

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: PHẠM ĐỨC TÀI

Giảng 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

ĐỀ TÀI:
ƯỚC LƯỢNG VẬN TỐC CỦA XE Ô TÔ DI
CHUYỂN TRÊN MỘT QUÃNG ĐƯỜNG QUA DỮ
LIỆU HÌNH ẢNH

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

Sinh viên: PHẠM ĐỨC TÀI

Giảng viên hướng dẫn: TS. HỒ THỊ HƯƠNG THƠM

HẢI PHÒNG, THÁNG 1 - 2025

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

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên: Phạm Đức Tài - **MSV:**1812111014

Lớp: CT2201M

Ngành: Công Nghệ Thông Tin

Tên đề tài: Ước lượng vận tốc của xe ô tô di chuyển trên một
quãng đường qua dữ liệu hình ảnh

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 ước lượng vận tốc ô tô qua dữ liệu hình ảnh, thiết kế xây dựng chương trình ước lượng vận tốc.

b) 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 ô tô để huấn luyện phát hiện đối tượng.
- Cài đặt mô hình phát hiện đối tượng ô tô bằng YOLOv8
- Thử nghiệm mô hình trên video để ước lượng vận tốc của ô tô.
- 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 ô tô bằng YOLOv8.
- Xây dựng được ứng dụng ước lượng vận tốc ô tô qua dữ liệu hình ảnh.
- 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 Giải Pháp Phần Mềm 5 Sao

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 ô tô để huấn luyện phát hiện đối tượng.
- Cài đặt mô hình phát hiện đối tượng ô tô bằng YOLOv8
- Thử nghiệm mô hình trên video để ước lượng vận tốc của ô tô.
- 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 ô tô bằng YOLOv8.
- Xây dựng được ứng dụng ước lượng vận tốc ô tô qua dữ liệu hình ảnh.
- Viết báo cáo đồ án tốt nghiệp.

Đề tài tốt nghiệp được giao ngày ... tháng 10 năm 2024

Yêu cầu phải hoàn thành xong trước ngày 11 Tháng 01 năm 2025

Đã nhận nhiệm vụ ĐTTN

Sinh viên

Đã giao nhiệm vụ ĐTTN

Giảng viên hướng dẫn

Phạm Đức Tài

TS. Hồ Thị Hương Thơm

Hải Phòng, ngày 11 tháng 01 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 Thơm

Đơn vị công tác: Trường Đại Học Hàng Hải

Họ và tên sinh viên: Phạm Đức Tài

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 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:**Chuyên ngành:**

Đề tài tốt nghiệp:

.....

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

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 “Ước lượng vận tốc của xe ô tô di chuyển trên một quãng đường qua dữ liệu hình ảnh” 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 2018 – 2022 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 11 tháng 1 năm 2025

Người Viết

Phạm Đức Tài

LỜI CAM ĐOAN

Sinh viên xin cam đoan đề tài “Ước lượng vận tốc của xe ô tô di chuyển trên một quãng đường qua dữ liệu hình ảnh” 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 để 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

Phạm Đức Tài

MỤC LỤC

LỜI CẢM ƠN.....	VI
LỜI CAM ĐOAN	VII
MỤC LỤC	VIII
DANH MỤC HÌNH ẢNH	X
DANH MỤC BẢNG	XI
BẢNG CÁC TỪ KHÓA VIẾT TẮT	XII
CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN.....	1
1.1.Đặt vấn đề	1
1.2.Phát hiện đối tượng.....	2
1.2.1. Khái niệm	2
1.2.2. Cách phát hiện đối tượng	4
1.2.3. Một số phương pháp phát hiện đối tượng phổ biến hiện nay.....	6
1.2.3.1. Phương pháp mô tả đặc trưng (HOG)	6
1.2.3.2. Mạng nơ-ron.....	7
CHƯƠNG 2. ƯỚC LƯỢNG VẬN TỐC CỦA XE Ô TÔ.....	12
2.1. Bài toán ước lượng vận tốc	12
2.2. Các loại phương tiện.....	13
2.2.1. Xe ô tô (Car).....	13
2.2.2. Xe tải (Truck)	14
2.2.3. Xe máy (Motorcycle).....	15
2.2.4. Xe khách (Bus)	15
2.3. Phát hiện đối tượng bằng YOLOv8.....	17
2.4. Thuật toán ước lượng vận tốc của đối tượng bằng YOLO.....	23
CHƯƠNG 3: CÀI ĐẶT, THỬ NGHIỆM VÀ ĐÁNH GIÁ.....	28
3.1. Môi trường	28
3.1.1. Python.....	28
3.1.1.1. Khái niệm	28

3.1.1.2. Đặc tính	28
3.1.1.3. Các phiên bản	29
3.1.2. Pycharm.....	30
3.1.2.1. Khái niệm	30
3.1.2.2. Các đặc trưng.....	31
3.1.2.3. Các phiên bản	31
3.1.3. Google Collab.....	32
3.1.3.1. Khái niệm	32
3.1.3.2. Những đặc trưng chính trong Google Colab	32
3.2. Dữ liệu thử nghiệm và cài đặt	33
3.2.1. Tạo bộ dữ liệu học để huấn luyện	33
3.2.2. Đào tạo mô hình phát hiện phương tiện	35
3.3. Kết quả 38	
3.3.1. Giao diện ứng dụng Ước lượng vận tốc	38
3.3.2. Thử nghiệm mô hình phát hiện đối tượng và ước lượng vận tốc ..	39
3.4. Đánh giá mô hình sau thử nghiệm.....	41
3.4.1. Kết quả huấn luyện	42
3.4.2. Thời gian huấn luyện	42
3.4.3. Đánh giá mô hình Ước lượng vận tốc	42
3.4.4. Ưu điểm của mô hình.....	42
3.4.5. Nhược điểm của mô hình:.....	43
3.4.6. Đề xuất cải tiến:	43
3.4.7. Hướng phát triển của đề tài	43
KẾT LUẬN.....	44
TÀI LIỆU THAM KHẢO	45

DANH MỤC HÌNH ẢNH

Hình 1.1 Tai nạn giao thông do chạy quá tốc độ.....	1
Hình 1.2: Giao thông trật tự khi được giám sát	2
Hình 1.3: Phát hiện đối tượng xe ô tô	3
Hình 1.4: Phát hiện đối tượng dựa trên học sâu cho các phương tiện.....	5
Hình 1.5: Kiến trúc hệ thống HOG để phát hiện đối tượng.[2].....	6
Hình 1.6: Mô hình R-CNN.[2].....	8
Hình 1.7: Faster R-CNN – Object Detection Algorithm. [2]	9
Hình 1.8: Mạng đề xuất vùng (RPN).[3].....	10
Hình 2.1: Hình dạng ô tô	13
Hình 2.2: Hình dạng xe tải.....	14
Hình 2.3: Hình dạng xe máy	15
Hình 2.4: Hình dạng xe khách.....	16
Hình 2.5: Kiến trúc YOLOv8 [6].....	19
Hình 2.6: Hình dung anchor box trong YOLO [6]	20
Hình 2.8: Đầu ra của YOLOv8 [6]	21
Hình 2.9: Hiệu suất trên tập dữ liệu phát hiện đối tượng MS COCO [11]	22
Hình 2.10: Sơ đồ thuật toán.....	23
Hình 3.1: Kết nối với drive, cài đặt thư viện và tải xuống tập dữ liệu huấn luyện	36
Hình 3.2: Tải mô hình yolov8n.pt để train	37
Hình 3.3: Đoạn mã để train	37
Hình 3.4: Giao diện chương trình	38
Hình 3.5: Kết quả xử lý video	41

DANH MỤC BẢNG

Bảng	Tên bảng	Trang
3.1	Số lượng ảnh từng loại	50
3.2	Bảng thử nghiệm với các epochs khác nhau	53
3.3	Tọa độ và hình ảnh video	55
3.4	Kết quả sau xử lý	56

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
RPN	Region Proposal Network	Mạng đề xuất vùng
NMS	Non-Maximum Suppression	Sự ức chế không tối đa
GPU	Graphics Processing Unit	Đơn vị xử lý đồ họa
HOG	Histogram of Oriented Gradient	Biểu đồ Gradient có hướng
IDE	Integrated Development Environ ment	Môi trường phát triển tích hợp
ID	Identifier	Định danh
TPU	Tensor Processing Unit	Đơn vị Xử lý Tensor
YOLO	You Only Look One	Bạn chỉ nhìn một lần
RPN	Region Proposal Network	Mạng đề xuất vùng

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN

1.1. Đặt vấn đề

Với sự phát triển nhanh chóng của công nghệ và trí tuệ nhân tạo, đặc biệt trong lĩnh vực thị giác máy tính, chúng ta đang chứng kiến những ứng dụng vượt bậc trong các lĩnh vực như nông nghiệp, công nghiệp, và giao thông vận tải. Khả năng này sẽ vào đóng vai trò quan trọng trong việc cải thiện an toàn giao thông.

Tai nạn giao thông là một trong những nguyên nhân hàng đầu gây thiệt hại về người và tài sản. Nguyên nhân phổ biến là do vi phạm tốc độ, thiếu ý thức, hoặc các yếu tố môi trường. Việc giám sát tốc độ xe cộ trên đường hiện nay thường dựa vào radar hoặc thiết bị cảm biến, nhưng những giải pháp này có chi phí cao và không phải lúc nào cũng triển khai hiệu quả trên diện rộng.



Hình 1.1 Tai nạn giao thông do chạy quá tốc độ

Trong bối cảnh này, ứng dụng thị giác máy tính để phân tích dữ liệu video và hình ảnh trở thành một giải pháp khả thi. Thuật toán nhận dạng và

theo dõi đối tượng, kết hợp với biến đổi phối cảnh và phân tích quỹ đạo, cho phép xác định vị trí và tính toán vận tốc của phương tiện trong thời gian thực. Điều này giúp:

- **Giám sát giao thông hiệu quả hơn:** Phát hiện vi phạm tốc độ, cảnh báo nguy cơ tai nạn tại các khu vực đông đúc.
- **Tiết kiệm chi phí:** Không yêu cầu lắp đặt nhiều thiết bị phần cứng đắt tiền.
- **Ứng dụng rộng rãi:** Có thể triển khai tại các nút giao thông, đoạn đường cao tốc, hoặc khu vực đô thị.



Hình 1.2: Giao thông trật tự khi được giám sát

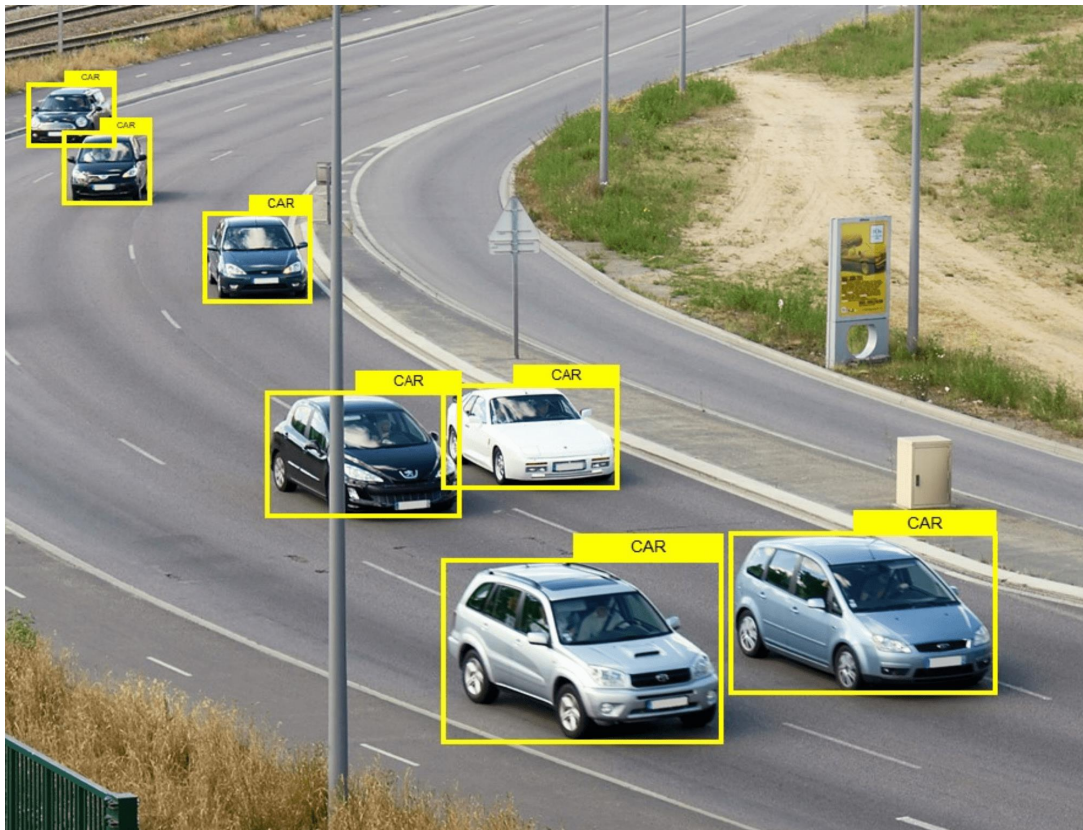
1.2. Phát hiện đối tượng

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 xe ô tô

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 trái cây, 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 phương tiện. [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.

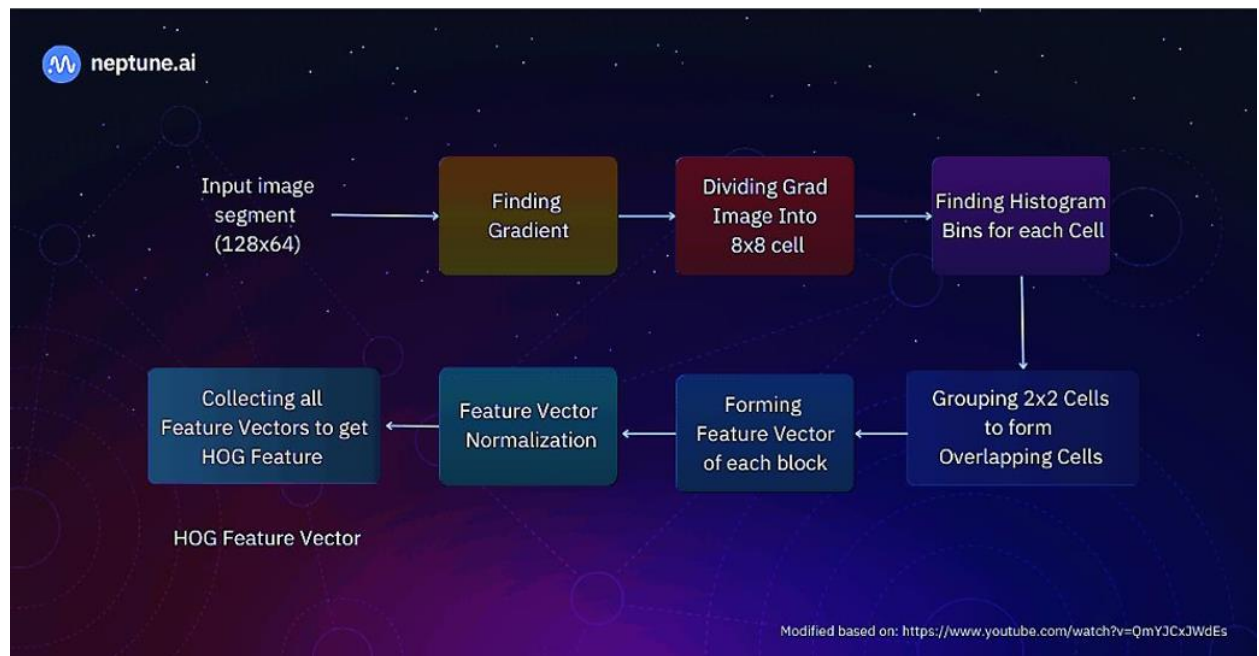
- Cần có năng lực tính toán cao, đặc biệt là GPU, để xử lý và đào tạo mô hình. Để giảm bớt khó khăn trong việc thu thập và gán nhãn dữ liệu, có thể sử dụng các bộ dữ liệu đã được gán nhãn sẵn như MS COCO, Caltech, KITTI, PAS CAL VOC, và V5. Những bộ dữ liệu này cung cấp một lượng lớn hình ảnh đã được gán nhãn, giúp rút ngắn quá trình thu thập và chuẩn bị dữ liệu cho việc đào tạo mô hình học sâu. [1]



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

1.2.3. Một số phương pháp phát hiện đối tượng 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.[2]

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. [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]

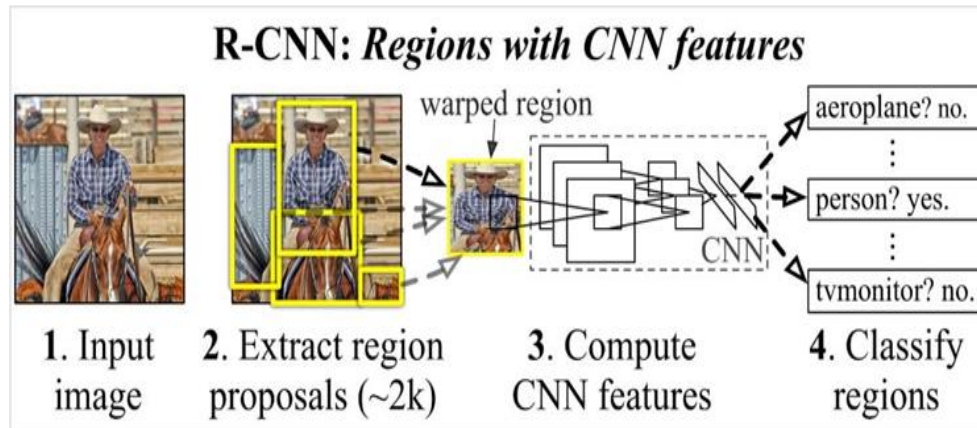
Trước khi chúng ta hiểu kiến trúc tổng thể của HOG, hãy xem cách nó hoạt động. Đối với một điểm ảnh cụ thể trong hình ảnh, biểu đồ tần suất của gradient được tính bằng cách xem xét các giá trị dọc và ngang để có được các vector đối tượng. Với sự trợ giúp của độ lớn gradient và các góc gradient, chúng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

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]

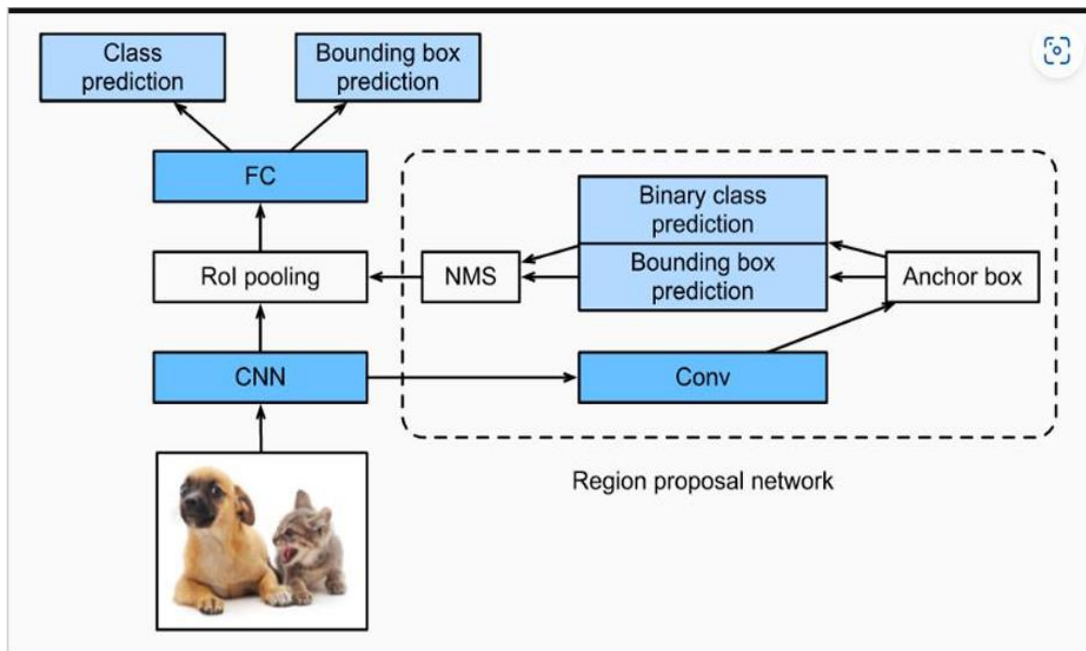
Quy trình làm việc của thuật toán tìm kiếm chọn lọc để chọn các đề xuất khu vực quan trọng nhất bao gồm việc tạo nhiều phân đoạn phụ trên một hình ảnh cụ thể và chọn các mục ứng cử viên cho nhiệm vụ của bạn. Thuật toán tham lam sau đó có thể được sử dụng để kết hợp các mục hiệu quả phù hợp cho một quy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

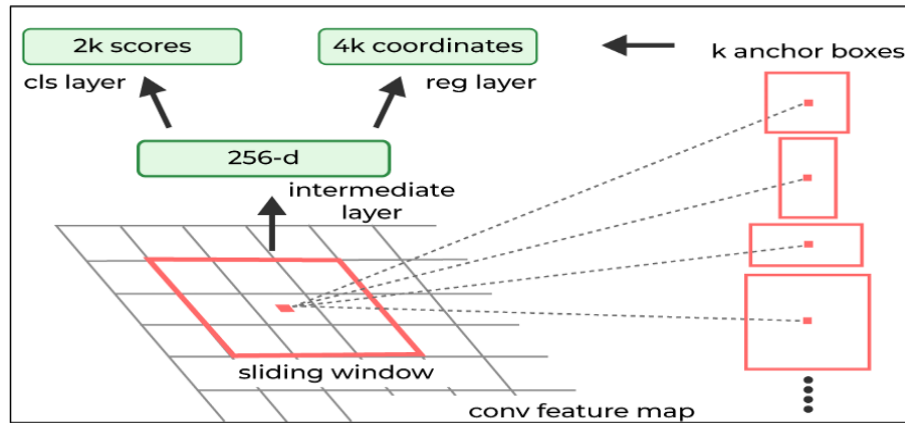
- **Faster R-CNN**

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 – Object Detection Algorithm. [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).[3]

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. [4]

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 YOLOv11.

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. [4]

Một số 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. [4]

YOLOv4: Bản cập nhật darknet gốc cho YOLOv3, được phát hành bởi Alexey Bochkovskiy vào năm 2020. [4]

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. [4]

YOLOv6: Được Meituan phát hành vào năm 2022 và được sử dụng trong nhiều robot giao hàng tự động của công ty. [4]

YOLOv7: Cập nhật YOLO mô hình được phát hành vào năm 2022 bởi các tác giả của YOLOv4. [4]

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. [4]

YOLOv9: Một mô hình thử nghiệm được đào tạo trên cơ sở mã nguồn mở YOLOv5 của Ultralytics. [4]

YOLOv11: Ra mắt vào tháng 10 năm 2024, YOLOv11 mang đến hiệu suất xử lý cao, hiệu quả và khả năng tương thích cho cả các thiết bị đám mây và biên. [4]

CHƯƠNG 2. ƯỚC LƯỢNG VẬN TỐC CỦA XE Ô TÔ

2.1. Bài toán ước lượng vận tốc

Với bài toán ước lượng vận tốc ô tô, bài toán cần giải quyết những vấn đề cơ bản nhất của quá trình giám sát giao thông bằng thị giác máy tính:

- **Ứng dụng mô hình học sâu hiệu quả trong nhận dạng và theo dõi phương tiện:** Để xác định vận tốc chính xác, cần nghiên cứu và triển khai các mô hình học sâu hiện đại, chẳng hạn như YOLO hoặc DeepSORT. Các mô hình này phải đủ mạnh để nhận diện phương tiện trong điều kiện thực tế, bao gồm ánh sáng thay đổi, mật độ giao thông cao, và sự đa dạng của các loại phương tiện.
- **Xây dựng và tổng hợp tập dữ liệu video giao thông đầy đủ:** Để mô hình hoạt động hiệu quả, cần có một tập dữ liệu video giao thông phong phú, bao gồm các tình huống giao thông đa dạng (giao lộ, đường cao tốc, khu dân cư), nhiều loại phương tiện và điều kiện môi trường khác nhau (mưa, nắng, ban đêm). Tập dữ liệu phải đảm bảo chất lượng cao để mô hình có thể học chính xác.
- **Thực hiện các thử nghiệm để đánh giá độ chính xác và hiệu quả:** Sau khi xây dựng mô hình và chuẩn bị dữ liệu, cần thực hiện các thử nghiệm thực tế. Những thử nghiệm này không chỉ đo độ chính xác trong nhận dạng mà còn đánh giá khả năng dự đoán vận tốc bằng cách so sánh với các giá trị thực tế. Các chỉ số cần xem xét bao gồm độ chính xác, độ trễ tính toán, và khả năng hoạt động trong các tình huống giao thông phức tạp.

Những yếu tố trên không chỉ đảm bảo bài toán được triển khai một cách hiệu quả mà còn góp phần xây dựng các hệ thống giao thông thông minh, hỗ trợ việc giảm thiểu tai nạn và tối ưu hóa lưu thông trên đường phố.

2.2. Các loại phương tiện

2.2.1. Xe ô tô (Car)

- **Khái niệm:** Xe ô tô là phương tiện giao thông cơ giới, có động cơ, được thiết kế để di chuyển trên các tuyến đường bộ. Xe ô tô có thể chở người hoặc hàng hóa và thường có 4 bánh. Đây là một trong những phương tiện phổ biến nhất dùng cho việc di chuyển cá nhân hoặc gia đình.
- **Đặc điểm:**
 - Có động cơ đốt trong hoặc động cơ điện.
 - Thường có sức chứa từ 4 đến 7 người, nhưng cũng có các loại xe lớn hơn như SUV hoặc minivan.
 - Có thể chạy với tốc độ cao, giúp di chuyển nhanh chóng qua các quãng đường dài.
 - Có các loại xe như sedan, hatchback, SUV, coupe, v.v.
- **Ứng dụng:** Di chuyển cá nhân, gia đình, vận chuyển hàng hóa nhẹ, công tác, du lịch [7]



Hình 2.1: Hình dạng ô tô

2.2.2. Xe tải (Truck)

- **Khái niệm:** Xe tải là loại xe ô tô có kích thước lớn, được thiết kế chủ yếu để vận chuyển hàng hóa. Xe tải có thể có nhiều loại với kích thước và tải trọng khác nhau, từ xe tải nhẹ cho đến xe tải nặng và xe đầu kéo.
- **Đặc điểm:**
 - Xe tải có thùng để chứa hàng hóa, có thể là thùng kín hoặc thùng mở.
 - Tải trọng và công suất động cơ của xe tải thường lớn, cho phép vận chuyển hàng hóa với số lượng lớn.
 - Có 4 bánh hoặc nhiều hơn tùy vào loại xe.
- **Ứng dụng:** Vận chuyển hàng hóa, vật liệu xây dựng, hàng hóa nông sản, giao nhận, v.v. [8]



Hình 2.2: Hình dạng xe tải

2.2.3. Xe máy (Motorcycle)

- **Khái niệm:** Xe máy là phương tiện giao thông cơ giới có hai bánh, sử dụng động cơ để di chuyển. Đây là phương tiện phổ biến ở nhiều quốc gia, đặc biệt là ở những khu vực có giao thông đông đúc, nơi việc sử dụng xe ô tô không thuận tiện.
- **Đặc điểm:**
 - Xe máy có kích thước nhỏ, gọn nhẹ, dễ dàng di chuyển qua các con phố chật hẹp hoặc trong giao thông đông đúc.
 - Thường có động cơ từ 50cc đến 1500cc hoặc hơn.
 - Xe máy có hai bánh.
- **Ứng dụng:** Di chuyển cá nhân, vận chuyển hàng hóa nhỏ, giao hàng, xe máy điện, phương tiện di chuyển trong các khu vực đô thị hoặc nông thôn. [9]



Hình 2.3: Hình dạng xe máy

2.2.4. Xe khách (Bus)

- **Khái niệm:** Xe khách là loại phương tiện giao thông công cộng hoặc chuyên dụng có sức chứa lớn, được thiết kế để vận chuyển hành khách

qua các tuyến đường dài hoặc ngắn. Xe khách có thể được sử dụng trong các dịch vụ vận tải công cộng hoặc cho các chuyến du lịch.

- **Đặc điểm:**

- Thường có sức chứa từ 20 đến 50 người hoặc hơn tùy vào loại xe. Xe khách có kích thước lớn và chiều dài dài hơn so với các phương tiện giao thông cá nhân khác. Nó thường có chiều dài từ 8 đến 12 mét (hoặc hơn, tùy thuộc vào loại xe).
- Được thiết kế với không gian thoải mái cho hành khách, có thể có ghế ngồi hoặc chỗ đứng. Là một khối hình chữ nhật lớn.
- Các loại xe khách có thể là xe buýt đô thị, xe khách đường dài, hoặc xe du lịch.

- **Ứng dụng:** Vận chuyển hành khách trong các tuyến giao thông công cộng, xe du lịch, xe đưa đón học sinh, nhân viên công ty. [10]



Hình 2.4: Hình dạng xe khách

2.3. Phát hiện đối tượng bằng YOLOv8

YOLOv8 Là một mô hình tiên tiến, hiện đại, YOLOv8 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. YOLOv8 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. [5]

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. [5]

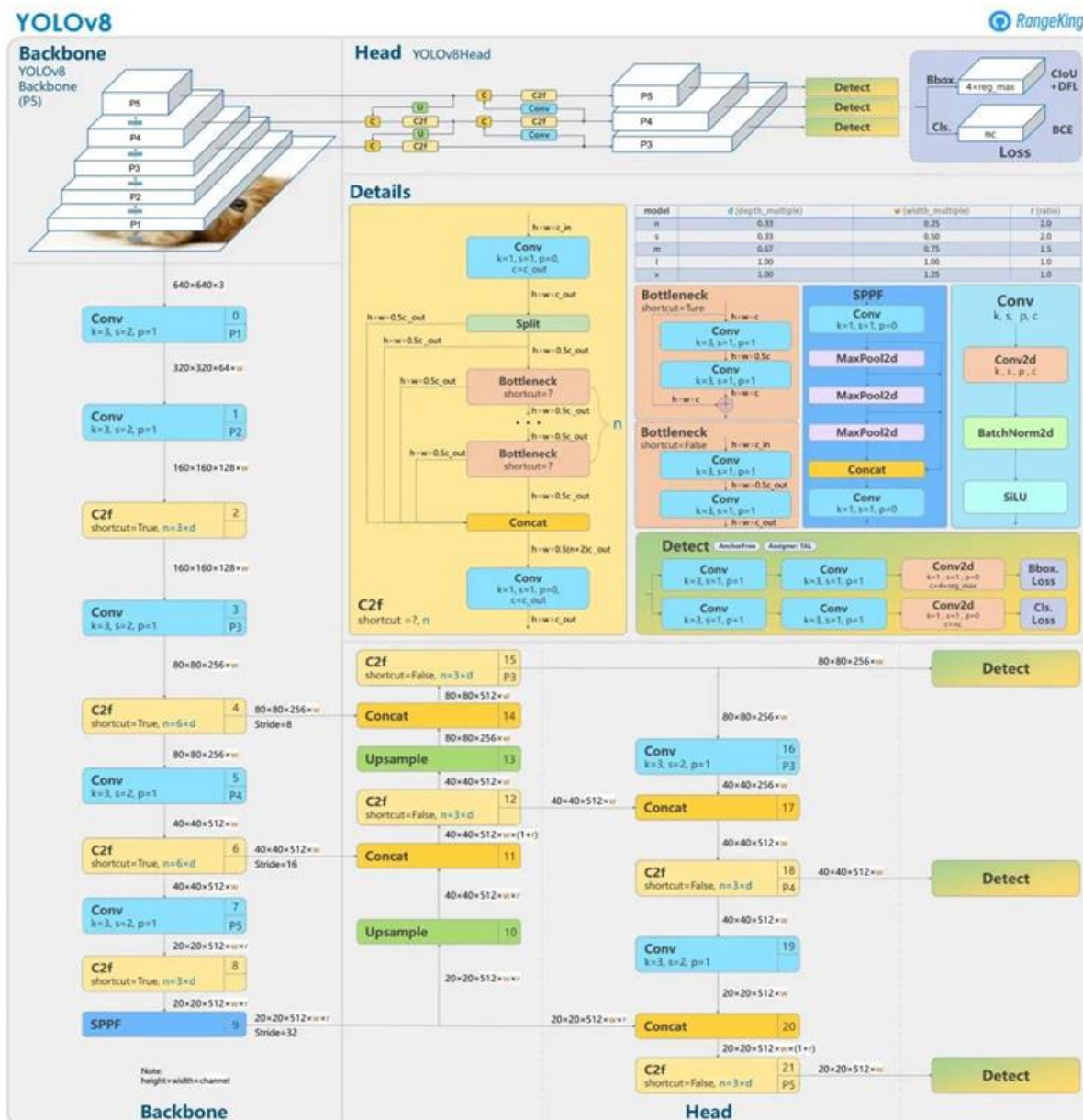
Cấu trúc của YOLOv8 được chia thành ba thành phần chính: xương sống, cổ và đầu. Xương sống chịu trách nhiệm trích xuất các đặc trưng từ hình ảnh đầu vào, với các tùy chọn như CSPDarknet53 và EfficientDet để phù hợp với nhu cầu cụ thể. Phần cổ kết nối xương sống với đầu, đóng vai trò quan trọng trong việc hợp nhất các đặc trưng. Phần đầu đảm nhận việc dự đoán các hộp giới hạn, lớp đối tượng và điểm tin cậy. [5]

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. [5]

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. [5]

Hơn nữa, YOLOv8 có hệ thống cấu trúc 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. [5]

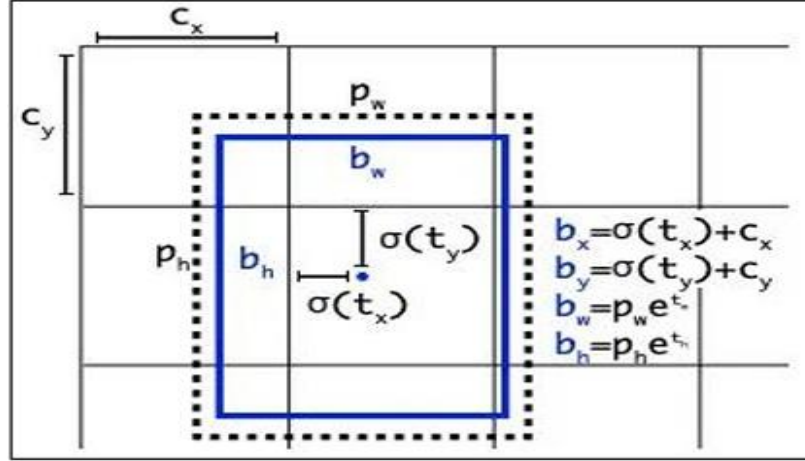
Tóm lại, kiến trúc của YOLOv8 nổi bật với thiết kế mô-đun, các biến thể có thể mở rộng, xương sống được cải thiện và các chiến lược đào tạo nâng cao. Những đặc điểm này cùng nhau góp phần vào sự thành công của YOLOv8 trong việc phát hiện đối tượng thời gian thực, khiến nó trở thành lựa chọn phổ biến cho các nhà nghiên cứu và chuyên gia trong lĩnh vực thị giác máy tính. [5]



Hình 2.5: Kiến trúc YOLOv8 [6]

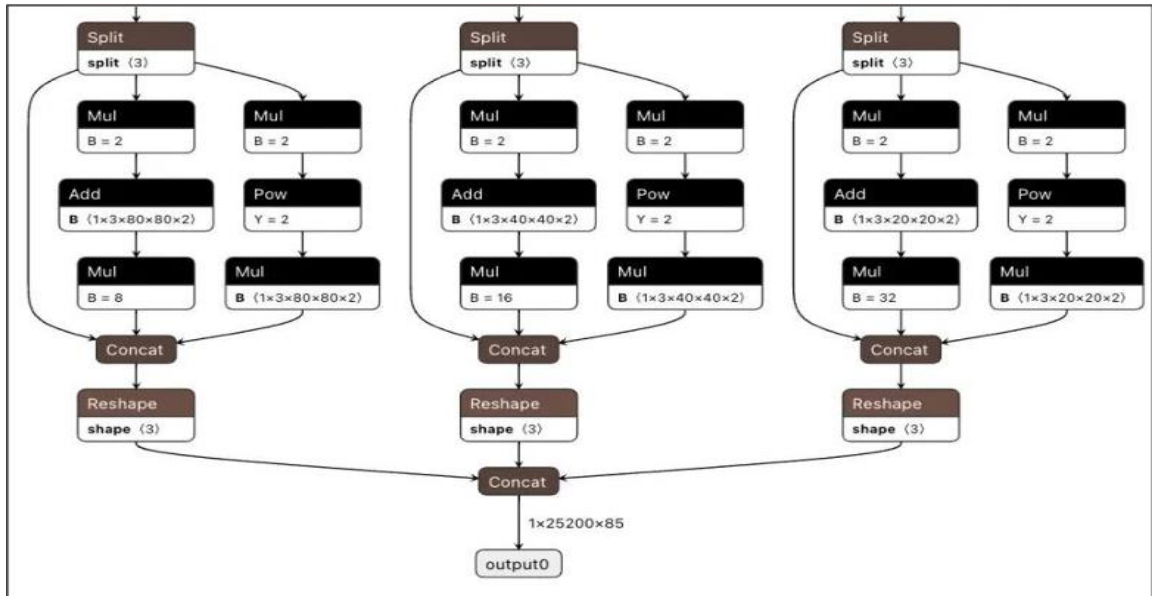
Phát hiện không neo

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.



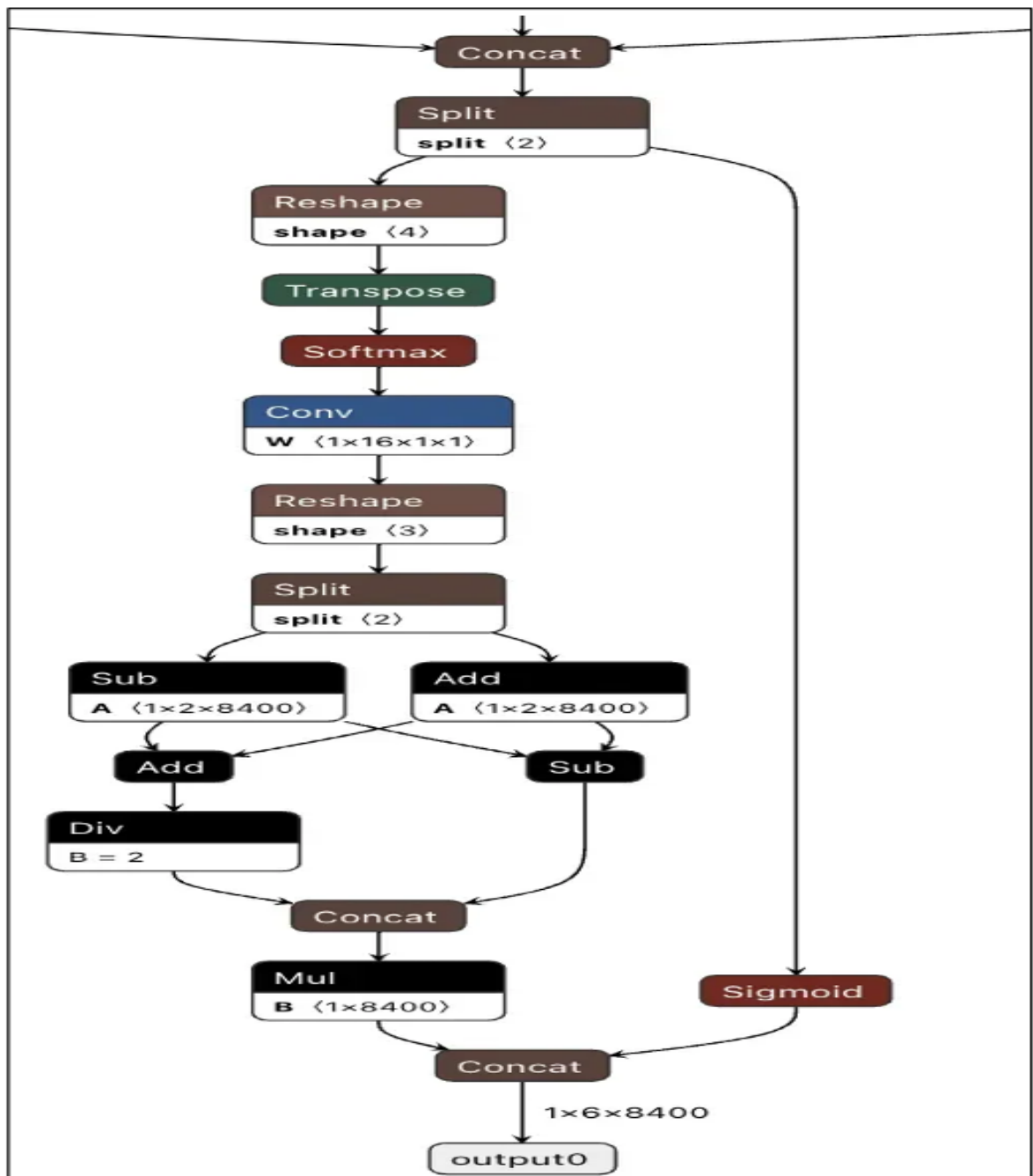
Hình 2.6: Hình dung anchor box trong YOLO [6]

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 2.7: Đầu ra của YOLOv5 [6]

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.

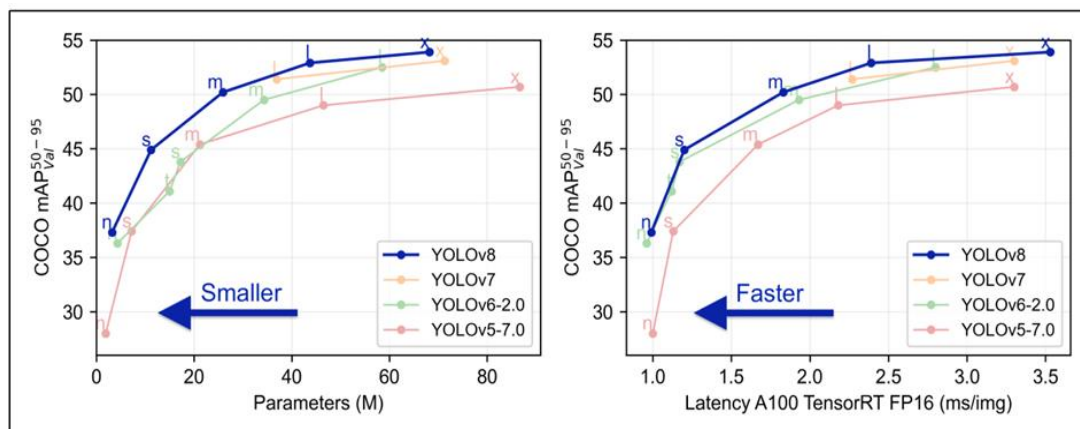


Hình 2.8: Đầu ra của YOLOv8 [6]

Các đặc trưng chính:

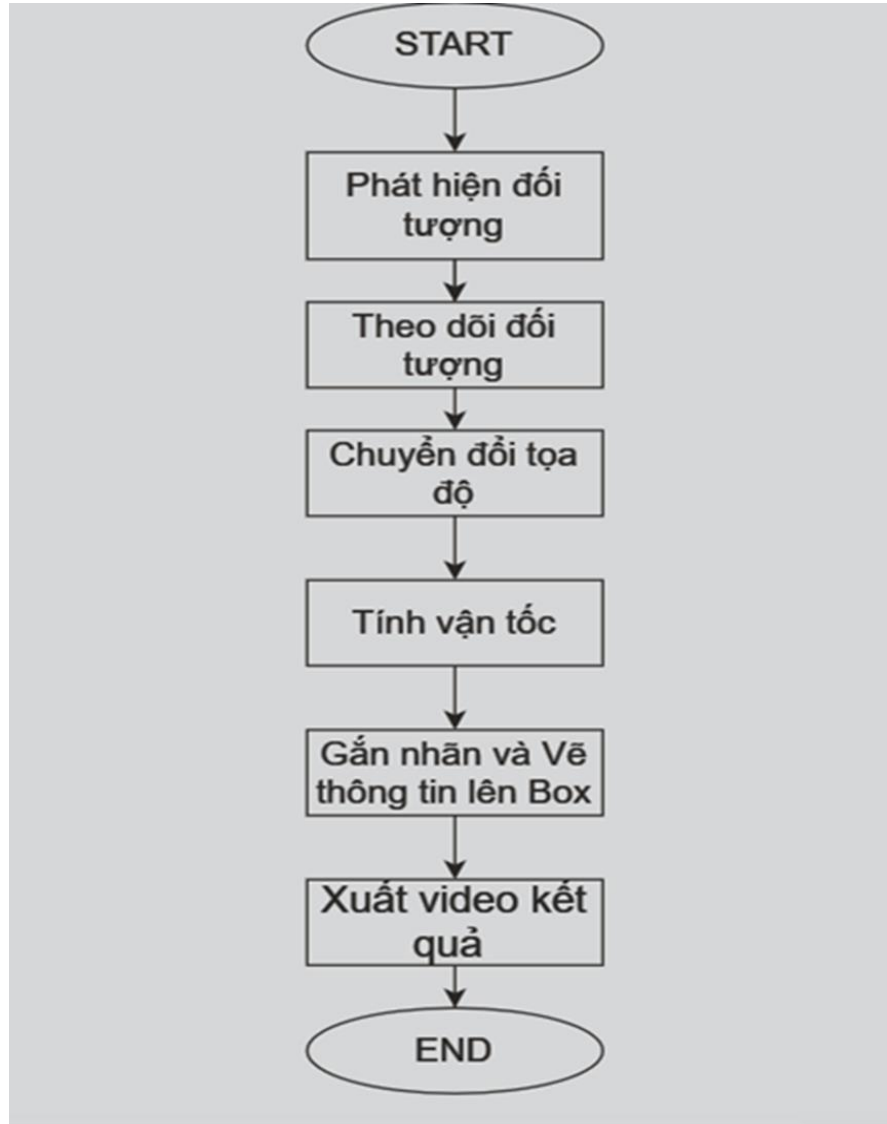
- Kiến trúc Backbone và Neck tiên tiến: YOLOv8 sử dụng các kiến trúc backbone và neck hiện đại, dẫn đến việc cải thiện khả năng trích xuất đặc trưng và hiệu suất phát hiện đối tượng.

- 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 2.9: Hiệu suất trên tập dữ liệu phát hiện đối tượng MS COCO [11]

2.4. Thuật toán ước lượng vận tốc của đối tượng bằng YOLO



Hình2.10: Sơ đồ thuật toán

Các bước của thuật toán:

- B1: Phát hiện đối tượng
- B2: Theo dõi đối tượng
- B3: Ước lượng vận tốc
- B4: Hiển thị kết quả vận tốc

Sau đây là phần trình bày chi tiết của các bước trong thuật toán.

B1: Phát hiện đối tượng

- Công cụ:
 - Sử dụng YOLOv8 bản nano (YOLOv8n) để phát hiện các đối tượng trong khung hình.
- Thực hiện:
 - Mỗi khung hình được đưa qua mô hình YOLO.
 - Bộ lọc **PolygonZone** đảm bảo chỉ giữ lại các đối tượng nằm trong khu vực được định nghĩa trước (SOURCE). Điều này giảm nhiễu từ các đối tượng bên ngoài khu vực quan tâm.
 - Loại bỏ trùng lặp: Sử dụng Non-Maximum Suppression (NMS) để giữ lại các hộp phát hiện tốt nhất cho cùng một đối tượng.
 - Kết quả trả về là các hộp giới hạn được gán ID.

B2: Theo dõi đối tượng

- Công cụ:
 - Sử dụng thuật toán ByteTrack để gán ID duy nhất cho mỗi đối tượng trong khung hình.
- Thực hiện:
 - Input: Nhận danh sách các hộp giới hạn và điểm neo của các đối tượng đã phát hiện.
 - ByteTrack gán ID theo dõi cho các đối tượng.
 - Nếu ByteTrack không tìm được đối tượng phù hợp trong khung hình tiếp theo, nó sẽ xem xét đối tượng đó đã "mất tích".
 - ByteTrack duy trì tracker_id (ID duy nhất) cho mỗi đối tượng, giúp theo dõi liên tục đối tượng qua nhiều khung hình.

B3: Ước lượng vận tốc

- Thu thập dữ liệu vị trí:

- Dữ liệu đầu vào:
 - Các tọa độ của đối tượng được phát hiện trong mỗi khung hình.
 - Tọa độ điểm neo của đối tượng được chọn (ở phần dưới hoặc giữa đối tượng) để xác định vị trí di chuyển theo thời gian.
- Lưu trữ lịch sử vị trí:
 - Lưu lại lịch sử vị trí (tọa độ y) của từng đối tượng để theo dõi dịch chuyển của đối tượng trong một khoảng thời gian.

- Biến đổi tọa độ

Dữ liệu vị trí từ không gian khung hình video được chuyển đổi về không gian thực tế thông qua một phép biến đổi phối cảnh (Perspective Transform):

- Nguồn (SOURCE): Tọa độ các điểm trong khung hình video (theo pixel).
- Mục tiêu (TARGET): Tọa độ các điểm trong không gian thực tế, với chiều dài và chiều cao xác định trước.
- Ma trận biến đổi: Sử dụng `cv2.getPerspectiveTransform()` để tạo ma trận M giúp chuyển đổi tọa độ.
 - Kết quả là tọa độ y của đối tượng phản ánh chính xác vị trí trong không gian thực tế (tính theo mét).
- Ước lượng khoảng cách dịch chuyển
 - Khoảng cách dịch chuyển theo trục y được ước lượng bằng công thức:

$$\text{Distance} = |y_{end} - y_{start}|$$
 - y_{end} : Giá trị tọa độ y tại thời điểm hiện tại.
 - y_{start} : Giá trị tọa độ y tại thời điểm trước đó (trong lịch sử lưu trữ).
- Ước lượng thời gian dịch chuyển

- Thời gian dịch chuyển được xác định dựa trên số lượng khung hình và FPS của video:

$$\text{time} = \frac{\text{len}(\text{coordinates})}{\text{FPS}}$$

-len(coordinates): Số khung hình mà đối tượng được theo dõi.

-FPS: Số khung hình mỗi giây của video.

- Ước lượng vận tốc
 - Vận tốc được ước lượng theo công thức:

$$\text{speed} = \frac{\text{distance}}{\text{time}} \times 3.6$$

-Kết quả ban đầu là vận tốc tính bằng m/s.

-Nhân 3.6 để chuyển đổi vận tốc sang km/h.

B4: Hiện thị kết quả vận tốc

- Vận tốc ước lượng được ở B3 sẽ được gán cho từng ID tương ứng và hiển thị lên các khung hình.

Chú thích:

- *Non-Maximum Suppression (NMS)*: là một kỹ thuật quan trọng trong xử lý ảnh và thị giác máy tính, đặc biệt trong các bài toán phát hiện đối tượng. Mục tiêu của NMS là loại bỏ các hộp giới hạn (bounding boxes) bị trùng lặp và chỉ giữ lại hộp tốt nhất đại diện cho một đối tượng, giúp hệ thống phát hiện hoạt động hiệu quả hơn và chính xác hơn.
- *ByteTrack*: là một thuật toán tiên tiến dùng để theo dõi đối tượng qua nhiều khung hình (multi-object tracking, MOT) trong các video. Nó được thiết kế để cải thiện độ chính xác và hiệu quả của việc gán ID duy nhất cho mỗi đối tượng trong các video, đảm bảo rằng đối tượng được theo dõi liên tục ngay cả khi gặp điều kiện khó khăn như mất dấu tạm thời hoặc chuyển động nhanh.

Ưu điểm:

- Công nghệ sử dụng: Sử dụng mô hình YOLOv8, một trong những mô hình nhận diện đối tượng hiện đại, với độ chính xác cao trong việc phát hiện đối tượng.
- Đo tốc độ phương tiện: Thuật toán sử dụng các phương pháp tính toán tốc độ dựa trên khoảng cách giữa các đường (zone) được định sẵn và thời gian di chuyển qua các vùng này. Tốc độ được tính toán rất chính xác vì có sự đo lường trực tiếp của khoảng cách và thời gian.
- Giám sát đa đối tượng: Thuật toán có thể theo dõi nhiều đối tượng trong cùng một lúc (từng xe ô tô, xe máy, v.v.) và có thể tính toán tốc độ của từng đối tượng.
- Áp dụng các phương pháp mới: Thuật toán sử dụng công cụ ByteTrack để theo dõi đối tượng, giúp duy trì theo dõi chính xác cho các đối tượng qua các khung hình.

Nhược điểm:

- Hiệu suất: Việc xử lý video có thể khá nặng, đặc biệt khi sử dụng mô hình YOLOv8 cho các video có độ phân giải cao. Điều này có thể gây tốn tài nguyên máy tính và thời gian xử lý lâu hơn.
- Cài đặt phức tạp: Thuật toán yêu cầu một số công cụ và thư viện như supervision và ultralytics, có thể cần một chút thời gian để cài đặt và cấu hình, đặc biệt cho người dùng chưa quen với các công cụ này.

CHƯƠNG 3: CÀI ĐẶT, THỬ NGHIỆM VÀ ĐÁNH GIÁ

3.1. Môi trường

3.1.1. 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. [16]

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. [12]

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 [16]

3.1.2. 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. [13]

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. [13]

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. [13]

3.1.3. 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. [14]

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. [15]

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 khối cho phép thực thi mã Python hoặc viết tiêu đề 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.

3.2. Dữ liệu thử nghiệm và cài đặt

3.2.1. 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.

Bộ dữ liệu gồm có 4483 ảnh về các phương tiện giao thông được thu thập qua các trang mạng xã hội: facebook, pinterest, pixel, ...và một số ảnh chụp trực tiếp từ bên ngoài, còn lại là ảnh trên còn lại được thêm vào thông qua đặc trưng tiền xử lý hay tăng cường hình ảnh trên Roboflow. [17]

Bảng 3.1: Số lượng ảnh từng loại

Phương tiện	Số ảnh
Car	676
Truck	3892
Bus	1278
Motorcycle	35

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 2804 ảnh chiếm 66%.

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 468 ảnh chiếm 11%.

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 989 ảnh chiếm 23%.

3.2.2. Đào tạo mô hình phát hiện phương tiện

Để huấn luyện mô hình “Phát hiện phương tiện” 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 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.

Các mô hình để huấn luyện của YOLOv8:

- YOLOv8n (Nano): Nhẹ nhất và nhanh nhất, dành cho thiết bị tài nguyên thấp như điện thoại hoặc IoT.
- YOLOv8s (Small): Nhẹ, nhanh, và có độ chính xác tốt, thích hợp cho các ứng dụng thời gian thực.
- YOLOv8m (Medium): Cân bằng giữa độ chính xác và tốc độ.
- YOLOv8l (Large): Độ chính xác cao hơn, phù hợp với các ứng dụng yêu cầu nhiều tài nguyên.
- YOLOv8x (Extra Large): Mạnh mẽ nhất, dành cho các hệ thống có tài nguyên lớn và yêu cầu độ chính xác cao.

```

0 giây
%cd /content/drive/MyDrive/ColabNotebooks/dulieu

/content/drive/MyDrive/ColabNotebooks/dulieu

!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="um3vg2Bu6kyQBsyvj5ob")
project = rf.workspace("aerial-image-object-detection")
project("vehicle-ywmkq-jo579-1lekt")
version = project.version(1)
dataset = version.download("yolov8")

Hiện kết quả đã ẩn

4 giây
[3] !pip install ultralytics

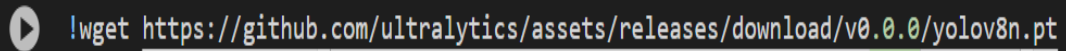
Hiện kết quả đã ẩn

0 giây
[4] %cd /content/drive/MyDrive/ColabNotebooks/yolov8

```

Hình3.1: Kết nối với drive, cài đặt thư viện và tải xuống tập dữ liệu huấn luyện

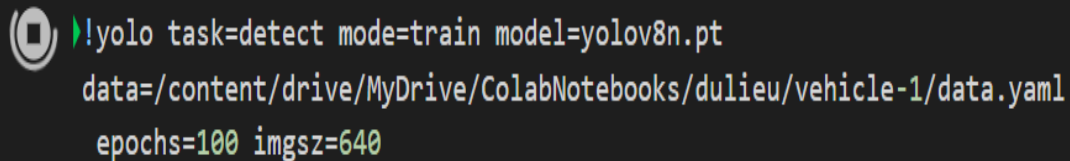
Vì Google Colab bản miễn phí chỉ cho giới hạn về thời gian chạy nên ta có thể liên kết với drive để lưu trữ để tránh mất dữ liệu sau khi hết phiên



```
!wget https://github.com/ultralytics/assets/releases/download/v0.0.0/yolov8n.pt
```

Hình 3.2: Tải mô hình yolov8n.pt để train

YOLOv8n là một phiên bản nhẹ của **YOLOv8**, được thiết kế để cân bằng giữa hiệu suất và tốc độ, đặc biệt phù hợp cho các ứng dụng yêu cầu tài nguyên thấp, chẳng hạn như chạy trên thiết bị di động hoặc các hệ thống có phần cứng hạn chế.



```
!yolo task=detect mode=train model=yolov8n.pt  
data=/content/drive/MyDrive/ColabNotebooks/dulieu/vehicle-1/data.yaml  
epochs=100 imgsz=640
```

Hình 3.3: Đoạn mã để train

Trong đó:

- **Epochs**: Số lần ảnh được train (số càng lớn dữ liệu train sẽ được cải thiện nhưng đổi lại thì sẽ tốn thời gian hơn)
- **Imgsz**: size ảnh (đối với YOLOv8 nhà phát triển có yêu cầu size ảnh là 640x640).
- **Model**: mô hình được Ultralytics cung cấp để đào tạo, ở đây là YOLOv8n.
- **Data**: đường dẫn đến tệp cấu hình tập dữ liệu (ví dụ: data.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.

Với từng epochs ta có bảng sau:

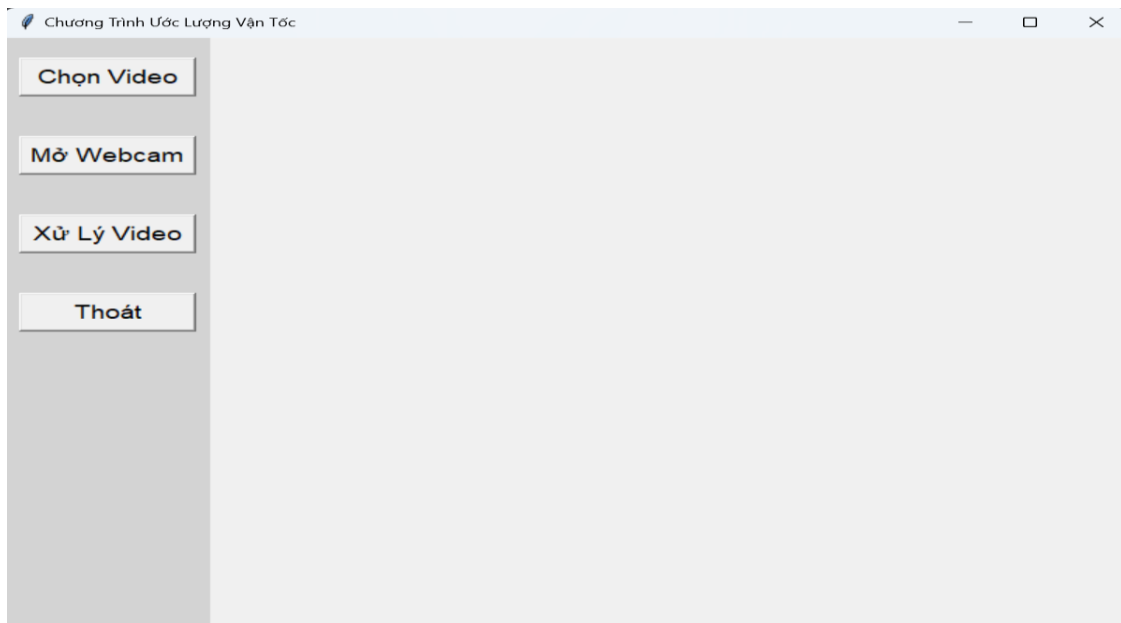
Bảng 3.2: Bảng thử nghiệm với các epochs khác nhau

Epochs	Ảnh kiểm thử	Thời gian	P	R	mAP	mAP ⁵⁰⁻⁹⁵
80	989	1h50'40s	0,689	0,664	0,641	0,433
90	989	2h3'20s	0,679	0,641	0,668	0,439
100	989	2h17'10s	0,730	0,678	0,693	0,449
110	989	2h59'25s	0,708	0,689	0,683	0,446

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 100 epochs ta được độ chính xác là 0.730 trong thời gian 2 tiếng 17 phút là tốt nhất. Từ đó ta có thể kết luận với dữ liệu tầm 4 nghìn ảnh đối với mô hình này thì 100 epochs là sự lựa chọn tốt nhất.

3.3. Kết quả

3.3.1. Giao diện ứng dụng Ước lượng vận tốc



Hình3.4: Giao diện chương trình

Giao diện có các chức năng cơ bản gồm:

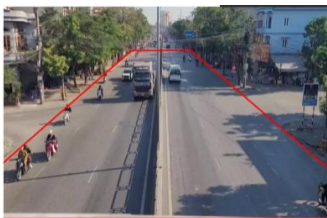
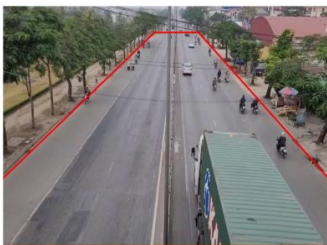
- Chọn video: Dùng để chọn video cần xử lý.
- Mở Webcam: Dùng webcam của máy để xử lý hình ảnh trực tiếp.
- Xử lý video: Dùng model đã train ở Google Colab để xử lý video được chọn sau đó xuất ra kết quả với định dạng mp4 và lưu vào 1 tệp chỉ định.
- Thoát: Thoát chương trình

3.3.2. Thử nghiệm mô hình phát hiện đối tượng và ước lượng vận tốc

Chúng ta tiến hành thử nghiệm với năm video khác nhau trong đó có ba video được tải từ trên mạng và hai video tự quay tại đường Nguyễn Văn Linh và đường Cầu Niệm. Với mỗi video khác nhau thì tọa độ của vùng quan tâm sẽ khác nhau nên ta có bảng 3.3.

Bảng 3.3: Tọa độ và hình ảnh video

Stt	Video	Tọa độ vùng quan tâm	Hình ảnh
1	Vehicles	SOURCE = np. Array ([[417, 262], [766, 268], [1680, 720], [-183, 720]])	
2	Highway	SOURCE = np. Array ([[733, 167], [480, 167], [-793, 720], [1853, 720]])	

Stt	Video	Tọa độ vùng quan tâm	Hình ảnh
3	Caotoc	SOURCE = np. Array ([[380, 20], [740, 20], [3080, 720], [-1080, 720]])	
4	Đường Nguyễn Văn Linh	SOURCE = np. Array ([[735, 150], [500, 150], [-250, 720], [1500, 720]])	
5	Đường Cầu Niệm	SOURCE = np. Array ([[750, 80], [580, 80], [-270, 720], [1480, 720]])	

Sau khi đã xác định được tọa độ vùng quan tâm của từng video thì video sẽ được xử lý chính xác. Kết quả thu được trình bày tại bảng 3.4.

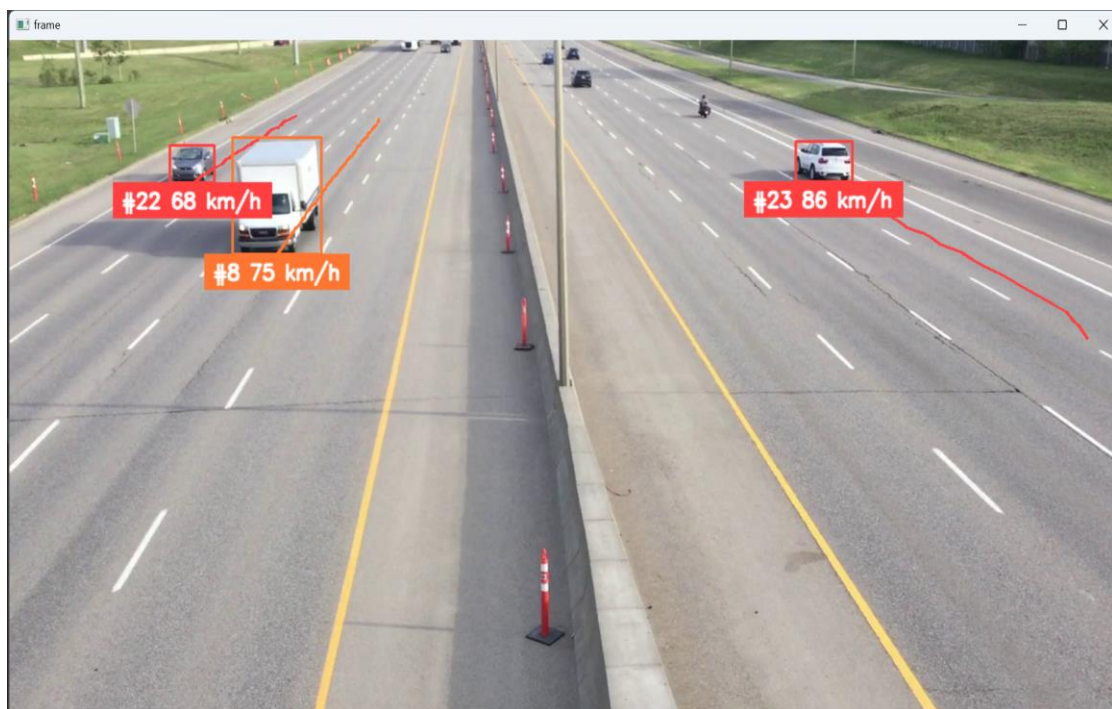
Bảng 3.4: Kết quả sau xử lý

Stt	Video	Số phương tiện	Số phương tiện đếm sai	Số phương tiện ước lượng sai vận tốc
1	Vehicles	30	2 (6,7%)	3 (10%)
2	Highway	174	40 (23%)	5 (2,8%)
3	Caotoc	110	3 (2,72%)	6 (5,4%)
4	Đường NVL	206	50 (24,2%)	7 (3,3%)

5	Đường Cầu Niệm	133	4 (3%)	8 (6%)
---	----------------	-----	--------	--------

Từ đó ta rút ra được kết luận. Khi chọn 1 video để xử lý mô hình đã đạt được những kết quả là:

- Phát hiện được đối tượng
- Ước lượng vận tốc của đối tượng
- Gắn nhãn cho đối tượng



Hình3.5: Kết quả xử lý video

Tuy nhiên vẫn còn một số hạn chế đó là:

- Khi đối tượng bị che khuất một phần thì kết quả ước lượng vận tốc sẽ không được chính xác.
- Nhận diện đối tượng kém khi nhiều đối tượng giao cắt nhau hoặc chất lượng hình ảnh kém.

3.4. Đánh giá mô hình sau thử nghiệm

Mô hình YOLOv8n đã được sử dụng để phát hiện và ước lượng vận tốc xe ô tô trên Google Colab, với bộ dữ liệu gồm khoảng hơn 4000 hình ảnh của 4 loại phương tiện. Mô hình đã được huấn luyện trong 110 epochs. 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 100, mô hình đạt độ chính xác cao nhất. Khi tăng hoặc giảm từ mốc epoch 100 thì các chỉ số đo độ chính xác ở bảng giảm.

3.4.2. Thời gian huấn luyện

- Với bộ dữ liệu 4000 ảnh và 110 epochs, thời gian huấn luyện kéo dài khoảng 5 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. Đánh giá mô hình Ước lượng vận tốc

Ta có công thức tính sai số trung bình (Mean Absolute Error - MAE) là:

$$\text{MEA} = \frac{\text{Tổng sai số}}{\text{Số lượng video}}$$

Suy ra sai số trung bình của năm video ở bảng 3.4 là:

$$\text{MEA} = \frac{10+2,8+5,4+3,3+6}{5} = \frac{27,5}{5} = 5,5\%$$

$$\Rightarrow \text{Độ tin cậy} = 100\% - 5,5\% = 94,5\%$$

Kết luận: Sai số trung bình là 5.5% đây là một mức sai số tương đối thấp và chấp nhận được trong nhiều ứng dụng thực tế.

3.4.4. 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 video ít bị che khuất và mật độ xe không quá cao.
- Khả năng xử lý: Hệ thống xử lý nhanh.

3.4.5. Nhược điểm của mô hình:

- Độ chính xác khi ước lượng vận tốc: Với những phương tiện ở rìa vùng quan tâm hoặc bị che khuất 1 phần vận tốc được ước lượng xuất hiện sai số lớn hoặc không được phát hiện.
- Sai sót trong nhận diện: Mô hình còn gặp khó khăn khi nhận diện các phương tiện bị che khuất một phần.

3.4.6. 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 YOLOv11 mới ra mắt vào tháng 9/2024 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. Tuy nhiên, vẫn cần cải thiện thêm để tăng độ chính xác và giảm sai sót trong quá trình nhận diện các phương tiện. 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.

3.4.7. Hướng phát triển của đề tài

Với những gì đã đạt được với trong đề tài có thể có những hướng phát triển sau:

- Phát hiện vi phạm giao thông: Tự động ghi nhận xe vượt quá tốc độ cho cơ quan chức năng.
- Hệ thống quản lý giao thông thông minh: Theo dõi mật độ và vận tốc phương tiện để tối ưu hóa dòng chảy giao thông.
- Ứng dụng trong logistic: Giám sát vận tốc xe tải trên đường nhằm đảm bảo an toàn và giảm chi phí vận hành.

KẾT LUẬN

Sau thời gian hơn ba 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: Ước lượng vận tốc ô tô di chuyển trên một quãng đường qua dữ liệu hình ảnh.

Một số mục tiêu mà đồ án đã đạt được:

- Hoàn thành đề tài “Ước lượng vận tốc ô tô di chuyển trên một quãng đường qua dữ liệu hình ảnh”.
- Đào tạo thành công mô hình nhận diện ô tô bằng mô hình YOLOv8.
- Mô hình nhận diện ô tô với độ chính xác tốt.
- Ước lượng được vận tốc xe ô tô di chuyển trên một quãng đường.

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 tích hợp vào các camera giao thông để quản lý giao thông tốt hơn. Góp phần cải thiện tình trạng giao thông phức tạp ở Việt Nam.

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 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]. 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 <https://neptune.ai/blog/object-detection-algorithms-and-libraries>
- [3]. Mạng đề xuất vùng (RPN) từ <https://medium.com/@codeplumber/region-proposal-network-rpn-backbone-of-faster-r-cnn-4a744a38d7f9>
- [4]. YOLO là gì? Các phiên bản của YOLO từ <https://blog.roboflow.com/guide-to-yolo-models/>
- [5]. Ultralytics - GitHub - ultralytics/ultralytics: NEW - YOLOv8 in PyTorch > ONNX > OpenVINO > CoreML > TFLite từ <https://github.com/ultralytics/ultralytics>
- [6]. What is YOLOv8? A Complete Guide từ <https://blog.roboflow.com/what-is-yolov8/>
- [7]. Xe ô tô – Wikipedia tiếng Việt từ [Ô tô – Wikipedia tiếng Việt](#)
- [8]. Xe tải – Wikipedia tiếng Việt từ [Xe tải – Wikipedia tiếng Việt](#)
- [9]. Xe máy – Wikipedia tiếng Việt từ [Xe máy – Wikipedia tiếng Việt](#)
- [10]. Xe khách – Wikipedia tiếng Việt từ [Xe khách – Wikipedia tiếng Việt](#)

[11]. So sánh hiệu suất của từng phiên bản YOLO được đào tạo trên tập dữ liệu MS COCO từ

https://www.researchgate.net/figure/Performance-comparison-of-each-YOLO-version-trained-on-the-MS-COCO-dataset-Compared-to_fig3_377321392

[12]. 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/>

[13]. PyCharm - Wikipedia từ

<https://en.wikipedia.org/wiki/PyCharm>

[14]. wu1503 - 01/06/2020 - [Google Colab Là Gì Và Dùng Để Làm Gì? | CodeLearn](#)

[15]. Google Colab là gì? Khám phá nền tảng sổ ghi chép tính toán trực tuyến từ <https://wikihow.com.vn/google-colab-la-gi/>

[16]. 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))

[17]. Bộ dữ liệu hình ảnh phương tiện từ

<https://universe.roboflow.com/aerial-image-object-detection/vehicle-ywmkq-jo579-llekt>