|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| BỘ NÔNG NGHIỆP VÀ PHÁT TRIỂN NÔNG THÔN  **TRƯỜNG ĐẠI HỌC LÂM NGHIỆP**    C:\Documents and Settings\ADMIN\Desktop\logo.JPG  **ĐỒ ÁN**  **MÔN : KHAI PHÁ DỮ LIỆU**  **ĐỀ TÀI:**  **HUẤN LUYỆN MÔ HÌNH NHẬN BIẾT RAU THƠM**   |  |  | | --- | --- | | **Sinh viên thực hiện:** | **Nguyễn Văn Đức**  **Đoàn Quang Khải**  **Vũ Tiến Đạt** | | **Lớp:** | **K66-HTTT** | | **Giảng viên hướng dẫn:** | **Ths.Mai Hà An** |   **Hà Nội, ngày 17 tháng 03năm 2025** |

**MỤC LỤC**

Contents

[**I .Giới thiệu** 4](#_Toc195268508)

[**1. Lý do chọn đề tài** 4](#_Toc195268509)

[**2. Mục tiêu nghiên cứu** 5](#_Toc195268510)

[**2.1 Mục tiêu tổng quát** 5](#_Toc195268511)

[**2.2 Mục tiêu cụ thể** 5](#_Toc195268512)

[**3. Phạm vi nghiên cứu** 5](#_Toc195268513)

[**3.1 Phạm vi công nghệ và thuật toán** 5](#_Toc195268514)

[**3.2 Phạm vi về dữ liệu** 5](#_Toc195268515)

[**3.3 Phạm vi về đối tượng** 5](#_Toc195268516)

[**II. Cơ sở lý thuyết** 6](#_Toc195268517)

[**1. Tổng quan về nhận diện đối tượng** 6](#_Toc195268518)

[**1.1 Khái niệm** 6](#_Toc195268519)

[**1.2 Ứng dụng thực tế** 6](#_Toc195268520)

[**1.3 Cách hoạt động** 6](#_Toc195268521)

[**2. Phương pháp nhận diện đối tượng** 6](#_Toc195268522)

[**2.1. R-CNN (Region-based Convolutional Neural Network)** 6](#_Toc195268523)

[**2.2. Fast R-CNN** 7](#_Toc195268524)

[**2.3. Faster R-CNN** 7](#_Toc195268525)

[**2.4. YOLO (You Only Look Once)** 8](#_Toc195268526)

[**2.5. SSD (Single Shot MultiBox Detector)** 8](#_Toc195268527)

[**2.6. Mask R-CNN** 8](#_Toc195268528)

[**2.7. RetinaNet** 9](#_Toc195268529)

[**2.8. EfficientDet** 9](#_Toc195268530)

[**3. Mô hình được chọn trong nghiên cứu** 10](#_Toc195268531)

[**3.1 Mô hình Yolo** 10](#_Toc195268532)

[**3.2 Cách thức hoạt động** 10](#_Toc195268533)

[**III. Xây dựng mô hình nhận diện đối tượng** 12](#_Toc195268534)

[**1. Chuẩn bị dữ liệu** 12](#_Toc195268535)

[**1.1. Thu thập dữ liệu tùy chỉnh** 12](#_Toc195268536)

[**1.2. Xử lý dữ liệu** 12](#_Toc195268537)

[**2. Huấn luyện mô hình** 14](#_Toc195268538)

[**IV. Kết quả và đánh giá** 17](#_Toc195268539)

[**1. Kết quả** 17](#_Toc195268540)

[**2. Hạn chế và đề xuất cải tiến** 20](#_Toc195268541)

[**2.1 Những vấn đề gặp phải** 20](#_Toc195268542)

[**2.2 Đề xuất cải tiến** 20](#_Toc195268543)

[**V. Kết luận** 21](#_Toc195268544)

[**1. Tóm tắt nội dung nghiên cứu** 21](#_Toc195268545)

[**2. Định hướng phát triển tương lai** 22](#_Toc195268546)

# **I .Giới thiệu**

## **1. Lý do chọn đề tài**

Trong bối cảnh công nghệ số phát triển mạnh mẽ, trí tuệ nhân tạo (AI) và thị giác máy tính (Computer Vision) đã trở thành những trụ cột quan trọng, mở ra tiềm năng ứng dụng rộng lớn trong nhiều lĩnh vực như nông nghiệp, ẩm thực, thương mại điện tử và giáo dục. Khả năng nhận diện đối tượng qua hình ảnh không chỉ hỗ trợ tự động hóa các quy trình phân loại, giám sát mà còn mang lại độ chính xác cao, giảm thiểu sai sót so với phương pháp thủ công truyền thống.

Việc xây dựng mô hình và ứng dụng demo để nhận diện rau thơm qua hình ảnh là một hướng nghiên cứu thiết thực, đáp ứng nhu cầu thực tế trong việc phân biệt nhanh chóng các loại rau thơm – vốn là nguyên liệu phổ biến nhưng dễ nhầm lẫn trong đời sống hàng ngày. Với sự phát triển vượt bậc của các thuật toán học máy (Machine Learning) và học sâu (Deep Learning), đặc biệt là mạng nơ-ron tích chập (Convolutional Neural Networks – CNN), hệ thống nhận diện rau thơm có thể đạt hiệu suất cao, phù hợp với các ứng dụng thời gian thực.

Lý do chọn đề tài này xuất phát từ mong muốn nghiên cứu và ứng dụng các công nghệ tiên tiến để xây dựng một hệ thống nhận diện rau thơm hiệu quả, đồng thời kiểm chứng tính khả thi của mô hình thông qua ứng dụng demo. Việc phát triển hệ thống không chỉ giúp giải quyết bài toán phân loại rau thơm trong thực tế mà còn tạo cơ sở để tối ưu hóa và mở rộng sang các đối tượng khác. Hơn nữa, đề tài mang lại cơ hội nâng cao kiến thức chuyên sâu về AI và thị giác máy tính, góp phần vào sự phát triển của các giải pháp công nghệ thông minh.

Bên cạnh đó, đề tài còn có tính ứng dụng cao khi có thể triển khai trong nhiều lĩnh vực thực tiễn như nông nghiệp thông minh (giám sát cây trồng, tự động thu hoạch), ẩm thực (hỗ trợ phân biệt rau thơm cho người dùng), thương mại (phân loại sản phẩm trong siêu thị) và giáo dục (dạy nhận biết thực vật qua hình ảnh). Hệ thống nhận diện rau thơm không chỉ cải thiện hiệu quả công việc mà còn mang lại giá trị thiết thực cho cộng đồng, đặc biệt trong việc hỗ trợ người tiêu dùng, nông dân và các doanh nghiệp nhỏ.

Với những lý do trên, đề tài “Xây dựng mô hình và ứng dụng demo để nhận diện rau thơm qua hình ảnh” là một lựa chọn phù hợp, kết hợp giữa tính ứng dụng thực tiễn và giá trị nghiên cứu khoa học, đồng thời góp phần thúc đẩy sự phát triển của công nghệ AI trong đời sống hiện đại.

## **2. Mục tiêu nghiên cứu**

### **2.1 Mục tiêu tổng quát**

Xây dựng và huấn luyện mô hình để nhận diện đối tượng qua hình ảnh nhằm nâng cao độ chính xác và hiệu quả trong xử lý dữ liệu hình ảnh.

### **2.2 Mục tiêu cụ thể**

* Nghiên cứu và lựa chọn mô hình phù hợp với bài toán.
* Thu thập và gán nhãn hình ảnh.
* Huấn luyện và tối ưu hóa mô hình.
* Kiểm thử, đánh giá và cải thiện hệ thống trong tương lai.

## **3. Phạm vi nghiên cứu**

### **3.1 Phạm vi công nghệ và thuật toán**

### **3.2 Phạm vi về dữ liệu**

* Dữ liệu hình ảnh được thu thập trực tiếp từ rau thơm , có độ phân giải phù hợp và chứa các nhãn rõ ràng để phục vụ quá trình huấn luyện và đánh giá mô hình.
* Xử lý dữ liệu hình ảnh trước khi đưa vào mô hình, bao gồm gắn nhãn, tiền xử lý, tăng cường dữ liệu (data augmentation) và làm sạch dữ liệu để cải thiện độ chính xác.

### **3.3 Phạm vi về đối tượng**

* Hệ thống sẽ nhận diện một đối tượng cụ thể (rau thơm) thay vì tất cả các loại đối tượng có thể có trong hình ảnh.
* Giới hạn về số lượng lớp (categories) nhận diện tùy thuộc vào dữ liệu và khả năng của mô hình.

# **II. Cơ sở lý thuyết**

## **1. Tổng quan về nhận diện đối tượng**

### **1.1 Khái niệm**

* Nhận diện đối tượng không chỉ đơn thuần là phân loại (classification) một hình ảnh chứa đối tượng gì, mà còn bao gồm việc xác định vị trí của đối tượng đó trong hình ảnh, thường dưới dạng các hộp bao (bounding boxes), và gắn nhãn cho từng đối tượng (ví dụ: "xe hơi", "người", "chó"). Một số hệ thống nâng cao hơn thậm chí có thể phân đoạn đối tượng (segmentation) để xác định chính xác biên của chúng.

### **1.2 Ứng dụng thực tế**

* Giao thông: Nhận diện biển báo, xe cộ, người đi bộ trong xe tự hành.
* An ninh: Phát hiện đối tượng đáng ngờ trong camera giám sát.
* Y tế: Nhận diện khối u hoặc tổn thương trong ảnh chụp X-quang, MRI.
* Bán lẻ: Đếm sản phẩm, theo dõi hành vi khách hàng.
* Giải trí: Hiệu ứng thực tế ảo (AR), nhận diện khuôn mặt trong ảnh.

### **1.3 Cách hoạt động**

Nhận diện đối tượng thường dựa trên các mô hình học sâu (deep learning), đặc biệt là mạng nơ-ron tích chập (Convolutional Neural Networks - CNN). Quy trình cơ bản bao gồm:

* **Phát hiện vùng quan tâm (Region Proposal)**: Xác định các khu vực có khả năng chứa đối tượng.
* **Phân loại**: Gán nhãn cho từng vùng (ví dụ: "mèo", "xe đạp").
* **Tinh chỉnh vị trí**: Điều chỉnh hộp bao để chính xác hơn.

Các phương pháp hiện đại thường tích hợp tất cả các bước này vào một mô hình duy nhất (end-to-end).

## **2. Phương pháp nhận diện đối tượng**

### **2.1. R-CNN (Region-based Convolutional Neural Network)**

* **Cách hoạt động**:
  + Sử dụng thuật toán Selective Search để tạo ra khoảng 2000 vùng đề xuất (region proposals) có khả năng chứa đối tượng.
  + Mỗi vùng được đưa qua mạng CNN để trích xuất đặc trưng, sau đó phân loại bằng SVM và tinh chỉnh hộp bao bằng hồi quy.
* **Ưu điểm**: Độ chính xác cao nhờ xử lý từng vùng riêng lẻ.
* **Nhược điểm**: Chậm do phải xử lý nhiều vùng độc lập (không phù hợp với thời gian thực).
* **Năm ra mắt**: 2014.

### **2.2. Fast R-CNN**

* **Cách hoạt động**:
  + Cải tiến từ R-CNN, thay vì xử lý từng vùng qua CNN, Fast R-CNN đưa toàn bộ hình ảnh qua CNN một lần để tạo bản đồ đặc trưng (feature map).
  + Các vùng đề xuất được chiếu lên bản đồ này, sau đó phân loại và tinh chỉnh.
* **Ưu điểm**: Nhanh hơn R-CNN (gấp khoảng 10 lần), vẫn giữ độ chính xác cao.
* **Nhược điểm**: Vẫn phụ thuộc vào Selective Search, gây chậm ở bước tạo vùng đề xuất.
* **Năm ra mắt**: 2015.

### **2.3. Faster R-CNN**

* **Cách hoạt động**:
  + Loại bỏ Selective Search, thay bằng Region Proposal Network (RPN) – một mạng nơ-ron nhỏ tích hợp trong hệ thống.
  + RPN dự đoán vùng đề xuất trực tiếp từ bản đồ đặc trưng, sau đó phân loại và tinh chỉnh như Fast R-CNN.
* **Ưu điểm**: Nhanh hơn Fast R-CNN, độ chính xác cao, gần với thời gian thực.
* **Nhược điểm**: Vẫn phức tạp và chậm hơn các phương pháp "one-stage" như YOLO.
* **Năm ra mắt**: 2016.

### **2.4. YOLO (You Only Look Once)**

* **Cách hoạt động**:
  + Xem nhận diện đối tượng như một bài toán hồi quy duy nhất: chia hình ảnh thành lưới (grid), mỗi ô lưới dự đoán hộp bao, xác suất đối tượng và nhãn lớp.
  + Xử lý toàn bộ hình ảnh trong một lần duy nhất (single pass).
* **Phiên bản nổi bật**: YOLOv1 (2015), YOLOv3, YOLOv5, YOLOv8 (cập nhật gần đây).
* **Ưu điểm**: Rất nhanh, phù hợp với ứng dụng thời gian thực (real-time).
* **Nhược điểm**: Độ chính xác thấp hơn với đối tượng nhỏ hoặc cảnh đông đúc.
* **Năm ra mắt**: 2015 (YOLOv1).

### **2.5. SSD (Single Shot MultiBox Detector)**

* **Cách hoạt động**:
  + Tương tự YOLO, xử lý trong một lần duy nhất (single shot).
  + Sử dụng nhiều lớp đặc trưng (feature layers) với các tỷ lệ khác nhau để phát hiện đối tượng ở kích thước đa dạng.
* **Ưu điểm**: Cân bằng tốt giữa tốc độ và độ chính xác, hiệu quả với đối tượng nhỏ hơn YOLO.
* **Nhược điểm**: Độ chính xác vẫn thua các phương pháp "two-stage" như Faster R-CNN.
* **Năm ra mắt**: 2016.

### **2.6. Mask R-CNN**

* **Cách hoạt động**:
  + Mở rộng từ Faster R-CNN, không chỉ nhận diện và định vị mà còn tạo mặt nạ (mask) cho từng đối tượng (instance segmentation).
  + Thêm một nhánh dự đoán mask song song với nhánh phân loại và hộp bao.
* **Ưu điểm**: Độ chính xác cao, hỗ trợ phân đoạn chi tiết.
* **Nhược điểm**: Chậm hơn, phức tạp, không phù hợp với thời gian thực.
* **Năm ra mắt**: 2017.

### **2.7. RetinaNet**

* **Cách hoạt động**:
  + Là một phương pháp "one-stage" nhưng cải tiến bằng Focal Loss để giải quyết vấn đề mất cân bằng giữa lớp nền (background) và đối tượng (foreground).
  + Sử dụng Feature Pyramid Network (FPN) để phát hiện đối tượng ở nhiều tỷ lệ.
* **Ưu điểm**: Độ chính xác gần bằng "two-stage" nhưng nhanh hơn.
* **Nhược điểm**: Phức tạp hơn YOLO hoặc SSD.
* **Năm ra mắt**: 2017.

### **2.8. EfficientDet**

* **Cách hoạt động**:
  + Kết hợp mạng backbone hiệu quả (EfficientNet) với Feature Pyramid Network hai chiều (BiFPN) và tối ưu hóa cho hiệu suất.
  + Được thiết kế để cân bằng giữa tốc độ, độ chính xác và tài nguyên tính toán.
* **Ưu điểm**: Hiệu quả cao, phù hợp với cả thiết bị mạnh và yếu.
* **Nhược điểm**: Cần tinh chỉnh để đạt hiệu suất tối ưu.
* **Năm ra mắt**: 2020.

## **3. Mô hình được chọn trong nghiên cứu**

### **3.1 Mô hình Yolo**

Mô hình YOLO (You Only Look Once) là một trong những kiến trúc học sâu (deep learning) phổ biến nhất để phát hiện đối tượng (object detection) trong thời gian thực. YOLO nổi bật nhờ tốc độ nhanh và khả năng xử lý hình ảnh chỉ trong một lần quét, thay vì chia nhỏ quá trình như các phương pháp truyền thống (ví dụ: R-CNN)

### **3.2 Cách thức hoạt động**

**Bước 1: Chia hình ảnh thành lưới (Grid)**

* Hình ảnh đầu vào được chia thành một lưới kích thước S×S S \times S S×S (ví dụ: 7×7 7 \times 7 7×7 hoặc 13×13 13 \times 13 13×13, tùy phiên bản).
* Mỗi ô trong lưới chịu trách nhiệm dự đoán các đối tượng có tâm (center) nằm trong ô đó.

**Bước 2: Dự đoán cho mỗi ô lưới**

Mỗi ô trong lưới dự đoán:

* **B hộp giới hạn (bounding boxes)**: Mỗi ô dự đoán B B B hộp giới hạn (thường là 2 hoặc 3, tùy phiên bản). Mỗi hộp bao gồm:
  + Tọa độ tâm (x,y)(x, y)(x,y): Vị trí tâm của hộp so với ô lưới.
  + Kích thước (w,h)(w, h)(w,h): Chiều rộng và chiều cao của hộp, thường được chuẩn hóa theo kích thước hình ảnh.
  + **Độ tin cậy (confidence score)**: Xác suất rằng hộp chứa một đối tượng, được tính bằng công thức: Confidence=P(Object)×IoUpred, truth\text{Confidence} = P(\text{Object}) \times \text{IoU}\_{\text{pred, truth}}Confidence=P(Object)×IoUpred, truth​ Trong đó, IoU \text{IoU} IoU (Intersection over Union) đo lường mức độ chồng lấn giữa hộp dự đoán và hộp thực tế.
* **Xác suất lớp (class probabilities)**: Mỗi ô dự đoán xác suất cho C C C lớp (ví dụ: 20 lớp nếu phát hiện 20 loại đối tượng). Xác suất này được tính có điều kiện, giả sử ô chứa một đối tượng.

Kết quả đầu ra của mỗi ô là một vector chứa thông tin:

(B×(x,y,w,h,confidence))+C(B \times (x, y, w, h, \text{confidence})) + C(B×(x,y,w,h,confidence))+C

Ví dụ: Với lưới 7×7 7 \times 7 7×7, B=2 B = 2 B=2, C=20 C = 20 C=20, đầu ra sẽ có kích thước 7×7×(2×5+20)=7×7×30 7 \times 7 \times (2 \times 5 + 20) = 7 \times 7 \times 30 7×7×(2×5+20)=7×7×30.

**Bước 3: Kết hợp dự đoán**

* Mạng nơ-ron tổng hợp các dự đoán từ tất cả các ô để tạo ra một tập hợp các hộp giới hạn và xác suất lớp.
* Để loại bỏ các hộp trùng lặp hoặc không chính xác, YOLO sử dụng **Non-Maximum Suppression (NMS)**:
  + Chọn hộp có độ tin cậy cao nhất.
  + Loại bỏ các hộp khác có IoU cao với hộp đã chọn (ngưỡng IoU thường là 0.5).
  + Lặp lại cho đến khi chỉ còn các hộp tốt nhất.

**Bước 4: Đầu ra cuối cùng**

* Kết quả là danh sách các hộp giới hạn, mỗi hộp đi kèm với:
  + Tọa độ và kích thước.
  + Lớp đối tượng (ví dụ: "chó", "xe").
  + Độ tin cậy.

# **III. Xây dựng mô hình nhận diện đối tượng**

## **1. Chuẩn bị dữ liệu**

### **1.1. Thu thập dữ liệu tùy chỉnh**

Để xây dựng mô hình nhận diện rau thơm, việc thu thập dữ liệu là bước đầu tiên và quan trọng. Dữ liệu được thu thập từ các nguồn thực tế nhằm đảm bảo tính đa dạng và phù hợp với mục tiêu của dự án. Cụ thể:

* Nguồn dữ liệu:

Hình ảnh các loại rau thơm (húng, bạc hà, đinh lăng, hành, diếp cá) được chụp trực tiếp từ chợ, vườn rau, hoặc các bữa ăn hàng ngày.

* Số lượng và phân loại:

Tổng cộng 250 hình ảnh được thu thập, bao gồm 5 loại rau thơm: húng (50 ảnh), bạc hà (50 ảnh), đinh lăng (50 ảnh), hành (50 ảnh), và diếp cá (50 ảnh).

Các hình ảnh được chụp trong nhiều điều kiện ánh sáng (sáng, tối, ánh sáng tự nhiên, ánh sáng nhân tạo) và góc chụp khác nhau (góc trên, góc ngang, góc nghiêng) để đảm bảo tính tổng quát.

Định dạng và chất lượng:

Hình ảnh được lưu ở định dạng JPEG với độ phân giải tối thiểu 640x640 pixel để đảm bảo chất lượng khi huấn luyện mô hình.

### **1.2. Xử lý dữ liệu**

Sau khi thu thập, dữ liệu thô cần được xử lý để đảm bảo phù hợp với yêu cầu của mô hình YOLOv8. Quá trình xử lý bao gồm các bước: gán nhãn, làm sạch, và tăng cường dữ liệu.

#### **1.2.1 Gán nhãn dữ liệu**

Gán nhãn là bước quan trọng để cung cấp thông tin cho mô hình học máy về các đối tượng cần nhận diện trong hình ảnh.

* Công cụ gán nhãn:

Sử dụng công cụ LabelImg để gán nhãn cho các hình ảnh. LabelImg hỗ trợ định dạng Pascal VOC (XML), phù hợp với YOLOv8 sau khi chuyển đổi.

* Quy trình gán nhãn:

Mỗi hình ảnh được mở trong LabelImg, sau đó vẽ khung bao (bounding box) quanh từng loại rau thơm có trong ảnh.

Gán nhãn tương ứng cho từng khung bao: hung, bacha, dinhlang, hanh, hoặc diepca.

Lưu kết quả dưới dạng file XML (định dạng Pascal VOC) cho mỗi hình ảnh. Ví dụ: bacha\_1.jpg sẽ có file bacha\_1.xml đi kèm.

Kết quả:

Tổng cộng 250 file XML được tạo, mỗi file chứa thông tin về vị trí khung bao (xmin, ymin, xmax, ymax) và nhãn của từng đối tượng trong ảnh.

Các file XML được lưu cùng thư mục với hình ảnh tương ứng, ví dụ: images/train/bacha\_1.xml.

#### **1.2.2 Làm sạch dữ liệu**

Dữ liệu thô thường chứa các lỗi hoặc không đồng nhất, do đó cần làm sạch để đảm bảo chất lượng huấn luyện.

* Kiểm tra và loại bỏ dữ liệu không hợp lệ:

Loại bỏ các hình ảnh bị mờ, quá tối, hoặc không chứa rau thơm (do lỗi khi thu thập).

Kiểm tra các file XML để đảm bảo không có khung bao bị gán nhãn sai (ví dụ: nhãn hung nhưng thực tế là hanh)

* Chuẩn hóa dữ liệu:

Đổi tên file hình ảnh và XML để đồng nhất định dạng: tenrau\_sothutu.jpg (ví dụ: bacha\_1.jpg, hung\_50.jpg).

Chuyển đổi tất cả hình ảnh về độ phân giải 640x640 pixel để đồng bộ kích thước đầu vào cho mô hình YOLOv8.

* Tổ chức dữ liệu:

Chia dữ liệu thành 2 tập: train (80%, 200 hình ảnh) và validation (20%, 50 hình ảnh).

#### **1.2.3 Tăng cường dữ liệu**

Để tăng tính tổng quát của mô hình và tránh hiện tượng overfitting, dữ liệu được tăng cường (data augmentation) bằng các kỹ thuật sau:

* Kỹ thuật tăng cường:
* Xoay (Rotation): Xoay hình ảnh ngẫu nhiên trong khoảng ±30 độ để mô phỏng các góc chụp khác nhau.
* Lật (Flip): Lật ngang và lật dọc hình ảnh để tăng tính đa dạng.
* Thay đổi độ sáng và độ tương phản (Brightness & Contrast): Điều chỉnh độ sáng và độ tương phản ngẫu nhiên trong khoảng ±20% để mô phỏng các điều kiện ánh sáng khác nhau.
* Cắt ngẫu nhiên (Random Crop): Cắt ngẫu nhiên một phần hình ảnh (vẫn đảm bảo đối tượng rau thơm nằm trong khung) để mô phỏng các trường hợp rau bị che khuất.
* Thêm nhiễu (Noise): Thêm nhiễu Gaussian để mô phỏng hình ảnh chất lượng thấp.

## **2. Huấn luyện mô hình**

from google.colab import drive

drive.mount('/content/drive')

dùng để kết nối với drive

yaml\_content = """train: /content/drive/MyDrive/khaiphadulieu/images/train

val: /content/drive/MyDrive/khaiphadulieu/images/validation

nc: 5

names: ["hung", "bacha", "dinhlang", "hanh", "diepca"]

"""

with open("/content/drive/MyDrive/khaiphadulieu/data.yaml", "w") as f:

  f.write(yaml\_content)

tạo ra file yami

from ultralytics import YOLO

model = YOLO("yolov8n.pt")

model.train(

    data="/content/drive/MyDrive/khaiphadulieu/data.yaml",

    epochs=50,

    imgsz=640,

    batch=8

)

Chạy huấn luyện

import shutil

# Danh sách các file best.pt

best\_pt\_files = [

    "/content/runs/detect/train2/weights/best.pt",

    "/content/runs/detect/rau\_thom\_train/weights/best.pt",

    "/content/runs/detect/rau\_thom\_train2/weights/best.pt"

]

# Sao chép từng file vào Google Drive

save\_dir = "/content/drive/MyDrive/khaiphadulieu/weights/"

os.makedirs(save\_dir, exist\_ok=True)

for best\_pt\_path in best\_pt\_files:

    if os.path.exists(best\_pt\_path):

        # Lấy tên thư mục huấn luyện (ví dụ: train2, rau\_thom\_train)

        train\_dir = os.path.basename(os.path.dirname(os.path.dirname(best\_pt\_path)))

        save\_path = os.path.join(save\_dir, f"best\_{train\_dir}.pt")

        shutil.copy(best\_pt\_path, save\_path)

        print(f"Đã sao chép {best\_pt\_path} vào {save\_path}")

    else:

        print(f"Không tìm thấy {best\_pt\_path}")

# Kiểm tra các file đã sao chép

print("Các file trong thư mục Google Drive:", os.listdir(save\_dir))

xuất file ra drive

# **IV. Kết quả và đánh giá**

## **1. Kết quả**

A green leaf on a tile

AI-generated content may be incorrect.

A green leaf on a tile surface

AI-generated content may be incorrect.

A close up of a plant

AI-generated content may be incorrect.

A leaf on a surface

AI-generated content may be incorrect.

A leaf on a surface

AI-generated content may be incorrect.

## **2. Hạn chế và đề xuất cải tiến**

### **2.1 Những vấn đề gặp phải**

* **Độ chính xác thấp với đối tượng nhỏ**: Do chia lưới, YOLO khó phát hiện các đối tượng nhỏ hoặc gần nhau.
* **Giới hạn số đối tượng**: Mỗi ô chỉ dự đoán một số lượng giới hạn hộp, có thể bỏ sót nếu có nhiều đối tượng trong một ô.
* **Phụ thuộc vào anchor boxes** (trong các phiên bản sau): Cần thiết kế anchor phù hợp với tập dữ liệu.

### **2.2 Đề xuất cải tiến**

Thu thập và làm giàu dữ liệu: Bổ sung thêm hình ảnh từ nhiều nguồn khác nhau để tăng độ đa dạng của bộ dữ liệu. Thực hiện data augmentation (xoay ảnh, thay đổi độ sáng, làm mờ…) để giúp mô hình học tốt hơn.

Tối ưu hóa tốc độ xử lý: Sử dụng model quantization và pruning để giảm kích thước mô hình. Triển khai trên GPU hoặc TPU để tăng tốc độ nhận diện trong các ứng dụng thời gian thực.

# **V. Kết luận**

## **1. Tóm tắt nội dung nghiên cứu**

Đề tài “Xây dựng mô hình và ứng dụng demo để nhận diện rau thơm qua hình ảnh” đã được thực hiện với mục tiêu phát triển một hệ thống nhận diện đối tượng dựa trên hình ảnh, tập trung vào việc phân loại các loại rau thơm phổ biến. Nghiên cứu đã trải qua các giai đoạn chính bao gồm:

* **Thu thập và xử lý dữ liệu**: Một tập dữ liệu hình ảnh rau thơm đã được xây dựng, bao gồm các loại như húng, bạc hà, thì là, hành, và diếp cá. Dữ liệu được gán nhãn, làm sạch và tăng cường thông qua các kỹ thuật như xoay ảnh, thay đổi độ sáng để cải thiện độ đa dạng và chất lượng.
* **Lựa chọn và huấn luyện mô hình**: Mô hình YOLO (You Only Look Once), cụ thể là phiên bản YOLOv8, được chọn nhờ khả năng nhận diện nhanh và hiệu quả trong các ứng dụng thời gian thực. Mô hình đã được huấn luyện trên tập dữ liệu tùy chỉnh với các tham số như số epoch là 50, kích thước ảnh 640x640 và batch size là 8.
* **Triển khai và đánh giá**: Sau quá trình huấn luyện, mô hình đã được kiểm thử và đánh giá dựa trên các chỉ số như độ chính xác (precision), độ nhạy (recall), và mAP (mean Average Precision). Kết quả cho thấy mô hình có khả năng nhận diện các loại rau thơm với độ chính xác tương đối cao, dù vẫn tồn tại một số hạn chế.
* **Ứng dụng demo**: Một ứng dụng demo đơn giản đã được phát triển để minh họa khả năng nhận diện rau thơm trong thực tế, giúp kiểm chứng tính khả thi của mô hình.

Nghiên cứu không chỉ đáp ứng mục tiêu xây dựng một hệ thống nhận diện rau thơm hiệu quả mà còn đóng góp vào việc ứng dụng trí tuệ nhân tạo và thị giác máy tính trong các lĩnh vực thực tiễn như nông nghiệp, ẩm thực và giáo dục. Kết quả đạt được đã khẳng định tiềm năng của các thuật toán học sâu, đặc biệt là YOLO, trong việc giải quyết các bài toán nhận diện đối tượng cụ thể.

## **2. Định hướng phát triển tương lai**

Để nâng cao hiệu quả và mở rộng phạm vi ứng dụng của hệ thống, nhóm nghiên cứu đề xuất một số định hướng phát triển trong tương lai như sau:

* **Mở rộng tập dữ liệu**: Tiếp tục thu thập thêm hình ảnh từ nhiều nguồn khác nhau, bao gồm các điều kiện ánh sáng, góc chụp và môi trường đa dạng hơn (ví dụ: rau thơm trong vườn, siêu thị, hoặc món ăn). Điều này sẽ giúp cải thiện khả năng tổng quát hóa của mô hình, đặc biệt với các trường hợp thực tế phức tạp.
* **Tăng cường số lượng lớp nhận diện**: Hiện tại, mô hình tập trung vào năm loại rau thơm phổ biến. Trong tương lai, có thể mở rộng để nhận diện thêm các loại rau thơm khác hoặc thậm chí các loại thực vật, rau củ khác để tăng tính ứng dụng.
* **Cải thiện độ chính xác với đối tượng nhỏ**: Để khắc phục hạn chế của YOLO trong việc nhận diện các đối tượng nhỏ hoặc gần nhau, có thể thử nghiệm các phiên bản YOLO mới hơn (như YOLOv9 hoặc các biến thể tối ưu) hoặc tích hợp các kỹ thuật như Feature Pyramid Network (FPN) để cải thiện hiệu suất.
* **Tối ưu hóa cho thiết bị di động**: Phát triển ứng dụng di động tích hợp mô hình nhận diện, cho phép người dùng (nông dân, đầu bếp, hoặc người tiêu dùng) sử dụng trực tiếp trên điện thoại thông minh. Điều này đòi hỏi tối ưu hóa mô hình bằng các kỹ thuật như quantization hoặc pruning để giảm kích thước và tăng tốc độ xử lý trên các thiết bị có tài nguyên hạn chế.
* **Ứng dụng trong các lĩnh vực liên quan**: Mở rộng phạm vi ứng dụng của mô hình sang các lĩnh vực khác như giám sát cây trồng trong nông nghiệp thông minh, phân loại sản phẩm trong thương mại điện tử, hoặc hỗ trợ giáo dục qua các ứng dụng học nhận biết thực vật.
* **Nghiên cứu tích hợp đa phương thức**: Kết hợp nhận diện hình ảnh với các dữ liệu khác (ví dụ: thông tin văn bản về đặc điểm rau thơm hoặc dữ liệu cảm biến) để xây dựng một hệ thống thông minh hơn, có khả năng cung cấp thông tin chi tiết về đối tượng được nhận diện.
* **Triển khai trên nền tảng thời gian thực**: Tăng cường khả năng xử lý thời gian thực của hệ thống bằng cách triển khai trên các thiết bị phần cứng chuyên dụng (như GPU, TPU) hoặc các nền tảng đám mây, đảm bảo tốc độ và độ chính xác đáp ứng yêu cầu của các ứng dụng thực tế.