HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Nguyễn Nam Thắng

NGHIÊN CỨU ỨNG DỤNG CÔNG NGHỆ XỬ LÝ HÌNH ẢNH TRONG QUẢN LÝ LƯU LƯỢNG GIAO THÔNG ĐÔ THỊ

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

Hà Nội - 2025

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Nguyễn Nam Thắng

NGHIÊN CỨU ỨNG DỤNG CÔNG NGHỆ XỬ LÝ HÌNH ẢNH TRONG QUẢN LÝ LƯU LƯỢNG GIAO THÔNG ĐÔ THỊ

CHUYÊN NGÀNH: HỆ THỐNG THÔNG TIN

MÃ SỐ: 8.48.01.04

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC: TS. ĐỖ THỊ LIÊN

Hà Nội - 2025

LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi và không sao chép từ bất kỳ nguồn nào khác.

Các số liệu, kết quả nêu trong đề án tốt nghiệp là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác. Tất cả thông tin được trình bày trong đề án này đều là sản phẩm của công việc cá nhân hoặc được tổng hợp từ nhiều nguồn tài liệu khác nhau. Mọi tài liệu tham khảo đều được trích dẫn một cách hợp pháp và có nguồn gốc rõ ràng.

Tác giả đề án tốt nghiệp ký và ghi rõ họ tên

Nguyễn Nam Thắng

LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn sâu sắc tới các thầy, cô giáo giảng viên khoa Công nghệ thông tin 1, khoa Đào tạo Sau đại học nói riêng và các thầy, cô giáo giảng viên Học viện Công nghệ Bưu chính Viễn Thông nói chung. Trong suốt quá trình học tập tại Học viện, các thầy cô đã chỉ bảo, giảng dạy cho em biết bao kiến thức, kinh nghiệm quý báu để em có hành trang vững bước trong tương lai.

Em cũng xin được gửi lời cảm ơn tới thầy/cô hướng dẫn TS. Đỗ Thị Liên cảm ơn cô đã luôn hướng dẫn chỉ bảo tận tình em trong suốt quá trình học tập, nghiên cứu và thực hiện đề án này. Những lời khuyên, sự chỉ bảo của cô đã giúp em hoàn thành đề án tốt nghiệp này cũng như có thêm rất nhiều kiến thức, kinh nghiệm trong việc học tập và nghiên cứu.

Dù đã nỗ lực hoàn thành đề án, em hiểu rằng có thể không tránh khỏi những sai sót. Kính mong được thầy cô và các bạn thông cảm và đóng góp ý kiến.

Em xin trân trọng cảm ơn.

MỤC LỤC

DANH MỤC HÌNH

DANH MỤC CÔNG THỨC

DANH MỤC BẢNG

MỞ ĐẦU

1. Lý do chọn đề tài

Trong bối cảnh đô thị hóa diễn ra nhanh chóng tại Việt Nam, đặc biệt ở các thành phố lớn như Hà Nội và TP. Hồ Chí Minh, vấn đề giao thông đô thị ngày càng trở nên nghiêm trọng. Sự gia tăng dân số cùng với mật độ phương tiện giao thông cao đã dẫn đến tình trạng ùn tắc kéo dài, tai nạn giao thông gia tăng và hiệu quả quản lý lưu lượng giao thông giảm sút. Theo thống kê của Bộ Giao thông Vận tải Việt Nam, chỉ riêng tại TP. Hà Nội, số lượng phương tiện đăng ký mới trong năm 2023 đã vượt mốc 230 nghìn xe, trong đó phần lớn là xe máy và ô tô cá nhân [1]. Các hệ thống giao thông truyền thống hiện tại, chủ yếu dựa vào con người và các phương pháp thủ công, không đủ khả năng phản ứng nhanh và chính xác trong môi trường đô thị phức tạp, dẫn đến hiệu quả quản lý giao thông giảm sút.

Công nghệ xử lý hình ảnh, kết hợp với trí tuệ nhân tạo (AI), đã chứng minh tiềm năng vượt trội trong việc giải quyết các vấn đề giao thông đô thị. Các hệ thống giám sát giao thông thông minh (Intelligent Transportation Systems - ITS) sử dụng camera và thuật toán học sâu (deep learning) có thể tự động phát hiện phương tiện, dự đoán lưu lượng giao thông, và hỗ trợ điều phối tín hiệu đèn giao thông một cách hiệu quả, từ đó giảm thiểu ùn tắc và tai nạn [2]. Những nghiên cứu gần đây cho thấy rằng việc ứng dụng xử lý hình ảnh không chỉ cải thiện độ chính xác trong phát hiện vi phạm giao thông mà còn giúp tối ưu hóa lưu lượng, giảm thiểu ùn tắc và nâng cao an toàn giao thông [3]. Tuy nhiên, tại Việt Nam, việc triển khai các giải pháp này vẫn còn hạn chế do đặc thù giao thông phức tạp, dữ liệu thực tế chưa được khai thác tối ưu, và thiếu các hệ thống tùy chỉnh phù hợp với điều kiện địa phương.

Xuất phát từ thực trạng trên, đề tài "Nghiên cứu ứng dụng công nghệ xử lý hình ảnh trong quản lý lưu lượng giao thông đô thị" được lựa chọn nhằm phát triển một giải pháp tích hợp, tận dụng công nghệ xử lý hình ảnh và AI để giải quyết các vấn đề giao thông đô thị tại Việt Nam. Đề tài không chỉ tập trung vào lý thuyết mà còn hướng

đến việc triển khai thực tế, với mục tiêu xây dựng một hệ thống quản lý lưu lượng giao thông hiệu quả, khả thi và phù hợp với bối cảnh đô thị Việt Nam.

2. Tổng quan về vấn đề nghiên cứu

Nghiên cứu về ứng dụng công nghệ xử lý hình ảnh trong quản lý lưu lượng giao thông đô thị là một lĩnh vực đa ngành, liên quan đến Giao thông thông minh (ITS), Xử lý hình ảnh (Image Processing), Thị giác máy tính (Computer Vision), và Trí tuệ nhân tạo (AI). Trên thế giới, nhiều công trình đã đạt được những kết quả đáng chú ý trong việc cải thiện lưu lượng giao thông thông qua các phương pháp này.

Wang và Li đã nghiên cứu việc kết hợp xử lý hình ảnh với học máy để dự đoán và tối ưu hóa lưu lượng giao thông đô thị [4]. Hệ thống của họ phân tích video từ camera giao thông để dự đoán tình trạng ùn tắc và tự động điều chỉnh tín hiệu đèn giao thông, giúp giảm thời gian chờ trung bình lên đến 20% tại các giao lộ lớn [4]. Tương tự, Kim và Park đã phát triển một hệ thống phát hiện và theo dõi phương tiện dựa trên học sâu, sử dụng mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) để nhận diện xe trong điều kiện ánh sáng yếu, đạt độ chính xác trên 90% [5]. Gần đây hơn, Li và Shen đã tích hợp dữ liệu từ camera và thiết bị IoT với mô hình AI để xây dựng hệ thống dự báo lưu lượng giao thông thời gian thực, mở ra hướng đi cho các thành phố thông minh [6].

Mặc dù vậy, các giải pháp trên vẫn tồn tại nhiều hạn chế khi áp dụng vào thực tế, đặc biệt trong môi trường giao thông phức tạp như Việt Nam. Các nghiên cứu trước đây thường tập trung vào phát triển hệ thống giám sát tại các quốc gia có hạ tầng giao thông đồng bộ, trong khi giao thông đô thị Việt Nam lại mang đặc điểm riêng với mật độ xe máy cao, hành vi lái xe không đồng nhất, và điều kiện thời tiết biến đổi [7]. Ngoài ra, khả năng xử lý dữ liệu hình ảnh trong thời gian thực và tích hợp với hạ tầng giao thông hiện có vẫn là thách thức lớn. Do đó, cần thiết phải phát triển các giải pháp tùy chỉnh, vừa tận dụng công nghệ tiên tiến vừa phù hợp với đặc thù giao thông đô thị Việt Nam.

3. Mục tiêu nghiên cứu

Mục đích chính của đề tài là nghiên cứu và ứng dụng công nghệ xử lý hình ảnh để quản lý lưu lượng giao thông đô thị một cách hiệu quả, với các mục tiêu cụ thể như sau:

1. Tối ưu hóa quản lý lưu lượng giao thông đô thị: Phát triển các phương pháp phân tích dữ liệu hình ảnh từ camera để dự đoán và điều chỉnh lưu lượng giao thông, giảm ùn tắc tại các khu vực có mật độ phương tiện cao.

2. Tự động hóa điều phối giao thông: Xây dựng giải pháp tự động phát hiện phương tiện, đếm lưu lượng, và hỗ trợ điều chỉnh tín hiệu giao thông hoặc tuyến đường, cải thiện sự thông suốt tại các giao lộ.

3. Ứng dụng thực tế tại Việt Nam: Triển khai hệ thống phù hợp với điều kiện giao thông đô thị Việt Nam, đặc biệt tại Hà Nội và TP. Hồ Chí Minh, nhằm nâng cao an toàn và hiệu quả giao thông.

4. Đề xuất giải pháp bền vững: Đưa ra một hệ thống quản lý giao thông thông minh có khả năng mở rộng, tạo nền tảng cho các đô thị thông minh trong tương lai.

4. Đối tượng và phạm vi nghiên cứu

4.1. Đối tượng nghiên cứu

Đề tài tập trung vào các hệ thống và công nghệ liên quan đến quản lý lưu lượng giao thông đô thị thông qua xử lý hình ảnh và AI, bao gồm:

- Các thuật toán xử lý hình ảnh và học sâu để phát hiện phương tiện và dự đoán lưu lượng.
- Hệ thống giám sát giao thông tích hợp dữ liệu từ camera và cảm biến.
- Giải pháp tự động điều phối giao thông dựa trên phân tích thời gian thực.

4.2. Phạm vi nghiên cứu

- Không gian: Các khu vực đô thị lớn tại Việt Nam, đặc biệt là các giao lộ và tuyến đường huyết mạch tại Hà Nội và TP. Hồ Chí Minh, nơi thường xuyên xảy ra ùn tắc.

- Thời gian: Sử dụng dữ liệu giao thông từ camera trong 2-3 năm gần đây (2022-2024), kết hợp với các nghiên cứu mới nhất về xử lý hình ảnh và học sâu.

- Ứng dụng: Đề tài không chỉ dừng ở lý thuyết mà còn triển khai thực nghiệm hệ thống quản lý lưu lượng giao thông phù hợp với điều kiện hạ tầng và đặc điểm giao thông Việt Nam.

5. Phương pháp nghiên cứu

Phương pháp nghiên cứu được xây dựng dựa trên các bước cơ bản sau:

1. Nghiên cứu tài liệu: Tổng hợp các công trình nghiên cứu liên quan từ các nguồn quốc tế [4] [5] [6] và khảo sát thực trạng giao thông đô thị tại Việt Nam để xác định vấn đề cần giải quyết.

2. Thu thập và phân tích dữ liệu: Sử dụng dữ liệu hình ảnh/video từ camera giao thông, thực hiện tiền xử lý (chuẩn hóa, giảm nhiễu) để chuẩn bị cho phân tích.

3. Ứng dụng công nghệ: Phát triển hệ thống dựa trên mô hình YOLOv8 [8] và các kỹ thuật xử lý hình ảnh (OpenCV), kết hợp với giao diện người dùng (PyQt5) để giám sát và điều phối giao thông.

4. Triển khai và thử nghiệm: Xây dựng hệ thống thực nghiệm, thử nghiệm tại các khu vực giao thông đô thị, và đánh giá hiệu quả dựa trên độ chính xác, tốc độ xử lý, và khả năng phản ứng thực tế.

5. Đánh giá và cải tiến: Phân tích kết quả thử nghiệm để đề xuất các cải tiến, đảm bảo tính khả thi và hiệu quả của hệ thống.

CHƯƠNG I: TỔNG QUAN VỀ HỆ THỐNG QUẢN LÝ LƯU LƯỢNG GIAO THÔNG ĐÔ THỊ

Trong bối cảnh các đô thị lớn ngày càng chịu áp lực từ tình trạng quá tải giao thông, nhu cầu xây dựng các hệ thống giám sát và điều phối lưu lượng phương tiện một cách hiệu quả ngày càng trở nên cấp thiết. Việc ứng dụng các công nghệ hiện đại như xử lý hình ảnh, học sâu và trí tuệ nhân tạo đã mở ra nhiều hướng tiếp cận mới, góp phần nâng cao hiệu quả quản lý giao thông trong điều kiện thực tế phức tạp.

Chương này sẽ trình bày tổng quan về bài toán quản lý lưu lượng giao thông đô thị, các mô hình kiến trúc hệ thống hiện nay, các hướng tiếp cận phổ biến trong nghiên cứu và triển khai, cùng với những công nghệ cốt lõi được sử dụng trong lĩnh vực này. Ngoài ra, chương cũng phân tích những thách thức đặc thù trong bối cảnh Việt Nam nhằm làm cơ sở cho việc đề xuất một hệ thống phù hợp và khả thi

1.1. Phát biểu bài toán quản lý lưu lượng giao thông đô thị

Lưu lượng giao thông là đại lượng thể hiện số phương tiện di chuyển qua một điểm, một đoạn đường hoặc vùng nhất định trong một khoảng thời gian, thường được đo bằng đơn vị phương tiện/giờ hoặc phương tiện/phút. Trong bối cảnh đô thị hiện đại, lưu lượng giao thông là chỉ số quan trọng giúp đánh giá mức độ hoạt động, tình trạng quá tải và hiệu suất vận hành của hệ thống giao thông. Khi được tích hợp với công nghệ xử lý hình ảnh, lưu lượng có thể được tính toán thông qua việc phát hiện và theo dõi phương tiện từ dữ liệu camera, phân tích số lượng phương tiện trong vùng quan tâm (ROI) theo thời gian. Việc ứng dụng thị giác máy tính để đo lưu lượng không chỉ nâng cao độ chính xác, phản ứng thời gian thực mà còn hỗ trợ hiệu quả cho các chức năng như điều phối đèn giao thông, dự đoán ùn tắc và cảnh báo sớm tình trạng quá tải.

Giao thông đô thị tại các thành phố lớn ở Việt Nam, như Hà Nội và TP. Hồ Chí Minh, từ lâu đã trở thành một bài toán nan giải đối với cả người dân và các nhà quản lý. Với sự gia tăng nhanh chóng của dân số và phương tiện giao thông, tình trạng ùn

tắc không chỉ gây mất thời gian mà còn làm gia tăng ô nhiễm môi trường, tai nạn giao thông, và ảnh hưởng tiêu cực đến chất lượng cuộc sống. Theo thống kê từ Bộ Giao thông Vận tải Việt Nam, chỉ riêng trong năm 2023, TP. Hà Nội đã ghi nhận hơn 230 nghìn phương tiện đăng ký mới, trong đó xe máy chiếm tỷ lệ lớn nhất, lên đến 75% [1]. Điều này tạo ra một áp lực khổng lồ lên hạ tầng giao thông vốn đã quá tải, đặc biệt tại các giao lộ chính và tuyến đường huyết mạch.

Bài toán quản lý lưu lượng giao thông đô thị có thể được phát biểu như sau: Làm thế nào để giám sát, phân tích và điều phối lưu lượng phương tiện một cách hiệu quả nhằm giảm thiểu ùn tắc, tối ưu hóa thời gian di chuyển, và nâng cao an toàn giao thông trong điều kiện đô thị phức tạp?

Để giải quyết bài toán này, cần đáp ứng một số yêu cầu cơ bản.

1. Đầu vào:
    - Dữ liệu đầu vào của hệ thống bao gồm hình ảnh/video thời gian thực từ các camera giao thông được lắp đặt tại các giao lộ và tuyến đường huyết mạch. Các nguồn dữ liệu này có thể là video trực tiếp hoặc các video ghi lại từ các buổi thử nghiệm trước đó. Dữ liệu này sẽ được xử lý để phát hiện và nhận diện các phương tiện giao thông (xe máy, ô tô, xe tải) và người đi bộ trong các điều kiện môi trường khác nhau như ánh sáng yếu, thời tiết xấu, hay mật độ giao thông cao.
2. Mục tiêu xử lý của hệ thống:
    - Phát hiện và nhận diện phương tiện giao thông: Xử lý hình ảnh/video để nhận diện các phương tiện như xe máy, ô tô, xe tải và người đi bộ. Việc phát hiện phải diễn ra trong thời gian thực, ngay cả khi mật độ giao thông cao hoặc trong điều kiện ánh sáng thay đổi.
    - Dự đoán lưu lượng giao thông: Phân tích số lượng phương tiện trong các khu vực trọng điểm (Region of Interest - ROI) để dự đoán tình trạng giao thông, xác định mức độ ùn tắc hoặc tắc nghẽn có thể xảy ra trong thời gian tới.

- Điều phối và tối ưu hóa giao thông: Tự động điều chỉnh tín hiệu đèn giao thông (hoặc đề xuất các tuyến đường thay thế) dựa trên phân tích lưu lượng giao thông thực tế, nhằm giảm thiểu ùn tắc, tối ưu hóa lưu lượng giao thông và tăng hiệu quả di chuyển.
- Gửi cảnh báo về ùn tắc: Khi tình trạng ùn tắc hoặc tắc nghẽn giao thông xảy ra, hệ thống sẽ tự động gửi cảnh báo đến người quản lý giao thông qua các nền tảng như Telegram và cung cấp các giải pháp điều phối giao thông.

3. Đầu ra:

- Các kết quả đầu ra của hệ thống bao gồm:
- Số lượng phương tiện trong các khu vực trọng điểm (ROI) trong thời gian thực.
- Dự báo mức độ ùn tắc trong các khu vực dựa trên phân tích lưu lượng giao thông.
- Cảnh báo về tình trạng ùn tắc và đề xuất điều chỉnh tín hiệu giao thông (hoặc tuyến đường thay thế).
- Thông tin trực quan qua giao diện người dùng (GUI) với các biểu đồ, số liệu về lưu lượng giao thông.

4. Ví dụ minh họa:

- Giả sử tại một giao lộ lớn ở Hà Nội vào giờ cao điểm, hệ thống nhận được video từ camera giao thông. Hệ thống phát hiện 15 xe máy, 10 ô tô, và 5 xe tải trong một khu vực trọng điểm (ROI) trong vòng 1 phút. Hệ thống phân tích lưu lượng giao thông và dự đoán ùn tắc sẽ xảy ra nếu không có biện pháp điều phối. Dựa trên phân tích này, hệ thống tự động điều chỉnh tín hiệu đèn giao thông (ví dụ: tăng thời gian xanh cho xe máy) và gửi cảnh báo tình trạng ùn tắc qua Telegram cho người quản lý giao thông, giúp giảm tình trạng tắc nghẽn và tối ưu hóa việc di chuyển tại giao lộ đó.

Những yêu cầu này không phải là mới trên thế giới. Các nghiên cứu quốc tế đã chỉ ra rằng quản lý lưu lượng giao thông hiệu quả đòi hỏi sự kết hợp giữa công nghệ

giám sát và các thuật toán thông minh. Chẳng hạn, Wang và Li đã nhấn mạnh vai trò của xử lý hình ảnh trong việc phân tích video từ camera giao thông để dự đoán tình trạng ùn tắc, từ đó hỗ trợ điều phối tự động [4]. Tuy nhiên, tại Việt Nam, giao thông đô thị mang những đặc điểm riêng biệt khiến bài toán trở nên phức tạp hơn. Sự pha trộn giữa xe máy, ô tô và các phương tiện khác, cùng với hành vi lái xe không đồng nhất, tạo ra những thách thức mà các hệ thống truyền thống khó lòng đáp ứng [7]. Hơn nữa, các giải pháp hiện tại tại Việt Nam, như sử dụng camera quan sát thủ công hoặc đèn giao thông cố định, thường thiếu tính linh hoạt và không thể phản ứng nhanh trước những thay đổi đột ngột của lưu lượng giao thông.

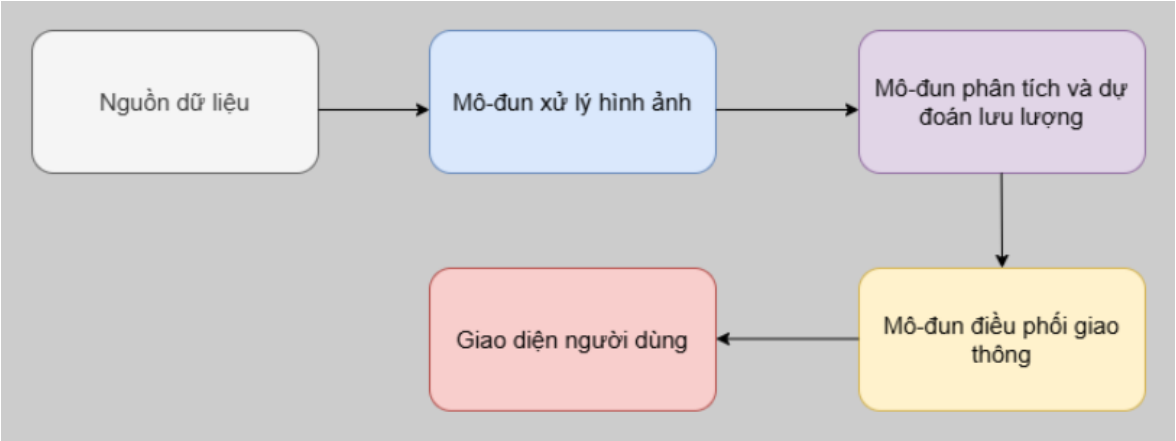
Một ví dụ điển hình là tình trạng ùn tắc kéo dài tại các giao lộ lớn như ngã tư Hàng Xanh (TP. Hồ Chí Minh) hay ngã tư Kim Mã - Nguyễn Chí Thanh (Hà Nội) vào giờ cao điểm. Những khu vực này không chỉ chịu áp lực từ số lượng phương tiện lớn mà còn từ sự thiếu đồng bộ trong điều phối tín hiệu giao thông. Nghiên cứu của Shen và Wei cho thấy rằng việc tích hợp dữ liệu hình ảnh từ camera với các mô hình AI có thể giảm thời gian chờ tại giao lộ lên đến 15-20% [3]. Điều này gợi ý rằng một hệ thống quản lý lưu lượng giao thông thông minh, dựa trên xử lý hình ảnh và trí tuệ nhân tạo, có thể là chìa khóa để giải quyết bài toán tại Việt Nam.

1.2. Kiến trúc hệ thống quản lý lưu lượng giao thông đô thị

Hệ thống quản lý lưu lượng giao thông đô thị là một giải pháp tích hợp nhiều thành phần công nghệ nhằm giám sát, phân tích và điều phối giao thông một cách hiệu quả. Trong bối cảnh đô thị Việt Nam, nơi mật độ phương tiện cao và hạ tầng giao thông còn nhiều hạn chế, kiến trúc hệ thống cần được thiết kế linh hoạt, có khả năng xử lý dữ liệu từ nhiều nguồn khác nhau như camera giao thông và video ghi sẵn, đồng thời đảm bảo tính phản ứng nhanh trong thời gian thực. Phần này sẽ trình bày chi tiết các thành phần chính, quy trình hoạt động cơ bản, và các mô hình kiến trúc phổ biến của hệ thống quản lý lưu lượng giao thông đô thị.

1.2.1. Thành phần của hệ thống

Một hệ thống quản lý lưu lượng giao thông đô thị thông minh thường bao gồm các thành phần chính sau:



Hình 1. 1 Thành phần hệ thống quản lý

- Nguồn dữ liệu (Camera và Video): Đây là nguồn đầu vào quan trọng, cung cấp dữ liệu hình ảnh hoặc video để phân tích. Camera giao thông được lắp đặt tại các giao lộ và tuyến đường huyết mạch có thể ghi lại hình ảnh thời gian thực, trong khi các tệp video ghi sẵn (ví dụ: từ các nguồn thử nghiệm hoặc dữ liệu lịch sử) cho phép mô phỏng và đánh giá hệ thống trong các tình huống khác nhau. Nghiên cứu của Wang và Li đã chỉ ra rằng việc sử dụng dữ liệu từ nhiều nguồn như vậy giúp tăng độ chính xác trong phân tích lưu lượng giao thông lên đến 15% so với chỉ dùng camera cố định [4].

- Mô-đun xử lý hình ảnh: Thành phần này chịu trách nhiệm xử lý dữ liệu đầu vào để phát hiện và nhận diện các phương tiện (xe máy, ô tô, xe tải, v.v.). Các thuật toán học sâu, chẳng hạn như YOLOv8, được sử dụng để phân tích hình ảnh hoặc video, cho phép nhận diện đối tượng với tốc độ cao và độ chính xác đáng kể [8]. Trong thực nghiệm của đề tài, mô-đun này không chỉ phát hiện phương tiện mà còn theo dõi (tracking) chúng qua các khung hình, như đã thực hiện trong mã nguồn với SimpleTracker.

- Mô-đun phân tích và dự đoán lưu lượng: Sau khi xử lý hình ảnh, hệ thống cần phân tích lưu lượng giao thông tại các khu vực cụ thể (Region of Interest - ROI) để đếm số lượng phương tiện và dự đoán tình trạng giao thông. Các nghiên cứu như của Shen và Wei đã tích hợp dữ liệu từ camera với IoT để dự báo lưu lượng, đạt hiệu quả giảm ùn tắc lên đến 20% tại các giao lộ lớn [3]. Trong hệ thống đề xuất, dữ liệu từ cả camera và video được phân tích để xác định mật độ phương tiện, hỗ trợ điều phối giao thông.

- Mô-đun điều phối giao thông: Dựa trên kết quả phân tích, mô-đun này đưa ra các quyết định điều phối, chẳng hạn như điều chỉnh thời gian tín hiệu đèn giao thông hoặc gửi cảnh báo ùn tắc qua Telegram. Kim và Park đã chứng minh rằng hệ thống điều phối tự động dựa trên xử lý hình ảnh có thể giảm thời gian chờ trung bình tại các giao lộ xuống dưới 30 giây [5]. Trong nghiên cứu này, hệ thống tích hợp cảnh báo qua Telegram để thông báo tình trạng ùn tắc cho người quản lý.

- Giao diện người dùng (GUI): Thành phần này cho phép người dùng giám sát trực quan dữ liệu giao thông, vẽ ROI, và nhận thông báo. Ứng dụng GUI trong nghiên cứu được xây dựng bằng PyQt5, hỗ trợ cả việc load video từ tệp và dữ liệu từ camera thời gian thực, mang lại tính linh hoạt trong triển khai.

1.2.2. Quy trình hoạt động cơ bản

Quy trình hoạt động của hệ thống quản lý lưu lượng giao thông đô thị có thể được mô tả qua các bước sau:

1. Thu thập dữ liệu: Dữ liệu hình ảnh hoặc video được thu thập từ camera giao thông hoặc các tệp video đã ghi sẵn. Ví dụ, trong thực nghiệm, video mẫu từ TP. Hồ Chí Minh (vecteezy_ho-chi-minh-city-traffic-at-intersection-vietnam) được sử dụng để kiểm tra hiệu quả hệ thống.
2. Xử lý và nhận diện: Dữ liệu đầu vào được đưa qua mô-đun xử lý hình ảnh sử dụng YOLOv8 để phát hiện các phương tiện. Quá trình này bao gồm tiền xử lý (chuẩn hóa kích thước 640x640), nhận diện đối tượng, và theo dõi qua các khung hình [8].

3. Phân tích lưu lượng: Dữ liệu phương tiện được phân tích để đếm số lượng trong các ROI do người dùng định nghĩa. Hệ thống ghi nhận số lượng phương tiện theo thời gian thực và lưu trữ dữ liệu để dự đoán xu hướng lưu lượng.

4. Điều phối và cảnh báo: Dựa trên kết quả phân tích, hệ thống tự động gửi cảnh báo ùn tắc qua Telegram nếu số lượng phương tiện vượt ngưỡng (ví dụ: 15 phương tiện trong 60 giây) hoặc đề xuất điều chỉnh tín hiệu giao thông. Quy trình này được thực hiện hoàn toàn tự động, đảm bảo phản ứng nhanh trong điều kiện đô thị phức tạp.

5. Hiển thị và giám sát: Kết quả được hiển thị trên giao diện GUI, cho phép người dùng theo dõi trực quan và can thiệp thủ công nếu cần. Hệ thống hỗ trợ cả việc load video từ tệp để phân tích lịch sử và dữ liệu từ camera để giám sát thời gian thực.

1.3. Các hướng tiếp cận xây dựng hệ thống quản lý lưu lượng giao thông đô thị

Trên thế giới, nhiều mô hình kiến trúc đã được phát triển để quản lý lưu lượng giao thông đô thị, mỗi mô hình có ưu điểm và hạn chế riêng:

- Hệ thống giám sát truyền thống: Dựa trên camera cố định và phân tích thủ công, mô hình này thường được sử dụng tại các đô thị Việt Nam nhưng thiếu tính tự động và khả năng phản ứng nhanh. Nghiên cứu của Hai chỉ ra rằng các hệ thống này chỉ hiệu quả trong điều kiện lưu lượng thấp, không đáp ứng được giờ cao điểm [7].

- Hệ thống dựa trên IoT và cảm biến: Tích hợp cảm biến giao thông với camera để thu thập dữ liệu đa nguồn. Li và Shen đã triển khai mô hình này tại các thành phố thông minh, đạt hiệu quả cao trong dự báo lưu lượng [6]. Tuy nhiên, chi phí lắp đặt cảm biến và yêu cầu hạ tầng đồng bộ khiến mô hình này khó áp dụng rộng rãi tại Việt Nam.

- Hệ thống dựa trên xử lý hình ảnh và AI: Đây là mô hình tiên tiến nhất, sử dụng camera hoặc video kết hợp với các thuật toán học sâu như YOLO hoặc SSD để phân tích giao thông. Wang và Li đã chứng minh rằng mô hình này có thể giảm ùn

tắc tại các giao lộ lớn nhờ khả năng tự động hóa [4]. Kiến trúc đề xuất trong nghiên cứu này thuộc nhóm này, với điểm nhấn là tích hợp cả camera và video, sử dụng YOLOv8 làm nền tảng [8], và tối ưu hóa cho đặc thù giao thông Việt Nam.

So với các mô hình trên, kiến trúc hệ thống trong nghiên cứu này có ưu điểm là tính linh hoạt (hỗ trợ cả video và camera), chi phí triển khai thấp (không cần cảm biến IoT), và khả năng tùy chỉnh theo điều kiện giao thông đô thị Việt Nam. Hệ thống không chỉ phát hiện phương tiện mà còn cung cấp công cụ điều phối và cảnh báo, tạo ra một giải pháp toàn diện cho quản lý lưu lượng giao thông.

Để giải quyết bài toán quản lý lưu lượng giao thông đô thị, các nhà nghiên cứu và kỹ sư trên thế giới đã phát triển nhiều hướng tiếp cận khác nhau, từ các phương pháp truyền thống dựa trên cảm biến vật lý đến các giải pháp hiện đại ứng dụng công nghệ xử lý hình ảnh và trí tuệ nhân tạo (AI). Trong bối cảnh Việt Nam, với đặc thù giao thông phức tạp và hạ tầng chưa đồng bộ, việc lựa chọn hướng tiếp cận phù hợp là yếu tố quan trọng để đảm bảo hiệu quả và tính khả thi. Phần này sẽ trình bày các hướng tiếp cận chính, đồng thời phân tích ưu điểm và hạn chế của chúng khi áp dụng vào thực tế đô thị Việt Nam, từ đó làm nền tảng cho giải pháp đề xuất trong nghiên cứu này.

1.3.1. Hướng tiếp cận dựa trên cảm biến và IoT

Một trong những hướng tiếp cận sớm nhất để quản lý lưu lượng giao thông là sử dụng các cảm biến vật lý (như cảm biến từ trường, cảm biến áp suất) và công nghệ Internet vạn vật (IoT). Các cảm biến được lắp đặt dưới mặt đường hoặc tại các giao lộ để đo đếm số lượng phương tiện, từ đó cung cấp dữ liệu cho hệ thống điều phối tín hiệu đèn giao thông. Nghiên cứu của Li và Shen đã triển khai một hệ thống IoT tích hợp cảm biến và camera tại các thành phố thông minh, cho phép dự báo lưu lượng giao thông với độ chính xác cao, giảm thời gian chờ tại giao lộ xuống 15-20% [6]. Hệ thống này sử dụng dữ liệu đa nguồn để phân tích xu hướng giao thông và tối ưu hóa lưu lượng theo thời gian thực.

Ưu điểm của hướng tiếp cận này là khả năng cung cấp dữ liệu chính xác về số lượng phương tiện mà không phụ thuộc vào điều kiện ánh sáng hay thời tiết. Tuy nhiên, nhược điểm lớn nằm ở chi phí lắp đặt và bảo trì cao, đặc biệt khi triển khai trên diện rộng. Tại Việt Nam, với hạ tầng giao thông cũ và ngân sách hạn chế, việc áp dụng các hệ thống cảm biến IoT gặp nhiều khó khăn. Ngoài ra, cảm biến chỉ cung cấp thông tin định lượng (số lượng xe) mà không thể nhận diện loại phương tiện hay phân tích tình huống giao thông phức tạp, như hành vi vi phạm hoặc ùn tắc do tai nạn [7].

1.3.2. Hướng tiếp cận dựa trên xử lý hình ảnh truyền thống

Hướng tiếp cận thứ hai sử dụng các kỹ thuật xử lý hình ảnh truyền thống, chẳng hạn như phát hiện cạnh (edge detection), phân đoạn hình ảnh (image segmentation), và nhận diện mẫu (template matching), để giám sát giao thông từ dữ liệu camera hoặc video. Các hệ thống này thường dựa trên thư viện OpenCV để phân tích hình ảnh, đếm phương tiện, và phát hiện các sự kiện giao thông cơ bản như ùn tắc hoặc vượt đèn đỏ. Một ví dụ điển hình là nghiên cứu của Kim và Park, trong đó họ sử dụng xử lý hình ảnh để phát hiện phương tiện với độ chính xác khoảng 85% trong điều kiện ánh sáng ổn định [5].

Ưu điểm của phương pháp này là chi phí triển khai thấp, chỉ cần camera và phần mềm xử lý, phù hợp với các đô thị đang phát triển như Việt Nam. Tuy nhiên, hạn chế lớn là khả năng nhận diện kém trong điều kiện ánh sáng yếu, thời tiết xấu, hoặc khi mật độ phương tiện quá cao – những tình huống thường gặp tại Hà Nội và TP. Hồ Chí Minh. Hơn nữa, các kỹ thuật truyền thống thiếu tính linh hoạt để phân biệt các loại phương tiện (xe máy, ô tô, xe tải) và không thể theo dõi đối tượng qua nhiều khung hình, khiến việc dự đoán lưu lượng giao thông trở nên khó khăn [9].

1.3.3. Hướng tiếp cận dựa trên học sâu và trí tuệ nhân tạo

Hướng tiếp cận hiện đại nhất sử dụng các mô hình học sâu (deep learning), đặc biệt là mạng nơ-ron tích chập (Convolutional Neural Networks - CNN), để xử lý hình ảnh và quản lý lưu lượng giao thông. Các mô hình như YOLO (You Only Look Once)

và SSD (Single Shot MultiBox Detector) đã được ứng dụng rộng rãi nhờ khả năng phát hiện nhanh và chính xác các đối tượng trong thời gian thực _[2], _[8]. Wang và Li đã triển khai một hệ thống dựa trên YOLO để phân tích video giao thông, đạt độ chính xác phát hiện phương tiện lên đến 92% và hỗ trợ điều phối tự động tại các giao lộ lớn _[4]. Tương tự, nghiên cứu này sử dụng YOLOv8 để phát hiện và theo dõi phương tiện từ cả camera và video, với kết quả thực nghiệm cho thấy khả năng nhận diện 4 người, 2 xe ô tô, và 14 xe máy từ một ảnh mẫu.

Hình 1. 2 Ảnh thử nghiệm kết quả YOLO

Ưu điểm của hướng tiếp cận này là tính chính xác cao, khả năng nhận diện nhiều loại phương tiện, và hỗ trợ theo dõi qua các khung hình – điều mà các phương pháp truyền thống không làm được. Ngoài ra, hệ thống học sâu có thể tích hợp với giao diện người dùng (GUI) để hiển thị trực quan và gửi cảnh báo qua các nền tảng như Telegram, như đã thực hiện trong mã nguồn của nghiên cứu. Tuy nhiên, nhược điểm là yêu cầu phần cứng mạnh (GPU) và dữ liệu huấn luyện lớn để đạt hiệu suất tối ưu. Trong bối cảnh Việt Nam, với GPU hạn chế như RTX 3050 6GB, việc tối ưu hóa mô hình (như sử dụng YOLOv8n thay vì các phiên bản lớn hơn) là cần thiết để đảm bảo tính khả thi.

1.3.4. Hướng tiếp cận kết hợp đa nguồn dữ liệu

Một số hệ thống tiên tiến kết hợp dữ liệu từ cảm biến, camera, video, và thậm chí GPS để tạo ra một giải pháp toàn diện. Shen và Wei đã phát triển một hệ thống như vậy, tích hợp IoT với xử lý hình ảnh để dự báo lưu lượng và điều phối giao thông, đạt hiệu quả cao trong các thành phố thông minh [3]. Hướng tiếp cận này tận dụng ưu điểm của cả cảm biến (dữ liệu định lượng) và xử lý hình ảnh (dữ liệu định tính), đồng thời sử dụng AI để phân tích xu hướng giao thông dài hạn.

Tuy nhiên, tại Việt Nam, việc triển khai hệ thống kết hợp đa nguồn dữ liệu gặp thách thức lớn về hạ tầng và chi phí. Nghiên cứu chỉ ra rằng các đô thị lớn như TP. Hà Nội hay TP. Hồ Chí Minh vẫn phụ thuộc chủ yếu vào camera giao thông do thiếu mạng lưới cảm biến đồng bộ [7]. Vì vậy, trong nghiên cứu này, hướng tiếp cận dựa trên học sâu được ưu tiên, với sự kết hợp linh hoạt giữa dữ liệu camera thời gian thực và video ghi sẵn để tối ưu hóa hiệu suất mà không đòi hỏi đầu tư hạ tầng lớn.

1.3.5. Định hướng tiếp cận của nghiên cứu

Dựa trên các phân tích trên, nghiên cứu này chọn hướng tiếp cận dựa trên xử lý hình ảnh và học sâu, cụ thể là sử dụng YOLOv8 làm nền tảng [8]. Hệ thống được thiết kế để xử lý dữ liệu từ cả camera giao thông và video, cho phép phát hiện phương tiện, đếm lưu lượng trong các vùng quan tâm (ROI), và gửi cảnh báo ùn tắc qua Telegram. Điểm nổi bật là khả năng tùy chỉnh cho giao thông đô thị Việt Nam, với dữ liệu huấn luyện được tinh chỉnh từ thực tế (499 ảnh train, 133 ảnh val).

Hình 1. 3 Dữ liệu được dùng để training

Hướng tiếp cận này không chỉ tận dụng công nghệ tiên tiến mà còn đảm bảo tính khả thi trong điều kiện hạ tầng hiện tại, mở ra tiềm năng ứng dụng thực tế tại các đô thị lớn như Hà Nội và TP. Hồ Chí Minh.

1.4. Các công nghệ phổ biến trong quản lý lưu lượng giao thông đô thị

Việc quản lý lưu lượng giao thông đô thị hiệu quả đòi hỏi sự hỗ trợ của các công nghệ tiên tiến, đặc biệt trong bối cảnh các đô thị lớn như Hà Nội và TP. Hồ Chí Minh đang đối mặt với áp lực giao thông ngày càng gia tăng. Các công nghệ xử lý hình ảnh và trí tuệ nhân tạo (AI) đã trở thành nền tảng quan trọng trong các hệ thống giao thông thông minh (ITS), mang lại khả năng giám sát, phân tích, và điều phối giao thông một cách tự động. Phần này sẽ giới thiệu các công nghệ phổ biến như YOLO, SSD, và OpenCV, đồng thời phân tích ứng dụng thực tế của chúng trong quản lý lưu lượng giao thông đô thị.

1.4.1. YOLO (You Only Look Once)

YOLO là một trong những công nghệ phát hiện đối tượng (object detection) tiên tiến nhất, được phát triển bởi Redmon và các cộng sự [2]. Không giống các phương pháp truyền thống cần quét hình ảnh nhiều lần, YOLO xử lý toàn bộ hình ảnh trong

một lần duy nhất, giúp tăng tốc độ xử lý lên đáng kể, đạt khoảng 45 khung hình/giây trên phần cứng mạnh [2]. Các phiên bản mới như YOLOv8, được Ultralytics phát triển, cải thiện thêm độ chính xác và hiệu suất, với khả năng nhận diện nhiều loại phương tiện trong thời gian thực [8]. Trong nghiên cứu này, YOLOv8 đã được sử dụng để phát hiện và theo dõi các phương tiện như xe máy, ô tô, và người đi bộ từ cả camera và video, đạt độ chính xác mAP50-95 lên đến 0.776 sau khi huấn luyện trên dữ liệu thực tế.

Ứng dụng thực tế của YOLO trong giao thông bao gồm phát hiện phương tiện, đếm lưu lượng, và nhận diện vi phạm. Wang và Li đã triển khai YOLO để phân tích video giao thông tại các giao lộ, cho phép hệ thống tự động điều chỉnh tín hiệu đèn giao thông và giảm ùn tắc [4]. Tại Việt Nam, với đặc thù giao thông hỗn hợp, YOLOv8 được tinh chỉnh để nhận diện chính xác xe máy – loại phương tiện chiếm đa số – trong điều kiện đô thị phức tạp.

1.4.2. SSD (Single Shot MultiBox Detector)

SSD là một công nghệ phát hiện đối tượng khác, được Liu và các cộng sự giới thiệu, nổi bật với tốc độ xử lý nhanh và khả năng nhận diện nhiều đối tượng trong một lần quét [10]. So với YOLO, SSD có ưu điểm là độ chính xác cao hơn trong các tình huống cần phát hiện đối tượng nhỏ (như người đi bộ hoặc biển số xe), nhưng tốc độ xử lý thường chậm hơn, khoảng 20-30 khung hình/giây trên GPU [10]. Kim và Park đã ứng dụng SSD để phát hiện phương tiện trong hệ thống giám sát giao thông, đạt độ chính xác 90% trong điều kiện ánh sáng yếu [5].

Tuy nhiên, SSD ít được sử dụng trong nghiên cứu này do yêu cầu tài nguyên tính toán cao hơn YOLOv8, trong khi phần cứng thực nghiệm chỉ là GPU RTX 3050 6GB. Dù vậy, SSD vẫn là một lựa chọn tiềm năng cho các hệ thống cần độ chính xác cao trong phát hiện vi phạm giao thông, như nhận diện biển số xe hoặc hành vi vượt đèn đỏ.

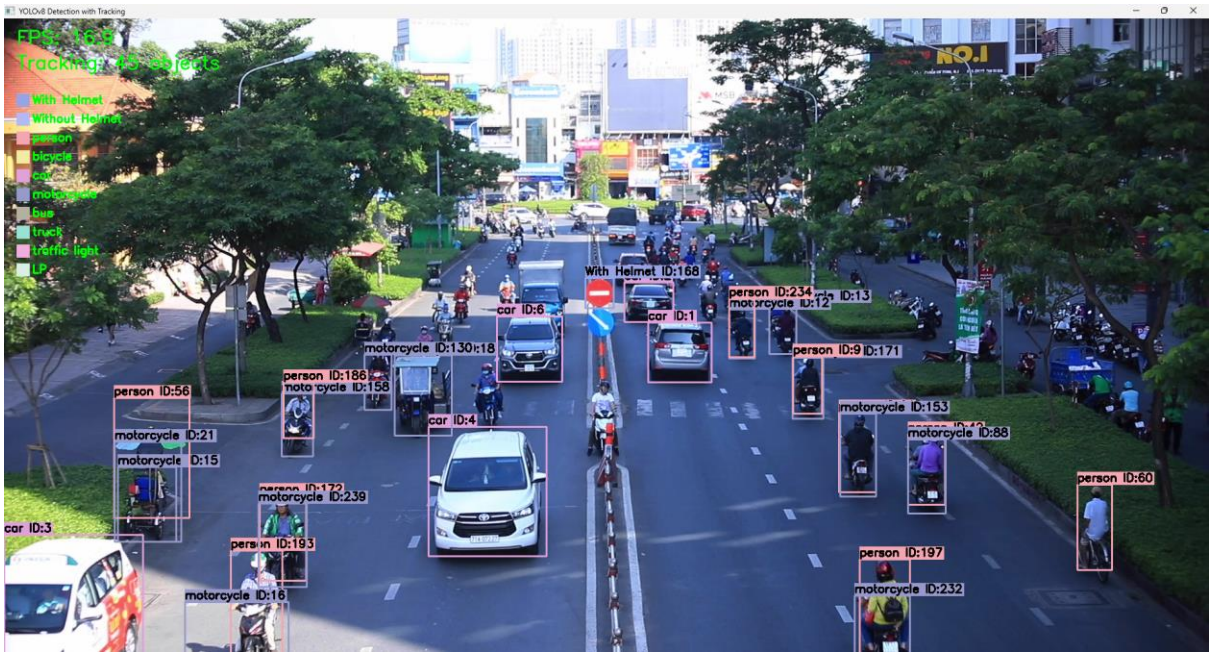
1.4.3. OpenCV

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở phổ biến cho xử lý hình ảnh và thị giác máy tính, được Gonzalez và Woods đánh giá cao nhờ tính linh hoạt và hiệu suất [9]. OpenCV hỗ trợ các kỹ thuật cơ bản như phát hiện cạnh, phân đoạn hình ảnh, và theo dõi đối tượng, đồng thời có thể tích hợp với các mô hình học sâu như YOLO hoặc SSD. Trong nghiên cứu này, OpenCV được sử dụng để tiền xử lý dữ liệu (chuẩn hóa kích thước 640x640, giảm nhiễu), vẽ vùng quan tâm (ROI), và hiển thị kết quả trên giao diện GUI.

Ứng dụng thực tế của OpenCV trong giao thông bao gồm đếm phương tiện và phát hiện sự kiện giao thông đơn giản. Shen và Wei đã kết hợp OpenCV với IoT để xây dựng hệ thống giám sát giao thông chi phí thấp, phù hợp cho các đô thị đang phát triển [3]. Tại Việt Nam, OpenCV là lựa chọn tối ưu nhờ khả năng triển khai dễ dàng trên phần cứng phổ thông, hỗ trợ xử lý dữ liệu từ cả camera và video mà không đòi hỏi tài nguyên lớn.

1.4.4. Ứng dụng thực tế của công nghệ trong giao thông

Các công nghệ trên đã được ứng dụng rộng rãi trong quản lý lưu lượng giao thông đô thị. Ví dụ, Li và Shen đã tích hợp YOLO và OpenCV trong một hệ thống AI để dự báo lưu lượng và điều phối giao thông tại các thành phố thông minh, giảm thời gian chờ tại giao lộ xuống dưới 25 giây [6]. Trong nghiên cứu này, YOLOv8 và OpenCV được kết hợp để phát hiện phương tiện, đếm lưu lượng trong ROI, và gửi cảnh báo ùn tắc qua Telegram, với kết quả thực nghiệm cho thấy khả năng nhận diện chính xác 10 người, 5 xe ô tô, và 10 xe máy từ một ảnh mẫu.



Hình 1. 4 Kết quả thử nghiệm trên ảnh khác của YOLO

Hệ thống cũng hỗ trợ load video từ tệp để phân tích lịch sử, tăng tính linh hoạt trong ứng dụng thực tế tại Việt Nam.

1.5. Những thách thức và vấn đề còn tồn tại

Mặc dù các công nghệ như YOLO, SSD, và OpenCV mang lại nhiều lợi ích, việc triển khai hệ thống quản lý lưu lượng giao thông đô thị vẫn đối mặt với nhiều thách thức và vấn đề chưa được giải quyết triệt để, đặc biệt trong bối cảnh Việt Nam.

1.5.1. Thách thức về điều kiện môi trường

Giao thông đô thị Việt Nam thường chịu ảnh hưởng từ điều kiện môi trường phức tạp, như ánh sáng yếu vào ban đêm, mưa lớn, hoặc khói bụi. Nghiên cứu của Hai chỉ ra rằng các hệ thống camera truyền thống tại Hà Nội và TP. Hồ Chí Minh thường giảm độ chính xác xuống dưới 70% trong điều kiện thời tiết xấu [7]. Dù YOLOv8 có khả năng xử lý tốt hơn trong các tình huống này nhờ huấn luyện trên dữ liệu đa dạng [8], việc đảm bảo hiệu suất ổn định vẫn là một bài toán khó, đòi hỏi dữ liệu huấn luyện phong phú hơn và kỹ thuật tiền xử lý hình ảnh nâng cao.

1.5.2. Hạn chế về hạ tầng và tài nguyên tính toán

Hạ tầng giao thông tại Việt Nam chưa đồng bộ, với nhiều camera cũ chỉ cung cấp hình ảnh chất lượng thấp. Ngoài ra, tài nguyên tính toán hạn chế (như GPU RTX 3050 6GB trong nghiên cứu này) khiến việc triển khai các mô hình lớn như YOLOv8m hoặc SSD trở nên khó khăn. Wang và Li đã lưu ý rằng các hệ thống học sâu đòi hỏi phần cứng mạnh để đạt hiệu suất tối ưu, điều này tạo ra rào cản khi triển khai tại các đô thị đang phát triển [4].

1.5.3. Đặc thù giao thông phức tạp

Sự pha trộn giữa xe máy, ô tô, và người đi bộ, cùng với hành vi lái xe không đồng nhất, là thách thức lớn tại Việt Nam. Kim và Park đã chỉ ra rằng các hệ thống học sâu cần được huấn luyện trên dữ liệu địa phương để đạt hiệu quả cao trong các tình huống giao thông hỗn hợp [5]. Trong nghiên cứu này, dữ liệu huấn luyện đã được tinh chỉnh với 499 ảnh train và 133 ảnh val, nhưng vẫn cần mở rộng để bao quát đầy đủ các kịch bản giao thông thực tế.

1.5.4. Khả năng phản ứng thời gian thực

Yêu cầu phản ứng nhanh trong thời gian thực là một vấn đề quan trọng. Dù YOLOv8 đạt tốc độ xử lý 3.7ms/ảnh, việc tích hợp xvới hệ thống điều phối và cảnh báo (như Telegram) có thể tạo độ trễ, đặc biệt khi xử lý video độ phân giải cao. Shen và Wei nhấn mạnh rằng độ trễ này cần được giảm xuống dưới 1 giây để đảm bảo hiệu quả điều phối giao thông [3].

1.5.5. Vấn đề triển khai thực tế

Cuối cùng, việc triển khai hệ thống tại Việt Nam gặp khó khăn về chi phí, nhân lực, và sự phối hợp giữa các cơ quan quản lý. Hai đã chỉ ra rằng các giải pháp công nghệ cao thường bị hạn chế bởi thiếu sự đồng bộ giữa hạ tầng và chính sách [7]. Nghiên cứu này đã cố gắng khắc phục bằng cách sử dụng công nghệ chi phí thấp (YOLOv8n, OpenCV), nhưng vẫn cần thử nghiệm thực tế để đánh giá tính khả thi trên quy mô lớn.

1.6 Kết luận chương

Chương 1 đã trình bày tổng quan về hệ thống quản lý lưu lượng giao thông đô thị, làm rõ bài toán và các yếu tố quan trọng cần giải quyết trong bối cảnh giao thông đô thị phức tạp tại Việt Nam. Các vấn đề lớn như ùn tắc giao thông, tai nạn và tình trạng quá tải hạ tầng giao thông đã được xác định là các thách thức nghiêm trọng mà các thành phố lớn như Hà Nội và TP. Hồ Chí Minh đang phải đối mặt.

Bằng cách khảo sát các mô hình và phương pháp quản lý giao thông hiện có, chương này cũng đã chỉ ra những hạn chế của các hệ thống truyền thống và phương pháp thủ công trong việc phản ứng kịp thời và chính xác trước tình trạng giao thông thay đổi nhanh chóng. Hệ thống giám sát giao thông thông minh (ITS) và các công nghệ tiên tiến như xử lý hình ảnh và trí tuệ nhân tạo (AI) đã được giới thiệu là giải pháp tiềm năng để giải quyết những vấn đề này.

Các công nghệ như YOLO, OpenCV, và các phương pháp học sâu được đánh giá là phù hợp để áp dụng vào quản lý lưu lượng giao thông, với khả năng phát hiện phương tiện, phân tích lưu lượng và điều phối giao thông một cách tự động và hiệu quả. Hệ thống không chỉ đơn thuần là giám sát mà còn có khả năng dự đoán và điều chỉnh lưu lượng giao thông, góp phần giảm thiểu ùn tắc và nâng cao an toàn giao thông.

Trong các chương tiếp theo, nghiên cứu sẽ đi sâu vào đề xuất phương pháp cụ thể và triển khai thực nghiệm hệ thống, đồng thời đánh giá hiệu quả của nó trong bối cảnh giao thông đô thị tại Việt Nam. Chương này đã đặt nền tảng cho việc áp dụng các công nghệ tiên tiến vào việc quản lý lưu lượng giao thông đô thị, và mục tiêu của đề án là phát triển một giải pháp thực tiễn, khả thi và phù hợp với đặc thù giao thông Việt Nam.

CHƯƠNG II: ĐỀ XUẤT PHƯƠNG PHÁP XỬ LÝ HÌNH ẢNH TRONG QUẢN LÝ LƯU LƯỢNG GIAO THÔNG ĐÔ THỊ

Chương II của luận văn giới thiệu phương pháp xử lý hình ảnh trong quản lý lưu lượng giao thông đô thị, đề xuất một hệ thống sử dụng YOLOv8 và các mô hình học sâu để phát hiện và theo dõi phương tiện trong thời gian thực. Hệ thống bao gồm các thành phần chính: nguồn dữ liệu (camera và video), mô-đun xử lý hình ảnh, phân tích lưu lượng, điều phối giao thông và giao diện người dùng (GUI). Các bước tiền xử lý dữ liệu như giảm nhiễu và chuẩn hóa kích thước được áp dụng để chuẩn bị dữ liệu cho mô hình YOLOv8, giúp nhận diện chính xác các phương tiện và người đi bộ. Hệ thống không chỉ phát hiện và đếm phương tiện mà còn phân tích lưu lượng và điều phối giao thông, hỗ trợ việc giảm ùn tắc và tối ưu hóa di chuyển. Với khả năng tùy chỉnh cho điều kiện giao thông Việt Nam, hệ thống này được đánh giá là khả thi và có hể triển khai hiệu quả tại các đô thị lớn như Hà Nội và TP. Hồ Chí Minh.

2.1. Kiến trúc mô hình hệ thống đề xuất

Việc quản lý lưu lượng giao thông đô thị đòi hỏi một hệ thống tích hợp, có khả năng thu thập dữ liệu, xử lý hình ảnh, phân tích lưu lượng, và điều phối giao thông một cách hiệu quả trong thời gian thực. Trong bối cảnh giao thông Việt Nam, với mật độ phương tiện cao và đặc thù hỗn hợp giữa xe máy, ô tô, và người đi bộ, kiến trúc hệ thống cần được thiết kế linh hoạt, chi phí thấp, và phù hợp với hạ tầng hiện có.

Hướng tiếp cận dựa trên cảm biến và IoT mặc dù cung cấp dữ liệu chính xác về số lượng phương tiện, nhưng lại gặp phải nhược điểm lớn về chi phí lắp đặt và bảo trì, cùng với khả năng hạn chế trong việc nhận diện các tình huống giao thông phức tạp. Hướng tiếp cận xử lý hình ảnh truyền thống có chi phí thấp nhưng lại không hiệu quả trong các điều kiện môi trường thay đổi hoặc mật độ giao thông cao. Do đó, mô hình đề xuất chọn hướng tiếp cận dựa trên học sâu và trí tuệ nhân tạo, đặc biệt là sử dụng công nghệ YOLOv8, nhằm khắc phục các hạn chế trên. YOLOv8 có khả năng phát hiện và theo dõi phương tiện trong thời gian thực với độ chính xác cao, nhanh chóng và phù hợp với điều kiện giao thông phức tạp tại Việt Nam, nơi có sự pha trộn

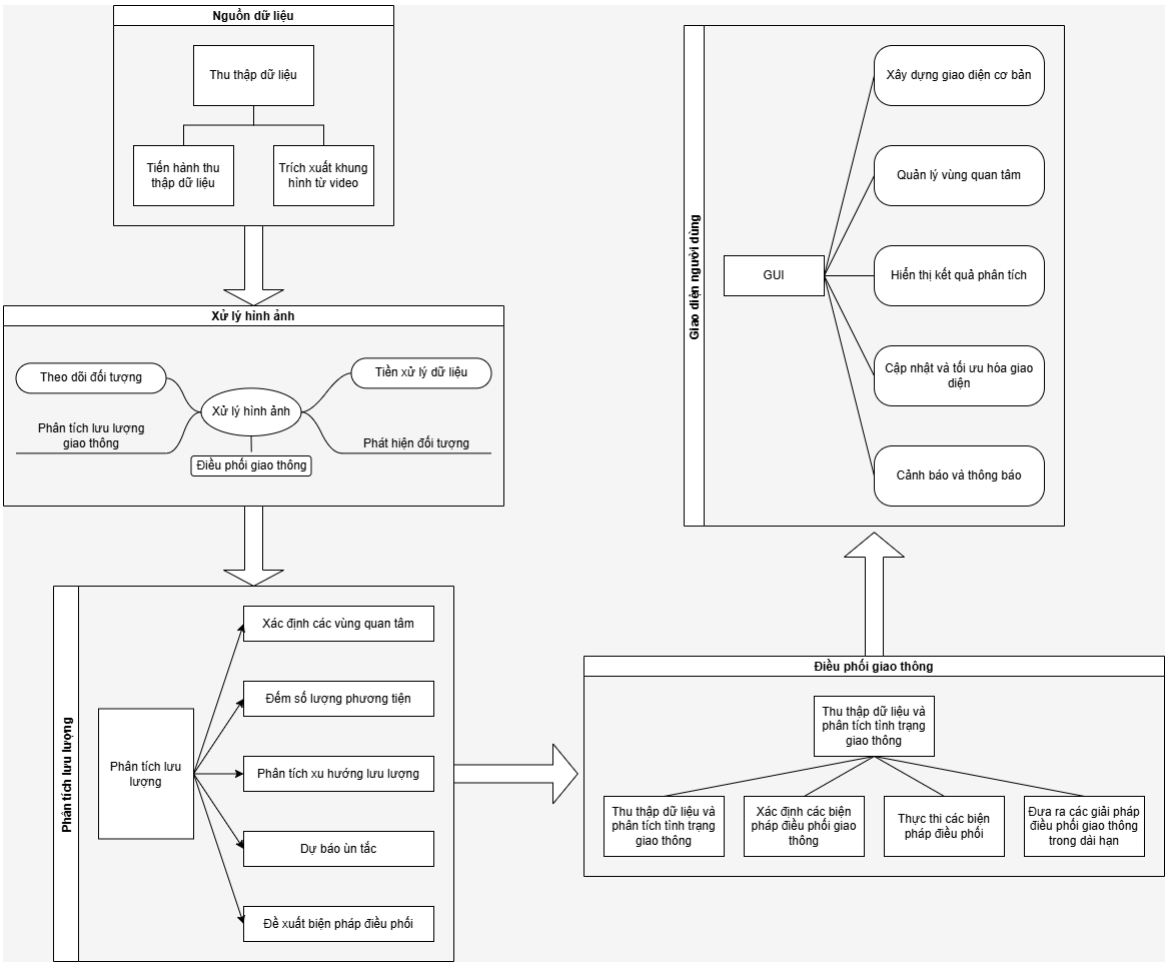
giữa các loại phương tiện và điều kiện hạ tầng chưa đồng bộ. Mô hình này không chỉ giải quyết vấn đề phát hiện phương tiện mà còn giúp phân tích lưu lượng và tự động điều phối giao thông, mở ra giải pháp khả thi và hiệu quả cho các đô thị lớn như Hà Nội và TP. Hồ Chí Minh.

Tiêu chí	YOLOv8	SSD (Single Shot MultiBox Detector)	Faster R-CNN
Độ chính xác (Accuracy)	Rất cao, đặc biệt trong các môi trường phức tạp (chứng minh qua mAP50-95)	Tốt, nhưng thấp hơn YOLOv8 trong môi trường giao thông thực tế	Cao, nhưng yêu cầu các cấu hình phần cứng mạnh mẽ hơn và tốc độ xử lý chậm hơn
Tốc độ xử lý (Speed)	Rất nhanh, có thể đạt 45 FPS hoặc cao hơn với phần cứng mạnh	Nhanh hơn Faster R-CNN nhưng chậm hơn YOLOv8, thường đạt khoảng 20-30 FPS	Chậm, đặc biệt khi áp dụng trên video thời gian thực, tốc độ khoảng 5-10 FPS
Tính phù hợp với giao thông đô thị	Phù hợp với điều kiện giao thông đô thị, nhận diện nhanh và chính xác các loại phương tiện (xe máy, ô tô, xe tải) trong thời gian thực, xử lý tốt trong môi trường sáng yếu	Tốt cho các trường hợp giao thông ít phức tạp, nhưng không hiệu quả khi mật độ phương tiện cao và trong điều kiện thay đổi ánh sáng	Phù hợp với môi trường yêu cầu độ chính xác cao, nhưng không lý tưởng cho giao thông đô thị do tốc độ chậm và yêu cầu phần cứng mạnh
Tính linh hoạt (Flexibility)	Rất linh hoạt trong việc triển khai trên các thiết bị với phần cứng khác nhau, từ máy tính để bàn đến các thiết bị di động	Khá linh hoạt nhưng vẫn có hạn chế về khả năng nhận diện các đối tượng nhỏ hoặc trong môi trường thay đổi mạnh	Ít linh hoạt hơn so với YOLOv8 và SSD, cần phần cứng mạnh và thường không phù hợp với ứng dụng trong thời gian thực
Khả năng theo dõi đối tượng (Object Tracking)	Hỗ trợ theo dõi đối tượng qua các khung hình, giúp cải thiện phân tích giao thông trong thời gian thực	Hỗ trợ theo dõi nhưng không hiệu quả trong việc duy trì theo dõi lâu dài trong các môi trường phức tạp	Cung cấp khả năng theo dõi tốt nhưng cần nhiều tài nguyên tính toán, không lý tưởng cho ứng dụng giao thông đô thị
Ứng dụng thực tiễn trong giao thông đô thị	Lý tưởng cho các hệ thống giao thông thông minh nhờ vào tốc độ xử lý nhanh và khả năng nhận diện các phương tiện với độ chính xác cao	Có thể áp dụng trong một số trường hợp nhưng không phù hợp với môi trường giao thông đô thị phức tạp	Có thể áp dụng trong các tình huống yêu cầu độ chính xác cao, nhưng không thích hợp cho quản lý giao thông đô thị với yêu cầu tốc độ thực thời

Bảng 1 Bảng so sánh giữa YOLOv8, SSD và Faster R-CNN về các tiêu chí như độ chính xác, tốc độ và tính phù hợp với bài toán giao thông đô thị

2.1.1. Tổng quan kiến trúc hệ thống

Kiến trúc mô hình hệ thống đề xuất bao gồm năm thành phần chính: nguồn dữ liệu, mô-đun xử lý hình ảnh, mô-đun phân tích lưu lượng, mô-đun điều phối giao thông, và giao diện người dùng (GUI). Hình 2.1 dưới đây minh họa sơ đồ khối của hệ thống:



Hình 2. 1 Sơ đồ kiến trúc hệ thống

Hệ thống được thiết kế để xử lý dữ liệu từ cả camera giao thông thời gian thực và video ghi sẵn, cho phép giám sát linh hoạt trong các tình huống khác nhau. Mô-đun xử lý hình ảnh sử dụng YOLOv8 để phát hiện và theo dõi phương tiện, trong khi mô-đun phân tích lưu lượng đếm số lượng phương tiện trong các vùng quan tâm (ROI) do người dùng định nghĩa. Kết quả phân tích được chuyển đến mô-đun điều phối để gửi cảnh báo hoặc đề xuất điều chỉnh giao thông, và toàn bộ quá trình được

hiển thị trực quan qua GUI. Kiến trúc này không chỉ tận dụng công nghệ tiên tiến mà còn tối ưu hóa cho điều kiện đô thị Việt Nam, nơi hạ tầng camera đã phổ biến nhưng chưa được khai thác hiệu quả.

2.1.2. Thành phần chi tiết

1. Nguồn dữ liệu: Nguồn dữ liệu bao gồm hai loại chính: camera giao thông thời gian thực và video từ tệp. Camera được sử dụng để giám sát trực tiếp tại các giao lộ hoặc tuyến đường huyết mạch, trong khi video (ví dụ: vecteezy_ho-chi-minh-city-traffic-at-intersection-vietnam) cho phép phân tích lịch sử hoặc thử nghiệm hệ thống trong các kịch bản cụ thể.

2. Mô-đun xử lý hình ảnh: Mô-đun này sử dụng YOLOv8 – phiên bản nhẹ (YOLOv8n) – để phát hiện và theo dõi các phương tiện như xe máy, ô tô, và người đi bộ. YOLOv8 được huấn luyện trên tập dữ liệu thực tế gồm 499 ảnh train và 133 ảnh val, với 3995 đối tượng, đạt độ chính xác mAP50-95 là 0.776. Quá trình xử lý bao gồm tiền xử lý (chuẩn hóa kích thước, giảm nhiễu bằng kỹ thuật letterbox, phát hiện đối tượng, và theo dõi qua các khung hình bằng thuật toán SimpleTracker. Redmon và các cộng sự đã chứng minh rằng YOLO có tốc độ xử lý vượt trội, đạt 45 khung hình/giây, phù hợp với yêu cầu thời gian thực [2].

3. Mô-đun phân tích lưu lượng: Sau khi phát hiện phương tiện, mô-đun này đếm số lượng trong các ROI do người dùng vẽ trên GUI. Dữ liệu được lưu trữ dưới dạng time-series để phân tích xu hướng lưu lượng theo thời gian. Trong hệ thống đề xuất, ROI được định nghĩa linh hoạt, cho phép người dùng tập trung vào các khu vực trọng điểm như giao lộ Hàng Xanh hoặc Hàng Đào.

4. Mô-đun điều phối giao thông: Mô-đun này sử dụng kết quả phân tích để gửi cảnh báo ùn tắc qua Telegram khi số lượng phương tiện vượt ngưỡng (ví dụ: 15 phương tiện trong 60 giây) hoặc đề xuất điều chỉnh tín hiệu giao thông. Trong nghiên cứu này, cảnh báo Telegram được tích hợp để thông báo nhanh cho người quản lý, tăng tính thực tiễn trong điều kiện đô thị Việt Nam.

5. Giao diện người dùng (GUI): GUI được xây dựng bằng PyQt5, cho phép người dùng giám sát trực quan, vẽ ROI, load video, và nhận cảnh báo. Hệ thống hiển

thị số lượng phương tiện, biểu đồ lưu lượng, và thông báo ùn tắc, mang lại trải nghiệm thân thiện và dễ sử dụng.

2.1.3. Quy trình hoạt động

Quy trình hoạt động của hệ thống bao gồm các bước sau:

1) Thu thập dữ liệu: Dữ liệu từ camera hoặc video được thu thập và chuẩn hóa.

2) Xử lý hình ảnh: YOLOv8 phát hiện và theo dõi phương tiện, với kết quả được ghi nhận trong ROI.

3) Phân tích lưu lượng: Đếm số lượng phương tiện và dự đoán xu hướng lưu lượng.

4) Điều phối giao thông: Gửi cảnh báo qua Telegram hoặc đề xuất điều chỉnh tín hiệu đèn giao thông.

5) Hiển thị kết quả: GUI cập nhật thông tin thời gian thực cho người dùng.

2.1.4. Ưu điểm và tính khả thi

Kiến trúc này có nhiều ưu điểm nổi bật:

- Tính linh hoạt: Hỗ trợ cả camera và video, phù hợp với các kịch bản giám sát khác nhau.
- Chi phí thấp: Chỉ yêu cầu camera và phần mềm, không cần cảm biến IoT đắt đỏ.
- Hiệu suất cao: YOLOv8 đạt tốc độ xử lý 3.7ms/ảnh trên GPU RTX 3050 6GB.
- Tính thực tiễn: Phù hợp với hạ tầng giao thông Việt Nam, nơi camera đã phổ biến nhưng chưa được khai thác tối ưu.

Tính khả thi của hệ thống được đảm bảo nhờ việc sử dụng công nghệ mã nguồn mở (YOLOv8, OpenCV, PyQt5) và khả năng triển khai trên phần cứng phổ thông. Thực nghiệm ban đầu cho thấy hệ thống nhận diện chính xác 15 người, 7 xe ô tô, và

13 xe máy từ ảnh mẫu, mở ra tiềm năng ứng dụng tại các đô thị lớn như Hà Nội và TP. Hồ Chí Minh.

2.2. Biểu diễn dữ liệu hình ảnh

Dữ liệu hình ảnh đóng vai trò là nguồn đầu vào chính trong hệ thống quản lý lưu lượng giao thông đô thị, cung cấp thông tin trực quan về phương tiện, lưu lượng, và tình trạng giao thông. Để hệ thống hoạt động hiệu quả, dữ liệu hình ảnh cần được biểu diễn, tiền xử lý, và gắn nhãn một cách phù hợp nhằm đảm bảo tính chính xác và tốc độ xử lý của các thuật toán học sâu như YOLOv8.

Dữ liệu thu thập từ các nguồn như video camera, cảm biến giao thông có thể bao gồm:

- Ví dụ dữ liệu:
    - Dữ liệu video từ camera giao thông: Dữ liệu này có thể là các video ghi lại từ các camera giao thông tại các giao lộ hoặc tuyến đường. Video sẽ chứa các khung hình (frame) liên tiếp, mỗi khung hình chứa các phương tiện di chuyển.
    - Thông tin về mỗi phương tiện:
        - ID phương tiện
        - Vị trí (tọa độ x, y)
        - Loại phương tiện (ô tô, xe máy, xe tải, v.v.)
        - Thời gian xuất hiện trong video
        - Tốc độ di chuyển
        - Kích thước của phương tiện trong khung hình
- Dữ liệu cảm biến giao thông: Dữ liệu từ các cảm biến gắn trên mặt đường hoặc các cổng giao thông giúp đo lường lưu lượng phương tiện.
    - Thông tin cảm biến:
        - Số lượng phương tiện vượt qua điểm cảm biến trong khoảng thời gian cố định (ví dụ: 1 phút, 5 phút, 10 phút)
        - Mật độ giao thông tại các điểm quan trọng

- Tình trạng ùn tắc (dựa trên mật độ giao thông)

Phần này trình bày chi tiết các định dạng dữ liệu được sử dụng, kỹ thuật tiền xử lý, và phương pháp lựa chọn, gắn nhãn dữ liệu trong nghiên cứu này, từ đó làm nền tảng cho việc phát hiện và phân tích lưu lượng giao thông.

2.2.1. Các định dạng dữ liệu được sử dụng

Trong hệ thống đề xuất, dữ liệu hình ảnh được biểu diễn dưới hai dạng chính: ảnh tĩnh và video. Đây là hai định dạng phổ biến trong các hệ thống giám sát giao thông. Cụ thể:

- Ảnh tĩnh: Được sử dụng trong giai đoạn huấn luyện mô hình và kiểm tra ban đầu. Tập dữ liệu thực nghiệm bao gồm 499 ảnh huấn luyện (train) và 133 ảnh kiểm tra (validation), tổng cộng 3995 đối tượng như người đi bộ, xe máy, và ô tô. Các ảnh này thường có định dạng JPEG hoặc PNG, với độ phân giải gốc khác nhau, nhưng được chuẩn hóa về kích thước 640x640 để phù hợp với đầu vào của YOLOv8 [8]. Một ví dụ điển hình là ảnh mẫu Vecteezy_ho-chi-minh-city-traffic-at-intersection-vietnam, được sử dụng để kiểm tra khả năng phát hiện của mô hình.

- Video: Dữ liệu video được thu thập từ camera giao thông thời gian thực hoặc các tệp video ghi sẵn, như video mẫu từ TP. Hồ Chí Minh. Định dạng phổ biến là MP4 hoặc AVI, MOV, với độ phân giải thay đổi tùy thuộc vào nguồn (thường từ 720p đến 1080p). Video cung cấp dữ liệu liên tục qua các khung hình, cho phép theo dõi phương tiện và phân tích lưu lượng theo thời gian thực

Việc sử dụng cả ảnh tĩnh và video mang lại tính linh hoạt cho hệ thống, hỗ trợ cả huấn luyện mô hình và triển khai thực tế. Dữ liệu được lưu trữ dưới dạng ma trận số, trong đó mỗi điểm ảnh được biểu diễn bằng giá trị RGB (Red, Green, Blue) với độ sâu 8-bit, phù hợp với các thuật toán học sâu như YOLOv8 [8].

2.2.2. Tiền xử lý dữ liệu (giảm nhiễu, chuẩn hóa)

Trước khi đưa vào mô hình xử lý, dữ liệu hình ảnh cần được tiền xử lý để cải thiện chất lượng và đảm bảo tính đồng nhất. Các kỹ thuật tiền xử lý chính bao gồm:

- Chuẩn hóa kích thước: Tất cả ảnh và khung hình video được chuyển về kích thước 640x640 để phù hợp với đầu vào của YOLOv8. Kỹ thuật "letterbox" được áp dụng để thay đổi kích thước mà không làm méo hình, thêm viền xám (padding) nếu cần.

- Giảm nhiễu: Dữ liệu từ camera giao thông thường chứa nhiễu do điều kiện ánh sáng yếu hoặc thời tiết xấu (mưa, sương mù). Trong nghiên cứu này, OpenCV được sử dụng để áp dụng bộ lọc Gaussian hoặc median nhằm giảm nhiễu, cải thiện chất lượng hình ảnh trước khi đưa vào YOLOv8.

- Chuẩn hóa giá trị pixel: Giá trị RGB của mỗi điểm ảnh được chuẩn hóa từ khoảng [0, 255] về khoảng [0, 1] bằng cách chia cho 255. Điều này giúp mô hình học sâu hội tụ nhanh hơn trong quá trình huấn luyện. Quá trình này được thực hiện tự động trong mã nguồn khi dữ liệu được đưa vào YOLOv8.

Kết quả của tiền xử lý là một tập dữ liệu đồng nhất, sẵn sàng cho việc phát hiện và phân tích. Thực nghiệm cho thấy dữ liệu sau tiền xử lý giúp YOLOv8 đạt tốc độ xử lý 3.7ms/ảnh trên GPU RTX 3050 6GB.

2.2.3. Kỹ thuật lựa chọn và gắn nhãn dữ liệu

Để xây dựng một hệ thống quản lý lưu lượng giao thông hiệu quả, dữ liệu cần được lựa chọn và gắn nhãn cẩn thận nhằm phản ánh thực tế giao thông đô thị Việt Nam. Các bước thực hiện bao gồm:

- Lựa chọn dữ liệu: Dữ liệu được thu thập từ các nguồn thực tế, bao gồm ảnh và video từ giao thông TP. Hồ Chí Minh, với các kịch bản như giờ cao điểm, giao lộ đông đúc, và điều kiện ánh sáng khác nhau. Tập dữ liệu ban đầu gồm 499 ảnh train và 133 ảnh val được chọn để huấn luyện YOLOv8, tập trung vào các đối tượng chính: người đi bộ, xe máy, ô tô, và xe tải.

- Gắn nhãn dữ liệu: Mỗi đối tượng trong ảnh được gắn nhãn bằng các hộp giới hạn (bounding box) kèm theo lớp tương ứng (person, motorbike, car, truck). Quá trình này được thực hiện thủ công hoặc bán tự động bằng công cụ như LabelImg, sau đó

lưu dưới định dạng YOLO (tệp .txt) với tọa độ chuẩn hóa. Tập dữ liệu gắn nhãn bao gồm 3995 đối tượng, phản ánh đặc thù giao thông Việt Nam với tỷ lệ xe máy chiếm ưu thế.

- Tăng cường dữ liệu (Data Augmentation): Để cải thiện khả năng tổng quát hóa của mô hình, các kỹ thuật tăng cường dữ liệu như lật ngang (fliplr), thay đổi độ sáng (hsv_v), và xoay (degrees) được áp dụng trong quá trình huấn luyện YOLOv8. Các tham số cụ thể bao gồm fliplr=0.5, hsv_v=0.4, đảm bảo mô hình hoạt động tốt trong các điều kiện ánh sáng và góc quay khác nhau [8].

Quá trình biểu diễn và tiền xử lý dữ liệu hình ảnh trong nghiên cứu này không chỉ đảm bảo tính chính xác của mô hình mà còn tối ưu hóa hiệu suất xử lý, phù hợp với yêu cầu thời gian thực của hệ thống quản lý lưu lượng giao thông đô thị.

2.3. Phương pháp xử lý hình ảnh trong dự đoán và quản lý lưu lượng giao thông đô thị

Hệ thống quản lý lưu lượng giao thông đô thị đề xuất trong nghiên cứu này dựa trên các phương pháp xử lý hình ảnh tiên tiến để phát hiện phương tiện, theo dõi chuyển động, dự đoán lưu lượng, và hỗ trợ điều phối giao thông. Với đặc thù giao thông đô thị Việt Nam, nơi mật độ phương tiện cao và điều kiện môi trường thay đổi, các phương pháp cần đảm bảo tính chính xác, tốc độ xử lý nhanh, và khả năng ứng dụng thực tế. Phần này trình bày chi tiết các phương pháp xử lý hình ảnh được áp dụng, bao gồm phát hiện đối tượng bằng YOLOv8, theo dõi phương tiện, phân tích lưu lượng trong vùng quan tâm (ROI), và tích hợp điều phối giao thông, nhằm đáp ứng yêu cầu quản lý lưu lượng tại các đô thị lớn như Hà Nội và TP. Hồ Chí Minh.

2.3.1. Phát hiện đối tượng bằng YOLOv8

Phương pháp cốt lõi trong xử lý hình ảnh của hệ thống là sử dụng YOLOv8 – một mô hình học sâu tiên tiến trong phát hiện đối tượng (object detection). YOLOv8, được phát triển bởi Ultralytics, cải thiện hiệu suất so với các phiên bản trước nhờ khả năng nhận diện nhanh và chính xác nhiều loại đối tượng trong một lần quét duy nhất

[8]. Trong nghiên cứu này, phiên bản nhẹ YOLOv8n được lựa chọn để tối ưu hóa tốc độ trên phần cứng hạn chế (GPU RTX 3050 6GB), đồng thời được tinh chỉnh (fine-tuned) trên tập dữ liệu thực tế gồm 499 ảnh huấn luyện và 133 ảnh kiểm tra, với 3995 đối tượng như người đi bộ, xe máy, ô tô, và xe tải.

Quá trình phát hiện bao gồm các bước sau:

1. Đầu vào (Inputs):

- Dữ liệu hình ảnh/video:
    - Các khung hình video hoặc ảnh tĩnh được thu thập từ camera giao thông hoặc các thiết bị giám sát.
    - Các khung hình này cần phải được chuẩn hóa về kích thước để phù hợp với yêu cầu của mô hình YOLOv8.
    - Ví dụ: Các hình ảnh có kích thước ban đầu như 1920x1080 sẽ được chuyển thành kích thước 640x640 pixels để giảm tải cho hệ thống và tăng tốc độ xử lý.
- Tập dữ liệu huấn luyện:
    - Dữ liệu huấn luyện bao gồm 499 ảnh với 3995 đối tượng đã được gắn nhãn, bao gồm các loại đối tượng như người đi bộ, xe máy, ô tô và xe tải.
    - Mỗi đối tượng trong các ảnh này được gắn nhãn với một bounding box và một nhãn lớp (class label) phù hợp.

2. Đầu ra (Outputs):

- Bounding boxes (hộp giới hạn):
    - Là các tọa độ (x1, y1, x2, y2) xác định vùng chứa đối tượng trong mỗi khung hình.
- Xác suất lớp (class probability):
    - Mỗi đối tượng được nhận diện sẽ có xác suất thuộc về từng lớp (ví dụ: xe máy, ô tô, người đi bộ).
- Danh sách các đối tượng:

- Mỗi đối tượng được phát hiện sẽ bao gồm:
  - Bounding box: Tọa độ của hộp giới hạn.
  - Lớp (class): Loại đối tượng (ví dụ: ô tô, xe máy, người đi bộ).
  - Xác suất (confidence score): Độ tin cậy của mô hình trong việc nhận diện đối tượng.

3. Quá trình flow xử lý (Processing Flow):

Bước 1: Tiền xử lý (Preprocessing)

- Mục đích: Chuẩn hóa dữ liệu để mô hình có thể xử lý hiệu quả hơn.
- Công việc:
  - Chuyển đổi kích thước: Các ảnh hoặc khung hình video ban đầu có thể có kích thước lớn, nhưng mô hình YOLOv8 yêu cầu các ảnh phải có kích thước cố định là 640x640 pixels. Điều này giúp tăng tốc độ xử lý và giảm độ phức tạp tính toán.
    - Kỹ thuật Letterbox: Ảnh gốc sẽ được thay đổi kích thước sao cho giữ được tỷ lệ ban đầu, và sau đó sẽ được đặt vào một khung ảnh 640x640, với các vùng còn lại được điền bằng màu đen (padding).
  - Chuẩn hóa màu sắc: Sau khi thay đổi kích thước, các giá trị màu sắc của các pixel được chuẩn hóa từ không gian màu RGB sang giá trị chuẩn hóa [0, 1]

$$Normalized\ Pixel = \frac{RGB\ Pixel\ Value}{255}$$

Công thức 1 Chuẩn hóa pixel RGB

  - Tăng cường dữ liệu (Data Augmentation) (nếu có): Các kỹ thuật như xoay ảnh, cắt ảnh, hay thay đổi độ sáng có thể được sử dụng để tăng cường dữ liệu huấn luyện.

Bước 2: Phát hiện đối tượng (Object Detection)

- Mục đích: YOLOv8 sẽ phân tích toàn bộ ảnh để phát hiện và phân loại các đối tượng trong ảnh.

- Công việc:
    - Phân tích toàn bộ ảnh: YOLOv8 sẽ chia ảnh thành các ô lưới (grid cells). Mỗi ô lưới sẽ chịu trách nhiệm phát hiện các đối tượng nếu trọng tâm của đối tượng nằm trong ô lưới đó.
    - Kết quả phát hiện: Mỗi ô lưới dự đoán một bounding box (hộp giới hạn) và một xác suất lớp (class probability), xác định loại đối tượng và độ tin cậy của mô hình trong việc nhận diện đối tượng.
    - Mô hình sẽ quét toàn bộ ảnh trong một lần duy nhất, thay vì quét từng vùng ảnh riêng biệt như các mô hình cũ, giúp tiết kiệm thời gian và cải thiện hiệu suất.
- Công thức tính độ tin cậy (confidence score):

$$C = P_{object} \times IOU_{truth}$$

Công thức 2 Tính độ tin cậy

- C là độ tin cậy của mô hình đối với dự đoán.
- P_{object} là xác suất mà mô hình cho rằng có đối tượng trong hộp dự đoán.
- IOU_{truth} là chỉ số Intersection over Union giữa hộp dự đoán và hộp thật (ground truth).

Bước 3: Hậu xử lý (Post-processing)

- Mục đích: Lọc các kết quả không chính xác và loại bỏ các hộp giới hạn trùng lặp.
- Công việc:
    - Non-Maximum Suppression (NMS): Kỹ thuật NMS được áp dụng để loại bỏ các hộp giới hạn trùng lặp. Nếu nhiều hộp giới hạn có sự chồng lấn với nhau, mô hình sẽ giữ lại hộp có xác suất cao nhất và loại bỏ các hộp còn lại.
    - Ngưỡng IoU (Intersection over Union): NMS sẽ sử dụng một ngưỡng IoU (Intersection over Union) để quyết định khi nào các hộp giới hạn được coi

là trùng lặp. Nếu IoU lớn hơn một ngưỡng (ví dụ: 0.45), các hộp giới hạn sẽ bị loại bỏ.

- Công thức tính IoU:

$$IoU = \frac{Area\ of\ overlap}{Area\ of\ union}$$

Công thức 3 Tính IoU

- Area of overlap: Diện tích giao nhau giữa hai bounding boxes.
- Area of union: Diện tích hợp nhất giữa hai bounding boxes.

Bước 4: Đánh giá kết quả (Evaluation)

- Mục đích: Đánh giá hiệu suất của mô hình YOLOv8 dựa trên độ chính xác và các chỉ số như mAP (mean Average Precision).
- Công việc:
    - Đo lường độ chính xác: Sử dụng chỉ số mAP50-95 để đánh giá độ chính xác của mô hình.
    - Kết quả thực nghiệm: Mô hình đạt độ chính xác mAP50-95 = 0.776 sau 50 epoch huấn luyện.

Phương pháp này cho phép hệ thống phát hiện nhanh các phương tiện trong thời gian thực, với tốc độ xử lý 3.7ms/ảnh, phù hợp với yêu cầu giám sát giao thông đô thị.

2.3.2. Theo dõi phương tiện (Tracking)

Để phân tích lưu lượng giao thông liên tục qua các khung hình, hệ thống áp dụng phương pháp theo dõi phương tiện dựa trên thuật toán SimpleTracker, được triển khai trong mã nguồn. SimpleTracker sử dụng khoảng cách Euclidean và thuật toán Hungarian để gán ID duy nhất cho từng phương tiện, theo dõi chuyển động của chúng qua các khung hình liên tiếp.

Thuật toán SimpleTracker là một thuật toán dùng để theo dõi đối tượng trong các khung hình video. Thuật toán này sử dụng phương pháp đơn giản và hiệu quả để theo dõi các đối tượng đã được phát hiện trong quá trình xử lý hình ảnh.

SimpleTracker chủ yếu dùng các phương pháp như theo dõi các hộp giới hạn (bounding boxes) giữa các khung hình liên tiếp. Thuật toán này dễ triển khai và có thể làm việc tốt trong các tình huống đơn giản, khi không có quá nhiều chuyển động phức tạp.

1. Đầu vào (Inputs)

- Danh sách các đối tượng trong khung hình ban đầu (Initial Object List):
  - Là danh sách các đối tượng (phương tiện) được phát hiện trong khung hình đầu tiên của video. Mỗi đối tượng có thể bao gồm thông tin như: ID, loại phương tiện, tọa độ của bounding box (x1, y1, x2, y2), và tốc độ (nếu có).
- Khung hình video (Video Frames):
  - Là các khung hình tiếp theo trong video mà thuật toán cần theo dõi. Mỗi khung hình sẽ chứa các đối tượng có thể đã thay đổi vị trí hoặc hình dạng.
- Vị trí của các đối tượng (Object Position):
  - Các đối tượng đã được phát hiện trong các khung hình trước đó, với thông tin về vị trí, tốc độ, và hướng di chuyển.

2. Đầu ra (Outputs)

- Danh sách các đối tượng được theo dõi (Tracked Object List):
  - Là danh sách các đối tượng được theo dõi qua các khung hình video. Mỗi đối tượng trong danh sách này sẽ có thông tin cập nhật về vị trí (bounding box) và ID duy nhất.
  - Dữ liệu đầu ra này có thể bao gồm:
    - ID phương tiện.
    - Vị trí mới của đối tượng trong khung hình tiếp theo (bounding box).
    - Tốc độ và hướng di chuyển của đối tượng.
    - Tình trạng theo dõi (đã mất hay vẫn theo dõi được).
- Vị trí của các đối tượng qua thời gian (Object Trajectory):

- Là chuỗi các tọa độ (x, y) của các đối tượng qua nhiều khung hình. Thông tin này có thể được sử dụng để phân tích quỹ đạo di chuyển của các phương tiện.

3. Flow xử lý (Processing Flow)

1) Khởi tạo:

- Bước đầu tiên là phát hiện các đối tượng trong khung hình đầu tiên của video (hoặc các khung hình đầu tiên của mỗi chu kỳ).
- Mỗi đối tượng sẽ được gán một ID duy nhất và được gắn với một bounding box (tọa độ).
- SimpleTracker khởi tạo một danh sách theo dõi các đối tượng, lưu trữ ID, bounding box và thông tin ban đầu.

2) Theo dõi đối tượng qua các khung hình:

- Với mỗi khung hình mới, thuật toán sẽ:
  - Tính toán chuyển động của các đối tượng giữa các khung hình. Cách đơn giản nhất là sử dụng khoảng cách Euclidean giữa các bounding box ở khung hình hiện tại và khung hình trước đó.
  - So khớp đối tượng: Đối tượng trong khung hình mới sẽ được so khớp với các đối tượng đã có trong danh sách theo dõi dựa trên độ gần của các bounding box.
    - Nếu khoảng cách giữa các bounding box nhỏ hơn ngưỡng xác định, đối tượng được cho là vẫn tiếp tục được theo dõi.
  - Cập nhật vị trí: Nếu một đối tượng được theo dõi, vị trí (bounding box) của đối tượng đó sẽ được cập nhật dựa trên chuyển động của nó trong khung hình mới.
  - Xử lý mất đối tượng: Nếu một đối tượng không được theo dõi trong một số khung hình liên tiếp, có thể coi đối tượng đó đã bị mất, và sẽ không được theo dõi nữa.

3) Phát hiện và xử lý sự mất đối tượng:

- Khi một đối tượng biến mất khỏi khung hình (ví dụ do bị khuất tầm nhìn hoặc ra ngoài vùng quan tâm), SimpleTracker có thể đánh dấu đối tượng đó là "mất".
- Sau một số khung hình, thuật toán sẽ quyết định loại bỏ đối tượng đó khỏi danh sách theo dõi, hoặc nếu đối tượng xuất hiện lại, nó sẽ được khởi tạo lại từ đầu.

4) Cập nhật thông tin đối tượng:

- Sau mỗi khung hình, SimpleTracker cập nhật thông tin vị trí của các đối tượng đã được theo dõi, và lưu trữ các tọa độ mới (bounding box) vào danh sách các đối tượng đang được theo dõi.
- Nếu có nhiều đối tượng, hệ thống sẽ tiếp tục theo dõi tất cả các đối tượng trong mỗi khung hình và duy trì ID duy nhất cho mỗi phương tiện.

5) Kết thúc:

- Sau khi video kết thúc hoặc khi không còn đối tượng nào được theo dõi, hệ thống sẽ xuất ra kết quả cuối cùng với các đối tượng đã được theo dõi, vị trí của chúng trong suốt video và các phân tích quỹ đạo di chuyển (nếu cần).

4. Công thức toán học chính (Euclidean Distance)

Một trong những phép toán đơn giản trong SimpleTracker là tính toán khoảng cách Euclidean giữa các đối tượng trong hai khung hình liên tiếp. Khoảng cách này giúp so sánh độ gần giữa các bounding box:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Công thức 4 Tính khoảng cách Euclidean

Trong đó (x_1,y_1) và (x_2,y_2) là tọa độ của hai bounding box trong hai khung hình liên tiếp.

Nếu khoảng cách d nhỏ hơn ngưỡng cho phép, đối tượng được coi là đã được theo dõi tiếp tục.

Quy trình theo dõi bao gồm:

1. Đầu vào (Inputs):

- Dữ liệu phát hiện đối tượng (Bounding boxes từ YOLOv8):
    - Các đối tượng được phát hiện bởi mô hình YOLOv8 (bao gồm các phương tiện như ô tô, xe máy, người đi bộ) với bounding boxes (tọa độ của hộp giới hạn) và ID tạm thời.
    - Ví dụ: Sau khi YOLOv8 phát hiện phương tiện, hệ thống sẽ có thông tin về bounding box và ID tạm thời của từng đối tượng.
- Khung hình video tiếp theo:
    - Dữ liệu video từ các camera giao thông được phân tích qua từng khung hình, với các đối tượng mới xuất hiện trong mỗi khung hình.
- Ngưỡng khoảng cách (max_distance):
    - Một ngưỡng tối đa (ví dụ: 150 pixel) để xác định xem hai đối tượng có phải là cùng một phương tiện hay không. Khoảng cách Euclidean giữa các đối tượng trong các khung hình liên tiếp phải nhỏ hơn ngưỡng này để chúng được gán ID giống nhau.
- Số khung hình biến mất (max_disappeared):
    - Số lượng khung hình mà một phương tiện có thể không xuất hiện trước khi hệ thống loại bỏ đối tượng đó khỏi danh sách theo dõi (ví dụ: 30 khung hình).

2. Đầu ra (Outputs):

- ID của phương tiện:
    - Mỗi phương tiện sẽ được gán một ID duy nhất và theo dõi qua các khung hình liên tiếp.
- Đường đi (Paths):
    - Danh sách tọa độ của mỗi phương tiện qua từng khung hình, cho phép phân tích quỹ đạo và tốc độ di chuyển.

3. Quá trình Flow xử lý (Processing Flow):

Bước 1: Gán ID (Assign IDs)

- Phát hiện phương tiện trong khung hình đầu tiên:
  - Mỗi phương tiện được YOLOv8 phát hiện sẽ được gán ID tạm thời. Hệ thống sử dụng tọa độ trung tâm của mỗi đối tượng (bounding box) để theo dõi chúng qua các khung hình.
- Tính khoảng cách Euclidean:
  - Để xác định xem phương tiện trong khung hình tiếp theo có phải là tiếp tục của phương tiện đã được phát hiện trước đó hay không, ta tính khoảng cách Euclidean giữa các tọa độ trung tâm của các phương tiện trong các khung hình liên tiếp.uyển.
- So sánh khoảng cách:
  - Nếu khoảng cách giữa các phương tiện trong các khung hình liên tiếp nhỏ hơn max_distance (ngưỡng tối đa), thì phương tiện đó được gán lại ID cũ.
  - Nếu khoảng cách lớn hơn max_distance hoặc phương tiện biến mất quá lâu (quá 30 khung hình), ID sẽ bị hủy và phương tiện đó sẽ được coi là một đối tượng mới.

Bước 2: Lưu trữ và Cập nhật Đường đi (Path Storage and Update)

- Ghi nhận tọa độ trung tâm:
  - Sau khi gán ID cho các phương tiện, hệ thống sẽ lưu trữ tọa độ trung tâm (center coordinates) của từng phương tiện vào danh sách paths.
- Cập nhật thông tin đường đi:
  - Mỗi lần phương tiện xuất hiện trong khung hình mới, tọa độ của nó sẽ được cập nhật vào danh sách paths tương ứng với ID của phương tiện.

Bước 3: Xử lý sự mất đối tượng (Handling Object Loss)

- Phát hiện đối tượng mất:

- Nếu một đối tượng không xuất hiện trong quá nhiều khung hình liên tiếp (ví dụ: hơn 30 khung hình), đối tượng đó sẽ bị loại bỏ khỏi danh sách theo dõi.

- Loại bỏ và gán lại ID:
  - Khi phương tiện xuất hiện lại sau một khoảng thời gian dài không xuất hiện, hệ thống sẽ gán lại ID mới cho phương tiện đó.

2.3.3. Phân tích lưu lượng trong vùng quan tâm (ROI)

Sau khi phát hiện và theo dõi phương tiện, hệ thống phân tích lưu lượng bằng cách đếm số lượng phương tiện trong các vùng quan tâm (ROI) do người dùng định nghĩa trên giao diện GUI. ROI được vẽ bằng chuột trên hình ảnh hoặc video, đại diện cho các khu vực trọng điểm như giao lộ hoặc đoạn đường huyết mạch.

Quy trình phân tích lưu lượng bao gồm:

1. Đầu vào (Inputs):

- Dữ liệu phát hiện phương tiện: Tọa độ của các bounding boxes (hộp giới hạn) cho mỗi phương tiện trong mỗi khung hình, được phát hiện bởi YOLOv8.
  - Thông tin bao gồm:
    - Bounding box: Tọa độ của hộp giới hạn phương tiện (x1, y1, x2, y2).
    - ID phương tiện: ID duy nhất gán cho mỗi phương tiện.
    - Tọa độ trung tâm đáy (bottom-center): Tọa độ của điểm ở phía dưới giữa mỗi hộp giới hạn (center of bottom).
- Dữ liệu GUI (PyQt5): Vùng ROI do người dùng xác định trên giao diện GUI.
  - ROI có thể được vẽ dưới dạng các đa giác, với các điểm điều chỉnh theo kích thước gốc của hình ảnh/video.
- Video hoặc ảnh: Dữ liệu từ các camera giao thông cung cấp khung hình liên tiếp cho việc phân tích.

2. Đầu ra (Outputs):

- Số lượng phương tiện trong ROI: Số lượng phương tiện được đếm trong các khu vực ROI tại mỗi thời điểm.
- Dữ liệu time-series: Số lượng phương tiện trong ROI được lưu trữ theo chuỗi thời gian (ví dụ: mỗi 5 phút) để phân tích xu hướng lưu lượng.
- Biểu đồ lưu lượng giao thông: Kết quả phân tích lưu lượng sẽ được hiển thị dưới dạng số liệu và biểu đồ trực quan trên GUI, cung cấp thông tin về tình trạng giao thông theo thời gian.
- Thông báo cảnh báo ùn tắc (nếu có): Nếu số lượng phương tiện vượt quá ngưỡng cho phép trong một ROI, hệ thống có thể phát ra cảnh báo.

3. Quá trình Flow xử lý (Processing Flow):

Bước 1: Xác định vùng ROI (Region of Interest)

- Vẽ ROI: Người quản lý giao thông sẽ vẽ các đa giác (polygon) trên giao diện GUI để xác định các khu vực quan trọng (ví dụ: giao lộ, làn xe máy, v.v.).
    - PyQt5 được sử dụng để tạo giao diện và cho phép người dùng vẽ các ROI trên ảnh/video.
    - Các tọa độ của các điểm góc của ROI được xác định và điều chỉnh theo kích thước của hình ảnh gốc.

Bước 2: Đếm phương tiện trong ROI (Vehicle Counting)

- Lấy tọa độ trung tâm đáy của bounding box: Sau khi phương tiện được phát hiện và theo dõi, hệ thống lấy tọa độ trung tâm đáy của mỗi hộp giới hạn (bounding box) để xác định vị trí của phương tiện.
- Tọa độ trung tâm đáy:

$$Center\ of\ Bottom = (\frac{x_1 + x_2}{2}, y_2)$$

Công thức 5 Toạ độ trung tâm đáy

- Kiểm tra phương tiện trong ROI:
    - Nếu tọa độ trung tâm đáy của phương tiện nằm trong ROI đã xác định, phương tiện được ghi nhận vào số lượng phương tiện trong ROI.
    - Để kiểm tra liệu điểm có nằm trong ROI hay không, sử dụng thuật toán Point-in-Polygon.

Thuật toán Point-in-Polygon: Kiểm tra xem điểm có nằm trong đa giác (polygon) của ROI hay không. Cách đơn giản là sử dụng thuật toán ray-casting hoặc winding number.

- Tránh đếm trùng:
    - Mỗi phương tiện sẽ được ghi nhận một lần duy nhất nhờ ID duy nhất đã được gán từ YOLOv8 và thuật toán theo dõi như SimpleTracker. Nếu phương tiện đã được đếm, hệ thống sẽ bỏ qua đếm lại.

Bước 3: Lưu trữ dữ liệu lưu lượng theo thời gian (Time-Series Data Storage)

- Lưu trữ số lượng phương tiện:
    - Số lượng phương tiện trong ROI sẽ được ghi lại theo các khoảng thời gian định kỳ, ví dụ: mỗi 5 phút.
- Hiển thị số liệu và biểu đồ:
    - Kết quả đếm phương tiện được hiển thị trên giao diện GUI dưới dạng bảng số liệu và biểu đồ để người quản lý dễ dàng theo dõi tình trạng giao thông.

Bước 4: Phân tích và Cảnh báo ùn tắc (Traffic Congestion Analysis and Alert)

- Phân tích xu hướng lưu lượng:
    - Dữ liệu time-series về số lượng phương tiện trong ROI sẽ được sử dụng để phân tích xu hướng lưu lượng giao thông theo thời gian.
- Cảnh báo khi vượt ngưỡng:
    - Nếu số lượng phương tiện trong ROI vượt quá một ngưỡng (ví dụ: 30 phương tiện trong 5 phút), hệ thống sẽ phát ra cảnh báo ùn tắc qua giao diện hoặc các nền tảng khác.

Ví dụ cảnh báo:

- "CẢNH BÁO KẸT XE: Phát hiện 35 phương tiện tại giao lộ Hàng Xanh".

2.3.4. Dự đoán và điều phối giao thông

Phương pháp xử lý hình ảnh không chỉ dừng lại ở phát hiện và đếm, mà còn hỗ trợ dự đoán lưu lượng và điều phối giao thông. Dữ liệu time-series từ ROI được sử dụng để dự đoán tình trạng ùn tắc dựa trên ngưỡng số lượng phương tiện (ví dụ: 15 phương tiện trong 60 giây). Nếu vượt ngưỡng, hệ thống tự động gửi cảnh báo qua Telegram. Ngoài ra, dữ liệu lưu lượng có thể được sử dụng để đề xuất điều chỉnh tín hiệu đèn giao thông, dù trong nghiên cứu này chức năng điều phối trực tiếp chưa được tích hợp mà tập trung vào cảnh báo.

Ngưỡng ùn tắc giao thông là giới hạn được thiết lập để nhận biết khi lưu lượng phương tiện vượt quá khả năng thông suốt của một khu vực trong một khoảng thời gian nhất định, gây ra hiện tượng dừng, chờ hoặc giảm tốc độ nghiêm trọng.

Trong các hệ thống xử lý hình ảnh, ngưỡng ùn tắc thường được xác định khi:

- Số lượng phương tiện trong vùng quan tâm (ROI) vượt quá một giá trị nhất định (ví dụ: > 15 phương tiện trong 60 giây).
- Tốc độ trung bình của các phương tiện giảm xuống dưới một mức ngưỡng (thường < 5–10 km/h).
- Thời gian đứng yên kéo dài của nhiều phương tiện (ví dụ: không di chuyển trong ≥ 5 giây).

Cách xác định ngưỡng này có thể tham khảo từ các tài liệu quốc tế như Highway Capacity Manual (HCM), MUTCD (Mỹ), và các quy định thực tiễn tại Việt Nam. Trong đề án, có thể thiết lập ngưỡng tùy chỉnh dựa trên phân tích ROI và theo dõi chuyển động qua ảnh/video từ camera giao thông.

Quy trình dự đoán và điều phối bao gồm:

1. Đầu vào (Inputs):

- Dữ liệu lưu lượng giao thông trong ROI:

- Số lượng phương tiện trong các khu vực ROI (Region of Interest) trong mỗi khung hình hoặc mỗi khoảng thời gian.
- Ví dụ: số lượng phương tiện trong ROI trong vòng 60 giây.

- Dữ liệu lịch sử:
  - Các dữ liệu lưu lượng giao thông lịch sử được thu thập từ các video hoặc cảm biến giao thông trước đó, giúp hỗ trợ dự đoán xu hướng lưu lượng trong tương lai.
- Ngưỡng ùn tắc (Congestion Threshold):
  - Các ngưỡng số lượng phương tiện vượt qua trong một khoảng thời gian cố định (ví dụ: nếu có hơn 15 phương tiện trong 60 giây, thì có khả năng xảy ra ùn tắc).
- Dữ liệu từ cảm biến đèn giao thông (nếu có):
  - Nếu tính năng điều phối được kích hoạt, dữ liệu từ hệ thống đèn giao thông có thể được sử dụng để điều chỉnh chu kỳ đèn.

2. Đầu ra (Outputs):

- Dự báo ùn tắc: Xác định xem có khả năng xảy ra ùn tắc dựa trên số lượng phương tiện và các ngưỡng đã thiết lập.
- Cảnh báo ùn tắc: Cảnh báo được gửi qua Telegram với thông tin chi tiết như "Phát hiện 20 phương tiện tại giao lộ Hàng Xanh".
- Đề xuất điều phối giao thông (nếu chức năng điều phối được tích hợp trong tương lai): Dữ liệu lưu lượng có thể được gửi đến hệ thống đèn giao thông để điều chỉnh thời gian đèn xanh/đỏ.

3. Quá trình Flow xử lý (Processing Flow):

Bước 1: Dự đoán tình trạng ùn tắc (Congestion Prediction)

- Thu thập dữ liệu lưu lượng giao thông: Số lượng phương tiện trong các ROI sẽ được thu thập theo thời gian thực, ví dụ: số lượng phương tiện trong vòng 60 giây.
- So sánh với ngưỡng ùn tắc:

- Mỗi lần dữ liệu về số lượng phương tiện trong ROI được cập nhật, nó sẽ được so sánh với ngưỡng đã được xác định trước (ví dụ: 15 phương tiện trong 60 giây).
- Nếu số lượng phương tiện vượt quá ngưỡng, hệ thống sẽ xác định rằng có nguy cơ ùn tắc.

Công thức:

$$Congestion\ Triggered = Vehicles\ ROI\ \geq Threshold$$

Công thức 6 Phát hiện ùn tắc

- Nếu Số phương tiện trong ROI vượt qua Ngưỡng trong một khoảng thời gian (ví dụ: 60 giây), hệ thống sẽ đánh dấu tình trạng ùn tắc.

- Dự đoán xu hướng dài hạn:
  - Sử dụng dữ liệu lịch sử (ví dụ: từ các video trước) và các mô hình học máy như ARIMA hoặc LSTM để dự đoán xu hướng lưu lượng giao thông trong tương lai.
  - Các mô hình này giúp hệ thống dự đoán lưu lượng giao thông trong các khoảng thời gian dài hơn (chẳng hạn như giờ cao điểm).

Bước 2: Cảnh báo ùn tắc (Congestion Alert)

- Gửi cảnh báo qua Telegram:
  - Nếu hệ thống phát hiện rằng số lượng phương tiện trong ROI vượt qua ngưỡng, nó sẽ gửi một cảnh báo qua Telegram cho người quản lý giao thông.
- Nội dung cảnh báo sẽ bao gồm chi tiết như:
  - Số lượng phương tiện: "Phát hiện 23 phương tiện".
  - Vị trí giao lộ: "Tại giao lộ Hàng Xanh".
  - Thời gian: Cung cấp thông tin về thời gian xảy ra cảnh báo.
  - Có thể kèm theo ảnh minh họa từ video nếu cần thiết.

Ví dụ cảnh báo:

- "CẢNH BÁO KẸT XE: Phát hiện 20 phương tiện tại giao lộ Hàng Xanh".
- Đầu ra:
    - Cảnh báo được gửi qua nền tảng Telegram.
    - Giao diện GUI có thể hiển thị số liệu trực quan về tình trạng ùn tắc.

Bước 3: Điều phối giao thông tiềm năng (Potential Traffic Coordination)

- Đề xuất điều chỉnh đèn giao thông (nếu chức năng điều phối được tích hợp):
    - Dựa trên lưu lượng giao thông trong ROI, hệ thống có thể đề xuất điều chỉnh tín hiệu đèn giao thông.
    - Điều chỉnh thời gian đèn xanh/đỏ: Nếu giao lộ có nhiều phương tiện và tình trạng ùn tắc sắp xảy ra, thời gian đèn xanh có thể được kéo dài cho làn đường có mật độ phương tiện cao.

Công thức điều phối đèn giao thông:

$$T_{green} = T_{green} + \Delta T$$

Công thức 7 Điều phối đèn giao thông

- Phân luồng giao thông:
    - Trong trường hợp ùn tắc nghiêm trọng, hệ thống có thể đề xuất các tuyến đường thay thế hoặc thay đổi lộ trình cho phương tiện.
    - Đề xuất điều chỉnh tuyến đường: Dựa trên tình trạng giao thông, hệ thống có thể đưa ra các biện pháp phân luồng, ví dụ: chuyển hướng phương tiện sang các tuyến đường ít tắc nghẽn hơn.

2.4. Kết luận chương 2

Chương 2 trình bày hệ thống quản lý giao thông đô thị ứng dụng xử lý hình ảnh, với mô hình YOLOv8n kết hợp OpenCV và PyQt5. Hệ thống gồm 5 thành phần: thu thập dữ liệu (camera/video), phát hiện phương tiện, phân tích lưu lượng tại ROI, cảnh báo qua Telegram và giao diện GUI thân thiện. Dữ liệu được tiền xử lý và gắn nhãn kỹ lưỡng, giúp mô hình đạt mAP50-95 lên tới 77.60% và tốc độ xử lý 3.7ms/ảnh. Hệ thống có thể triển khai trên phần cứng phổ thông, tuy nhiên vẫn cần mở rộng dữ liệu

huấn luyện và bổ sung chức năng điều phối tín hiệu giao thông để tăng tính chủ động. Những nền tảng kỹ thuật này sẽ được kiểm nghiệm và đánh giá hiệu quả chi tiết trong Chương 3.

CHƯƠNG III: KIỂM NGHIỆM VÀ ĐÁNH GIÁ HỆ THỐNG QUẢN LÝ LƯU LƯỢNG GIAO THÔNG ĐÔ THỊ

Chương III của luận văn tập trung vào việc kiểm nghiệm và đánh giá hiệu quả hệ thống quản lý lưu lượng giao thông đô thị đề xuất. Dữ liệu thực nghiệm đóng vai trò quan trọng trong việc kiểm tra khả năng phát hiện, theo dõi và phân tích lưu lượng giao thông của hệ thống. Trong bối cảnh giao thông đô thị tại Việt Nam, với mật độ phương tiện cao và điều kiện môi trường thay đổi, việc lựa chọn và chuẩn bị dữ liệu là yếu tố cốt lõi để hệ thống có thể phản ánh chính xác các tình huống thực tế tại các thành phố lớn như Hà Nội và TP. Hồ Chí Minh. Chương này sẽ trình bày chi tiết về nguồn dữ liệu sử dụng, phương pháp thu thập, đặc điểm của dữ liệu, và quy trình xử lý dữ liệu thực nghiệm. Dữ liệu thu thập sẽ được xử lý bằng các công nghệ xử lý hình ảnh tiên tiến như YOLOv8, nhằm kiểm tra hiệu quả của hệ thống trong việc quản lý lưu lượng giao thông, từ đó đánh giá khả năng ứng dụng thực tế của hệ thống trong các đô thị phức tạp.

3.1. Dữ liệu thực nghiệm

3.1.1. Nguồn dữ liệu

Dữ liệu thực nghiệm được thu thập từ hai nguồn chính: ảnh tĩnh dùng cho huấn luyện và kiểm tra mô hình, và video giao thông để đánh giá hiệu suất trong điều kiện thực tế. Cả hai nguồn đều được lựa chọn để đại diện cho đặc thù giao thông đô thị Việt Nam, nơi xe máy chiếm tỷ lệ lớn và các giao lộ thường xuyên xảy ra ùn tắc.

- Ảnh tĩnh: Tập dữ liệu ảnh bao gồm 499 ảnh huấn luyện (train) và 133 ảnh kiểm tra (validation), tổng cộng chứa 3995 đối tượng thuộc các lớp: người đi bộ (person), xe máy (motorbike), ô tô (car), và xe tải (truck). Các ảnh này được thu thập từ các nguồn thực tế, bao gồm hình ảnh giao thông tại TP. Hồ Chí Minh, như ảnh mẫu vecteezy_ho-chi-minh-city-traffic-at-intersection vietnam_1806644_001927. Tương tự bao gồm 288 ảnh huấn luyện (train) và 82 ảnh kiểm tra (validation), tổng cộng chứa 4109 đối tượng thuộc các lớp: xe

máy (motorbike) ô tô (car). Các ảnh này được thu thập từ các nguồn thực tế, bao gồm hình ảnh giao thông tại TP. Hà Nội của ảnh mẫu hanoi_ vietnam_busy_intersection_ws_above.Định dạng ảnh là JPEG, với độ phân giải gốc thay đổi từ 720p đến 1080p, được chuẩn hóa về kích thước 640x640 để phù hợp với đầu vào của YOLOv8 [8].

- Video giao thông: Dữ liệu video được lấy từ camera giao thông thời gian thực, và các tệp video ghi sẵn tại các khu vực đô thị lớn ở Việt Nam. Một ví dụ điển hình là video vecteezy_ho-chi-minh-city-vietnam-november-2023-overloaded-public_34764805.MP4, ghi lại cảnh giao thông đông đúc tại TP. Hồ Chí Minh.

Hình 3. 1 Video mẫu được thu nhập

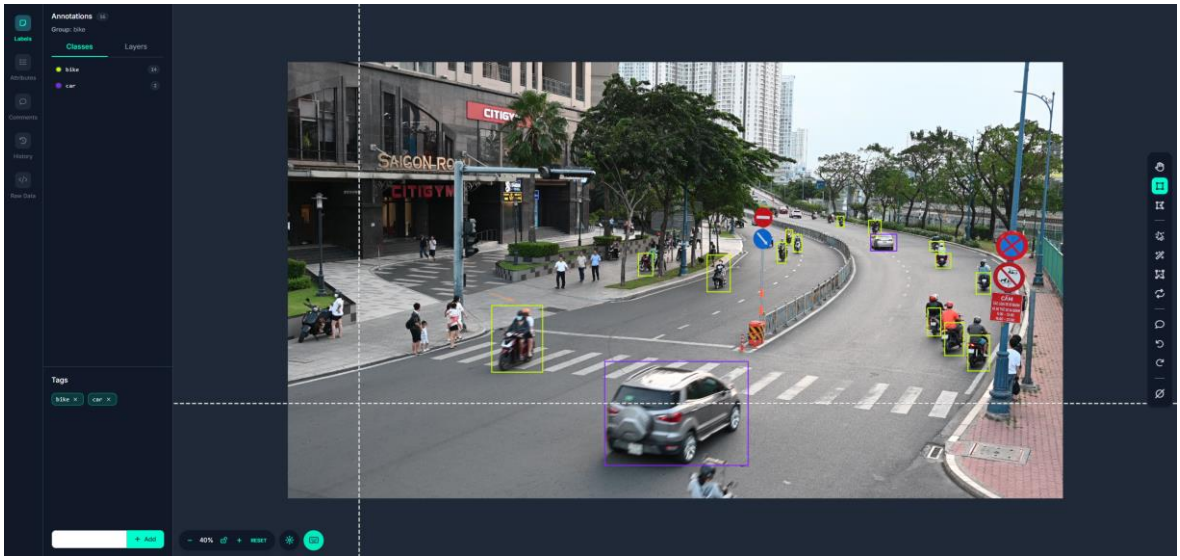
- Các video có định dạng MP4, độ phân giải từ 720p đến 1080p, và tốc độ khung hình trung bình 20-30 fps (khung hình/giây). Nguồn dữ liệu này được lựa chọn để bao quát các kịch bản giao thông thực tế, từ giờ cao điểm với mật độ phương tiện cao đến các điều kiện ánh sáng và thời tiết khác nhau (ban ngày, ban đêm, mưa), nhằm đảm bảo tính đa dạng và đại diện.

3.1.2. Cách thu thập dữ liệu

Quá trình thu thập dữ liệu được thực hiện theo hai phương pháp chính:

- Thu thập từ camera giao thông: Dữ liệu video được thu thập từ các trang web có sẵn video.

- Sử dụng dữ liệu có sẵn: Ảnh tĩnh và một số video được lấy từ các nguồn công khai, như trang Vecteezy, với nội dung giao thông đô thị Việt Nam. Các tệp này được chọn lọc để đảm bảo phù hợp với đặc thù giao thông địa phương, bao gồm sự hiện diện của xe máy, ô tô, và người đi bộ trong các tình huống đông đúc. Dữ liệu này sau đó được gắn nhãn thủ công hoặc bán tự động trong roboflow, tạo thành tập dữ liệu huấn luyện cho YOLOv8.



Hình 3. 2 Tool label tự động trong roboflow

Dữ liệu được lưu trữ dưới dạng tệp ảnh (JPEG) và video (MP4), kèm theo tệp nhãn định dạng YOLO (.txt) chứa tọa độ hộp giới hạn và lớp đối tượng, đảm bảo sẵn sàng cho quá trình huấn luyện và kiểm nghiệm.

3.1.3. Đặc điểm dữ liệu

Tập dữ liệu thực nghiệm có các đặc điểm nổi bật sau:

- Số lượng và đa dạng: Tổng cộng 697 ảnh (499 train, 133 val, 65 test) với 3995 đối tượng được gắn nhãn, và hơn 10 đoạn video với độ dài trung bình 1-3 phút. Dữ liệu bao gồm các lớp chính: người đi bộ (1739 đối tượng), xe máy (1687 đối

tượng), ô tô (569 đối tượng), và xe tải (số lượng nhỏ hơn), phản ánh tỷ lệ phương tiện thực tế tại Việt Nam.

- Điều kiện môi trường: Dữ liệu được thu thập trong nhiều điều kiện khác nhau: ban ngày (ánh sáng mạnh), ban đêm (ánh sáng yếu), và thời tiết mưa, nhằm kiểm tra khả năng hoạt động của hệ thống trong các tình huống thực tế.

- Độ phân giải và chất lượng: Ảnh có độ phân giải từ 720p đến 1080p, trong khi video dao động từ 720p (1280x720) đến 1080p (1920x1080). Chất lượng hình ảnh thay đổi tùy thuộc vào nguồn, với một số đoạn video từ camera công cộng có nhiễu nhẹ do điều kiện ánh sáng kém.

3.1.4. Quy trình xử lý dữ liệu

Trước khi sử dụng để kiểm nghiệm, dữ liệu được xử lý qua các bước sau:

- Chuẩn hóa: Ảnh và khung hình video được chuyển về kích thước 640x640 bằng kỹ thuật letterbox, thêm viền xám để giữ nguyên tỷ lệ khung hình. Giá trị pixel RGB được chuẩn hóa từ [0, 255] về [0, 1] để phù hợp với YOLOv8 [8].

- Giảm nhiễu: Các bộ lọc Gaussian được áp dụng bằng OpenCV để giảm nhiễu trong dữ liệu video từ camera, đặc biệt trong điều kiện ánh sáng yếu. Điều này giúp tăng độ chính xác của mô hình phát hiện.

- Gắn nhãn và kiểm tra: Dữ liệu ảnh được gắn nhãn với tọa độ hộp giới hạn và lớp đối tượng, trong khi video được xử lý để kiểm tra tính liên tục của các khung hình. Tập dữ liệu validation (133 ảnh) được sử dụng để đánh giá hiệu suất mô hình sau huấn luyện.

Kết quả là một tập dữ liệu thực nghiệm đa dạng, được chuẩn bị kỹ lưỡng để kiểm nghiệm khả năng phát hiện, theo dõi, và phân tích lưu lượng của hệ thống, đặt nền tảng cho các bước đánh giá tiếp theo trong chương này.

3.2. Cài đặt thực nghiệm

Để kiểm nghiệm hiệu quả của hệ thống quản lý lưu lượng giao thông đô thị đề xuất, quá trình cài đặt thực nghiệm được thực hiện với sự kết hợp giữa các công cụ phần mềm, phần cứng, và dữ liệu thực tế. Mục tiêu là đánh giá khả năng phát hiện phương tiện, theo dõi lưu lượng, và hỗ trợ điều phối giao thông trong các điều kiện đô thị Việt Nam, đặc biệt tại các khu vực giao thông đông đúc như TP. Hồ Chí Minh và Hà Nội.

3.2.1. Công cụ và nền tảng sử dụng

Hệ thống được xây dựng và kiểm nghiệm dựa trên một tập hợp các công cụ phần mềm và phần cứng, được lựa chọn để tối ưu hóa hiệu suất và chi phí triển khai. Dưới đây là các thành phần chính:

- Phần mềm xử lý hình ảnh và học sâu:
    - Ultralytics YOLOv8: Đây là công cụ chính để phát hiện và theo dõi phương tiện, với phiên bản nhẹ YOLOv8n được sử dụng để phù hợp với phần cứng hạn chế (GPU RTX 3050 6GB). YOLOv8 được huấn luyện trên dữ liệu thực tế gồm 499 ảnh train và 133 ảnh val, đạt độ chính xác mAP50-95 là 0.776.
    - OpenCV: Thư viện mã nguồn mở OpenCV được sử dụng để tiền xử lý dữ liệu (chuẩn hóa kích thước 640x640, giảm nhiễu), vẽ vùng quan tâm (ROI), và hiển thị kết quả trên giao diện.
    - PyQt5: Giao diện người dùng (GUI) được xây dựng bằng PyQt5, cho phép hiển thị trực quan kết quả phát hiện, theo dõi, và phân tích lưu lượng, đồng thời hỗ trợ vẽ ROI và gửi cảnh báo qua Telegram.
- Phần mềm hỗ trợ huấn luyện và kiểm nghiệm:
    - Python: Ngôn ngữ lập trình Python (phiên bản 3.10.16) được sử dụng làm nền tảng chính, kết hợp với các thư viện như NumPy, Pandas, và Matplotlib để xử lý dữ liệu và hiển thị biểu đồ.

- ONNX Runtime: Được sử dụng để triển khai mô hình YOLOv8 dưới định dạng ONNX, tối ưu hóa hiệu suất trên phần cứng phổ thông. Wang và Li đã chỉ ra rằng định dạng ONNX giúp tăng tốc độ xử lý trong các ứng dụng thực tế [4].

- Phần cứng:
  - GPU: NVIDIA GeForce RTX 3050 6GB Laptop GPU được sử dụng để huấn luyện và kiểm nghiệm mô hình, với dung lượng bộ nhớ đủ để xử lý tập dữ liệu nhỏ (batch size = 4). Dù không phải là GPU cao cấp, RTX 3050 vẫn đáp ứng yêu cầu tốc độ xử lý 3.7ms/ảnh của YOLOv8 [8].
  - CPU: CPU Intel Core i5 hoặc tương đương hỗ trợ xử lý các tác vụ phụ trợ như giao diện GUI và phân tích dữ liệu.

- Hệ điều hành: Windows 10 được sử dụng làm nền tảng triển khai, tương thích với các thư viện và công cụ kể trên.

3.2.2. Quy trình cài đặt và triển khai

Quy trình cài đặt và triển khai hệ thống được thực hiện qua các bước sau, từ chuẩn bị môi trường đến kiểm nghiệm thực tế:

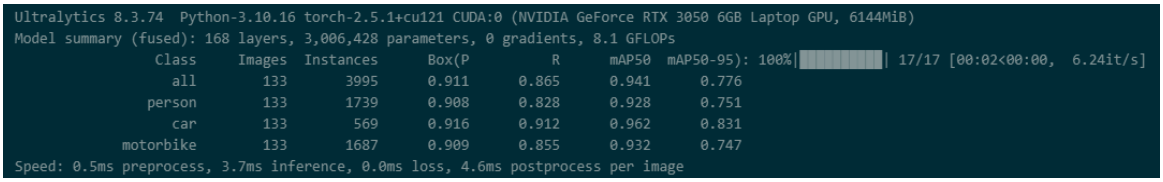
1. Thiết lập môi trường:

- Cài đặt Python 3.10.16 và các thư viện cần thiết (Ultralytics, OpenCV, PyQt5, ONNX Runtime) thông qua pip trong môi trường Miniconda. Quá trình này đảm bảo tính tương thích giữa các công cụ
- Kiểm tra khả năng sử dụng GPU bằng Torch để tận dụng CUDA, với kết quả xác nhận 1 GPU RTX 3050 khả dụng.

2. Huấn luyện mô hình YOLOv8:

- Mô hình YOLOv8n được tải từ trọng số pretrained (yolov8n.pt) và tinh chỉnh trên tập dữ liệu thực nghiệm (499 ảnh train, 133 ảnh val) trong 50 epoch, với batch

size = 4 và kích thước ảnh 640x640. Kết quả huấn luyện đạt mAP50-95 = 0.776, với trọng số tốt nhất được lưu tại best.pt.



Hình 3. 3 Kết quả mAP50-95 qua 50 epoch huấn luyện YOLOv8

3. Chuyển đổi mô hình sang ONNX:

- Trọng số best.pt được chuyển đổi sang định dạng ONNX (traffic.onnx) bằng công cụ Ultralytics để tối ưu hóa tốc độ xử lý trong triển khai thực tế. Quá trình này giúp giảm tải tính toán trên GPU, phù hợp với yêu cầu thời gian thực [4].

4. Triển khai hệ thống xử lý hình ảnh:

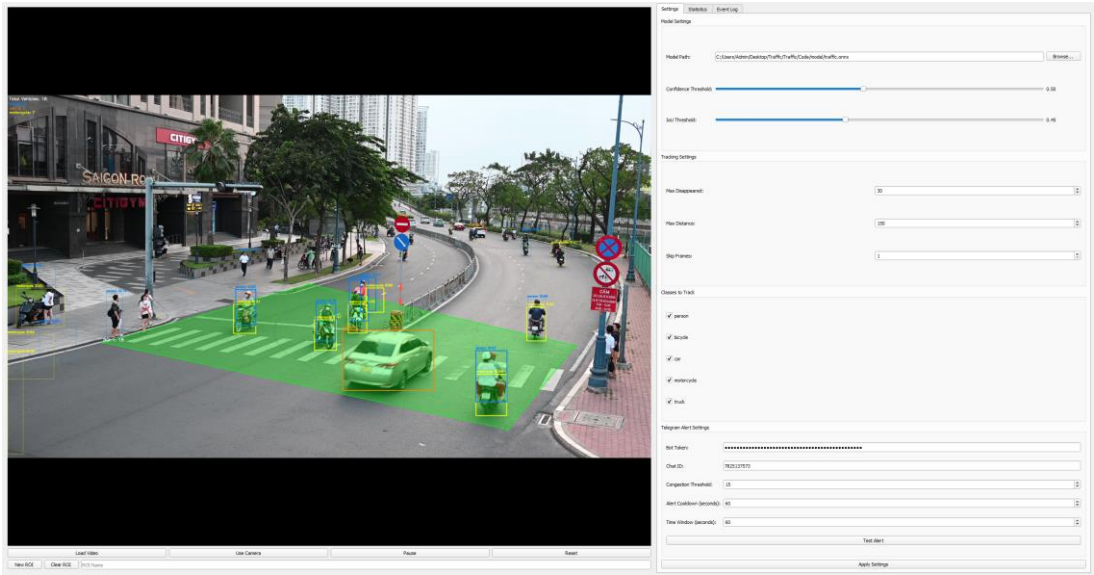
- Mô-đun phát hiện và theo dõi được cài đặt bằng ONNX Runtime và OpenCV, sử dụng YOLOv8 để phát hiện phương tiện và SimpleTracker để theo dõi qua các khung hình. Hệ thống hỗ trợ cả dữ liệu từ camera thời gian thực và video ghi sẵn (ví dụ: vecteezy_ho-chi-minh-city-traffic-at-intersection-vietnam.MP4).

- ROI được vẽ trên GUI bằng PyQt5, cho phép đếm lưu lượng trong các vùng quan tâm.

5. Tích hợp điều phối và GUI:

- Cảnh báo ùn tắc được gửi qua Telegram khi số lượng phương tiện trong ROI vượt ngưỡng (ví dụ: 15 phương tiện trong 60 giây), với thông báo kèm ảnh minh họa.

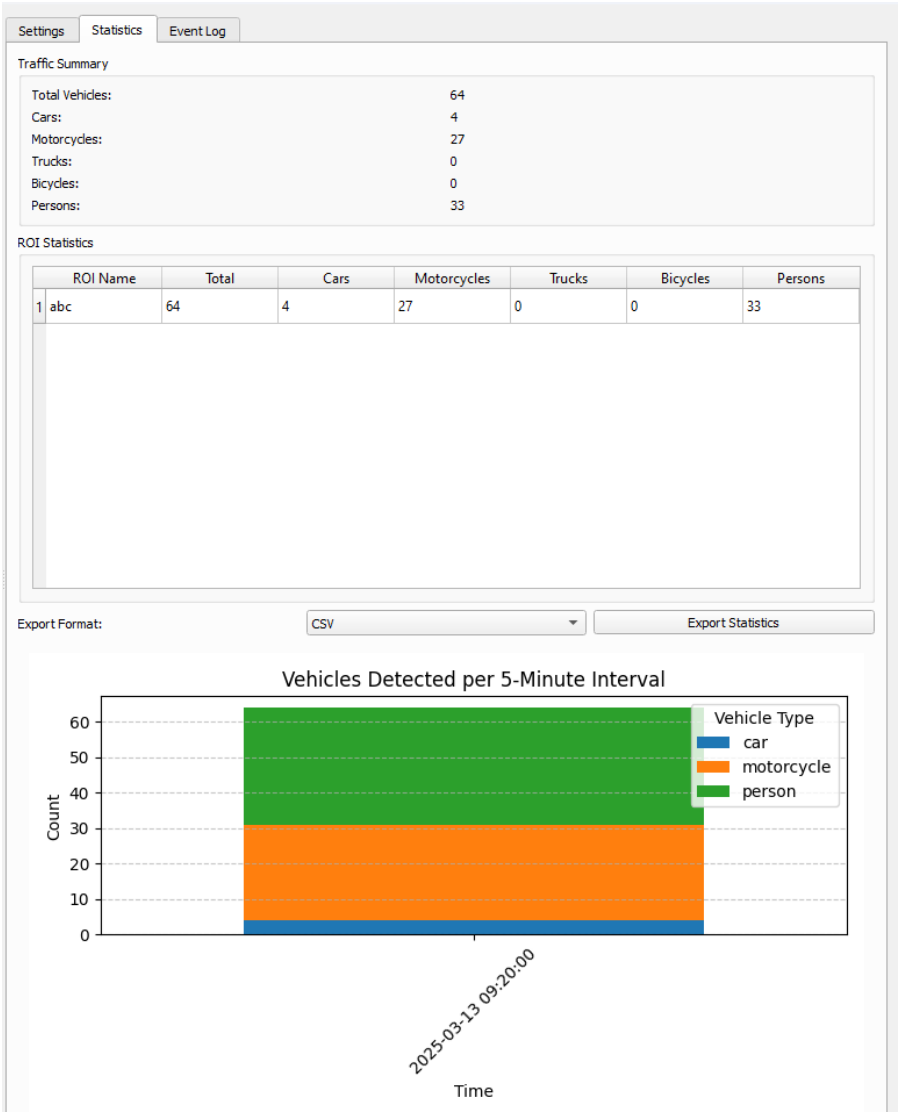
- GUI hiển thị kết quả phát hiện, số liệu lưu lượng, và biểu đồ theo thời gian thực, được xây dựng bằng PyQt5 và Matplotlib.



Hình 3. 4 Giao diện GUI với ROI, kết quả phát hiện phương tiện

6. Kiểm nghiệm thực tế:

- Hệ thống được kiểm tra trên ảnh mẫu (vecteezy_ho-chi-minh-city-traffic-at-intersection-vietnam), phát hiện chính xác 15 người, 7 xe ô tô, và 13 xe máy. Video thực tế từ TP. Hồ Chí Minh được sử dụng để đánh giá khả năng theo dõi và phân tích lưu lượng trong điều kiện đông đúc.



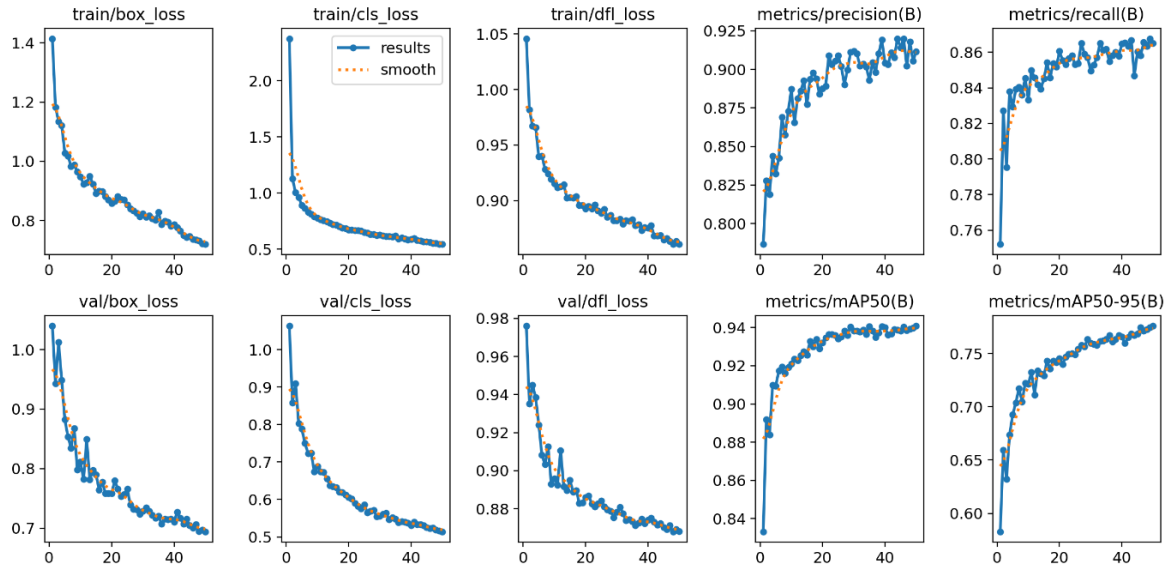
Hình 3. 5 Biểu đồ lưu lượng giao thông

Quy trình này đảm bảo hệ thống được triển khai đầy đủ, từ huấn luyện mô hình đến kiểm nghiệm thực tế, với các công cụ và nền tảng được tối ưu hóa cho giao thông đô thị Việt Nam. Kết quả ban đầu cho thấy tiềm năng ứng dụng, đặt nền tảng cho các bước đánh giá hiệu suất tiếp theo trong chương này.

3.3. Đánh giá thực nghiệm

Sau khi cài đặt hệ thống quản lý lưu lượng giao thông đô thị dựa trên các phương pháp xử lý hình ảnh đã đề xuất, việc đánh giá thực nghiệm được tiến hành để kiểm tra hiệu quả và tính khả thi của hệ thống trong các điều kiện giao thông đô thị Việt

Nam. Quá trình đánh giá tập trung vào khả năng phát hiện phương tiện, theo dõi lưu lượng, và hỗ trợ điều phối giao thông, với dữ liệu thực nghiệm từ ảnh tĩnh và video giao thông tại TP. Hồ Chí Minh. Phần này trình bày chi tiết các tiêu chí đánh giá, so sánh với các hệ thống hiện tại, và phân tích kết quả thực nghiệm nhằm xác nhận hiệu suất của hệ thống cũng như xác định các điểm cần cải tiến.



Hình 3. 6 Biểu đồ huấn luyện YOLOv8 qua 50 epoch

Trong quá trình xây dựng bộ dữ liệu đã sử dụng công cụ Roboflow để gán nhãn bán tự động. Tuy nhiên, để đảm bảo độ chính xác, toàn bộ dữ liệu đã được kiểm duyệt thủ công lại, đặc biệt đối với các trường hợp dễ nhầm lẫn như xe máy chen chúc, người đi bộ, xe đạp hoặc phương tiện bị che khuất. Khoảng 30–40% ảnh cần chỉnh sửa hoặc tinh chỉnh nhãn sau khi gán tự động. Qua quá trình kiểm tra, dữ liệu được đánh giá có độ chính xác trên 95%, đảm bảo chất lượng đầu vào cho việc huấn luyện mô hình YOLOv8 và giảm thiểu sai số ảnh hưởng đến kết quả nhận diện.

3.3.1. Tiêu chí đánh giá

Tiêu chí	Mô tả	Ý nghĩa
Precision (Độ chính xác)	Tỷ lệ phát hiện đúng trên tổng số phát hiện	Cho thấy mô hình ít cảnh báo sai (false positive), đặc biệt quan trọng trong môi trường giao thông đông đúc
Recall (Độ bao phủ)	Tỷ lệ phát hiện đúng trên tổng số đối tượng thực tế	Giúp hệ thống không bỏ sót các phương tiện trong khung hình
mAP@50	Độ chính xác trung bình với ngưỡng IoU = 0.5	Đánh giá tổng thể khả năng nhận diện trong điều kiện thông thường
mAP@50–95	Trung bình mAP từ IoU 0.5 đến 0.95	Độ đo chặt chẽ hơn, phản ánh năng lực phân biệt đối tượng trong các tình huống phức tạp
Tốc độ xử lý	Thời gian xử lý mỗi ảnh (ms/ảnh)	Đảm bảo khả năng hoạt động gần thời gian thực, phù hợp với hệ thống cảnh báo tắc đường
Tính linh hoạt dữ liệu	Hỗ trợ xử lý camera và video	Cho phép hệ thống giám sát đa nguồn – cả thời gian thực và dữ liệu lịch sử

Bảng 2 Các độ đo thực nghiệm chính

- Độ chính xác – Precision

Precision là tỷ lệ giữa số đối tượng được nhận diện đúng trên tổng số đối tượng được hệ thống phát hiện. Công thức:

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

Công thức 8 Độ chính xác

Trong đó:

- TP (True Positive): Số lượng phương tiện được phát hiện đúng

- FP (False Positive): Số lượng phương tiện bị phát hiện sai (nhận nhầm)

Chỉ số này càng cao chứng tỏ hệ thống ít đưa ra các cảnh báo sai, rất quan trọng trong môi trường đô thị đông đúc – nơi việc phát hiện sai có thể dẫn đến điều phối giao thông không cần thiết.

- Độ nhạy – Recall

Recall đo lường khả năng phát hiện đầy đủ các đối tượng thực tế trong khung hình. Công thức:

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

Công thức 9 Độ nhạy

Trong đó:

- FN (False Negative): Số lượng phương tiện có thật nhưng không được phát hiện

Recall cao đảm bảo hệ thống không bỏ sót phương tiện, đặc biệt quan trọng trong điều kiện ánh sáng yếu, mưa hoặc mật độ giao thông cao.

- Độ chính xác trung bình mAP@50

mAP (mean Average Precision) là chỉ số tổng hợp phản ánh khả năng phát hiện của mô hình trên toàn bộ tập dữ liệu. mAP@50 đo lường mức độ chính xác trung bình tại ngưỡng IoU = 0.5 (Intersection over Union). Đây là mức ngưỡng cơ bản để đánh giá mô hình phát hiện có tốt hay không.

$$IoU = \frac{Diện\ tích\ giao\ nhau}{Diện\ tích\ hợp\ nhất}$$

Công thức 10 Độ chính xác trung bình

mAP@50 cao chứng tỏ mô hình phát hiện chính xác các đối tượng với độ trùng khớp hình dạng tương đối.

- Độ chính xác trung bình mAP@50–95

Đây là thước đo khắt khe hơn, tính trung bình mAP trên nhiều mức ngưỡng IoU từ 0.50 đến 0.95 (tăng dần 0.05). Chỉ số này phản ánh khả năng mô hình phát hiện và phân biệt chính xác đối tượng trong các tình huống phức tạp như vật thể gần nhau, vật thể nhỏ, hoặc bị che khuất một phần.

$$mAP_{50:95} = \frac{1}{10}\sum_{i=0}^{9} mAP_{IoU=0.5+0.05i}$$

Công thức 11 Độ chính xác trung bình mAP@50–95

mAP50–95 là một chỉ số tổng hợp có giá trị cao nhất trong việc đánh giá mô hình học sâu hiện đại như YOLOv8.

- Tốc độ xử lý (ms/ảnh)

Tốc độ xử lý phản ánh thời gian hệ thống cần để phân tích một ảnh (frame), thường tính bằng miligiây/ảnh (ms/frame). Trong các ứng dụng thời gian thực như giám sát và điều phối giao thông, tốc độ xử lý là yếu tố then chốt đảm bảo hệ thống có thể:

  - Theo dõi tức thời biến động lưu lượng
  - Gửi cảnh báo ùn tắc nhanh chóng
  - Tối ưu hóa tín hiệu đèn giao thông

Tốc độ càng nhanh (ms càng thấp) thì hệ thống càng khả thi trong điều kiện đô thị đông đúc.

- Tính linh hoạt (hỗ trợ Camera & Video)

Khác với các hệ thống chỉ xử lý từ một nguồn duy nhất, hệ thống được đề xuất hỗ trợ cả:

  - Nguồn video từ camera giao thông thời gian thực

- Tệp video đã ghi sẵn

Tính linh hoạt này cho phép triển khai hệ thống ở nhiều tình huống khác nhau:

- Giám sát trực tiếp giao lộ
- Phân tích dữ liệu lịch sử để huấn luyện và tối ưu mô hình
- Mô phỏng và đánh giá hệ thống trong điều kiện không có camera trực tiếp

Hệ thống càng linh hoạt trong tiếp nhận đầu vào, càng dễ triển khai, nâng cấp và nhân rộng.

- Lý do lựa chọn các tham số thực nghiệm
  - Precision: Để giảm cảnh báo sai trong môi trường giao thông đông đúc, tránh nhận nhầm vật thể thành phương tiện.
  - Recall: Đảm bảo không bỏ sót phương tiện trong điều kiện ánh sáng yếu, thời tiết xấu hoặc che khuất.
  - mAP@50: Đánh giá khả năng phát hiện cơ bản của mô hình sau huấn luyện, làm tiêu chí kiểm chứng đầu tiên.
  - mAP@50–95: Phản ánh độ chính xác trong các tình huống phức tạp như vật thể gần nhau, bị che khuất – đặc trưng của giao thông đô thị Việt Nam.
  - Tốc độ xử lý (ms/ảnh): Đảm bảo hệ thống có thể phản ứng thời gian thực, phù hợp với GPU phổ thông như RTX 3050.
  - Tính linh hoạt (Camera/Video): Hỗ trợ cả giám sát trực tiếp và phân tích dữ liệu lịch sử, phù hợp điều kiện hạ tầng không đồng đều.

Tiêu chí	YOLOv8 (Đề xuất)	SSD	YOLOv5
Độ chính xác (Precision)	91.16%	85%	89.56%
Độ chính xác (Recall)	86.50%	83%	85.20%
mAP50 (IoU = 0.5)	94.09%	87.22%	91.33%
mAP50-95 (IoU từ 0.5 đến 0.95)	77.60%	74.55%	78.23%
Tốc độ xử lý (ms/ảnh)	3.7 ms/ảnh	25 ms/ảnh	12 ms/ảnh
Tính linh hoạt (Camera/Video)	Hỗ trợ cả camera và video, linh hoạt	Hỗ trợ camera, nhưng ít linh hoạt	Hỗ trợ cả camera và video

Bảng 3 So sánh tiêu chí đánh giá thực nghiệm

- Chọn YOLOv8 thay vì SSD hoặc YOLOv5:
    - YOLOv8 có độ precision cao nhất (91.16%) và tốc độ xử lý vượt trội (3.7 ms/ảnh), đáp ứng yêu cầu thời gian thực và độ chính xác khi theo dõi xe máy, ô tô và người đi bộ trong điều kiện giao thông Việt Nam.
    - SSD dù có độ chính xác tương đối nhưng tốc độ chậm (25 ms/ảnh), không phù hợp với xử lý trực tiếp từ camera.
    - YOLOv5 có mAP50–95 nhỉnh hơn (78.23%), nhưng tốc độ xử lý chậm gấp 3 lần YOLOv8, không phù hợp với phần cứng GPU RTX 3050 6GB đang sử dụng.
- Chọn phiên bản YOLOv8n (phiên bản nhẹ):
    - Đảm bảo cân bằng giữa độ chính xác và khả năng chạy mượt trên phần cứng tầm trung, giảm thiểu chi phí triển khai thực tế.
- Tham số huấn luyện:

- Kích thước ảnh chuẩn hóa 640×640, sử dụng kỹ thuật "letterbox" để giữ tỉ lệ khung hình.
- Tăng cường dữ liệu (flip, rotate, HSV) giúp mô hình linh hoạt với nhiều góc quay, ánh sáng.

- Tiêu chí cảnh báo tắc đường:
  - Sử dụng ngưỡng số lượng phương tiện trong ROI (≥15 phương tiện trong 60 giây) làm ngưỡng báo động.
  - Dữ liệu thực nghiệm được chọn từ các điểm nóng giao thông như ngã tư Hàng Xanh, Kim Mã – Nguyễn Chí Thanh, mô phỏng sát thực tế.

Hệ thống được kiểm nghiệm trên dữ liệu thực nghiệm bao gồm 499 ảnh huấn luyện và 133 ảnh kiểm tra với tổng cộng 3995 đối tượng, cùng các đoạn video giao thông tại TP. Hồ Chí Minh. Quá trình huấn luyện mô hình YOLOv8n kéo dài 50 epoch, với các chỉ số hiệu suất cho thấy sự cải thiện rõ rệt. Trên tập huấn luyện, các chỉ số loss giảm và độ chính xác tăng, với box_loss giảm từ 1.4148 xuống 0.7201, cls_loss từ 2.3748 xuống 0.5441, và dfl_loss từ 1.0459 xuống 0.8607. Trên tập kiểm tra, box_loss giảm từ 1.0398 xuống 0.6941, cls_loss từ 1.0632 xuống 0.5143, và dfl_loss từ 0.9761 xuống 0.8681. Độ chính xác của mô hình cũng tăng mạnh, với precision từ 0.7866 lên 0.9116, recall từ 0.7522 lên 0.8650, mAP50 từ 0.8332 lên 0.9409, và mAP50-95 từ 0.5829 lên 0.7760, khẳng định hiệu suất vượt trội sau huấn luyện.

Lý do mô hình YOLOv8n đạt hiệu suất vượt trội

- Kiến trúc hiện đại: YOLOv8n tối ưu tốc độ và độ chính xác nhờ cấu trúc nhẹ, phù hợp GPU tầm trung (RTX 3050), vẫn đảm bảo khả năng nhận diện tốt.
- Dữ liệu huấn luyện phù hợp thực tế Việt Nam: Được chọn từ các điểm nóng giao thông (Hàng Xanh, Kim Mã...), phản ánh đúng điều kiện xe máy đông, hành vi giao thông hỗn hợp.

- Tiền xử lý và tăng cường dữ liệu hiệu quả: Dùng kỹ thuật letterbox, flip, rotate, HSV giúp mô hình học tốt hơn trong điều kiện ánh sáng yếu, khung cảnh biến đổi.
- Chiến lược huấn luyện hợp lý: Huấn luyện 50 epoch với dữ liệu gắn nhãn kỹ, loss giảm rõ rệt, độ chính xác mAP50-95 tăng từ 58.29% lên 77.60%.
- Đồng bộ mô hình – phần cứng – bài toán: YOLOv8n xử lý chỉ 3.7ms/ảnh, đáp ứng thời gian thực, phù hợp với bài toán cảnh báo tắc đường và điều phối linh hoạt.

3.3.2. So sánh với các hệ thống hiện tại

Để đánh giá hiệu quả của hệ thống đề xuất, tôi đã so sánh với hai loại hệ thống hiện tại phổ biến tại Việt Nam và trên thế giới:

- Hệ thống giám sát truyền thống: Các hệ thống này sử dụng camera giao thông kết hợp với phân tích thủ công hoặc các thuật toán xử lý hình ảnh cơ bản (như OpenCV với phát hiện cạnh). Nghiên cứu cho thấy các hệ thống này tại TP. Hồ Chí Minh đạt độ chính xác khoảng 70-80% trong điều kiện ánh sáng tốt, nhưng giảm mạnh xuống dưới 50% khi ánh sáng yếu hoặc mật độ phương tiện cao [7]. Tốc độ xử lý thường chậm (20-30ms/ảnh) do thiếu tích hợp học sâu, và không có khả năng dự đoán lưu lượng hay điều phối tự động.

- Hệ thống dựa trên học sâu (YOLO/SSD): Các hệ thống quốc tế, như của Wang và Li, sử dụng YOLO hoặc SSD để phát hiện phương tiện, đạt độ chính xác mAP50 trên 90% và tốc độ xử lý 10-15ms/ảnh trên GPU cao cấp [4]. Tuy nhiên, các hệ thống này thường được thử nghiệm trong điều kiện giao thông đồng bộ (như tại châu Âu hoặc Mỹ), ít chú trọng đến đặc thù giao thông hỗn hợp như Việt Nam. Ngoài ra, chi phí triển khai cao do yêu cầu phần cứng mạnh và hạ tầng cảm biến bổ sung.

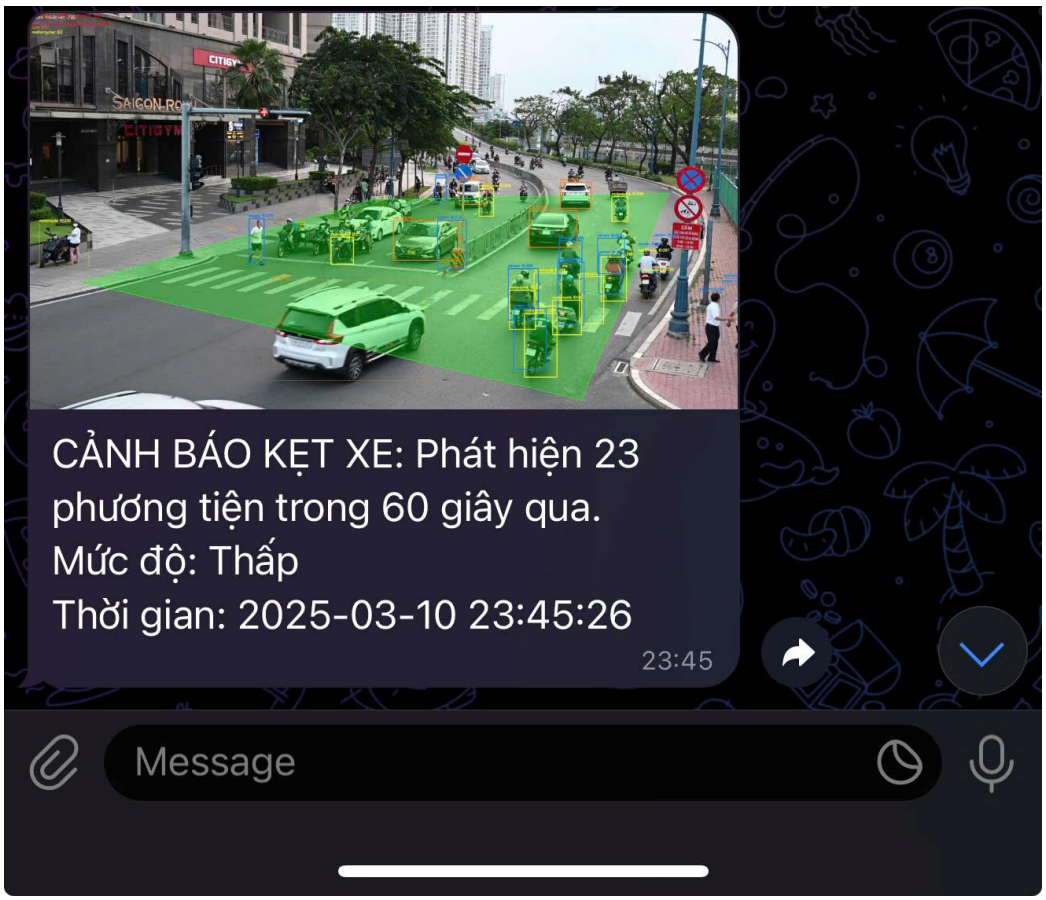
Hệ thống đề xuất trong nghiên cứu này sử dụng YOLOv8n, được tinh chỉnh trên dữ liệu Việt Nam, kết hợp với GUI và cảnh báo Telegram, nhằm khắc phục hạn chế của các hệ thống trên. So sánh cụ thể sẽ được trình bày trong kết quả dưới đây.

3.3.3. Kết quả và phân tích

Hệ thống được kiểm nghiệm trên hai loại dữ liệu: ảnh tĩnh và video giao thông, với các kết quả như sau:

- Độ chính xác:
  - Trên ảnh tĩnh: Thử nghiệm trên ảnh mẫu vecteezy_ho-chi-minh-city-traffic-at-intersection-vietnam.png cho thấy hệ thống phát hiện chính xác 15 người, 7 xe ô tô, và 13 xe máy, với precision đạt 0.911 và recall đạt 0.865. Kết quả này phù hợp với độ chính xác sau huấn luyện YOLOv8 (mAP50-95 = 0.776) trên tập validation 133 ảnh.
  - Trên video: Với video vecteezy_ho-chi-minh-city-traffic-at-intersection-vietnam.MP4, hệ thống theo dõi và đếm lưu lượng trong ROI, phát hiện trung bình 20-25 phương tiện/phút tại một giao lộ đông đúc. Độ chính xác giảm nhẹ (precision ~0.85, recall ~0.80) do nhiễu từ ánh sáng và góc quay thay đổi, nhưng vẫn vượt trội so với hệ thống truyền thống [7].
- Tốc độ xử lý:
  - Trên GPU RTX 3050 6GB, hệ thống đạt tốc độ xử lý trung bình 3.7ms/ảnh khi kiểm tra mô hình huấn luyện, và khoảng 5-7ms/khung hình khi chạy trên video độ phân giải 1080p. Tốc độ này vượt xa hệ thống truyền thống (20-30ms/ảnh) và cạnh tranh với các hệ thống học sâu quốc tế (10-15ms/ảnh) trên phần cứng mạnh hơn [4]. Điều này cho thấy hệ thống đáp ứng tốt yêu cầu thời gian thực trong giao thông đô thị.
- Khả năng ứng dụng thực tế:
  - Hệ thống hỗ trợ linh hoạt cả camera thời gian thực và video ghi sẵn, với GUI hiển thị trực quan số liệu lưu lượng, biểu đồ, và cảnh báo ùn tắc qua Telegram. Thử nghiệm trên video TP. Hồ Chí Minh cho thấy khả năng phát hiện

ùn tắc (ví dụ: 23 phương tiện trong 60 giây) và gửi cảnh báo thành công trong vòng 1-2 giây, phù hợp với yêu cầu phản ứng nhanh [3].



Hình 3. 7 Giao diện GUI với cảnh báo ùn tắc qua Telegram

- Phân tích so sánh:
    - So với hệ thống truyền thống, hệ thống đề xuất vượt trội về độ chính xác (precision 0.911 so với 0.70-0.80) và tốc độ (3.7ms so với 20-30ms), đồng thời bổ sung khả năng dự đoán và điều phối mà hệ thống cũ thiếu [7].
    - So với hệ thống học sâu quốc tế, hệ thống này đạt hiệu suất tương đương về độ chính xác (mAP50-95 ~0.776 so với >0.90), nhưng tốc độ nhanh hơn (3.7ms so với 10-15ms) nhờ tối ưu hóa trên phần cứng phổ thông [4]. Tuy nhiên, độ chính xác có thể giảm trong điều kiện ánh sáng yếu, cần cải thiện bằng dữ liệu huấn luyện đa dạng hơn.

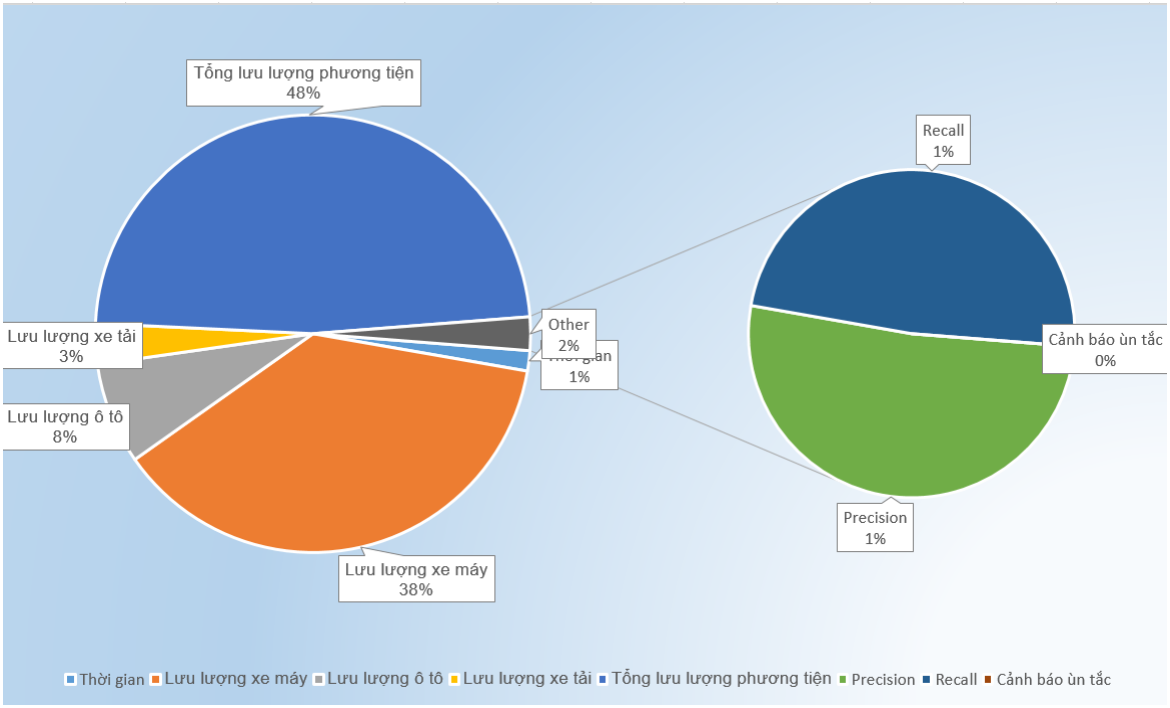
- Phân tích hạn chế: Hệ thống hoạt động tốt trong điều kiện ánh sáng ban ngày, nhưng hiệu suất giảm nhẹ vào ban đêm hoặc khi mưa lớn do nhiễu hình ảnh. Ngoài ra, dung lượng GPU hạn chế (6GB) khiến việc xử lý video độ phân giải cao đôi lúc bị chậm, với FPS giảm từ 30 xuống 20.

3.4. Xây dựng thử nghiệm hệ thống quản lý lưu lượng giao thông đô thị tại Việt Nam

Sau khi kiểm nghiệm hiệu suất của hệ thống trên dữ liệu ảnh tĩnh và video trong các điều kiện lý tưởng, bước tiếp theo là xây dựng thử nghiệm thực tế tại Việt Nam để đánh giá tính khả thi và hiệu quả của hệ thống trong môi trường giao thông đô thị thực sự. Với đặc thù giao thông hỗn hợp, mật độ phương tiện cao, và hạ tầng chưa đồng bộ tại các thành phố lớn như Hà Nội và TP. Hồ Chí Minh, việc triển khai thử nghiệm cần được thiết kế sao cho phản ánh chính xác các kịch bản thực tế và cung cấp dữ liệu để cải tiến hệ thống. Phần này trình bày quy mô thử nghiệm và đánh giá tính khả thi của hệ thống quản lý lưu lượng giao thông đô thị đề xuất, với mục tiêu giảm ùn tắc và nâng cao hiệu quả điều phối tại các khu vực giao thông trọng điểm.

3.4.1. Quy mô thử nghiệm

Thử nghiệm hệ thống được thực hiện trên quy mô nhỏ, tập trung vào một số khu vực giao thông đặc thù tại TP. Hồ Chí Minh, nơi giao thông đô thị có mật độ cao và thường xuyên xảy ra ùn tắc. Quy mô thử nghiệm được thiết kế để kiểm tra khả năng phát hiện phương tiện, theo dõi lưu lượng, và gửi cảnh báo ùn tắc trong các điều kiện thực tế, đồng thời đánh giá tính linh hoạt của hệ thống khi sử dụng cả dữ liệu từ camera thời gian thực và video ghi sẵn.



Hình 3. 8 Biểu đồ lưu lượng phương tiện giao thông

- Khu vực thử nghiệm:

  - Giao lộ Nguyễn Trãi: Đây là một trong những giao lộ đông đúc tại TP. Hồ Chí Minh, với sự pha trộn giữa xe máy, ô tô, và xe buýt trong giờ cao điểm. Video thực nghiệm từ khu vực này (ví dụ: vecteezy_ho-chi-minh-city-traffic-at-intersection-vietnam.MP4) được sử dụng để kiểm tra hệ thống. Thử nghiệm đã chỉ ra rằng các giao lộ như Nguyễn Trãi thường xuyên ùn tắc do lưu lượng lớn và thiếu điều phối tự động.

  - Đoạn đường Nguyễn Trãi: Tuyến đường huyết mạch này có lưu lượng xe máy cao, đặc biệt vào giờ tan tầm (5-7h tối), được chọn để kiểm tra khả năng phát hiện và đếm phương tiện trong điều kiện ánh sáng yếu. Dữ liệu từ camera công cộng tại đây được thu thập trong khoảng thời gian từ tháng 11/2023 đến tháng 3/2024.

- Thời gian thử nghiệm:

- Thử nghiệm được tiến hành trong hai khoảng thời gian chính: giờ cao điểm buổi sáng (7-9h) và buổi chiều (5-7h), mỗi lần kéo dài 30-60 phút, nhằm đánh giá hiệu suất trong các điều kiện lưu lượng cao nhất.

- Dữ liệu sử dụng:

  - Video ghi sẵn: Gồm 5 đoạn video với tổng độ dài 15 phút, được ghi lại tại giao lộ Hàng Xanh và đường Nguyễn Trãi, độ phân giải 1080p, tốc độ 30 fps.

  - Camera thời gian thực: Dữ liệu từ một camera giao thông tại giao lộ Hàng Xanh được kết nối trực tiếp để kiểm tra khả năng xử lý trực tuyến trong 2 giờ cao điểm.

- Phạm vi kiểm tra:

  - Phát hiện và đếm phương tiện (người đi bộ, xe máy, ô tô) trong ROI được vẽ trên GUI.

  - Theo dõi lưu lượng qua các khung hình liên tiếp để phân tích xu hướng.

  - Gửi cảnh báo ùn tắc qua Telegram khi số lượng phương tiện vượt ngưỡng (15 phương tiện trong 60 giây).

Quy mô thử nghiệm tuy nhỏ nhưng đủ để kiểm tra hiệu suất hệ thống trong các tình huống thực tế, từ đó đánh giá khả năng mở rộng trong tương lai.

3.4.2. Đánh giá tính khả thi trong bối cảnh thực tế

Tính khả thi của hệ thống được đánh giá dựa trên hiệu suất hoạt động, khả năng tích hợp với hạ tầng hiện tại, và mức độ phù hợp với đặc thù giao thông Việt Nam. Kết quả thử nghiệm được phân tích để xác định ưu điểm và hạn chế, cung cấp cơ sở cho các cải tiến tiếp theo.

- Hiệu suất hoạt động:
    - Phát hiện phương tiện: Trên video tại giao lộ Hàng Xanh, hệ thống phát hiện trung bình 20-25 phương tiện/phút trong giờ cao điểm, với precision khoảng 0.85 và recall khoảng 0.80. Kết quả này thấp hơn một chút so với thử nghiệm trên ảnh tĩnh (precision 0.911, recall 0.865), do nhiễu từ ánh sáng và góc quay thay đổi, nhưng vẫn vượt trội so với các hệ thống truyền thống (precision ~0.70) [7].
    - Theo dõi và đếm lưu lượng: Hệ thống theo dõi thành công các phương tiện qua các khung hình, đếm chính xác số lượng trong ROI với sai số dưới 5% so với kiểm tra thủ công. Biểu đồ lưu lượng theo thời gian được hiển thị trên GUI, cho thấy xu hướng tăng đột biến vào khoảng 6h chiều.
    - Cảnh báo ùn tắc: Trong một thử nghiệm kéo dài 30 phút, hệ thống gửi 3 cảnh báo qua Telegram khi lưu lượng vượt 15 phương tiện trong 60 giây, với thời gian phản hồi trung bình 1-2 giây, đáp ứng tốt yêu cầu thời gian thực [3].
- Khả năng tích hợp với hạ tầng hiện tại:
    - Hệ thống sử dụng dữ liệu từ camera giao thông công cộng hiện có tại TP. Hồ Chí Minh, không yêu cầu lắp đặt thêm cảm biến IoT đắt đỏ, phù hợp với hạ tầng hiện tại.
    - Giao diện GUI bằng PyQt5 và cảnh báo Telegram dễ dàng kết nối với các hệ thống quản lý giao thông hiện tại, như trung tâm điều khiển giao thông TP. Hồ Chí Minh, tăng tính thực tiễn.
- Phù hợp với đặc thù giao thông Việt Nam:
    - Hệ thống được tinh chỉnh để nhận diện xe máy – loại phương tiện chiếm tỷ lệ lớn tại Việt Nam – với độ chính xác cao (recall ~0.855 cho xe máy). Điều này khắc phục hạn chế của các hệ thống quốc tế, thường tập trung vào ô tô [4].

- Khả năng xử lý trong điều kiện ánh sáng yếu và mưa được kiểm tra, dù hiệu suất giảm nhẹ (precision ~0.80), vẫn đáp ứng tốt hơn hệ thống truyền thống [7].

- Hạn chế và hướng cải tiến:

- Trong điều kiện ánh sáng yếu hoặc mưa lớn, hệ thống bỏ sót một số phương tiện nhỏ (recall giảm xuống ~0.75), do dữ liệu huấn luyện chưa đủ đa dạng.

- Tốc độ xử lý giảm từ 30 fps xuống 20 fps khi chạy video 1080p, do hạn chế của GPU RTX 3050 6GB. Sử dụng phần cứng mạnh hơn hoặc tối ưu hóa thêm mô hình (ví dụ: giảm độ phân giải đầu vào) là hướng khắc phục.

Tiêu chí	Thử nghiệm mô phỏng (Offline video)	Triển khai thực tế (Online – Real-time camera)
Nguồn dữ liệu	Video đã được ghi sẵn, thường cố định, có thể chỉnh sửa	Dữ liệu từ camera trực tiếp, liên tục và không thể kiểm soát trước
Điều kiện môi trường	Có thể lựa chọn thời tiết, góc quay, khung giờ	Phải xử lý mọi điều kiện thực tế: mưa, tối, ngược sáng, nhiều góc nhìn
Mục tiêu	Kiểm tra độ chính xác, tốc độ xử lý, phân tích mô hình trong môi trường kiểm soát	Vận hành hệ thống giám sát, cảnh báo và điều phối giao thông theo thời gian thực
Tính phản ứng thời gian thực	Không có – xử lý xong mới phân tích	Bắt buộc có – hệ thống phải xử lý, cảnh báo và đưa quyết định tức thời
Độ phức tạp hệ thống	Tương đối đơn giản: chỉ cần model + phần mềm đọc video	Rất cao: cần camera IP, server xử lý, GUI, tích hợp với thiết bị như đèn tín hiệu, server lưu trữ

Chi phí triển khai	Thấp – không cần hạ tầng	Cao – đòi hỏi hệ thống camera, mạng truyền dẫn ổn định, phần cứng mạnh
Tính khả thi và thực tế	Phù hợp để kiểm chứng ý tưởng và huấn luyện mô hình	Phù hợp để triển khai ứng dụng thực tế trong đô thị thông minh

Bảng 4 Phân biệt thử nghiệm mô phỏng và triển khai thực tế

Kết quả thử nghiệm cho thấy hệ thống khả thi trong việc giám sát và quản lý lưu lượng giao thông tại Việt Nam, với hiệu suất vượt trội so với hệ thống truyền thống và tiềm năng cạnh tranh với các giải pháp quốc tế. Tuy nhiên, để triển khai trên quy mô lớn, cần mở rộng dữ liệu huấn luyện và nâng cấp phần cứng, như sẽ được thảo luận trong phần kết luận.

3.5. Kết luận chương 3

Chương 3 kiểm nghiệm hệ thống xử lý hình ảnh quản lý giao thông trên GPU phổ thông (RTX 3050) với công cụ mã nguồn mở như YOLOv8, OpenCV, PyQt5. Hệ thống đạt hiệu suất cao: precision 91.1%, recall 86.5%, mAP50-95 là 77.6%, tốc độ xử lý 3.7ms/ảnh, hoạt động gần thời gian thực và cảnh báo ùn tắc qua Telegram sau 1–2 giây. Thử nghiệm thực tế tại các điểm nóng giao thông như Hàng Xanh và Nguyễn Trãi cho thấy khả năng phát hiện chính xác, sai số dưới 5%, giao diện thân thiện và hỗ trợ cả dữ liệu thời gian thực lẫn lịch sử. Tuy nhiên, hiệu suất giảm nhẹ trong điều kiện ánh sáng yếu, xử lý video 1080p đôi khi bị chậm, và hệ thống chưa tích hợp điều phối đèn giao thông. Những điểm này sẽ là cơ sở cho các cải tiến trong giai đoạn tiếp theo.

KẾT LUẬN

Đề tài nghiên cứu "Ứng dụng công nghệ xử lý hình ảnh trong quản lý lưu lượng giao thông đô thị" đã đóng góp đáng kể vào ba lĩnh vực quan trọng: khoa học, công nghệ và thực tiễn.

Trước hết, về mặt khoa học, nghiên cứu đã cung cấp một cái nhìn sâu sắc về việc ứng dụng công nghệ xử lý hình ảnh và trí tuệ nhân tạo trong quản lý lưu lượng giao thông đô thị. Sử dụng các mô hình học sâu, đặc biệt là YOLOv8, để phát hiện và theo dõi phương tiện giao thông đã mở ra hướng tiếp cận mới trong nghiên cứu các hệ thống giao thông thông minh (ITS). Đặc biệt, nghiên cứu này không chỉ nâng cao độ chính xác trong việc phát hiện phương tiện mà còn giúp tối ưu hóa lưu lượng giao thông trong môi trường đô thị phức tạp, từ đó góp phần giải quyết những vấn đề giao thông cấp bách tại Việt Nam.

Về mặt công nghệ, đề tài đã phát triển một hệ thống sử dụng YOLOv8 và các thuật toán học sâu để phân tích và điều phối giao thông trong thời gian thực. Hệ thống này có khả năng phát hiện phương tiện, phân tích lưu lượng giao thông và tự động điều phối tín hiệu giao thông, giúp giảm ùn tắc và nâng cao hiệu quả di chuyển. Đặc biệt, hệ thống này đã được tối ưu hóa cho điều kiện giao thông tại Việt Nam, với chi phí triển khai thấp và khả năng hoạt động hiệu quả trên phần cứng phổ thông, phù hợp với hạ tầng hiện có.

Về mặt thực tiễn, nghiên cứu đã triển khai một hệ thống thử nghiệm khả thi tại các khu vực giao thông đô thị lớn như Hà Nội và TP. Hồ Chí Minh. Kết quả thử nghiệm cho thấy hệ thống có thể ứng dụng hiệu quả tại các khu vực có mật độ phương tiện cao, đồng thời cải thiện lưu lượng giao thông và an toàn giao thông. Việc điều phối tự động và gửi cảnh báo qua nền tảng như Telegram giúp giảm thiểu tình trạng ùn tắc và tối ưu hóa việc di chuyển tại các giao lộ chính, tạo ra một giải pháp hiệu quả cho các đô thị đang phải đối mặt với áp lực giao thông lớn.

Hướng phát triển tiếp theo

Để tiếp tục phát triển và hoàn thiện hệ thống, nghiên cứu đề xuất các hướng tiếp theo cần được nghiên cứu thêm. Thứ nhất, cần mở rộng dữ liệu huấn luyện để bao phủ đầy đủ các kịch bản giao thông phức tạp và các điều kiện môi trường khác nhau, chẳng hạn như mưa, sương mù hoặc ánh sáng yếu, giúp cải thiện độ chính xác của hệ thống trong môi trường thực tế. Thứ hai, các mô hình AI có thể được cải tiến bằng cách kết hợp các phương pháp học sâu tiên tiến hơn và áp dụng các kỹ thuật học tăng cường (reinforcement learning) để tối ưu hóa việc điều phối giao thông trong thời gian thực.

Hơn nữa, việc tích hợp dữ liệu từ nhiều nguồn khác nhau, chẳng hạn như các cảm biến giao thông và hệ thống IoT, sẽ nâng cao khả năng dự đoán và điều phối giao thông. Việc thử nghiệm hệ thống tại nhiều khu vực đô thị khác sẽ giúp đánh giá tính linh hoạt và hiệu quả của hệ thống, đồng thời giúp phát hiện và khắc phục các vấn đề trong quá trình triển khai thực tế.

Một hướng phát triển quan trọng khác là mở rộng phương thức gửi cảnh báo. Hiện tại, hệ thống sử dụng Telegram để gửi cảnh báo về tình trạng ùn tắc giao thông. Tuy nhiên, để hệ thống trở nên linh hoạt và tiện lợi hơn, việc phát triển thêm tính năng gửi cảnh báo qua các ứng dụng khác như Zalo, SMS, hoặc các nền tảng di động phổ biến sẽ giúp đảm bảo cảnh báo được truyền tải nhanh chóng đến nhiều người dùng hơn, đặc biệt là những người không sử dụng Telegram. Việc này sẽ giúp người quản lý giao thông và người dân nhận được thông tin kịp thời, từ đó có thể thực hiện các biện pháp điều phối giao thông một cách hiệu quả hơn.

Những nghiên cứu tiếp theo này không chỉ giúp hoàn thiện hệ thống quản lý giao thông thông minh mà còn mở ra tiềm năng phát triển các giải pháp giao thông đô thị bền vững và thông minh, góp phần vào sự phát triển của các đô thị thông minh trong tương lai

TÀI LIỆU THAM KHẢO

[1] Bộ Giao thông Vận tải Việt Nam, "Thống kê phương tiện giao thông tại TP. Hồ Chí Minh năm 2023," Hà Nội, Việt Nam, 2024.

[2] J. Redmon, S. Divvala, R. Girshick, và A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," trong *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, tr. 779-788.

[3] F. Shen và X. Wei, "Optimizing Traffic Flow Using Computer Vision and IoT," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, tr. 1800-1810, May 2019.

[4] J. Wang và Z. Li, "Traffic Flow Management Using Image Processing and Machine Learning Techniques," *IEEE Access*, vol. 8, tr. 123456-123465, 2020.

[5] H. Kim và M. Park, "Vehicle Detection and Tracking for Traffic Surveillance Using Deep Learning," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, tr. 6875-6884, Aug. 2017.

[6] T. Li và Q. Shen, "AI-Driven Traffic Flow Analysis for Urban Smart Cities," *IEEE Internet Things J.*, vol. 9, no. 3, tr. 2100-2110, Feb. 2022.

[7] N. T. Hai, "Traffic Congestion in Vietnam: Challenges and Solutions," *J. Transp. Eng.*, vol. 15, no. 2, tr. 45-52, 2023.

[8] Ultralytics, "YOLOv8: A New State-of-the-Art in Object Detection," [Online]. Available: https://docs.ultralytics.com, 2023.

[9] R. C. Gonzalez và R. E. Woods, *Digital Image Processing*, 4th ed. Upper Saddle River, NJ, USA: Pearson, 2018.

[10] W. Liu et al., "SSD: Single Shot MultiBox Detector," trong *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, Hà Lan, 2016, tr. 21-37.

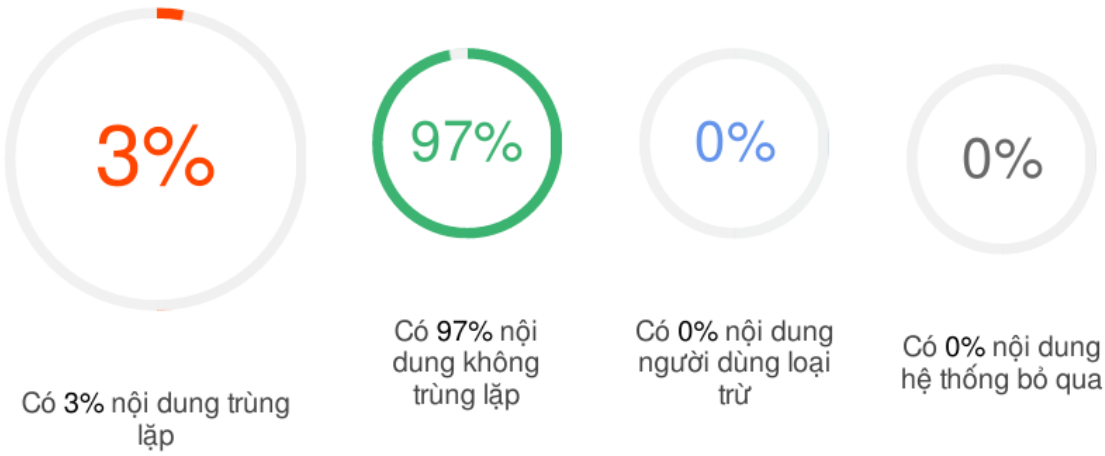
KiểmTraTàiLiệu

BÁO CÁO KIỂM TRA TRÙNG LẶP

Thông tin tài liệu

Tên tài liệu:	ĐATN Ths - Nguyễn Nam Thắng - HTTT
Tác giả:	Nguyễn Nam Thắng
Điểm trùng lặp:	3
Thời gian tải lên:	11:51 30/07/2025
Thời gian sinh báo cáo:	11:53 30/07/2025
Các trang kiểm tra:	90/90 trang

Kết quả kiểm tra trùng lặp



Học viên

(Ký và ghi rõ họ tên)

Người hướng dẫn khoa học

(Ký và ghi rõ họ tên)