hướng tới việc xây dựng một hệ thống đơn giản nhưng hiệu quả để phát hiện và phân loại các loại trái cây như táo, chuối, cam, xoài,… thông qua hình ảnh.

* 1. **Mục tiêu nghiên cứu**
* Tìm hiểu và ứng dụng mô hình YOLO trong bài toán nhận diện vật thể.
* Xây dựng bộ dữ liệu hoa quả có gán nhãn đầy đủ.
* Triển khai mô hình nhận diện hình ảnh các loại hoa quả.
* Đánh giá hiệu quả của mô hình trên tập dữ liệu thử nghiệm.
  1. **Lý do chọn đề tài**

Trong thời đại trí tuệ nhân tạo phát triển mạnh mẽ, nhận diện vật thể bằng xử lý ảnh đang được ứng dụng rộng rãi trong nhiều lĩnh vực, đặc biệt là nông nghiệp thông minh. Đề tài “Ứng dụng YOLO trong nhận diện hoa quả” được chọn vì có tính thực tiễn cao, giúp tự động hóa việc phân loại trái cây, tiết kiệm thời gian, nhân lực, đồng thời giúp nhóm sinh viên hiểu rõ hơn về mô hình YOLO và quy trình xây dựng hệ thống nhận diện hình ảnh.

* 1. **Lý do lựa chọn Yolo V8**

YOLOv8 là phiên bản mới nhất trong dòng mô hình YOLO, nổi bật với độ chính xác cao, tốc độ xử lý nhanh và khả năng triển khai linh hoạt. So với các phiên bản trước, YOLOv8 cải tiến kiến trúc mạng, hỗ trợ tốt hơn cho các thiết bị edge và dễ dàng tùy chỉnh cho nhiều bài toán thực tế. Vì vậy, nhóm lựa chọn YOLOv8 nhằm đảm bảo hiệu quả cao nhất trong việc nhận diện nhanh và chính xác các loại trái cây qua hình ảnh.

* 1. **phạm vi nghiên cứu**

Đề tài tập trung vào việc triển khai mô hình YOLOv8 để nhận diện một số loại trái cây phổ biến như táo, chuối, cam, xoài… thông qua hình ảnh. Phạm vi nghiên cứu bao gồm: thu thập và gán nhãn dữ liệu hình ảnh, tiền xử lý ảnh, huấn luyện mô hình YOLOv8, đánh giá độ chính xác và thử nghiệm nhận diện trên ảnh tĩnh và video đơn giản. Đề tài không đi sâu vào tối ưu phần cứng, không so sánh với tất cả các mô hình khác và chưa triển khai trên thiết bị thực tế

* 1. **Đối tượng và ý nghĩa nghiên cứu**
     1. *Đối tượng:*
* Đối tượng của đề tài là hình ảnh các loại trái cây và mô hình phát hiện vật thể YOLOv8 – một mô hình học sâu được ứng dụng để nhận diện và phân loại trái cây qua ảnh.
  + 1. *Ý nghĩa thực tiễn:*
* Đề tài góp phần ứng dụng công nghệ thị giác máy tính vào lĩnh vực nông nghiệp thông minh, hỗ trợ tự động hóa quy trình phân loại sản phẩm. Đồng thời, giúp sinh viên tiếp cận thực tiễn việc xây dựng hệ thống xử lý ảnh, từ khâu dữ liệu đến huấn luyện và triển khai mô hình, phục vụ cho các ứng dụng trong công nghiệp, thương mại và giáo dục

# **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

## **2.1. Xử lý ảnh và thị giác máy tính**

Xử lý ảnh (Image Processing) là quá trình áp dụng các kỹ thuật toán học và lập trình để phân tích, biến đổi hoặc trích xuất thông tin từ ảnh số. Mục tiêu chính là cải thiện chất lượng ảnh, làm nổi bật đặc điểm quan trọng, hoặc chuẩn bị ảnh cho các ứng dụng cụ thể như nhận diện vật thể, phân loại, theo dõi, và phân đoạn hình ảnh.

Ảnh số là tập hợp các điểm ảnh (pixels) được sắp xếp dưới dạng ma trận. Mỗi pixel mang giá trị thể hiện mức độ sáng hoặc màu sắc. Ví dụ, một ảnh có kích thước 640x640 sẽ bao gồm:

* 640 × 640 = 409.600 pixel

Các loại ảnh phổ biến trong xử lý ảnh số:

* Ảnh xám (Grayscale): Mỗi pixel có giá trị từ 0 (đen) đến 255 (trắng). Đây là loại ảnh phổ biến trong các tác vụ tiền xử lý vì đơn giản và tiết kiệm tài nguyên tính toán.
* Ảnh nhị phân (Binary): Mỗi pixel chỉ nhận giá trị 0 hoặc 1, tương ứng với đen và trắng. Loại ảnh này thường dùng trong phân đoạn và phát hiện biên.
* Ảnh màu (RGB): Mỗi pixel gồm ba giá trị đại diện cho ba kênh màu Đỏ (Red), Lục (Green) và Lam (Blue). Đây là dạng ảnh đầu vào phổ biến trong các ứng dụng thị giác máy tính.

Các bước cơ bản trong xử lý ảnh gồm:

1. Tiền xử lý (Preprocessing): Làm sạch ảnh, thay đổi kích thước, chuẩn hóa dữ liệu để phục vụ huấn luyện mô hình.
2. Phân đoạn ảnh (Segmentation): Chia ảnh thành các vùng nhỏ có ý nghĩa, giúp dễ nhận diện vật thể.
3. Trích xuất đặc trưng (Feature Extraction): Rút ra các đặc điểm quan trọng như hình dạng, màu sắc, cạnh viền…
4. Phân loại (Classification): Sử dụng các mô hình học máy hoặc học sâu để nhận diện đối tượng.

## **2.2. Nhận diện vật thể**

Nhận diện vật thể (Object Detection) là một bước tiến cao hơn so với phân loại ảnh (Image Classification). Thay vì chỉ xác định một nhãn cho toàn ảnh, nhận diện vật thể cho biết có những đối tượng gì, ở vị trí nào trong ảnh.

So sánh các tác vụ trong thị giác máy tính:

|  |  |
| --- | --- |
| **Tác vụ** | **Đầu ra** |
| Phân loại ảnh | |  | | --- | | Một nhãn cho toàn bộ ảnh | |
| Nhận diện vật thể | Nhãn + vị trí (bounding box) của từng đối tượng trong ảnh |
| Phân đoạn ảnh | Gán nhãn cho từng pixel trong ảnh (các đối tượng tách biệt chính xác) |

Một số mô hình phổ biến:

* CNN (Convolutional Neural Networks): Cơ sở cho hầu hết mô hình xử lý ảnh hiện nay.
* SSD (Single Shot Detector): Phát hiện vật thể nhanh, nhưng độ chính xác không cao bằng các mô hình mới hơn.
* Faster R-CNN: Độ chính xác cao, nhưng tốc độ chậm, khó dùng trong thời gian thực.
* YOLO (You Only Look Once): Mô hình nhận diện một bước (one-stage), nổi bật với tốc độ xử lý nhanh và độ chính xác tốt, đặc biệt phù hợp với ứng dụng thời gian thực như nhận diện hoa quả trong video.

## **2.3. Công nghệ sử dụng**

YOLO là mô hình phát hiện vật thể một bước, nghĩa là chỉ cần một lần quét toàn bộ ảnh để dự đoán vị trí và nhãn của tất cả vật thể. Điều này giúp YOLO trở thành mô hình lý tưởng cho các ứng dụng cần tốc độ cao như xe tự lái, giám sát an ninh, và nhận diện sản phẩm.

Nguyên lý hoạt động:

YOLO (You Only Look Once) hoạt động dựa trên cách tiếp cận chia ảnh đầu vào thành một lưới (grid), thường là lưới S×S. Mỗi ô lưới sẽ chịu trách nhiệm phát hiện các vật thể mà tâm của vật thể nằm trong ô đó. Với mỗi ô, mô hình sẽ dự đoán một số lượng cố định các bounding boxes (hộp giới hạn), đồng thời dự đoán:

* Tọa độ của hộp (bao gồm x, y là tọa độ trung tâm của vật thể, và w, h là chiều rộng, chiều cao của hộp)
* Độ tin cậy (confidence score) của mỗi hộp: thể hiện xác suất có vật thể trong hộp và mức độ chính xác của dự đoán vị trí.
* Xác suất thuộc về từng lớp (class probabilities): xác suất vật thể thuộc vào mỗi loại nhãn (ví dụ: người, xe, chó, v.v.)

Các phiên bản chính của YOLO:

* YOLOv1 → YOLOv4: Các phiên bản đầu giúp khẳng định tên tuổi YOLO với tốc độ xử lý cực nhanh.
* YOLOv5: Phổ biến nhất, dễ dùng với nhiều công cụ hỗ trợ, hỗ trợ triển khai linh hoạt.
* YOLOv7: Cải tiến mạnh về độ chính xác và tối ưu hóa tốt hơn cho GPU.
* YOLOv8: Phiên bản mới nhất, hỗ trợ tốt cho cả detection, segmentation và classification. Kiến trúc mạng nhẹ hơn, tốc độ nhanh hơn và chính xác cao hơn.

Ưu điểm của YOLOv8:

* Xử lý ảnh thời gian thực với tốc độ cao.
* Hiệu suất tốt trên ảnh độ phân giải thấp.
* Dễ triển khai, tích hợp vào các hệ thống sản xuất, bán hàng hoặc tự động hóa.

Hạn chế:

* Hiệu quả phát hiện vật thể nhỏ hoặc chồng chéo chưa tối ưu như một số mô hình hai bước.
* Phụ thuộc nhiều vào chất lượng dữ liệu và công cụ gán nhãn.

Các phiên bản chính của YOLO:

* YOLOv1 → YOLOv4: Các phiên bản đầu giúp khẳng định tên tuổi YOLO với tốc độ xử lý cực nhanh.
* YOLOv5: Phổ biến nhất, dễ dùng với nhiều công cụ hỗ trợ, hỗ trợ triển khai linh hoạt.
* YOLOv7: Cải tiến mạnh về độ chính xác và tối ưu hóa tốt hơn cho GPU.
* YOLOv8: Phiên bản mới nhất, hỗ trợ tốt cho cả detection, segmentation và classification. Kiến trúc mạng nhẹ hơn, tốc độ nhanh hơn và chính xác cao hơn.

Ưu điểm của YOLOv8:

* Xử lý ảnh thời gian thực với tốc độ cao.
* Hiệu suất tốt trên ảnh độ phân giải thấp.
* Dễ triển khai, tích hợp vào các hệ thống sản xuất, bán hàng hoặc tự động hóa.

Hạn chế:

* Hiệu quả phát hiện vật thể nhỏ hoặc chồng chéo chưa tối ưu như một số mô hình hai bước.
* Phụ thuộc nhiều vào chất lượng dữ liệu và công cụ gán nhãn.

.

# **CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

## **3.1. Kiến trúc hệ thống Hệ thống nhận diện hoa quả sử dụng mô hình YOLOv8 được thiết kế với các thành phần chính như sau:**

## Bộ thu thập dữ liệu (Dataset Collector): Thu thập hình ảnh các loại trái cây từ nhiều nguồn (trực tuyến hoặc thực tế), đảm bảo đa dạng góc chụp, ánh sáng và bối cảnh.

## Bộ gán nhãn dữ liệu (Annotation Tool): Sử dụng công cụ như LabelImg hoặc Roboflow để gán nhãn cho từng vật thể trong ảnh (bounding box và class).

## Tiền xử lý dữ liệu (Preprocessing): Chuyển đổi kích thước ảnh về chuẩn (thường là 640x640), chuẩn hóa pixel, và chia tập dữ liệu thành train/val/test.

## Huấn luyện mô hình (Training): Sử dụng YOLOv8 với các siêu tham số phù hợp (batch size, learning rate, epochs...) để học từ dữ liệu.

## Kiểm thử và đánh giá (Evaluation): Sử dụng các chỉ số như mAP (mean Average Precision), Precision, Recall để đánh giá độ chính xác của mô hình.

## Triển khai mô hình (Deployment): Tích hợp vào ứng dụng nhận diện ảnh tĩnh hoặc video.

## **3.2. Quy trình thực hiện**

1. Thu thập dữ liệu: Chụp hoặc tải hình ảnh hoa quả đa dạng (mỗi loại ~200–500 ảnh).
2. Gán nhãn: Gán nhãn cho mỗi đối tượng trong ảnh bằng bounding box và nhãn tương ứng.
3. Tiền xử lý: Chuyển ảnh về kích thước chuẩn, chia train/test, kiểm tra chất lượng dữ liệu.
4. Huấn luyện mô hình YOLOv8: Sử dụng framework như Ultralytics YOLO với file YAML cấu hình lớp và đường dẫn dữ liệu.
5. Đánh giá: Sử dụng tập test để đánh giá mô hình bằng chỉ số mAP@0.5, mAP@0.5:0.95.
6. Triển khai: Tạo giao diện demo hoặc ứng dụng thực tế để nhận diện hoa quả từ webcam hoặc video.

# **CHƯƠNG 4: THỰC NGHIỆM VÀ KẾT QUẢ**

## **4.1. Dữ liệu huấn luyện**

Sử dụng roboflow để gán nhãn và train dữ liệu bằng kaggle



Sử dụng dòng lệnh : ”!yolo task=detect mode=train model=yolov8n.pt data=/kaggle/working/Object-Detection-7/data.yaml epochs=100 imgsz=640 save\_period=20 #save sau 20 epoch” để train model.

Sau đó dùng Arduino để truyền dữ liệu vào trong cam Esp32 để nhận diện hoa quả và kết nối với flask để đẩy lên web để thống kê và tổng hợp số lượng quả hỏng hay tươi.

* **Code Arduino:**

#include "esp\_camera.h"

#include <WiFi.h>

#include <WebServer.h>

#include "esp\_http\_server.h"

// 🛜 Thay thông tin Wi-Fi tại đây

const char\* ssid = "test123";

const char\* password = "12345678";

// 🧠 Khai báo chân cho AI Thinker ESP32-CAM

#define PWDN\_GPIO\_NUM 32

#define RESET\_GPIO\_NUM -1

#define XCLK\_GPIO\_NUM 0

#define SIOD\_GPIO\_NUM 26

#define SIOC\_GPIO\_NUM 27

#define Y9\_GPIO\_NUM 35

#define Y8\_GPIO\_NUM 34

#define Y7\_GPIO\_NUM 39

#define Y6\_GPIO\_NUM 36

#define Y5\_GPIO\_NUM 21

#define Y4\_GPIO\_NUM 19

#define Y3\_GPIO\_NUM 18

#define Y2\_GPIO\_NUM 5

#define VSYNC\_GPIO\_NUM 25

#define HREF\_GPIO\_NUM 23

#define PCLK\_GPIO\_NUM 22

// ⚙️ Khởi tạo luồng MJPEG

esp\_err\_t stream\_handler(httpd\_req\_t \*req) {

camera\_fb\_t \*fb = NULL;

esp\_err\_t res = ESP\_OK;

res = httpd\_resp\_set\_type(req, "multipart/x-mixed-replace; boundary=frame");

if (res != ESP\_OK) return res;

while (true) {

fb = esp\_camera\_fb\_get();

if (!fb) {

Serial.println("Camera capture failed");

continue;

}

char part\_buf[64];

size\_t hlen = snprintf(part\_buf, sizeof(part\_buf),

"--frame\r\nContent-Type: image/jpeg\r\nContent-Length: %u\r\n\r\n", fb->len);

res = httpd\_resp\_send\_chunk(req, part\_buf, hlen);

if (res == ESP\_OK) res = httpd\_resp\_send\_chunk(req, (const char \*)fb->buf, fb->len);

if (res == ESP\_OK) res = httpd\_resp\_send\_chunk(req, "\r\n", 2);

esp\_camera\_fb\_return(fb);

if (res != ESP\_OK) break;

}

return res;

}

void startCameraServer() {

httpd\_config\_t config = HTTPD\_DEFAULT\_CONFIG();

httpd\_uri\_t uri\_handler = {

.uri = "/",

.method = HTTP\_GET,

.handler = stream\_handler,

.user\_ctx = NULL

};

httpd\_handle\_t server = NULL;

httpd\_start(&server, &config);

httpd\_register\_uri\_handler(server, &uri\_handler);

}

void setup() {

Serial.begin(115200);

Serial.setDebugOutput(true); // Bật debug để xem chi tiết lỗi Wi-Fi

// Kết nối Wi-Fi với timeout

WiFi.begin(ssid, password);

Serial.println("Connecting to Wi-Fi...");

unsigned long startAttemptTime = millis();

const unsigned long timeout = 20000; // Timeout sau 10 giây

while (WiFi.status() != WL\_CONNECTED && millis() - startAttemptTime < timeout) {

delay(500);

Serial.print(".");

}

// Kiểm tra kết nối Wi-Fi

if (WiFi.status() != WL\_CONNECTED) {

Serial.println("\nFailed to connect to Wi-Fi. Check credentials, signal, or band (2.4GHz).");

Serial.print("WiFi Status Code: ");

Serial.println(WiFi.status());

return; // Dừng chương trình nếu không kết nối được

}

Serial.println("");

Serial.println("WiFi connected");

Serial.print("ESP32-CAM IP Address: http://");

Serial.println(WiFi.localIP());

// Cấu hình camera

camera\_config\_t config;

config.ledc\_channel = LEDC\_CHANNEL\_0;

config.ledc\_timer = LEDC\_TIMER\_0;

config.pin\_d0 = Y2\_GPIO\_NUM;

config.pin\_d1 = Y3\_GPIO\_NUM;

config.pin\_d2 = Y4\_GPIO\_NUM;

config.pin\_d3 = Y5\_GPIO\_NUM;

config.pin\_d4 = Y6\_GPIO\_NUM;

config.pin\_d5 = Y7\_GPIO\_NUM;

config.pin\_d6 = Y8\_GPIO\_NUM;

config.pin\_d7 = Y9\_GPIO\_NUM;

config.pin\_xclk = XCLK\_GPIO\_NUM;

config.pin\_pclk = PCLK\_GPIO\_NUM;

config.pin\_vsync = VSYNC\_GPIO\_NUM;

config.pin\_href = HREF\_GPIO\_NUM;

config.pin\_sscb\_sda = SIOD\_GPIO\_NUM;

config.pin\_sscb\_scl = SIOC\_GPIO\_NUM;

config.pin\_pwdn = PWDN\_GPIO\_NUM;

config.pin\_reset = RESET\_GPIO\_NUM;

config.xclk\_freq\_hz = 20000000;

config.pixel\_format = PIXFORMAT\_JPEG;

config.frame\_size = FRAMESIZE\_QVGA;

config.jpeg\_quality = 6;

config.fb\_count = 1;

// Khởi động camera

if (esp\_camera\_init(&config) != ESP\_OK) {

Serial.println("Camera init failed");

return;

}

startCameraServer();

Serial.println("Camera Stream Ready!");

}

void loop() {

// Không cần gì trong loop

}

* **Code app.py:**

# app.py (hoàn chỉnh với YOLO, IPFS, Blockchain, CSV Export)

from flask import Flask, request, render\_template, send\_from\_directory, send\_file

from ultralytics import YOLO

import os

from werkzeug.utils import secure\_filename

import threading

import cv2

import uuid

import webbrowser

from web3 import Web3

import json

import requests

import datetime

import csv

# ========== Flask setup ==========

app = Flask(\_\_name\_\_)

app.config['UPLOAD\_FOLDER'] = 'uploads'

app.config['RESULT\_FOLDER'] = 'runs/detect/predict'

app.config['DASHBOARD\_FOLDER'] = 'dashboard'

os.makedirs(app.config['UPLOAD\_FOLDER'], exist\_ok=True)

os.makedirs(app.config['DASHBOARD\_FOLDER'], exist\_ok=True)

# ========== YOLO model ==========

model = YOLO('weights/best.pt')

video\_extensions = ['.mp4', '.avi', '.mov', '.mkv']

ESP32\_STREAM\_URL = "http://192.168.109.61/"

# ========== Web3 & Contract Setup ==========

w3 = Web3(Web3.HTTPProvider("http://127.0.0.1:7545"))

account = w3.eth.accounts[0]

with open("abi.json") as f:

    abi = json.load(f)

if not os.path.exists("contract\_address.txt"):

    raise FileNotFoundError("❌ Lỗi: Chưa tìm thấy file contract\_address.txt. Hãy tạo file này với địa chỉ contract đã deploy trong Remix.")

with open("contract\_address.txt") as f:

    contract\_address = f.read().strip()

contract = w3.eth.contract(address=contract\_address, abi=abi)

# ========== IPFS Config ==========

PINATA\_API\_KEY = '395644cd1afc16115605'

PINATA\_SECRET\_API\_KEY = '33f6e83cb40f066e4590f50a5a0733b0ccdcb209c35cec8361352f26c5c2ab0f'

def upload\_to\_ipfs(file\_path):

    url = "https://api.pinata.cloud/pinning/pinFileToIPFS"

    headers = {

        "pinata\_api\_key": PINATA\_API\_KEY,

        "pinata\_secret\_api\_key": PINATA\_SECRET\_API\_KEY

    }

    with open(file\_path, 'rb') as file:

        response = requests.post(url, files={"file": file}, headers=headers)

    if response.status\_code == 200:

        return response.json()['IpfsHash']

    else:

        print("IPFS upload error:", response.text)

        return None

# ========== Export to CSV ==========

def export\_to\_csv(start\_date=None, end\_date=None):

    count = contract.functions.getRecordCount().call()

    with open("records.csv", "w", newline="") as f:

        writer = csv.writer(f)

        writer.writerow(["Timestamp", "DateTime", "Rotten", "Fresh"])

        for i in range(count):

            ts, r, fsh = contract.functions.getRecord(i).call()

            dt = datetime.datetime.fromtimestamp(ts)

            if start\_date and end\_date:

                if not (start\_date <= dt.date() <= end\_date):

                    continue

            writer.writerow([ts, dt.strftime("%Y-%m-%d %H:%M:%S"), r, fsh])

@app.route('/')

def index():

    return render\_template('index.html')

@app.route('/download-records')

def download\_records():

    start = request.args.get('start')

    end = request.args.get('end')

    if start and end:

        start\_date = datetime.datetime.strptime(start, "%Y-%m-%d").date()

        end\_date = datetime.datetime.strptime(end, "%Y-%m-%d").date()

    else:

        start\_date = end\_date = None

    export\_to\_csv(start\_date, end\_date)

    return send\_file("records.csv", as\_attachment=True)

@app.route('/filter')

def filter\_form():

    return render\_template('filter\_form.html')

@app.route('/esp32-live')

def esp32\_live():

    threading.Thread(target=stream\_yolo\_from\_esp32).start()

    webbrowser.open("http://127.0.0.1:5000/dashboard")

    return "\U0001f680 ESP32-CAM stream started."

def stream\_yolo\_from\_esp32():

    cap = cv2.VideoCapture(ESP32\_STREAM\_URL)

    if not cap.isOpened():

        print("ESP32 not connected.")

        return

    while True:

        ret, frame = cap.read()

        if not ret:

            break

        frame = cv2.resize(frame, (320, 240))

        results = model(frame, conf=0.5)

        labels = []

        save\_image = False

        if results[0].boxes.shape[0] > 0:

            for box in results[0].boxes:

                label = model.names[int(box.cls[0])]

                labels.append(label)

                if 'rotten' in label:

                    save\_image = True

        if save\_image:

            annotated\_frame = results[0].plot()

            count\_rotten = sum(1 for l in labels if 'rotten' in l)

            count\_fresh = len(labels) - count\_rotten

            try:

                tx = contract.functions.updateCounts(count\_rotten, count\_fresh).transact({'from': account})

                w3.eth.wait\_for\_transaction\_receipt(tx)

            except Exception as e:

                print("Blockchain error:", e)

            filename = f"{uuid.uuid4().hex}.jpg"

            path = os.path.join(app.config['DASHBOARD\_FOLDER'], filename)

            cv2.imwrite(path, annotated\_frame)

            upload\_to\_ipfs(path)

        display\_frame = annotated\_frame if save\_image else frame

        cv2.imshow("ESP32 + YOLO", display\_frame)

        if cv2.waitKey(1) == ord('q'):

            break

    cap.release()

    cv2.destroyAllWindows()

@app.route('/dashboard')

def dashboard():

    images = os.listdir(app.config['DASHBOARD\_FOLDER'])

    images.sort(reverse=True)

    rotten\_count = contract.functions.rottenCount().call()

    fresh\_count = contract.functions.freshCount().call()

    return render\_template('dashboard.html', images=images, rotten\_count=rotten\_count, fresh\_count=fresh\_count)

@app.route('/dashboard\_img/<filename>')

def dashboard\_image(filename):

    return send\_from\_directory(app.config['DASHBOARD\_FOLDER'], filename)

@app.route('/uploads/<filename>')

def uploaded\_file(filename):

    return send\_from\_directory(app.config['UPLOAD\_FOLDER'], filename)

@app.route('/result/<filename>')

def result\_file(filename):

    return send\_from\_directory(app.config['RESULT\_FOLDER'], filename)

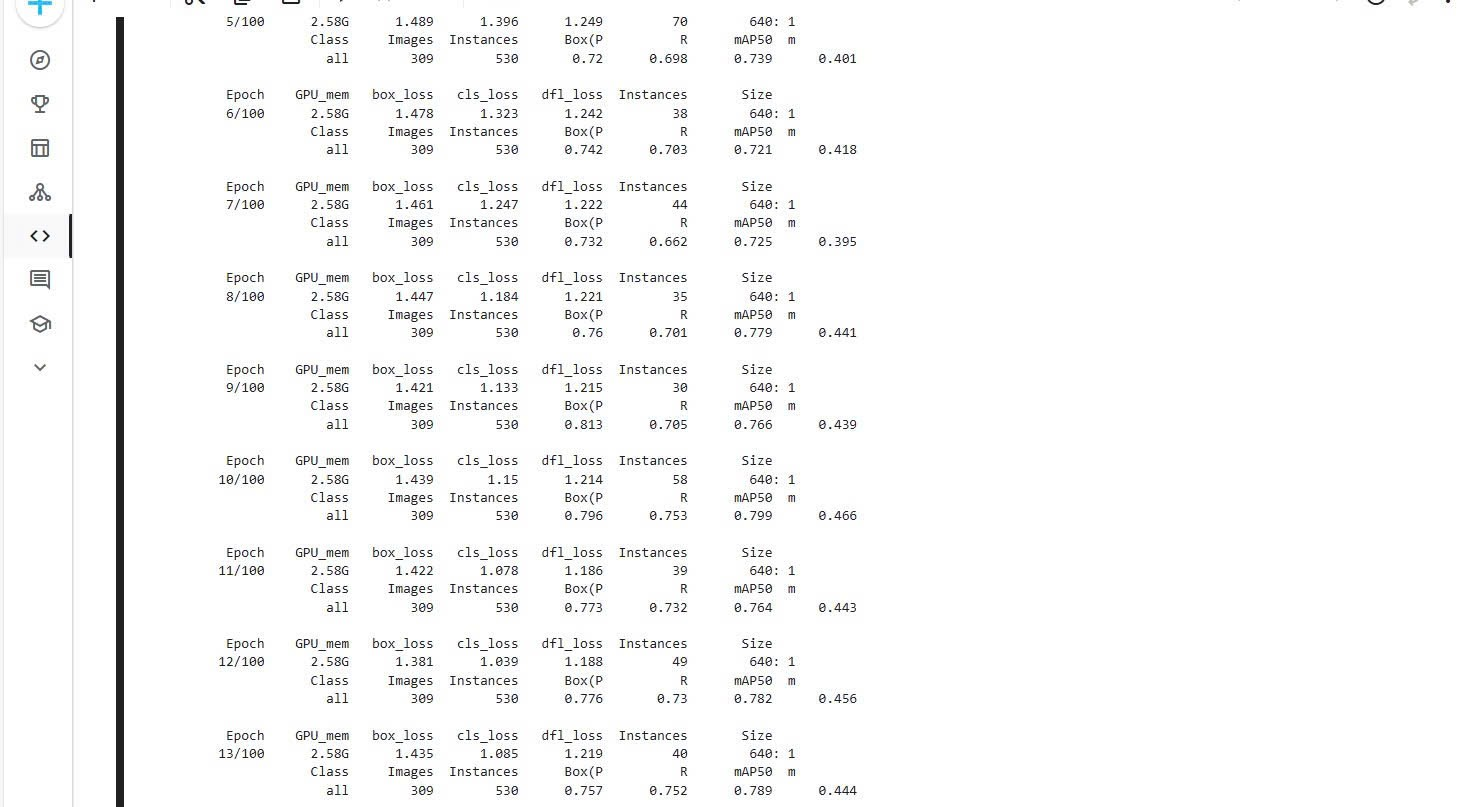
# ========== Chạy Flask ==========

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(host='127.0.0.1', port=5000, debug=True)

## **4.2. Thông số huấn luyện**

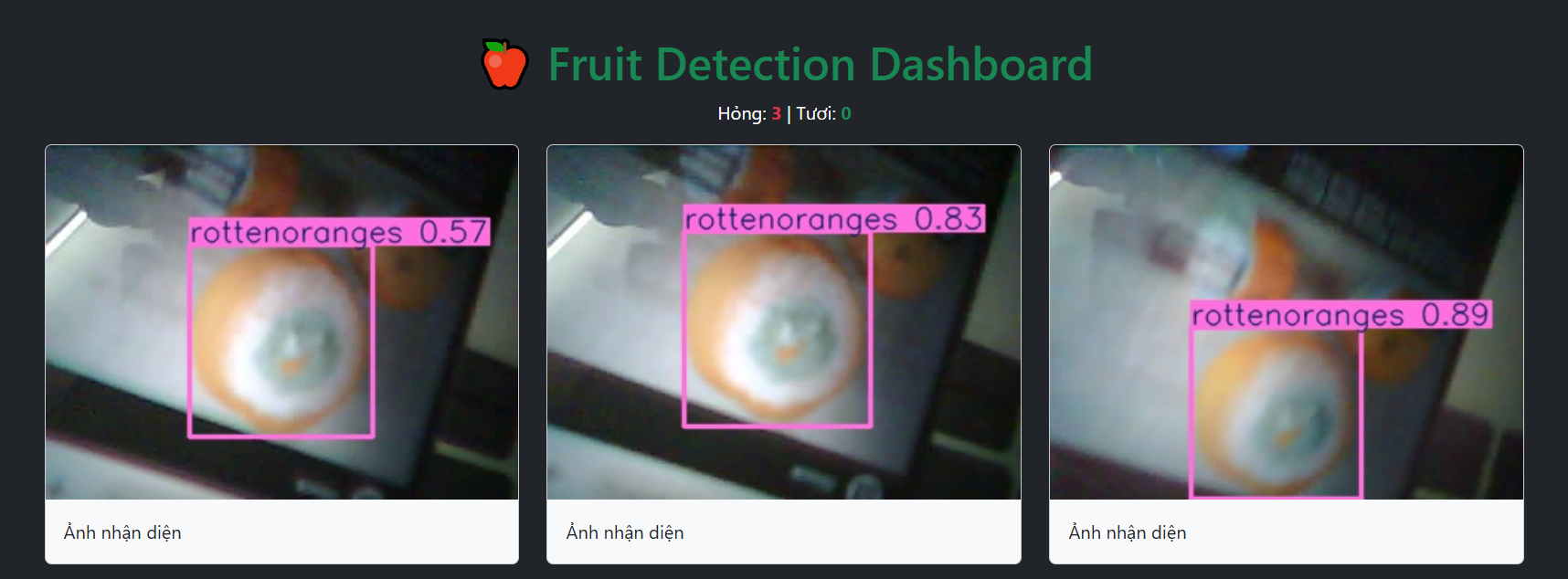
Số liệu sau khi huấn luyện được các thông số dưới ảnh sau:



**4.3. Kết quả**

**A screenshot of a computer

AI-generated content may be incorrect.**

****

**4.4. Đánh giá kết quả**

Kết quả nhận diện đúng nhưng vẫn sẽ có một vại yếu tố gây ảnh hưởng tới kết quả nhận diện như ánh sáng độ phân giải cam delay,…..

# **CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## **5.1. Kết luận**

Đề tài được thực hiện nhằm mục đích cải thiện trong việc nhận diện hoa quả hỏng và tổng hợp lại nhằm mục đích tăng sự chính xác trong việc phân loại các loại sản phẩm hoa quả trong đời sống nhằm cải thiện đời sống.

## **5.2. Hướng phát triển**

Tích hợp thêm đèn led và cảm biến báo động khi phát hiện ra hoa quả bị hỏng và tích hợp thêm blockchain để đẩy và lưu trữ dữ liệu vào các khối block nhằm tránh việc thay đổi kết quả và giúp cho trung thực hơn trong các hoạt động của các công ty.

# **KẾT LUẬN**

Trong quá trình nghiên cứu và triển khai hệ thống ERP quản lý khách hàng trên nền tảng Odoo. Hệ thống không chỉ giúp tối ưu hóa quy trình quản lý thông tin khách hàng, hợp đồng, và các giao dịch kinh doanh mà còn tích hợp nhiều model hỗ trợ như phân tích khách hàng, ghi nhận phản hồi, theo dõi tương tác và quản lý cơ hội bán hàng. Qua đó, doanh nghiệp có thể nắm bắt được toàn bộ hành trình của khách hàng từ lúc tiếp cận đến chăm sóc, tạo điều kiện thuận lợi cho việc đưa ra các chiến lược kinh doanh chính xác.

Với khả năng phân tích sâu và báo cáo trực quan, ban lãnh đạo dễ dàng theo dõi tiến trình và đánh giá hiệu quả của từng bộ phận, từ đó có những điều chỉnh kịp thời để tối ưu hóa quy trình làm việc. Hơn nữa, còn tích hợp thêm các module quản lý nội bộ như nhân viên, nhiệm vụ và chiến lược marketing.

Tuy nhiên, hệ thống hiện tại vẫn tồn tại một số hạn chế cần được khắc phục. Một số module, đặc biệt là các chức năng phân tích khách hàng và dự đoán xu hướng, vẫn chủ yếu dựa trên dữ liệu định tính và chưa đạt được mức độ tự động hóa cao. Ngoài ra, giao diện người dùng mặc dù thân thiện nhưng vẫn có thể được cải thiện thêm về mặt trực quan để phù hợp hơn với các xu hướng thiết kế mới, nhằm nâng cao trải nghiệm của người dùng cuối.

Trong hướng phát triển tương lai, việc mở rộng tích hợp các công nghệ tiên tiến như trí tuệ nhân tạo (AI) và học máy (machine learning) sẽ là một bước đột phá quan trọng. AI có thể giúp dự đoán xu hướng, phân tích hành vi khách hàng một cách chính xác hơn, từ đó đưa ra các chiến lược marketing cá nhân hóa và nâng cao hiệu quả bán hàng. Bên cạnh đó, việc phát triển ứng dụng di động cho phép khách hàng và nhân viên truy cập hệ thống mọi lúc, mọi nơi sẽ góp phần cải thiện khả năng tương tác và nâng cao hiệu quả quản lý. Các module nội bộ khác cũng có thể được mở rộng, tích hợp với các công cụ quản lý dự án và CRM để tạo ra một hệ thống tổng thể hoàn thiện.

# **DANH MỤC TÀI LIỆU THAM KHẢO**

[1]. <https://www.odoo.com/documentation>

[2]. <https://ubuntu.com/server/docs>

[3]. Sumner, M. (2005). Enterprise Resource Planning (ERP): The Dynamics of Operations Management

[4]. Bradford, M. (2010). Modern ERP: Select, Implement, and Use Today's Advanced Business Systems

[5]. Leon, A. (2008). ERP Demystified

[6]. Odoo Community Association (OCA)