TRƯỜNG ĐẠI HỌC LÂM NGHIỆP

**KHOA CƠ ĐIỆN CÔNG TRÌNH**



**BÁO CÁO ĐỒ ÁN**

**Môn học: Khai phá dữ liệu**

**Đề tài: Xây dựng mô hình và ứng dụng demo để nhận diện đối tượng qua hình ảnh.**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | **ThS. Mai Hà An** |
| **Sinh viên thực hiện:** | **Trương Văn Bình**  **Nguyễn Thị Mai**  **Tống Thị Khánh Trang** |
| **Lớp:** | **K66-HTTT** |

**Hà Nội, tháng 3 năm 2025**

**MỤC LỤC**

[Chương 1. TỔNG QUAN VẤN ĐỀ 3](#_Toc194247006)

[1. Giới thiệu 3](#_Toc194247007)

[1.1. Huấn luyện mô hình là gì? 3](#_Toc194247008)

[1.2. Quy trình huấn luyện mô hình 3](#_Toc194247009)

[1.3. YOLO là gì? 3](#_Toc194247010)

[2. Mục tiêu của đề tài 4](#_Toc194247011)

[3. Đối tượng nhận diện 4](#_Toc194247012)

[4. Phương pháp thực hiện 4](#_Toc194247013)

[Chương 2. QUY TRÌNH THỰC HIỆN 5](#_Toc194247014)

[1. Yêu cầu dữ liệu 5](#_Toc194247015)

[2. Gán nhãn đối tượng 5](#_Toc194247016)

[3. Chuẩn bị dữ liệu 6](#_Toc194247017)

[4. Huấn luyện mô hình 6](#_Toc194247018)

[5. Tinh chỉnh mô hình 7](#_Toc194247019)

[6. Ứng dụng Web Demo 7](#_Toc194247020)

[Chương 3. KẾT QUẢ THỰC HIỆN VÀ ĐÁNH GIÁ 9](#_Toc194247021)

[1. Kết quả thực hiện 9](#_Toc194247022)

[2. Đánh giá và nhận xét 11](#_Toc194247023)

[3. Kết luận 12](#_Toc194247024)

[4. Đề xuất phát triển 12](#_Toc194247025)

# Tổng quan vấn đề

## Giới thiệu

### Huấn luyện mô hình là gì?

Huấn luyện mô hình là quá trình sử dụng dữ liệu để dạy cho một thuật toán học máy nhận diện hoặc dự đoán các đặc trưng cụ thể. Trong bài toán nhận diện hình ảnh, huấn luyện mô hình giúp hệ thống có thể phân loại và xác định đối tượng trong ảnh một cách tự động.

### Quy trình huấn luyện mô hình

Quy trình huấn luyện bao gồm:

* Thu thập dữ liệu: Chụp ảnh và thu thập dữ liệu hình ảnh của các loại hoa cần nhận diện.
* Chuẩn bị dữ liệu: Tiền xử lý dữ liệu, chuẩn hóa kích thước ảnh.
* Gán nhãn dữ liệu: Xác định vùng chứa hoa và dán nhãn cho từng loại hoa trong ảnh.
* Huấn luyện mô hình: Sử dụng thuật toán YOLO để học các đặc trưng từ dữ liệu.
* Đánh giá mô hình: Kiểm tra độ chính xác và tối ưu hóa tham số.
* Triển khai mô hình: Đưa mô hình vào ứng dụng thực tế.

### YOLO là gì?

YOLO (You Only Look Once) là một trong những thuật toán hàng đầu về nhận diện đối tượng trong thời gian thực, nổi bật nhờ tốc độ nhanh và độ chính xác cao. So với các thuật toán khác như Faster R-CNN hay SSD, YOLO có khả năng xử lý ảnh nhanh chóng, giúp nhận diện đối tượng hiệu quả trong các ứng dụng thời gian thực.

Hiện tại, YOLO đã phát triển qua nhiều phiên bản:

* YOLOv1: Phiên bản đầu tiên, nhanh nhưng độ chính xác chưa cao.
* YOLOv2 (YOLO9000): Cải thiện tốc độ và độ chính xác, hỗ trợ nhận diện nhiều đối tượng hơn.
* YOLOv3: Tăng cường khả năng nhận diện đối tượng nhỏ, sử dụng nhiều quy mô feature map.
* YOLOv4: Tối ưu hóa tốc độ và độ chính xác, được
* sử dụng rộng rãi trong các ứng dụng thực tế.
* YOLOv5: Một phiên bản tối ưu hóa bởi cộng đồng, phổ biến nhờ dễ sử dụng và hiệu suất tốt.
* YOLOv6, YOLOv7: Các phiên bản cải tiến, tập trung vào hiệu suất trên thiết bị edge.
* YOLOv8: Phiên bản mới nhất với nhiều cải tiến về tốc độ, độ chính xác và khả năng huấn luyện dễ dàng hơn.

## Mục tiêu của đề tài

* Xây dựng mô hình có khả năng nhận diện chính xác các loại hoa từ hình ảnh đầu vào.
* Ứng dụng mô hình vào thực tế thông qua một ứng dụng Web demo, giúp người dùng có thể tải ảnh lên và nhận diện các loại hoa một cách nhanh chóng.

## Đối tượng nhận diện

Ứng dụng AI để nhận diện năm loại hoa: hoa sen cạn, hoa đào, hoa ban, hoa lan, và me chua đất. Đối tượng sử dụng mô hình có thể là các nhà nghiên cứu thực vật, nông dân hoặc bất kỳ ai quan tâm đến việc phân loại hoa tự động.

## Phương pháp thực hiện

* Nghiên cứu lý thuyết: Tìm hiểu về huấn luyện mô hình, YOLO và các thuật toán nhận diện ảnh.
* Thực nghiệm: Huấn luyện mô hình trên tập dữ liệu thu thập được.

# QUY TRÌNH THỰC HIỆN

## Yêu cầu dữ liệu

* Đối tượng ảnh: Hoa đào, hoa lan, hoa ban, hoa sen cạn, me đất.
* Số lượng ảnh: Mỗi loại hoa có 50 ảnh (tổng cộng 250 ảnh).
* Kích thước ảnh: Chuẩn hóa về 640x640 pixel.
* Định dạng nhãn: Sử dụng định dạng YOLO (label.txt).
* Công cụ gán nhãn: LabelImg.

## Gán nhãn đối tượng

Các bước thực hiện:

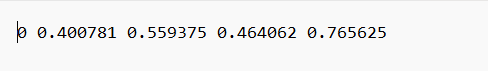
Bước 1: Cài đặt Lableimg

Bước 2: Gán nhãn cho các đối tượng trong ảnh

* Chọn thư mục chứa ảnh
* Mở ảnh, để gán nhãn. Chọn format YOLO rồi chọn “Create RectBox” từ thanh công cụ hoặc nhấn phím W trên bàn phím để vẽ một hộp bao quanh đối tượng trong ảnh.
* Vẽ bounding box: Kéo chuột để vẽ một hộp bao quanh đối tượng mà bạn muốn gán nhãn. Bạn có thể vẽ nhiều hộp cho nhiều đối tượng khác nhau trong cùng một ảnh.
* Gán nhãn: nhập nhãn cho đối tượng. Bạn cần nhập tên lớp (class) cho đối tượng này (ví dụ: "Hoa Ban", "Sen Cạn", v.v.). Enter để lưu nhãn và save để lưu kết quả vừa thực hiện.
* Thực hiện gán nhãn lần lượt với toàn bổ ảnh trong thư mục ảnh đã chọn.

Bước 3: Lưu nhãn

LabelImg sẽ tự động tạo một tệp nhãn dưới định dạng .txt (YOLO) trong thư mục nhãn đã chọn. Ví dụ: Nếu ảnh là Ban1 tệp nhãn sẽ là Ban1.txt và có nội dung như hình:



Trong đó:

* 0: số thứ tự của tên nhãn.
* 0.400781 0.559375 0.464062 0.765625: kích thước bounding box của đối tượng.

Bước 4: Kiểm tra và hoàn thành

Quay lại kiểm tra các nhãn đã dán. Kiểm tra tên nhãn, bounding box và căn chỉnh cho phù hợp với nhu cầu cần thiết.

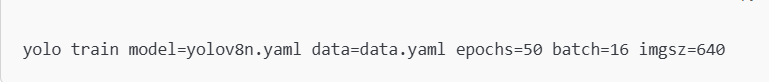
*\*NOTE*: Nếu trước đó nhãn chưa được sử dụng ở định dạng YOLO thì có thể sử dụng tính năng **"Pascal VOC to YOLO"** từ menu View → Auto Save Mode → YOLO để chuyển đổi tự động.

## Chuẩn bị dữ liệu

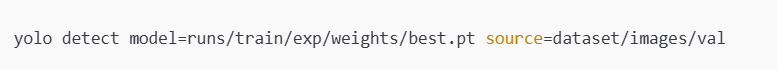
* Chuyển toàn bộ dữ liệu ảnh và nhãn lên Google Drive.
* Sử dụng Google Colab để chạy huấn luyện với YOLOv8.
* Cấu hình mô hình:
* Sử dụng YOLOv8n (phiên bản nhẹ).
* Số epoch: 250.
* Bộ tối ưu hóa: SGD.
* Kích thước batch: 16.

## Huấn luyện mô hình

* Tải mô hình mẫu từ Ultralytics.
* Cấu hình tham số phù hợp với bài toán



* model=yolov8n.yaml: Chỉ định mô hình YOLOv8 nhỏ nhất (yolov8n) hoặc bạn có thể thay thế bằng mô hình lớn hơn như yolov8s, yolov8m, yolov8l, yolov8x.
* data=data.yaml: Đường dẫn tới tệp cấu hình dữ liệu bạn đã tạo.
* epochs=50: Số lượng vòng lặp huấn luyện. Bạn có thể thay đổi con số này tùy thuộc vào bộ dữ liệu và yêu cầu.
* batch=16: Số lượng ảnh trong mỗi batch.
* imgsz=640: Kích thước ảnh đầu vào. Có thể thay đổi kích thước này để phù hợp với yêu cầu.
* Chạy huấn luyện trên GPU (Tesla T4).



* model: Đường dẫn tới tệp trọng số mô hình.
* source: Đường dẫn tới thư mục chứa các hình ảnh kiểm tra.

Kết quả sẽ được lưu trong thư mục runs/detect/exp, nơi bạn có thể kiểm tra hình ảnh với các bounding box và nhãn đối tượng đã được nhận diện.

* Giám sát quá trình huấn luyện chỉ số mAP (Mean Average Precision).
* Sau khi huấn luyện xong, bạn có thể xem các chỉ số đánh giá như mAP, precision, recall, và F1-score từ kết quả huấn luyện.
* Điều này sẽ giúp bạn hiểu rõ mô hình hoạt động như thế nào và có thể giúp tinh chỉnh thêm.

## Tinh chỉnh mô hình

* Nếu mô hình chưa đạt được kết quả mong muốn, bạn có thể thử điều chỉnh các siêu tham số như learning rate, batch size, hoặc thậm chí thay đổi mô hình (từ YOLOv8n lên YOLOv8s hoặc YOLOv8m).
* Tinh chỉnh lại mô hình trên một bộ dữ liệu nhỏ hoặc bộ dữ liệu có những đặc điểm đặc biệt.
* Nếu cần, bạn có thể huấn luyện lại mô hình với các tham số mới.

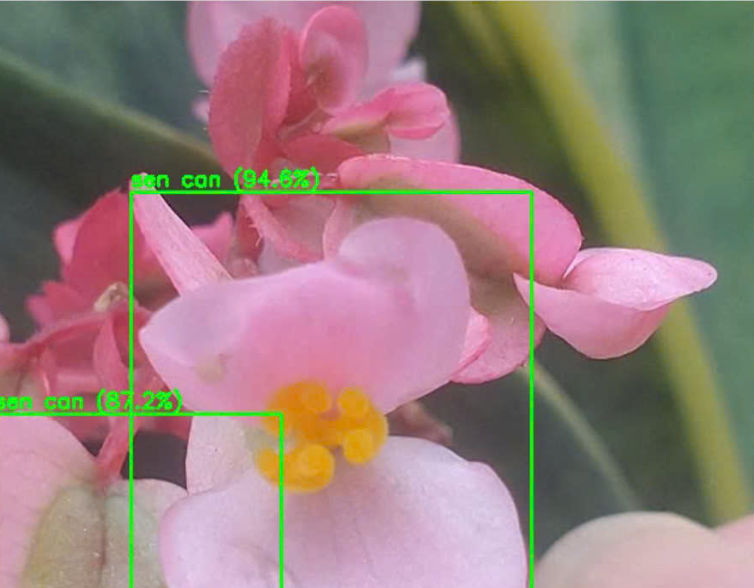
## Ứng dụng Web Demo

* Xây dựng ứng dụng Web để tải ảnh và nhận diện hoa bằng mô hình đã huấn luyện.
* Triển khai mô hình YOLO lên Web.
* Ứng dụng hiển thị ảnh với kết quả nhận diện, vẽ bounding box và hiển thị thông tin chi tiết về loại hoa.
* Khi người dùng tải ảnh lên, hệ thống sẽ phân tích, nhận diện và cung cấp thông tin mô tả về loại hoa được nhận diện từ Wikipedia.
* Giao diện trực quan, dễ sử dụng, giúp người dùng nhận diện hoa một cách nhanh chóng và chính xác.

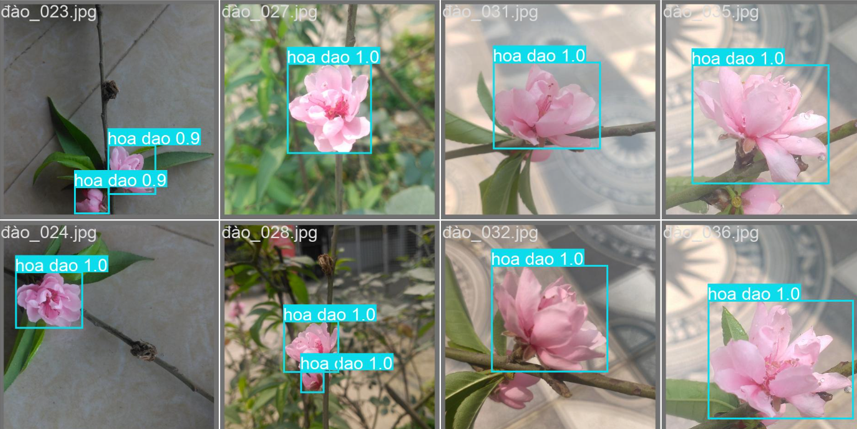
# KẾT QUẢ THỰC HIỆN VÀ ĐÁNH GIÁ

## Kết quả thực hiện

* Kết quả huấn luyện:

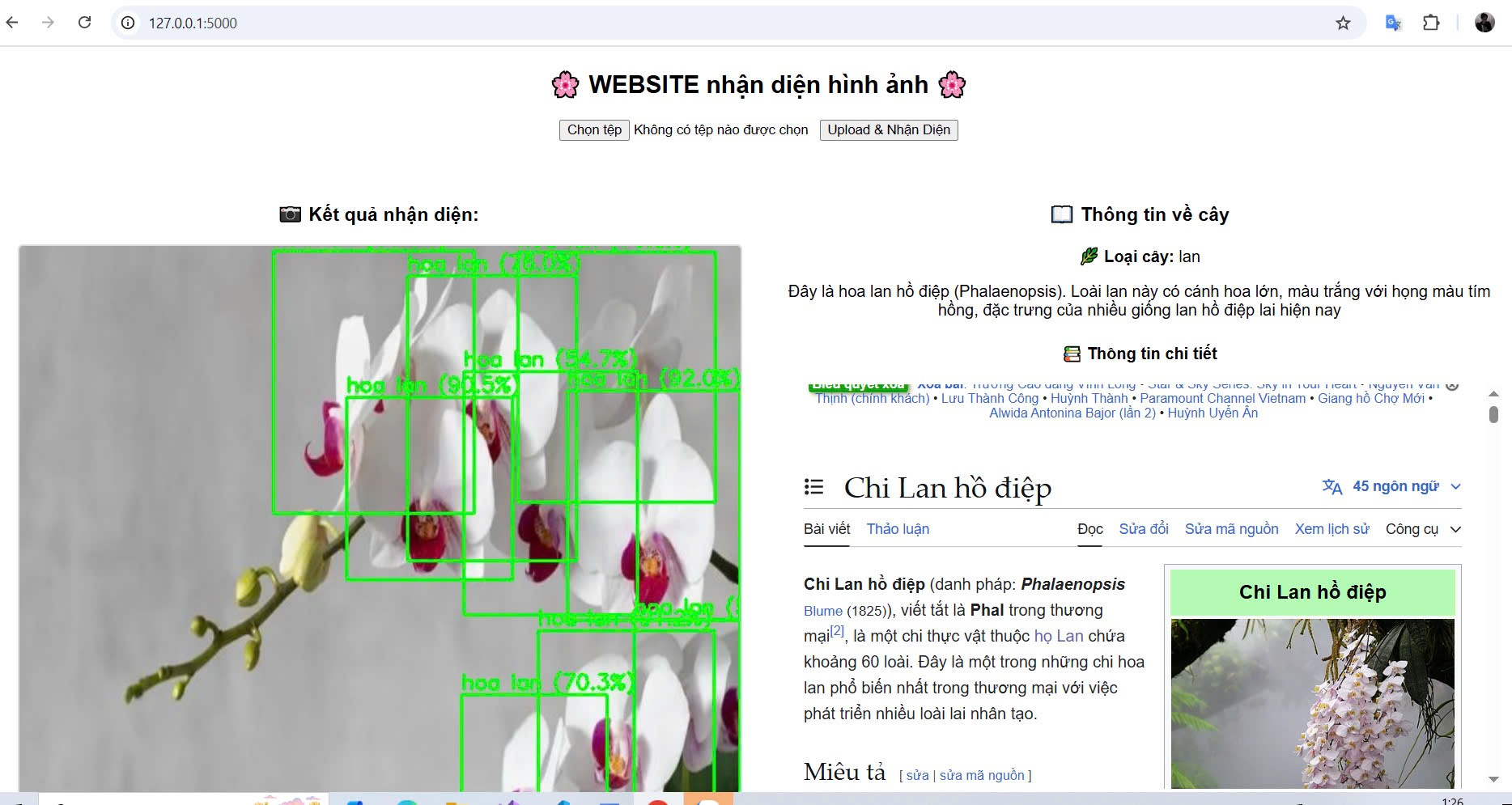


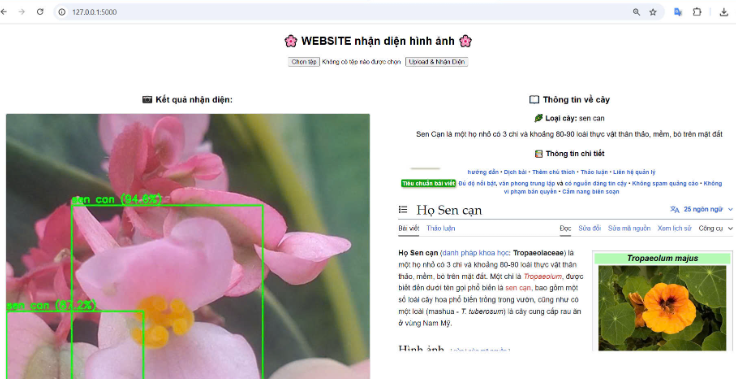
Kết quả thu được sau huấn luyện nhận diện sen cạn



Kết quả thu được sau huấn luyện nhận diện hoa đào

* Hình ảnh web hoàn thiện:





Khi huấn luyện 250 epochs, kết quả thu được:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Class | Images | Instances | Box(P) | R | mAP50 | mAP50-95 |
| all | 250 | 954 | 0.992 | 0.991 | 0.994 | 0.955 |
| hoa ban | 50 | 50 | 0.994 | 1 | 0.995 | 0.984 |
| hoa đào | 50 | 70 | 0.995 | 1 | 0.995 | 0.984 |
| sen cạn | 50 | 204 | 0.998 | 0.991 | 0.995 | 0.970 |
| hoa lan | 50 | 95 | 0.985 | 1 | 0.995 | 0.964 |
| me dat | 50 | 535 | 0.977 | 0.955 | 0.992 | 0.875 |

Chú thích:

* Images → Số lượng ảnh trong tập kiểm tra.
* Instances → Số lượng đối tượng của từng loại trong tập kiểm tra.
* Box(P) (Precision - Độ chính xác)
  + Tỷ lệ dự đoán đúng trên tổng số dự đoán (nếu có 100 dự đoán và 98 đúng thì P = 0.98).
* R (Recall - Độ nhạy)
* Tỷ lệ mô hình tìm được đúng số lượng thực tế có trong ảnh (nếu có 50 hoa Sen Cạn mà tìm được 41 thì R = 0.82).
* mAP50 (Mean Average Precision ở ngưỡng IoU 50%)
* Giá trị trung bình của độ chính xác dự đoán khi IoU ≥ 50%.
* Cao nhất là 1.0 (hoàn hảo), thường trên 0.9 là tốt.
* mAP50-95 (Mean Average Precision trung bình từ IoU 50% đến 95%)
* Giá trị chính xác tổng quát ở nhiều mức IoU, phản ánh khả năng mô hình dự đoán chính xác vị trí đối tượng.
* Thường thấp hơn mAP50, trên 0.8 là ổn.

## Đánh giá và nhận xét

* Mô hình đạt độ chính xác khá cao với hầu hết các loại hoa.
* Lớp “hoa ban”, “hoa dao”, “sen can”, “hoa lan” đều đạt [mAP@0.5](mailto:mAP@0.5): 0.95 > 0.96 → rất tốt.
* Me chua đất có số lượng mẫu cao nhất (535 instances), nhưng lại có Recall thấp nhất (0.955) → Có thể dữ liệu mẫu hơi khó phân biệt.
* Thời gian inferencing nhanh (~2.7ms/ảnh), phù hợp cho ứng dụng thời gian thực.

## Kết luận

* Quá trình huấn luyện mô hình YOLO để nhận diện hoa đã đạt kết quả tốt với độ chính xác khá cao.
* Ứng dụng web demo hoạt động ổn định, giúp kiểm tra mô hình dễ dàng.
* Tuy nhiên, mô hình vẫn cần cải thiện về khả năng nhận diện trong điều kiện thực tế phức tạp hơn.

## Đề xuất phát triển

* Mở rộng tập dữ liệu bằng cách thu thập thêm ảnh từ các nguồn khác nhau.
* Tăng số lượng epoch và sử dụng kỹ thuật augmentation nâng cao để cải thiện hiệu suất mô hình.
* Thử nghiệm các mô hình khác như EfficientDet hoặc Faster R-CNN để so sánh kết quả.
* Cải thiện ứng dụng Web bằng cách tối ưu hóa tốc độ inference và giao diện người dùng.