

TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP

Tên đề tài:

**HỆ THỐNG NHẬN DẠNG MẬT ĐỘ GIAO THÔNG
TỪ NGUỒN CAMERA CÔNG CỘNG**

SVTH : NGUYỄN TIẾN TRỌNG. 20TCLC_KHDL
NGUYỄN KIỀU QUỐC.
GVHD: PGS.TS. NGUYỄN TẤN KHÔI

Đà Nẵng, 2024

NHẬN XÉT ĐỒ ÁN TỐT NGHIỆP

I. Thông tin chung:

- Họ và tên sinh viên 1: NGUYỄN TIẾN TRỌNG Số thẻ SV: 102200291
- Họ và tên sinh viên 2: NGUYỄN KIỀU QUỐC Số thẻ SV: 102200282
- Tên đề tài: Hệ thống nhận dạng mật độ giao thông từ nguồn camera công cộng
- Người hướng dẫn: Nguyễn Tấn Khôi Học hàm/ học vị: Phó giáo sư, Tiến sĩ

II. Nhận xét, đánh giá đồ án tốt nghiệp:

- Về tính cấp thiết, tính mới, khả năng ứng dụng của đề tài: (điểm tối đa là 2đ)
.....
- Về kết quả giải quyết các nội dung nhiệm vụ yêu cầu của đồ án: (điểm tối đa là 4đ)
.....
- Về hình thức, cấu trúc, bố cục của đồ án tốt nghiệp: (điểm tối đa là 2đ)
.....
- Đề tài có giá trị khoa học/ có bài báo/ giải quyết vấn đề đặt ra của doanh nghiệp hoặc nhà trường: (điểm tối đa là 1đ)
.....
.....
- Các tồn tại, thiếu sót cần bổ sung, chỉnh sửa:
.....
.....

III. Tinh thần, thái độ làm việc của sinh viên: (điểm tối đa 1đ)

.....

IV. Đánh giá:

- Điểm đánh giá:/10 (lấy đến 1 số lẻ thập phân)
- Đề nghị: ☐ Được bảo vệ đồ án ☐ Bổ sung đề bảo vệ ☐ Không được bảo vệ

Đà Nẵng, ngày tháng năm 202
Người hướng dẫn

TÓM TẮT

Tên đề tài: Hệ thống nhận dạng mật độ giao thông từ nguồn camera công cộng.

Sinh viên 1 thực hiện: Nguyễn Tiến Trọng

Số thẻ SV: 102200291

Sinh viên 2 thực hiện: Nguyễn Kiều Quốc

Số thẻ SV: 102200282

Lớp: 20TCLC_KHDL

Trong quá trình điều khiển và quản lý giao thông, việc giám sát mật độ giao thông đóng vai trò quan trọng trong việc cung cấp thông tin kịp thời và chính xác, giúp giảm thiểu tắc nghẽn và tai nạn. Để giải quyết vấn đề này, chúng em đã xây dựng một hệ thống nhận dạng mật độ giao thông từ nguồn camera công cộng, nhằm hỗ trợ công tác điều hành giao thông hiệu quả hơn.

Để đạt được mục tiêu này, chúng em đã nghiên cứu và áp dụng các phương pháp và thuật toán nhận diện hình ảnh, kết hợp với các công nghệ tiên tiến như học sâu (deep learning) và xử lý ảnh số. Hệ thống này sử dụng dữ liệu từ các camera công cộng để phân tích và xác định mật độ giao thông tại các khu vực cụ thể, từ đó cung cấp thông tin chính xác và kịp thời cho các cơ quan quản lý giao thông.

Chúng em cũng đã xây dựng một cơ sở dữ liệu phong phú và đa dạng, chứa thông tin về các điều kiện giao thông thường gặp trong thực tế. Qua quá trình nghiên cứu và thử nghiệm, hệ thống đã cho thấy hiệu quả cao trong việc nhận dạng mật độ giao thông. Tuy nhiên, vẫn còn một số điểm cần cải thiện để hệ thống trở nên hoàn thiện hơn.

Trong thời gian tới, chúng em sẽ tiếp tục phát triển và hoàn thiện hệ thống, nhằm nâng cao độ chính xác và hiệu quả trong việc nhận dạng mật độ giao thông, góp phần vào việc quản lý giao thông thông minh và hiệu quả hơn.

ĐẠI HỌC ĐÀ NẴNG TRƯỜNG ĐẠI HỌC BÁCH KHOA KHOA CÔNG NGHỆ THÔNG TIN	CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM <u>Độc lập - Tự do - Hạnh phúc</u>
---	--

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên 1: Nguyễn Tiến Trọng Số thẻ sinh viên: 102200291

Họ tên sinh viên 2: Nguyễn Kiều Quốc Số thẻ sinh viên: 102200282

Lớp: 20TCLC_KHDL Khoa: Công nghệ thông tin

Ngành: Khoa học dữ liệu và trí tuệ nhân tạo

1. Tên đề tài đồ án:

Hệ thống nhận dạng mật độ giao thông từ nguồn camera công cộng

2. Đề tài thuộc diện: ☐ Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện

3. Các số liệu và dữ liệu ban đầu:

Dữ liệu hình ảnh và video thu thập từ các camera công cộng được lắp đặt tại các nút giao thông chính trong thành phố.

4. Nội dung các phần thuyết minh và tính toán:

Chương 1: Cơ sở lý thuyết

Chương 2: Phân tích thiết kế hệ thống

Chương 3: Kết quả thực nghiệm và đánh giá

5. Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ)

6. Họ tên người hướng dẫn: PGS. TS. Nguyễn Tấn Khôi

7. Ngày giao nhiệm vụ đồ án: 15/04/2024.

8. Ngày hoàn thành đồ án: 19/06/2024.

Đà Nẵng, ngày tháng năm 201

Trưởng Bộ môn

Người hướng dẫn

LỜI NÓI ĐẦU

Trong bối cảnh phát triển mạnh mẽ của đô thị hóa và tăng trưởng dân số, việc quản lý và điều tiết giao thông trở thành một thách thức lớn đối với các cơ quan chức năng. Mật độ giao thông ngày càng gia tăng đòi hỏi các biện pháp quản lý phải hiệu quả và kịp thời. Để hỗ trợ cho công tác này, việc ứng dụng công nghệ vào giám sát và phân tích giao thông đang trở thành một xu hướng tất yếu. Đồ án "Hệ thống nhận dạng mật độ giao thông từ nguồn camera công cộng" được thực hiện với mục tiêu nghiên cứu và phát triển một hệ thống có khả năng giám sát và nhận dạng mật độ giao thông, nhằm cung cấp thông tin chính xác và kịp thời cho việc điều hành giao thông.

Hệ thống này sử dụng các thuật toán nhận diện hình ảnh và công nghệ học sâu (deep learning) để phân tích dữ liệu từ các camera công cộng. Qua đó, chúng em hy vọng sẽ góp phần nâng cao hiệu quả quản lý giao thông, giảm thiểu tình trạng ùn tắc và tai nạn giao thông, đồng thời tạo ra một môi trường giao thông an toàn và thuận tiện hơn cho người dân.

Để hoàn thành đồ án này, chúng em đã nhận được sự giúp đỡ và hỗ trợ từ nhiều cá nhân và tổ chức. Chúng em xin bày tỏ lòng biết ơn sâu sắc đến Thầy Nguyễn Tấn Khôi, anh Nguyễn Minh Hoàng đã tận tình hướng dẫn, cung cấp kiến thức và kinh nghiệm quý báu, cũng như động viên chúng em trong suốt quá trình thực hiện đồ án.

Với những kết quả đạt được, hệ thống sẽ góp phần cải thiện tình hình giao thông và nâng cao chất lượng cuộc sống cho cộng đồng.

CAM ĐOAN

Chúng tôi xin cam đoan rằng toàn bộ nội dung của đồ án tốt nghiệp "Hệ thống nhận dạng mật độ giao thông từ nguồn camera công cộng" là do chính chúng tôi thực hiện dưới sự hướng dẫn của thầy Nguyễn Tấn Khôi và anh Nguyễn Minh Hoàng.

Trong quá trình thực hiện đồ án, chúng tôi đã nghiên cứu, thu thập và tổng hợp thông tin từ các nguồn tài liệu tham khảo, và đã trích dẫn đầy đủ các nguồn này theo quy định về liêm chính học thuật. Tất cả các số liệu, kết quả và nhận xét trong đồ án đều là trung thực, khách quan và chưa từng được sử dụng để bảo vệ một đồ án, luận văn hoặc công trình nghiên cứu nào khác.

Chúng tôi xin chịu trách nhiệm hoàn toàn về tính trung thực và chính xác của nội dung trong đồ án này.

Đà Nẵng, ngày 16 tháng 06 năm 2024

Sinh viên 1

Sinh viên 2

Nguyễn Tiến Trọng

Nguyễn Kiều Quốc

MỤC LỤC

TÓM TẮT.....	2
LỜI NÓI ĐẦU.....	1
CAM ĐOAN.....	2
DANH SÁCH CÁC HÌNH VẼ VÀ BẢNG	3
DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT.....	5
MỞ ĐẦU	6
Chương 1: CƠ SỞ LÝ THUYẾT	8
1.1. Thực trạng sản phẩm.....	8
1.2. Mật độ giao thông	9
1.2.1. Định nghĩa và đo lường.....	9
1.2.2. Yếu tố ảnh hưởng.....	9
1.3. Giới thiệu về YOLOV9	9
1.3.1. Kiến trúc mạng YOLOv9.....	9
1.3.2. Những cải tiến cốt lõi trong YOLOv9	12
1.4. Nhận dạng đối tượng	13
1.5. Theo dõi đối tượng.....	15
1.5.1. Giới thiệu.....	15
1.5.2. Các vấn đề trong Object Tracking	18
1.5.3. Simple Online Realtime Object Tracking.....	19
1.5.4. Bytetrack	23
1.6. Đếm đối tượng	25
Chương 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	27
2.1 Tổ chức chương trình.....	27
2.2 Lưu trữ và quản lý dữ liệu với PostgreSQL	28
Chương 3: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ	31
3.1. Các vấn đề cần giải quyết	31
3.1.1. Phân tích bài toán	31
3.1.2. Các vấn đề cần giải quyết	32
3.2. Đề xuất giải pháp tổng quan	32
3.3. Giải pháp.....	33
3.3.1. Sơ đồ tổng quan hệ thống.....	33
3.3.2. Giải pháp nhận diện phương tiện	34

3.4. Nhận diện phương tiện.....	34
3.4.1. Tập dữ liệu	34
3.4.2. Huấn luyện	37
3.4.3. Các công thức đánh giá kết quả	37
3.5. Kết quả mô hình.....	40
3.5.1. Kết quả định tính.....	40
3.5.2. Kết quả định lượng.....	41
3.6. Kết quả phần mềm	43
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	49
1. Đánh giá.....	49
1.1. Kết quả đạt được	49
1.2. Hạn chế.....	49
2. Hướng phát triển	49
TÀI LIỆU THAM KHẢO	50
PHỤ LỤC	1

DANH SÁCH CÁC HÌNH VẼ VÀ BẢNG

Hình 1: Kiến trúc mạng yolov9	11
Hình 2: Kết quả trực quan hóa của bản đồ tính năng đầu ra trọng số ban đầu ngẫu nhiên cho các kiến trúc mạng khác nhau: (a) Input image, (b) PlainNet, (c) ResNet, (d) CSPNet và (e) GELAN	13
Hình 3: Ví dụ về Object Detection	14
Hình 4: Single Object Tracking.....	16
Hình 5: Mutiple Object Tracking.....	16
Hình 6: Accuracy và Speed của SORT	19
Hình 7: Sơ đồ các bước xử lý của Kalman Filter.....	22
Hình 8: Bước 1 của ByteTrack.....	24
Hình 9: Bước 2 của ByteTrack.....	25
Hình 10: Ví dụ về Object Counting	26
Hình 11: Sơ đồ tổng quan hệ thống	27
Hình 12: Sơ đồ usecase của hệ thống	28
Hình 13: Sơ đồ ERD (Entity – Relationship Diagram).....	29
Hình 14: Sơ đồ huấn luyện	31
Hình 15: Sơ đồ nhận diện.....	31
Hình 16: Sơ đồ tổng quan hệ thống	33
Hình 17: Giải pháp nhận diện phương tiện giao thông	34
Hình 18: Gán nhãn trên Roboflow	35
Hình 19: Biểu đồ về số lượng nhãn có trong ảnh.....	36
Hình 20: Cấu trúc thư mục dataset.....	37
Hình 21: Intersection Over Union (IoU)	38
Hình 22: Confusion Matrix.....	39
Hình 23: Kết quả nhận diện trên ảnh	41
Hình 24: Kết quả train mô hình với yolov9.....	42
Hình 25: Ma trận nhầm lẫn của yolov9	43
Hình 26: Giao diện Dashboard.....	44
Hình 27: Giao diện Multimedia Detection: Detect Image	44
Hình 28: Giao diện Multimedia Detection: Detect Video	45
Hình 29: Giao diện Multimedia Detection: vẽ ROI.....	45
Hình 30: Cấu hình tham số đầu vào trang Multi Camera Monitor	46
Hình 31: camera giám sát số 1	46

Hình 32: camera giám sát số 2	47
Hình 33: Bản đồ hiển thị 2 camera giám sát và mật độ lưu thông tại đường Nguyễn Huệ	48

Bảng 1: Giải pháp tổng quan	32
Bảng 2: Thông tin tập dữ liệu	35
Bảng 3: Kết quả mô hình nhận diện phương tiện.....	41

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

Từ viết tắt	Từ diễn giải
YOLOv9	You Only Look Once version 9
PGI	Programmable Gradient Information
GELAN	Generalized Efficient Layer Aggregation Network
SSD	Single Shot MultiBox Detector
TP	True Positives
FP	False Positives
CNN	Convolutional Neural Network
FN	False Negative
GT	Ground Truth

MỞ ĐẦU

1. Mục đích thực hiện đề tài

Đề tài "Hệ thống nhận dạng mật độ giao thông từ nguồn camera công cộng" được thực hiện nhằm phát triển một hệ thống giám sát giao thông thông minh, sử dụng công nghệ nhận diện hình ảnh và học sâu (deep learning) để phân tích và đánh giá mật độ giao thông từ dữ liệu thu thập qua các camera công cộng. Mục đích chính của đề tài là hỗ trợ các cơ quan chức năng trong việc quản lý giao thông hiệu quả, giảm thiểu ùn tắc và tai nạn giao thông, và nâng cao chất lượng cuộc sống đô thị.

2. Mục tiêu đề tài

a. Xây dựng một hệ thống nhận dạng mật độ giao thông:

- Phân tích và xử lý dữ liệu hình ảnh từ các camera giao thông công cộng ở Đà Nẵng.
- Áp dụng các thuật toán học sâu để nhận diện và đánh giá mật độ giao thông.

b. Phát triển cơ sở dữ liệu giao thông:

- Tạo lập và quản lý một cơ sở dữ liệu đa dạng và phong phú, chứa thông tin về các tình huống giao thông thực tế.

c. Kiểm tra và đánh giá hiệu quả hệ thống:

- Thử nghiệm hệ thống trong các điều kiện giao thông khác nhau để đánh giá độ chính xác và hiệu quả của hệ thống.

3. Phạm vi và đối tượng nghiên cứu

a. Phạm vi nghiên cứu:

- Nghiên cứu các thuật toán nhận diện hình ảnh và học sâu (deep learning) ứng dụng trong lĩnh vực giao thông.
- Thu thập và xử lý dữ liệu từ các camera giao thông công cộng tại một số khu vực ở Đà Nẵng.
- Phát triển và thử nghiệm hệ thống trong môi trường giả lập và thực tế.

b. Đối tượng nghiên cứu:

- Các camera công cộng ở Đà Nẵng và dữ liệu hình ảnh thu thập từ các camera này.
- Các thuật toán và mô hình học sâu dùng để phân tích và nhận dạng mật độ giao thông.

4. Phương pháp nghiên cứu

a. Nghiên cứu lý thuyết:

- Tìm hiểu các khái niệm cơ bản về nhận diện hình ảnh, xử lý ảnh số và học sâu.
- Tham khảo các tài liệu, công trình nghiên cứu liên quan đến giám sát và quản lý giao thông.

b. Phân tích và thiết kế hệ thống:

- Phân tích yêu cầu, thiết kế kiến trúc hệ thống và các thành phần chính.
- Xây dựng mô hình và lựa chọn các thuật toán phù hợp.

c. Thực nghiệm và đánh giá:

- Thu thập dữ liệu từ các camera giao thông.
- Triển khai các thuật toán học sâu để phân tích dữ liệu.
- Thử nghiệm hệ thống trong các tình huống giao thông khác nhau và đánh giá hiệu quả.

5. Bố cục

Mở đầu

Chương 1: Cơ sở lý thuyết

Chương 2: Phân tích thiết kế hệ thống

Chương 3: Kết luận và hướng phát triển

Chương 1: CƠ SỞ LÝ THUYẾT

Trong chương này, trình bày tổng quan về thực trạng sản phẩm, các vấn đề cần giải quyết và giải pháp để nhận dạng mật độ giao thông. Bên cạnh đó, trình bày về kiến trúc lý thuyết và kiến trúc về mạng YOLOv9 và hướng tiếp cận bài toán.

1.1. Thực trạng sản phẩm

Hiện nay, các sản phẩm về nhận dạng mật độ giao thông từ nguồn camera công cộng đã đạt được những bước tiến đáng kể và được áp dụng trong nhiều ứng dụng thực tế. Nhiều hệ thống đã đạt được độ chính xác cao trong việc phân tích và nhận dạng mật độ giao thông thông qua sự kết hợp giữa các công nghệ học máy và học sâu, đồng thời cải thiện đáng kể về tốc độ xử lý trong thời gian thực. Những hệ thống này đáp ứng yêu cầu của các ứng dụng thực tế như quản lý và điều hành giao thông, hỗ trợ lái xe thông minh, và giám sát an ninh giao thông.

Tuy nhiên, mặc dù đã có sự phát triển tích cực, sản phẩm nhận dạng mật độ giao thông từ nguồn camera công cộng vẫn đối mặt với một số thách thức như:

- Điều kiện ánh sáng yếu và thời tiết khắc nghiệt: Hệ thống có thể gặp khó khăn trong việc nhận diện chính xác mật độ giao thông vào ban đêm hoặc trong điều kiện thời tiết xấu như mưa, sương mù.
- Góc quay và vị trí camera: Việc đặt camera ở các vị trí không tối ưu hoặc góc quay không phù hợp có thể làm giảm hiệu quả của việc nhận diện và phân tích giao thông.
- Đa dạng tình huống giao thông: Giao thông phức tạp với nhiều tình huống khác nhau, chẳng hạn như các sự kiện giao thông bất ngờ, công trường xây dựng hoặc các khu vực có mật độ giao thông cao đột ngột, cũng là những thách thức lớn.
- Chất lượng và độ phân giải hình ảnh: Nhiều camera công cộng hiện tại sử dụng công nghệ cũ, độ phân giải thấp, ảnh hưởng đến khả năng phân tích và nhận diện mật độ giao thông.
- Khả năng mở rộng và tính tương thích: Hệ thống cần khả năng mở rộng để tích hợp với các hệ thống hiện có và tương thích với nhiều loại camera khác nhau để đảm bảo tính linh hoạt và hiệu quả.

Những thách thức này cần được giải quyết để cải thiện hiệu suất và độ tin cậy của hệ thống nhận dạng mật độ giao thông từ nguồn camera công cộng. Việc áp dụng các công nghệ mới, tối ưu hóa thuật toán và cải thiện cơ sở hạ tầng là những hướng đi cần thiết để phát triển các sản phẩm này trong tương lai.

1.2. Mật độ giao thông

1.2.1. Định nghĩa và đo lường

- Mật độ giao thông thường được đo bằng số lượng phương tiện (xe hơi, xe máy, xe đạp, v.v.) trên một km hoặc một dặm đường. Đơn vị phổ biến là xe/km hoặc xe/mile.
- Mật độ giao thông cũng có thể được đo bằng số lượng phương tiện trên một đơn vị diện tích nhất định, chẳng hạn như số lượng xe trong một khu vực trung tâm thành phố.

1.2.2. Yếu tố ảnh hưởng

- Giờ cao điểm: Thời gian trong ngày khi mật độ giao thông tăng cao, thường là buổi sáng khi mọi người đi làm và buổi chiều khi tan tầm.
- Sự cố giao thông: Tai nạn, công trình xây dựng hoặc sự cố bất ngờ khác có thể làm tăng mật độ giao thông cục bộ.
- Cơ sở hạ tầng: Độ rộng và chất lượng của đường, số lượng làn đường, hệ thống đèn tín hiệu và các biện pháp quản lý giao thông khác.

1.3. Giới thiệu về YOLOv9

YOLOv9 (You Only Look Once version 9) là một mô hình học sâu tiên tiến được thiết kế cho việc nhận dạng đối tượng thời gian thực. Nó nhằm cải thiện các phiên bản trước về độ chính xác, hiệu quả tính toán và độ phức tạp của mô hình. [1]

1.3.1. Kiến trúc mạng YOLOv9

Kiến trúc mạng yolov9 gồm 3 phần là Backbone, Neck, và Head. [2]

a. Backbone

Backbone trong YOLOv9 chịu trách nhiệm trích xuất các đặc trưng (features) từ ảnh đầu vào. Nó thực hiện các tác vụ sau:

- **Trích xuất đặc trưng cốt lõi:** Backbone sử dụng các lớp convolutional để trích xuất các đặc trưng quan trọng từ ảnh đầu vào. Các lớp này dần dần giảm kích thước không gian của ảnh, trong khi tăng chiều sâu của các đặc trưng (feature maps).
- **Giảm kích thước:** Sử dụng các lớp convolutional và pooling (ADown) để giảm kích thước không gian của các feature maps, giúp giảm số lượng tính toán và tăng cường khả năng trừu tượng hóa.
- **Kết hợp các đặc trưng:** Các lớp như RepNCSPelan4 giúp kết hợp và trừu tượng hóa các đặc trưng từ các tầng khác nhau.

b. Neck

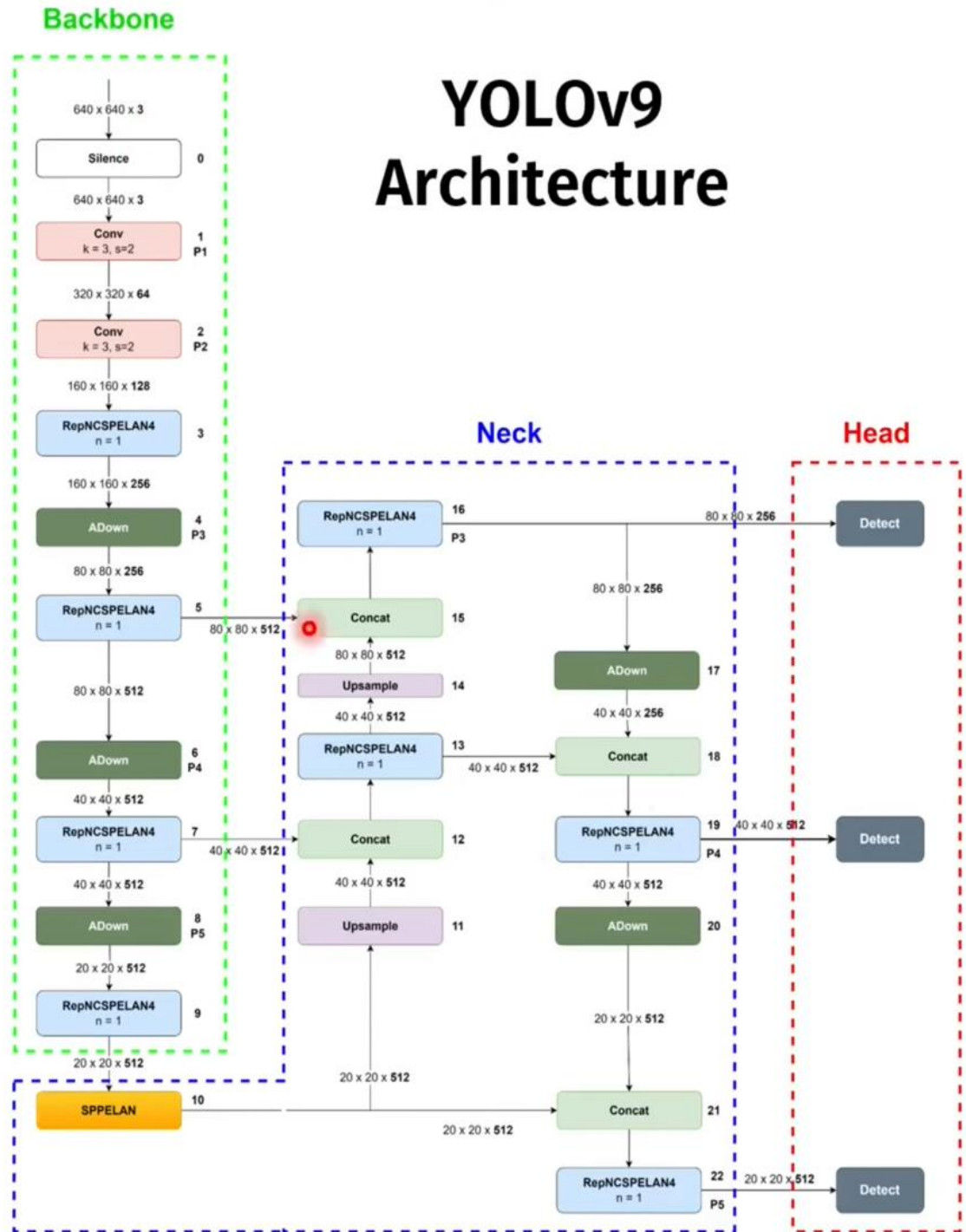
Neck trong YOLOv9 chịu trách nhiệm xử lý và kết hợp các đặc trưng từ Backbone để chuẩn bị cho quá trình phát hiện đối tượng. Nó thực hiện các tác vụ sau:

- **Tăng kích thước (Upsampling):** Sử dụng các lớp upsampling để tăng kích thước không gian của các feature maps, giúp giữ lại các đặc trưng quan trọng từ các tầng sâu hơn.
- **Kết hợp đặc trưng (Concat):** Kết hợp các feature maps từ các tầng khác nhau của Backbone, giúp tổng hợp thông tin từ các mức độ trừu tượng khác nhau.
- **Tạo ra các feature maps đa quy mô:** Sử dụng các lớp như SPP và các lớp convolutional khác để tạo ra các feature maps có kích thước khác nhau, giúp phát hiện đối tượng ở các kích thước khác nhau.

c. Head

Head trong YOLOv9 chịu trách nhiệm dự đoán các bounding boxes và các lớp (class) của các đối tượng trong ảnh. Nó thực hiện các tác vụ sau:

- **Phát hiện đối tượng (Detection):** Sử dụng các lớp convolutional cuối cùng để dự đoán các bounding boxes và xác suất lớp cho mỗi đối tượng trong các feature maps.
- **Phát hiện đa quy mô:** Sử dụng các feature maps từ các kích thước khác nhau (80x80, 40x40, 20x20) để phát hiện các đối tượng ở các kích thước khác nhau, giúp cải thiện khả năng phát hiện các đối tượng nhỏ và lớn.
- **Dự đoán tọa độ và lớp của bounding boxes:** Sử dụng các lớp cuối cùng để dự đoán tọa độ và lớp của các bounding boxes cho mỗi đối tượng được phát hiện.



Hình 1: Kiến trúc mạng yolov9

Mô tả chi tiết các module trong YOLOv9:

- Silence: Khởi khởi đầu, không thực hiện phép toán nào mà chỉ chuyển tiếp dữ liệu.
- Conv (Convolution): Các lớp tích chập, có thể có kernel size và stride khác nhau, thực hiện phép tích chập để trích xuất đặc trưng từ hình ảnh.
- RepNCSP (ELAN4): Khối tập hợp các lớp tích chập và ReLU (Rectified Linear Unit) để tạo ra các đặc trưng phức tạp hơn.

- ADown (Average Downsample): Khối giảm kích thước bằng phép toán trung bình để giảm số lượng thông tin nhưng giữ nguyên thông tin quan trọng.
- Concat (Concatenate): Ghép nối các đặc trưng từ nhiều mức độ khác nhau để tạo ra thông tin phong phú hơn.
- Upsample: Phép toán tăng kích thước của đặc trưng để phù hợp với các đặc trưng khác trong quá trình ghép nối.
- RepNCSP(ELAN4): Tiếp tục sử dụng các khối tích chập để làm giàu thêm thông tin đặc trưng.
- ADown: Giảm kích thước để phù hợp với các phép toán tiếp theo.
- Detect: Các khối thực hiện phép toán dự đoán, bao gồm việc xác định vị trí, kích thước và loại của các đối tượng trong hình ảnh.

1.3.2. Những cải tiến cốt lõi trong YOLOv9

Những tiến bộ của YOLOv9 bắt nguồn sâu sắc trong việc giải quyết những thách thức do mất thông tin trong các mạng nơ-ron sâu. Nguyên tắc tắc nghẽn thông tin và việc sử dụng sáng tạo các Chức năng đảo ngược là trọng tâm trong thiết kế của nó, đảm bảo YOLOv9 duy trì hiệu quả và độ chính xác cao [3].

a. Information Bottleneck Principle

- Nguyên tắc tắc nghẽn thông tin (Information Bottleneck Principle) cho thấy một thách thức cơ bản trong học sâu: khi dữ liệu đi qua các lớp liên tiếp của mạng, khả năng mất thông tin tăng lên. YOLOv9 chống lại thách thức này bằng cách triển khai thông tin chuyển màu có thể lập trình (PGI), hỗ trợ bảo quản dữ liệu thiết yếu trên độ sâu của mạng, đảm bảo tạo gradient đáng tin cậy hơn và do đó, hội tụ mô hình và hiệu suất tốt hơn.

b. Reversible Functions

- Khái niệm về chức năng đảo ngược (Reversible Functions) là một nền tảng khác trong thiết kế của YOLOv9. Một hàm được coi là có thể đảo ngược nếu nó có thể được đảo ngược mà không bị mất thông tin YOLOv9 kết hợp các chức năng đảo ngược trong kiến trúc của nó để giảm thiểu nguy cơ suy thoái thông tin, đặc biệt là ở các lớp sâu hơn, đảm bảo bảo toàn dữ liệu quan trọng cho các nhiệm vụ phát hiện đối tượng.

c. Impact on Lightweight Models

- Giải quyết mất thông tin đặc biệt quan trọng đối với các mô hình nhẹ, thường không được tham số hóa và dễ bị mất thông tin quan trọng trong quá trình chuyển tiếp. Kiến trúc của YOLOv9, thông qua việc sử dụng PGI và các chức năng đảo

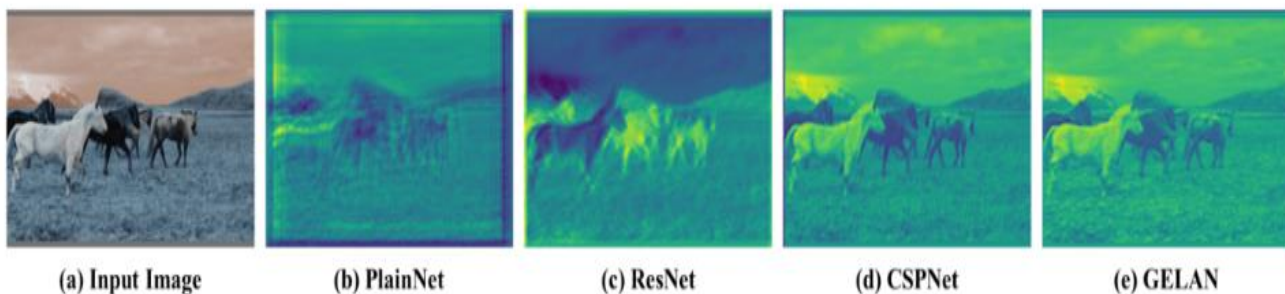
ngược, đảm bảo rằng ngay cả với một mô hình được sắp xếp hợp lý, thông tin cần thiết để phát hiện đối tượng chính xác vẫn được giữ lại và sử dụng hiệu quả.

d. Programmable Gradient Information (PGI):

- PGI được giới thiệu để giải quyết các thách thức về mất thông tin trong quá trình trích xuất đặc trưng và chuyển đổi không gian trong mạng sâu.
- Nó đảm bảo thông tin đầu vào hoàn chỉnh có sẵn cho nhiệm vụ mục tiêu, tạo điều kiện thuận lợi cho thông tin gradient đáng tin cậy để cập nhật trọng số của mạng.
- PGI có thể thích nghi cho các mô hình khác nhau, từ mô hình nhẹ đến mô hình lớn, và giúp đạt được kết quả tốt hơn ngay cả khi được huấn luyện từ đầu so với các mô hình được tiền huấn luyện trên các tập dữ liệu lớn.

e. Generalized Efficient Layer Aggregation Network (GELAN):

- GELAN là một kiến trúc mạng nhẹ được thiết kế dựa trên lập kế hoạch đường dẫn gradient.
- Nó tối ưu hóa việc sử dụng các toán tử tích chập thông thường để đạt được hiệu quả sử dụng tham số tốt hơn so với các phương pháp hiện đại sử dụng tích chập theo chiều sâu.



Hình 2: Kết quả trực quan hóa của bản đồ tính năng đầu ra trọng số ban đầu ngẫu nhiên cho các kiến trúc mạng khác nhau: (a) Input image, (b) PlainNet, (c) ResNet, (d) CSPNet và (e) GELAN

Từ hình, chúng ta có thể thấy rằng trong các kiến trúc khác nhau, thông tin được cung cấp cho hàm mục tiêu để tính toán tổn thất bị mất ở các mức độ khác nhau và kiến trúc GELAN có thể giữ lại thông tin đầy đủ nhất và cung cấp thông tin độ dốc đáng tin cậy nhất để tính toán hàm mục tiêu.

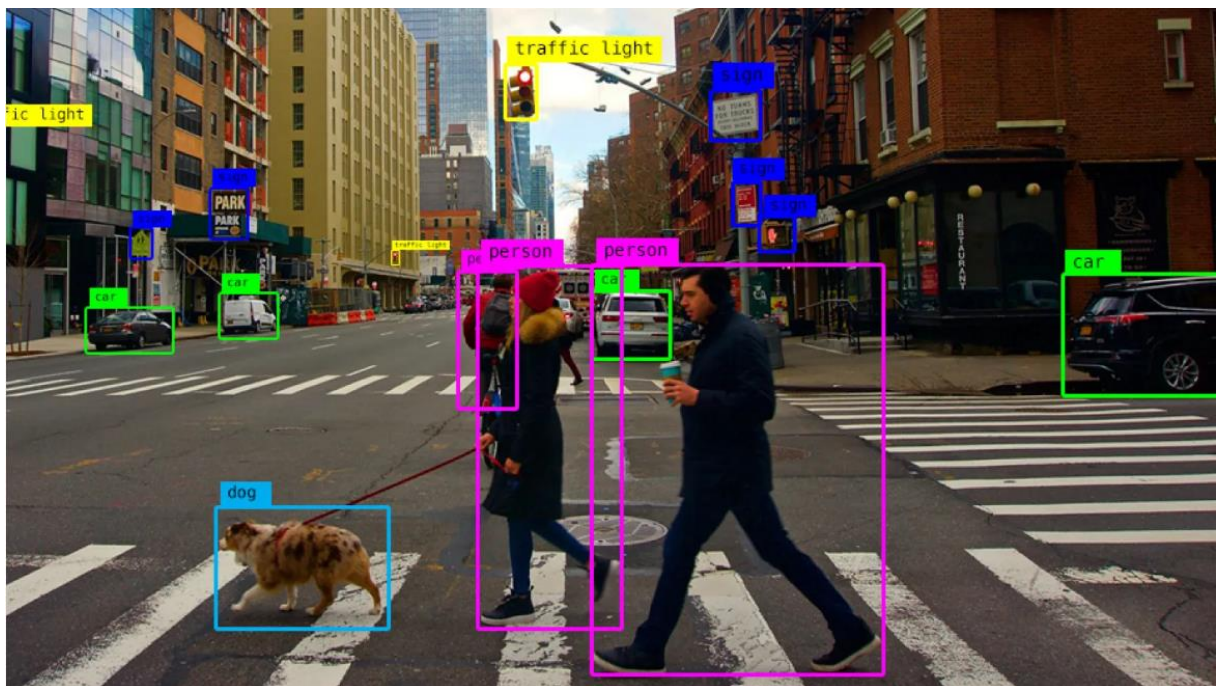
1.4. Nhận dạng đối tượng

Nhận dạng đối tượng (Object Detection) là quá trình nhận diện và xác định vị trí của các đối tượng trong một hình ảnh hoặc khung hình video. Nhiệm vụ này bao gồm hai

phần: phân loại các đối tượng và xác định chính xác vị trí của chúng, thường bằng các hộp bao quanh (bounding boxes). [4]

Các phương pháp:

- a. Phương pháp truyền thống: Các phương pháp này dựa vào việc trích xuất đặc trưng và các bộ phân loại học máy. Các kỹ thuật như Biểu đồ Gradient (Histogram of Oriented Gradients - HOG), Trường quan tâm (Regions of Interest - ROI), và bộ phân loại SVM đã từng rất phổ biến.
- b. Phương pháp hiện đại: Với sự phát triển của học sâu (deep learning), các phương pháp dựa trên mạng nơ-ron tích chập (CNN) trở nên phổ biến. Một số mô hình nổi tiếng bao gồm:
 - + R-CNN (Regions with CNN features): Sử dụng các vùng quan tâm để phát hiện đối tượng.
 - + Fast R-CNN và Faster R-CNN: Cải tiến tốc độ và hiệu suất của R-CNN.
 - + YOLO (You Only Look Once): Phát hiện đối tượng trong một lần duy nhất, rất nhanh chóng.
 - + SSD (Single Shot MultiBox Detector): Kết hợp tính chính xác và tốc độ.



Hình 3: Ví dụ về Object Detection

1.5. Theo dõi đối tượng

Theo dõi đối tượng (Object Tracking) là quá trình theo dõi vị trí của một đối tượng cụ thể trong suốt các khung hình liên tiếp của video. Mục tiêu là duy trì định danh của đối tượng và theo dõi chuyển động của nó. [5]

1.5.1. Giới thiệu

a. Khái niệm

Object Tracking là bài toán theo dõi một hoặc nhiều đối tượng chuyển động theo thời gian trong một video. Hiểu một cách đơn giản nhất, nó là bài toán ở mức độ cao hơn so với object detection, khi đối tượng được xử lý không đơn giản là một hình ảnh mà là một chuỗi các hình ảnh.

Nếu đơn giản như vậy, chẳng phải cứ tách video ra thành các frame rồi áp dụng object detection với từng frame là đủ rồi sao? Cũng không có gì khó lắm! Đương nhiên không chỉ đơn giản như vậy, việc tracking bên cạnh việc xác định các bounding box, còn quan tâm đến khá nhiều yếu tố hoặc nhiều khác nhau:

- ID của mỗi đối tượng cần đảm bảo luôn không đổi qua các frame
- Khi đối tượng bị che khuất hoặc biến mất sau 1 vài frame, hệ thống vẫn cần đảm bảo nhận diện lại được đúng ID khi đối tượng xuất hiện
- Các vấn đề liên quan đến tốc độ xử lý để đảm bảo realtime và tính ứng dụng cao

b. Phân loại

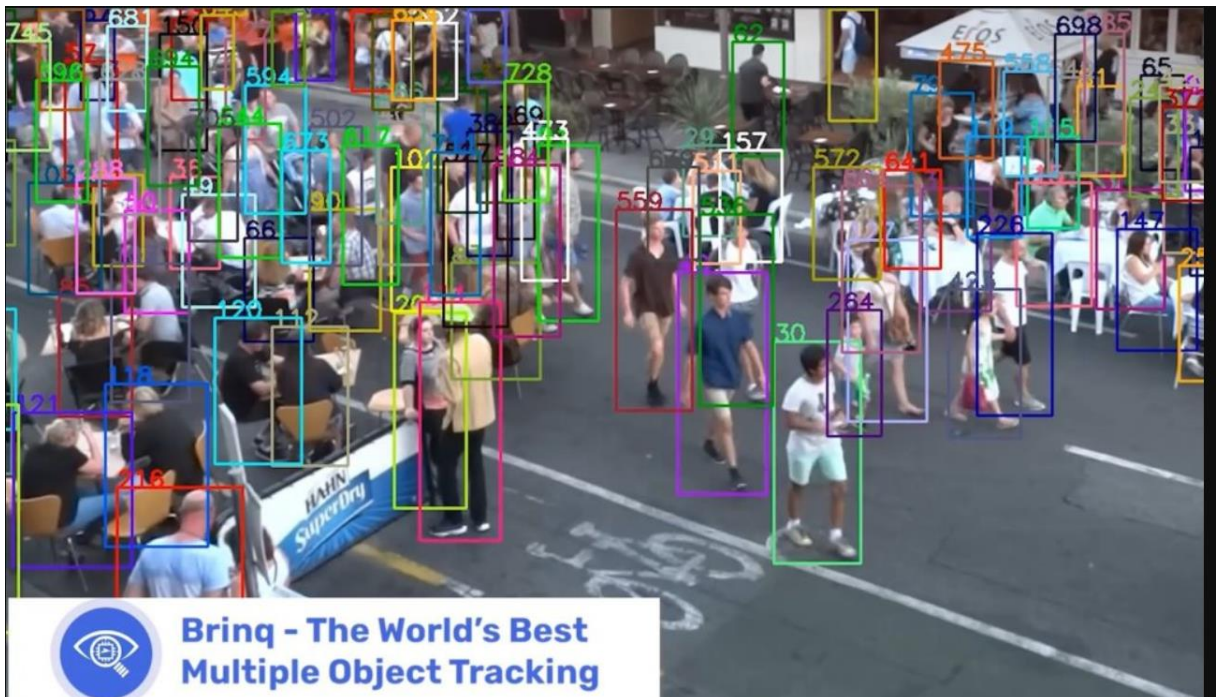
Object Tracking có thể chia thành 2 cách tiếp cận chính:

- **Single Object Tracking (SOT):** Cái tên nói lên tất cả, Single Object Tracking tập trung vào việc theo dõi một đối tượng duy nhất trong toàn bộ video. Và tất nhiên, để biết được cần theo dõi đối tượng nào, việc cung cấp một bounding box từ ban đầu là việc bắt buộc phải có.



Hình 4: Single Object Tracking

- **Mutiple Object Tracking (MOT):** Mutiple Object Tracking hướng tới các ứng dụng có tính mở rộng cao hơn. Bài toán cố gắng phát hiện đồng thời theo dõi tất cả các đối tượng trong tầm nhìn, kể cả các đối tượng mới xuất hiện trong video. Vì điều này, MOT thường là những bài toán khó hơn SOT và nhận được rất nhiều sự quan tâm của giới nghiên cứu.



Hình 5: Mutiple Object Tracking

Bên cạnh việc tiếp cận, các phương pháp giải lớp bài toán này cũng được phân chia rất đa dạng, phổ biến nhất là:

- Online Tracking : Khi xử lý video, Online Tracking chỉ sử dụng frame hiện tại và frame ngay trước đó để tracking. Cách xử lý này có thể sẽ làm giảm độ chính xác của thuật toán, tuy nhiên nó lại phản ánh đúng cách vấn đề được xử lý trong thực tế, khi mà tính "online" là cần thiết
- Offline Tracking : Các phương pháp Offline thường sử dụng toàn bộ frame của video, do đó thường đạt được độ chính xác cao hơn nhiều so với Online Tracking

Ngoài ra còn phân chia theo:

- Detection based Tracking : Tập trung vào mối liên kết chặt chẽ giữa object detection và object tracking, từ đó dựa vào các kết quả của detection để theo dõi đối tượng qua các frame.
- Detection Free Tracking : Coi video như 1 dạng dữ liệu dạng chuỗi, từ đó, áp dụng những phương pháp dành riêng cho "chuỗi" như RNN, LSTM, ...

c. Metric đánh giá

- **FP** (False Positive) : tổng số lần xuất hiện một đối tượng được phát hiện mặc dù không có đối tượng nào tồn tại
- **FN** (False Negative) : tổng số lần mà đối tượng hiện có không được phát hiện.
- **ID Switches** : tổng số lần 1 đối tượng bị gán cho 1 ID mới trong suốt quá trình tracking video
- **MOTA**: Mutiple Object Tracking Accuracy

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDS_t)}{\sum_t (GT_t)}$$

- **MOTP**: Mutiple Object Tracking Precision

$$MOTP = \frac{\sum_{i,t} (d_t^i)}{\sum_t (c_t)}$$

- + dt: Khoảng cách giữa việc định vị các đối tượng trong thực tế và đầu ra phát hiện
- + ct: tổng số kết quả khớp được thực hiện giữa nhận diện thực tế và đầu ra phát hiện được

- **MT** (Most Tracked Target) : tính trong ít nhất 80% video
- **ML** (Most Lost Target) : tính trong 20% video
- **Hz (FPS)**: Tốc độ tracking

1.5.2. Các vấn đề trong Object Tracking

a. Multiple Object Tracking

Một phương pháp Multiple Object Tracking cố gắng hướng đến việc theo dõi tất cả các đối tượng xuất hiện trong khung hình bằng việc phát hiện và gán định danh cho từng đối tượng. Bên cạnh đó, các ID đã được gán cho 1 đối tượng cần đảm bảo nhất quán qua từng frame. Vậy, có những vấn đề gì đáng quan tâm ở đây ?

- Phát hiện "tất cả" các đối tượng: Đây vẫn luôn là vấn đề được quan tâm nhất trong object detection và vẫn không ngừng có những phương pháp, những thuật toán cải thiện vấn đề này. Trong object tracking, đặc biệt là detection based tracking, việc đảm bảo tính chính xác của quá trình detect cũng vô cùng quan trọng
- Đối tượng bị che khuất 1 phần hoặc toàn bộ: Khi 1 ID được gán cho 1 đối tượng, ID cần đảm bảo nhất quán trong suốt video, tuy nhiên, khi một đối tượng bị che khuất, nếu chỉ dựa riêng vào object detection là không đủ để giải quyết vấn đề này
- Đối tượng ra khỏi phạm vi của khung hình và sau đó xuất hiện lại : Tương tự như vấn đề trước đó, chúng ta vẫn đang nói về chỉ số ID switches. Cần giải quyết tốt vấn đề nhận dạng lại đối tượng kể cả việc che khuất hay biến mất để giảm số lượng ID_switches xuống mức thấp nhất có thể
- Các đối tượng có quỹ đạo chuyển động giao nhau hoặc chồng chéo lên nhau. : Việc các đối tượng có quỹ đạo chồng chéo lên nhau cũng có thể dẫn đến hậu quả gán nhầm ID cho các đối tượng, đây cũng là vấn đề chúng ta cần chú ý xử lý khi làm việc với Multiple Object Tracking

b. Realtime Object Tracking

Realtime Object Tracking lại quan tâm nhiều hơn về tốc độ xử lý của phương pháp. Vẫn chưa có định nghĩa, tốc độ như thế nào mới được gọi là realtime (30 FPS, 40 FPS hay phải nhanh hơn thế nữa). Thay vì xét về tốc độ, chúng ta hãy định nghĩa realtime một cách dễ hình dung hơn. Phương pháp được gọi là realtime cần đảm bảo tốc độ đưa ra output là nhanh hơn hoặc ít nhất là nhanh bằng tốc độ đưa vào input.

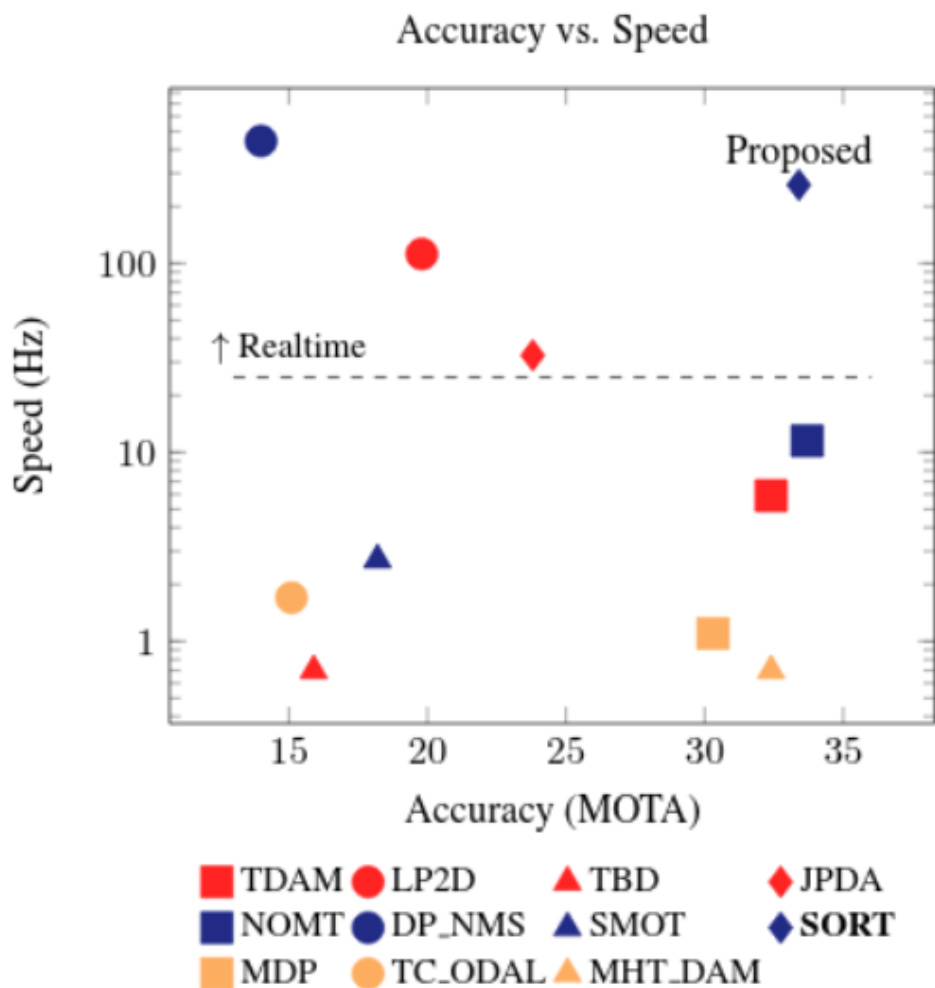
Trong thực tế, nếu việc xử lý từng frame chỉ khiến video có độ trễ 1s so với tốc độ bình thường của nó, việc xử lý này cũng có thể chấp nhận rằng đó là realtime. Tuy nhiên,

ngay cả khi chấp nhận có độ trễ, việc đảm bảo tính realtime vẫn luôn là một vấn đề nan giải. Thông thường, chúng ta có thể bỏ qua 1 vài frame không xử lý cho đến khi frame hiện tại xử lý xong, sau đó tiếp tục các frame sau - pha xử lý này vẫn sẽ đem lại cảm giác là việc xử lý đang là realtime, tuy nhiên, bù lại, việc tracking mỗi x frame lại làm giảm đáng kể tính chính xác mong muốn.

Hiện nay, các nghiên cứu mới nhất vẫn luôn tìm kiếm những phương pháp đủ nhanh để hướng tới tính realtime trong xử lý.

1.5.3. Simple Online Realtime Object Tracking

Phần này mình sẽ trình bày về Simple Online Realtime Object Tracking (SORT), một thuật toán thuộc dạng Tracking-by-detection (hay Detection based Tracking).



Hình 6: Accuracy và Speed của SORT

Một đặc điểm của lớp các thuật toán Tracking-by-detection là tách object detection ra như một bài toán riêng biệt và cố gắng tối ưu kết quả trong bài toán này. Công việc sau

đó là tìm cách liên kết các bounding box thu được ở mỗi frame và gán ID cho từng đối tượng. Do đó, chúng ta có một khung quá trình xử lý với mỗi frame mới như sau:

- Detect: phát hiện vị trí các đối tượng trong frame
- Predict: Dự đoán vị trí mới của các đối tượng dựa vào các frame trước đó
- Associate: Liên kết các vị trí detected với các vị trí dự đoán được để gán ID tương ứng

Vậy SORT xử lý các phần Detect, Predict và Associate như thế nào ?

a. Giải thuật Hungary trong Object Tracking

- Giải thuật Hungary [6] được phát triển và công bố vào năm 1955, đề xuất để giải bài toán phân công công việc (assignment problem).
- Thuật toán Hungary trong Object Tracking: Có n detection ($i = 1, 2, \dots, n$) và n track predicted ($j = 1, 2, \dots, n$). Để liên kết một detection i với một track j giả sử dựa vào 1 độ đo D - D là khoảng cách giữa i và j trong không gian vector. Bài toán đặt ra là cần liên kết mỗi detection với mỗi track tương ứng sao cho sai số của việc liên kết là nhỏ nhất.
- Dựa vào 2 định lý :
 - o [Định lý]. Giả sử ma trận chi phí của bài toán giao việc là không âm và có ít nhất n phần tử bằng 0. Hơn nữa nếu n phần tử 0 này nằm ở n hàng khác nhau và n cột khác nhau thì phương án giao cho người i thực hiện công việc tương ứng với số 0 này ở hàng i sẽ là phương án tối ưu (lời giải) của bài toán.
 - o [Định lý]. Cho $C=[c_{ij}]$ là ma trận chi phí của bài toán giao việc (n người, n việc) và $X^*=[x_{ij}]$ là một lời giải (phương án tối ưu) của bài toán này. Giả sử ' C ' là ma trận nhận được từ C bằng cách thêm số $\alpha \neq 0$ (dương hoặc âm) vào mỗi phần tử ở hàng r của C . Khi đó X^* cũng là lời giải của bài toán giao việc với ma trận chi phí ' C '.
- Thuật toán Hungary dựa vào 2 định lý này, từ đó hình thành được hướng xử lý bài toán: Biến đổi ma trận (cộng trừ vào các hàng hoặc cột) để đưa về ma trận có n phần tử bằng 0 nằm ở các hàng và cột khác nhau, sau đó, lấy ra phương án tối ưu là các vị trí chứa các phần tử 0 này.
- Cụ thể hơn, có thể chia thuật toán thành các bước sau:
 - Bước 1 (Bước chuẩn bị). Trừ các phần tử trên mỗi hàng của C cho phần tử nhỏ nhất trên hàng đó, tiếp theo trừ các phần tử trên mỗi cột cho phần tử nhỏ nhất trên cột đó. Kết quả ta nhận được ma trận C' có tính chất: trên mỗi hàng, cột có ít nhất một phần tử 0 và bài toán giao việc với ma trận C' có cùng lời giải như bài toán với ma trận C .

- Bước 2: Vẽ một số tối thiểu các đường thẳng trên dòng và cột để đảm bảo mọi phần tử 0 đều được đi qua.
- Bước 3: Nếu có n đường thẳng được vẽ, kết thúc thuật toán và tiến hành phân công công việc. Nếu số đường thẳng được vẽ nhỏ hơn n , vẫn chưa tìm được phương án phân công tối ưu, tiến hành bước tiếp theo.
- Bước 4: Mỗi hàng (hoặc cột) có đường thẳng vẽ qua, ta gọi các hàng (cột) đó là các hàng (cột) thiết yếu. Các hàng (cột) còn lại là các hàng (cột) không thiết yếu. Tìm phần tử nhỏ nhất không nằm trong các hàng (cột) thiết yếu, tiến hành trừ mỗi hàng không thiết yếu cho phần tử nhỏ nhất ấy và cộng giá trị nhỏ nhất ấy cho cột thiết yếu. Ta được ma trận C'' có cùng lời giải với ma trận C' . Sau đó quay lại Bước 2

b. Bộ lọc Kalman trong Object Tracking

- Trong object tracking, Kalman Filter [7] được biết đến nhiều nhất với vai trò dự đoán các trạng thái của đối tượng hiện tại dựa vào các track trong quá khứ và update lại các detection sau khi đã được liên kết với các track trước đó.
- Linear Kalman Filter giả định các mô hình của quá trình (θ_k , H_k) đều là các mô hình tuyến tính. Khi đó:

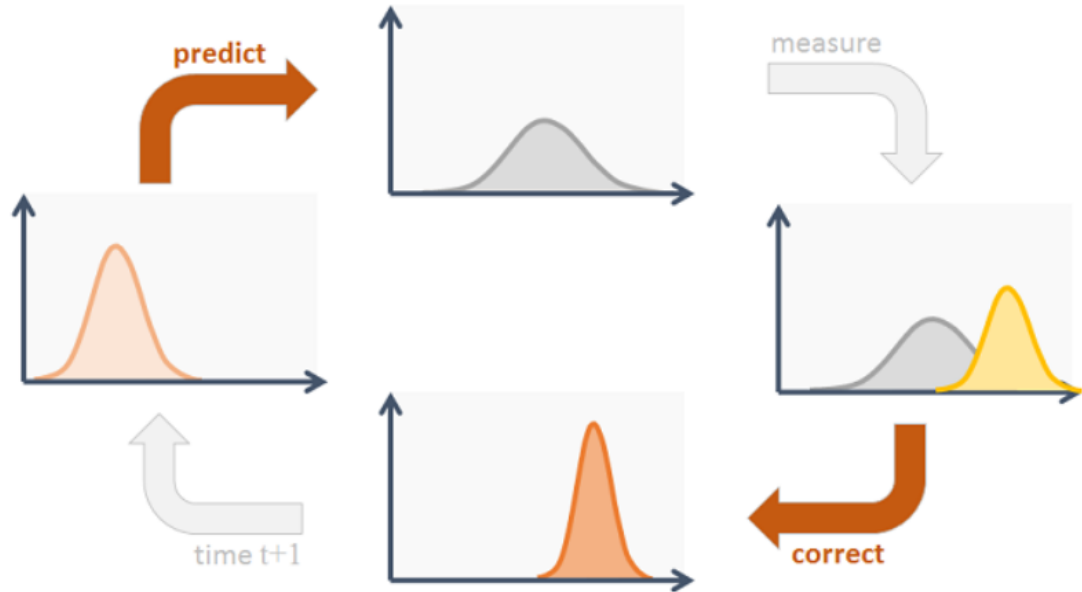
$$\begin{aligned}x_k &= F_{k-1}x_{k-1} + u_{k-1} \\z_k &= H_{k-1}x_k + w_k\end{aligned}$$

Với:

- $x_k \sim N(\hat{x}, \Sigma)$:

$$p(x_k) = \frac{1}{(2\pi^{D/2})|\Sigma_k|^{1/2}} \exp \left[-\frac{(x_k - \hat{x}_k)^T \Sigma_k^{-1} (x_k - \hat{x}_k)}{2} \right]$$

- F_k, H_k là các ma trận $n * n$ (state transition matrix) và $m*n$ (measurement function)
- $u_k \sim N(0, Q)$, $w_k \sim N(0, R)$



Hình 7: Sơ đồ các bước xử lý của Kalman Filter

Giai đoạn Dự đoán (Prediction)

$$x_k = F_{k-1}x_{k-1} + u_{k-1}$$

Để dự đoán các giá trị trạng thái của quá trình ngẫu nhiên, ta dự đoán các giá trị mean $\hat{x}$ và covariance Σ . Theo tính chất kỳ vọng và ma trận hiệp phương sai với vector ngẫu nhiên, ta có :

$$\hat{x}_k = E[x_k] = E[F_{k-1}x_{k-1} + u_{k-1}] = F_{k-1}E[x_{k-1}] + E[u_{k-1}] = F_{k-1}\hat{x}_{k-1}$$

Giai đoạn hiệu chỉnh (Update)

Trong giai đoạn cập nhật, trạng thái dự đoán và ma trận hiệp phương sai được cập nhật dựa trên phép đo mới:

- **Tính toán độ đoán Kalman (Kalman Gain):**

$$K_k = \Sigma_k H_k^T (R_k + H_k \Sigma_k H_k^T)^{-1}$$

- K_k : Độ đoán Kalman tại thời điểm k.
- Σ_k : Ma trận hiệp phương sai của lỗi dự đoán tại thời điểm k.
- H_k : Ma trận đo lường, hay còn gọi là ma trận quan sát, tại thời điểm k.
- H_k^T : Ma trận chuyển vị của H_k
- R_k : Ma trận hiệp phương sai của lỗi đo lường tại thời điểm k

- **Cập nhật trạng thái:**

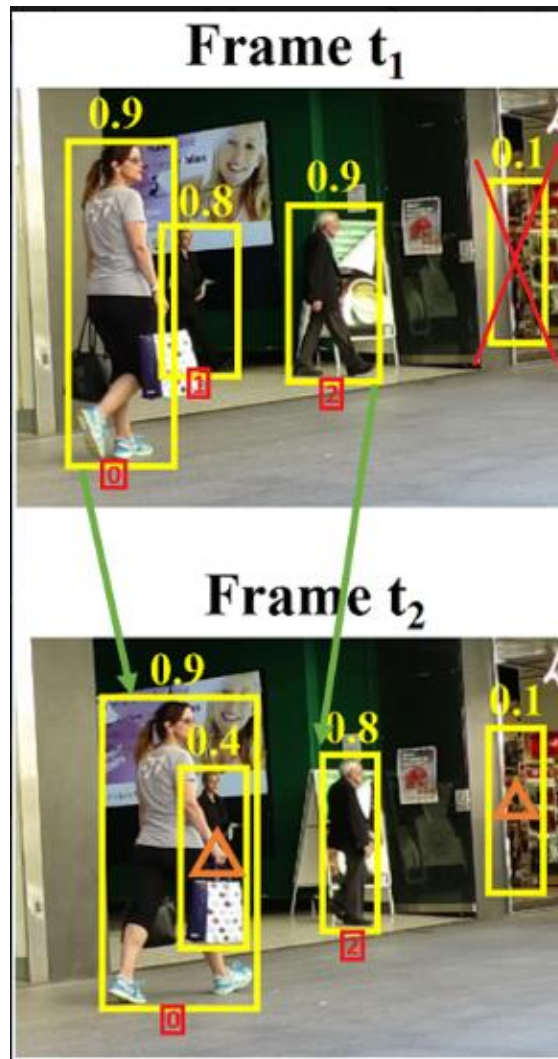
$$\hat{x}_k^F = \hat{x}_k + K_k(z_k - H_k \hat{x}_k)$$

- $\hat{x}_k^F$: Trạng thái ước lượng sau khi cập nhật (filtered state estimate) tại thời điểm k.
- $\hat{x}_k$: Trạng thái ước lượng trước khi cập nhật (predicted state estimate) tại thời điểm k.
- **Cập nhật ma trận hiệp phương sai:**
 - $\Sigma_k^F = \Sigma_k^{\square}(I - K_k H_k)$
 - I: Ma trận đơn vị.

1.5.4. ByteTrack

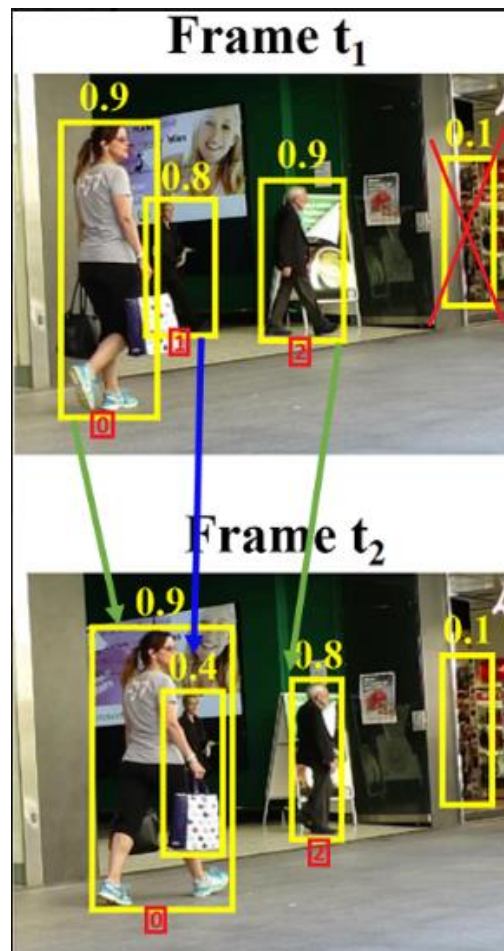
ByteTrack là một phương pháp theo dõi đa đối tượng (MOT) được thiết kế để giải quyết những hạn chế của các hệ thống theo dõi hiện tại [8]. Các phương pháp MOT truyền thống thường loại bỏ các hộp phát hiện có độ tin cậy thấp, dẫn đến việc bỏ sót các phát hiện và các đường đi bị phân đoạn. Tuy nhiên, ByteTrack liên kết mọi hộp phát hiện, bao gồm cả những hộp có độ tin cậy thấp, để cải thiện độ chính xác và độ bền của việc theo dõi, đặc biệt là trong các tình huống khó khăn như bị che khuất.

Đầu tiên, quá trình matching sẽ được thực hiện giữa boxes của các đối tượng đã được track tại frame t1 và các boxes được detect tại frame t2, những boxes này có Confidence score cao (lớn hơn threshold: 0.5) và các bước cũng giống như SORT được giới thiệu ở trên.



Hình 8: Bước 1 của ByteTrack

Sau khi match xong những bboxes mà có Confidence score cao, trong frame thứ 2 ta có thể thấy còn 2 bboxes nữa với Confidence score lần lượt là: 0.4 và 0.1 chưa được match (những bboxes có hình tam giác màu cam bên trong). Với 2 bboxes còn lại này, các tác giả lại cho nó đi qua quá trình matching một lần nữa để tính IoU giữa nó và các đối tượng được track ở các frame trước.



Hình 9: Bước 2 của ByteTrack

Từ đó có thể thấy box của người thứ 2 từ bên trái qua khi bị che khuất bây giờ đã được match. Còn box được detect với Confidence score 0.1 thì được loại bỏ vì IoU giữa nó và box được track từ các frame phía trước là 0.

Khi áp dụng kỹ thuật này vào các phương pháp theo dõi hiện có, nó cho thấy rằng hiệu suất được cải thiện một cách nhất quán. Đặc biệt có thể thấy rằng, số lượng IDs giảm đi rất nhiều (IDs càng ít càng tốt). Ở đây, số lượng chuyển đổi ID có nghĩa là số lần ID được thay đổi trong quá trình theo dõi đối với từng đối tượng.

1.6. Đếm đối tượng

Đếm đối tượng (Object Counting) là quá trình đếm số lượng các đối tượng cụ thể trong một hình ảnh hoặc video. Đây có thể là việc đếm các đối tượng trong một khung hình đơn hoặc theo dõi và đếm chúng trong một chuỗi video. [9]

Các phương pháp:

- Phương pháp truyền thống: Sử dụng các kỹ thuật xử lý ảnh như ngưỡng (thresholding), phân đoạn (segmentation), và phân cụm (clustering) để đếm đối tượng.

b. Phương pháp dựa trên học máy:

- + Sử dụng phát hiện đối tượng: Đếm số lượng các hộp bao quanh đối tượng được phát hiện.
- + Phương pháp dựa trên mật độ (Density-based methods): Sử dụng các mạng nơ-ron tích chập để ước lượng bản đồ mật độ, từ đó tính toán số lượng đối tượng.



Counting number: 292 cars

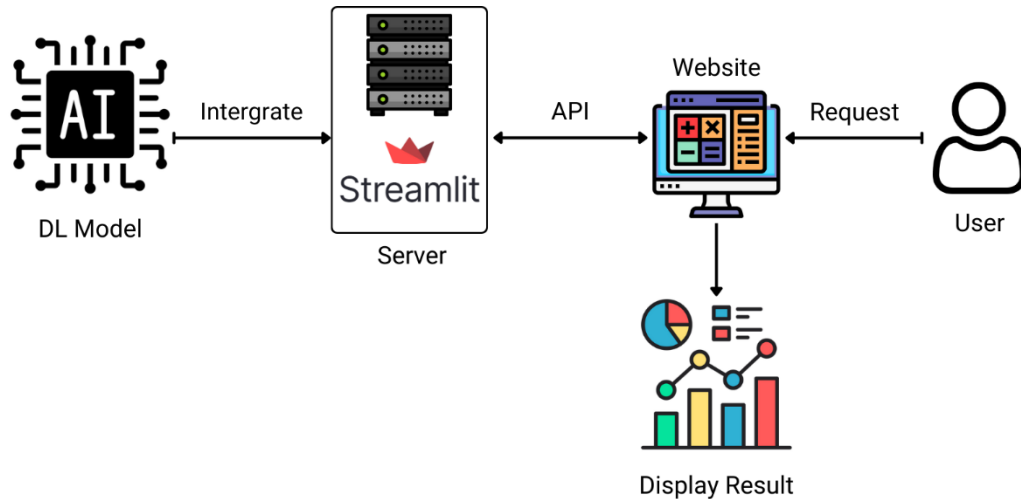
Ground Truth: 299 cars

Hình 10: Ví dụ về Object Counting

Chương 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

Trong chương này trình bày về tổ chức hệ thống nhận diện phương tiện

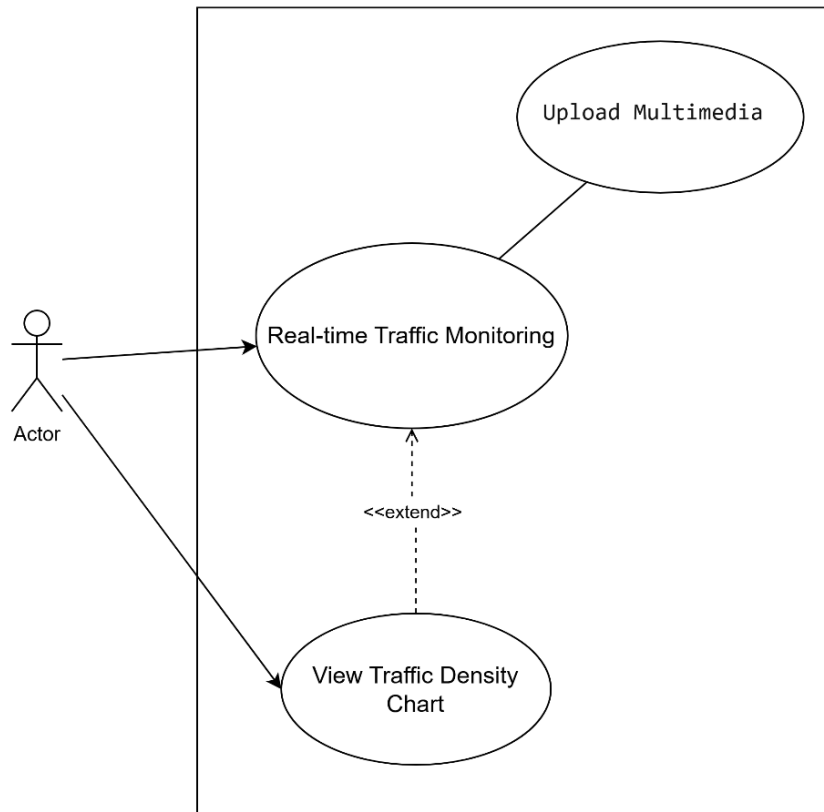
2.1 Tổ chức chương trình



Hình 11: Sơ đồ tổng quan hệ thống

Đồ án đã phát triển và triển khai một ứng dụng web sử dụng Streamlit Framework, tận dụng khả năng tương tác và hiển thị dữ liệu mạnh mẽ của nó để hỗ trợ mô hình YOLOv9 trong việc nhận dạng và phân tích mật độ giao thông. Hệ thống bao gồm các chức năng chính sau:

- Đa nguồn dữ liệu: Có khả năng xử lý và phân tích dữ liệu từ nhiều nguồn đầu vào khác nhau bao gồm ảnh, video, URL YouTube, livestream YouTube, RTSP và webcam.
- Nhận diện và theo dõi thời gian thực: Nhận diện, theo dõi và đếm đối tượng trong thời gian thực, cung cấp thông tin tức thì về mật độ giao thông.
- Vùng quan tâm định hình: Cho phép người dùng vẽ vùng Region of Interest (ROI) dưới dạng hình polygon để chỉ đếm những đối tượng xuất hiện trong vùng đó, tập trung vào các khu vực cần quan sát đặc biệt.
- Biểu đồ phân tích dữ liệu: Hiển thị chart line chứa số lượng phương tiện đã phát hiện được theo thời gian và pie chart để cho thấy phân bố của loại phương tiện trong khu vực đã phát hiện, cung cấp cái nhìn trực quan về tình hình giao thông.

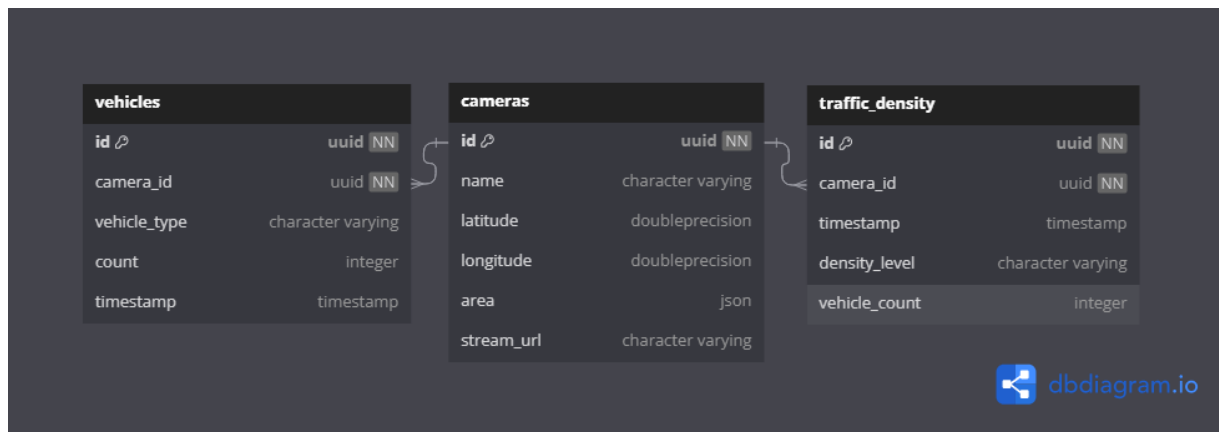


Hình 12: Sơ đồ usecase của hệ thống

2.2 Lưu trữ và quản lý dữ liệu với PostgreSQL

Hệ thống đã tích hợp tính năng lưu trữ kết quả số lượng xe đã được nhận diện và đếm từ mô hình YOLOv9 vào một cơ sở dữ liệu PostgreSQL. Điều này giúp duy trì và quản lý các dữ liệu thống kê giao thông một cách hiệu quả và có thể truy xuất lại được sau này.

PostgreSQL được chọn lựa như một giải pháp cơ sở dữ liệu phù hợp nhờ vào tính bảo mật, khả năng mở rộng và khả năng xử lý dữ liệu phức tạp. Mỗi lần phát hiện và đếm số lượng xe, hệ thống tự động ghi nhận thời gian, số lượng xe và vị trí camera tương ứng vào cơ sở dữ liệu. Điều này cho phép hệ thống phân tích và xây dựng các biểu đồ thống kê dựa trên dữ liệu lịch sử, từ đó cung cấp thông tin hữu ích cho quản lý giao thông và điều hành.



Hình 13: Sơ đồ ERD (Entity – Relationship Diagram)

1. Vehicles:

- **Vehicle_type:** Loại xe được phát hiện (ví dụ: ô tô, xe máy, xe buýt).
- **Count:** Số lượng xe được đếm trong một khoảng thời gian nhất định.
- **Timestamp:** Thời điểm ghi nhận số lượng xe.
- **Camera_id:** Khóa ngoại liên kết đến Camera tương ứng.

Mỗi khi một xe được nhận diện và đếm, dữ liệu về loại xe, số lượng và thời điểm sẽ được ghi vào bảng Vehicles của cơ sở dữ liệu PostgreSQL. Điều này giúp cung cấp thông tin chi tiết về lưu lượng giao thông và phân bố các loại xe qua các thời điểm khác nhau.

2. Camera:

- **Name:** Tên của camera.
- **Latitude:** Vĩ độ của camera.
- **Longitude:** Kinh độ của camera.
- **Area:** Vùng quan sát của camera, được lưu dưới dạng dữ liệu JSON để định hình các khu vực quan tâm.
- **Stream_url:** Địa chỉ URL stream của camera.

Thông tin về các camera được sử dụng để ghi nhận dữ liệu giao thông và vị trí của chúng. Các thông tin này cung cấp sự liên kết giữa dữ liệu nhận diện và vị trí vật lý của các thiết bị quan sát.

3. TrafficDensity:

- **timestamp:** Thời điểm ghi nhận mật độ giao thông.
- **density_level:** Mức độ mật độ giao thông (ví dụ: thấp, trung bình, cao).
- **vehicle_count:** Tổng số lượng phương tiện được nhận diện trong vùng quan sát.
- **camera_id:** Khóa ngoại liên kết đến Camera tương ứng.

Dữ liệu về mật độ giao thông được ghi lại trong bảng TrafficDensity, kèm theo thông tin chi tiết về số lượng phương tiện và thời gian ghi nhận. Điều này giúp theo dõi và phân tích xu hướng và biến động của lưu lượng giao thông theo thời gian.

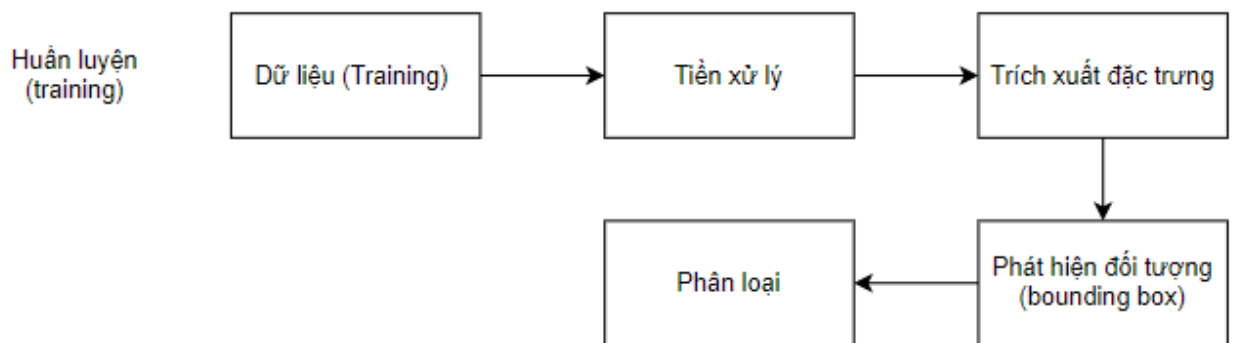
Chương 3: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

Trong chương này trình bày các nội dung bao gồm: các vấn đề cần giải quyết của bài toán nhận diện phương tiện, giải giáp, cách triển khai và đánh giá kết quả sau khi huấn luyện.

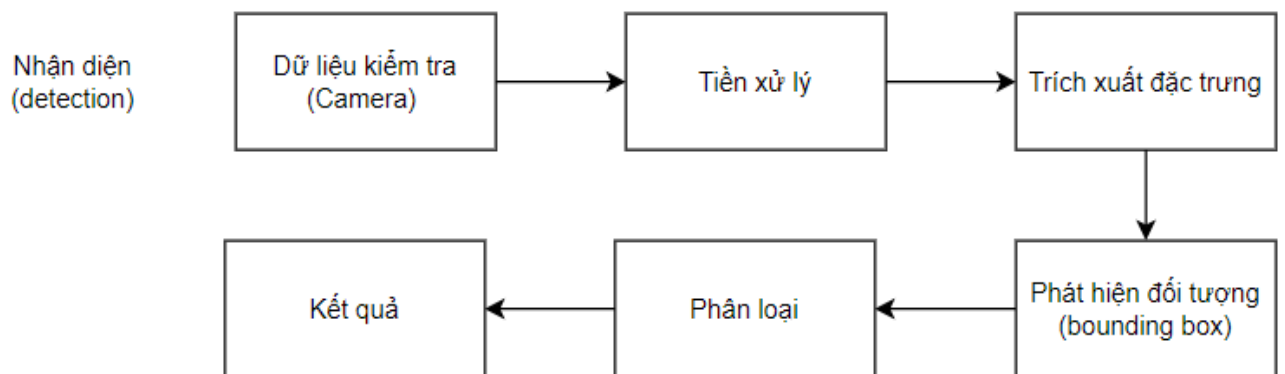
3.1. Các vấn đề cần giải quyết

3.1.1. Phân tích bài toán

Một hệ thống nhận dạng đối tượng thông thường gồm các bước sau đây:



Hình 14: Sơ đồ huấn luyện



Hình 15: Sơ đồ nhận diện

- Tiền xử lý: Bước này nhằm mục đích lọc nhiễu, nâng cao chất lượng ảnh, trong bước này bao gồm các bước: căn chỉnh ảnh, chuẩn hóa ánh sáng

- Trích rút đặc trưng: Ảnh sẽ được đưa qua 1 mạng neural network (thường là 1 mạng CNN) để trích rút đặc trưng
- Phát hiện đối tượng và phân loại: Ở bước này 1 phương pháp phát hiện và phân loại vật thể như SSD, YOLO, Faster R-CNN, ... được sử dụng. Kết quả là một danh sách các bounding box chứa đối tượng được xác định và định vị trong ảnh, cùng với class predictions và confidence scores tương ứng.

3.1.2. Các vấn đề cần giải quyết

- Cần thu thập và gán nhãn dữ liệu
- Phát hiện và nhận diện các loại phương tiện
- Cần ứng dụng để kiểm soát và kiểm tra kết quả
- Hệ thống chạy theo thời gian thực

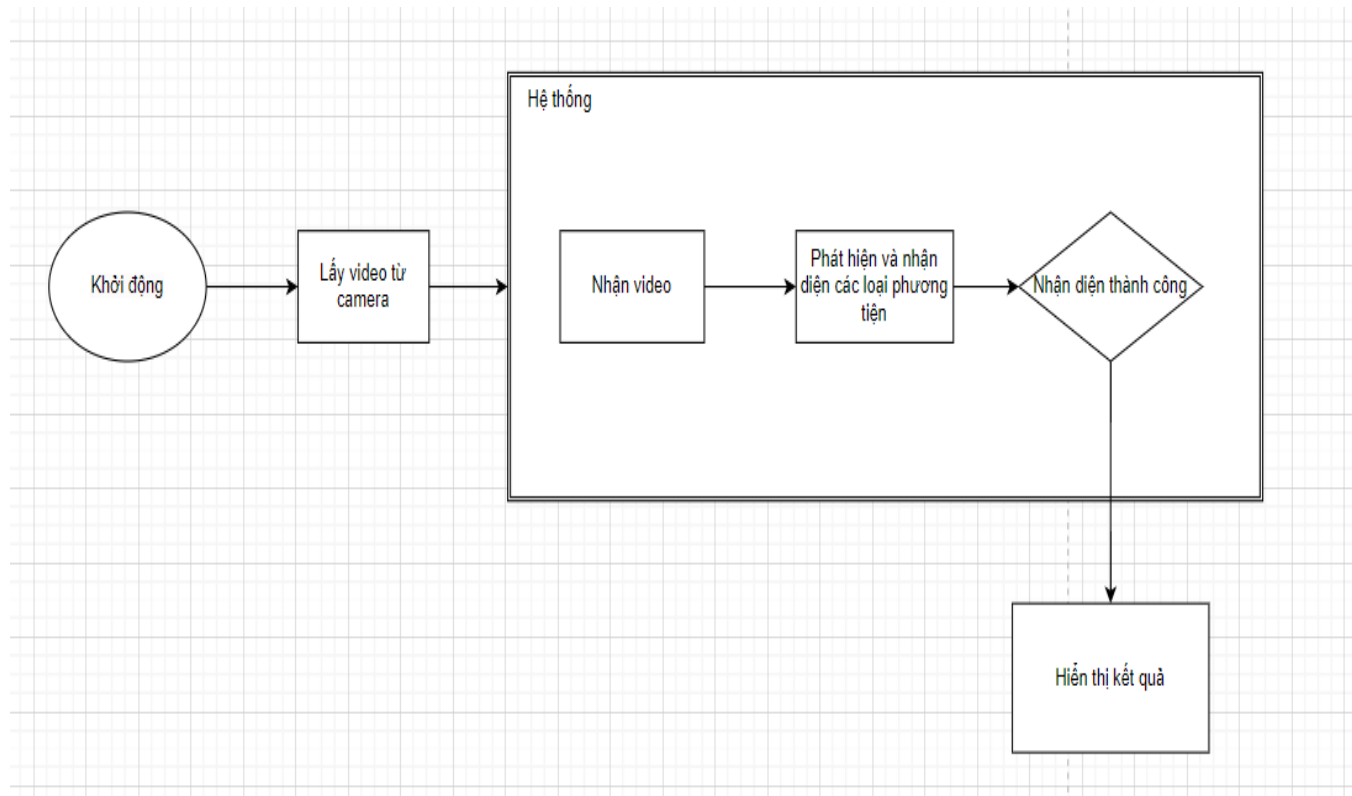
3.2. Đề xuất giải pháp tổng quan

Bảng 1: Giải pháp tổng quan

Dữ liệu	Thu thập từ các nguồn dữ liệu có sẵn công khai Tự thu thập và gán nhãn
Nhận diện phương tiện	Xây dựng và huấn luyện model nhận diện biển báo Thử nghiệm với các model: YOLO, Faster R-CNN, ... Huấn luyện trên Kaggle
Ứng dụng	Xây dựng web Có chức năng nhận diện và đếm phương tiện trên ảnh hoặc video Hiển thị kết quả lên màn hình

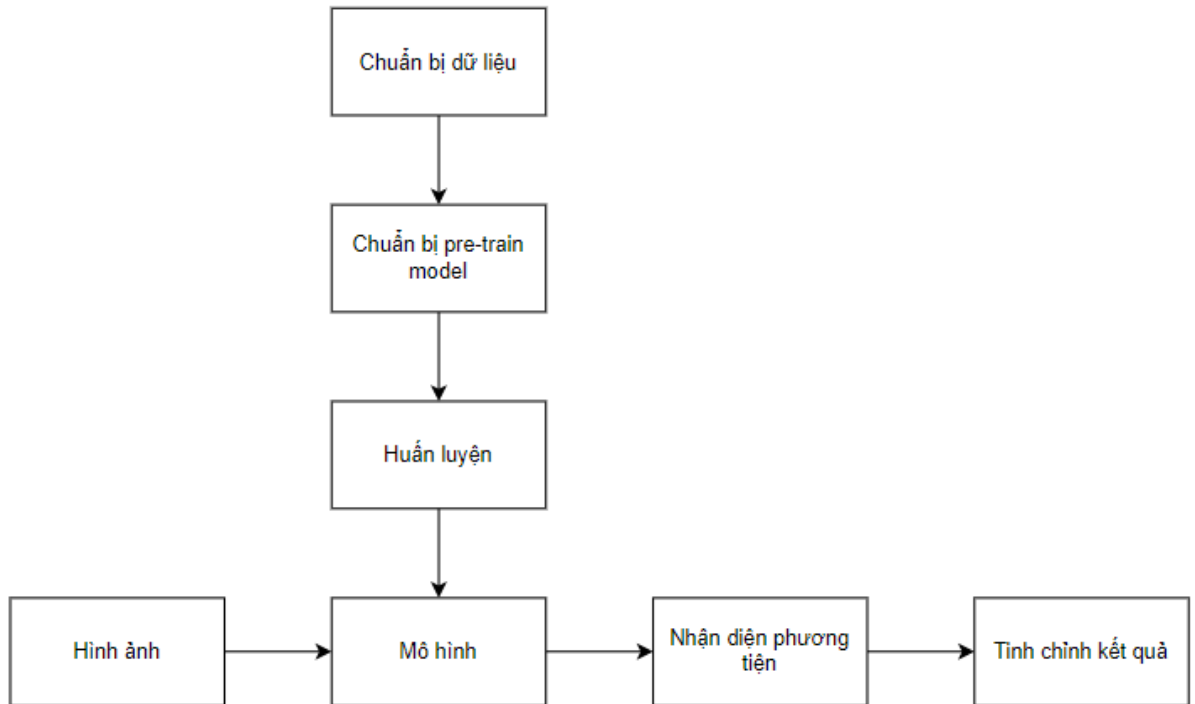
3.3. Giải pháp

3.3.1. Sơ đồ tổng quan hệ thống



Hình 16: Sơ đồ tổng quan hệ thống

3.3.2. Giải pháp nhận diện phương tiện



Hình 17: Giải pháp nhận diện phương tiện giao thông

3.4. Nhận diện phương tiện

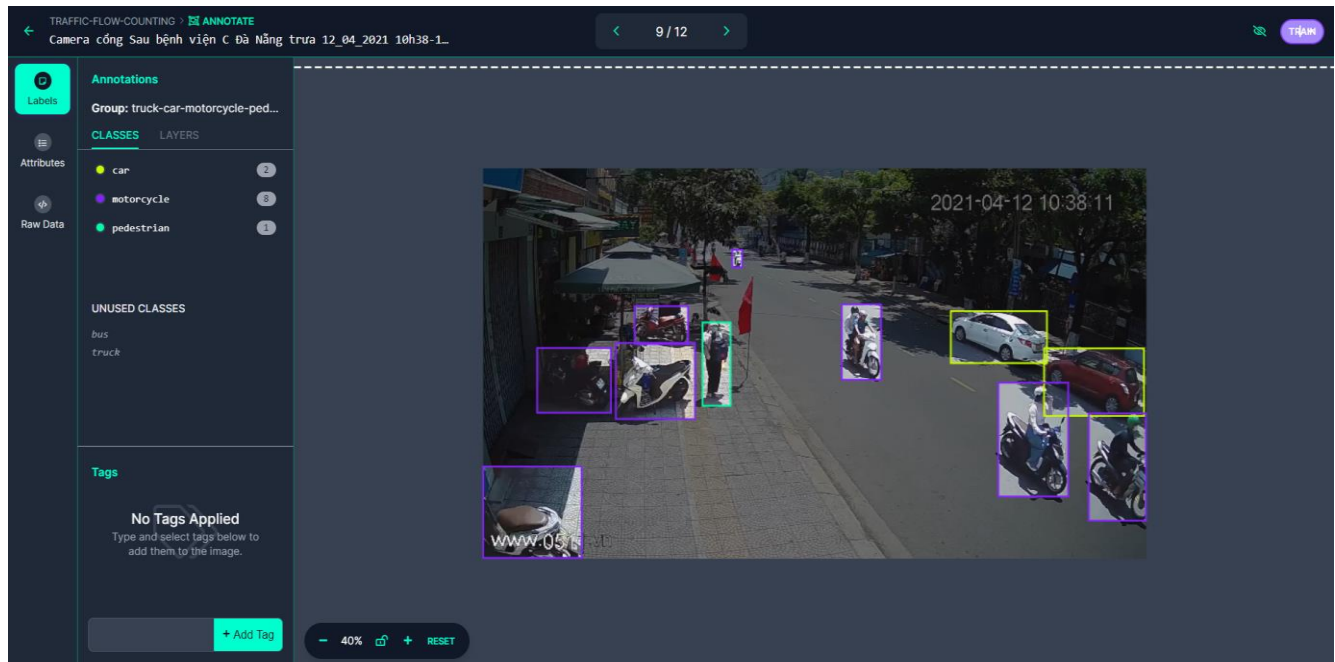
3.4.1. Tập dữ liệu

a. Nguồn gốc và cách thức thu thập

Trong phần huấn luyện mô hình nhận diện biển báo giao thông, tập dữ liệu là các biển báo giao thông của hệ thống giao thông đường bộ Việt Nam. Tập dữ liệu được thu thập từ 2 nguồn:

- Nguồn 1: Bộ dữ liệu có sẵn công khai từ nhiều khu vực khác nhau (998 ảnh).
- Nguồn 2: Dữ liệu tự thu thập bằng cách lấy từ các camera công cộng trong khu vực thành phố Đà Nẵng (3478 ảnh).

Dữ liệu sau đó được tổng hợp lên Roboflow và thực hiện gán nhãn.



Hình 18: Gán nhãn trên Roboflow

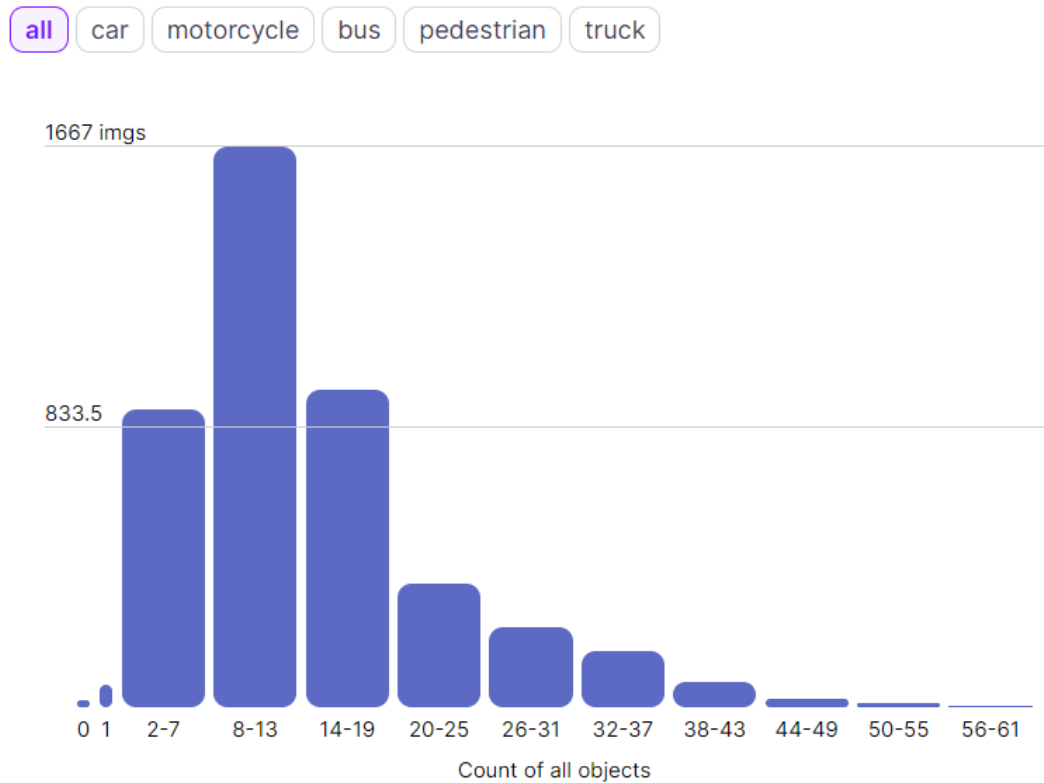
b. Thông tin tập dữ liệu

- Tập dữ liệu gồm có 4476 hình ảnh phương tiện và người ở Đà Nẵng.
- Sau khi gán nhãn, bao gồm 62856 nhãn tương ứng với 5 class (4 phương tiện và người).

Bảng 2: Thông tin tập dữ liệu

Class	Ý nghĩa	Số nhãn
Motorcycle	Xe máy	37403
Car	Xe ô tô	19018
Pedestrian	Người đi bộ	3730
Truck	Xe tải	1581
Bus	Xe buýt	1124

Histogram of Object Count by Image



Hình 19: Biểu đồ về số lượng nhân có trong ảnh

c. Tiền xử lý

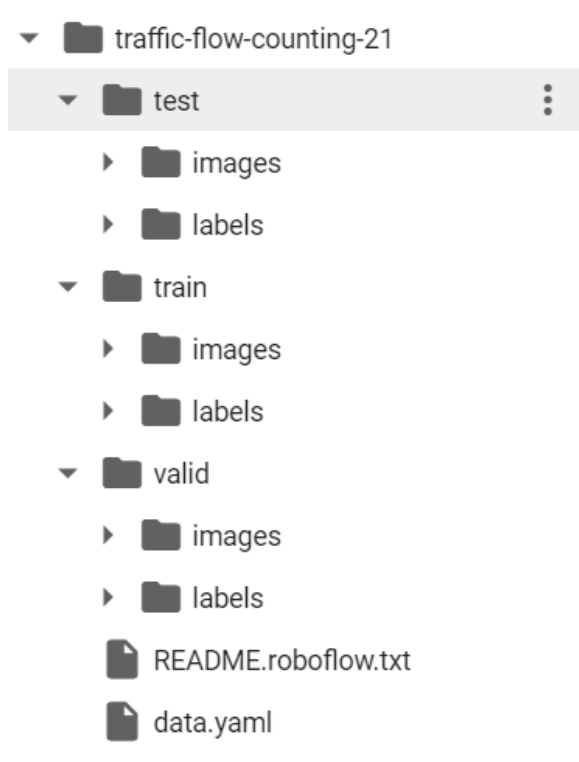
Đầu tiên dữ liệu sẽ được chuẩn hoá về cùng một kích thước 640x640. Sau đó, dữ liệu sẽ được chia thành 3 tập huấn luyện: 89%, đánh giá: 7%, kiểm thử: 4%.

Để tăng tính đa dạng và khả năng tổng quát hoá của mô hình, nhóm đã áp dụng các kỹ thuật tăng cường dữ liệu trên tập huấn luyện:

- + Hue: điều chỉnh màu sắc của hình ảnh. Giá trị được chỉ định là từ -20° đến $+20^{\circ}$. Thay đổi màu sắc giúp tăng tính đa dạng của dữ liệu huấn luyện.
- + Saturation: điều chỉnh độ bão hòa của hình ảnh. Giá trị được chỉ định là từ - 20% đến +20%
- + Exposure: điều chỉnh độ phơi bày của hình ảnh. Giá trị được chỉ định là từ -15% đến 15%
- + Blur: áp dụng hiệu ứng mờ lên hình ảnh, lên tới 1.2px.
- + Noise: áp dụng hiệu ứng nhiễu lên hình ảnh, lên tới 1.25% mỗi pixels

Kết quả là tập huấn luyện đã tăng từ 4476 hình ảnh lên 11012 hình ảnh, tỉ lệ sau cùng của tập dữ liệu là huấn luyện: 89 (%), đánh giá: 7 (%), kiểm thử: 4 (%).

Sau khi tăng cường và phân chia tập dữ liệu, dữ liệu sẽ được chuyển đổi về định dạng dữ liệu của YOLOv9:



Hình 20: Cấu trúc thư mục dataset

3.4.2. Huấn luyện

Trong quá trình huấn luyện mô hình yolo, việc sử dụng GPU là cần thiết để cải thiện tốc độ xử lý và hiệu suất của mô hình. Vì thế nhóm sử dụng tài nguyên GPU từ Kaggle nhằm tiết kiệm chi phí và tăng tốc độ huấn luyện mô hình.

Cấu hình huấn luyện:

- + Dịch vụ sử dụng : Kaggle
- + GPU : GPU T4*2
- + Epochs: 50
- + Batch size: 16
- + Pre-train model : YOLOv8, YOLOv9

3.4.3. Các công thức đánh giá kết quả

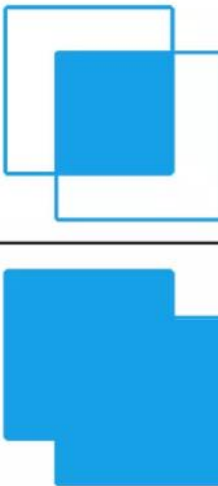
Để tiến hành đánh giá hiệu suất mô hình, nhóm sử dụng mAP (mean Average Precision). mAP là độ đo được sử dụng rất phổ biến cho các bài toán Object Detection. Công thức mAP dựa trên các chỉ số phụ sau đây

- Intersection Over Union (IoU)
- Confusion Matrix

- Precision
- Recall
- F1 Score
- mAP

a. Intersection Over Union (IoU)

IOU là chỉ số đánh giá được sử dụng để đánh giá độ chính xác của phát hiện đối tượng dựa trên tập dữ liệu cụ thể. IOU được tính bằng:


$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

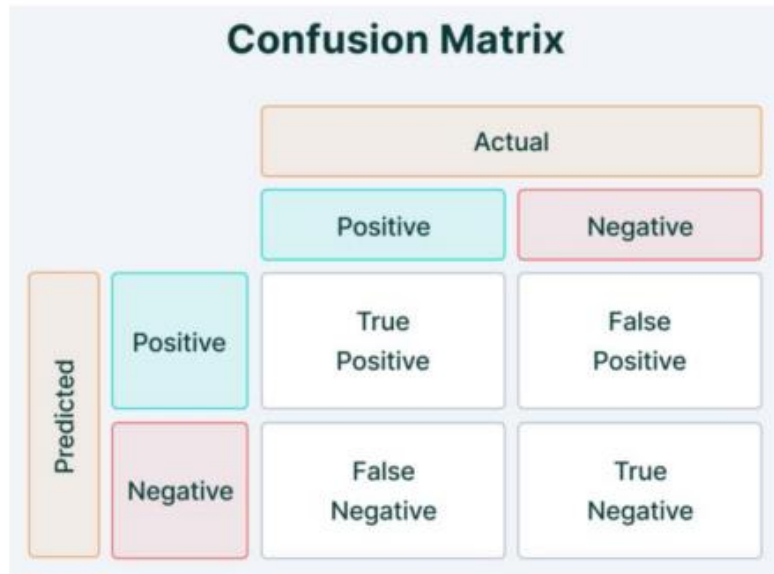
Hình 21: Intersection Over Union (IoU)

Trong đó Area of Overlap là diện tích phần giao nhau giữa predicted bounding box với growth-truth bounding box, còn Area of Union là diện tích phần hợp giữa predicted bounding box với growth-truth bounding box. Nếu IOU lớn hơn 1 ngưỡng (thường bằng 0.5) thì prediction được đánh giá là tốt.

b. Confusion Matrix

Để tạo một confusion matrix, ta cần 4 thuộc tính :

- True Positive (TP) : Một kết quả phát hiện đúng ($\text{IOU} \geq \text{ngưỡng}$).
- False Positive (FP) : Một kết quả phát hiện sai ($\text{IOU} < \text{ngưỡng}$).
- True Negative (TN) : Không có phương tiện hoặc người dân và không phát hiện ra phương tiện hoặc người dân
- False Negative (FN) : Có phương tiện hoặc người dân nhưng không phát hiện ra



Hình 22: Confusion Matrix

c. Precision

Độ chính xác (Precision) là tỷ lệ giữa số lượng phát hiện đúng (True Positives - TP) và tổng số phát hiện (bao gồm cả đúng và sai).

$$Precision = \frac{TP}{TP + FP}$$

d. Recall

Độ nhạy (Recall) là tỷ lệ giữa số lượng phát hiện đúng (TP) và tổng số đối tượng thực tế (bao gồm cả đúng và bỏ sót).

$$Recall = \frac{TP}{TP + FN}$$

e. F1 Score

F1 Score là trung bình điều hòa của độ chính xác và độ nhạy, cung cấp một thước đo cân bằng giữa hai chỉ số này.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

f. Mean Average Precision (mAP)

Mean Average Precision là một trong những chỉ số quan trọng nhất trong đánh giá mô hình phát hiện đối tượng. mAP là trung bình của chỉ số Average Precision (AP) tính cho từng lớp đối tượng.

- **Average Precision (AP):**

AP là diện tích dưới đường cong Precision-Recall. Để tính AP, ta cần:

- + Sắp xếp các phát hiện theo thứ tự giảm dần của độ tin cậy (confidence).
- + Tính Precision và Recall tại mỗi ngưỡng.
- + Vẽ đồ thị Precision-Recall và tính diện tích dưới đường cong.

- **Mean Average Precision (mAP):**

mAP là trung bình của AP trên tất cả các lớp đối tượng.

$$mAP = \frac{1}{N} \sum_{k=0}^n \binom{N}{i=1} AP_i$$

Trong đó N là số lớp đối tượng và AP_i là Average Precision của lớp thứ i.

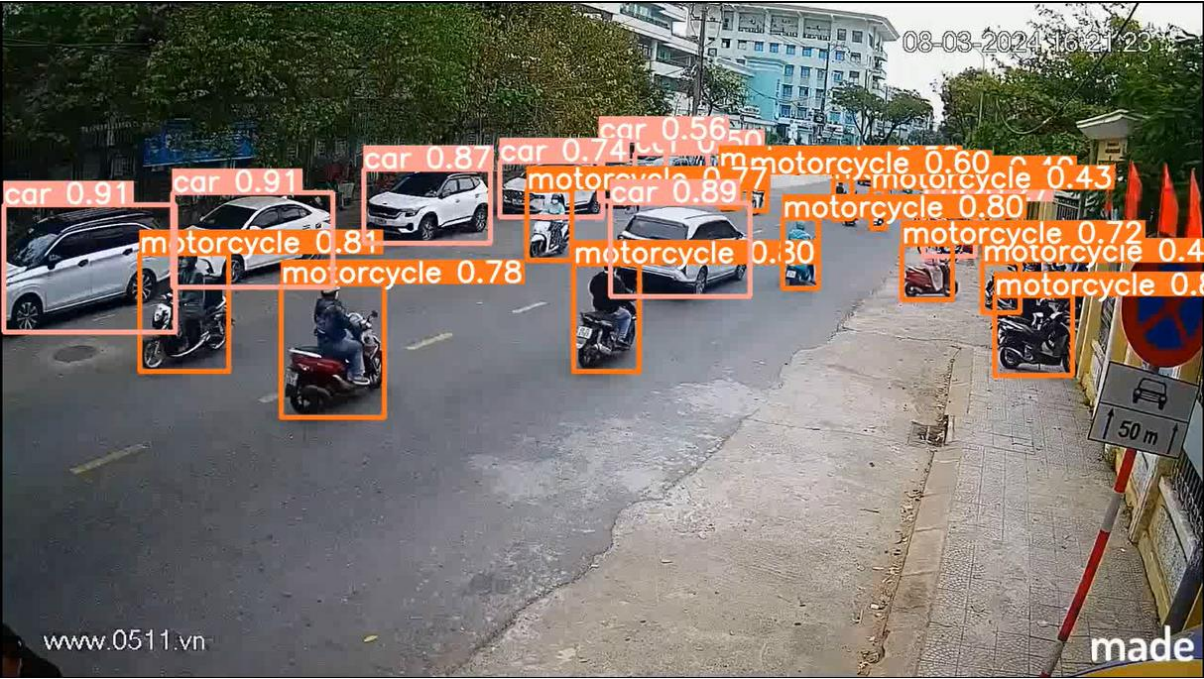
g. Các chỉ số khác

- FPS (Frames Per Second): Tốc độ xử lý của mô hình, quan trọng đối với các ứng dụng thời gian thực.
- Latency (Độ trễ): Thời gian xử lý một khung hình.

3.5. Kết quả mô hình

3.5.1. Kết quả định tính

Kết quả mô hình nhận diện có độ chính xác khá tốt. Tuy nhiên, vẫn còn 1 số trường hợp phương tiện không nhận diện được do khoảng cách xa và một số trường hợp phương tiện nhận diện chưa chính xác (nhầm lẫn với phương tiện khác hoặc nhầm lẫn với môi trường)

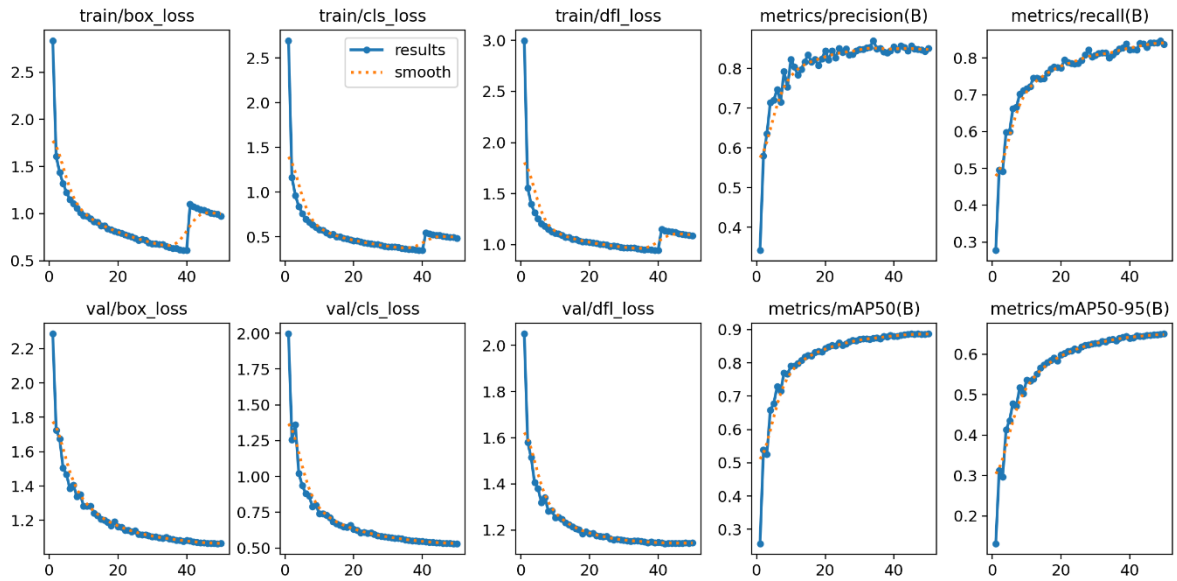


Hình 23: Kết quả nhận diện trên ảnh

3.5.2. Kết quả định lượng

Bảng 3: Kết quả mô hình nhận diện phương tiện

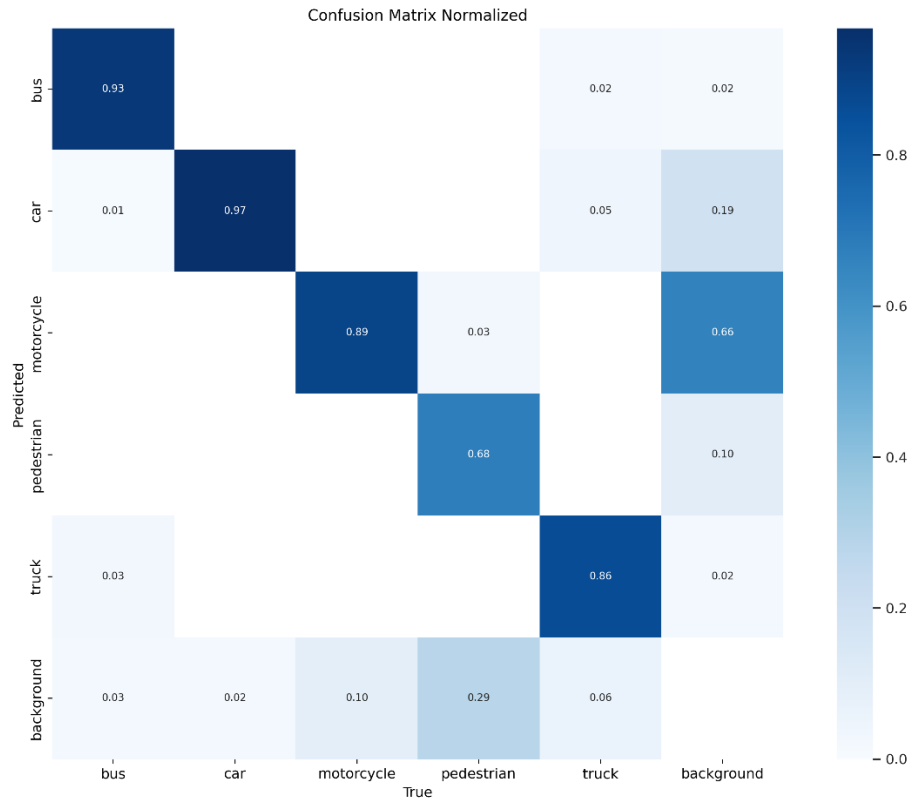
Model	Image Size	Precision	Recall	mAP@.5	mAP@.5:.95	Pre-train
Yolov8	640	0.839	0.789	0.849	0.606	
Yolov8	640	0.89	0.833	0.901	0.671	X
Yolov9	640	0.852	0.838	0.887	0.651	
Yolov9	640	0.871	0.87	0.909	0.684	X



Hình 24: Kết quả train mô hình với yolov9

Nhận xét:

- **Hiệu suất mô hình:** Mô hình YOLO này đang học rất tốt và đạt hiệu suất cao, với các chỉ số chính xác (precision), thu hồi (recall), và mAP đều tăng dần và ổn định sau khoảng 20 epoch.
- **Mất mát giảm dần:** Các giá trị mất mát (loss) giảm đều trong cả quá trình huấn luyện và kiểm tra, cho thấy mô hình đang dần dần tối ưu hóa và học tốt các đặc trưng của dữ liệu.
- **Sự ổn định:** Các giá trị của metrics (precision, recall, mAP) đều ổn định sau một số epoch nhất định, chứng tỏ mô hình đã hội tụ và có khả năng dự đoán chính xác.



Hình 25: Ma trận nhầm lẫn của yolov9

Nhận xét:

- **Độ chính xác cao cho các loại bus, car, và truck:**

Mô hình có độ chính xác cao đối với các loại bus, car, và truck, với các giá trị tương ứng là 93%, 97%, và 86%. Điều này cho thấy mô hình có khả năng phân loại chính xác cao cho các đối tượng này.

- **Nhầm lẫn chủ yếu ở motorcycle và pedestrian:**

Có một tỷ lệ nhầm lẫn đáng kể giữa motorcycle và pedestrian. Cụ thể, 29% của pedestrian bị nhầm lẫn là motorcycle và 66% của motorcycle bị nhầm lẫn là background. Điều này cho thấy mô hình có thể gặp khó khăn trong việc phân biệt hai đối tượng này, hoặc có thể do dataset không đủ phân biệt rõ ràng.

- **Background nhầm lẫn với nhiều đối tượng khác:**

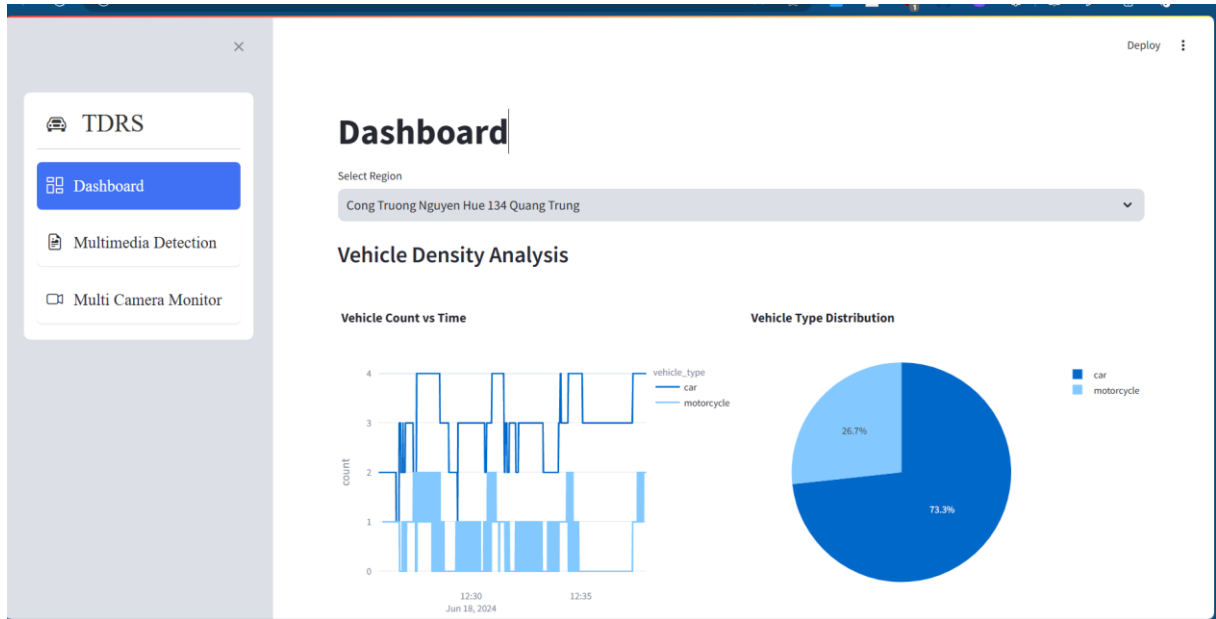
Background có tỷ lệ nhầm lẫn với nhiều loại đối tượng khác, đặc biệt là với pedestrian (29%) và motorcycle (66%). Điều này có thể chỉ ra rằng mô hình cần được cải thiện trong việc xác định ranh giới giữa các đối tượng và background.

3.6. Kết quả phần mềm

Gồm 3 trang Dashboard, Multimedia Detection, Multi Camera Monitor

a. Giao diện trang Dashboard

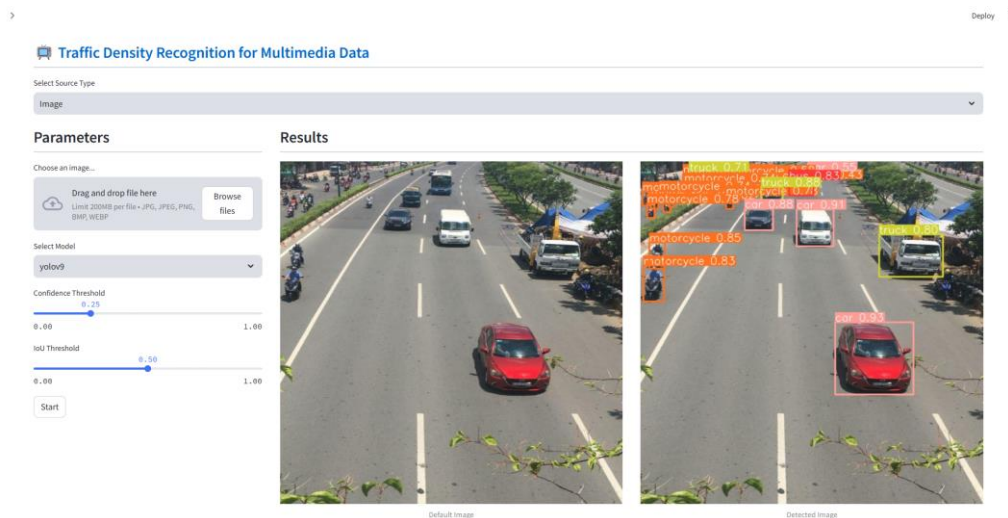
Trang sẽ hiển thị biểu đồ chart với số lượng đối tượng đã đếm được và pie chart cho thấy tỉ lệ phần trăm của các loại đối tượng khác nhau phát hiện trong khu vực như xe máy, ô tô, bus, ...



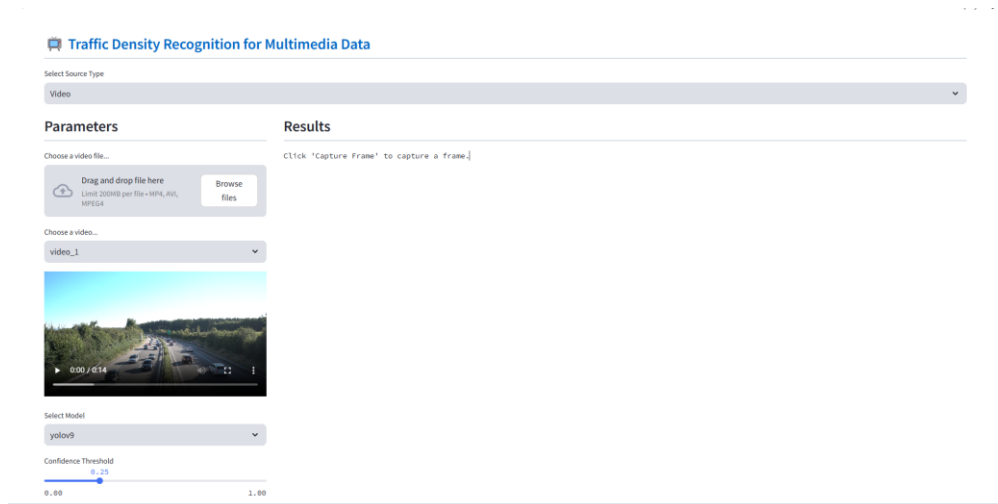
Hình 26: Giao diện Dashboard

b. Giao diện trang Multimedia Detection

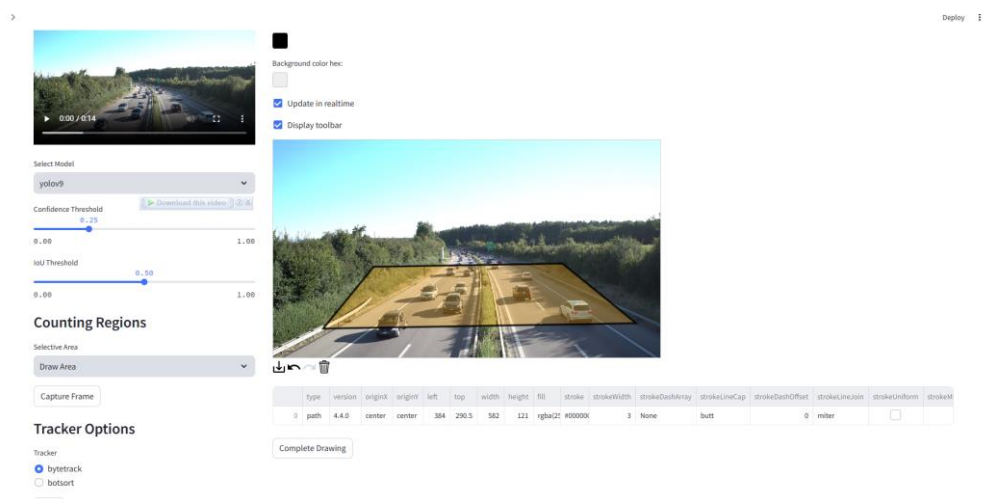
Trang có khả năng xử lý và phân tích dữ liệu từ nhiều nguồn đầu vào khác nhau bao gồm ảnh, video, URL YouTube, livestream YouTube, RTSP và webcam. Ngoài ra, trang còn cho phép người dùng vẽ vùng Region of Interest (ROI) dưới dạng hình polygon để chỉ đếm những đối tượng xuất hiện trong vùng đó, tập trung vào các khu vực cần quan sát đặc biệt.



Hình 27: Giao diện Multimedia Detection: Detect Image



Hình 28: Giao diện Multimedia Detection: Detect Video

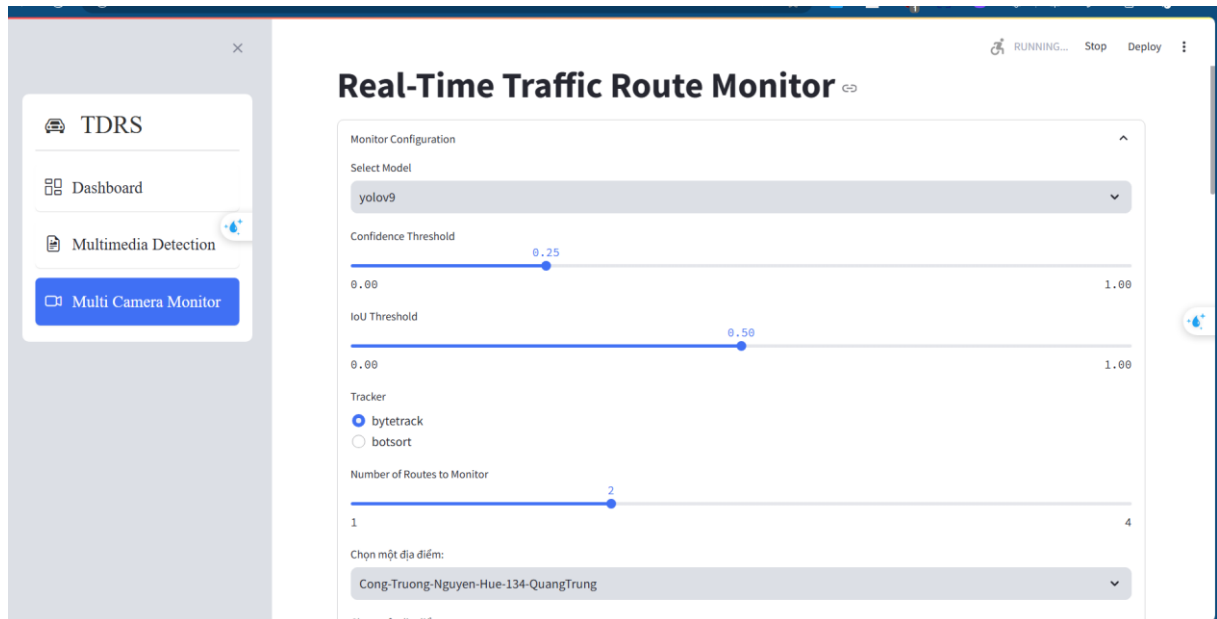


Hình 29: Giao diện Multimedia Detection: vẽ ROI

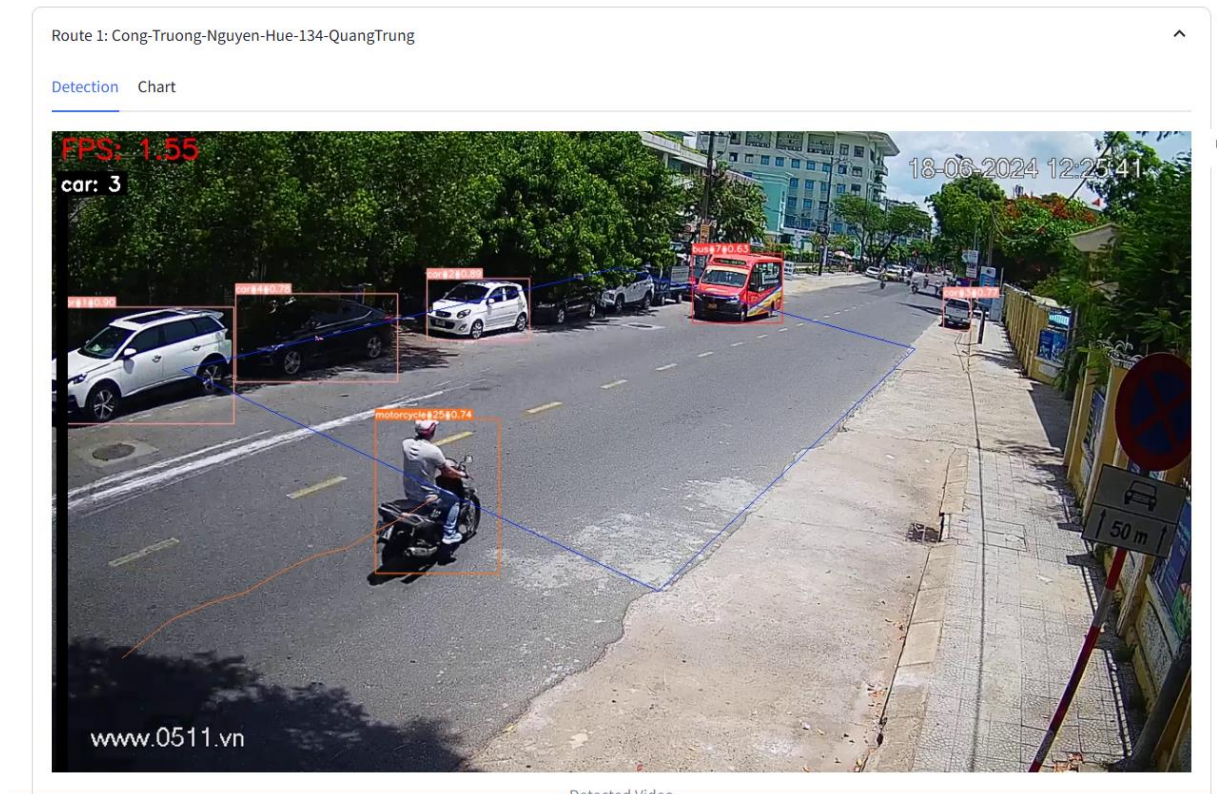
- Hình 12 và hình 13 cho thấy giao diện của trang Multimedia Detection với chức năng lần lượt là xử lý nguồn đầu vào với ảnh và video
- Hình 14 cho thấy chức năng vẽ ROI dưới dạng polygon lên ảnh được lấy từ đầu vào (ảnh, video, ...) để chỉ đếm những đối tượng xuất hiện trong vùng đó, tập trung vào các khu vực cần quan sát đặc biệt.

c. Giao diện trang Multi Camera Monitor

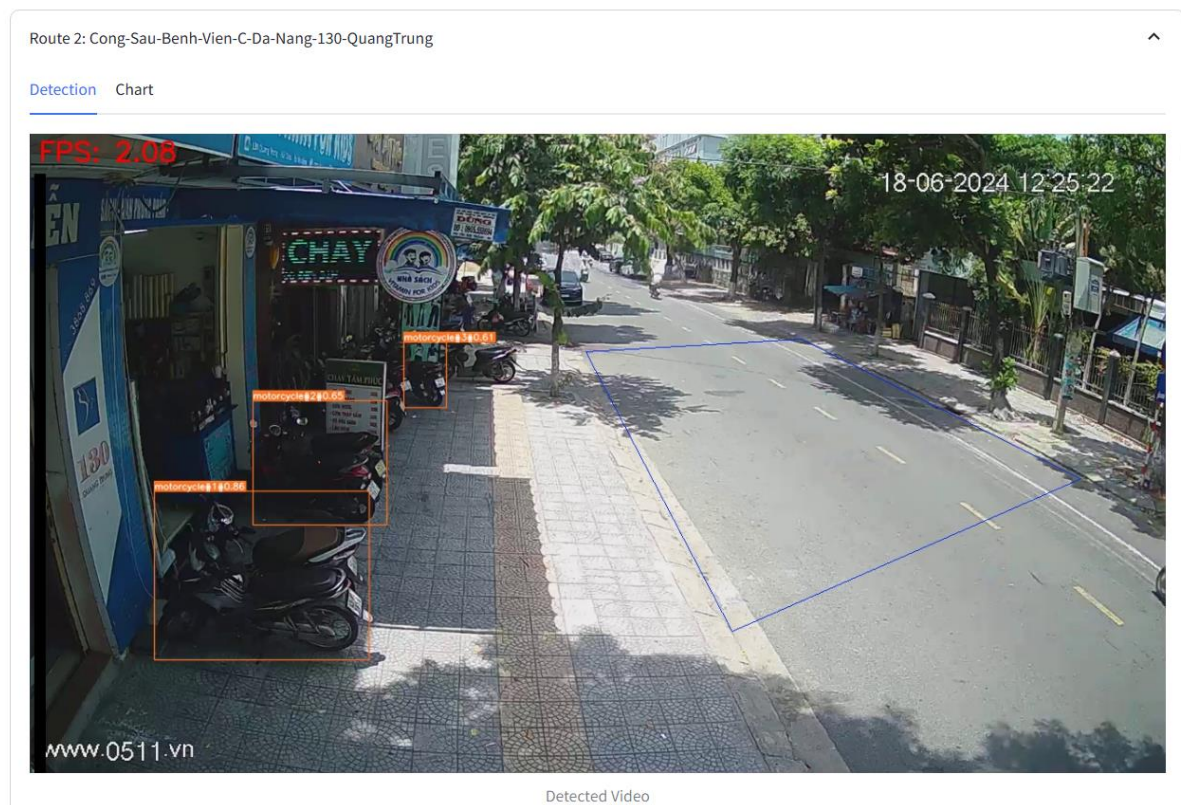
Trang dùng để xử lý luồng livestream camera công cộng đến từ kênh youtube camera giám sát Đà Nẵng. Trang có khả năng phát hiện, theo vết và đếm đối tượng theo thời gian thực với nhiều luồng livestream chạy đồng thời cùng lúc.



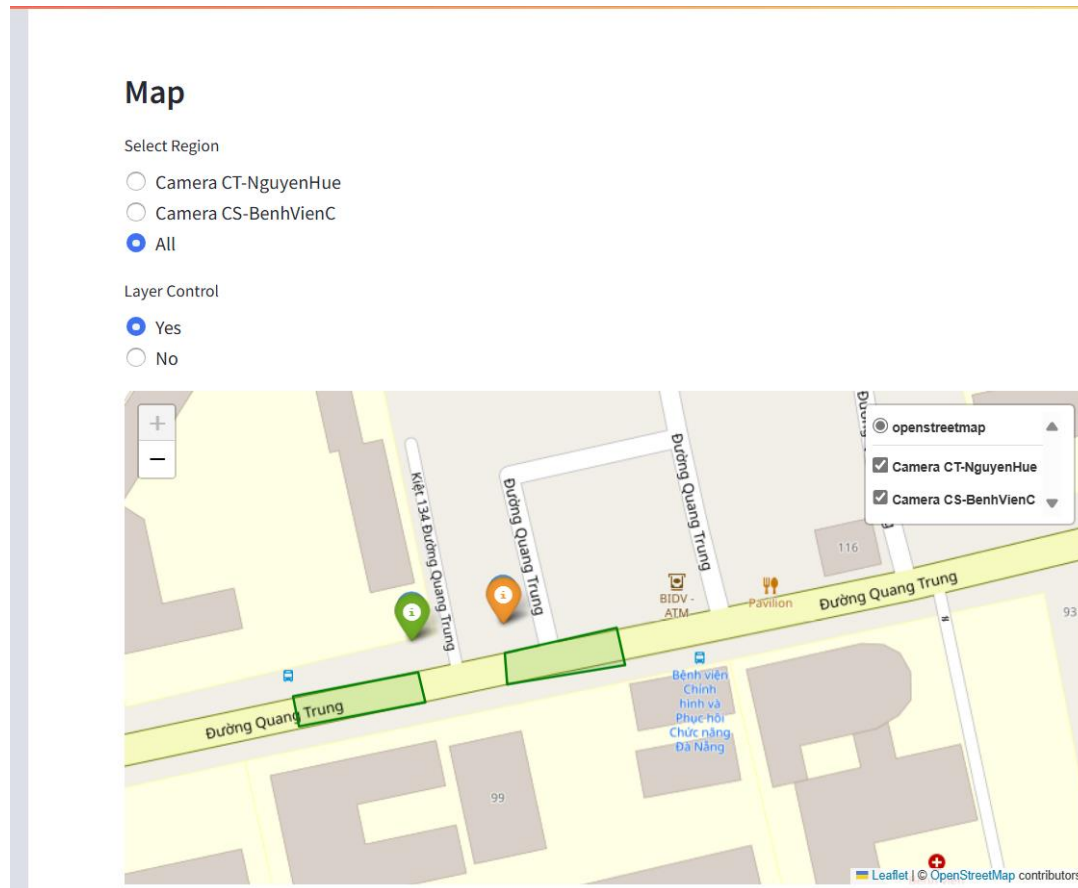
Hình 30: Cấu hình tham số đầu vào trang Multi Camera Monitor



Hình 31: camera giám sát số 1



Hình 32: camera giám sát số 2



Hình 33: Bản đồ hiển thị 2 camera giám sát và mật độ lưu thông tại đường Nguyễn Huệ

- Hình 15 Cấu hình tham số đầu vào như lựa chọn mô hình yolov9 hoặc yolov8 các chỉ số confidence, iou và lựa chọn số lượng camera để giám sát đồng thời.
- Hình 16 và Hình 17 cho thấy kết quả sau thực hiện giám sát đồng thời
- Hình 18: Bản đồ hiển thị 2 camera giám sát và mật độ lưu thông tại đường Nguyễn Huệ

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Đánh giá

1.1. Kết quả đạt được

- Tập dữ liệu: Thu thập được 5 class với dữ liệu đa dạng (vị trí, kích thước, sáng tối, đơn lẻ, hỗn hợp,...)
- Mô hình: Áp dụng được mô hình YOLO vào bài toán thực hiện nhận diện phương tiện giao thông.
- Phần mềm: Đủ các chức năng cơ bản để người dùng có thể thao tác sử dụng để nhận diện phương tiện và quan sát sự nhận diện đó.
- Thông qua việc áp dụng các thuật toán học máy tiên tiến, hệ thống của chúng tôi đã có thể nhận diện chính xác và nhanh chóng mật độ giao thông trên các tuyến đường. Đặc biệt, những thuật toán như phân loại đối tượng và phân tích hình ảnh đã chứng minh được tính hiệu quả cao trong việc phân biệt giữa các loại phương tiện khác nhau và đếm số lượng phương tiện di chuyển qua từng điểm quan sát. Hệ thống này không chỉ giúp cung cấp thông tin tức thời về tình trạng giao thông mà còn hỗ trợ trong việc dự báo và phân tích xu hướng giao thông trong tương lai.

1.2. Hạn chế

- Mô hình nhận diện hiện tại chưa hoàn thiện. Tốc độ thực nghiệm chưa được tốt (fps khá thấp), độ chính xác trong các điều kiện phức tạp như ánh sáng kém, thời tiết xấu, hoặc phương tiện bị che khuất một phần chưa cao.
- Phần mềm: chỉ dừng ở mức cơ bản, đủ dùng; chưa có nhiều chức năng mở rộng

2. Hướng phát triển

- Tập dữ liệu: Thu thập nhiều phương tiện (class) hơn với nhiều kiểu dữ liệu đa dạng.
- Mô hình: Tiếp tục huấn luyện mô hình để nâng cao độ chính xác nhận diện. Ngoài ra, tìm hiểu và xây dựng nhiều mô hình khác để tìm ra mô hình thích hợp với bài toán.
- App: Xây dựng app có giao diện đẹp hơn. Đồng thời, mở rộng ra nhiều chức năng phụ (đánh giá, học về các loại phương tiện,..) và tích hợp trên nhiều nền tảng.

TÀI LIỆU THAM KHẢO

- [1]. Chien-Yao Wang, I-Hau Yeh, Hong-Yuan Mark Liao, “YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information”, arXiv:2402.13616.
- [2]. Dr. Priyanto Hidayatullah, “YOLOv9 Architecture Explained”
- [3]. Laughing-q , glenn-jocher , RizwanMunawar , Burhan-Q, “YOLOv9: A Leap Forward in Object Detection Technology”
- [4]. Mert, “How to use YOLOv9 for Object Detection”, Mar 13, 2024.

- [5]. Bui Tien Tung, “SORT - Deep SORT : Một góc nhìn về Object Tracking (phần 1)”, Dec 16, 2020.
- [6]. “Hungarian algorithm”, https://en.wikipedia.org/wiki/Hungarian_algorithm
- [7]. “Bộ lọc Kalman”,
https://vi.wikipedia.org/wiki/B%E1%BB%99_l%E1%BB%8Dc_Kalman
- [8]. Trường Sơn, “ByteTrack: Multi-object detector hiện đại nhất hiện nay”
- [9]. Pyresearch, Real-Time Object Counting with YOLOv9 and Supervision, Apr 21, 2024.
- [10]. Đặng, Thị Mỹ Nhân. Ứng dụng Deep Learning để đếm số lượng xe ô tô trong nội thành Đà Nẵng. Diss. Trường Đại học Bách khoa-Đại học Đà Nẵng, 2019.
- [11]. Minh, Khoa Truong, et al. "Vehicle Counting on Vietnamese Street." 2023 IEEE Statistical Signal Processing Workshop (SSP). IEEE, 2023.
- [12]. Sadik, Md Nahid, Tahmim Hossain, and Faisal Sayeed. "Real-Time Detection and Analysis of Vehicles and Pedestrians using Deep Learning." arXiv preprint arXiv:2404.08081 (2024).
- [13]. Trinh, Truc, and Khang Nguyen. "A Vietnamese benchmark for vehicle detection and real-time empirical evaluation." CTU Journal of Innovation and Sustainable Development 14.3 (2022): 45-52.
- [14]. Nguyen, Nhu Phuc, et al. "AI-based traffic counting: A Case Study in Vietnam." 2022 International Conference on Advanced Computing and Analytics (ACOMPA). IEEE, 2022.
- [15]. Du, Yunhao, et al. "Strongsort: Make deepsort great again." IEEE Transactions on Multimedia (2023).

PHỤ LỤC