

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 thực hiện: Nguyễn Hải Bình

Giáo viên hướng dẫn: TS. Hồ Thị Hương Thơm

HẢI PHÒNG – 2025

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

**XÂY DỰNG MÔ HÌNH AI HỖ TRỢ XÁC ĐỊNH VÀ
PHÂN LOẠI MỘT SỐ SẢN PHẨM HÀNG HÓA**

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

NGÀNH: Công nghệ thông tin

Sinh viên thực hiện: Nguyễn Hải Bình

Giáo viên hướng dẫn: TS. Hồ Thị Hương Thơm

HẢI PHÒNG – 2025

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

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Tên đề tài: “XÂY DỰNG MÔ HÌNH AI HỖ TRỢ XÁC ĐỊNH VÀ PHÂN LOẠI
MỘT SỐ SẢN PHẨM HÀNG HÓA

Sinh viên làm đề tài tốt nghiệp

Sinh viên : Nguyễn Hải Bình

Mã SV: 2112111013

Lớp :CT2501C

Ngành : Công nghệ thông tin

Tên đề tài : Xây dựng mô hình phát hiện và phân loại hàng hóa

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 YOLO11 cho bài toán phát hiện loại hàng hóa , thiết kế xây dựng chương trình phát hiện loại hàng hóa.

b) Nội dung hướng dẫn

- Tìm hiểu về bài toán phát hiện đối tượng nói chung và phát hiện đối tượng bằng YOLO11.
- Phân tích, thu thập dữ liệu hình ảnh hàng hóa để huấn luyện phát hiện đối tượng.
- Cài đặt mô hình phát hiện đối tượng bằng YOLO11.
- Thử nghiệm mô hình trên hình ảnh và video để phát hiện hàng hóa.
- Nhận xét đánh giá, kết luận và đưa ra khuyến nghị khi ứng dụng.

c) Kết quả cần đạt được

Sau quá trình nghiên cứu, thu thập và xử lý dữ liệu, mô hình YOLO11 đã được huấn luyện thành công trên tập dữ liệu gồm nhiều loại hàng hóa khác nhau. Mục tiêu của quá trình huấn luyện là giúp mô hình có khả năng phát hiện và phân loại chính xác các sản phẩm trong ảnh được chụp trong môi trường thực tế như siêu thị, kho hàng hoặc cửa hàng bán lẻ.

Trong quá trình huấn luyện, các giá trị loss (box_loss, cls_loss, dfl_loss) trên cả tập train và validation đều giảm dần qua từng epoch. Điều này chứng tỏ mô hình học được mối quan hệ giữa đặc trưng hình ảnh và nhãn hàng hóa một cách ổn định, đồng thời không xuất hiện hiện tượng overfitting đáng kể.

Các hình ảnh minh họa từ tập kiểm thử cho thấy mô hình YOLO11 có khả năng khoanh vùng và gán nhãn chính xác các loại hàng hóa như bánh, kẹo, nước ngọt ngay cả khi có nhiều vật thể trong cùng một ảnh. Khung dự đoán (bounding box) khớp khá chính xác với biên của vật thể, nhãn hiển thị đúng lớp và có độ tin cậy (confidence) cao, trung bình đạt trên 0.85.

Kết quả cho thấy mô hình YOLO11 có hiệu năng vượt trội so với các phiên bản trước (YOLOv8, YOLOv5) về tốc độ và độ chính xác nhờ kiến trúc được tối ưu hóa và khả năng học đặc trưng sâu hơn. Với độ chính xác cao, tốc độ nhanh và tính ổn định, mô hình hoàn toàn có thể được áp dụng trong các bài toán tự động hóa quản lý hàng hóa, kiểm kê kho, giám sát bán hàng, hoặc hỗ trợ nhận dạng sản phẩm tại quầy thanh toán thông minh.

CÁC CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Họ và tên : Hồ Thị Hương Thom

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 YOLO11.
- Phân tích, thu thập dữ liệu hình ảnh hàng hóa để huấn luyện phát hiện đối tượng.
- Cài đặt mô hình phát hiện đối tượng bằng YOLO11.
- Thử nghiệm mô hình trên hình ảnh và video để phát hiện hàng hóa.
- 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 hàng hóa bằng YOLO11.
- Xây dựng được ứng dụng phát hiện loại hàng hóa trên hình ảnh và video.
- Viết báo cáo đồ án tốt nghiệp.

Đề tài tốt nghiệp được giao ngày 16 tháng 8 năm 2025

Yêu cầu phải hoàn thành xong

Đã nhận nhiệm vụ ĐTTN

Sinh viên

Đã giao nhiệm vụ ĐTTN

Giảng viên hướng dẫn

TS. Hồ Thị Hương Thom

Hải Phòng, ngày tháng năm 2025

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 Thom

Đơn vị công tác: Khoa Công nghệ thông tin - Trường Đại Học Hàng Hải Việt Nam

Họ và tên sinh viên : Nguyễn Hải Bình **Ngành:** Công nghệ Thông tin

Nội dung hướng dẫn : Toàn bộ đề tài

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

.....

.....

.....

.....

2.Đánh giá chất lượng của đồ 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 2025

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:.....

Đơn vị công tác:.....

Họ và tên sinh viên: Nguyễn Hải Bình **Ngành:** Công nghệ Thông tin

Đề tài tốt nghiệp: Xây dựng mô hình AI hỗ trợ xác định và phân loại một số sản phẩm hàng hóa

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 2025

Giảng viên chấm phản biện

(ký và ghi rõ họ tên)

LỜI CẢM ƠN

Trong quá trình làm đồ án vừa qua vì được sự chỉ dẫn nhiệt tình của thầy/cô TS. Hồ Thị Hương Thơm – Trường Đại học Hàng Hải Việt Nam, em đã hoàn thành đồ án của mình. Mặc dù em đã cố gắng với sự tận tâm của thầy/cô, nhưng vì thời gian và khả năng nên đồ án của em vẫn còn không tránh được những điều thiếu sót.

Em xin chân thành và bày tỏ lòng biết ơn sâu sắc đến thầy/cô vì đã tận tình chỉ bảo, hướng dẫn và giành thời gian quý báu của mình cho em trong thời gian qua để em có thể hoàn thành đồ án của mình đúng thời hạn.

Em xin cảm ơn tất cả thầy cô giáo trong khoa Công nghệ thông tin vì đã truyền đạt cho em rất nhiều các kiến thức nền tảng, chuyên ngành, chuyên môn và chuyên sâu cực kì vững chắc trong những năm qua để em có thể hoàn thành được đồ án này.

Em xin cảm ơn Trường Đại học Quản lý và Công nghệ Hải Phòng vì không ngừng hỗ trợ và đào tạo những điều kiện tốt nhất trong những năm vừa qua để em có thể học và thực hiện tốt đồ án.

Em xin chân thành cảm ơn !

LỜI CAM ĐOAN

Sinh viên xin cam đoan đề tài “ Xây dựng mô hình phát hiện người lái xe ô tô có hành vi sử dụng điện thoại khi lái xe” là công trình nghiên cứu độc lập dưới sự hướng dẫn của TS. Hồ Thị Hương Thơm – 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 Hải Bình

MỤC LỤC

LỜI CẢM ƠN.....	8
LỜI CAM ĐOAN	9
CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN	15
1.1. Mở đầu.....	15
1.2. Phát hiện đối tượng	15
1.2.1. Khái niệm.....	15
1.2.2. Cách phát hiện đối tượng	16
1.2.3. Các phương pháp phát hiện phổ biến hiện nay	17
1.2.4. Các phiên bản của YOLO.....	21
1.3. Đánh giá độ tin cậy của bài toán phân loại	33
1.3.1. Precision và Recall.....	33
1.3.2. AP	34
1.3.3. mAP.....	35
CHƯƠNG 2. NHẬN DẠNG ĐỐI TƯỢNG BẰNG PHƯƠNG PHÁP HỌC SÂU CHO BÀI TOÁN PHÂN LOẠI HÀNG HÓA.....	36
2.1. Mở đầu.....	36
2.1.1. Bài toán phát hiện loại hàng hóa	36
2.2. Sơ đồ hoạt động của mô hình hệ thống phát hiện hàng hóa	37
2.3. Một số hàng hóa.....	38
2.4. Quá trình huấn luyện.....	40
2.4.1. Thu thập ảnh	40
2.4.2 Đánh nhãn cho đối tượng	41
2.5. Giới thiệu về cấu trúc YOLO11.....	42
2.5.1. Giới thiệu	42
2.5.2. YOLO11 là gì?	42
2.5.3. Cấu trúc của YOLO11	43
2.6. Lợi ích của YOLO11 cho bài toán.....	48
CHƯƠNG 3 : CÀI ĐẶT, THỬ NGHIỆM VÀ ĐÁNH GIÁ	51
3.1. Mở đầu.....	51

3.2. Môi trường thử nghiệm	51
3.2.1. Python.....	51
3.2.2. Các đặc trưng Python	54
3.3. Google Collab	55
3.3.1. Khái niệm.....	55
3.3.2. Những đặc trưng chính trong Google Colab.....	55
3.4. Các phiên bản VS Code.....	56
3.5. Androi studio.....	58
3.5.1. Tổng quan về Android Studio.....	58
3.5.2. Cấu trúc và thành phần chính của Android Studio.....	58
3.5.3. Ưu điểm của Android Studio	59
3.5.4. Ứng dụng thực tế trong phát triển phần mềm	60
3.5.5. Kết luận.....	60
3.6. Tạo bộ dữ liệu học để huấn luyện.....	61
3.7. Đào tạo mô hình phát hiện loại hàng hóa	61
3.8. Giá trị các biểu đồ	65
3.10. Kết quả thử nghiệm trên mô hình đã huấn luyện trên desktop	66
3.11. Kết quả thử nghiệm mô hình đã huấn luyện trên android	67
3.12. Đánh giá mô hình sau khi thử nghiệm	68
KẾT LUẬN	70
TÀI LIỆU THAM KHẢO.....	71

Danh mục ảnh

Hình 1.1. Kiến trúc hệ thống HOG để phát hiện đối tượng.	17
Hình 1.2. Mô hình R-CNN.....	19
Hình 1.3. Faster R-CNN – Object Detection Algorithm.	20
Hình 1.4. Ứng dụng YOLOv3 cho tủ lạnh thông minh trong ẩm thực và nhà hàng.	23
Hình 1.5. Phát hiện đối tượng nhỏ với YOLOv5 trong phân tích giao thông.....	26
Hình 1.6. Ứng dụng hệ thống AI được đào tạo để phát hiện máy bay với YOLOv7.	27
Hình 1.7. Kiến trúc YOLOv8.	30
Hình 1.8. Hình dung anchor box trong YOLO.	30
Hình 1.9. Detection head của YOLOv8.....	31
Hình 1.10. Detection head của YOLOv8.....	31
Hình 1.11. Hiệu suất trên Tập dữ liệu Phát hiện Đối tượng MS COCO.....	32
Hình 1.12. Ví dụ biểu đồ thể hiện P và R để tính AP.	34
Hình 2.1. Sơ đồ hoạt động của mô hình.	37
Hình 2.2. Hình bim bim.	38
Hình 2.3. Hình coca.	39
Hình 2.4. Hình kẹo.....	39
Hình 2.5. Ảnh dữ liệu huấn luyện	41
Hình 2.5. Hình ảnh về Make sense.....	41
Hình 2.6. Các mô-đun kiến trúc chính trong YOLO11.....	44
Hình 2.7. So sánh YOLO11 với các phiên bản trước.....	47
Hình 3.1. Ngôn ngữ Python.....	54
Hình 3.2. Phần mềm VS Code.....	57
Hình 3.3. Các thư viện hỗ trợ.	62
Hình 3.4. Đoạn mã train.....	62
Hình 3.5. Code thư viện và chuyển đổi sang file tfile để chạy trên di động.....	63
Hình 3.6. Giao diện chương trình phát hiện trên laptop.....	64
Hình 3.7. Giao diện trên android studio	65

Hình 3.8. Biểu đồ thống kê sau thi đào tạo.....	65
Bảng 3.9. Nước ngọt bị nhận sai.	67
Hình 3.10. Kẹo bị nhận diện sai.	67

Danh mục bảng

Bảng 3.1. Số lượng ảnh của từng loại.....	61
Bảng 3.2. Thử nghiệm phân loại hình ảnh tĩnh.....	66
Bảng 3.3. Thử nghiệm phân loại hình ảnh tĩnh.....	68

Danh mục từ viết tắt

STT	Từ viết tắt	Nghĩa tiếng Anh	Nghĩa tiếng Việt
1	AI	Artificial Intelligence	Trí tuệ nhân tạo
2	CNN	Convolutional Neural Network	Mạng nơ-ron tích chập
3	R-CNN	Region-based Convolutional Neural Network	Mạng CNN dựa trên vùng
4	Fast R-CNN	Fast Region-based CNN	Mạng CNN nhanh dựa trên vùng
5	Faster R-CNN	Faster Region-based CNN	Mạng CNN nhanh hơn dựa trên vùng
6	YOLO	You Only Look Once	Thuật toán phát hiện đối tượng YOLO
8	HOG	Histogram of Oriented Gradient	Biểu đồ Gradien hướng
9	GPU	Graphics Processing Unit	Bộ xử lý đồ họa
10	TPU	Tensor Processing Unit	Bộ xử lý Tensor
11	CPU	Central Processing Unit	Bộ xử lý trung tâm

12	IDE	Integrated Development Environment	Môi trường phát triển tích hợp
13	IT	Information Technology	Công nghệ thông tin
14	HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
15	CSS	Cascading Style Sheets	Ngôn ngữ định kiểu trang
16	JS	JavaScript	Ngôn ngữ lập trình JavaScript
21	VS Code	Visual Studio Code	Trình soạn thảo mã nguồn VS Code
22	API	Application Programming Interface	Giao diện lập trình ứng dụng
23	Colab	Google Colaboratory	Công cụ lập trình Google Colab

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN

1.1. Mở đầu

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 hàng hóa, nơi việc phát hiện và phân loại hàng hóa 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 hàng hóa có thể cải thiện đáng kể hiệu suất và chất lượng sản phẩm.

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.

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?’.

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

cột, 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 hàng hóa.

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.

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.

- Độ chính xác cao hơn so với các phương pháp truyền thống.

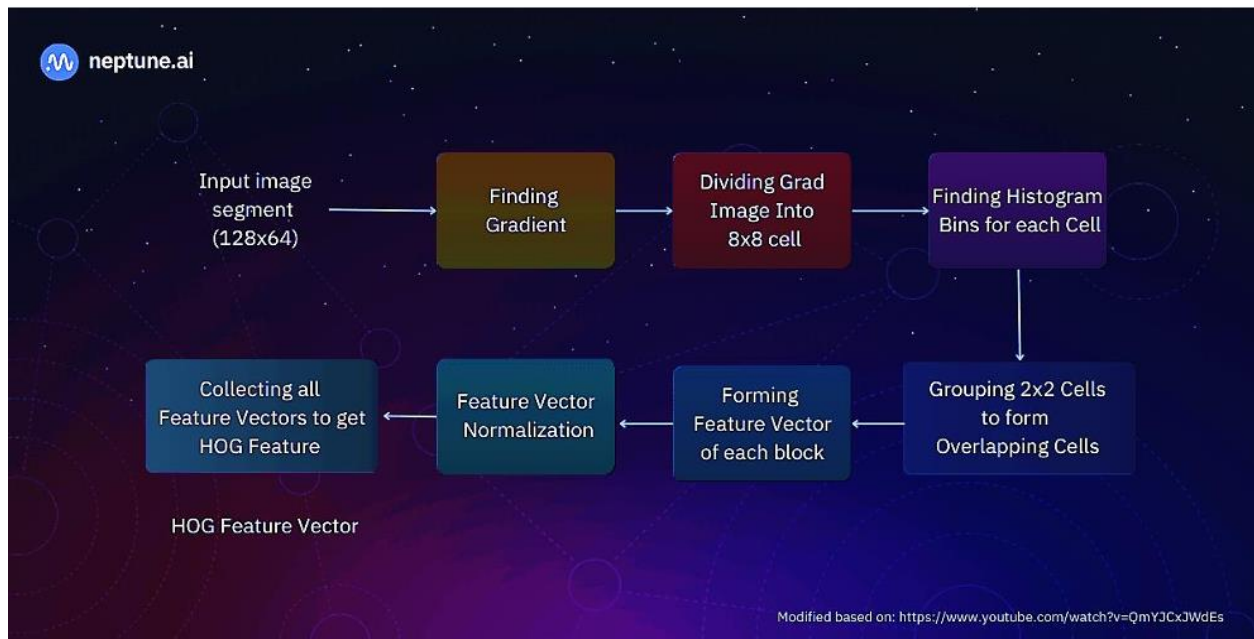
Nhược điểm:

- 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, PAS CAL 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.

1.2.3. Các phương pháp phát hiện phổ biến hiện nay

a. Phương pháp mô tả đặc trưng (HOG)



Hình 1.1. 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 trình trích xuất đặc trưng để xác định các đối tượng trong hình ảnh.

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.

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.

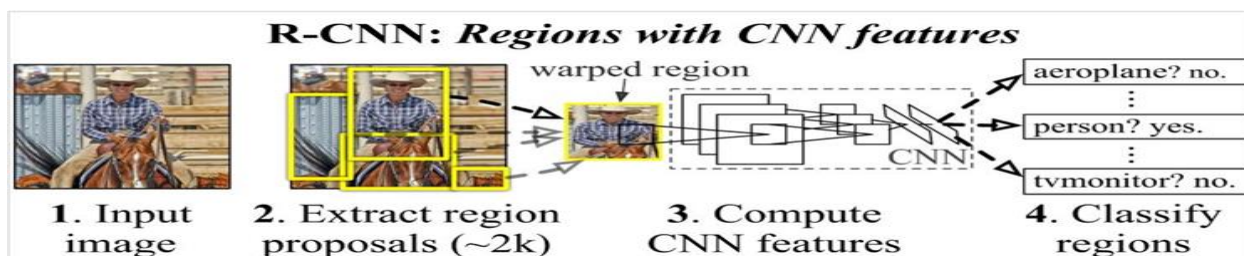
Như thể hiện trong biểu diễn hình ảnh ở trên, mô hình sẽ xem xét một phân đoạn hình ảnh có kích thước cụ thể. Bước đầu tiên là tìm gradient bằng cách chia toàn bộ tính toán của hình ảnh thành các biểu diễn gradient của 8×8 ô. Với sự trợ giúp của 64 vector gradient đạt được, chúng ta có thể chia mỗi ô 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.

Khi mô hình 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.

b. Mạng nơ-ron

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



Hình 1.2. Mô hình R-CNN.

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.

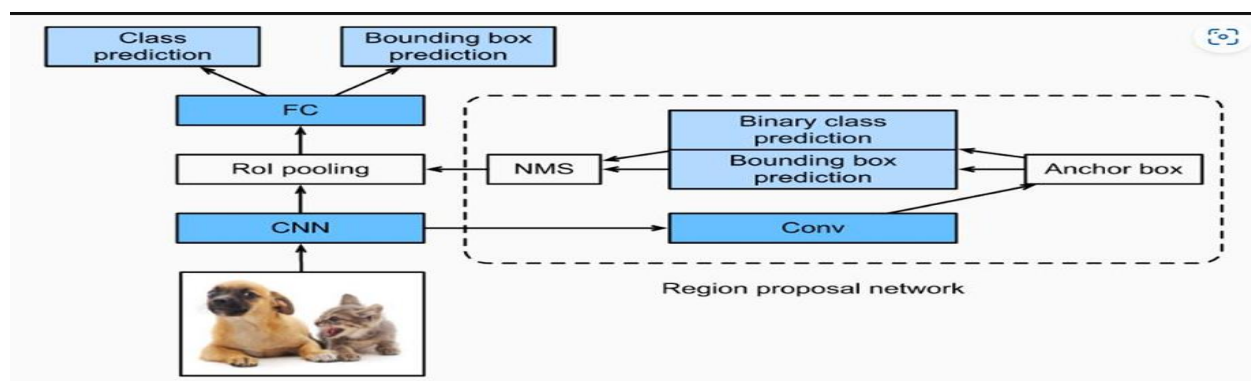
Quy trình làm việc của thuật toán tìm kiếm chọn lọc để chọn các đề xuất khu vực quan trọng nhất bao gồm việc tạo nhiều phân đoạn phụ trên một hình ảnh cụ thể và chọn các mục ứng cử viên cho nhiệm vụ. Thuật toán tham lam sau đó có thể được sử dụng để kết hợp các mục hiệu quả phù hợp cho một quy trình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

Khi thuật toán tìm kiếm chọn lọc hoàn thành thành công, nhiệm vụ tiếp theo là trích xuất các đặc trưng và đưa ra dự đoán thích hợp. Sau đó 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, ta có thể đạt được nhiệm vụ trích xuất đặc trưng.

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.

c. 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.



Hình 1.3. Faster R-CNN – Object Detection Algorithm.

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.

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 có thể dự đoán một lớp nhị phân cụ thể và tạo ra một hộp giới hạn cho nó.

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.

d. YOLO

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.

1.2.4. Các phiên bản của YOLO

a. 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.

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.

b. 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.

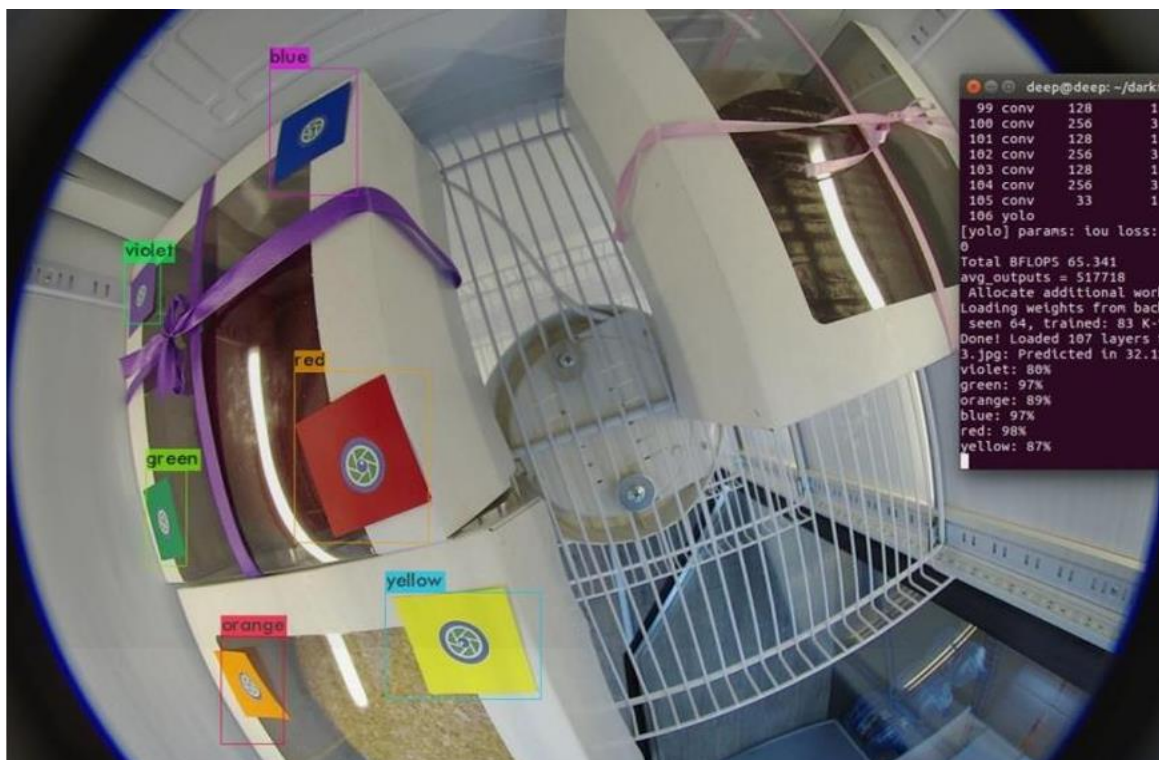
- Hiệu suất vượt trội: YOLOv2 đạt được độ chính xác trung bình (mean Average Precision - mAP) là 76.8 và tốc độ suy luận 67 khung hình mỗi giây (FPS) trên bộ dữ liệu VOC 2007.

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.

c. 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.



Hình 1.4. Ứng dụng YOLOv3 cho tủ lạnh thông minh trong ẩm 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.

1. 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.

2. 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.

3. 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.

d. YOLO v4

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.

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.

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.

e. YOLO v5

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.



Hình 1.5. Phát hiện đối tượng nhỏ với YOLOv5 trong phân tích giao thông.

f. 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 nổi 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.

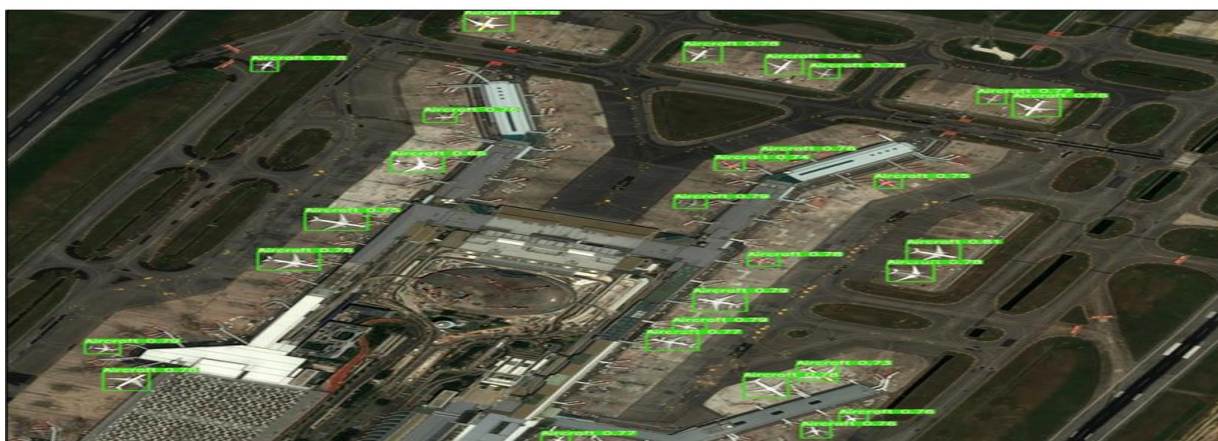
- Chiến lược đào tạo có sự hỗ trợ của neo (Anchor-Assisted Training): Mô hình này đề xuất Anchor-Assisted Training để tận hưởng những lợi ích của cả mô hình dựa trên neo và không có neo mà không ảnh hưởng đến hiệu quả suy luận.

- 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.

g. 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 1.6. 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 nhãn 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.

h. 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.

Kiến trúc của yolo v8

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.

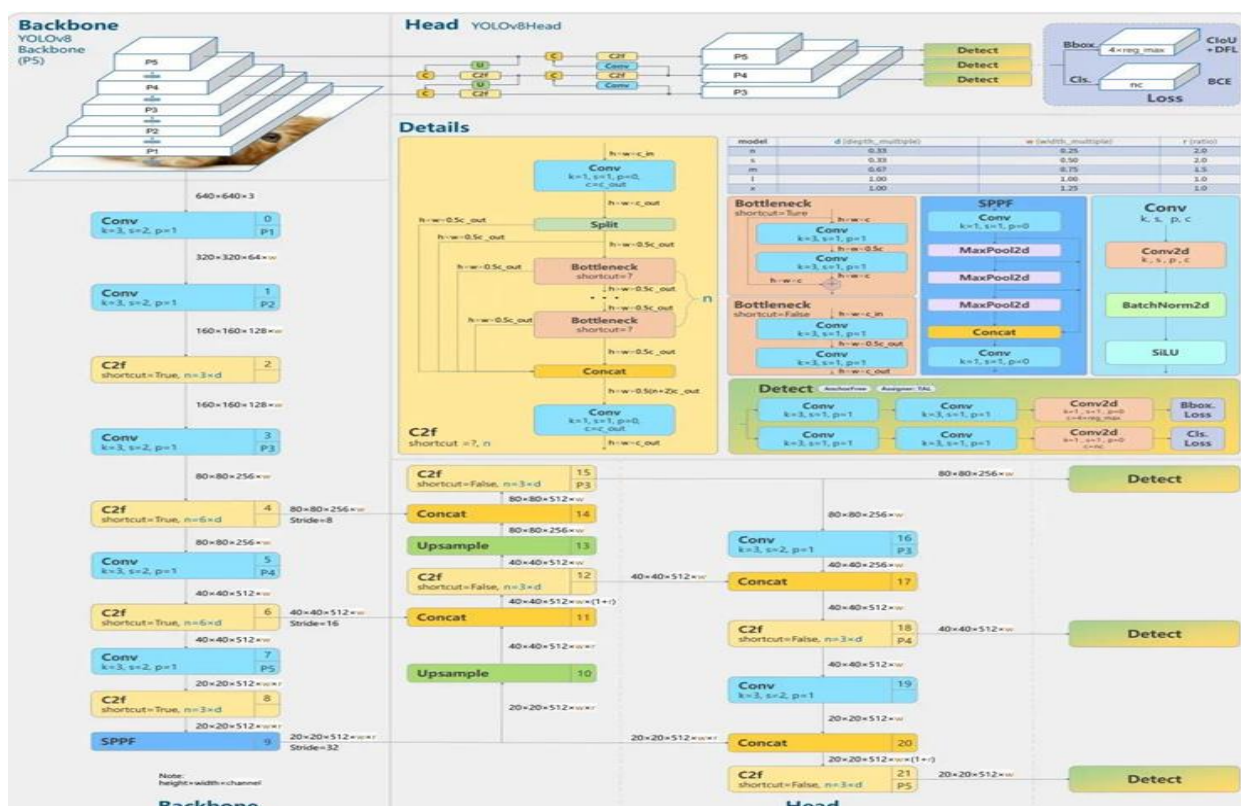
Cấu trúc của YOLOv8 được chia thành ba thành phần chính: xương sống, cổ và đầu. Xương sống chịu trách nhiệm trích xuất các đặc trưng từ hình ảnh đầu vào, với các tùy chọn như CSPDarknet53 và EfficientDet để phù hợp với nhu cầu cụ thể. Phần cổ kết nối xương sống với đầu, đóng vai trò quan trọng trong 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.

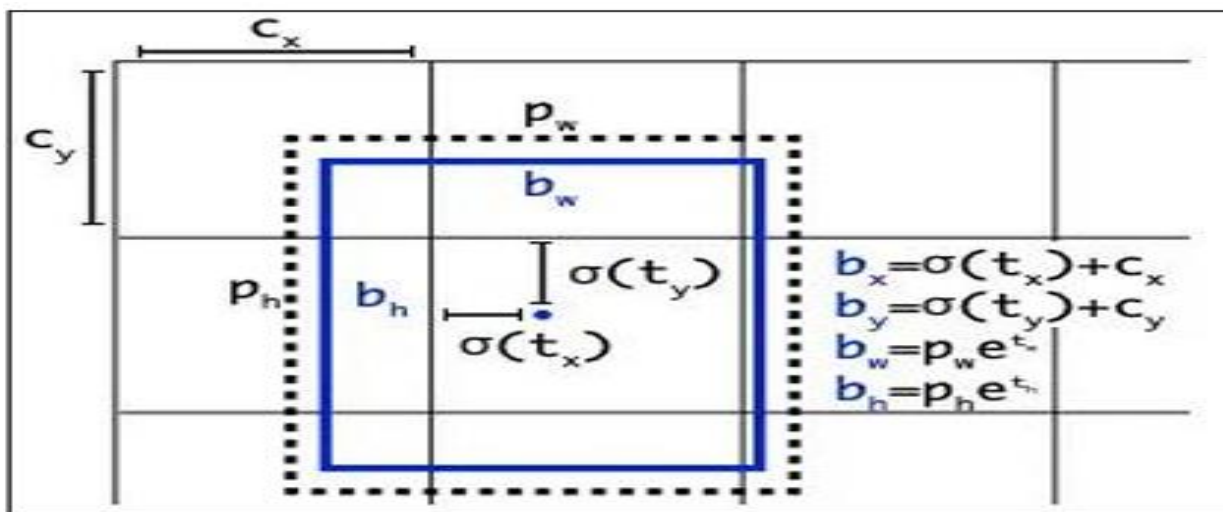
Tóm lại, kiến trúc của YOLOv8 nổi bật với thiết kế mô-đun, các biến thể có thể mở rộng, xương sống được cải thiện và các chiến lược đào tạo nâng cao. Những đặc điểm này cùng nhau góp phần vào sự thành công của YOLOv8 trong việc phát hiện đối tượng thời gian thực, khiến nó trở thành lựa chọn phổ biến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



Hình 1.7. Kiến trúc YOLOv8.

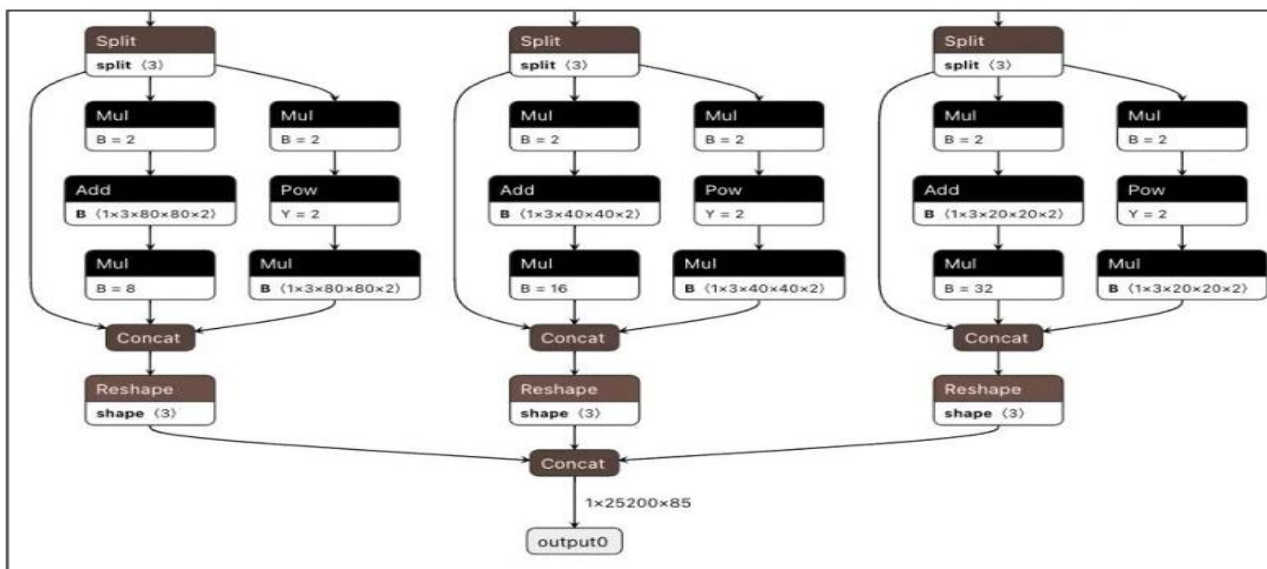
Phát hiện không neo

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.



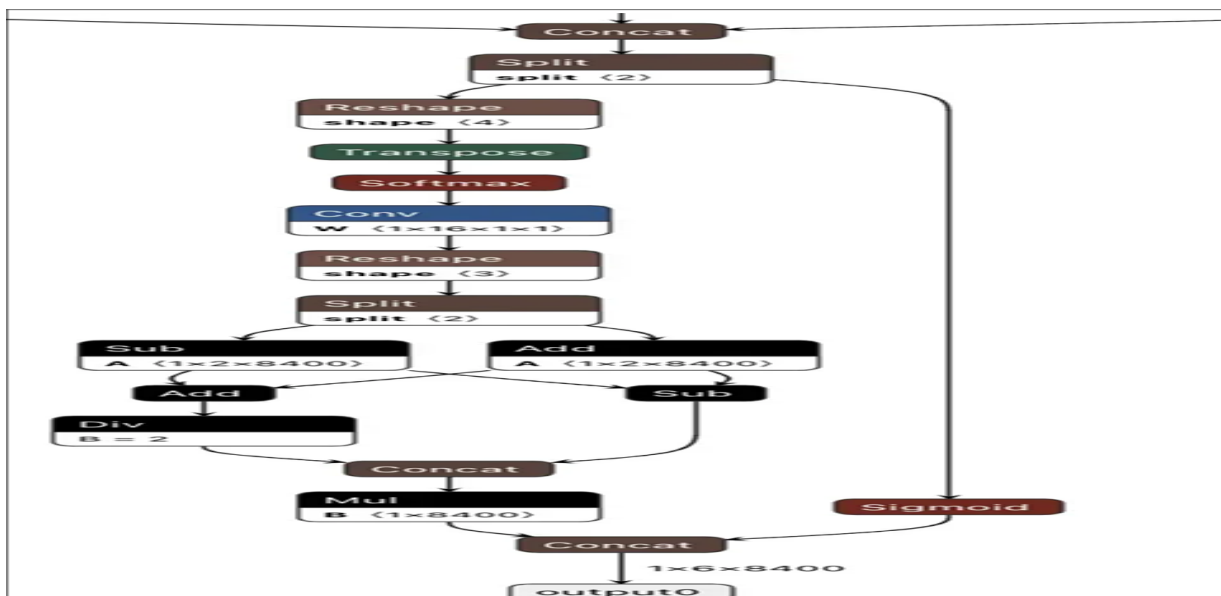
Hình 1.8. Hình dung anchor box trong YOLO.

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 1.9. Detection head của YOLOv8.

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 1.10. Detection head của YOLOv8.

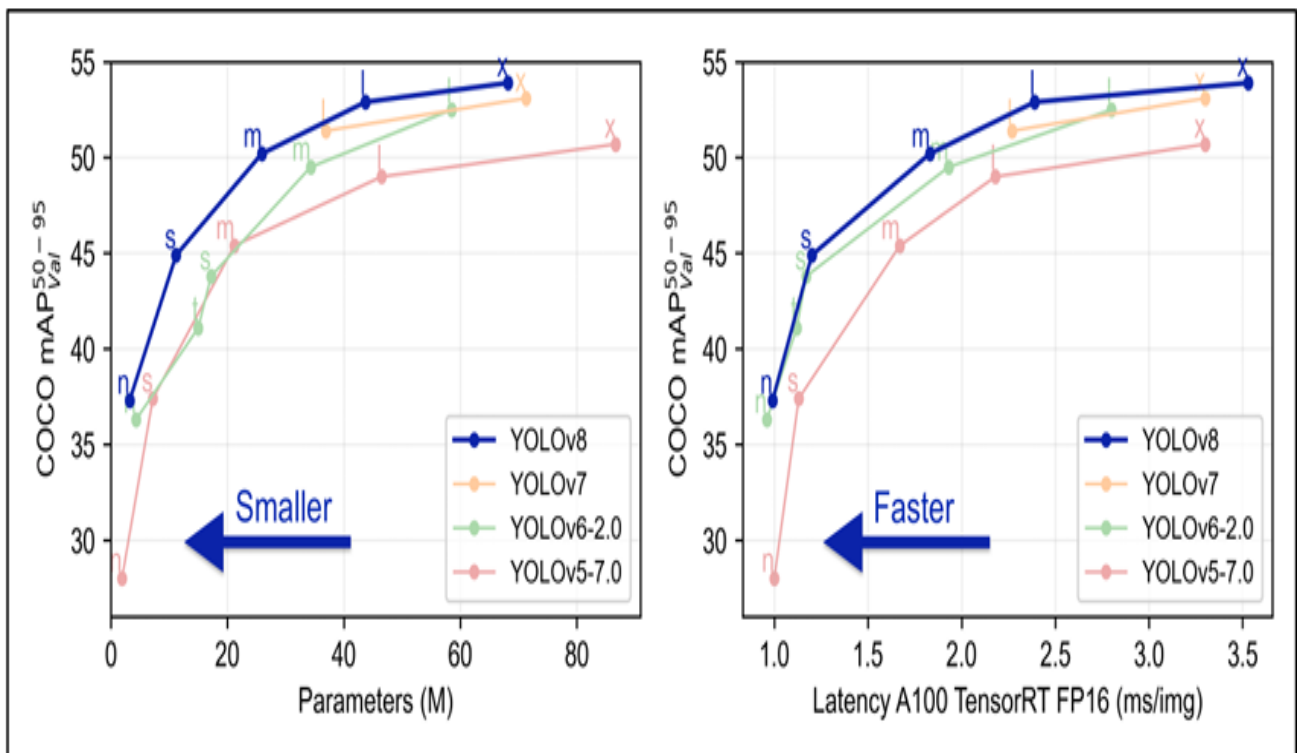
Các đặc trưng chính:

-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 1.11. Hiệu suất trên Tập dữ liệu Phát hiện Đối tượng MS COCO.

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

1.3.1. 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 :

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

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 :

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

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.

1.3.2. 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.

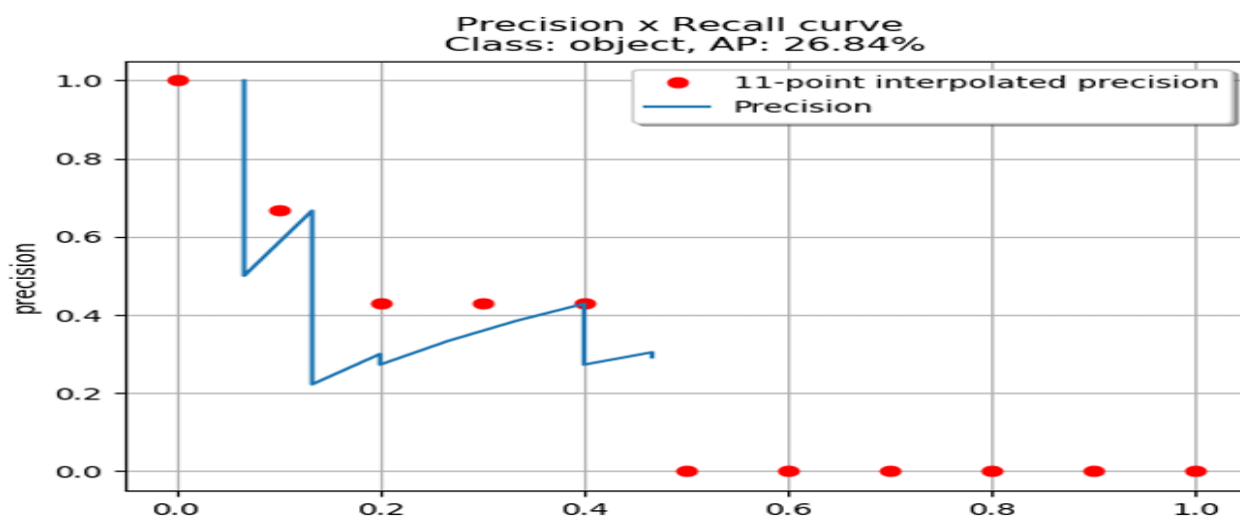
Sử dụng Tính toán nội suy 11 điểm là tính trung bình độ chính xác ở một tập hợp 11 mức thu hồi (0,0,1,...,1). Các giá trị chính xác nội suy thu được bằng cách lấy độ chính xác tối đa có giá trị thu hồi lớn hơn giá trị thu hồi hiện tại của nó như sau:

$$\text{Trong đó : } AP = \frac{1}{11} \sum_{r \in \{0,0.1, \dots, 1\}} p_{\text{interp}}(r) \quad (1-3)$$

$$p_{\text{interp}} = \max p(\tilde{r}) \quad (1-4)$$

$p(\tilde{r})$ 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 lấy độ chính xác tối đa có giá trị thu hồi lớn hơn.



Hình 1.12. Ví dụ biểu đồ thể hiện P và R để tính AP.

Bằng cách áp dụng nội suy 11 điểm:

$$AP = 1/11 (1 + 0.6666 + 0.4285 + 0.4285 + 0.4285 + 0 + 0 + 0 + 0 + 0 + 0)$$

$$AP = 26.84\%$$

1.3.3. *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:

$$mAP = (AP_a + AP_b + AP_c) / 3 \quad (1-5)$$

CHƯƠNG 2. NHẬN DẠNG ĐỐI TƯỢNG BẰNG PHƯƠNG PHÁP HỌC SÂU CHO BÀI TOÁN PHÂN LOẠI HÀNG HÓA

2.1. Mở đầu

Trong chương này, các kiến thức nền tảng về học sâu sẽ được trình bày nhằm cung cấp cơ sở lý thuyết cho việc xây dựng mô hình phân loại hàng hóa. Nội dung sẽ bao gồm khái niệm, đặc điểm, ưu điểm của học sâu và cách nó được áp dụng trong các bài toán nhận dạng đối tượng. Đây là bước quan trọng để làm rõ vai trò của học sâu trong việc giải quyết bài toán phân loại hàng hóa, đồng thời tạo tiền đề cho việc triển khai mô hình trong các phần tiếp theo.

2.1.1. Bài toán phát hiện loại hàng hóa

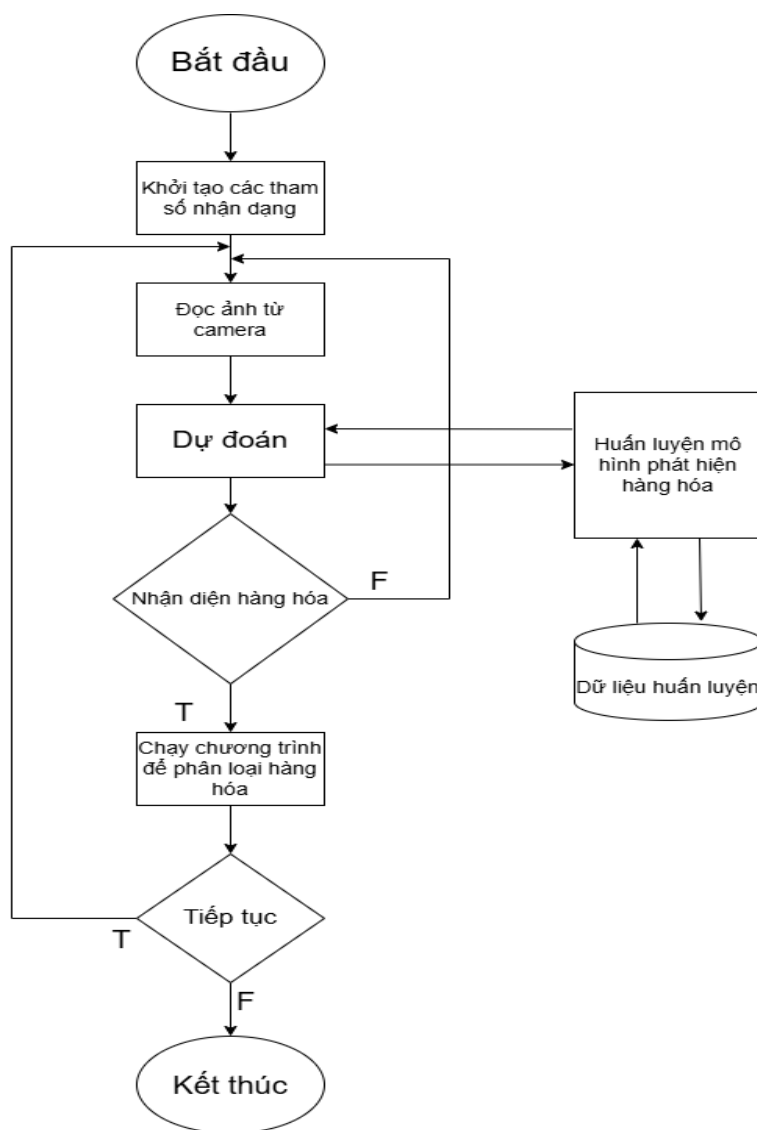
- Với bài toán phát hiện loại hàng hóa (bimbim, kẹo, nước ngọt, ...) bài toán cần đảm nhận được những vấn đề cơ bản nhất của một quá trình phát hiện đối tượng thông thường:

- Ứng dụng một mô hình học sâu hiệu quả trong phân loại và nhận dạng hàng hóa: 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 hàng hóa, 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 hàng hóa tương đối đầy đủ: Để phân loại và nhận dạng hàng hóa, cần phải có tập dữ liệu ảnh tương đối đầy đủ, đảm bảo các loại hàng hóa 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 hàng hóa khác nhau.

2.2. Sơ đồ hoạt động của mô hình hệ thống phát hiện hàng hóa



Hình 2.1. Sơ đồ hoạt động của mô hình.

Để phát hiện hàng hóa một cách hiệu quả, chúng ta cần một quy trình xử lý dữ liệu từ camera đến khi đưa ra được kết quả cuối cùng. Sơ đồ này là nền tảng giúp hệ thống hoạt động trơn tru và an toàn.

Đầu vào (Input): Dữ liệu này có thể là các khung hình (frame) tĩnh từ video hoặc một luồng hình ảnh trực tiếp.

Tiền xử lý dữ liệu (Preprocessing): Các khung hình được chuẩn hóa về kích thước và định dạng phù hợp với yêu cầu của mô hình. Bước này đảm bảo rằng tất cả dữ liệu đầu vào đều tương thích, giúp mô hình hoạt động ổn định.

Mô hình phát hiện (Detection Model): Khung hình đã được tiền xử lý sẽ được đưa vào mô hình học sâu đã được huấn luyện (ví dụ: mô hình YOLO của bạn). Mô hình này sẽ xử lý hình ảnh và đưa ra các dự đoán.

Hậu xử lý kết quả (Post-processing): Mô hình có thể tạo ra nhiều hộp giới hạn chồng chéo nhau cho cùng một đối tượng. Bước này sử dụng thuật toán Non-Maximum Suppression (NMS) để loại bỏ các hộp giới hạn trùng lặp và chỉ giữ lại hộp có độ chính xác cao nhất, giúp kết quả rõ ràng và chính xác hơn.

Đầu ra (Output): Kết quả cuối cùng là các hộp giới hạn được vẽ xung quanh vật thể (ví dụ: người đi bộ, ô tô, biển báo), cùng với nhãn và độ tin cậy tương ứng.

2.3. Một số hàng hóa

Bim bim

Bim bim, hay còn gọi là snack, là một loại thức ăn nhẹ phổ biến, được làm từ bột và gia vị (như vị tôm, bò, cua, cay, ngọt...) hoặc các loại hạt. Sản phẩm được sấy khô, tạo ra hương vị mặn, cay chủ đạo nhưng cũng có loại ngọt, và có nhiều hãng sản xuất khác nhau.



Hình 2.2. Hình bim bim.

Nước ngọt

Coca-Cola là một thương hiệu nước ngọt nổi tiếng toàn cầu, được phát minh bởi dược sĩ John Stith Pemberton vào năm 1886 tại Atlanta, Hoa Kỳ. Ban đầu được quảng cáo như một loại thuốc bổ, Coca-Cola giờ đây đã phát triển thành một công ty nước giải khát hàng đầu thế giới, sở hữu nhiều nhãn hiệu đồ uống khác nhau. Sản phẩm cốt lõi của công ty, Coca-Cola, nổi tiếng với hương vị đặc trưng từ đường, dầu cam, vanilla và quế, cùng với chất acid phosphoric.



Hình 2.3. Hình coca.

Kẹo

Kẹo là loại thực phẩm ở dạng viên, thời có chứa thành phần chính là đường ăn. Từ này cũng được gọi chung là bánh mút kẹo, bao hàm bất kỳ loại kẹo ngọt nào, gồm sô-cô-la, kẹo cao su và kẹo đường. Rau quả, các loại hạt mà được lên men và phủ đường được gọi là kẹo hoa quả.



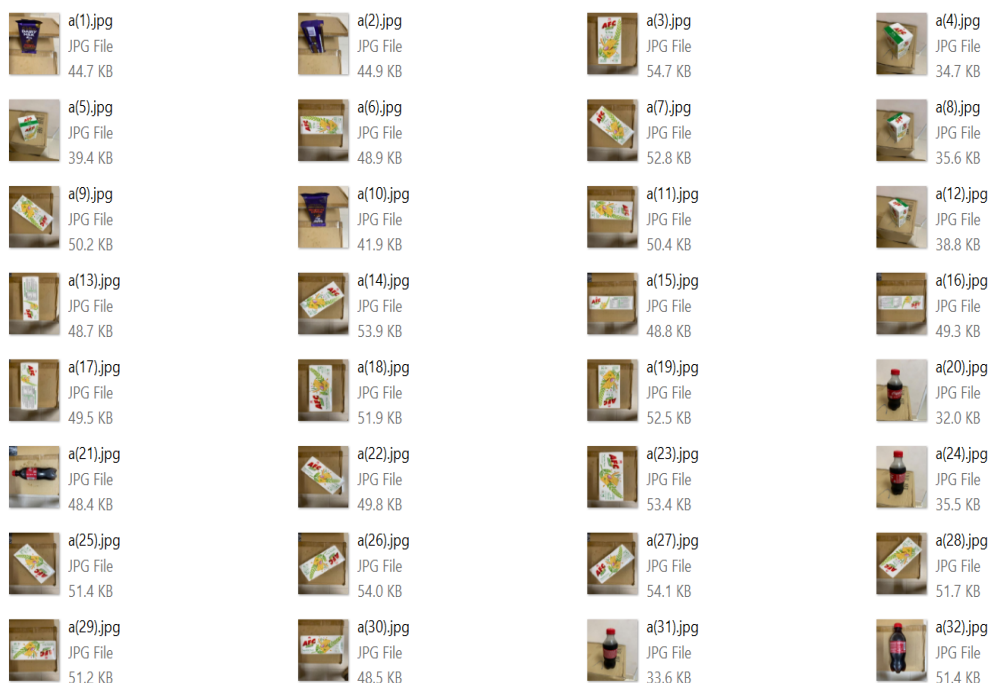
Hình 2.4. Hình kẹo.

2.4. Quá trình huấn luyện

2.4.1. Thu thập ảnh

Dữ liệu ảnh được thu thập từ hai nguồn chính:

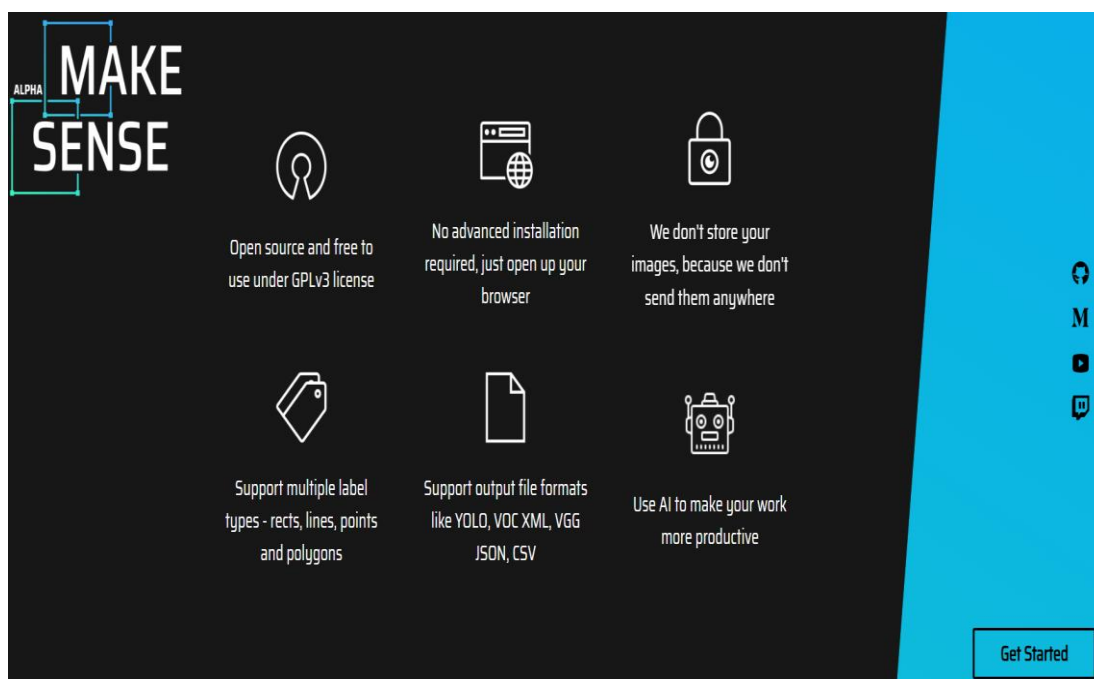
- Chụp trực tiếp bằng điện thoại hoặc máy ảnh tại siêu thị, cửa hàng tạp hóa hoặc kho hàng.
- Thu thập từ Internet thông qua các trang thương mại điện tử, catalog sản phẩm hoặc website nhà cung cấp.
- Việc kết hợp nhiều nguồn giúp tăng tính đa dạng của dữ liệu, giảm hiện tượng lệch dữ liệu.
- Chất lượng 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.
- 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 tệp 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.
- 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.



Hình 2.5. Ảnh dữ liệu huấn luyện

2.4.2 Đánh nhãn cho đối tượng

Sử dụng công cụ có sẵn trên mạng là makesense để gán nhãn cho đối tượng, trong bài này đối tượng sẽ là tài xế lái xe (<https://www.makesense.ai/>).



Hình 2.5. Hình ảnh về Make sense.

2.5. Giới thiệu về cấu trúc YOLO11

2.5.1. Giới thiệu

Tầm nhìn máy tính, một lĩnh vực phát triển nhanh chóng, cho phép máy móc diễn giải và hiểu dữ liệu trực quan. Một khía cạnh quan trọng của miền này là phát hiện đối tượng, bao gồm việc xác định chính xác và định vị các đối tượng trong hình ảnh hoặc luồng video. Những năm gần đây đã chứng kiến sự tiến bộ đáng kể trong các phương pháp tiếp cận thuật toán để giải quyết thách thức này.

Một bước đột phá quan trọng trong việc phát hiện đối tượng đến từ việc Redmon và cộng sự giới thiệu thuật toán You Only Look Once (YOLO) vào năm 2015. Phương pháp tiếp cận sáng tạo này, đúng như tên gọi của nó, xử lý toàn bộ hình ảnh chỉ trong một lần quét để phát hiện vật thể và vị trí của chúng. Phương pháp của YOLO khác biệt so với các quy trình phát hiện hai giai đoạn truyền thống bằng cách định hình việc phát hiện vật thể như một bài toán hồi quy. Nó sử dụng một mạng nơ-ron tích chập duy nhất để đồng thời dự đoán các hộp giới hạn và xác suất lớp trên toàn bộ hình ảnh, hợp lý hóa quy trình phát hiện so với các phương pháp truyền thống phức tạp hơn.

YOLO11 là phiên bản mới nhất trong chuỗi YOLO, được xây dựng dựa trên nền tảng của YOLOv1. Được công bố tại hội nghị YOLO Vision 2024 (YV24), YOLO11 đại diện cho một bước tiến vượt bậc trong công nghệ phát hiện đối tượng thời gian thực. Phiên bản mới này mang đến những cải tiến đáng kể về cả kiến trúc và phương pháp đào tạo, mở rộng ranh giới về độ chính xác, tốc độ và hiệu quả.

2.5.2. YOLO11 là gì?

Sự phát triển của thuật toán YOLO đạt đến tầm cao mới với sự ra đời của YOLO11, đại diện cho một bước tiến đáng kể trong công nghệ phát hiện đối tượng thời gian thực. Phiên bản mới nhất này kế thừa những điểm mạnh của các phiên bản

trước, đồng thời giới thiệu những tính năng mới giúp mở rộng tiện ích trên nhiều ứng dụng CV khác nhau.

YOLO11 nổi bật với khả năng thích ứng được cải tiến, hỗ trợ phạm vi tác vụ CV mở rộng vượt ra ngoài phạm vi phát hiện đối tượng truyền thống. Đáng chú ý trong số đó là ước tính tư thế và phân đoạn thực thể, mở rộng khả năng ứng dụng của mô hình trong nhiều lĩnh vực. Thiết kế của YOLO11 tập trung vào việc cân bằng giữa sức mạnh và tính thực tiễn, hướng đến việc giải quyết các thách thức cụ thể trong nhiều ngành công nghiệp với độ chính xác và hiệu quả cao hơn.

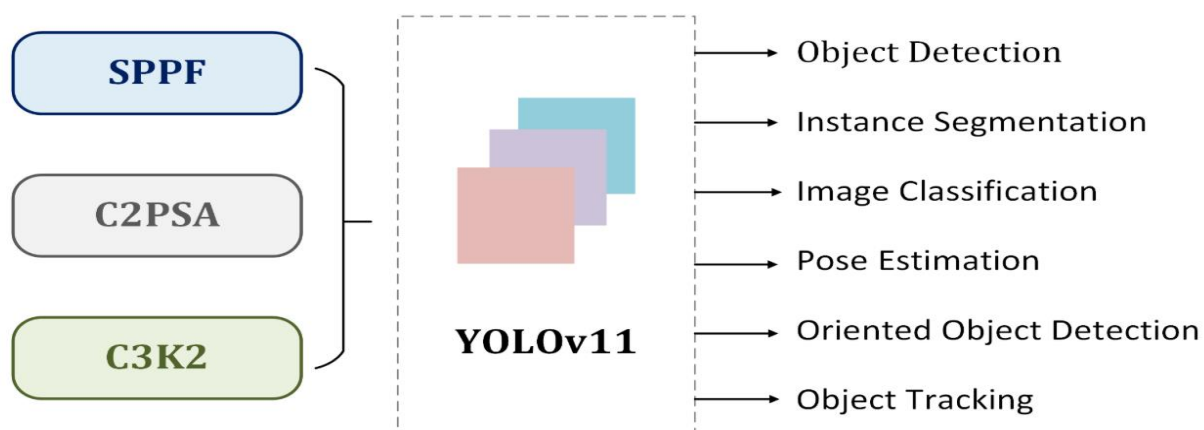
Mô hình mới nhất này chứng minh sự phát triển không ngừng của công nghệ phát hiện đối tượng thời gian thực, mở rộng giới hạn khả năng của các ứng dụng CV. Tính linh hoạt và cải tiến hiệu suất của nó định vị YOLO11 là một bước tiến đáng kể trong lĩnh vực này, có khả năng mở ra những hướng đi mới cho việc triển khai thực tế trên nhiều lĩnh vực khác nhau.

2.5.3. Cấu trúc của YOLO11

- Khung YOLO đã cách mạng hóa việc phát hiện đối tượng bằng cách giới thiệu một kiến trúc mạng nơ-ron thống nhất có thể xử lý đồng thời cả hồi quy hộp giới hạn và các tác vụ phân loại đối tượng. Phương pháp tiếp cận tích hợp này đánh dấu sự thay đổi đáng kể so với các phương pháp phát hiện hai giai đoạn truyền thống, cung cấp khả năng đào tạo toàn diện thông qua thiết kế có thể phân biệt hoàn toàn.

- Về cốt lõi, kiến trúc YOLO bao gồm ba thành phần cơ bản. Thứ nhất, xương sống đóng vai trò là bộ trích xuất đặc trưng chính, sử dụng mạng nơ-ron tích chập để chuyển đổi dữ liệu ảnh thô thành bản đồ đặc trưng đa tỷ lệ. Thứ hai, thành phần cổ đóng vai trò là giai đoạn xử lý trung gian, sử dụng các lớp chuyên biệt để tổng hợp và nâng cao biểu diễn đặc trưng trên các tỷ lệ khác nhau. Thứ ba, thành phần đầu hoạt động như một cơ chế dự đoán, tạo ra kết quả cuối cùng cho việc định vị và phân loại đối tượng dựa trên bản đồ đặc trưng đã được tinh chỉnh.

Dựa trên kiến trúc đã được thiết lập này, YOLO11 mở rộng và nâng cao nền tảng được đặt ra bởi YOLOv8, giới thiệu những cải tiến về kiến trúc và tối ưu hóa tham số để đạt được hiệu suất phát hiện vượt trội như minh họa trong.



Hình 2.6. Các mô-đun kiến trúc chính trong YOLO11.

a. Xương sống

Xương sống là một thành phần quan trọng của kiến trúc YOLO, chịu trách nhiệm trích xuất các đặc điểm từ ảnh đầu vào ở nhiều tỷ lệ khác nhau. Quá trình này bao gồm việc xếp chồng các lớp tích chập và các khối chuyên biệt để tạo ra các bản đồ đặc điểm ở nhiều độ phân giải khác nhau.

b. Các lớp tích chập

YOLO11 duy trì cấu trúc tương tự như các phiên bản tiền nhiệm, sử dụng các lớp tích chập ban đầu để giảm mẫu ảnh. Các lớp này tạo thành nền tảng của quy trình trích xuất đặc trưng, giảm dần kích thước không gian trong khi tăng số lượng kênh.

Một cải tiến đáng kể trong YOLO11 là việc giới thiệu khối C3k2, thay thế khối C2f được sử dụng trong các phiên bản trước. Khối C3k2 là một triển khai hiệu quả hơn về mặt tính toán của Nút Cổ Chai Phân Giai Đoạn Chéo (CSP). Khối này sử dụng hai phép tích chập nhỏ hơn thay vì một phép tích chập lớn, như được thấy trong YOLOv8. "k2" trong C3k2 biểu thị kích thước hạt nhân nhỏ hơn, góp phần xử lý nhanh hơn trong khi vẫn duy trì hiệu suất.

c. SPFF và C2PSA

YOLO11 vẫn giữ nguyên khối Spatial Pyramid Pooling - Fast (SPPF) từ các phiên bản trước nhưng giới thiệu một khối Cross Stage Partial with Spatial Attention (C2PSA) mới sau khi . Khối C2PSA là một bổ sung đáng chú ý giúp tăng cường khả năng tập trung không gian trong bản đồ đặc trưng. Cơ chế tập trung không gian này cho phép mô hình tập trung hiệu quả hơn vào các vùng quan trọng trong ảnh. Bằng cách gộp các đặc trưng theo không gian, khối C2PSA cho phép YOLO11 tập trung vào các khu vực quan tâm cụ thể, có khả năng cải thiện độ chính xác phát hiện đối với các vật thể có kích thước và vị trí khác nhau.

d. Cổ

Cổ kết hợp các đặc điểm ở các tỷ lệ khác nhau và truyền chúng đến đầu để dự đoán. Quá trình này thường bao gồm việc lấy mẫu lại và ghép nối các bản đồ đặc điểm từ các mức độ khác nhau, cho phép mô hình nắm bắt thông tin đa tỷ lệ một cách hiệu quả.

e. Khối C3k2

YOLO11 mang đến một thay đổi đáng kể bằng cách thay thế khối C2f ở cổ bằng khối C3k2. Khối C3k2 được thiết kế để nhanh hơn và hiệu quả hơn, nâng cao hiệu suất tổng thể của quy trình tổng hợp tính năng. Sau khi lấy mẫu và ghép nối, cổ trong YOLO11 tích hợp khối cải tiến này, mang lại tốc độ và hiệu suất được cải thiện.

f. Cơ chế chú ý

Một điểm bổ sung đáng chú ý cho YOLO11 là việc tăng cường tập trung vào sự chú ý không gian thông qua mô-đun C2PSA. Cơ chế chú ý này cho phép mô hình tập trung vào các vùng quan trọng trong ảnh, có khả năng dẫn đến khả năng phát hiện chính xác hơn, đặc biệt là đối với các vật thể nhỏ hơn hoặc bị che khuất một phần. Việc bổ sung C2PSA giúp YOLO11 khác biệt so với phiên bản tiền nhiệm YOLOv8, vốn thiếu cơ chế chú ý cụ thể này.

g. Cái đầu

Đầu của YOLO11 chịu trách nhiệm tạo ra các dự đoán cuối cùng về phát hiện và phân loại đối tượng. Nó xử lý các bản đồ đặc trưng được truyền từ đầu, cuối cùng đưa ra các hộp giới hạn và nhãn lớp cho các đối tượng trong ảnh.

h. Khối C3k2

Trong phần đầu, YOLO11 sử dụng nhiều khối C3k2 để xử lý và tinh chỉnh hiệu quả các bản đồ đặc trưng. Các khối C3k2 được đặt trên nhiều đường dẫn bên trong phần đầu, có chức năng xử lý các đặc trưng đa tỷ lệ ở các độ sâu khác nhau. Khối C3k2 thể hiện tính linh hoạt tùy thuộc vào giá trị của tham số $c3k$:

Khi $c3k = \text{False}$, mô-đun C3k2 hoạt động tương tự như khối C2f, sử dụng cấu trúc nút thắt tiêu chuẩn.

Khi $c3k = \text{True}$, cấu trúc nút thắt được thay thế bằng mô-đun C3, cho phép trích xuất tính năng sâu hơn và phức tạp hơn.

Các đặc điểm chính của khối C3k2

Xử lý nhanh hơn: Việc sử dụng hai phép tích chập nhỏ hơn giúp giảm chi phí tính toán so với một phép tích chập lớn, giúp trích xuất tính năng nhanh hơn.

Hiệu quả tham số: C3k2 là phiên bản nhỏ gọn hơn của nút thắt CSP, giúp kiến trúc hiệu quả hơn về số lượng tham số có thể đào tạo.

Một bổ sung đáng chú ý khác là khối C3k, mang lại tính linh hoạt cao hơn bằng cách cho phép tùy chỉnh kích thước hạt nhân. Khả năng thích ứng của C3k đặc biệt hữu ích trong việc trích xuất các đặc điểm chi tiết hơn từ hình ảnh, góp phần cải thiện độ chính xác phát hiện.

i. Khối CBS

Phần đầu của YOLO11 bao gồm một số CBS (Convolution-BatchNorm-Silu) các lớp sau khối C3k2. Các lớp này tinh chỉnh thêm bản đồ đặc trưng bằng cách:

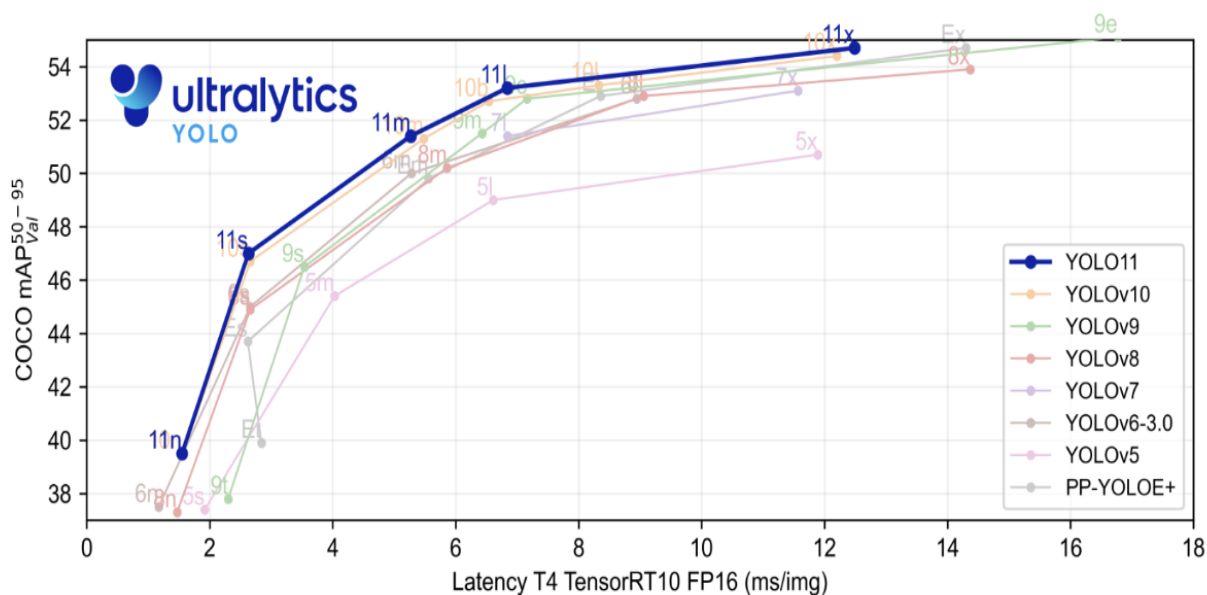
- Trích xuất các đặc điểm có liên quan để phát hiện đối tượng chính xác.
- Ổn định và chuẩn hóa luồng dữ liệu thông qua chuẩn hóa theo lô.
- Sử dụng hàm kích hoạt Đơn vị tuyến tính Sigmoid (SiLU) cho tính phi tuyến tính, giúp cải thiện hiệu suất mô hình.

Các khối CBS đóng vai trò là thành phần cơ bản trong cả quá trình trích xuất đặc điểm và phát hiện, đảm bảo rằng các bản đồ đặc điểm đã tinh chỉnh được chuyển đến các lớp tiếp theo để đưa ra hộp giới hạn và dự đoán phân loại.

j. Lớp tích chập cuối cùng và lớp phát hiện

Mỗi nhánh phát hiện kết thúc bằng một tập hợp các lớp Conv2D, giúp giảm các đặc trưng xuống số lượng đầu ra cần thiết cho tọa độ hộp giới hạn và dự đoán lớp. Lớp Phát hiện cuối cùng hợp nhất các dự đoán này, bao gồm:

- Tọa độ hộp giới hạn để xác định vị trí các đối tượng trong hình ảnh.
- Điểm số đối tượng cho biết sự hiện diện của đối tượng.
- Điểm lớp để xác định lớp của đối tượng được phát hiện.



Hình 2.7. So sánh YOLO11 với các phiên bản trước.

2.6. Lợi ích của YOLO11 cho bài toán

Trong bài toán phân loại hàng hóa, mục tiêu không chỉ dừng lại ở việc nhận biết sản phẩm thuộc nhóm nào mà còn phải xác định vị trí cụ thể của từng mặt hàng trong ảnh hoặc video. Do đó, việc sử dụng các mô hình chỉ phân loại (ví dụ: ResNet, MobileNet, EfficientNet) sẽ không tối ưu, bởi chúng yêu cầu ảnh đầu vào phải được cắt riêng từng sản phẩm trước khi đưa vào mô hình. Điều này gây mất thời gian và khó áp dụng trong môi trường thực tế, nơi hàng hóa thường nằm sát nhau, chồng lên nhau hoặc xuất hiện đồng thời trong khung hình.

Để giải quyết vấn đề này, đề tài sử dụng mô hình YOLO11 – một phiên bản cải tiến thuộc họ mô hình You Only Look Once (YOLO). Điểm mạnh của YOLO11 là khả năng phát hiện đối tượng và phân loại nhãn trong một lần suy luận duy nhất (one-stage detection). Nghĩa là mô hình vừa xác định tọa độ bounding box của từng sản phẩm, vừa gán nhãn chủng loại hàng hóa tương ứng ngay lập tức.

a. Kết hợp phát hiện và phân loại trong cùng một mô hình

- Thông thường, các mô hình phân loại truyền thống (như ResNet, EfficientNet, VGG,...) chỉ có khả năng phân loại ảnh đơn lẻ, yêu cầu sản phẩm phải được cắt riêng trước khi đưa vào nhận dạng. Tuy nhiên trong thực tế, hàng hóa thường:

- Xuất hiện nhiều đối tượng trong cùng một ảnh.
- Có thể chồng lên nhau hoặc sắp xếp sát nhau.
- Kích thước và góc chụp thay đổi liên tục.

- YOLO11 giải quyết vấn đề này bằng cơ chế “One-Stage Detection”: mô hình tự động phát hiện đối tượng và phân loại nhãn chỉ trong một lần suy luận, không cần bước cắt ảnh trung gian. Giúp giảm thời gian xử lý, giảm sai lệch trong cắt ảnh, Tối ưu hiệu suất khi triển khai thời gian thực.

b. Hiệu suất nhận dạng cao trong điều kiện môi trường phức tạp. Hàng hóa trong siêu thị, kho vận và dây chuyền sản xuất thường chịu ảnh hưởng của:

- Ánh sáng không ổn định.
- Phản chiếu bề mặt (bọc nilon, chai nhựa,...).
- Góc nhìn thay đổi liên tục.
- Nền phức tạp (kệ hàng nhiều màu sắc).

YOLO11 sử dụng backbone cải tiến có khả năng trích xuất đặc trưng tốt hơn, giúp mô hình:

- Nhận dạng sản phẩm nhỏ, bị che khuất một phần.
- Giảm nhầm lẫn giữa các mặt hàng có hình dạng hoặc bao bì tương tự.
- Giữ được độ chính xác cao ngay cả khi ảnh đầu vào không chuẩn hóa hoàn toàn.

c. Tốc độ xử lý nhanh, phù hợp triển khai thực tế

YOLOv11 được tối ưu hóa để đạt:

- Tốc độ suy luận nhanh hơn.

-Độ trễ thấp (low-latency) khi chạy trên GPU, CPU hoặc thiết bị Edge AI (Jetson, Raspberry Pi,...).

Điều này giúp mô hình đáp ứng tốt trong các ứng dụng:

Ứng dụng	Lợi ích khi dùng YOLOv11
Quét tự động tại quầy thanh toán	Không cần mã vạch, xử lý thời gian thực
Kiểm kê hàng hóa trong kho	Đếm nhanh và chính xác số lượng sản phẩm
Giám sát dây chuyền đóng gói	Nhận diện lỗi sai đóng gói hoặc nhầm sản phẩm
Bán lẻ thông minh	Tự động nhận diện mặt hàng trên kệ

d. Dễ dàng mở rộng và cập nhật tập dữ liệu

Khi xuất hiện mặt hàng mới, hệ thống chỉ cần:

- Thêm ảnh sản phẩm mới.
- Gán nhãn bounding box.
- Huấn luyện tiếp mô hình (fine-tuning).

Kết luận :

YOLO11 là lựa chọn phù hợp và tối ưu cho bài toán phân loại hàng hóa nhờ khả năng:

- Nhận diện và phân loại đồng thời.
- Duy trì độ chính xác cao trong môi trường thực tế.
- Xử lý nhanh, hỗ trợ triển khai thời gian thực.
- Dễ mở rộng khi số lượng loại hàng tăng.

Điều này đảm bảo tính khả thi, hiệu quả và bền vững khi triển khai hệ thống trong thực tế.

CHƯƠNG 3 : CÀI ĐẶT, THỬ NGHIỆM VÀ ĐÁNH GIÁ

3.1. Mở đầu

Trong số các phương pháp hiện đại, YOLO (You Only Look Once) nổi bật nhờ khả năng xử lý nhanh và độ chính xác cao. Phiên bản YOLOv11 được cải tiến với kiến trúc tối ưu hơn, giúp nhận diện và phân loại các loại hàng hóa ngay cả trong điều kiện phức tạp như ánh sáng thay đổi, nhiều vật thể chồng chéo hoặc đa dạng kiểu dáng bao bì. Việc ứng dụng YOLOv11 vào bài toán phân loại hàng hóa không chỉ giúp tăng tốc độ kiểm tra sản phẩm, mà còn hỗ trợ xây dựng các hệ thống thông minh như kiểm kê tự động, phân loại sản phẩm tại dây chuyền, hỗ trợ bán hàng tự động.

Chương này tập trung xây dựng mô hình phân loại hàng hóa dựa trên YOLO, tiến hành thu thập dữ liệu, huấn luyện mô hình và đánh giá kết quả. Mục tiêu là tạo ra một hệ thống có khả năng phân loại hàng hóa nhanh, chính xác và dễ dàng tích hợp vào các ứng dụng thực tế.

3.2. Môi trường thử nghiệm

3.2.1. Python

a. 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.

b. Đặc tính

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

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

2. Ngôn ngữ hướng đối tượng: Nó hỗ trợ các đặc trưng và kỹ thuật lập trình hướng đối tượng.

3. Ngôn ngữ lập trình tương tác: Người dùng có thể tương tác trực tiếp với trình thông dịch python để viết chương trình.

4. Ngôn ngữ dễ học: Python rất dễ học, đặc biệt là cho người mới bắt đầu.

5. 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.

6. Mã nguồn mở: Python cho phép người dùng có thể sử dụng, chỉnh sửa và phân phối mã nguồn một cách tự do.

7. Di động: Mã Python có thể chạy trên nhiều nền tảng phần cứng có cùng giao diện.

8. Có thể mở rộng: Người dùng có thể thêm các mô-đun cấp thấp vào trình thông dịch Python.

9. Có thể cải tiến: Python cung cấp một cấu trúc cải tiến để hỗ trợ các chương trình lớn sau đó là shell-script.

10. 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 cả 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.

c. Các phiên bản Python

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.



Hình 3.1. Ngôn ngữ Python.

3.2.2. Các đặc trưng Python

- 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.

3.3. Google Collab

3.3.1. 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.

3.3.2. 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.

- 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.

- 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 deep learning.

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.

- 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.

3.4. Các phiên bản VS Code

Khái niệm : Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn miễn phí, nhẹ và mạnh mẽ, phát triển bởi Microsoft, hoạt động trên đa nền tảng (Windows, macOS, Linux). Nó hỗ trợ nhiều ngôn ngữ lập trình và cung cấp các tính năng hữu ích như gỡ lỗi (debug), tô sáng cú pháp (syntax highlighting), tự động hoàn thành mã, và tích hợp Git. VS Code được ưa chuộng vì khả năng tùy biến cao và kho tiện ích mở rộng phong phú, phù hợp cho cả người mới bắt đầu và lập trình viên chuyên nghiệp. Nó là đa nền tảng, hoạt động trên Microsoft Windows, macOS và Linux.

Các đặc trưng :

Visual Studio Code là gì được rất nhiều người tìm hiểu. Đây cũng là một trong các ứng dụng được dân IT “săn đón” và tải về và sử dụng rất nhiều. Visual Studio Code cũng

luôn có những cải tiến và tạo ra đa dạng các tiện ích đi kèm từ đó giúp cho các lập trình viên sử dụng dễ dàng hơn. Trong đó có thể kể đến những ưu điểm sau:

1. Đa dạng ngôn ngữ lập trình giúp người dùng thỏa sức sáng tạo và sử dụng như HTML, CSS, JavaScript, C++,...
2. Ngôn ngữ, giao diện tối giản, thân thiện, giúp các lập trình viên dễ dàng định hình nội dung.
3. Các tiện ích mở rộng rất đa dạng và phong phú.
4. Hỗ trợ đa nền tảng: Linux, Mac, Windows,...
5. Hỗ trợ đa ngôn ngữ: C/C++, C#, F#, JavaScript, JSON, Visual Basic, HTML, CSS,...
6. Ít dung lượng.
7. Tính năng mạnh mẽ.
8. Intellisense chuyên nghiệp.
9. Giao diện thân thiện.
10. Kiến trúc mạnh mẽ và người dùng có thể khai thác mở rộng.
11. Số lượng người sử dụng lớn tạo nên ộng đồng hỗ trợ rộng rãi.



Hình 3.2. Phần mềm VS Code.

3.5. Androi studio

Android Studio là môi trường phát triển tích hợp (*Integrated Development Environment – IDE*) chính thức do Google phát hành, được sử dụng rộng rãi để phát triển các ứng dụng cho hệ điều hành Android. Android Studio được xây dựng dựa trên nền tảng IntelliJ IDEA – một IDE nổi tiếng của JetBrains – và được tối ưu hóa dành riêng cho việc phát triển ứng dụng di động Android.

3.5.1. Tổng quan về Android Studio

Android Studio được giới thiệu lần đầu tiên vào năm 2013 tại hội nghị Google I/O, thay thế cho công cụ Eclipse ADT (Android Development Tools) trước đây. Việc ra đời của Android Studio đánh dấu bước chuyển quan trọng trong phát triển ứng dụng Android, mang lại một môi trường làm việc chuyên nghiệp, hiện đại, có khả năng tự động hóa và tích hợp mạnh mẽ.

Công cụ này hỗ trợ lập trình viên trong toàn bộ quá trình phát triển phần mềm, từ thiết kế giao diện, viết mã nguồn, biên dịch, kiểm thử, đến triển khai ứng dụng lên thiết bị thật hoặc giả lập. Android Studio tương thích với các ngôn ngữ lập trình Java, Kotlin (ngôn ngữ chính thức của Google cho Android từ năm 2017) và C/C++ thông qua bộ công cụ Android NDK.

3.5.2. Cấu trúc và thành phần chính của Android Studio

Android Studio bao gồm nhiều thành phần quan trọng giúp tối ưu hóa quy trình phát triển ứng dụng:

a. Môi trường viết mã (Code Editor): Cung cấp khả năng hỗ trợ gợi ý cú pháp thông minh, tự động hoàn thiện mã, phân tích lỗi cú pháp theo thời gian thực và hỗ trợ refactor mã nguồn nhanh chóng. Trình soạn thảo có khả năng nhận diện cấu trúc code, hỗ trợ chuyển hướng dễ dàng giữa các lớp, hàm và tệp tin trong dự án.

b. Trình thiết kế giao diện (Layout Editor): Cho phép thiết kế giao diện ứng dụng một cách trực quan bằng thao tác kéo – thả (Drag and Drop) các thành phần UI

như nút bấm, hộp văn bản, hình ảnh, danh sách,... Người dùng có thể xem trước giao diện trên nhiều loại màn hình, kích thước và độ phân giải khác nhau mà không cần chạy ứng dụng.

c. Hệ thống biên dịch Gradle (Gradle Build System): Là công cụ tự động hóa việc biên dịch, đóng gói và quản lý thư viện trong dự án. Gradle giúp lập trình viên dễ dàng khai báo các thư viện phụ thuộc, cấu hình phiên bản SDK, thiết lập bản debug hoặc release chỉ bằng vài dòng lệnh trong file build.gradle.

d. Trình giả lập Android (Android Emulator): Cho phép chạy thử ứng dụng trên các thiết bị ảo (Virtual Devices) mà không cần điện thoại thật. Emulator mô phỏng đầy đủ phần cứng và phần mềm của thiết bị Android, giúp kiểm thử giao diện, hiệu năng và hành vi ứng dụng ở nhiều điều kiện khác nhau.

e. Trình quản lý thiết bị ảo (AVD Manager): Cho phép người dùng tạo và cấu hình các thiết bị ảo với thông số phần cứng khác nhau như độ phân giải màn hình, dung lượng RAM, phiên bản hệ điều hành Android, hay cấu trúc CPU (ARM, x86).

f. Android Profiler: Cung cấp bộ công cụ theo dõi hiệu năng ứng dụng (CPU, bộ nhớ, luồng xử lý, mạng, năng lượng tiêu thụ...) trong thời gian thực, giúp lập trình viên phát hiện các vấn đề gây chậm hoặc tốn pin.

g. Tích hợp Firebase và Google Play Services: Android Studio hỗ trợ kết nối trực tiếp với nền tảng Firebase để thêm các chức năng như xác thực người dùng (Firebase Auth), lưu trữ dữ liệu đám mây (Cloud Firestore), thông báo đẩy (Push Notifications) và phân tích hành vi người dùng (Analytics).

3.5.3. Ưu điểm của Android Studio

Android Studio mang lại nhiều lợi thế nổi bật so với các IDE khác:

- Giao diện trực quan, dễ sử dụng và thân thiện với người mới học.
- Tự động cập nhật SDK, API và thư viện hỗ trợ của Google.

- Hệ thống giả lập mạnh mẽ, tiết kiệm thời gian kiểm thử.
- Hỗ trợ tích hợp Git, GitHub, SVN, giúp quản lý mã nguồn hiệu quả.
- Cho phép phát triển ứng dụng đa nền tảng (điện thoại, máy tính bảng, TV, Wear OS, Auto).
- Khả năng tối ưu hóa hiệu năng, giảm thời gian biên dịch và hỗ trợ phát triển ứng dụng Android hiện đại theo kiến trúc MVVM hoặc Jetpack Compose.

3.5.4. Ứng dụng thực tế trong phát triển phần mềm

Trong các đề tài, Android Studio thường được sử dụng để:

- Xây dựng ứng dụng di động phục vụ học tập, quản lý, giám sát, hoặc nhận dạng đối tượng.
- Tích hợp với mô hình trí tuệ nhân tạo (AI), đặc biệt là mô hình YOLO trong nhận diện hình ảnh.
- Kết nối với các mô hình học sâu (*Deep Learning*) được huấn luyện trên Python để tạo ra ứng dụng thông minh chạy trên điện thoại Android.

3.5.5. Kết luận

Android Studio là công cụ phát triển ứng dụng Android toàn diện, mạnh mẽ và linh hoạt nhất hiện nay. Với khả năng tích hợp đa ngôn ngữ, hỗ trợ kiểm thử tự động, và khả năng tương thích với các nền tảng AI – IoT – Cloud, Android Studio đóng vai trò then chốt trong việc xây dựng các ứng dụng di động thông minh, đặc biệt trong bối cảnh chuyển đổi số và ứng dụng trí tuệ nhân tạo ngày càng phổ biến.

3.6. Tạo bộ dữ liệu học để huấn luyện

Tổng số lượng ảnh : 375

Bảng 3.1. Số lượng ảnh của từng loại

Hàng hóa	Số ảnh
Kẹo	100
Bánh	150
Nước ngọt	125

Các 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.

- 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.

3.7. Đào tạo mô hình phát hiện loại hàng hóa

Để huấn luyện mô hình “ Phát hiện loại hàng hóa “ 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 tập note book 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. Các thư viện hỗ trợ.

Sau khi cài xong các thư viện ta có thể bắt đầu train với đoạn mã sau

```
model=/content/drive/MyDrive/yolov11_project
/yolo11n.pt epochs=50 imgsz=640
```

Hình 3.4. Đoạn mã train.

Trong đó:

- Epochs: mô hình sẽ học qua toàn bộ dữ liệu 50 lần để tối ưu trọng số (số càng lớn dữ liệu train sẽ được cải thiện nhưng đổi lại thì sẽ tốn thời gian hơn).

- Imgsz: Kích thước đầu vào ảnh cho mô hình. Mỗi ảnh sẽ được resize về 640×640 pixel trước khi đưa vào huấn luyện. Đây là kích thước mặc định, giúp mô hình xử lý nhanh và đủ chính xác.

- Model: mô hình được Ultralytics cung cấp để đào tạo.

- Content: Đường dẫn tới file mô hình YOLO đã có sẵn (ở đây là YOLO11n.pt bản YOLO11 Nano). Đây là mô hình pretrained mà bạn dùng để huấn luyện tiếp (fine-tune) trên dữ liệu của mình.

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.

Bước tiếp theo là chuyển đổi từ file mô hình đã huấn luyện best.pt chạy trên desktop qua mô hình đã huấn luyện best.tflite để chạy trên android

Đầu tiên cài đặt thư viện Ultralytics

```
!pip install ultralytics

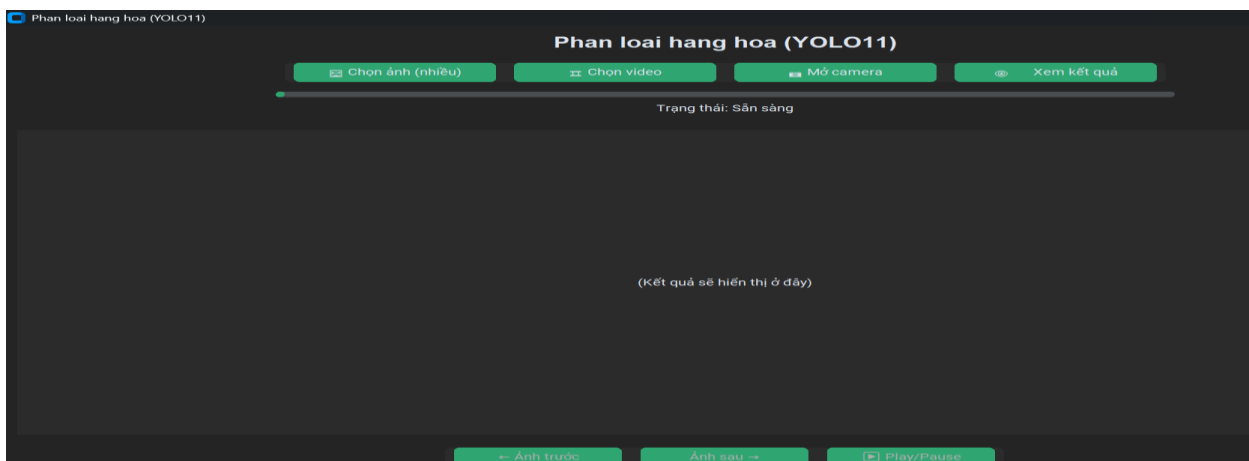
from ultralytics import YOLO

# Load mô hình PyTorch
model = YOLO("best.pt")

# Xuất sang định dạng TensorFlow Lite
model.export(format="tflite")
```

Hình 3.5. Code thư viện và chuyển đổi sang file tflite để chạy trên di động.

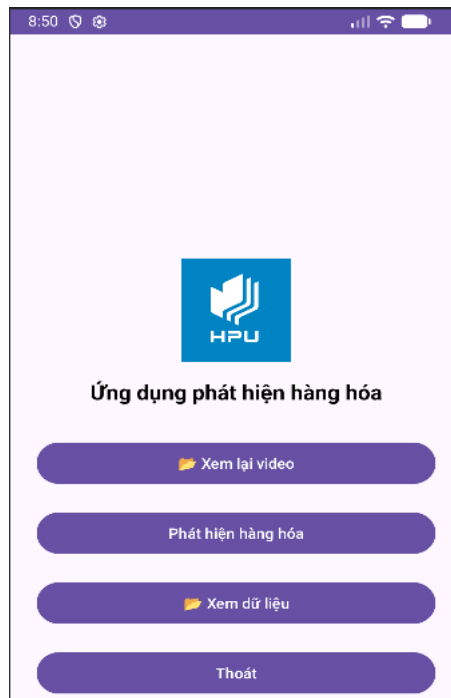
Bước tiếp theo chạy code chương trình phát hiện hoa quả hư hỏng trên desktop.



Hình 3.6. Giao diện chương trình phát hiện trên laptop.

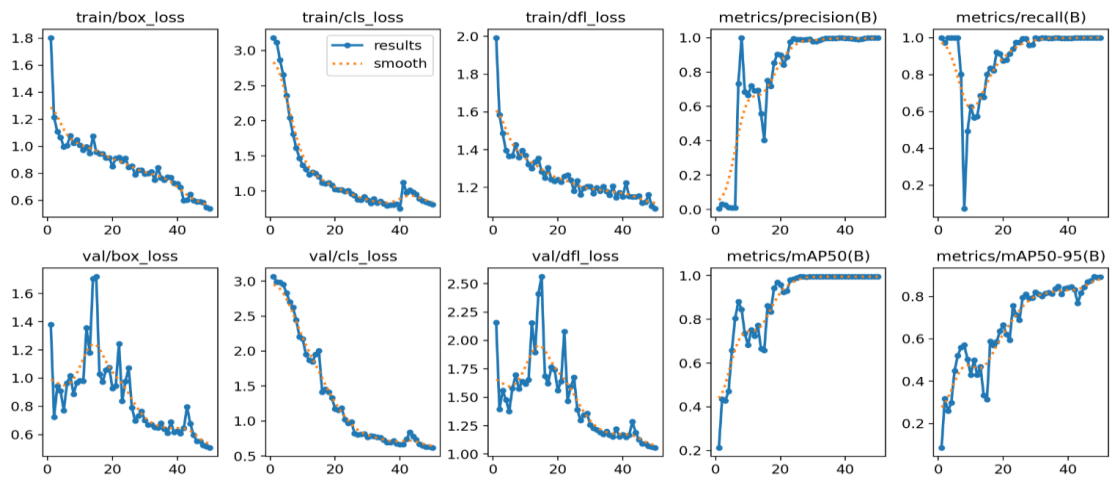
- Chọn ảnh (nhiều): Dùng để chọn nhiều hình ảnh từ máy tính và đưa vào hệ thống để mô hình xử lý.
- Chọn video: Cho phép chọn một tệp video để mô hình thực hiện nhận diện trên từng khung hình.
- Mở camera: Bật camera để nhận diện trái cây trực tiếp theo thời gian thực.
- Xem ảnh đã xử lý: Mở thư mục hoặc giao diện hiển thị các ảnh đã được mô hình phân tích.
- Xem video đã xử lý: Xem lại video đã được mô hình nhận diện (trong ảnh nút này đang bị mờ, có thể chưa kích hoạt).
- Dừng xử lý: Ngừng quá trình xử lý ảnh hoặc video đang chạy.
- Nút chuyển ảnh sang trái: Chuyển sang xem ảnh trước đó trong danh sách ảnh đã xử lý.
- Nút chuyển ảnh sang phải: Chuyển sang xem ảnh tiếp theo.

Tiếp đến là chạy chương trình phát hiện trái cây hỏng đã xây dựng trên android.



Hình 3.7. Giao diện trên android studio

3.8. Giá trị các biểu đồ



Hình 3.8. Biểu đồ thống kê sau thi đào tạo.

Theo như hình ta có thể thấy:

Biểu đồ trên thể hiện kết quả huấn luyện và kiểm định mô hình YOLO qua khoảng 50 epoch. Nhìn tổng thể, các chỉ số về loss (độ lỗi) và độ chính xác (precision,

recall, mAP) đều cho thấy xu hướng cải thiện rõ rệt và ổn định theo thời gian, chứng tỏ mô hình đã học được đặc trưng của dữ liệu và đạt được khả năng nhận diện tốt.

Các chỉ số precision, recall và mAP đều đạt giá trị cao :

- Precision và Recall đều tiến dần đến giá trị gần 1.0, chứng tỏ mô hình nhận diện chính xác hầu hết các đối tượng và rất ít bỏ sót.

-mAP@0.5 đạt xấp xỉ 1.0, thể hiện khả năng nhận diện chính xác tuyệt đối ở ngưỡng IoU = 0.5.

-mAP@0.5–0.95 đạt khoảng 0.85–0.9, phản ánh mô hình duy trì độ chính xác cao ngay cả khi tiêu chí so sánh trở nên khắt khe hơn.

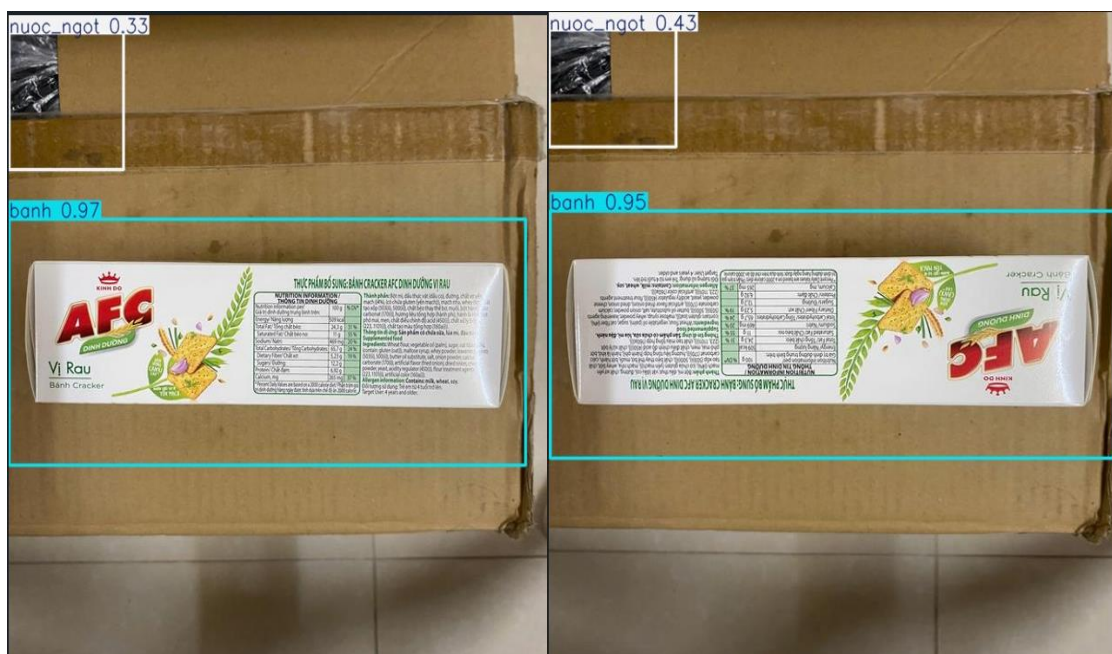
3.10. Kết quả thử nghiệm trên mô hình đã huấn luyện trên desktop

Đối với ảnh tĩnh 1 loại hàng hóa chương trình đem lại độ chính xác trong khoảng 90%.

Bảng 3.2. Thử nghiệm phân loại hình ảnh tĩnh

STT	Tên hàng hóa	Số lượng ảnh	Nhận đúng	Nhận sai	Độ chính xác
1	Kẹo	200	185	15	92%
2	Bánh	300	290	10	96.6%
3	Nước ngọt	250	225	20	90%

Theo như bảng trên nhận sai nhiều nhất nằm ở nước ngọt



Bảng 3.9. Nước ngọt bị nhận sai.

Tiếp theo là kẹo



Hình 3.10. Kẹo bị nhận diện sai.

3.11. Kết quả thử nghiệm mô hình đã huấn luyện trên android

Các chỉ số Precision và Recall đều đạt mức cao và ổn định sau khoảng 10–15 epoch, cho thấy mô hình phân biệt tốt giữa các đối tượng và phát hiện đầy đủ phần lớn các mẫu trong tập dữ liệu.

Tổng thể, mô hình YOLO sau huấn luyện đã đạt được hiệu năng tốt, độ chính xác cao và khả năng nhận dạng ổn định, đủ điều kiện để áp dụng vào giai đoạn kiểm thử và triển khai trong ứng dụng thực tế.

Đối với ảnh tĩnh 1 loại hàng chương trình đem lại độ chính xác trong khoảng 85%

Bảng 3.3. Thử nghiệm phân loại hình ảnh tĩnh

STT	Tên hàng hóa	Số lượng ảnh	Nhận đúng	Nhận sai	Độ chính xác
1	Kẹo	250	220	30	88%
2	Bánh	200	190	10	95%
2	Nước ngọt	300	245	55	81,6%

3.12. Đánh giá mô hình sau khi thử nghiệm

Mô hình YOLO11 đã được sử dụng để phân loại trái cây trên Google Colab, với bộ dữ liệu gồm khoảng 106 hình ảnh của 3 loại hàng hóa khác nhau. Mô hình đã được huấn luyện trong 50 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.

a. Kết quả huấn luyện

- Độ chính xác:

Dựa trên quá trình huấn luyện 50 epochs, mô hình đạt hiệu năng tốt và ổn định

Ở khoảng epoch thứ 40 trở đi, các chỉ số đánh giá đạt giá trị cao và gần như không còn biến động lớn.

- Kiểm tra thủ công trên bảng 3.5:

Khi tiến hành kiểm tra mô hình bằng các hình ảnh thực tế bên ngoài tập huấn luyện, mô hình cho kết quả nhận diện chính xác cao và ổn định.

Bộ ảnh kiểm tra có chất lượng tốt (rõ nét, đủ sáng) cho kết quả nhận diện rất chính xác, trong khi các ảnh chụp từ webcam hoặc điều kiện ánh sáng kém sẽ làm độ chính xác giảm nhẹ.

Điều này cho thấy chất lượng ảnh đầu vào có ảnh hưởng trực tiếp đến hiệu quả của mô hình, tuy nhiên mô hình vẫn chứng tỏ khả năng tổng quát hóa tốt trên dữ liệu thực tế.

b. Ưu điểm của mô hình:

- Hiệu suất và độ chính xác: YOLO11 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 hàng hóa.

c. 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, hoặc bị che khuất một phần.

d. 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 YOLOv12 mới ra mắt vào tháng 2/2025 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 YOLO11 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 3 tháng nghiên cứu và thực hiện đồ 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 đồ án với đề tài: Phát hiện loại hàng hóa

Một số mục tiêu mà đồ án đã đạt được:

Hoàn thành đề tài: "Phát hiện loại hàng hóa"

Đào tạo thành công mô hình nhận diện với 3 loại hàng hóa khác nhau: kẹo, bánh, nước ngọt bằng mô hình YOLO11

Mô hình phân loại được các loại hàng hóa với độ chính xác trung bình 75% với 50 epochs nhưng lại chính xác khá cao với ảnh tĩnh với một loại quả là 90% và vẫn còn nhiều sai sót với một số loại hàng hóa. 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 hàng hóa, 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 hàng hóa tại Việt Nam, một đất nước nổi tiếng với nhiều loại hàng hóa

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 đồ 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 đồ án của mình. Sinh viên xin trân trọng cảm ơn

TÀI LIỆU THAM KHẢO

1. Nguyễn Văn Vy, *Giáo trình Trí tuệ nhân tạo*. Hà Nội: NXB Khoa học và Kỹ thuật, 2023.
2. Phạm Văn Hùng, *Thị giác máy tính và ứng dụng*. Hà Nội: NXB Bách Khoa, 2022.
3. Ultralytics, “Tài liệu hướng dẫn sử dụng YOLOv8–v11,” *Trang chủ Ultralytics YOLO*, 2024. Có tại: <https://docs.ultralytics.com/vi/models/yolo11/>
4. Nguyễn Hữu Tùng, “Ứng dụng học sâu trong nhận diện khuôn mặt,” *Tạp chí Công nghệ Thông tin & Truyền thông*, 2024.
5. Python Software Foundation, “Python Documentation,” 2024. Có tại: <https://docs.python.org/3/>
6. Google Research, “Hướng dẫn sử dụng Google Colab,” *Google Research Documentation*, 2024. Có tại: <https://colab.research.google.com/>
7. Microsoft, “Visual Studio Code Documentation,” 2024. Có tại: <https://code.visualstudio.com/docs/>
8. Ultralytics, “YOLOv11 Model Release and Benchmark Report,” 2025. Có tại: <https://github.com/ultralytics/YOLOv11>
9. Roboflow, “Computer Vision Dataset Management,” 2024. Có tại: <https://roboflow.com/>
10. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016. Có tại: <https://www.deeplearningbook.org/>