

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG



ĐỒ ÁN TỐT NGHIỆP

NGÀNH : CÔNG NGHỆ THÔNG TIN

Sinh viên: NGUYỄN TUẤN HÙNG

Giảng viên hướng dẫn: TS. HỒ THỊ HƯƠNG THƠM

HẢI PHÒNG - 2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

ĐỀ TÀI : PHÁT HIỆN LOẠI TRÁI CÂY
(CAM, TÁO, LÊ, ĐÀO, ...)

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
NGÀNH : CÔNG NGHỆ THÔNG TIN

Sinh viên: NGUYỄN TUẤN HÙNG

Giảng viên hướng dẫn: TS. HỒ THỊ HƯƠNG THƠM

HẢI PHÒNG, THÁNG 5 - 2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên : Nguyễn Tuấn Hùng - **MSV** : 1912111009

Lớp : CT2301M

Ngành : Công Nghệ Thông Tin

Tên đề tài : Phát hiện loại trái cây

NHIỆM VỤ ĐỀ TÀI

❖ Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

a) Mô tả tóm tắt đề tài

Tìm hiểu về mô hình phát hiện đối tượng bằng YOLOv8 cho bài toán phát hiện loại trái cây, thiết kế xây dựng chương trình phát hiện loại trái cây.

b) Nội dung hướng dẫn

- Tìm hiểu về bài toán phát hiện đối tượng nói chung và phát hiện đối tượng bằng YOLOv8.
- Phân tích, thu thập dữ liệu hình ảnh trái cây để huấn luyện phát hiện đối tượng.
- Cài đặt mô hình phát hiện đối tượng bằng YOLOv8
- Thử nghiệm mô hình trên hình ảnh và video để phát hiện trái cây.
- Nhận xét đánh giá, kết luận và đưa ra khuyến nghị khi ứng dụng.

c) Kết quả cần đạt được

- Đã xây dựng được mô hình phát hiện đối tượng trái cây bằng YOLOv8.
- Xây dựng được ứng dụng phát hiện loại trái cây trên hình ảnh và video.
- Viết báo cáo đồ án tốt nghiệp.

❖ Các tài liệu, số liệu cần thiết

- Tài liệu tham khảo lập trình ngôn ngữ Python.
- Tài liệu về phương pháp nhận diện đối tượng bằng YOLO.

❖ Địa điểm thực tập tốt nghiệp

- Công Ty TNHH Đào Tạo và Tích Hợp Công Nghệ e-Teck

CÁC CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Họ và tên : Hồ Thị Hương Thơm

Học hàm, học vị : Tiến sĩ

Cơ quan công tác : Trường Đại Học Hàng Hải Việt Nam

Nội dung hướng dẫn :

- Tìm hiểu về bài toán phát hiện đối tượng nói chung và phát hiện phát hiện đối tượng bằng YOLOv8.
- Phân tích, thu thập dữ liệu hình ảnh trái cây để huấn luyện phát hiện đối tượng.
- Cài đặt mô hình phát hiện đối tượng bằng YOLOv8
- Thử nghiệm mô hình trên hình ảnh và video để phát hiện trái cây.
- Nhận xét đánh giá, kết luận và đưa ra khuyến nghị khi ứng dụng.

Kết quả cần đạt được:

- Đã xây dựng được mô hình phát hiện đối tượng trái cây bằng YOLOv8.
- Xây dựng được ứng dụng phát hiện loại trái cây trên hình ảnh và video.
- Viết báo cáo đồ án tốt nghiệp.

Đề tài tốt nghiệp được giao ngày tháng 05 năm 2024

Yêu cầu phải hoàn thành xong trước ngày ... Tháng 05 năm 2024

Đã nhận nhiệm vụ ĐTTN

Sinh viên

Đã giao nhiệm vụ ĐTTN

Giảng viên hướng dẫn

Nguyễn Tuấn Hùng

TS. Hồ Thị Hương Thơm

Hải Phòng, ngày 25 tháng 05 năm 2024

TRƯỞNG KHOA

Cộng Hòa Xã Hội Chủ Nghĩa Việt Nam

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN TỐT NGHIỆP

Họ và tên giảng viên: Hồ Thị Hương Thơm

Đơn vị công tác : Trường Đại Học Hàng Hải

Họ và tên sinh viên : Nguyễn Tuấn Hùng

Chuyên ngành : Công Nghệ Thông Tin

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp

.....
.....
.....
.....

2. Đánh giá chất lượng của đề án/khóa luận (so với nội dung yêu cầu đã đề ra trong nhiệm vụ Đ.T.T.N, trên các mặt lý luận, thực tiễn, tính toán số liệu...)

.....
.....
.....

3. Ý kiến của giảng viên hướng dẫn tốt nghiệp

Được bảo vệ Không được bảo vệ Điểm hướng dẫn

Hải Phòng, ngày.....tháng.....năm 2024

Giảng viên hướng dẫn

(ký và ghi rõ họ tên)

Cộng hòa xã hội chủ nghĩa Việt Nam

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN CHẤM PHẢN BIỆN

Họ và tên giảng viên: Phùng Anh Tuấn

Đơn vị công tác: Trường Quản Lý Và Công Nghệ Hải Phòng

Họ và tên sinh viên: Nguyễn Tuấn Hùng - **Chuyên ngành:** Công Nghệ Thông Tin

Đề tài tốt nghiệp: Phát Hiện Loại Trái Cây

1. Phần nhận xét của giảng viên chấm phản biện

.....
.....
.....
.....

2. Những mặt còn hạn chế

.....
.....
.....
.....

3. Ý kiến của giảng viên chấm phản biện

Được bảo vệ Không được bảo vệ Điểm phản biện

Hải Phòng, ngày.....tháng.....năm 2024

Giảng viên chấm phản biện

(ký và ghi rõ họ tên)

Lời cảm ơn

Đồ án tốt nghiệp được hoàn thành tại Trường Đại Học Quản Lý và Công Nghệ Hải Phòng. Trong thời gian học tập và làm đề tài đồ án “ Phân hiện loại trái cây (cam, táo, lê, đào, ...)” em đã nhận được rất nhiều sự hướng dẫn chỉ bảo để hoàn thiện đề tài.

Điều đầu tiên, em xin gửi lời cảm ơn trân thành đến có TS. Hồ Thị Hương Thom. Cô đã hướng dẫn, chỉ bảo, truyền đạt kiến thức tâm huyết để em có thể hoàn thành đề tài đồ án này.

Tiếp đến em xin chân thành cảm ơn đến Trường Đại Học Quản Lý và Công Nghệ Hải Phòng nơi em đã học tập, các học viên nhóm lớp Công Nghệ Thông Tin khóa 2019 – 2023 đã tạo điều kiện, giúp đỡ, cổ vũ về tinh thần cho em trong suốt quá trình học tập, hoàn thiện đồ án.

Trân trọng

Hải Phòng, ngày 15 tháng 5 năm 2024

Sinh Viên

Nguyễn Tuấn Hùng

LỜI CAM ĐOAN

Sinh viên xin cam đoan đề tài “Phát Hiện Loại Trái Cây (cam, táo, lê, đào, ...)” là công trình nghiên cứu độc lập dưới sự hướng dẫn của TS. Hồ Thị Hương Thom – Giảng viên Khoa Công nghệ thông tin Trường Đại Học Hàng Hải Việt Nam. Đề tài là sự cố gắng, quyết tâm của cá nhân sinh viên trong việc tiếp cận kiến thức chuyên môn mới để hoàn thành đạt mục tiêu của đề tài. Các dữ liệu đề nghiên cứu thực nghiệm là trung thực có nguồn gốc trích dẫn rõ ràng.

Sinh viên xin hoàn toàn chịu trách nhiệm với lời cam đoan này

Người cam đoan

Nguyễn Tuấn Hùng

MỤC LỤC

Lời Cam Đoan	7
Mục Lục	8
Danh mục hình ảnh.....	11
Danh sách bảng	13
Bảng các từ khóa Viết Tắt	13
Chương 1: Giới thiệu tổng quan.....	14
1.1. Đặt vấn đề.....	14
1.2. Phát hiện đối tượng	16
1.2.1. Khái niệm.....	16
1.2.2. Cách phát hiện đối tượng.....	17
1.2.3. Các Phương pháp phát hiện phổ biến hiện nay	19
1.2.3.1. Phương pháp mô tả đặc trưng (HOG)	19
1.2.3.2. Mạng nơ-ron.....	20
1.3. Bài toán phát hiện loại trái cây.....	24
1.4. Đánh giá độ tin cậy của bài toán phân loại.....	25
1.4.1. Precision và Recall	25
1.4.2. AP.....	26
1.4.3. mAP	27
Chương 2: mô hình hệ thống phân loại trái cây	28
2.1. Giới thiệu một số loại trái cây	28
2.4.1. Cam.....	28
2.4.2. Xoài.....	29
2.4.3. Táo	30

2.4.4. Chuối.....	30
2.4.5. Nho.....	31
2.4.6. Ổi.....	32
2.4.7. Dưa Hấu.....	33
2.2. Sơ đồ hoạt động của mô hình hệ thống phát hiện trái cây.....	34
2.3. Quá trình Huấn Luyện.....	34
2.3.1. Chuẩn bị ảnh.....	34
2.3.2. Quá trình đánh nhãn cho các đối tượng.....	35
2.4. Phát hiện loại trái cây bằng YOLO.....	35
2.4.1. YOLO.....	35
2.4.2. Các phiên bản.....	36
2.4.2.1. YOLOv1.....	36
2.4.2.2. YOLOv2.....	37
2.4.2.3. YOLOv3.....	37
2.4.2.4. YOLOv4.....	39
2.4.2.5. YOLOv5.....	40
2.4.2.6. YOLOv6.....	41
2.4.2.7. YOLOv7.....	42
2.4.2.8. YOLOv8.....	43
Chương 3: Cài đặt, thử nghiệm và đánh giá.....	49
3.1. Môi trường thử nghiệm.....	49
3.1.1. Ngôn ngữ lập trình Python.....	49
3.1.1.1. Khái niệm.....	49
3.1.1.2. Đặc tính.....	49

3.1.1.3. Các phiên bản.....	50
3.1.2. Môi trường phát triển tích hợp Pycharm.....	51
3.1.2.1. Khái niệm:.....	51
3.1.2.2. Các đặc trưng:	52
3.1.2.3. Các phiên bản.....	52
3.1.3. Công cụ trực tuyến Google Collab.....	53
3.1.3.1. Khái niệm.....	53
3.1.3.2. Những đặc trưng chính trong Google Colab	53
3.2. Tạo bộ dữ liệu học để huấn luyện.....	54
3.3. Đào tạo mô hình phát hiện loại trái cây	57
3.3.1. Giá trị các biểu đồ.....	60
3.3.2. Giao diện ứng dụng phát hiện loại trái cây	62
3.3.3. Kết quả thử nghiệm trên mô hình đã huấn luyện	63
3.4. Đánh giá mô hình sau khi thử nghiệm.....	65
3.4.1. Kết quả huấn luyện:	65
3.4.2. Thời gian huấn luyện:	66
3.4.3. Ưu điểm của mô hình:	66
3.4.4. Nhược điểm của mô hình:	66
3.4.5. Đề xuất cải tiến:	66
Kết Luận	67
TÀI LIỆU THAM KHẢO.....	68

DANH MỤC HÌNH ẢNH

Hình 1.1: Công nhân đang phân loại trái cây.....	14
Hình 1.2: Trái cây được xếp trong siêu thị.	15
Hình 1.3: Phát hiện đối tượng.....	16
Hình 1.4: Phát hiện đối tượng dựa trên học sâu cho các phương tiện.....	18
Hình 1.5: Kiến trúc hệ thống HOG để phát hiện đối tượng.	19
Hình 1.6: Mô hình R-CNN. [17].....	20
Hình 1.7: Faster R-CNN – Sơ đồ kiến trúc. [2].....	22
Hình 1.8: Mạng đề xuất vùng (RPN). [18]	22
Hình 1.9: ví dụ biểu đồ thể hiện P và R để tính AP	27
Hình 2.1 : Hình dạng quả cam	28
Hình 2.2 : Hình ảnh trái xoài	29
Hình 2.3: Hình ảnh trái táo	30
Hình 2.4: Trái Chuối	31
Hình 2.5: Hình ảnh nho.....	32
Hình 2.6: Trái ổi chín	32
Hình 2.8: Dưa Hấu	33
Hình 2.9: Một số hình ảnh dưa hấu khác.....	33
Hình 2.10 : Sơ đồ hoạt động của mô hình phát hiện trái cây	34
Hình 2.11: Ứng dụng YOLOv3 cho tử lạnh thông minh trong ẩm thực và nhà hàng. .	38
Hình 2.12: Phát hiện đối tượng nhỏ với YOLOv5 trong phân tích giao thông.....	41
Hình 2.13: Ứng dụng hệ thống AI được đào tạo để phát hiện máy bay với YOLOv7 ...	42
Hình 2.14: Kiến trúc YOLOv8. [22].....	45
Hình 2.15: Hình dung anchor box trong YOLO.....	46

Hình 2.16: detection head của YOLOv5	46
Hình 2.17: detection head của YOLOv8	47
Hình 2.18: Hiệu suất trên Tập dữ liệu Phát hiện Đối tượng MS COCO.....	48
Hình 3.1: đặc trưng tăng cường hình ảnh trên Roboflow.....	55
Hình 3.2: Health check được thống kê trên Roboflow.....	56
Hình 3.3: cài các thư viện hỗ trợ.....	57
Hình 3.4: Ấn chọn biểu tượng drive trong phân tệp.....	58
Hình 3.5: Đoạn mã kết nối Google Colab với Google Drive.....	58
Hình 3.6: đoạn mã để bắt đầu đào tạo	58
Hình 3.7: Biểu đồ thống kê sau khi đào tạo.....	60
Hình 3.8: ma trận nhầm lẫn (hiệu suất dự đoán của mô hình với epochs=100.....	61
Hình 3.9: giao diện chương trình.....	62
Hình 3.10: trái xoài bị nhận sai.....	63
Hình 3.11: chùm nho bị nhận diện sai	64
Hình 3.12: Kết quả phân loại trái cây trên ảnh tĩnh có nhiều loại quả.....	64
Hình 3.13: nhận diện bằng video	65

DANH SÁCH BẢNG

Bảng	Tên bảng	Trang
3-1	Số lượng ảnh của từng loại	58
3-2	Bảng thử nghiệm với các epochs khác nhau	54
3-3	Bảng thử nghiệm phân loại hình ảnh tĩnh	67

BẢNG CÁC TỪ KHÓA VIẾT TẮT

Từ Viết Tắt	Nghĩa Tiếng Anh	Nghĩa Tiếng Việt
AI	Artificial Intelligence	Trí tuệ nhân tạo
AP	Average Precision	Độ chính xác trung bình
CNN	Convolutional Neural Network	Mạng nơ ron tích chập
FN	False Negative	Âm tính giả
FP	False Positive	Dương tính giả
GPU	Graphics Processing Unit	Đơn vị xử lý đồ họa
HOG	Histogram of Oriented Gradient	Biểu đồ Gradient có hướng
IDE	Integrated Development Environment	môi trường phát triển tích hợp
IoU	Intersection over Union	Giao điểm qua Liên hiệp
P	Precision	Độ chính xác
R	Recall	Độ nhớ lại
TN	True Negative	Âm tính thực
TP	True Positive	Dương tính thực
TPU	Tensor Processing Unit	Đơn vị Xử lý Tensor
YOLO	You Only Look One	Bạn chỉ nhìn một lần

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN

1.1. Đặt vấn đề

Với sự tiến bộ không ngừng của công nghệ và sự tăng trưởng nhanh chóng của dữ liệu thông qua mạng lưới internet toàn cầu, chúng ta đã chứng kiến nhiều bước tiến vượt bậc trong lĩnh vực học máy, đặc biệt là thị giác máy tính. Khả năng này có thể được áp dụng một cách hiệu quả trong ngành sản xuất và bán lẻ trái cây, nơi việc phát hiện và phân loại trái cây đóng vai trò then chốt để đảm bảo chất lượng và tính nhất quán của sản phẩm. Tuy nhiên, việc này có thể tốn kém và mất thời gian nếu thực hiện bằng tay, đặc biệt khi xử lý với số lượng lớn. Do đó, việc sử dụng công nghệ để phát hiện và phân loại trái cây có thể cải thiện đáng kể hiệu suất và chất lượng sản phẩm.



Hình 1.1: Công nhân đang phân loại trái cây.

Trong nông nghiệp, người nông dân phải xử lý hàng nghìn quả từ nhiều loại trái cây khác nhau, việc phân loại quả hư, dập hoặc kém chất lượng là thách thức lớn, tốn nhiều thời gian và công sức. Bài toán đặt ra là làm sao để giúp nhanh

chống loại bỏ sản phẩm không đạt chất lượng, nâng cao chất lượng tổng thể và giá trị bán ra, cải thiện thu nhập cho nông dân. Ngoài ra làm cách nào để có thể phân biệt nguồn gốc trái cây, như nho Việt Nam hay nho Trung Quốc, giúp nhà phân phối đưa ra quyết định kinh doanh hiệu quả và người tiêu dùng an tâm hơn về nguồn gốc sản phẩm. Lúc này ta cần một công cụ để giúp ta giải quyết những vấn đề trên. Với đề tài “Phát hiện loại trái cây”, như là bước đầu tiên để bắt đầu xây dựng một mô hình nâng cao chất lượng nông nghiệp tạo nền tảng cho các mô hình tiên tiến hơn trong tương lai.



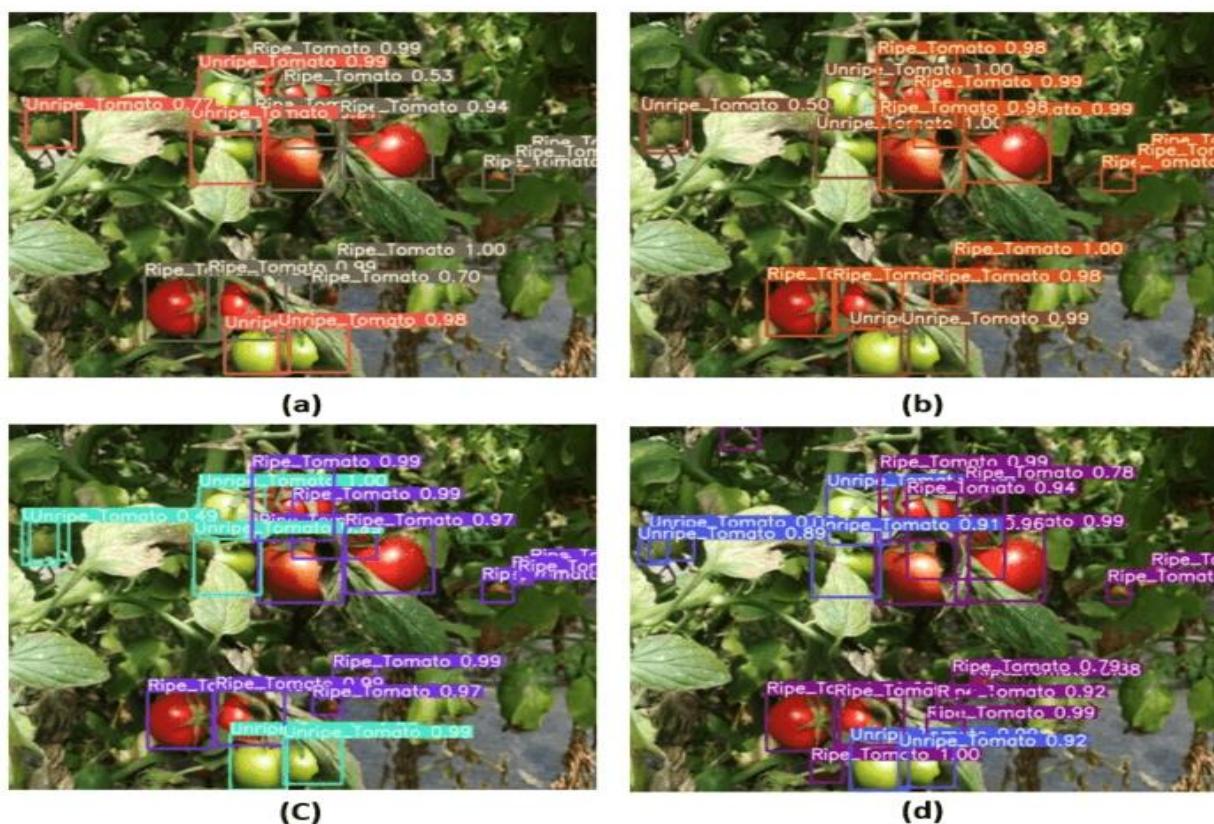
Hình 1.2: Trái cây được xếp trong siêu thị.

1.2. Phát hiện đối tượng

1.2.1. Khái niệm

Phát hiện đối tượng, một lĩnh vực quan trọng trong thị giác máy tính và xử lý ảnh, liên quan đến việc xác định các trường hợp cụ thể của đối tượng ngữ nghĩa thuộc một lớp nhất định trong hình ảnh và video kỹ thuật số. Các mô hình phát hiện đối tượng thường được huấn luyện để xác định các đối tượng cụ thể trong hình ảnh, video hoặc trong thời gian thực. [1]

Phát hiện đối tượng đóng vai trò quan trọng trong sự phát triển của học sâu và xử lý ảnh. Mục tiêu của nó là phát triển các mô hình tính toán để cung cấp thông tin cơ bản nhất cần thiết cho các ứng dụng thị giác máy tính, trả lời hai câu hỏi chính: ‘Đối tượng cần tìm có xuất hiện trong hình ảnh không?’ và ‘Nếu có, vị trí của nó là gì?’. [1]



Hình 1.3: Phát hiện đối tượng.

Phát hiện đối tượng là một trong những vấn đề cơ bản của thị giác máy tính, tạo nền tảng cho nhiều nhiệm vụ khác như phân đoạn thực thể và hình ảnh, tạo chú thích cho hình ảnh, theo dõi đối tượng, và nhiều hơn nữa. Các ứng dụng cụ thể của phát hiện đối tượng bao gồm phát hiện người đi bộ, phát hiện động vật, phát hiện xe cộ, đếm người, phát hiện khuôn mặt, phát hiện văn bản, phát hiện tư thế, nhận dạng biển số và như ở đây là phát hiện trái cây. [1]

1.2.2. Cách phát hiện đối tượng

Để phát hiện đối tượng, có thể sử dụng các kỹ thuật xử lý ảnh truyền thống hoặc mạng học sâu hiện đại. Mỗi phương pháp có ưu điểm và nhược điểm riêng, phù hợp với các yêu cầu và điều kiện khác nhau.

Cách 1: Kỹ thuật xử lý ảnh truyền thống

Các kỹ thuật xử lý ảnh truyền thống không yêu cầu dữ liệu lịch sử để đào tạo và tự giám sát. OpenCV là một công cụ phổ biến cho các nhiệm vụ xử lý ảnh truyền thống. Một số ưu và nhược điểm của phương pháp này bao gồm

Ưu điểm:

Không yêu cầu hình ảnh được gán nhãn thủ công, giúp giảm bớt khối lượng công việc và chi phí liên quan đến việc tạo dữ liệu đào tạo.

Nhược điểm:

- Hạn chế trong các tình huống phức tạp như nền không đồng nhất, đối tượng bị che khuất một phần, điều kiện ánh sáng và bóng tối thay đổi, và các hiệu ứng nhiễu.
- Độ chính xác và độ tin cậy thường không cao bằng các phương pháp học sâu. [1]

Cách 2: Mạng học sâu hiện đại

Các phương pháp học sâu thường phụ thuộc vào học có giám sát hoặc không giám sát, với các phương pháp học có giám sát là tiêu chuẩn trong các nhiệm vụ thị giác máy tính. Hiệu suất của các phương pháp này bị giới hạn bởi sức mạnh tính toán của GPU, đang tăng nhanh từng năm.

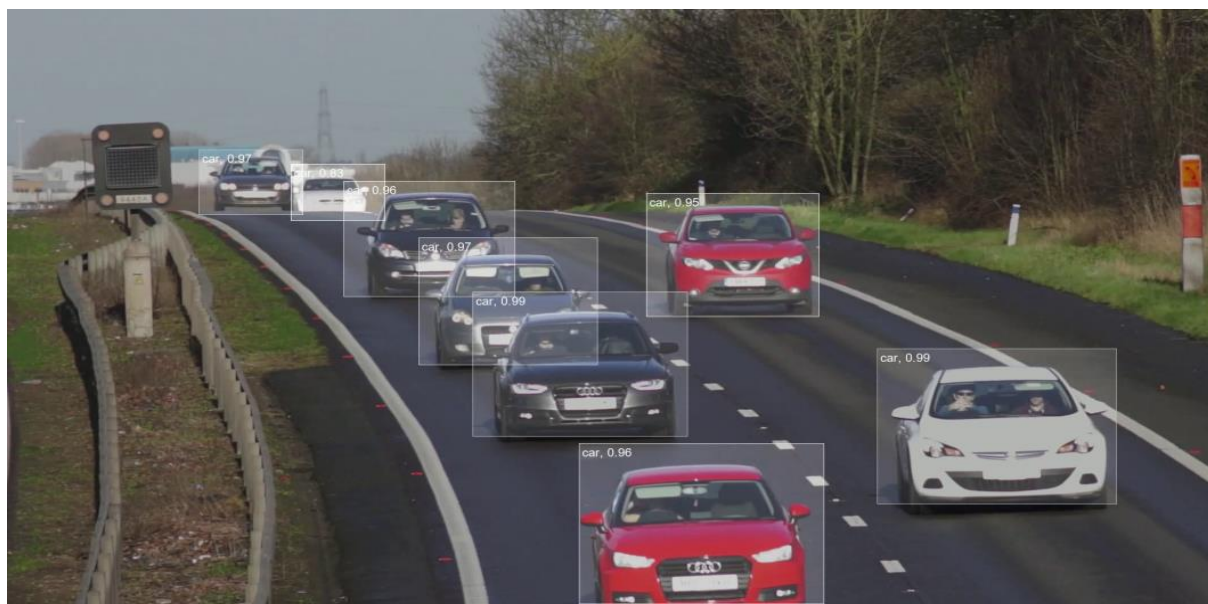
Ưu điểm:

- Phát hiện đối tượng bằng học sâu mạnh mẽ hơn đối với các trường hợp đối tượng bị che khuất, cảnh quan phức tạp và điều kiện ánh sáng thách thức.
- Độ chính xác cao hơn so với các phương pháp truyền thống.

Nhược điểm:

- Yêu cầu một lượng lớn dữ liệu đào tạo. Quá trình gán nhãn hình ảnh là công việc tốn nhiều thời gian và chi phí. Ví dụ, gán nhãn 500.000 hình ảnh để đào tạo thuật toán phát hiện đối tượng tùy chỉnh được coi là một tập dữ liệu nhỏ.
- Cần có sức mạnh tính toán cao, đặc biệt là GPU, để xử lý và đào tạo mô hình.

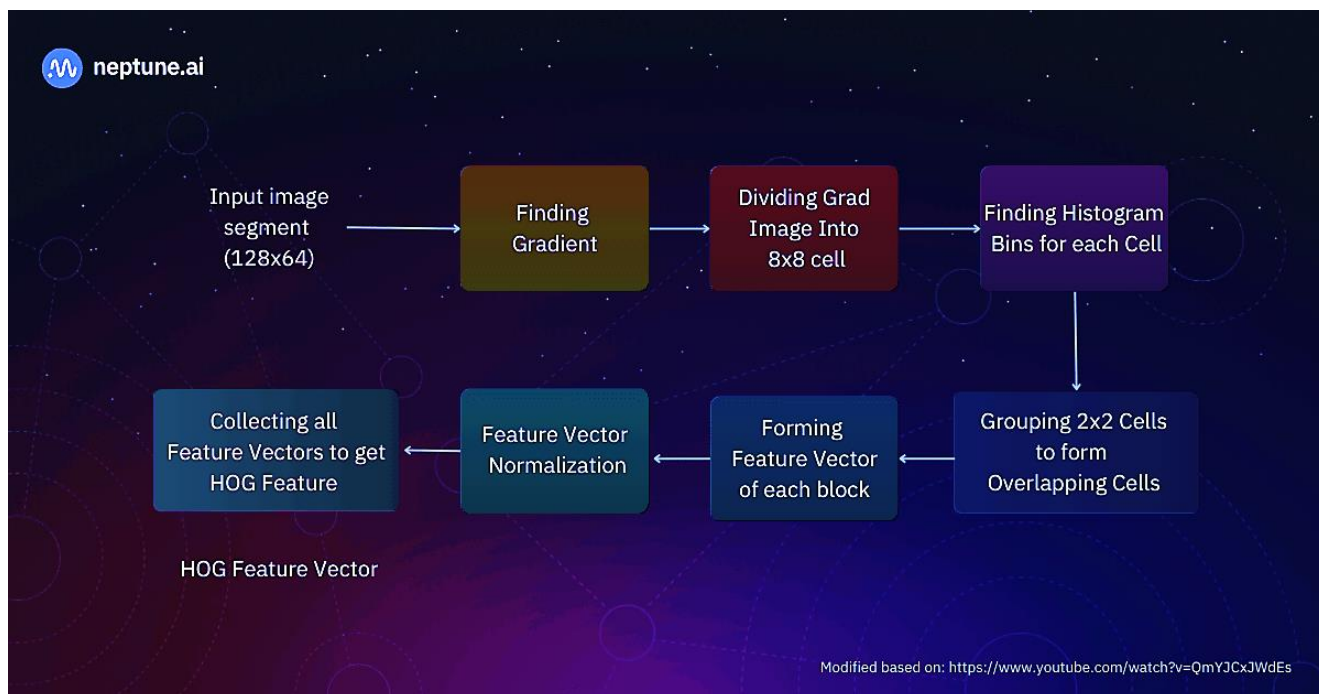
Để giảm bớt khó khăn trong việc thu thập và gán nhãn dữ liệu, có thể sử dụng các bộ dữ liệu đã được gán nhãn sẵn như MS COCO, Caltech, KITTI, PASCAL VOC, và V5. Những bộ dữ liệu này cung cấp một lượng lớn hình ảnh đã được gán nhãn, giúp rút ngắn quá trình thu thập và chuẩn bị dữ liệu cho việc đào tạo mô hình học sâu. [1]



Hình 1.4: Phát hiện đối tượng dựa trên học sâu cho các phương tiện.

1.2.3. Các Phương pháp phát hiện phổ biến hiện nay

1.2.3.1. Phương pháp mô tả đặc trưng (HOG)



Hình 1.5: Kiến trúc hệ thống HOG để phát hiện đối tượng.

Phương pháp mô tả đặc trưng (Histogram of Oriented Gradient – HOG) là một trong những phương pháp phát hiện đối tượng lâu đời nhất. Nó được giới thiệu lần đầu tiên vào năm 1986. Mặc dù có một số phát triển trong thập kỷ sau đó, cách tiếp cận này không trở nên phổ biến cho đến năm 2005 khi nó bắt đầu được sử dụng trong nhiều nhiệm vụ liên quan đến thị giác máy tính. HOG sử dụng trích xuất đặc trưng để xác định các đối tượng trong hình ảnh. [2]

Bản chất của phương pháp HOG là sử dụng thông tin về sự phân bố của các cường độ gradient hoặc của hướng biên để mô tả các đối tượng cục bộ trong ảnh. Các toán tử HOG được cài đặt bằng cách chia nhỏ một bức ảnh thành các vùng con, được gọi là cell và với mỗi cell, ta sẽ tính toán một histogram về các hướng của gradients cho các điểm nằm trong cell. Ghép các histogram lại với nhau ta sẽ có một biểu diễn cho bức ảnh ban đầu. Để tăng cường hiệu năng nhận dạng, các histogram cục bộ có thể được chuẩn hóa về độ tương phản bằng cách tính một ngưỡng cường độ trong một vùng lớn hơn cell, gọi là các khối và sử dụng giá trị ngưỡng đó để chuẩn hóa tất cả các

cell trong khối. Kết quả sau bước chuẩn hóa sẽ là một vector đặc trưng có tính bất biến đối với các thay đổi về điều kiện ánh sáng. [2]

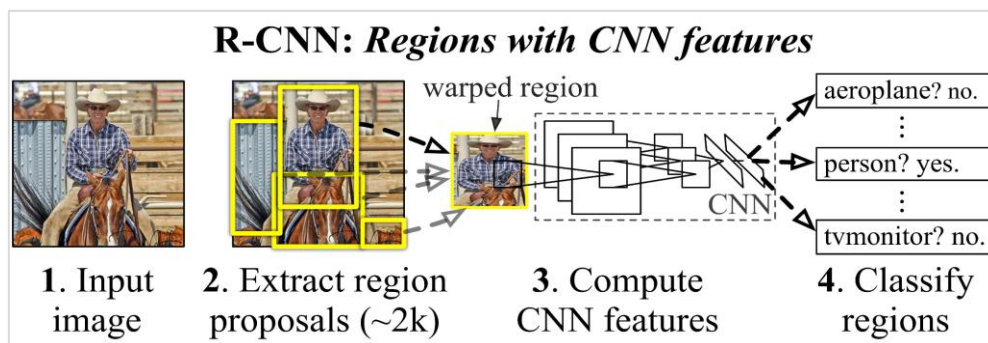
Trước khi chúng ta hiểu kiến trúc tổng thể của HOG, hãy xem cách nó hoạt động. Đối với một điểm ảnh cụ thể trong hình ảnh, biểu đồ tần suất của gradient được tính bằng cách xem xét các giá trị dọc và ngang để có được các vector đối tượng. Với sự trợ giúp của độ lớn gradient và các góc gradient, chúng ta có thể có được giá trị rõ ràng cho điểm ảnh hiện tại bằng cách khám phá các thực thể khác trong môi trường xung quanh ngang và dọc của chúng. [2]

Như thể hiện trong biểu diễn hình ảnh ở trên, chúng tôi sẽ xem xét một phân đoạn hình ảnh có kích thước cụ thể. Bước đầu tiên là tìm gradient bằng cách chia toàn bộ tính toán của hình ảnh thành các biểu diễn gradient của 8×8 ô. Với sự trợ giúp của 64 vector gradient đạt được, chúng ta có thể chia mỗi ô thành các góc và tính toán biểu đồ cho diện tích cụ thể. Quá trình này làm giảm kích thước của 64 vector xuống kích thước nhỏ hơn là 9 giá trị. [2]

Khi chúng tôi có được kích thước của các giá trị biểu đồ điểm 9 cho mỗi ô, chúng tôi có thể chọn tạo chồng chéo cho các khối ô. Các bước cuối cùng là tạo các khối đặc trưng, chuẩn hóa các vector đặc trưng thu được và thu thập tất cả các vector đặc trưng để có được đặc trưng HOG tổng thể. [2]

1.2.3.2. Mạng nơ-ron

- **Mạng nơ-ron tích chập theo vùng (R-CNN)**



Hình 1.6: Mô hình R-CNN. [17]

Các mạng nơ-ron tích chập dựa trên khu vực là một cải tiến trong quy trình phát hiện đối tượng từ các phương pháp HOG và SIFT (biến đổi đặc trưng không đổi theo tỉ lệ) trước đây. Trong các mô hình R-CNN, chúng ta cố gắng trích xuất các đặc trưng thiết yếu nhất (thường là khoảng 2000 đặc trưng) bằng cách sử dụng các đặc trưng chọn lọc. Quá trình lựa chọn các trích xuất quan trọng nhất có thể được thực hiện với sự trợ giúp của thuật toán tìm kiếm chọn lọc, nhằm đạt được các đề xuất khu vực quan trọng hơn. [2]

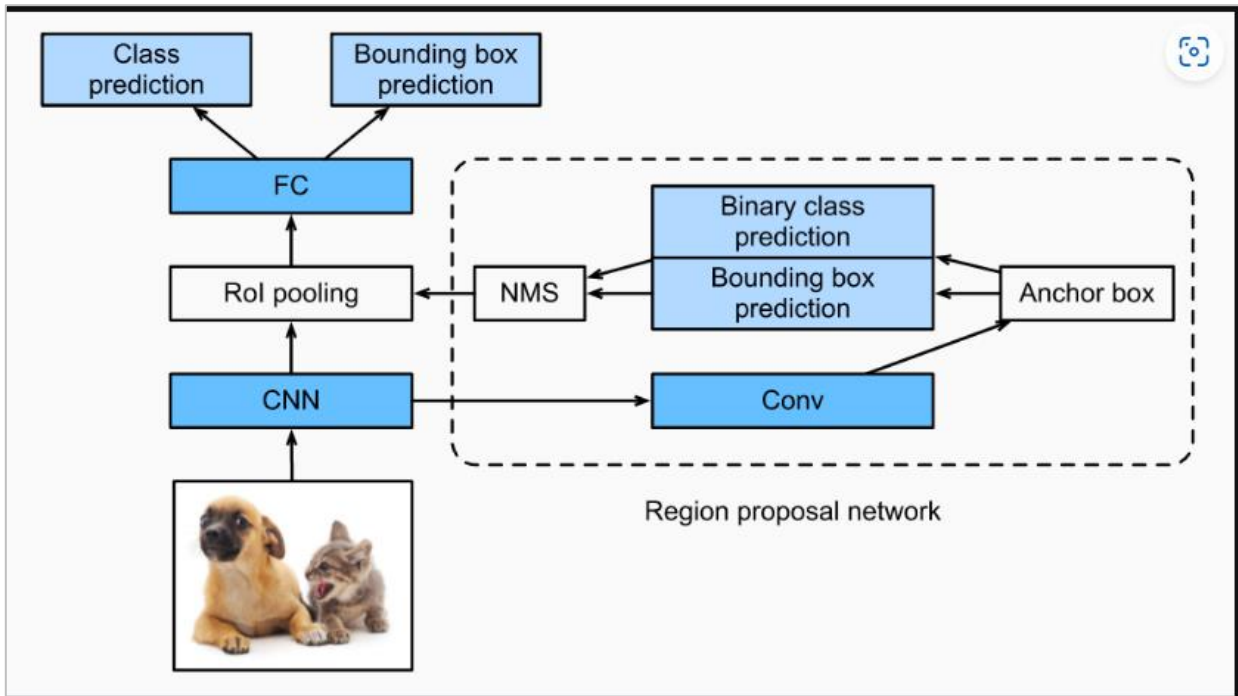
Quy trình làm việc của thuật toán tìm kiếm chọn lọc để chọn các đề xuất khu vực quan trọng nhất bao gồm việc tạo nhiều phân đoạn phụ trên một hình ảnh cụ thể và chọn các mục ứng cử viên cho nhiệm vụ của bạn. Thuật toán tham lam sau đó có thể được sử dụng để kết hợp các mục hiệu quả phù hợp cho một quy trình định kỳ, nhằm kết hợp các phân đoạn nhỏ hơn thành các phân đoạn lớn hơn phù hợp. [2]

Khi thuật toán tìm kiếm chọn lọc hoàn thành thành công, nhiệm vụ tiếp theo của chúng ta là trích xuất các đặc trưng và đưa ra dự đoán thích hợp. Sau đó, chúng ta có thể đưa ra các đề xuất ứng cử viên cuối cùng và các mạng nơ-ron tích chập có thể được sử dụng để tạo vector đặc trưng n chiều (2048 hoặc 4096) làm đầu ra. Với sự trợ giúp của mạng nơ-ron tích chập được đào tạo trước, chúng ta có thể đạt được nhiệm vụ trích xuất đặc trưng một cách dễ dàng. [2]

Bước cuối cùng của R-CNN là đưa ra các dự đoán thích hợp cho hình ảnh và gán nhãn hộp giới hạn tương ứng cho phù hợp. Để có được kết quả tốt nhất cho mỗi nhiệm vụ, các dự đoán được thực hiện bằng cách tính toán mô hình phân loại cho từng nhiệm vụ, trong khi mô hình hồi quy được sử dụng để sửa phân loại hộp giới hạn cho các vùng được đề xuất. [2]

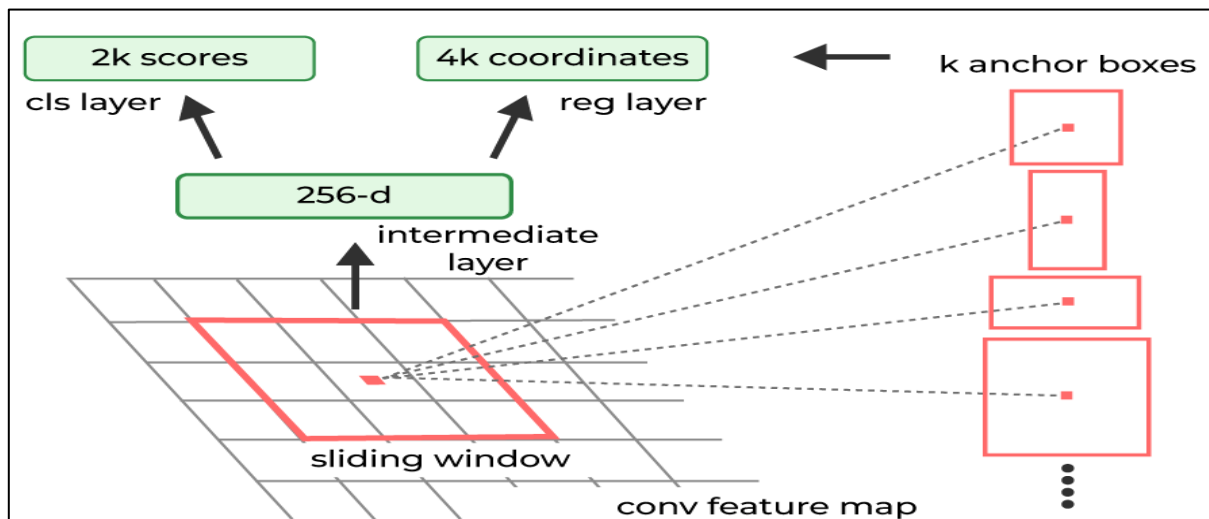
- **Faster R-CNN**

R-CNN, một mô hình phát hiện đối tượng, đã đạt được kết quả đáng chú ý nhưng lại gặp phải vấn đề về tốc độ. Để khắc phục điều này, Fast R-CNN đã được giới thiệu. Trong Fast R-CNN, toàn bộ hình ảnh được truyền qua một mạng nơ-ron tích chập đã được huấn luyện trước, thay vì chỉ xem xét các phân đoạn con của hình ảnh. [2]



Hình 1.7: Faster R-CNN – Sơ đồ kiến trúc. [2]

Tiếp theo, Faster R-CNN, một phiên bản nâng cấp của Fast R-CNN, đã được ra mắt. Faster R-CNN đã thay thế thuật toán tìm kiếm chọn lọc, được sử dụng trong R-CNN và Fast R-CNN để tính toán các đề xuất vùng, bằng một mạng đề xuất vùng vượt trội hơn. Mạng đề xuất vùng (RPN) tính toán hình ảnh từ một phạm vi rộng và các tỷ lệ khác nhau để tạo ra các đầu ra hiệu quả. [2]



Hình 1.8: Mạng đề xuất vùng (RPN). [18]

RPN giảm thời gian tính toán biên xuống còn khoảng 10 ms cho mỗi hình ảnh. Mạng này bao gồm một lớp tích chập từ đó chúng ta có thể lấy được các bản đồ đặc trưng cần thiết của mỗi pixel. Đối với mỗi bản đồ đặc trưng, chúng ta có nhiều hộp neo có các tỷ lệ, kích thước và tỷ lệ khía cạnh khác nhau. Đối với mỗi hộp neo, chúng ta dự đoán một lớp nhị phân cụ thể và tạo ra một hộp giới hạn cho nó. [2]

Thông tin sau đó được truyền qua giảm thiểu tối đa để loại bỏ bất kỳ dữ liệu không cần thiết nào vì nhiều chồng chéo được tạo ra khi tạo bản đồ đặc trưng. Đầu ra từ giảm thiểu tối đa được truyền qua vùng quan tâm, và phần còn lại của quá trình và tính toán tương tự như cách hoạt động của Fast R-CNN. [2]

- **YOLO**

YOLO (You Only Look Once) là một phương pháp phát hiện đối tượng rất phổ biến trong lĩnh vực thị giác máy tính. YOLO sử dụng một mạng nơ-ron tích chập (CNN) để xử lý hình ảnh và video. [21]

YOLO chia hình ảnh thành một lưới các ô và mỗi ô sẽ phát hiện các đối tượng bên trong nó. Mô hình này được biết đến với tốc độ nhanh và độ chính xác cao. Từ khi được giới thiệu lần đầu tiên bởi Joseph Redmon và cộng sự vào năm 2016, đã có nhiều phiên bản của YOLO, một trong những phiên bản gần đây nhất là YOLOv9.

Tuy nhiên, YOLO cũng có nhược điểm là khó khăn trong việc phát hiện các đối tượng nhỏ và có thể không phát hiện chính xác các đối tượng trong các cảnh đông đúc hoặc khi các đối tượng ở xa máy ảnh. [21]

Một sơ mô hình nổi bật của YOLO:

YOLOv3: Lần lặp lại thứ ba của YOLO họ mô hình, ban đầu bởi Joseph Redmon, được biết đến với khả năng phát hiện đối tượng thời gian thực hiệu quả.

YOLOv4: Bản cập nhật darknet gốc cho YOLOv3, được phát hành bởi Alexey Bochkovskiy vào năm 2020.

YOLOv5: Một phiên bản cải tiến của YOLO Kiến trúc của Ultralytics, cung cấp hiệu suất và tốc độ đánh đổi tốt hơn so với các phiên bản trước.

YOLOv6: Được Meituan phát hành vào năm 2022 và được sử dụng trong nhiều robot giao hàng tự động của công ty.

YOLOv7: Cập nhật YOLO mô hình được phát hành vào năm 2022 bởi các tác giả của YOLOv4.

YOLOv8: Phiên bản mới nhất của YOLO Dòng sản phẩm, có các chức năng nâng cao như phân đoạn phiên bản, ước tính tư thế/điểm chính và phân loại.

YOLOv9: Một mô hình thử nghiệm được đào tạo trên cơ sở mã nguồn mở YOLOv5 của Ultralytics, triển khai Thông tin Độ dốc Có thể Lập trình được.

Segment Anything Model: Mô hình phân khúc bất cứ thứ gì của Meta.

Mobile Segment Anything Model: cho các ứng dụng di động, bởi Đại học Kyung Hee.

Fast Segment Anything Model: bởi Nhóm phân tích hình ảnh & video, Viện Tự động hóa, Viện Hàn lâm Khoa học Trung Quốc.

YOLO-NAS: YOLO Mô hình tìm kiếm kiến trúc thần kinh.

Realtime Detection Transformers: Baidu của PaddlePaddle Biến áp phát hiện thời gian thực mô hình.

YOLO-World: Các mô hình phát hiện đối tượng từ vung mở thời gian thực từ Tencent AI Lab. [21]

1.3. Bài toán phát hiện loại trái cây

Với bài toán phát hiện loại trái cây (cam, táo, lê, đào, ...) bài toán cần đảm nhận được những vấn đề cơ bản nhất của một quá trình phát hiện đối tượng thông thường:

- Ứng dụng một mô hình học sâu hiệu quả trong phân loại và nhận dạng trái cây: Để đạt được mục tiêu này, cần tiến hành nghiên cứu và tìm hiểu về các mô hình học sâu phù hợp để phân loại và nhận dạng trái cây, từ đó xây dựng một mô hình học sâu chính xác và hiệu quả nhưng ít tốn kém.

- Tổng hợp và xây dựng tập dữ liệu ảnh trái cây tương đối đầy đủ: Để phân loại và nhận dạng trái cây, cần phải có tập dữ liệu ảnh tương đối đầy đủ, đảm bảo các loại trái cây được đại diện đầy đủ, chất lượng ảnh tốt và độ đa dạng cao.
- Thực hiện các thử nghiệm để đánh giá độ chính xác và hiệu quả của mô hình học sâu: Sau khi xây dựng mô hình học sâu và tập dữ liệu, cần thực hiện các thử nghiệm để đánh giá độ chính xác và hiệu quả của mô hình. Các thử nghiệm này bao gồm đo lường độ chính xác, độ nhạy và độ đặc hiệu của mô hình trong việc phân loại và nhận dạng các loại trái cây khác nhau.

1.4. Đánh giá độ tin cậy của bài toán phân loại

1.4.1. Precision và Recall

Precision – đại diện cho độ tin cậy của của model sẽ cho biết rằng trong những cái model dự đoán là Positive thì có bao nhiêu % là Positive thật.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (1-1)$$

Recall – đại diện cho độ nhạy của model sẽ cho biết model có thể tóm đúng được bao nhiêu Positive trong dữ liệu được cho.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (1-2)$$

True Positive (TP):

Đây là những trường hợp mà mô hình dự đoán đúng. Cụ thể, mô hình đã phát hiện chính xác một đối tượng thực sự có trong hình ảnh.

Ví dụ: Nếu có một quả táo trong hình và mô hình đúng là đã dự đoán đó là một quả táo, thì đây là một TP.

False Positive (FP):

Đây là những trường hợp mà mô hình dự đoán sai, tức là mô hình nhận diện một đối tượng mà thực tế không có.

Ví dụ: Nếu không có quả táo nào trong hình nhưng mô hình lại dự đoán rằng có một quả táo, thì đây là một FP.

True Negative (TN):

Đây là những trường hợp mà mô hình dự đoán đúng rằng không có đối tượng nào. Mô hình chính xác trong việc nhận diện sự vắng mặt của đối tượng.

Ví dụ: Nếu không có quả táo nào trong hình và mô hình cũng không dự đoán rằng có một quả táo, thì đây là một TN.

False Negative (FN):

Đây là những trường hợp mà mô hình bỏ sót đối tượng, tức là mô hình không phát hiện ra một đối tượng thực sự có trong hình.

Ví dụ: Nếu có một quả táo trong hình nhưng mô hình không phát hiện ra nó, thì đây là một FN.

1.4.2. AP

Average Precision là giá trị trung bình của Precision tại các mức Recall khác nhau. AP được tính cho từng lớp đối tượng và đại diện cho sự cân bằng giữa Precision và Recall.

Sử dụng Tính toán nội suy 11 điểm là tính trung bình độ chính xác ở một tập hợp 11 mức thu hồi (0,0,1,...,1). Các giá trị chính xác nội suy thu được bằng cách lấy độ chính xác tối đa có giá trị thu hồi lớn hơn giá trị thu hồi hiện tại của nó như sau:

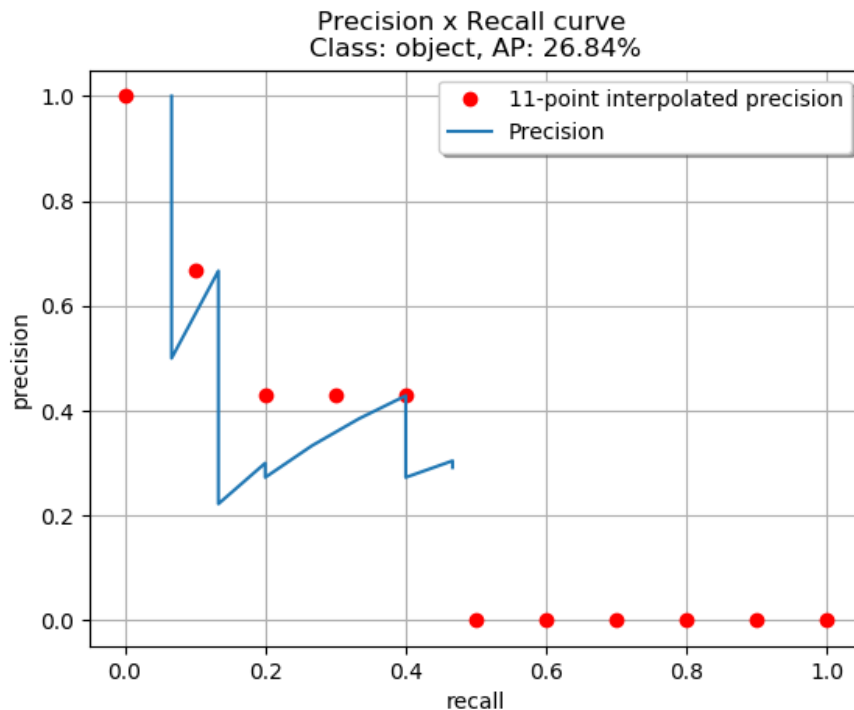
Trong đó :

$$AP = \frac{1}{11} \sum_{r \in \{0,0.1,\dots,1\}} P_{interp}(r) \quad (1-3)$$

$$p_{interp} = \max_{\tilde{r}: \tilde{r} \geq r} p(\tilde{r}) \quad (1-4)$$

$p(\tilde{r})$ là độ chính xác khi thu hồi \tilde{r}

Thay vì sử dụng độ chính xác quan sát được tại mỗi điểm, AP thu được bằng cách nội suy độ chính xác chỉ ở 11 mức r lấy độ chính xác tối đa có giá trị thu hồi lớn hơn r .



Hình 1.9: ví dụ biểu đồ thể hiện P và R để tính AP

Bằng cách áp dụng nội suy 11 điểm:

$$AP = \frac{1}{11} (1 + 0.6666 + 0.4285 + 0.4285 + 0.4285 + 0 + 0 + 0 + 0 + 0 + 0)$$

$$AP = 26.84\%$$

1.4.3. mAP

Mean Average Precision là một độ đo phổ biến trong các bài toán phát hiện đối tượng (object detection) để đánh giá hiệu suất của một mô hình.

Được tính bằng cách tính trung bình của tất cả các giá trị AP

Ví dụ: có tổng cộng 3 lớp AP lần lượt là a,b,c. Khi đó mAP được tính như sau:

$$mAP = \frac{AP_a + AP_b + AP_c}{3} \quad (1-5)$$

CHƯƠNG 2: MÔ HÌNH HỆ THỐNG PHÂN LOẠI TRÁI CÂY

2.1. Giới thiệu một số loại trái cây

2.4.1. Cam

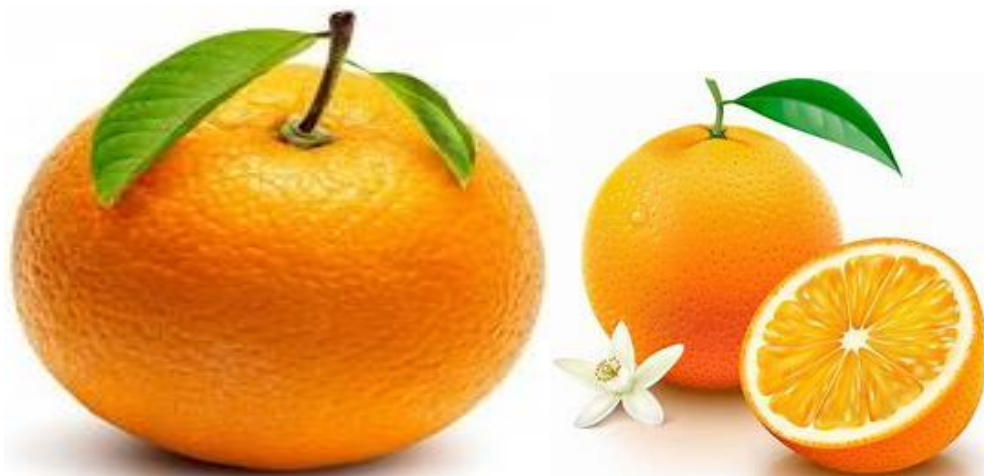
Cam quả cam tên tiếng anh là Orange, là một loại cây ăn quả cùng họ với bưởi nhưng kích thước nhỏ hơn và vỏ mỏng hơn, tùy vào các loại cam khác nhau mà khi chín có màu da cam, màu xanh hoặc màu vàng, có vị ngọt hoặc hơi chua. Cam bắt nguồn từ Châu Á, sau lan rộng ra khắp nơi trên thế giới, hiện nay cam có mặt ở khắp mọi nơi và nhân giống thành nhiều loại khác nhau. Ở nước ta cam có nhiều loại, phổ biến như:

- Cam sành: hình ảnh quả cam sành khá to, vỏ sần sùi, quả hơi dẹt, màu xanh, có mùi thơm và vị ngọt và chua dịu, nhiều nước, vụ thu hoạch thường đúng dịp Tết.

- Cam xoàn: trái nhỏ, thịt quả có màu vàng nhạt, hầu như không có hạt, nước ít nhưng rất ngọt thanh.

- Cam ham lin: có nguồn gốc từ Mỹ, du nhập vào Việt Nam khoảng từ năm 1971, vỏ quả mỏng, nhiều nước và có vị ngọt đậm đà.

- Cam Xã Đoài: tập trung nhiều ở vùng Xã Đoài, Nghệ An, quả ngọt thơm vị rất đặc trưng, nhiều nước nên rất được ưa chuộng. [3]



Hình 2.1 : Hình dạng quả cam

2.4.2. Xoài

Đã trở thành một biểu tượng của vùng nhiệt đới, phát triển mạnh mẽ trong khí hậu ẩm áp. Xoài có nguồn gốc từ Nam Á, đặc biệt Ấn Độ được biết đến như cái nôi của loại quả ngọt ngào này. Qua thời gian, xoài đã du nhập và phát triển mạnh mẽ ở nhiều quốc gia nhờ các cuộc hành trình của các nhà thám hiểm và thương nhân từ thời cổ đại. Quả xoài, từ lúc non mang màu xanh mát mắt, dần dần chuyển sang màu vàng rực rỡ khi chín. Thịt quả màu vàng tươi, mềm mại, đậm đà hương vị với một chút xơ nhẹ. Hình dạng của quả xoài thay đổi từ tròn đến bầu dục, tạo nên sự đa dạng trong từng loại.

Xoài Cát Hòa Lộc: có quả to, vỏ xanh, khi chín chuyển sang màu vàng.

Xoài Cát Chu: Xoài Cát Chu có quả nhỏ hơn xoài Cát Hòa Lộc, vỏ xanh, khi chín chuyển sang màu vàng.

Xoài keo: có quả nhỏ, vỏ xanh, khi chín chuyển sang màu vàng.

Xoài keo: có quả nhỏ, vỏ xanh, khi chín chuyển sang màu vàng. Thịt xoài dày, màu vàng cam, vị ngọt thanh, hơi chua, nhiều xơ.

Xoài Úc: có quả to, vỏ xanh, khi chín chuyển sang màu vàng. Thịt xoài dày, màu vàng cam. [4]



Hình 2.2 : Hình ảnh trái xoài

2.4.3. Táo

Táo có nguồn gốc từ châu Á và Kazakh; nó đã được phổ biến rộng rãi qua miền tây Trung Quốc, dọc theo Con đường Tơ lụa và qua Biển Đen. Bằng cách này, nó đã lan rộng khắp châu Âu và chinh phục thị hiếu của nhiều người, sau đó nó được du nhập vào lãnh thổ Châu Mỹ, cả Bắc Mỹ và Nam Mỹ, nơi nó có khả năng thích nghi cao ở những vùng lạnh nhất của lục địa và cho đến ngày nay nó được trồng với quy mô lớn, để buôn bán, tiêu dùng trong dân chúng và xuất khẩu.

Quả có màu hơi đỏ, một số quả có nhiều tói, những loại khác có tông màu sáng hơn và được coi là có kích thước trung bình, chỉ vài cm.



Hình 2.3: Hình ảnh trái táo

2.4.4. Chuối

Chuối là một loại cây ăn quả thuộc họ Musaceae, có nguồn gốc từ khu vực Đông Nam Á, đặc biệt là các nước như Malaysia, Indonesia, và Philippines. Từ đây, chuối đã được con người trồng và lan rộng ra khắp các khu vực nhiệt đới và cận nhiệt đới trên thế giới. Hiện nay, chuối được trồng phổ biến tại nhiều quốc gia, bao gồm Ấn Độ, Trung Quốc, Brazil, và các nước vùng Trung Mỹ và châu Phi. [6]

Quả chuối thường có hình dáng cong, vỏ màu vàng khi chín và màu xanh khi chưa chín. Bên trong là phần thịt màu trắng hoặc vàng nhạt, mềm, ngọt và nhiều nước. Chuối mọc thành từng nải, mỗi nải có nhiều quả. Mỗi cây chuối thường cho ra một buồng chuối, và mỗi buồng có thể chứa từ vài chục đến hàng trăm quả.

- Chuối tây (chuối tiêu): Có kích thước lớn, vỏ dày, thịt ngọt và mùi thơm đặc trưng.
- Chuối ngự: Kích thước nhỏ hơn, vỏ mỏng, thịt thơm ngon và thường được dùng trong các dịp lễ tết.
- Chuối sứ (chuối hột): Kích thước trung bình, vỏ mỏng, thịt ngọt dịu và thường được dùng làm nguyên liệu chế biến các món ăn.



Hình 2.4: Trái Chuối

2.4.5. Nho

Nho là một loại cây ăn quả có nguồn gốc từ khu vực Địa Trung Hải, Trung Á và Tây Nam Á. Từ đây, nho đã được trồng rộng rãi và phổ biến trên toàn thế giới, đặc biệt là ở các nước có khí hậu ôn đới và cận nhiệt đới. Những khu vực nổi tiếng về trồng nho bao gồm Pháp, Ý, Tây Ban Nha, Mỹ (California), Chile, và Úc. [8]

- Quả nho thường mọc thành chùm, mỗi chùm có thể chứa từ vài chục đến vài trăm quả.
- Quả nho có kích thước nhỏ, hình tròn hoặc hình bầu dục, vỏ mỏng, màu sắc đa dạng từ xanh, đỏ, tím đến đen tùy theo giống.
- Bên trong quả nho có thịt mềm, ngọt, chứa nhiều nước và thường có hạt nhỏ (một số giống nho không hạt cũng được trồng rộng rãi).



Hình 2.5: Hình ảnh nho

2.4.6. Ôi

Ôi là một loại cây ăn quả thuộc họ Myrtaceae, có nguồn gốc từ khu vực nhiệt đới châu Mỹ, cụ thể là Mexico, Trung Mỹ và phía bắc Nam Mỹ. Từ đây, ôi đã được trồng rộng rãi ở nhiều vùng nhiệt đới và cận nhiệt đới trên toàn thế giới, bao gồm các quốc gia châu Á như Ấn Độ, Thái Lan và Việt Nam. [9]

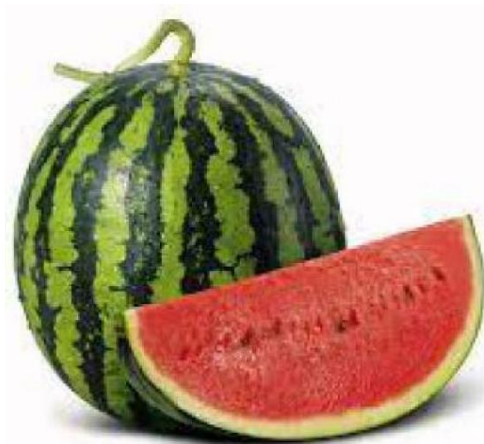


Hình 2.6: Trái ôi chín

- Quả ôi có hình dạng bầu dục hoặc hình tròn, kích thước thay đổi từ nhỏ đến trung bình, với đường kính khoảng 4-12 cm.
- Vỏ quả ôi có thể mỏng hoặc dày, màu sắc từ xanh nhạt đến vàng khi chín. Một số giống ôi có vỏ màu hồng nhạt.
- Thịt ôi bên trong có màu trắng, hồng, đỏ hoặc vàng, tùy thuộc vào giống. Thịt ôi giòn, ngọt và thơm, có nhiều hạt nhỏ cứng ở trung tâm.

2.4.7. Dưa Hấu

Dưa hấu là một loại cây ăn quả thuộc họ Cucurbitaceae. Dưa hấu có nguồn gốc từ vùng Tây Phi, nơi những dạng hoang dã của cây này vẫn còn tồn tại. Từ đây, dưa hấu đã được trồng và lan rộng ra khắp các vùng nhiệt đới và cận nhiệt đới trên thế giới. Hiện nay, dưa hấu được trồng phổ biến ở nhiều quốc gia như Trung Quốc, Ấn Độ, Thổ Nhĩ Kỳ, Brazil và Hoa Kỳ. [7]

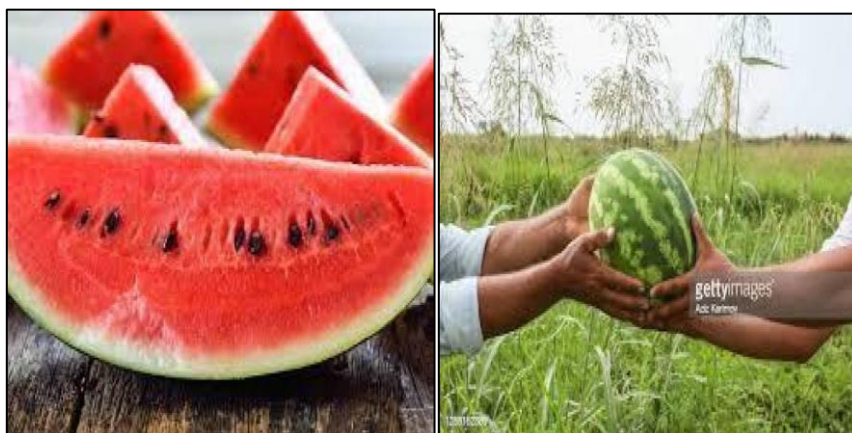


Hình 2.8: Dưa Hấu

Quả dưa hấu thường có hình tròn hoặc hình bầu dục, kích thước lớn, có thể nặng từ vài kilogram đến hơn 20 kilogram.

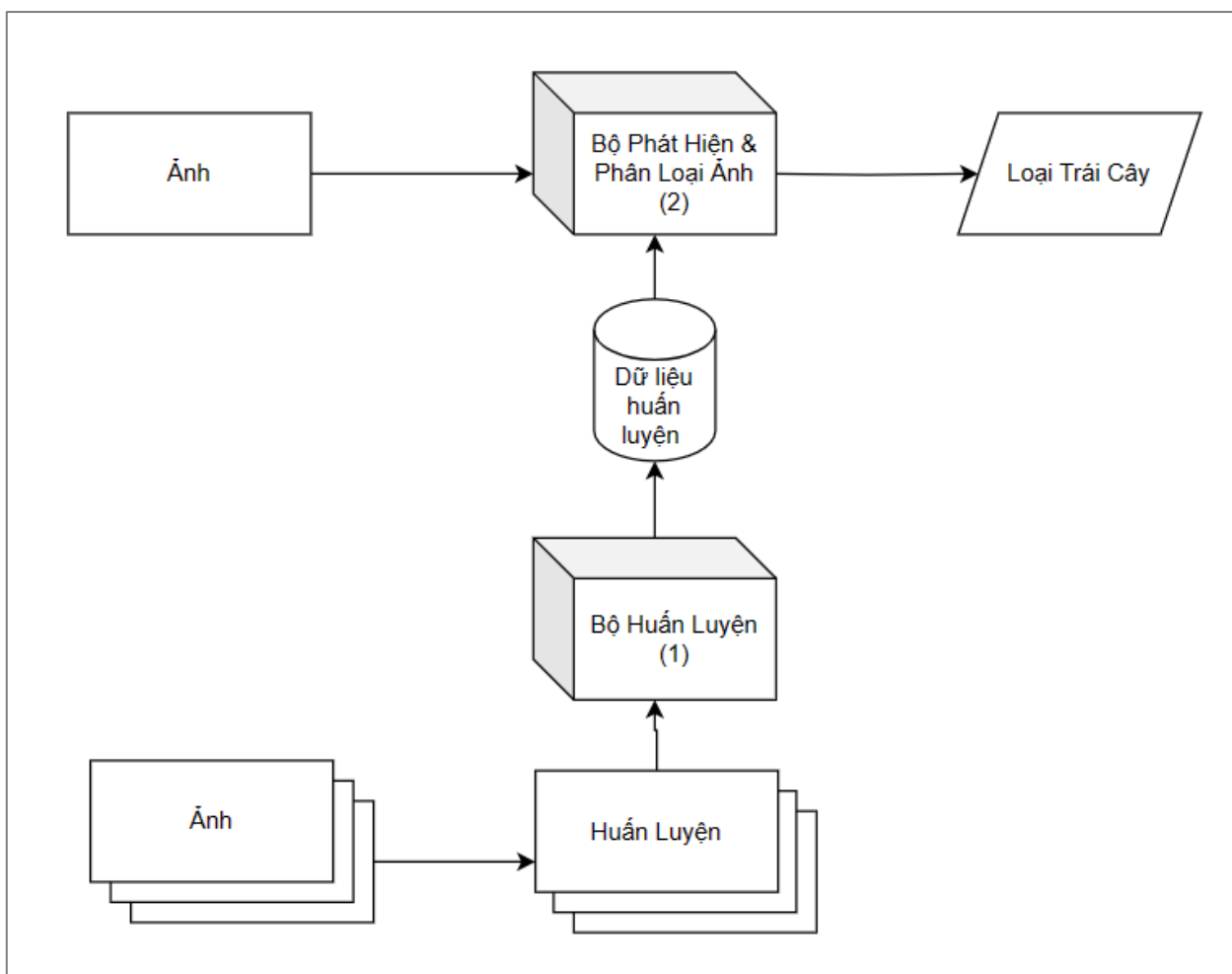
Vỏ dưa hấu thường dày, màu xanh lá cây, có thể có các vân hoặc đốm trắng.

Thịt quả dưa hấu có màu đỏ, hồng, vàng, hoặc cam, ngọt và nhiều nước. Bên trong thịt quả chứa nhiều hạt nhỏ, màu đen hoặc trắng. [7]



Hình 2.9: Một số hình ảnh dưa hấu khác

2.2. Sơ đồ hoạt động của mô hình hệ thống phát hiện trái cây.



Hình 2.10 : Sơ đồ hoạt động của mô hình phát hiện trái cây

(1) Bộ phát hiện và phân loại trái cây

(2) Bộ phát hiện loại trái cây

2.3. Quá trình Huấn Luyện

2.3.1. Chuẩn bị ảnh

- Chất lượng Ảnh: Ảnh cần có độ phân giải đủ cao để các đối tượng có thể được nhận diện rõ ràng, ngay cả khi chúng nhỏ hoặc ở xa.
- Số lượng ảnh: quả cam (1087), quả chuối (985), nho (875), dưa hấu (878), xoài (935), ổi (810), táo (1189).

- Đa dạng: Bộ dữ liệu nên bao gồm ảnh với các điều kiện ánh sáng khác nhau, góc chụp, và cảnh quan để mô hình có thể học cách phát hiện đối tượng trong nhiều hoàn cảnh.
- Gán Nhãn Chính Xác: Các bounding box cần được vẽ chính xác xung quanh đối tượng, không quá rộng hoặc quá hẹp so với kích thước thực tế của đối tượng.
- Cân Bằng Dữ Liệu: Tránh tình trạng mất cân bằng, nơi một số lớp có nhiều mẫu hơn đáng kể so với các lớp khác, điều này có thể dẫn đến độ chệch trong quá trình huấn luyện.
- Chia Tập Dữ Liệu: Chia bộ dữ liệu thành các tập train, validation, và test để đánh giá mô hình một cách khách quan.
- Tối Ưu Hóa Kích Thước Ảnh: Đôi khi cần phải điều chỉnh kích thước ảnh để phù hợp với yêu cầu đầu vào của mô hình, đồng thời giảm bớt gánh nặng tính toán. (640 là yêu cầu đối với YOLOv8).
- Augmentation: Sử dụng kỹ thuật tăng cường dữ liệu (data augmentation) như xoay, lật, thay đổi màu sắc, để tăng cường khả năng tổng quát hóa của mô hình.

2.3.2. Quá trình đánh nhãn cho các đối tượng

Sử dụng các công cụ có sẵn trên mạng như: CVAT, Labelbox, LabelMe, Roboflow.

Đối với việc đánh nhãn ta sẽ sử dụng Roboflow để là nơi tập chung ảnh và đánh nhãn cho các đối tượng (Cam, Xoài, Táo, Chuối, Nho, Ôi, Dưa Hấu).

2.4. Phát hiện loại trái cây bằng YOLO

2.4.1. YOLO

You Only Look Once (YOLO) đề xuất sử dụng một mạng nơ-ron đầu-cuối để đưa ra dự đoán về các hộp giới hạn và xác suất lớp cùng một lúc. Nó khác với cách tiếp cận của các thuật toán phát hiện đối tượng trước đó, mà sử dụng lại các bộ phân loại để thực hiện phát hiện.

Theo một cách tiếp cận hoàn toàn khác với phát hiện đối tượng, YOLO đạt được kết quả hàng đầu, vượt xa các thuật toán phát hiện đối tượng thời gian thực khác.

Trong khi các thuật toán như Faster RCNN hoạt động bằng cách phát hiện các vùng quan tâm có thể có bằng Mạng Đề xuất Vùng và sau đó thực hiện nhận dạng trên các vùng đó riêng biệt, YOLO thực hiện tất cả các dự đoán của nó với sự trợ giúp của một lớp kết nối đầy đủ duy nhất.

Các phương pháp sử dụng Mạng Đề xuất Vùng thực hiện nhiều lần lặp lại cho cùng một hình ảnh, trong khi YOLO chỉ cần một lần lặp.

Kể từ khi phát hành phiên bản YOLO đầu tiên vào năm 2015, đã có nhiều phiên bản mới của cùng một mô hình được đề xuất, mỗi phiên bản đều xây dựng và cải thiện phiên bản trước đó. Dưới đây là một dòng thời gian trình bày sự phát triển của YOLO trong những năm gần đây.

2.4.2. Các phiên bản

2.4.2.1. YOLOv1

YOLO chia hình ảnh thành nhiều lưới và tính toán điểm tin cậy và các hộp bao cho mỗi ô lưới, phản ánh xác suất của một đối tượng nằm trong ô lưới đó. Tiếp theo, dựa trên xác suất của đối tượng lớn hơn không, thuật toán tính toán các xác suất lớp tương ứng và nhân chúng với xác suất đối tượng để tạo ra một điểm xác suất tổng thể và hộp bao. Với kiến trúc này, YOLOv1 đã vượt qua R-CNN với độ chính xác trung bình (mean average precision - mAP) là 63.4 và tốc độ suy luận (inference speed) là 45 khung hình mỗi giây (FPS) trên bộ dữ liệu Pascal Visual Object Classes 2007 mã nguồn mở. [11]

Các điểm nổi bật của YOLOv1:

- Phân chia hình ảnh thành lưới: Hình ảnh được chia thành nhiều ô lưới để phân tích.
- Tính toán điểm tin cậy và hộp bao: Mỗi ô lưới sẽ tính toán điểm tin cậy và hộp bao cho các đối tượng có thể nằm trong ô đó.
- Tích hợp xác suất lớp: Nếu xác suất của đối tượng lớn hơn không, thuật toán sẽ tính toán các xác suất lớp tương ứng và nhân chúng với xác suất đối tượng.

- Hiệu suất vượt trội: YOLOv1 đạt mAP 63.4 và tốc độ suy luận 45 FPS, vượt trội hơn so với R-CNN trên bộ dữ liệu Pascal VOC 2007. [11]

2.4.2.2. YOLOv2

Năm 2016, Joseph Redmon và Ali Farhadi đã giới thiệu YOLOv2, có khả năng phát hiện hơn 9000 loại đối tượng. YOLOv2 mang đến một số cải tiến quan trọng, trong đó có việc giới thiệu anchor boxes – các hộp bao định sẵn được gọi là priors mà mô hình sử dụng để xác định vị trí lý tưởng của một đối tượng. Thuật toán tính toán điểm IoU cho hộp bao dự đoán so với anchor box. Nếu IoU đạt đến ngưỡng nhất định, mô hình sẽ tạo ra dự đoán.

Các điểm nổi bật của YOLOv2:

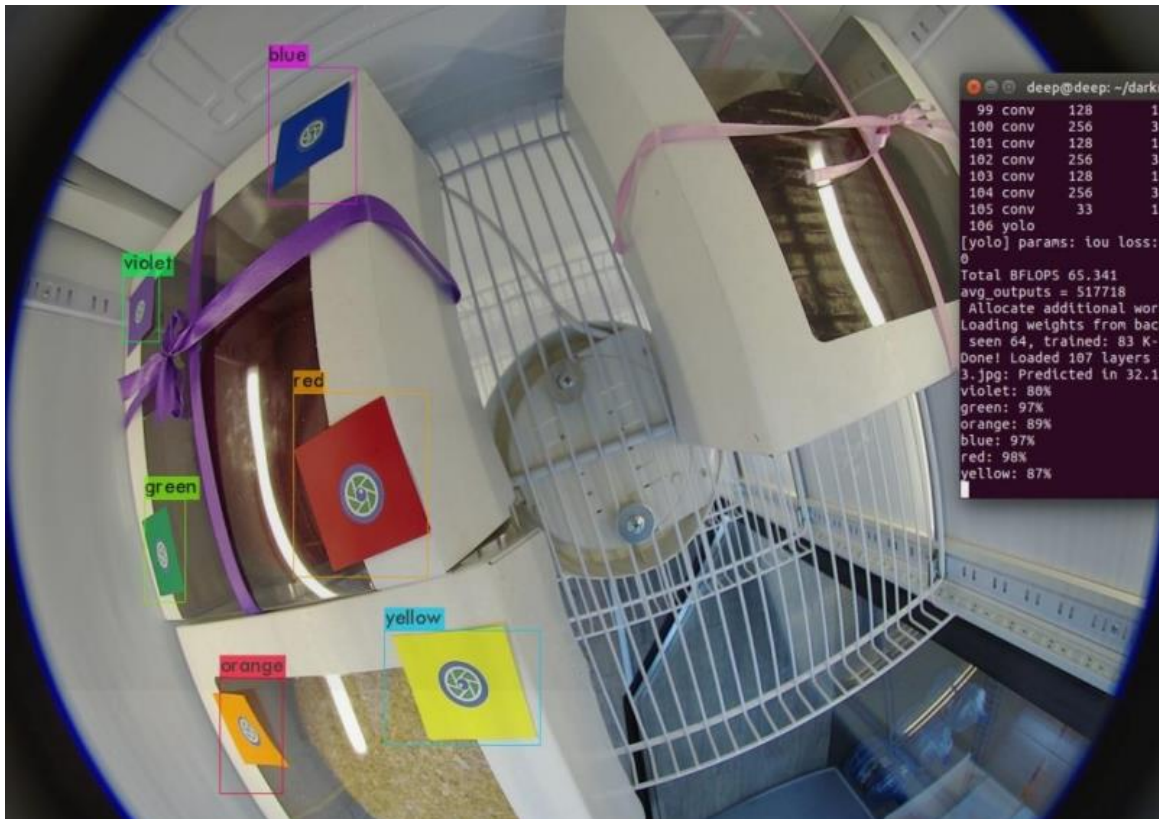
- Anchor Boxes: Sử dụng các hộp bao định sẵn (priors) để cải thiện độ chính xác trong việc xác định vị trí đối tượng. Điều này giúp mô hình dễ dàng định vị các đối tượng có hình dạng và kích thước khác nhau.
- Intersection over Union (IoU): Thuật toán tính toán điểm IoU giữa hộp bao dự đoán và anchor box. Nếu điểm IoU vượt qua ngưỡng xác định, mô hình sẽ tạo ra dự đoán cuối cùng cho vị trí và loại đối tượng.
- Hiệu suất vượt trội: YOLOv2 đạt được độ chính xác trung bình (mean Average Precision - mAP) là 76.8 và tốc độ suy luận 67 khung hình mỗi giây (FPS) trên bộ dữ liệu VOC 2007.

Nhờ các cải tiến này, YOLOv2 đã nâng cao khả năng phát hiện đối tượng, không chỉ mở rộng số lượng loại đối tượng có thể nhận diện mà còn cải thiện tốc độ và độ chính xác của quá trình phát hiện. Điều này giúp YOLOv2 trở thành một công cụ mạnh mẽ trong các ứng dụng thị giác máy tính yêu cầu phát hiện đối tượng nhanh chóng và chính xác. [11]

2.4.2.3. YOLOv3

Năm 2018, Joseph Redmon và Ali Farhadi đã công bố một bài báo giới thiệu YOLOv3, phiên bản mới với độ chính xác cao hơn các phiên bản trước đó.

YOLOv3 đạt được độ chính xác trung bình (mAP) là 28.2 với thời gian xử lý chỉ 22 mili giây. [11]



Hình 2.11: Ứng dụng YOLOv3 cho tủ lạnh thông minh trong âm thực và nhà hàng.

Darknet-53 Backbone:

YOLOv3 sử dụng Darknet-53 làm kiến trúc xương sống (backbone) cho mô hình. Darknet-53 là một mạng nơ-ron tích chập sâu với 53 lớp, được thiết kế để cân bằng giữa độ chính xác và hiệu suất tính toán.

Logistic Classifiers:

Để dự đoán các lớp, YOLOv3 sử dụng bộ phân loại logistic thay vì softmax. Việc sử dụng logistic classifiers giúp mô hình có thể dự đoán nhiều nhãn cho một đối tượng, phù hợp với các tình huống mà một đối tượng có thể thuộc nhiều loại cùng một lúc.

Binary Cross-entropy Loss:

YOLOv3 áp dụng hàm mất mát Binary Cross-entropy cho việc huấn luyện, thay vì sử dụng các hàm mất mát truyền thống. Binary Cross-entropy loss giúp cải thiện quá trình tối ưu hóa và độ chính xác của mô hình.

Hiệu suất:

Với những cải tiến trên, YOLOv3 đạt mAP 28.2 và có khả năng xử lý với tốc độ 22 mili giây cho mỗi hình ảnh, tương đương với khoảng 45 khung hình mỗi giây (FPS). Điều này cho thấy YOLOv3 không chỉ tăng cường độ chính xác mà còn duy trì tốc độ xử lý nhanh chóng.

Nhờ những cải tiến về kiến trúc và thuật toán, YOLOv3 trở thành một trong những mô hình phát hiện đối tượng hàng đầu, được sử dụng rộng rãi trong nhiều ứng dụng thị giác máy tính đòi hỏi độ chính xác và tốc độ cao. [11]

2.4.2.4. YOLOv4

Năm 2020, Alexey Bochkovskiy. phát hành YOLOv4, giới thiệu khái niệm Túi quà tặng (BoF) và Túi đặc biệt (BoS). BoF là một tập hợp các kỹ thuật tăng cường dữ liệu giúp tăng độ chính xác mà không mất thêm chi phí suy luận. (BoS tăng cường đáng kể độ chính xác với chi phí tăng nhẹ). Mô hình đạt được 43,5 mAP ở tốc độ 65 FPS trên bộ dữ liệu COCO. [11]

YOLOv4 sử dụng một số đặc trưng sáng tạo hoạt động cùng nhau để tối ưu hóa hiệu suất của nó. Chúng bao gồm Weighted-Residual-Connections, Cross-Stage-Partial-connections, Cross-mini-Batch Normalization, Self-adversarial-training, Mish-activation, Mosaic data augmentation, DropBlock regularization, và CIoU loss. Các đặc trưng này được kết hợp để đạt được kết quả hiện đại. [11]

Một máy dò đối tượng điển hình bao gồm một số bộ phận bao gồm đầu vào, xương sống, cổ và đầu. Xương sống của YOLOv4 được đào tạo trước trên ImageNet và được sử dụng để dự đoán các lớp và hộp giới hạn của các đối tượng. Xương sống có thể là từ một số mô hình bao gồm VGG, ResNet, ResNeXt hoặc DenseNet. Phần cổ của máy dò được sử dụng để thu thập bản đồ địa vật từ các giai

đoạn khác nhau và thường bao gồm một số đường dẫn từ dưới lên và một số đường dẫn từ trên xuống. Phần đầu là những gì được sử dụng để thực hiện phát hiện và phân loại đối tượng cuối cùng. [11]

2.4.2.5. YOLOv5

Không có tài liệu nghiên cứu chính thức, Ultralytics đã phát hành YOLOv5 cũng vào năm 2020. Mô hình này rất dễ đào tạo vì nó được triển khai trong PyTorch. Kiến trúc mô hình sử dụng khối kết nối một phần giai đoạn chéo làm xương sống cho luồng gradient tốt hơn để giảm chi phí tính toán. YOLOv5 sử dụng tệp YAML thay vì tệp cấu hình trong cấu hình mô hình.

Các đặc trưng chính:

- **Chia tách không neo Ultralytics Head:** Các mô hình phát hiện đối tượng truyền thống dựa vào các hộp neo được xác định trước để dự đoán vị trí đối tượng. Tuy nhiên, YOLOv5u hiện đại hóa cách tiếp cận này. Bằng cách áp dụng phân chia không neo Ultralytics Đầu tiên, nó đảm bảo một cơ chế phát hiện linh hoạt và thích ứng hơn, do đó nâng cao hiệu suất trong các tình huống đa dạng.
- **Tối ưu hóa sự đánh đổi độ chính xác-tốc độ:** Tốc độ và độ chính xác thường kéo theo hướng ngược nhau. Nhưng YOLOv5u thách thức sự đánh đổi này. Nó cung cấp một sự cân bằng hiệu chuẩn, đảm bảo phát hiện thời gian thực mà không ảnh hưởng đến độ chính xác. Đặc trưng này đặc biệt vô giá đối với các ứng dụng yêu cầu phản hồi nhanh, chẳng hạn như xe tự hành, robot và phân tích video thời gian thực.
- **Nhiều mô hình Pre-training:** Hiểu rằng các nhiệm vụ khác nhau đòi hỏi các bộ công cụ khác nhau, YOLOv5u cung cấp rất nhiều mô hình được đào tạo trước. Cho dù bạn đang tập trung vào Suy luận, Xác thực hay Đào tạo, có một mô hình được thiết kế riêng đang chờ bạn. Sự đa dạng này đảm bảo bạn không chỉ sử dụng giải pháp một kích thước phù hợp với tất cả, mà còn là một mô hình được tinh chỉnh đặc biệt cho thử thách độc đáo của bạn. [11]



Hình 2.12: Phát hiện đối tượng nhỏ với YOLOv5 trong phân tích giao thông

2.4.2.6. YOLOv6

Là một phiên bản không chính thức khác được giới thiệu vào năm 2022 bởi Meituan – một nền tảng mua sắm của Trung Quốc. Công ty nhắm mục tiêu mô hình cho các ứng dụng công nghiệp với hiệu suất tốt hơn so với người tiền nhiệm của nó. Những thay đổi dẫn đến việc YOLOv6n đạt được mAP là 37.5 ở 1187 FPS trên tập dữ liệu COCO và YOLOv6s đã đạt được 45 mAP ở 484 FPS.

Các đặc trưng chính:

- **Mô-đun nội hai chiều:** YOLOv6 giới thiệu một mô-đun ở cổ của máy dò, tăng cường tín hiệu nội địa hóa và mang lại hiệu suất tăng với sự suy giảm tốc độ không đáng kể.
- **Chiến lược đào tạo có sự hỗ trợ của neo (Anchor-Assisted Training):** Mô hình này đề xuất Anchor-Assisted Training để tận hưởng những lợi ích của cả mô hình dựa trên neo và không có neo mà không ảnh hưởng đến hiệu quả suy luận.
- **Thiết kế xương sống và cổ nâng cao:** Bằng cách đào sâu YOLOv6 để bao gồm một giai đoạn khác ở xương sống và cổ, mô hình này đạt được hiệu suất hiện đại trên bộ dữ liệu COCO ở đầu vào có độ phân giải cao.

- **Chiến lược tự chưng cất:** Một chiến lược tự chưng cất mới được thực hiện để tăng hiệu suất của các mô hình YOLOv6 nhỏ hơn, tăng cường nhánh hồi quy phụ trợ trong quá trình đào tạo và loại bỏ nó theo suy luận để tránh sự suy giảm tốc độ rõ rệt. [11]

2.4.2.7. YOLOv7

Vào tháng 7/2022, một nhóm các nhà nghiên cứu đã công bố mô hình mã nguồn mở YOLOv7, máy dò đối tượng nhanh nhất và chính xác nhất với mAP là 56,8% ở FPS dao động từ 5 đến 160. YOLOv7 dựa trên Mạng tổng hợp lớp hiệu quả mở rộng, giúp cải thiện đào tạo bằng cách cho phép mô hình học các đặc trưng đa dạng với tính toán hiệu quả.



Hình 2.13: Ứng dụng hệ thống AI được đào tạo để phát hiện máy bay với YOLOv7

Các đặc trưng chính:

- **Tham số hóa lại mô hình:** YOLOv7 đề xuất một mô hình tái tham số hóa theo kế hoạch, đây là một chiến lược áp dụng cho các lớp trong các mạng khác nhau với khái niệm đường dẫn lan truyền gradient.

- **Phân công nhân động:** Việc đào tạo mô hình với nhiều lớp đầu ra trình bày một vấn đề mới: "Làm thế nào để chỉ định các mục tiêu động cho đầu ra của các nhánh khác nhau?" Để giải quyết vấn đề này, YOLOv7 giới thiệu một phương pháp gán nhãn mới được gọi là gán nhãn có hướng dẫn từ thô đến mịn.
- **Mở rộng và chia tỷ lệ hợp chất:** YOLOv7 đề xuất các phương pháp "mở rộng" và "chia tỷ lệ hợp chất" cho máy dò đối tượng thời gian thực có thể sử dụng hiệu quả các tham số và tính toán.
- **Hiệu quả:** Phương pháp do YOLOv7 đề xuất có thể giảm hiệu quả khoảng 40% thông số và 50% tính toán của máy dò đối tượng thời gian thực hiện đại, đồng thời có tốc độ suy luận nhanh hơn và độ chính xác phát hiện cao hơn. [11]

2.4.2.8. YOLOv8

Là một mô hình tiên tiến, hiện đại, YOLO v8 Được xây dựng dựa trên sự thành công của các phiên bản trước, giới thiệu các đặc trưng và cải tiến mới để nâng cao hiệu suất, tính linh hoạt và hiệu quả. YOLO v8 hỗ trợ đầy đủ các tác vụ AI thị giác, bao gồm phát hiện, phân đoạn, ước tính tư thế, theo dõi và phân loại. Tính linh hoạt này cho phép người dùng tận dụng YOLOv8. Khả năng của họ trên các ứng dụng và lĩnh vực đa dạng. [10]

Kiến trúc

YOLOv8 được thiết kế để khắc phục những hạn chế của các phiên bản trước đồng thời duy trì sự cân bằng giữa tốc độ và độ chính xác. Một cải tiến đáng chú ý của YOLOv8 là thiết kế mô-đun và có thể mở rộng, cho phép mô hình linh hoạt thích ứng với nhiều yêu cầu khác nhau.

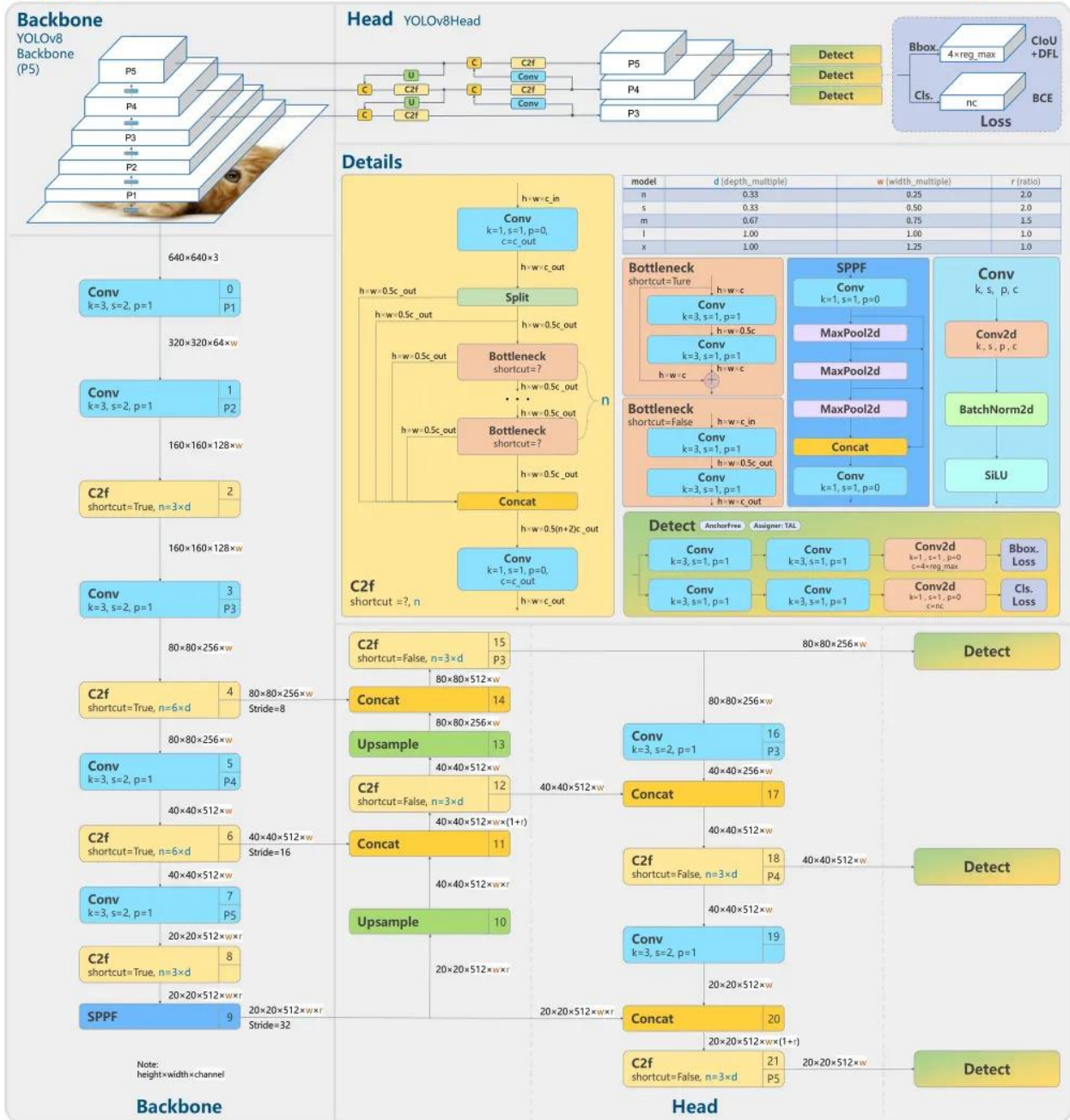
Cấu trúc của YOLOv8 được chia thành ba thành phần chính: xương sống, cổ và đầu. Xương sống chịu trách nhiệm trích xuất các đặc trưng từ hình ảnh đầu vào, với các tùy chọn như CSPDarknet53 và EfficientDet để phù hợp với nhu cầu cụ thể. Phần cổ kết nối xương sống với đầu, đóng vai trò quan trọng trong việc hợp nhất các đặc trưng. Phần đầu đảm nhận việc dự đoán các hộp giới hạn, lớp đối tượng và điểm tin cậy.

Một khía cạnh quan trọng khác của YOLOv8 là khả năng mở rộng mô hình. Nó cung cấp các biến thể khác nhau như YOLOv8-tiny và YOLOv8x, khác nhau về kích thước và độ phức tạp tính toán. Điều này cho phép người dùng chọn mô hình phù hợp với yêu cầu của mình, dù là trong môi trường hạn chế tài nguyên hay ứng dụng hiệu suất cao.

YOLOv8 cũng giới thiệu những cải tiến trong chiến lược đào tạo, bao gồm việc sử dụng tối ưu hóa Adam chỉnh lưu (RAdam) và phương pháp phát hiện đối tượng dựa trên neo hoặc không neo. Những cải tiến này giúp quá trình đào tạo hội tụ nhanh hơn và nâng cao hiệu suất trong các nhiệm vụ phát hiện đối tượng.

Hơn nữa, YOLOv8 có hệ thống cấu hình linh hoạt, cho phép người dùng dễ dàng tùy chỉnh các thông số như kích thước đầu vào, hộp neo và độ phức tạp của mô hình. Tính linh hoạt này làm cho YOLOv8 dễ dàng thích ứng với các bộ dữ liệu và kịch bản ứng dụng đa dạng.

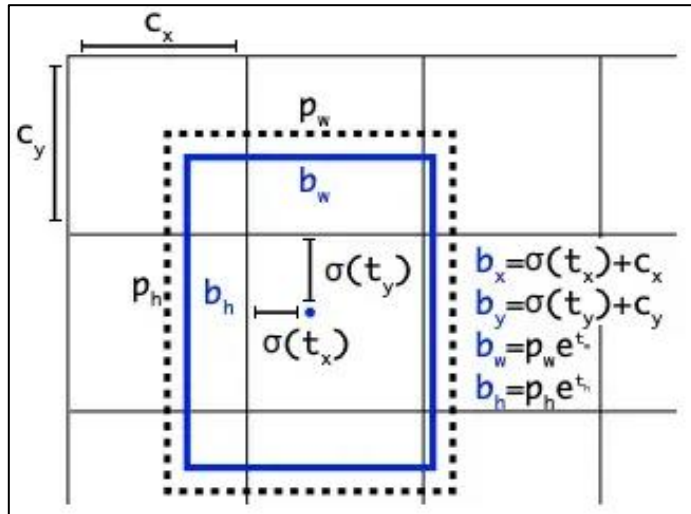
Tóm lại, kiến trúc của YOLOv8 nổi bật với thiết kế mô-đun, các biến thể có thể mở rộng, xương sống được cải thiện và các chiến lược đào tạo nâng cao. Những đặc điểm này cùng nhau góp phần vào sự thành công của YOLOv8 trong việc phát hiện đối tượng thời gian thực, khiến nó trở thành lựa chọn phổ biến cho các nhà nghiên cứu và chuyên gia trong lĩnh vực thị giác máy tính. [22]



Hình 2.14: Kiến trúc YOLOv8. [22]

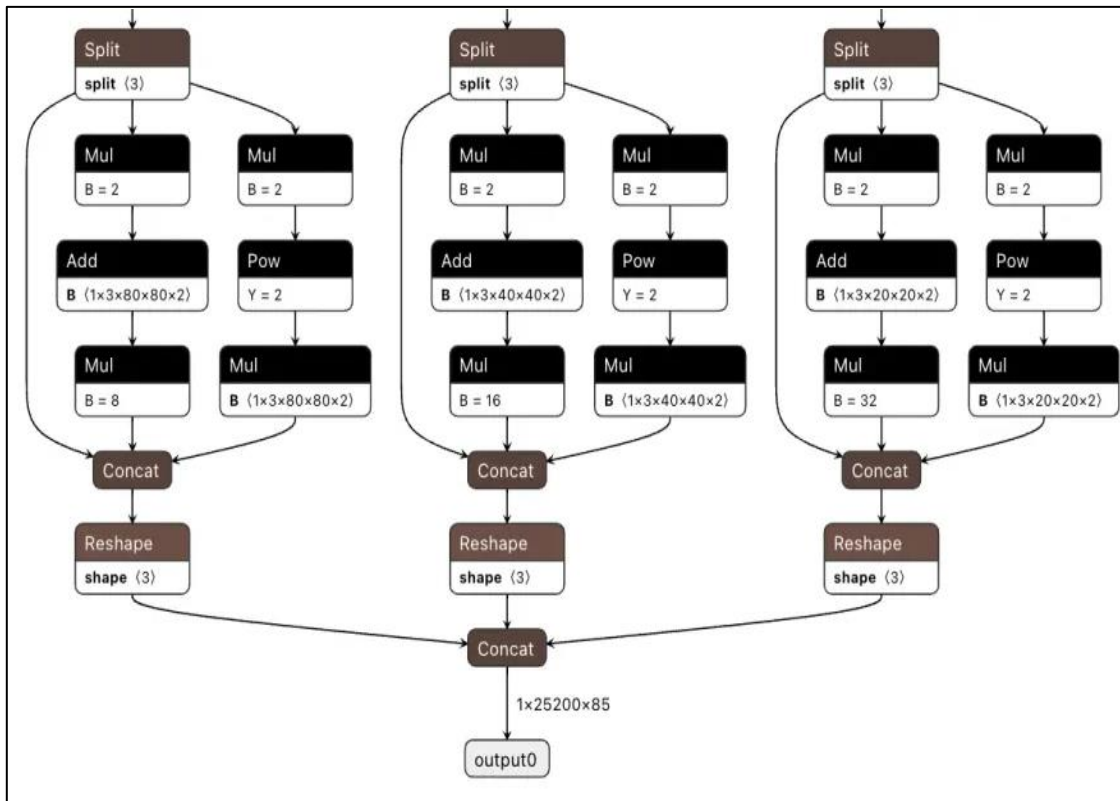
Phát hiện không neo

YOLOv8 không sử dụng anchor box, điều này có nghĩa là nó dự đoán trực tiếp tâm của một đối tượng thay vì dựa vào việc điều chỉnh từ một hộp neo đã được xác định trước.



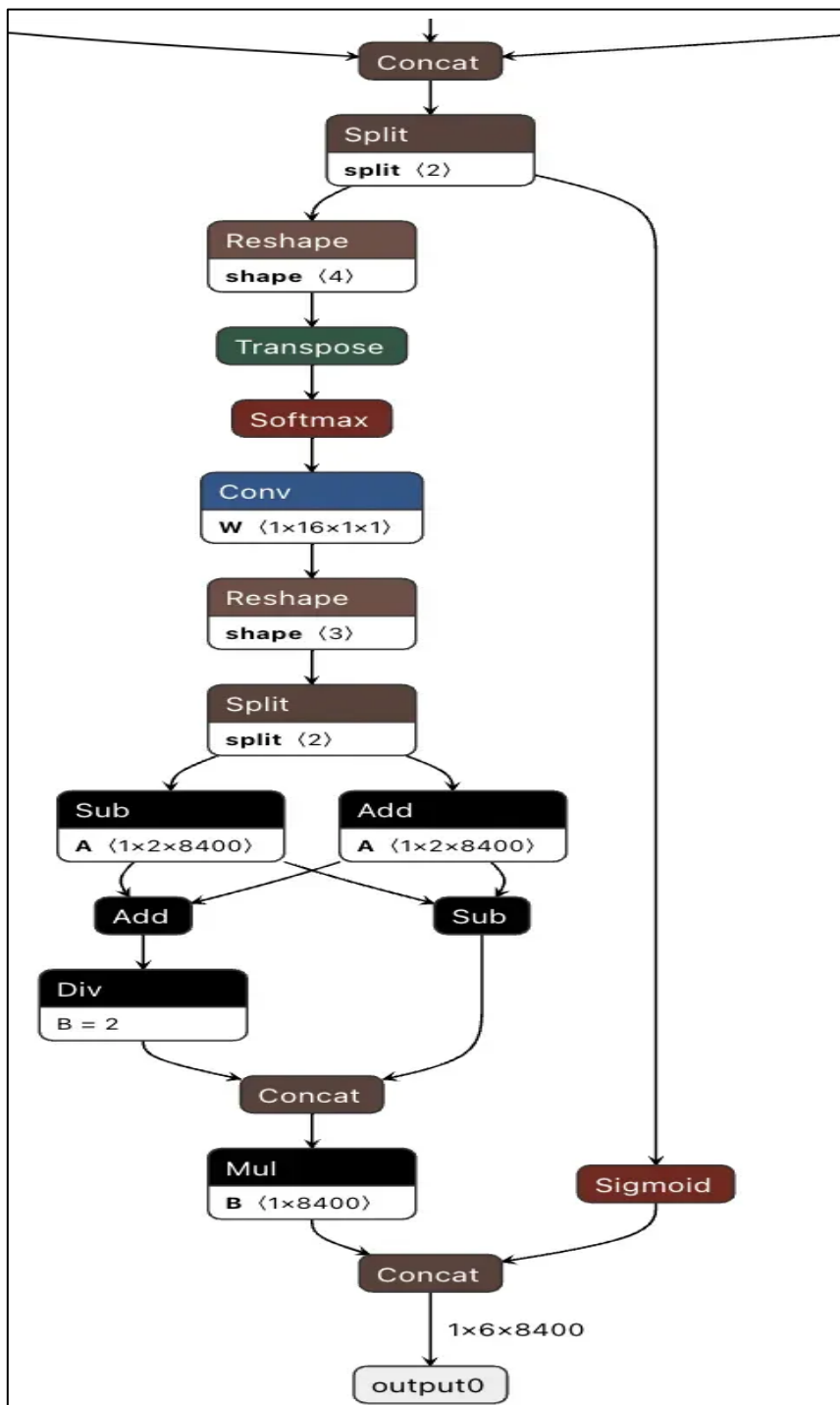
Hình 2.15: Hình dung anchor box trong YOLO

Trong các mô hình YOLO trước đó, anchor box đã trở nên phổ biến nhưng đồng thời cũng tạo ra sự phức tạp, bởi chúng có thể không phản ánh đúng sự đa dạng của các đối tượng trong tập dữ liệu.



Hình 2.16: detection head của YOLOv5

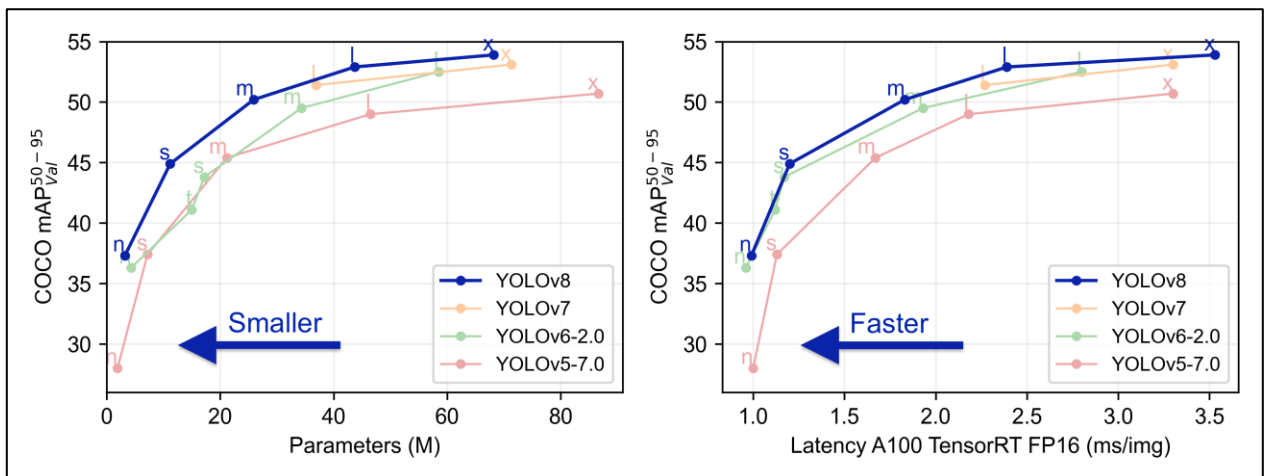
Sử dụng phát hiện không neo trong YOLOv8 giúp giảm số lượng dự đoán hộp, từ đó cải thiện tốc độ của quá trình Triệt tiêu không tối đa. Điều này giúp làm giảm phức tạp trong việc xử lý dữ liệu bằng cách lọc các phát hiện ứng cử viên sau khi suy luận.



Hình 2.17: detection head của YOLOv8

Các đặc trưng chính:

- Kiến trúc Backbone và Neck tiên tiến: YOLOv8 sử dụng các kiến trúc backbone và neck hiện đại, dẫn đến việc cải thiện khả năng trích xuất đặc trưng và hiệu suất phát hiện đối tượng.
- Đầu Ultralytics không neo: YOLOv8 áp dụng đầu Ultralytics không sử dụng anchor box, giúp tăng độ chính xác và cải thiện quá trình phát hiện so với các phương pháp sử dụng anchor box.
- Tối ưu hóa cân bằng độ chính xác và tốc độ: Với trọng tâm là duy trì sự cân bằng tối ưu giữa độ chính xác và tốc độ, YOLOv8 phù hợp cho các nhiệm vụ phát hiện đối tượng trong thời gian thực ở nhiều lĩnh vực ứng dụng khác nhau.
- Đa dạng các mô hình được huấn luyện trước: YOLOv8 cung cấp nhiều mô hình được huấn luyện trước để phục vụ cho các nhiệm vụ và yêu cầu hiệu suất khác nhau, giúp dễ dàng tìm ra mô hình phù hợp cho trường hợp sử dụng cụ thể.



Hình 2.18: Hiệu suất trên Tập dữ liệu Phát hiện Đối tượng MS COCO

CHƯƠNG 3: CÀI ĐẶT, THỬ NGHIỆM VÀ ĐÁNH GIÁ

3.1. Môi trường thử nghiệm

3.1.1. Ngôn ngữ lập trình Python

3.1.1.1. Khái niệm

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học, được dùng rộng rãi trong phát triển trí tuệ nhân tạo. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu. Vào tháng 7 năm 2018, Van Rossum đã từ chức lãnh đạo trong cộng đồng ngôn ngữ Python sau 30 năm làm việc.

Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động; do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.

Ban đầu, Python được phát triển để chạy trên nền Unix. Nhưng rồi theo thời gian, Python dần mở rộng sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido van Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.

Python luôn được xếp hạng vào những ngôn ngữ lập trình phổ biến nhất. [14]

3.1.1.2. Đặc tính

Python đang trở nên phổ biến trong cộng đồng lập trình nhờ có các đặc tính sau:

- Ngôn ngữ thông dịch: Python được xử lý trong thời gian chạy bởi trình thông dịch Python.

- Ngôn ngữ hướng đối tượng: Nó hỗ trợ các đặc trưng và kỹ thuật lập trình hướng đối tượng.
- Ngôn ngữ lập trình tương tác: Người dùng có thể tương tác trực tiếp với trình thông dịch python để viết chương trình.
- Ngôn ngữ dễ học: Python rất dễ học, đặc biệt là cho người mới bắt đầu.
- Cú pháp đơn giản: Việc hình thành cú pháp Python rất đơn giản và dễ hiểu, điều này cũng làm cho nó trở nên phổ biến.
- Mã nguồn mở: Python cho phép người dùng có thể sử dụng, chỉnh sửa và phân phối mã nguồn một cách tự do.
- Di động: Mã Python có thể chạy trên nhiều nền tảng phần cứng có cùng giao diện.
- Có thể mở rộng: Người dùng có thể thêm các mô-đun cấp thấp vào trình thông dịch Python.
- Có thể cải tiến: Python cung cấp một cấu trúc cải tiến để hỗ trợ các chương trình lớn sau đó là shell-script.
- Hỗ trợ cộng đồng: python có một cộng đồng lớn với nhiều người dùng và nhà phát triển trên khắp thế giới đóng góp vào việc phát triển và cải tiến Python. Cộng đồng này cũng cung cấp hỗ trợ và tài liệu để giúp người dùng python giải quyết các vấn đề một cách nhanh chóng và hiệu quả. [15]

3.1.1.3. Các phiên bản

Python đã được Guido van Rossum tạo ra vào những năm 1980 tại Trung tâm Toán học – Tin học (Centrum Wiskunde & Informatica, CWI) ở Hà Lan như là một ngôn ngữ kế tục ngôn ngữ ABC – một ngôn ngữ được lấy cảm hứng từ SETL (ngôn ngữ được phát triển bởi Jacob T. Schwartz và các đồng nghiệp), có khả năng xử lý ngoại lệ và giao tiếp với hệ điều hành Amoeba. Nó bắt đầu được triển khai vào tháng 12 năm 1989.

Python 2.0 được ra mắt vào ngày 16 tháng 10 năm 2000, với nhiều đặc trưng mới mẻ, bao gồm một bộ dọn rác phát hiện theo chu kỳ và khả năng hỗ trợ Unicode.

Python 3.0 được ra mắt vào ngày mùng 3 tháng 12 năm 2008. Đây là một phiên bản lớn của Python không tương thích ngược hoàn toàn. Nhiều đặc trưng lớn của nó đã được chuyển mã ngược (backport) về loạt phiên bản Python 2.6.x và 2.7.x. Các bản phát hành của Python 3 có đi kèm với công cụ 2to3, có tác dụng tự động hoá việc dịch mã Python 2 sang Python 3.

Python 3.9.2 và 3.8.8 được xúc tiến vì tất cả các phiên bản trước của Python (bao gồm cả 2.7) gặp một số vấn đề bảo mật, có thể dẫn đến thực thi mã từ xa và "đầu độc" bộ nhớ đệm.

Trong năm 2022, Python 3.10.4 và 3.9.12 được xúc tiến cùng với 3.8.13 và 3.7.13, nguyên nhân là do một vài vấn đề về bảo mật. Khi Python 3.9.13 được phát hành vào tháng Năm năm 2022, loạt phiên bản 3.9 (cùng với loạt 3.8 và 3.7) được thông báo rằng sẽ chỉ nhận được các bản vá bảo mật trong tương lai. Vào ngày 7 tháng Chín năm 2022, bốn bản cập nhật mới được phát hành do có khả năng xảy ra một cuộc tấn công từ chối dịch vụ: 3.10.7, 3.9.14, 3.8.14 và 3.7.14.

Tính đến tháng 10 năm 2023, Python 3.12 là bản phát hành ổn định mới nhất. Một số thay đổi đáng chú ý từ bản 3.11 bao gồm các thay đổi về ngôn ngữ và thư viện chuẩn.[\[14\]](#)

3.1.2. Môi trường phát triển tích hợp Pycharm

3.1.2.1. Khái niệm:

PyCharm là một môi trường phát triển tích hợp (IDE) được sử dụng để lập trình bằng Python. Nó cung cấp phân tích mã, trình gỡ lỗi đồ họa, trình kiểm tra đơn vị tích hợp, tích hợp với các hệ thống kiểm soát phiên bản và hỗ trợ phát triển web với Django. PyCharm được phát triển bởi công ty JetBrains của Séc.

Nó là đa nền tảng, hoạt động trên Microsoft Windows, macOS và Linux. PyCharm có Professional Edition, được phát hành theo giấy phép độc quyền và Community Edition được phát hành theo Giấy phép Apache. PyCharm Community Edition ít mở rộng hơn Professional Edition. [\[12\]](#)

3.1.2.2. Các đặc trưng:

- Hỗ trợ và phân tích mã hóa, với việc hoàn thành mã, tô sáng cú pháp và lỗi, tích hợp lớp lót và sửa lỗi nhanh
- Điều hướng dự án và mã: chế độ xem dự án chuyên biệt, chế độ xem cấu trúc tệp và nhảy nhanh giữa các tệp, lớp, phương thức và cách sử dụng
- Tái cấu trúc mã Python: bao gồm đổi tên, phương thức trích xuất, giới thiệu biến, giới thiệu hằng số, kéo lên, đẩy xuống và các biến khác
- Hỗ trợ cho các khung web: Django, web2py và Flask
- Trình gỡ lỗi Python tích hợp
- Kiểm tra đơn vị tích hợp, với phạm vi bao phủ từng dòng
- Công cụ ứng dụng Google Phát triển Python
- Tích hợp kiểm soát phiên bản: giao diện người dùng hợp nhất cho Mercurial, Git, Subversion, Perforce và CVS với danh sách thay đổi và hợp nhất
- Tích hợp công cụ khoa học: tích hợp với IPython Notebook, có bảng điều khiển Python tương tác và hỗ trợ Anaconda cũng như nhiều gói khoa học bao gồm Matplotlib và NumPy. [12]

3.1.2.3. Các phiên bản

PyCharm đã được phát hành ra thị trường các IDE tập trung vào Python để cạnh tranh với PyDev (cho Eclipse) hoặc Komodo IDE tập trung rộng rãi hơn của ActiveState.

Phiên bản beta của sản phẩm được phát hành vào tháng 7 năm 2010, với phiên bản 1.0 đến 3 tháng sau đó. Phiên bản 2.0 được phát hành vào ngày 13 tháng 12 năm 2011, phiên bản 3.0 được phát hành vào ngày 24 tháng 9 năm 2013 và phiên bản 4.0 được phát hành vào ngày 19 tháng 11 năm 2014.

PyCharm trở thành mã nguồn mở vào ngày 22 tháng 10 năm 2013. Biến thể Mã nguồn mở được phát hành dưới tên Community Edition – trong khi biến thể thương mại, Professional Edition, chứa các mô-đun nguồn đóng. [12]

3.1.3. Công cụ trực tuyến Google Collab

3.1.3.1. Khái niệm

Colaboratory hay còn gọi là Google Colab, là một sản phẩm từ Google Research, nó cho phép chạy các dòng code python thông qua trình duyệt, đặc biệt phù hợp với Data analysis, machine learning và giáo dục. Colab không cần yêu cầu cài đặt hay cấu hình máy tính, mọi thứ có thể chạy thông qua trình duyệt, bạn có thể sử dụng tài nguyên máy tính từ CPU tốc độ cao và cả GPUs và cả TPUs đều được cung cấp cho bạn.

Colab cung cấp nhiều loại GPU, thường là Nvidia K80s, T4s, P4s and P100s, tuy nhiên người dùng không thể chọn loại GPU trong Colab, GPU trong Colab thay đổi theo thời gian. Vì là dịch vụ miễn phí, nên Colab sẽ có những thứ tự ưu tiên trong việc sử dụng tài nguyên hệ thống, cũng như giới hạn thời gian sử dụng, thời gian sử dụng tối đa lên tới 12 giờ. [13]

3.1.3.2. Những đặc trưng chính trong Google Colab

- Sử dụng Jupyter Notebooks trực tuyến

Google Colab cho phép tạo và chạy các Jupyter Notebooks trực tuyến mà không cần cài đặt môi trường phát triển phức tạp trên máy tính cá nhân.

Giao diện sử dụng tương tự như Jupyter Notebook truyền thống với các cell cho phép thực thi mã Python hoặc viết markdown để tạo nội dung hướng dẫn. [16]

- Khả năng chia sẻ và cộng tác

Người dùng có thể chia sẻ notebook với những người khác để cùng làm việc trên cùng một notebook, tạo điều kiện thuận lợi cho việc học tập và làm việc nhóm.

Các đặc trưng như bình luận và chế độ chỉnh sửa đồng thời giúp tăng tính tương tác và hiệu quả của quá trình cộng tác. [16]

- **Dùng GPU và TPU miễn phí**

Google Colab cung cấp truy cập miễn phí đến GPU và TPU, đặc biệt hữu ích cho các tác vụ tính toán nặng về mặt số học, đặc biệt là trong lĩnh vực học máy và học sâu.

Việc sử dụng các card GPU hoặc TPU có sẵn giúp tăng tốc độ xử lý và huấn luyện mô hình so với việc sử dụng CPU thông thường. [16]

- **Lưu trữ dữ liệu trên Google Drive và tích hợp Google Cloud**

Người dùng có thể truy cập và lưu trữ dữ liệu trực tiếp từ Google Drive, tạo điều kiện thuận lợi cho việc làm việc với tập tin dữ liệu lớn.

Tích hợp với điện toán đám mây của Google cũng cho phép sử dụng các dịch vụ như BigQuery, Cloud Storage, và các API khác từ dịch vụ điện toán đám mây trong quá trình làm việc. [16]

3.2. Tạo bộ dữ liệu học để huấn luyện

Roboflow là một framework dành cho nhà phát triển Thị giác Máy tính để thu thập dữ liệu tốt hơn để xử lý trước và các kỹ thuật đào tạo mô hình. Roboflow có sẵn các tập dữ liệu công khai cho người dùng và cũng có quyền truy cập để người dùng tải lên dữ liệu tùy chỉnh của riêng họ. Roboflow chấp nhận các định dạng chú thích khác nhau. Trong quá trình xử lý trước dữ liệu, có các bước liên quan như định hướng hình ảnh, thay đổi kích thước, độ tương phản và tăng cường dữ liệu.

Toàn bộ quy trình làm việc có thể được điều phối với các nhóm trong khuôn khổ. Đối với đào tạo mô hình, có một loạt thư viện mô hình đã có mặt như EfficientNet, MobileNet, Yolo, TensorFlow, PyTorch, v.v. Sau đó, các tùy chọn trực quan và triển khai mô hình cũng có sẵn do đó bao gồm toàn bộ hiện đại.

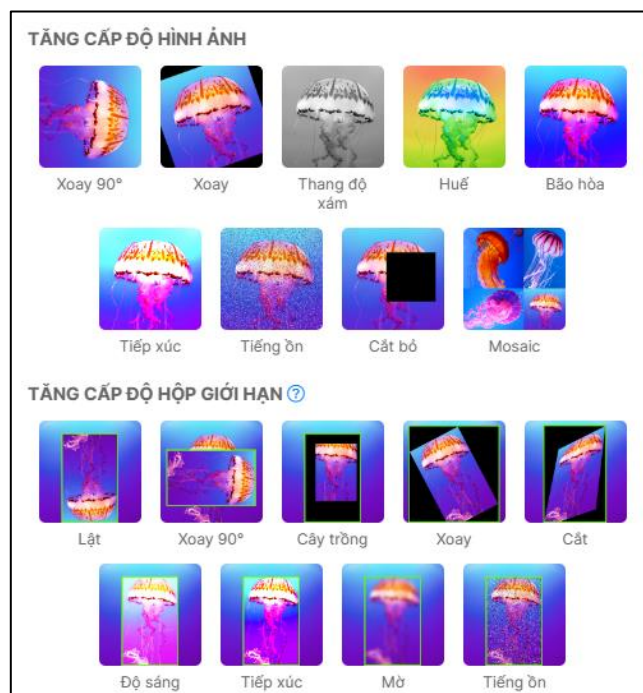
Roboflow được sử dụng trong các ngành công nghiệp thị giác máy tính khác nhau cho các trường hợp sử dụng như – phát hiện rò rỉ khí đốt, phát hiện thực vật và cỏ dại, bảo trì máy bay, ước tính thiệt hại mái nhà, hình ảnh vệ tinh, ô tô tự lái, bộ đếm giao thông, dọn rác và nhiều hơn nữa. [17]

Tổng số lượng ảnh : 12.076 ảnh

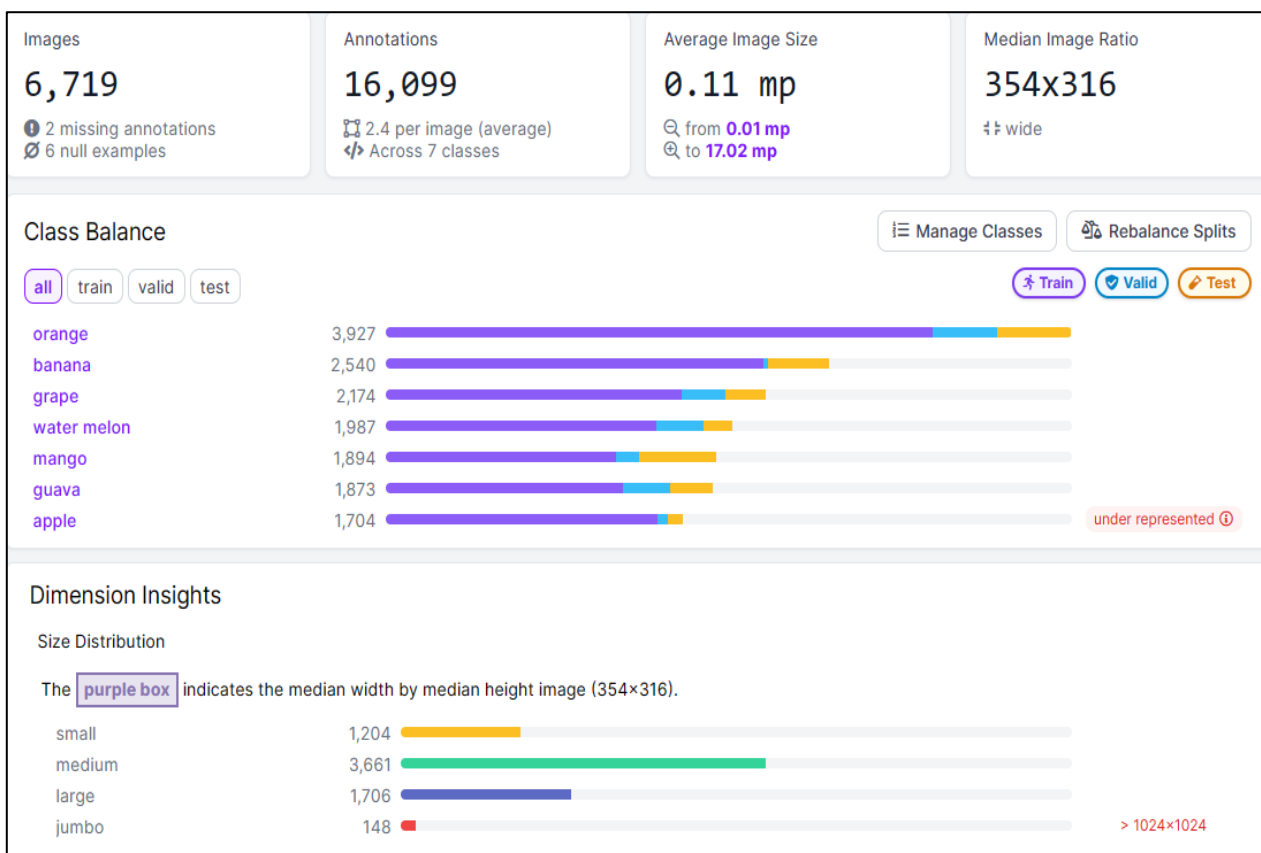
Trái Cây	Số Ảnh
Cam	1087
chuối	985
Nho	870
Dưa Hấu	878
Xoài	935
Ổi	810
Táo	1189

Bảng 3-1: Số Lượng Ảnh của từng loại

Với tổng số 12.076 ảnh trong đó 6.719 ảnh được thu thập thủ công trên các trang mạng xã hội: facebook, pinterest, pixel, ... và một số ảnh được thu thập từ việc chụp trực tiếp trái cây bên ngoài, với 3995 ảnh còn lại được thêm vào thông qua đặc trưng tiền xử lý hay tăng cường hình ảnh trên Roboflow.



Hình 3.1: đặc trưng tăng cường hình ảnh trên Roboflow



Hình 3.2: Health check được thông kê trên Roboflow.

Phân chia tập dữ liệu:

Train (Training Set): Đây là tập dữ liệu được sử dụng để huấn luyện mô hình. Mô hình học máy sẽ “học” từ dữ liệu này, điều chỉnh các tham số để có thể dự đoán hoặc phân loại chính xác nhất. Được chia nhiều nhất với 10.714 ảnh chiếm 89%.

Test (Testing Set): Sau khi mô hình đã được huấn luyện, tập dữ liệu này được sử dụng để kiểm tra hiệu suất của mô hình. Nó giúp đánh giá xem mô hình có khả năng tổng quát hóa trên dữ liệu mới mà nó chưa từng “thấy” trong quá trình huấn luyện hay không. Được chia 655 ảnh chiếm 6%.

Valid (Validation Set): Đây là tập dữ liệu được sử dụng để điều chỉnh các hyperparameters của mô hình. Nó giúp xác định cấu hình mô hình tốt nhất trước khi thực hiện kiểm tra cuối cùng trên tập testing set. Được chia 707 ảnh chiếm 6%.

3.3. Đào tạo mô hình phát hiện loại trái cây

Để huấn luyện mô hình “ Phát hiện loại trái cây “ bằng phương pháp học sâu thì em sử dụng một công cụ hỗ trợ ở đây là Google Colab. Google Colab là một dịch vụ máy tính đám mây của Google cho phép người dùng tạo ra các tệp notebook Jupyter để thực thi mã Python. Nó cung cấp một môi trường tính toán đám mây miễn phí, với các đặc trưng như GPU miễn phí, RAM miễn phí đến 12GB có thể mở rộng đến 25.5GB nếu có trả phí, lưu trữ đám mây và nhiều đặc trưng khác.

Ngoài ra Google Colab còn liên kết với driver để có thể hỗ trợ người dùng trong việc lưu trữ các dữ liệu quan trọng. Google Colab được thiết kế để hỗ trợ việc phát triển và huấn luyện các mô hình học máy và các ứng dụng AI. Nó cung cấp một môi trường tính toán đám mây miễn phí, mạnh mẽ và linh hoạt cho các nhu cầu học tập và nghiên cứu.

```
%pip install ultralytics
import ultralytics
ultralytics.checks()

Ultralytics YOLOv8.2.2 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Setup complete (2 CPUs, 12.7 GB RAM, 28.8/78.2 GB disk)

!pip install Pillow

Requirement already satisfied: Pillow in /usr/local/lib/python3.10/dist-packages (9.4.0)

from PIL import Image

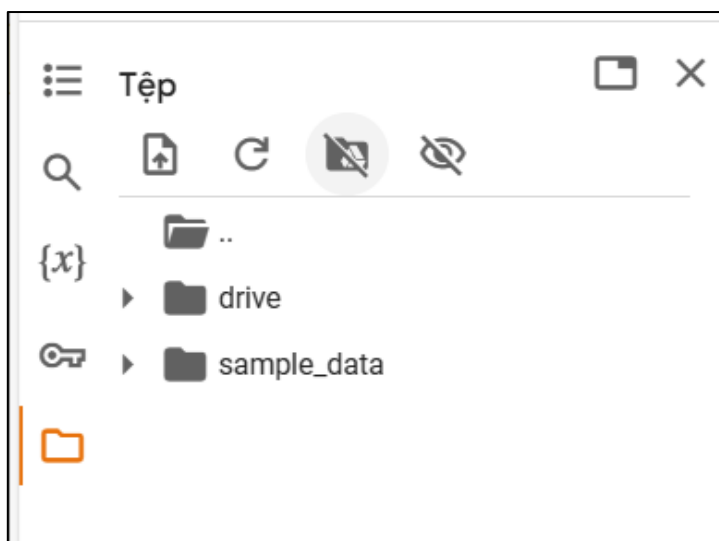
from IPython.display import Image

!apt install unzip
```

Hình 3.3: cài các thư viện hỗ trợ

Vì Google Colab bản miễn phí chỉ cho giới hạn về thời gian chạy nên ta có thể liên kết với drive để lưu trữ để tránh mất dữ liệu sau khi hết phiên

Ta có thể dễ dàng kết nối với drive với 2 cách:



Hình 3.4: Ẩn chọn biểu tượng drive trong phần tệp

```
from google.colab import drive
drive.mount('/content/drive')
```

Hình 3.5: Đoạn mã kết nối Google Colab với Google Drive

Sau khi cài xong một vài đoạn mã ta có thể bắt đầu đào tạo với đoạn mã sau

```
%cd {HOME}
yolo task=detect mode=train model=yolov8n.pt data={dataset.location}/data.yaml
epochs=100 imgsz=640
```

Hình 3.6: Đoạn mã để bắt đầu đào tạo

Trong đó:

- **%cd:** là di chuyển đến tệp ta sẽ lưu trữ các dữ liệu khi đang đào tạo
- **Epochs:** Số lần ảnh được đào tạo (số càng lớn dữ liệu đào tạo sẽ được cải thiện nhưng đổi lại thì sẽ tốn thời gian hơn)
- **Imgsz:** kích thước ảnh (đối với YOLOv8 nhà phát triển có yêu cầu kích thước là 640x640).
- **Model:** mô hình được Ultralytics cung cấp để đào tạo.

- **Data:** đường dẫn đến tệp cấu hình tập dữ liệu (ví dụ: coco8.yaml). Tệp này chứa các tham số dành riêng cho tập dữ liệu, bao gồm đường dẫn đến dữ liệu đào tạo và xác thực, tên lớp và số lượng lớp.

Ta cũng có một số thông số khác nhưng ta cũng không cần phải ghi hết tất cả để có thể chạy được mọi thông tin ta có thể tìm trên trang train của ultralytics. Ultralytics có đầy đủ các thông số nếu ai có yêu cầu chi tiết trong quá trình đào tạo.

Thời gian đào tạo với 20/100 epochs mất tầm 2 tiếng 15 phút

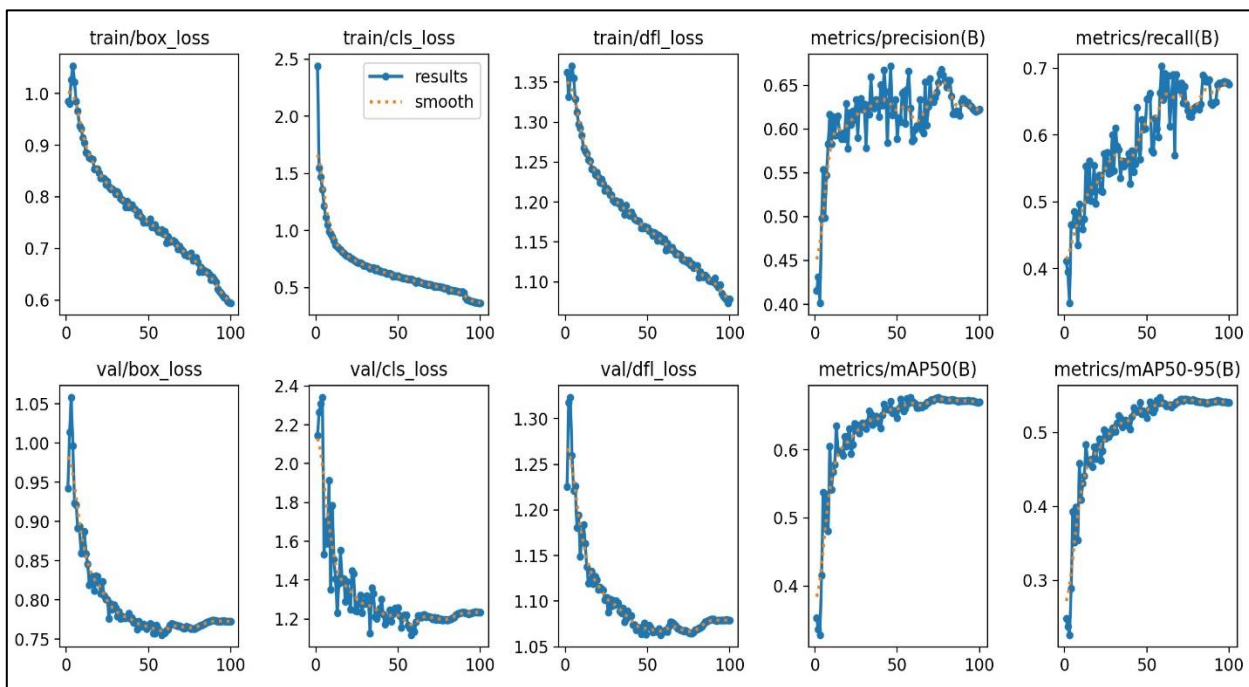
Với từng epochs ta có bảng sau:

Epochs	ảnh kiểm thử (valid)	Thời gian thực hiện	P	R	mAP	mAP 50- 95
80	707	43 phút 13 giây	0.725	0.74	0.773	0.642
90	707	1 tiếng 24 phút 46 giây	0.735	0.749	0.772	0.642
100	707	2 tiếng 6 phút 41 giây	0.722	0.776	0.77	0.641

Bảng 3-2: Bảng thử nghiệm với các epochs khác nhau

Quan sát (bảng 3-2) trên ta có thể nhận thấy kết quả thử nghiệm mô hình với các giá trị của epochs khác nhau nhưng không có nhiều sự chênh lệch. Tuy nhiên với 90 epochs ta được độ chính xác là 0.772 trong thời gian 1 tiếng 24 phút khi chạy 20/100 epochs. Khi thay đổi các số epochs ta có thể thấy từ 80 có sự tăng nhẹ trong khi đó lên 100 một vài chỉ số đã bị giảm đi. Từ đó ta có thể kết luận với dữ liệu tầm 12 nghìn ảnh đối với mô hình này thì 90 epochs là sự lựa chọn tốt nhất.

3.3.1. Giá trị các biểu đồ

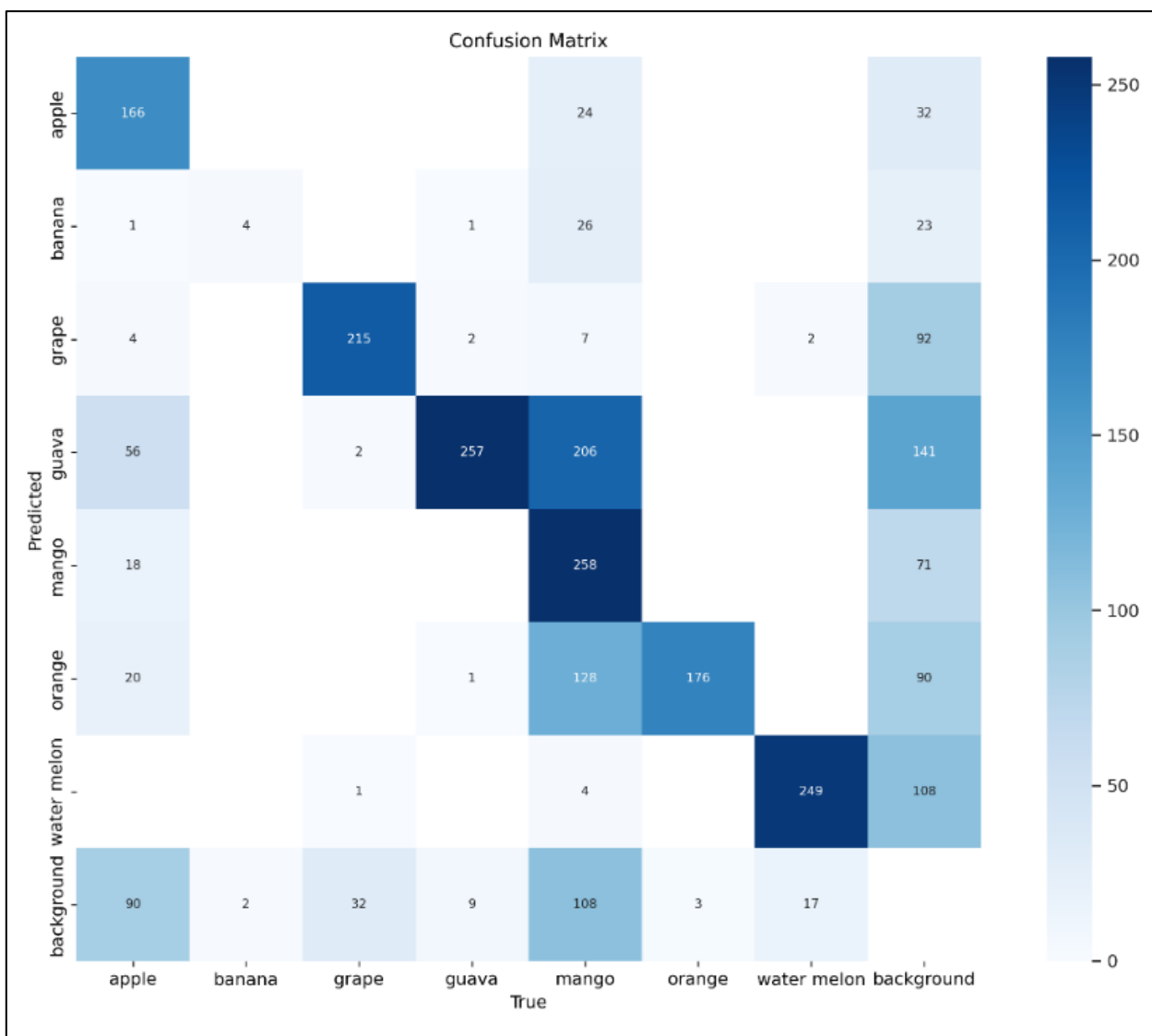


Hình 3.7: Biểu đồ thống kê sau khi đào tạo

Theo như hình ta có thể thấy:

Các biểu đồ mất mát đang giảm dần cho biết được mô hình đang được học tốt hơn.

Biểu đồ độ chính xác: thì lại tăng dần lên, và nhìn từ hình ảnh ta thấy được mô hình có sự tăng trưởng nhanh ở 50 epochs đầu tiên và chậm lại.



Hình 3.8: ma trận nhầm lẫn (hiệu suất dự đoán của mô hình với epochs=100)

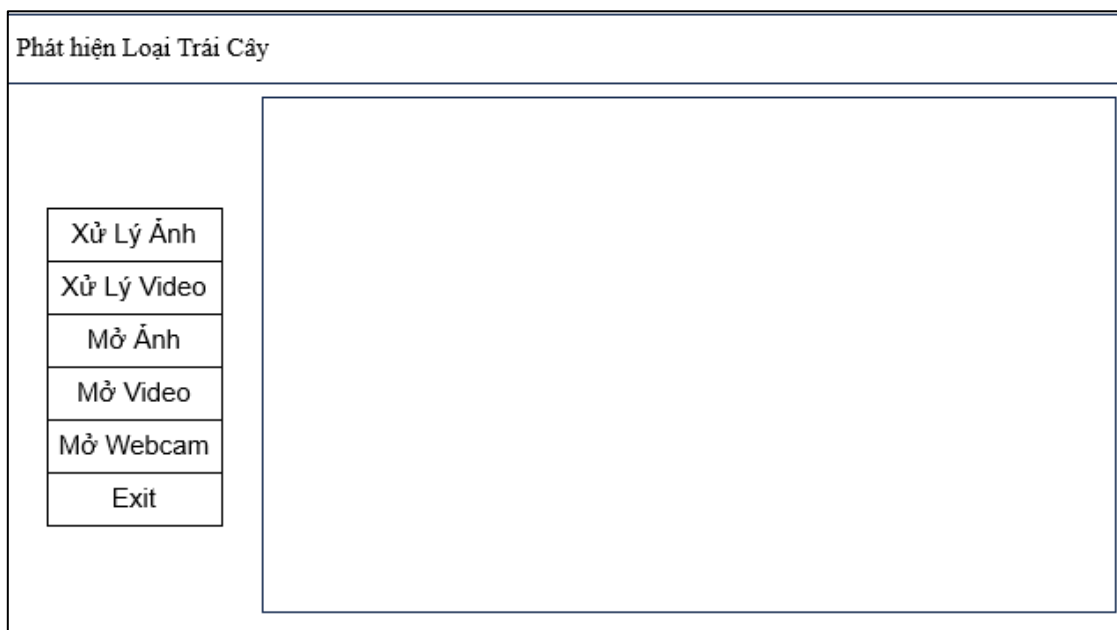
Confusion Matrix được sử dụng để đánh giá hiệu suất của một mô hình phân loại trong lĩnh vực học máy. Nó thể hiện số lượng các dự đoán chính xác và không chính xác giữa các lớp khác nhau. Cụ thể:

- Các Lớp: Bảng này bao gồm các lớp như táo, chuối, nho, ổi, xoài, cam, dưa hấu và nền.
- Giá Trị Ô: Các giá trị số trong mỗi ô của ma trận thể hiện số lượng trường hợp cho mỗi sự kết hợp của lớp thực tế và lớp dự đoán.
- Trục X và Y: Trục x với nhãn “True” biểu thị các lớp thực tế, trong khi trục y với nhãn “Predicted” biểu thị các lớp được mô hình dự đoán.

- Màu Sắc: Các ô có màu xanh đậm hơn chỉ ra giá trị cao hơn (Điều này thể hiện rằng mô hình đã dự đoán chính xác nhiều lần cho lớp đó); màu nhạt hơn chỉ ra giá trị thấp hơn (mô hình đã dự đoán ít lần cho lớp đó hoặc là không chính xác).

Ma trận nhầm lẫn giúp nhận diện được những lớp mà mô hình phân loại hoạt động tốt và những lớp mà mô hình có thể cải thiện.

3.3.2. Giao diện ứng dụng phát hiện loại trái cây



Hình 3.9: giao diện chương trình

Giao diện có các chức năng cơ bản gồm:

- Xử lý ảnh: Dùng model đã đào tạo trên Google Colab để xử lý nhận diện ảnh và lưu ảnh về định dạng .JPG và lưu vào 1 tệp chỉ định.
- Xử Lý Video: Dùng model đã đào tạo trên Google Colab để xử lý nhận diện vật trái cây trong video và xuất ra định dạng .mp4 và lưu vào 1 tệp chỉ định.
- Mở ảnh: dùng để chọn ảnh để nhận diện và xem kết quả
- Mở video: dùng để chọn video để nhận diện
- Mở webcam: dùng webcam của máy để nhận diện đối tượng (trái cây)
- Exit: thoát chương trình
- View: để xem ảnh, video

3.3.3. Kết quả thử nghiệm trên mô hình đã huấn luyện

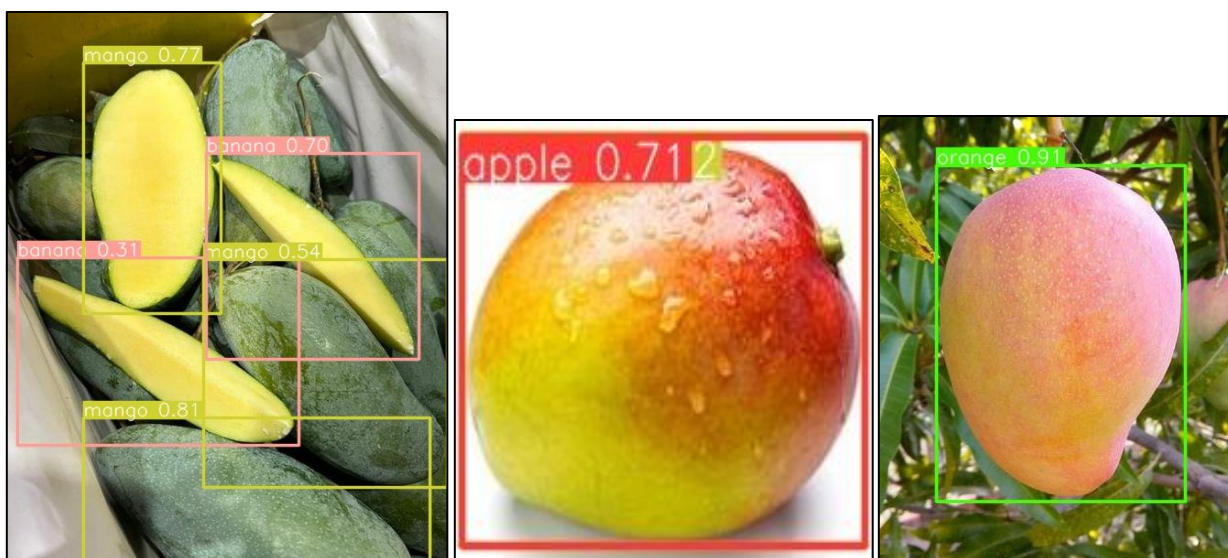
Thực hiện thủ công trên 2 loại dữ liệu đầu vào: ảnh tĩnh 1 loại quả và ảnh tĩnh nhiều loại quả.

Đối với ảnh tĩnh 1 loại quả chương trình đem lại độ chính xác trong khoảng 95%.

STT	Tên Quả	Số lượng ảnh	Nhận đúng	Nhận sai	Độ chính xác
1	Cam	30	30	0	100
2	Xoài	30	24	6	80
3	Táo	30	29	1	96
4	Chuối	30	29	1	96
5	Nho	30	28	2	93
6	Ổi	30	30	0	100
7	Dưa Hấu	30	30	0	100

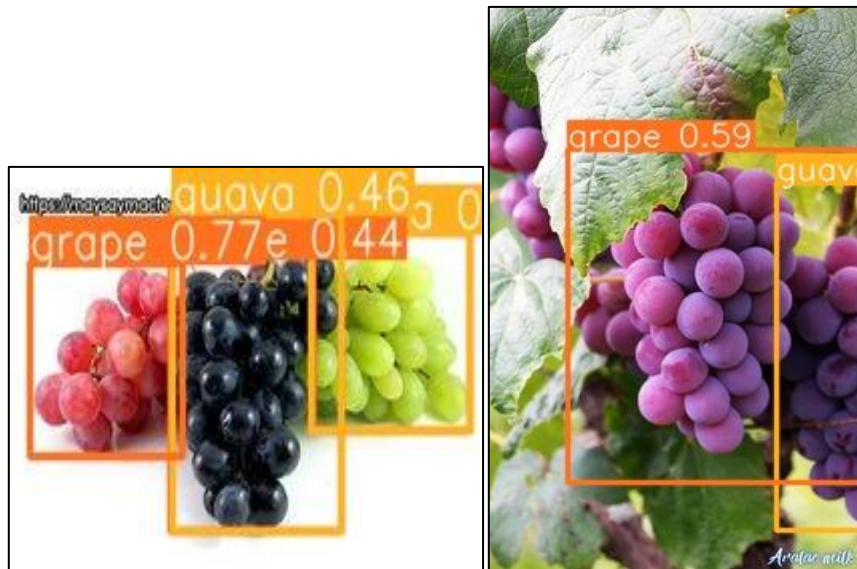
Bảng 3-3: Bảng thử nghiệm phân loại hình ảnh tĩnh

Theo như bảng trên ra có thể thấy nhận sai nhiều nhất nằm ở quả xoài với 80%.



Hình 3.10: trái xoài bị nhận sai

Tiếp theo sau là trái nho với 93%



Hình 3.11: chùm nho bị nhận diện sai

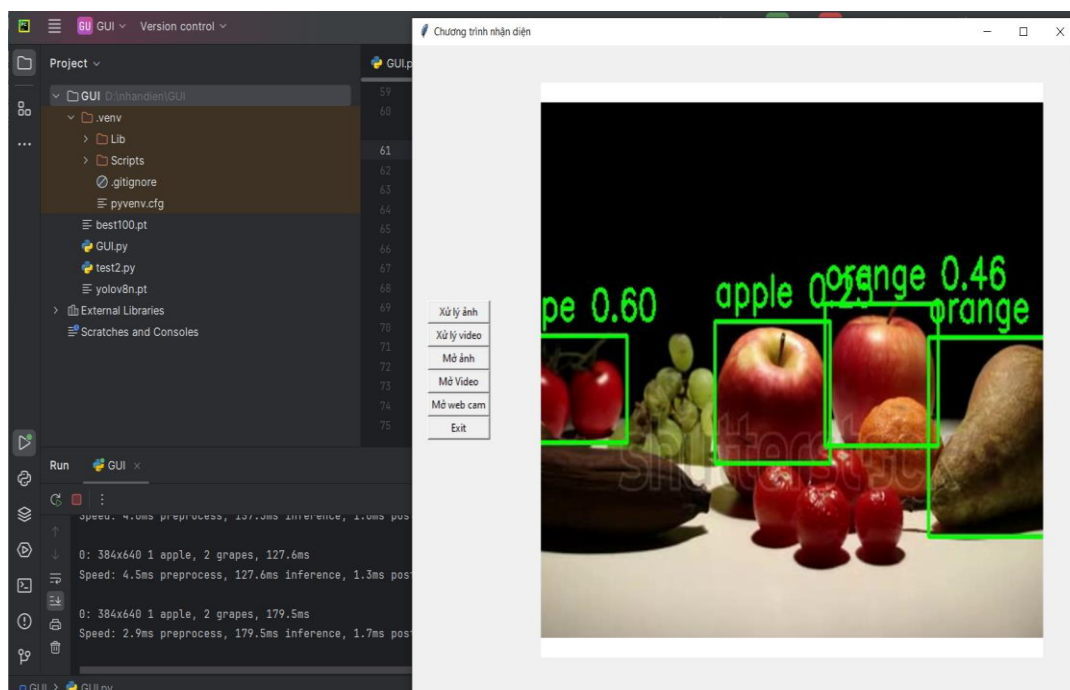
Đối với ảnh tĩnh nhiều quả trong 1 hình thì ra có được



Hình 3.12: Kết quả phân loại trái cây trên ảnh tĩnh có nhiều loại quả

Từ kết quả trên ta có thể thấy được chương trình nhận diện sai nhiều kể cả đối với các loại trái cây không có trong mô hình chương trình cũng nhận diện sai và thông báo độ chắc chắn lên tới ~51%.

Khi sử dụng video để nhận diện chương trình cũng đã thực hoàn thành tốt việc nhận diện những là video có nhiều góc độ khác nhau nên độ chắc chắn của mô hình sẽ có lúc không được quá cao.



Hình 3.13: nhận diện bằng video

3.4. Đánh giá mô hình sau khi thử nghiệm

Mô hình YOLOv8n đã được sử dụng để phân loại trái cây trên Google Colab, với bộ dữ liệu gồm khoảng 12,000 hình ảnh của 7 loại trái cây khác nhau. Mô hình đã được huấn luyện trong 100 epochs. Bài báo cáo này sẽ đánh giá các kết quả đạt được, cũng như nêu rõ các ưu điểm và nhược điểm của mô hình.

3.4.1. Kết quả huấn luyện:

- Độ chính xác:

Ở epoch thứ 90, mô hình đạt độ chính xác cao nhất là 75%.

Khi huấn luyện lên 100 epochs, độ chính xác giảm nhẹ so với tại epoch 90.

- Kiểm tra thủ công:

Khi kiểm tra thủ công trên bảng 3.2, độ chính xác trung bình đạt 95%.

Bộ ảnh kiểm tra thủ công lấy từ mạng có chất lượng cao và rõ nét, giúp mô hình đạt độ chính xác cao hơn so với bộ ảnh lấy từ webcam.

3.4.2. Thời gian huấn luyện:

- Mỗi 20/100 epochs tiêu tốn khoảng 2 giờ 6 phút 41 giây.
- Với bộ dữ liệu 12,000 ảnh và 100 epochs, thời gian huấn luyện kéo dài khoảng 10 ngày do giới hạn thời gian sử dụng phiên bản miễn phí của Google Colab (2-6 giờ mỗi phiên, phải chờ 2-3 ngày để có phiên mới).

3.4.3. Ưu điểm của mô hình:

- Hiệu suất và độ chính xác: YOLOv8n đã cho thấy hiệu suất khá tốt với độ chính xác cao đối với các hình ảnh tĩnh chứa một loại quả.
- Khả năng xử lý: Hệ thống xử lý nhanh và chính xác đối với các hình ảnh tĩnh và những video có 1 loại quả.

3.4.4. Nhược điểm của mô hình:

- Độ chính xác với hình ảnh phức tạp: Với các hình ảnh chứa nhiều loại trái cây khác nhau độ chính xác chỉ đạt mức trung bình.
- Sai sót trong nhận diện: Mô hình còn gặp khó khăn khi nhận diện các loại trái cây có màu sắc và hình dáng tương tự nhau, bị che khuất một phần hoặc để quá xa hay quá bé để mô hình có thể phát hiện được.

3.4.5. Đề xuất cải tiến:

- Thêm dữ liệu đào tạo: Bổ sung thêm dữ liệu đa dạng để cải thiện khả năng nhận diện.
- Nâng cấp mô hình: Sử dụng các phiên bản cao cấp hơn của YOLO như YOLOv9 mới ra mắt vào tháng 2/2024 để tăng độ chính xác.
- Tối ưu hóa tài nguyên: Sử dụng các dịch vụ đám mây khác hoặc phiên bản trả phí của Google Colab để có tài nguyên huấn luyện mạnh mẽ hơn.
- Mô hình YOLOv8n đã phần nào đáp ứng được các yêu cầu của đề tài, đặc biệt là đối với các hình ảnh tĩnh. Tuy nhiên, vẫn cần cải thiện thêm để tăng độ chính xác và giảm sai sót trong quá trình nhận diện trái cây. Việc nâng cấp mô hình và bổ sung dữ liệu sẽ là các bước quan trọng để đạt được kết quả tốt hơn trong tương lai.

KẾT LUẬN

Sau thời gian gần 5 tháng nghiên cứu và thực hiện đề án tốt nghiệp với sự hướng dẫn của cô TS. Hồ Thị Hương Thơm – Khoa Công Nghệ Thông Tin trường Đại Học Hàng Hải Việt Nam để hoàn thiện được đề án với đề tài: Phát hiện loại trái cây

Một số mục tiêu mà đề án đã đạt được:

Hoàn thành đề tài: ”Phát hiện loại trái cây”.

Đào tạo thành công mô hình nhận diện với 7 loại trái cây khác nhau: cam, xoài, táo, ổi, nho, dưa hấu, chuối bằng mô hình YOLOv8.

Mô hình phân loại được các loại trái cây với độ chính xác trung bình 75% với 100 epochs nhưng lại chính xác khá cao với ảnh tĩnh với một loại quả là 95% và vẫn còn nhiều sai sót với một số loại quả. Tuy nhiên khi đưa vào thực tiễn có thể đáp ứng được phần nào yêu cầu đặt ra nhưng có thể sẽ không được chính xác. Muốn cải thiện độ chính xác thì ảnh trước khi đưa vào phân loại cần phải được xử lý nâng cao chất lượng ảnh, hoặc ảnh được chụp quay từ các thiết bị chuyên dụng có chất lượng cao.

Chúng ta có thể phát triển chương trình này thành một mô-đun tiền xử lý hình ảnh cho các loại trái cây và rau củ, để tích hợp vào hệ thống robot phân loại tự động. Hệ thống này sẽ thực hiện bước tiền xử lý trước khi sản phẩm được đưa lên băng chuyền đóng gói hoặc nhận dạng tính tiền tự động trong siêu thị. Việc tự động hóa này sẽ góp phần hiện đại hóa ngành nông nghiệp và công nghiệp đóng gói, hướng tới quy trình sản xuất và đóng gói theo dây chuyền. Điều này sẽ hỗ trợ hiệu quả cho nền sản xuất trái cây và rau củ tại Việt Nam, một đất nước nổi tiếng với nhiều loại trái cây ngon xuất khẩu.

Trong quá trình học tập và nghiên cứu, tìm hiểu một lĩnh vực rất mới và nhiều thử thách đối với bản thân sinh viên cho nên đề án của sinh viên sẽ còn có những thiếu sót và hạn chế. Sinh viên rất cầu thị quý thầy cô trong khoa Công nghệ thông tin và Viện đào tạo sau đại học có cái nhìn cảm thông và chia sẻ đóng góp những lời nhận xét để sinh viên nâng cao sự hiểu biết về lĩnh vực AI vốn là lĩnh vực mới và hoàn thiện hơn đề án của mình. Sinh viên xin trân trọng cảm ơn.

TÀI LIỆU THAM KHẢO

- [1]. Gaudenz Boesch - Object Detection in 2024: The Definitive Guid từ <https://viso.ai/deep-learning/object-detection/>
- [2]. Bharath K - Object Detection Algorithms and Libraries Từ [Object Detection Algorithms and Libraries \(neptune.ai\)](https://neptune.ai/object-detection-algorithms-and-libraries)
- [3]. Nguyễn Thị Hậu - 2021-10-28 - Tổng quan về quả cam từ <https://ifarmer.vn/bai-viet/tong-quan-ve-qua-cam/>
- [4]. airnano - Cây Xoài: Đặc điểm Và Nguồn Gốc từ <https://airnano.vn/cay-xoai/>
- [5]. Táo tây – Wikipedia tiếng Việt từ https://vi.wikipedia.org/wiki/T%C3%A1o_t%C3%A2y
- [6]. Chuôi – Wikipedia tiếng Việt từ <https://vi.wikipedia.org/wiki/Chu%E1%BB%91i>
- [7]. Dưa hấu – Wikipedia tiếng Việt từ https://vi.wikipedia.org/wiki/D%C6%B0a_h%E1%BA%A5u
- [8]. Quả nho – Wikipedia tiếng Việt từ <https://vi.wikipedia.org/wiki/Nho>
- [9]. Ổi – Wikipedia tiếng Việt từ <https://vi.wikipedia.org/wiki/%E1%BB%94i>
- [10]. Ultralytics - GitHub - ultralytics/ultralytics: NEW - YOLOv8 in PyTorch > ONNX > OpenVINO > CoreML > TFLite từ <https://github.com/ultralytics/ultralytics>
- [11]. Gaudenz Boesch - A Guide to YOLOv8 in 2024 từ <https://viso.ai/deep-learning/yolov8-guide/>
- [12]. PyCharm - Wikipedia từ <https://en.wikipedia.org/wiki/PyCharm>
- [13]. wu1503 - 01/06/2020 - Google Colab Là Gì Và Dùng Để Làm Gì? (codelearn.io) từ <https://codelearn.io/sharing/google-colab-la-gi>
- [14]. Python (ngôn ngữ lập trình) – Wikipedia tiếng Việt từ [https://vi.wikipedia.org/wiki/Python_\(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh\)](https://vi.wikipedia.org/wiki/Python_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh))

- [15]. Python Là Gì? Tất Tần Tật Về Ngôn Ngữ Lập Trình Python từ <https://glints.com/vn/blog/ngon-ngu-lap-trinh-python-la-gi/>
- [16]. Google Colab là gì? Khám phá nền tảng sổ ghi chép tính toán trực tuyến từ <https://wikihow.com.vn/google-colab-la-gi/>
- [17]. R-CNN | Region Based CNNs từ <https://www.geeksforgeeks.org/r-cnn-region-based-cnns/>
- [18]. Tuan - Sep 19, 2020 - Bài 2. Object detection từ <https://medium.com/oho-software/b%C3%A0i-2-object-detection-c0adbcf505ba>
- [19]. Nguyễn Chiến Thắng – 14/10/2021 - Thử tìm hiểu về mAP - đo lường Object Detection model từ <https://www.miai.vn/2021/10/14/thu-tim-hieu-ve-map-do-luong-object-detection-model/>
- [20]. Jacob Solawetz, Francesco - JAN 11, 2023 - What is YOLOv8? The Ultimate Guide. [2024] (roboflow.com) từ <https://blog.roboflow.com/whats-new-in-yolov8/#yolov8-architecture-a-deep-dive>
- [21]. RizwanMunawar, glenn-jocher, AyushExel - 2023-11-12 - Ultralytics YOLOv8 Documents từ <https://docs.ultralytics.com/vi>