

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

TÓM TẮT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI – 2024

Đề án tốt nghiệp được hoàn thành tại:
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: TS. Đỗ Thị Liên

Phản biện 1:

.....

Phản biện 2:

.....

Đề án tốt nghiệp sẽ được bảo vệ trước Hội đồng chấm đề án tốt nghiệp
thạc sĩ tại Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: giờ ngày tháng năm

Có thể tìm hiểu đề án tốt nghiệp tại:

- Thư viện của Học viện Công nghệ Bưu chính Viễn thông.

LỜI MỞ ĐẦU

1. Lý do chọn của đề tài

Việt Nam đang đô thị hóa nhanh, đặc biệt ở các thành phố lớn như Hà Nội và TP. Hồ Chí Minh. Điều này kéo theo sự gia tăng nhanh chóng về phương tiện giao thông, dẫn đến ùn tắc kéo dài, tai nạn giao thông tăng cao, và hiệu quả điều phối giao thông giảm sút. Trong khi đó, các phương pháp quản lý truyền thống hiện nay (dựa vào con người hoặc thiết bị đơn lẻ) không còn đáp ứng được nhu cầu phản ứng nhanh trong điều kiện giao thông phức tạp.

Công nghệ xử lý hình ảnh kết hợp trí tuệ nhân tạo (AI) đã chứng minh khả năng:

- Phát hiện phương tiện tự động
- Dự đoán lưu lượng giao thông
- Hỗ trợ điều phối tín hiệu đèn giao thông

Tuy nhiên, tại Việt Nam, việc triển khai còn hạn chế do đặc thù giao thông hỗn hợp, hạ tầng không đồng bộ và dữ liệu thực tế chưa được khai thác tối ưu. Vì vậy, việc nghiên cứu một hệ thống xử lý hình ảnh phù hợp với điều kiện giao thông Việt Nam là cần thiết và cấp bách.

2. Tổng quan về vấn đề nghiên cứu:

Nghiên cứu thuộc lĩnh vực giao thông thông minh (ITS), xử lý hình ảnh, thị giác máy tính và trí tuệ nhân tạo. Trên thế giới, nhiều mô hình đã đạt kết quả tích cực:

- Wang & Li: Hệ thống AI điều phối đèn giao thông giảm thời gian chờ 20%
- Kim & Park: Phát hiện phương tiện bằng CNN đạt độ chính xác >90%
- Li & Shen: Kết hợp camera và IoT để dự báo lưu lượng thời gian thực

Tuy nhiên, phần lớn các nghiên cứu này triển khai tại quốc gia có hạ tầng giao thông hiện đại. Trong khi đó, giao thông Việt Nam có đặc thù như:

- Mật độ xe máy cao
- Hành vi lái xe khó đoán

- Điều kiện thời tiết thay đổi liên tục

=> Cần phát triển mô hình tùy chỉnh, phù hợp với thực tế địa phương.

3. Mục đích nghiên cứu

- Tối ưu hóa quản lý lưu lượng giao thông bằng xử lý hình ảnh và AI
- Tự động hóa việc phát hiện, đếm, và điều phối phương tiện
- Ứng dụng thực tế tại Hà Nội, TP.HCM với điều kiện giao thông đặc thù
- Đề xuất mô hình có tính mở rộng cho các đô thị thông minh tương lai

4. Đối tượng và phạm vi nghiên cứu

4.1. Đối tượng nghiên cứu

- Các thuật toán xử lý hình ảnh và học sâu (YOLO, OpenCV...)
- Hệ thống giám sát giao thông từ camera/video
- Mô hình điều phối và cảnh báo dựa trên phân tích thời gian thực

4.2. Phạm vi nghiên cứu

Không gian: Các đô thị lớn như Hà Nội, TP.HCM

- Thời gian: Dữ liệu giao thông trong giai đoạn 2022–2024
- Ứng dụng: Triển khai thử nghiệm hệ thống giao thông thông minh phù hợp với thực tế Việt Nam

5. Phương pháp nghiên cứu

- Tổng hợp tài liệu trong và ngoài nước về ITS, AI, xử lý ảnh
- Thu thập & xử lý dữ liệu từ camera giao thông (chuẩn hóa, giảm nhiễu...)
- Xây dựng hệ thống sử dụng YOLOv8 + OpenCV + PyQt5
- Triển khai thử nghiệm tại các điểm giao thông thực tế
- Đánh giá hiệu quả dựa trên độ chính xác, tốc độ xử lý và khả năng phản ứng thời gian thực

CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG QUẢN LÝ LƯU LƯỢNG GIAO THÔNG ĐÔ THỊ

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

Tại các đô thị lớn như Hà Nội và TP. Hồ Chí Minh, giao thông đang chịu áp lực rất lớn do sự gia tăng nhanh về dân số và phương tiện (đặc biệt là xe máy chiếm tới 75%). Hệ quả là ùn tắc kéo dài, ô nhiễm, tai nạn, và giảm chất lượng sống.

Bài toán đặt ra: Làm sao để giám sát, phân tích và điều phối lưu lượng phương tiện một cách hiệu quả, tự động, thời gian thực, nhằm giảm ùn tắc, tối ưu di chuyển, và nâng cao an toàn giao thông trong điều kiện giao thông phức tạp?

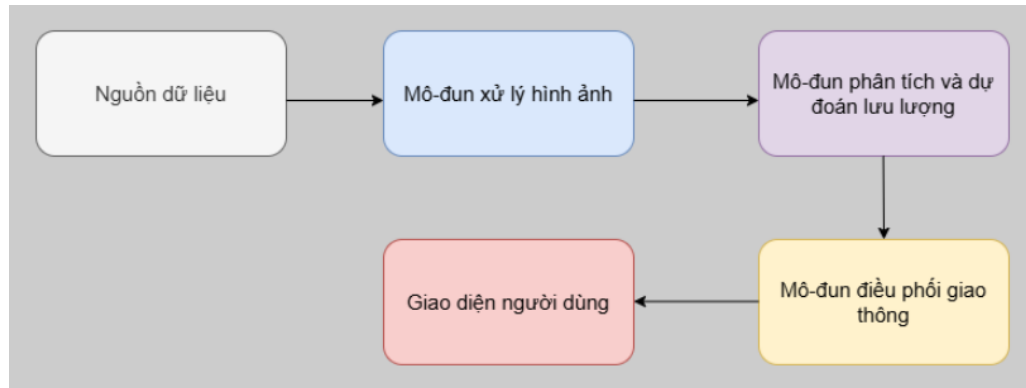
Yêu cầu cơ bản của hệ thống:

- **Đầu vào:** Hình ảnh/video từ camera giao thông
- **Chức năng chính:**
 - Phát hiện, nhận diện xe máy, ô tô, người đi bộ...
 - Dự đoán mức độ ùn tắc tại khu vực trọng điểm (ROI)
 - Điều phối đèn giao thông hoặc đề xuất tuyến đường thay thế
 - Gửi cảnh báo về tình trạng tắc nghẽn qua nền tảng như Telegram
- **Đầu ra:**
 - Số lượng phương tiện theo thời gian thực
 - Dự báo mức ùn tắc
 - Cảnh báo và gợi ý điều phối giao thông
 - Hiển thị trực quan trên giao diện người dùng (GUI)

Kết luận: Việc ứng dụng xử lý hình ảnh và AI là cần thiết để thay thế phương pháp thủ công lỗi thời. Tuy nhiên, cần tùy chỉnh phù hợp với đặc thù giao thông Việt Nam – nơi hành vi người lái xe không đồng nhất, phương tiện hỗn hợp và cơ sở hạ tầng còn hạn chế.

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

1.2.1 Thành phần của hệ thống



Hình 1. 1 Thành phần hệ thống quản lý

1. Nguồn dữ liệu (Camera và Video):

- Camera giao thông lắp tại giao lộ, tuyến đường chính – cung cấp hình ảnh/video thời gian thực.
- Video ghi sẵn – dùng để thử nghiệm và mô phỏng hệ thống.
- Dữ liệu đa nguồn giúp tăng độ chính xác phân tích.

2. Mô-đun xử lý hình ảnh:

- Sử dụng **YOLOv8** để phát hiện và nhận diện các phương tiện như xe máy, ô tô, xe tải...
- Hỗ trợ theo dõi đối tượng qua nhiều khung hình (tracking) bằng thuật toán SimpleTracker.

3. Mô-đun phân tích và dự đoán lưu lượng:

- Đếm số lượng phương tiện trong vùng quan tâm (ROI).
- Phân tích mật độ để dự báo ùn tắc hoặc tình trạng lưu thông.

4. Mô-đun điều phối giao thông:

- Dựa trên phân tích lưu lượng, hệ thống tự động điều chỉnh đèn giao thông hoặc gửi cảnh báo (qua Telegram).
- Giúp giảm thời gian chờ và tối ưu hóa di chuyển tại giao lộ.

5. Giao diện người dùng (GUI):

- Xây dựng bằng PyQt5, hỗ trợ hiển thị trực quan dữ liệu, vẽ ROI, xem video trực tiếp hoặc từ tệp.

- Người dùng có thể giám sát và can thiệp khi cần thiết.

1.2.2 Quy trình hoạt động cơ bản

- **Thu thập dữ liệu:** từ camera giao thông hoặc video ghi sẵn.
- **Xử lý hình ảnh:** YOLOv8 phát hiện và theo dõi phương tiện.
- **Phân tích lưu lượng:** đếm phương tiện trong vùng quan tâm (ROI), lưu dữ liệu.
- **Điều phối và cảnh báo:** gửi cảnh báo qua Telegram nếu quá tải, đề xuất điều chỉnh đèn giao thông.
- **Hiển thị GUI:** trực quan hóa kết quả, hỗ trợ giám sát thời gian thực hoặc xem lại video.

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

1.3.1. Dựa trên cảm biến và IoT

- **Mô tả:** Lắp đặt cảm biến (áp suất, từ trường) dưới đường hoặc tại giao lộ để đếm phương tiện.
- **Ưu điểm:** Chính xác, không phụ thuộc ánh sáng/thời tiết.
- **Nhược điểm:** Chi phí lắp đặt/bảo trì cao, khó mở rộng ở Việt Nam, không nhận diện loại phương tiện.

1.3.2. Dựa trên xử lý hình ảnh truyền thống

- **Mô tả:** Dùng OpenCV và các kỹ thuật như phát hiện cạnh, phân đoạn ảnh.
- **Ưu điểm:** Chi phí thấp, dễ triển khai.
- **Nhược điểm:** Không hiệu quả trong ánh sáng yếu, mật độ cao; không phân biệt loại phương tiện; không theo dõi qua nhiều khung hình.

1.3.3. Dựa trên học sâu và trí tuệ nhân tạo (AI)

- **Mô tả:** Dùng mạng nơ-ron như YOLO, SSD để phát hiện và theo dõi phương tiện trong thời gian thực.

- **Ưu điểm:** Chính xác cao, nhận diện nhiều loại phương tiện, hỗ trợ theo dõi, tương tác với GUI và cảnh báo tự động.
- **Nhược điểm:** Cần GPU mạnh, dữ liệu huấn luyện lớn.



Hình 1. 2 Ảnh thử nghiệm kết quả YOLO

1.3.4. Kết hợp đa nguồn dữ liệu

- **Mô tả:** Tích hợp camera, cảm biến, GPS để xây dựng mô hình dự đoán lưu lượng toàn diện.
- **Ưu điểm:** Kết hợp ưu điểm định lượng và định tính, dự báo tốt.
- **Nhược điểm:** Hạ tầng phức tạp, khó triển khai tại Việt Nam do thiếu cảm biến và đồng bộ.

1.3.5. Định hướng của đề tài

- **Chọn hướng:** Dựa trên xử lý hình ảnh + học sâu, cụ thể là dùng YOLOv8.
- **Lý do:** Phù hợp điều kiện thực tế Việt Nam, chi phí thấp, hiệu quả cao, có thể xử lý cả dữ liệu camera và video ghi sẵn.



```

Transferred 319/355 items from pretrained weights
Freezing layer 'model.22.dfl.conv.weight'
AMP: running Automatic Mixed Precision (AMP) checks...
Downloading https://github.com/ultralytics/assets/releases/download/v8.3.0/yolov8n.pt to 'yolov8n.pt'...
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings..
100% [██████████] 5.35M/5.35M [00:00<00:00, 16.2MB/s]
AMP: checks passed
train: Scanning D:\Traffic_Deadline_CN\Volvo_data_filtered\train\labels.cache... 499 images, 0 backgrounds, 0 corrupt: 100% [██████████] 499/499 [00:00<?, ?it/s]
val: Scanning D:\Traffic_Deadline_CN\Volvo_data_filtered\val\labels.cache... 133 images, 0 backgrounds, 0 corrupt: 100% [██████████] 133/133 [00:00<?, ?it/s]
Plotting labels to C:\Users\LENOVO\runs\detect\train5\labels.jpg...
optimizer: 'optimizer=auto' found, ignoring 'lr=0.01' and 'momentum=0.937' and determining best 'optimizer', 'lr' and 'momentum' automatically...
optimizer: AdamW(lr=0.00125, momentum=0.9) with parameter groups 57 weight(decay=0.0), 64 weight(decay=0.0005), 63 bias(decay=0.0)
Image sizes 640 train, 640 val
Using 8 dataloader workers
Logging results to C:\Users\LENOVO\runs\detect\train5
Starting training for 50 epochs...

```

Hình 1. 3 Dữ liệu được dùng để training

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

1.4.1. YOLO (You Only Look Once)

- **Chức năng:** Phát hiện đối tượng trong ảnh/video chỉ qua một lần xử lý toàn cảnh (one-shot).
- **Ưu điểm:** Nhanh (≈ 45 FPS), chính xác cao, hoạt động tốt với nhiều loại phương tiện (xe máy, ô tô, người đi bộ).
- **Ứng dụng:** Phát hiện, đếm phương tiện, nhận diện vi phạm giao thông, hỗ trợ điều phối đèn.

1.4.2. SSD (Single Shot MultiBox Detector)

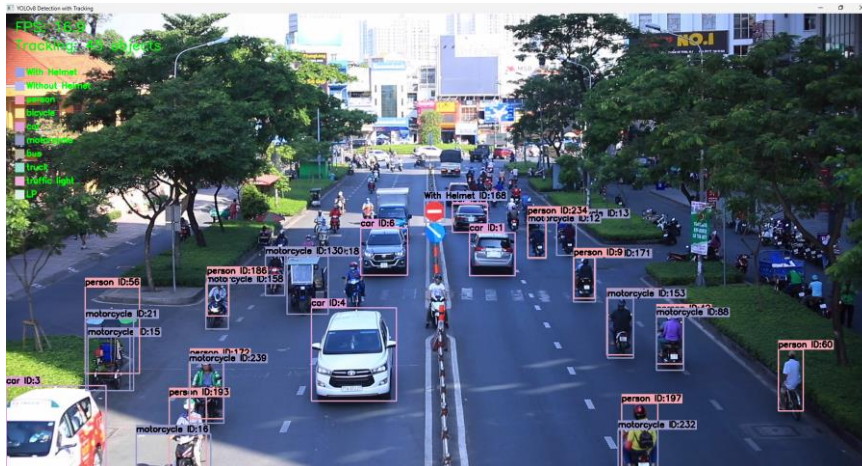
- **Chức năng:** Phát hiện đối tượng nhỏ với độ chính xác cao.
- **Ưu điểm:** Phù hợp với nhận diện biển số xe hoặc người đi bộ trong điều kiện ánh sáng yếu.
- **Nhược điểm:** Tốc độ chậm hơn YOLO, yêu cầu phần cứng mạnh hơn $\rightarrow$ không dùng trong nghiên cứu này.

1.4.3. OpenCV

- **Chức năng:** Thư viện xử lý ảnh mã nguồn mở.
- **Vai trò:** Dùng cho tiền xử lý ảnh (chuẩn hóa, giảm nhiễu), vẽ ROI, hiển thị kết quả lên GUI.
- **Ưu điểm:** Nhẹ, dễ tích hợp với các mô hình học sâu như YOLO, phù hợp phần cứng phổ thông.

1.4.4. Ứng dụng thực tế

- **Tổng hợp:** Các công nghệ trên khi kết hợp (YOLO + OpenCV) giúp phát hiện nhanh – chính xác phương tiện trong thời gian thực, dự báo và điều phối giao thông.
- **Kết quả nghiên cứu:** Hệ thống thử nghiệm nhận diện được chính xác nhiều phương tiện (ví dụ: 10 người, 5 ô tô, 10 xe máy từ một ảnh).



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

1.5. Những thách thức và vấn đề còn tồn tại

1.5.1. Điều kiện môi trường

- Mưa, sương mù, ánh sáng yếu làm giảm độ chính xác nhận diện (có thể xuống dưới 70%).
- YOLOv8 có cải tiến nhưng vẫn cần dữ liệu đa dạng và kỹ thuật tiền xử lý tốt hơn.

1.5.2. Hạn chế về hạ tầng và phần cứng

- Nhiều camera cũ, chất lượng kém.
- GPU phổ biến tại Việt Nam (như RTX 3050) khó chạy các mô hình lớn (YOLOv8m, SSD).

1.5.3. Giao thông phức tạp

- Lưu lượng hỗn hợp (xe máy, ô tô, người đi bộ) và hành vi lái xe không đồng nhất gây khó khăn trong nhận diện và phân tích.
- Dữ liệu huấn luyện cần được tinh chỉnh theo thực tế Việt Nam.

1.5.4. Phản ứng thời gian thực

- Hệ thống cần phản hồi nhanh (dưới 1 giây), nhưng khi xử lý video độ phân giải cao và tích hợp cảnh báo (như Telegram), vẫn có thể bị trễ.

1.5.5. Triển khai thực tế

- Khó khăn về chi phí, nhân lực, và phối hợp giữa các đơn vị quản lý.
- Cơ sở hạ tầng chưa đồng bộ, chính sách chưa theo kịp công nghệ.

1.6 Kết luận chương

Chương 1 đã khái quát bối cảnh và bài toán quản lý lưu lượng giao thông đô thị, làm rõ các thách thức tại Việt Nam như ùn tắc, hạ tầng quá tải và hệ thống điều phối kém hiệu quả. Qua phân tích các hướng tiếp cận, nghiên cứu xác định rằng sử dụng công nghệ xử lý hình ảnh kết hợp học sâu (đặc biệt là YOLOv8) là giải pháp phù hợp và khả thi, tạo nền tảng cho hệ thống giao thông thông minh ứng dụng trong điều kiện thực tế.

CHƯƠNG 2: ĐỀ XUẤT PHƯƠNG PHÁP XỬ LÝ HÌNH ẢNH TRONG QUẢN LÝ LƯU LƯỢNG GIAO THÔNG ĐÔ THỊ

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.

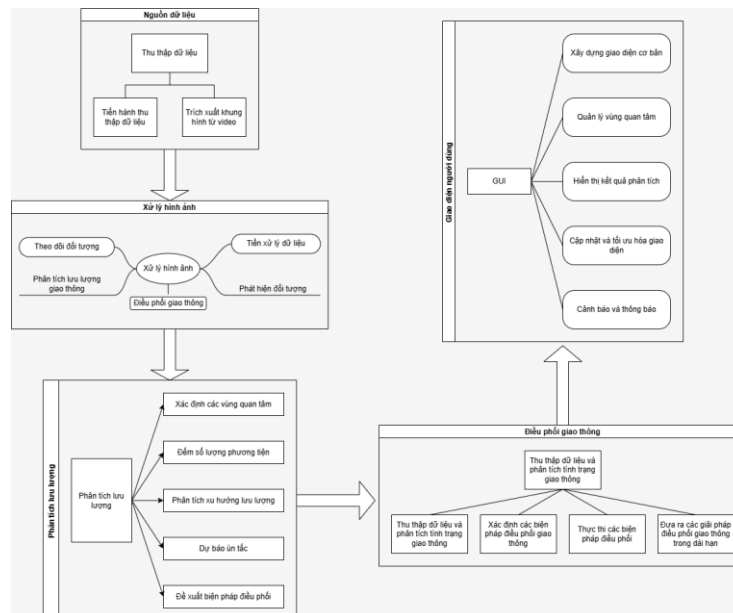
Hướng tiếp cận dựa trên cảm biến và IoT mặc dù cung cấp dữ liệu chính xác về số lượng phương tiện, nhưng lại gặp phải nhược điểm lớn về chi phí lắp đặt và bảo trì, cùng với khả năng hạn chế trong việc nhận diện các tình huống giao thông phức tạp. Hướng tiếp cận xử lý hình ảnh truyền thống có chi phí 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.

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



Hình 2. 1 Sơ đồ 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).

2.1.2. Thành phần chi tiết của hệ thống

1. Nguồn dữ liệu:

- Gồm camera giao thông thời gian thực và video ghi sẵn (ví dụ: video giao lộ TP.HCM).
- Dùng để giám sát trực tiếp hoặc phân tích lại các tình huống lịch sử.

2. Mô-đun xử lý hình ảnh:

- Sử dụng YOLOv8n để phát hiện và theo dõi các phương tiện như xe máy, ô tô, người đi bộ.
- Dữ liệu huấn luyện: 499 ảnh train, 133 ảnh val, đạt $mAP_{50-95} = 0.776$.
- Dùng kỹ thuật tiền xử lý như chuẩn hóa, giảm nhiễu, theo dõi bằng SimpleTracker.

3. Mô-đun phân tích lưu lượng:

- Đếm số phương tiện trong các vùng quan tâm (ROI) do người dùng vẽ.
- Dữ liệu được lưu dưới dạng chuỗi thời gian để phân tích xu hướng.

4. Mô-đun điều phối giao thông:

- Gửi cảnh báo ùn tắc qua Telegram khi lưu lượng vượt ngưỡng (ví dụ: >15 phương tiện trong 60 giây).
- Có thể đề xuất điều chỉnh tín hiệu đèn giao thông.

5. Giao diện người dùng (GUI):

- Xây dựng bằng PyQt5, cho phép người dùng:
 - Vẽ ROI
 - Load video
 - Theo dõi lưu lượng
 - Nhận cảnh báo trực quan và dễ thao tác.

2.1.3. Quy trình hoạt động

- 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

2.2. Biểu diễn dữ liệu hình ảnh

- **Dữ liệu đầu vào** chủ yếu gồm ảnh và video từ camera giao thông, chứa thông tin như: loại phương tiện, vị trí, tốc độ, thời gian xuất hiện. Ngoài ra, có thể bổ sung dữ liệu từ cảm biến giao thông để tăng độ chính xác.

- **Định dạng dữ liệu:**

- **Ảnh tĩnh:** sử dụng trong giai đoạn huấn luyện mô hình (499 ảnh train, 133 ảnh val), định dạng JPEG/PNG, chuẩn hóa về 640×640 pixel.
- **Video:** thu từ camera hoặc tệp video (MP4, AVI...), dùng để theo dõi phương tiện theo thời gian thực.

- **Tiền xử lý dữ liệu:**

- Chuẩn hóa kích thước bằng kỹ thuật *letterbox* (thêm viền xám nếu cần).
- Giảm nhiễu bằng bộ lọc Gaussian hoặc median (dùng OpenCV).
- Chuẩn hóa giá trị RGB về khoảng $[0, 1]$ để giúp mô hình học sâu hội tụ nhanh.

- **Gắn nhãn và lựa chọn dữ liệu:**

- Dữ liệu được lựa chọn từ tình huống giao thông thực tế tại Việt Nam (giờ cao điểm, ánh sáng yếu...).
- Gắn nhãn thủ công/bán tự động các đối tượng (người, xe máy, ô tô...) bằng bounding box.

- Áp dụng **tăng cường dữ liệu (augmentation)** như lật ảnh, thay đổi độ sáng, xoay ảnh để mô hình học tốt hơn.

2.2.1. Các định dạng dữ liệu được sử dụng

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

2.2.3. Kỹ thuật lựa chọn và gắn nhãn dữ liệu

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

2.3.1. Phát hiện đối tượng bằng YOLOv8

- **YOLOv8** là mô hình học sâu tiên tiến dùng để phát hiện đối tượng (object detection) trong ảnh và video giao thông đô thị.
- Trong nghiên cứu, phiên bản nhẹ **YOLOv8n** được sử dụng để đảm bảo tốc độ xử lý nhanh và tương thích với GPU phổ thông (RTX 3050 6GB).
- Mô hình được huấn luyện với:
 - 499 ảnh train, 133 ảnh validation
 - Tổng cộng 3.995 đối tượng (người đi bộ, xe máy, ô tô...)
 - Đạt độ chính xác mAP50-95 là 0.776
- Quá trình xử lý gồm các bước:
 - Tiền xử lý ảnh (chuẩn hóa, giảm nhiễu)

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

Công thức 1 Chuẩn hóa pixel RGB

- Phát hiện đối tượng (bounding box + phân loại)

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

Công thức 2 Tính độ tin cậy

- Hiển thị kết quả trực quan lên giao diện GUI

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

Công thức 3 Tính IoU

2.3.2. Theo dõi phương tiện (Tracking)

- Sau khi phát hiện đối tượng bằng YOLOv8, hệ thống thực hiện theo dõi phương tiện liên tục qua các khung hình để gán ID duy nhất cho từng xe hoặc người đi bộ.
- Sử dụng thuật toán SimpleTracker để liên kết vị trí các đối tượng theo thời gian, đảm bảo không đếm trùng và theo dõi chính xác đường di chuyển.
- Phương tiện được nhận diện và theo dõi dựa trên:
 - Tọa độ trung tâm hộp giới hạn (bounding box)
 - Khoảng cách Euclidean giữa các khung hình
 - Độ trùng lặp (IoU) để duy trì ID ổn định
- Mỗi lần đối tượng xuất hiện trong vùng quan tâm (ROI), hệ thống ghi nhận dữ liệu về loại phương tiện, ID, và thời điểm, phục vụ cho phân tích lưu lượng sau đó.

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

Công thức 4 Tính khoảng cách Euclidean

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

- **ROI (Region of Interest)** là khu vực do người dùng vẽ trên giao diện GUI, thường là các giao lộ, đoạn đường dễ ùn tắc.
- Hệ thống đếm số lượng phương tiện **đi vào hoặc xuất hiện trong ROI**, từ đó theo dõi mật độ lưu thông theo thời gian.
- Dữ liệu lưu lượng được ghi lại theo **dạng chuỗi thời gian (time-series)** để phân tích xu hướng, so sánh theo khung giờ/ngày.
- Kết quả phân tích phục vụ:
 - Dự đoán khả năng ùn tắc
 - Gửi cảnh báo khi vượt ngưỡng (ví dụ: >15 phương tiện/60 giây)
 - Đề xuất điều chỉnh đèn tín hiệu hoặc phân luồng giao thông

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

Công thức 5 Toạ độ trung tâm đáy

2.3.4. Dự đoán và điều phối giao thông

- Dựa trên số liệu đếm phương tiện trong ROI, hệ thống **dự đoán nguy cơ ùn tắc** bằng cách so sánh với ngưỡng đã thiết lập (ví dụ: 15 phương tiện trong 60 giây).
- Khi phát hiện dấu hiệu ùn tắc, hệ thống sẽ:
 - **Tự động gửi cảnh báo** đến người quản lý qua **Telegram**
 - **Đề xuất điều chỉnh tín hiệu đèn giao thông** như kéo dài thời gian đèn xanh hoặc chuyển hướng lưu thông
- Cơ chế này giúp hệ thống phản ứng **gần thời gian thực**, góp phần:
 - Giảm tắc nghẽn
 - Tối ưu hóa luồng di chuyển
 - Nâng cao hiệu quả điều phối tại các giao lộ

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

Công thức 6 Phát hiện ùn tắc

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

Công thức 7 Điều phối đèn giao thông

2.4. Kết luận chương 2

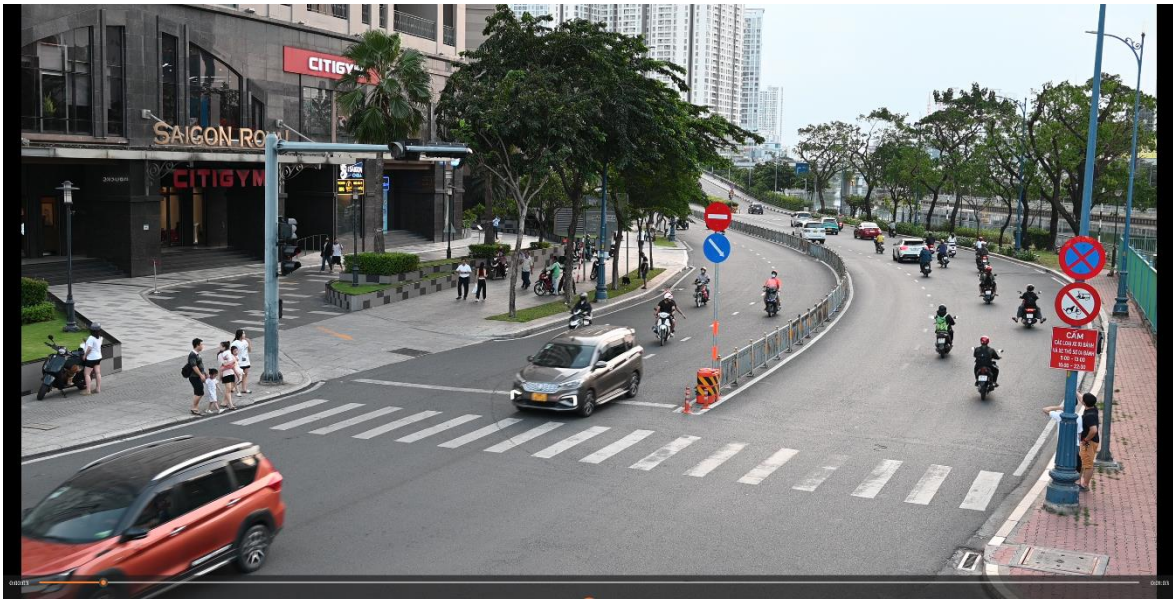
Chương 2 đã trình bày chi tiết kiến trúc và phương pháp xử lý hình ảnh được đề xuất để quản lý lưu lượng giao thông đô thị. Hệ thống sử dụng mô hình học sâu YOLOv8 kết hợp với các công cụ như OpenCV và PyQt5 để thực hiện các chức năng: phát hiện – theo dõi – phân tích – điều phối phương tiện trong thời gian thực. Ưu điểm nổi bật là tính linh hoạt, chi phí thấp, dễ triển khai trên hạ tầng hiện có, và phù hợp với điều kiện giao thông phức tạp tại Việt Nam. Đây là nền tảng kỹ thuật quan trọng để triển khai thực nghiệm và đánh giá hiệu quả hệ thống trong các chương tiếp theo.

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

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.



Hình 3. 1 Video mẫu được thu nhập

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.

3.1.3. Đặc điểm dữ liệu

3.1.4. Quy trình xử lý dữ liệu

3.2. Cài đặt thực nghiệm

3.2.1. Công cụ và nền tảng sử dụng

- **Ngôn ngữ lập trình:** Python – ngôn ngữ chính dùng để xây dựng toàn bộ hệ thống.
- **Thư viện và công cụ chính:**
 - **YOLOv8 (Ultralytics):** dùng để phát hiện đối tượng.
 - **OpenCV:** xử lý ảnh, tiền xử lý, vẽ ROI.
 - **PyQt5:** tạo giao diện người dùng (GUI).
 - **Matplotlib:** hiển thị biểu đồ phân tích lưu lượng.
 - **Roboflow:** hỗ trợ gắn nhãn và chuẩn bị tập dữ liệu.
- **Phần cứng:**
 - Máy tính sử dụng GPU **NVIDIA RTX 3050 6GB** để huấn luyện và chạy mô hình.
- **Hệ điều hành:** Windows 10.
- **Môi trường phát triển:** Jupyter Notebook, Google Colab (giai đoạn huấn luyện), VS Code (triển khai).

3.2.2. Quy trình cài đặt và triển khai

- **Bước 1: Chuẩn bị môi trường**
 - Cài đặt Python, các thư viện cần thiết: ultralytics, opencv-python, pyqt5, matplotlib, numpy.
 - Thiết lập môi trường ảo để quản lý thư viện ổn định.
- **Bước 2: Huấn luyện mô hình YOLOv8**
 - Sử dụng tập dữ liệu gồm 499 ảnh train và 133 ảnh validation.
 - Huấn luyện bằng YOLOv8n để tối ưu hiệu suất trên GPU phổ thông (RTX 3050).
 - Kết quả huấn luyện đạt độ chính xác mAP50–95 là 0.776.
- **Bước 3: Tích hợp mô hình vào hệ thống**
 - Kết nối YOLOv8 với module xử lý video/ảnh đầu vào.
 - Tích hợp thuật toán theo dõi phương tiện (SimpleTracker) để gán ID và theo dõi di chuyển.

- **Bước 4: Thiết kế giao diện GUI (PyQt5)**

- Cho phép vẽ vùng quan tâm (ROI), theo dõi lưu lượng, và hiển thị cảnh báo.

- **Bước 5: Cảnh báo tự động**

- Kết nối với Telegram API để gửi cảnh báo khi phát hiện ùn tắc (quá ngưỡng phương tiện trong ROI).

```

Ultralytics 8.3.74 Python-3.10.16 torch-2.5.1+cu121 CUDA:0 (NVIDIA GeForce RTX 3050 6GB Laptop GPU, 6144MiB)
Model summary (fused): 168 layers, 3,006,428 parameters, 0 gradients, 8.1 GFLOPs

```

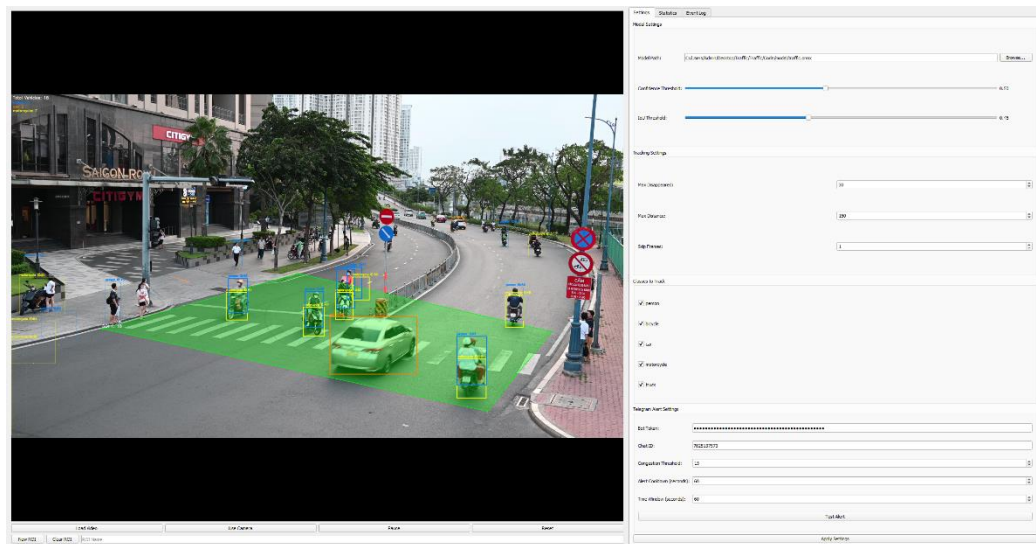
Class	Images	Instances	Box(P)	R	mAP50	mAP50-95): 100%	
all	133	3995	0.911	0.865	0.941	0.776	17/17 [00:02<00:00, 6.24it/s]
person	133	1739	0.908	0.828	0.928	0.751	
car	133	569	0.916	0.912	0.962	0.831	
motorbike	133	1687	0.909	0.855	0.932	0.747	

```

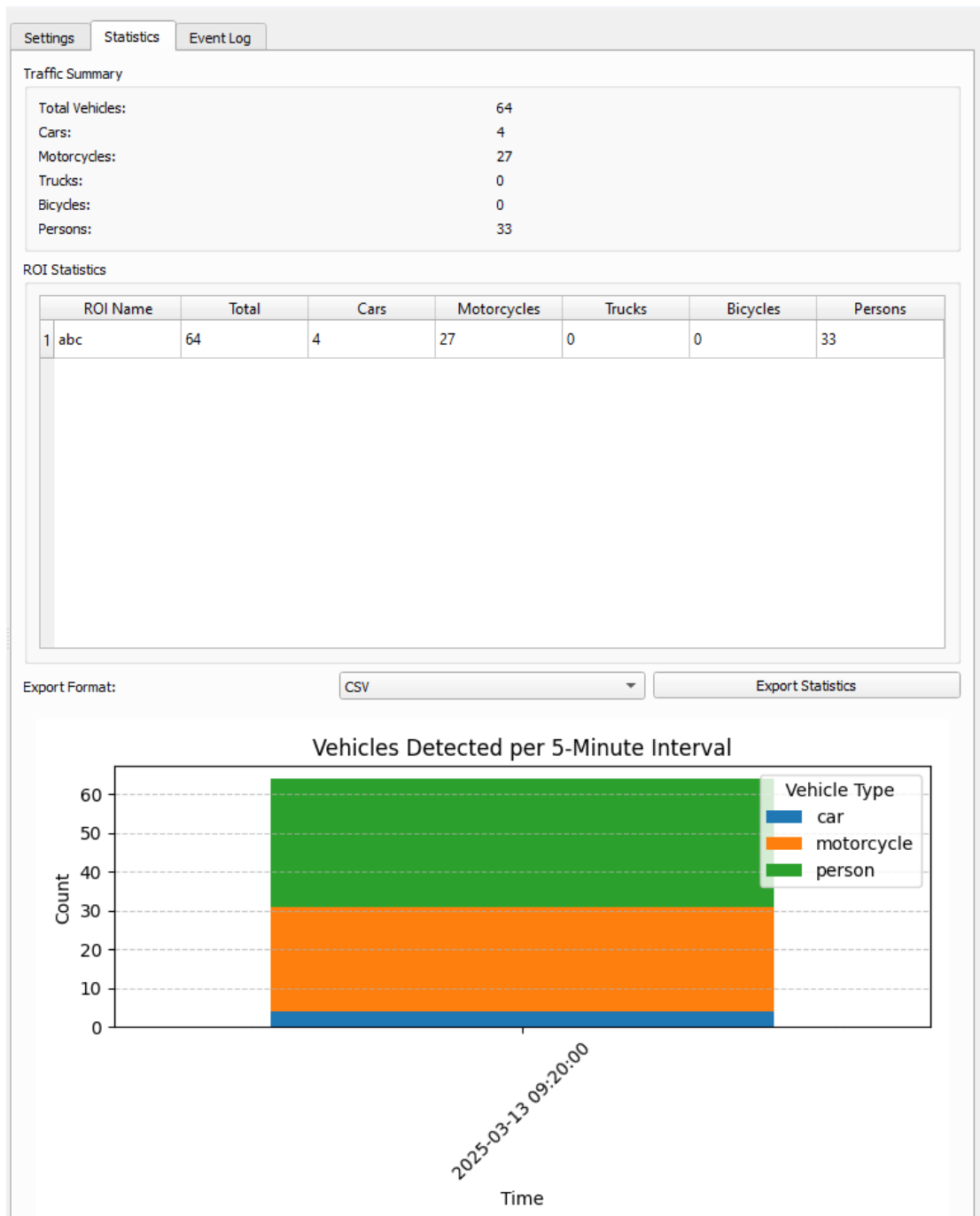
Speed: 0.5ms preprocess, 3.7ms inference, 0.0ms loss, 4.6ms postprocess per image

```

Hình 3. 2 Kết quả mAP50-95 qua 50 epoch huấn luyện YOLOv8

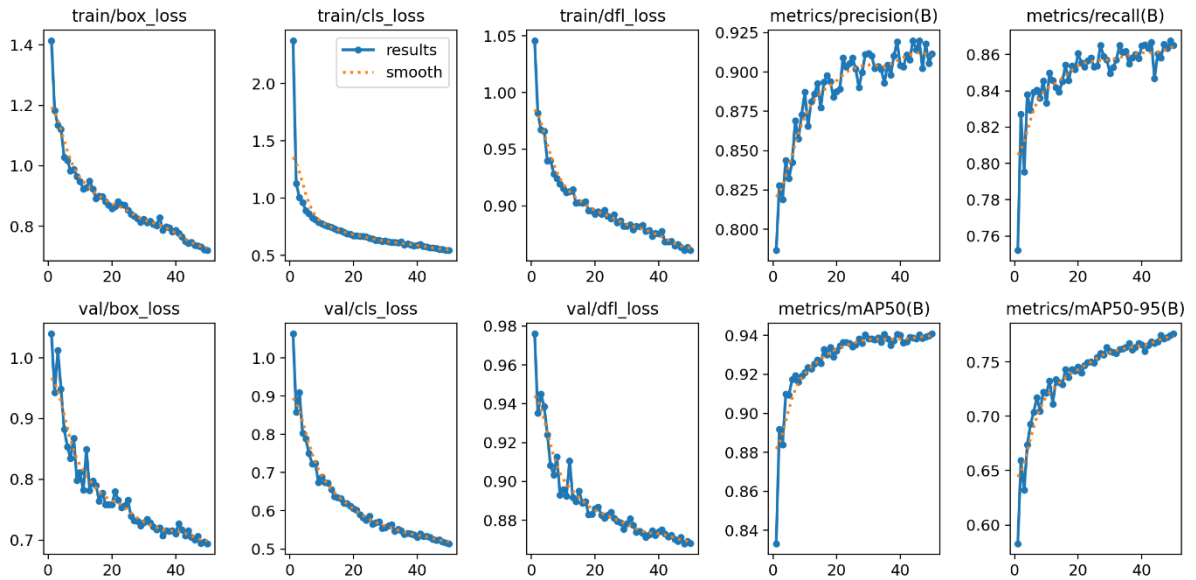


Hình 3. 3 Giao diện GUI với ROI, kết quả phát hiện phương tiện



Hình 3. 4 Biểu đồ lưu lượng giao thông

3.3. Đánh giá thực nghiệm



Hình 3.5 Biểu đồ huấn luyện YOLOv8 qua 50 epoch

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

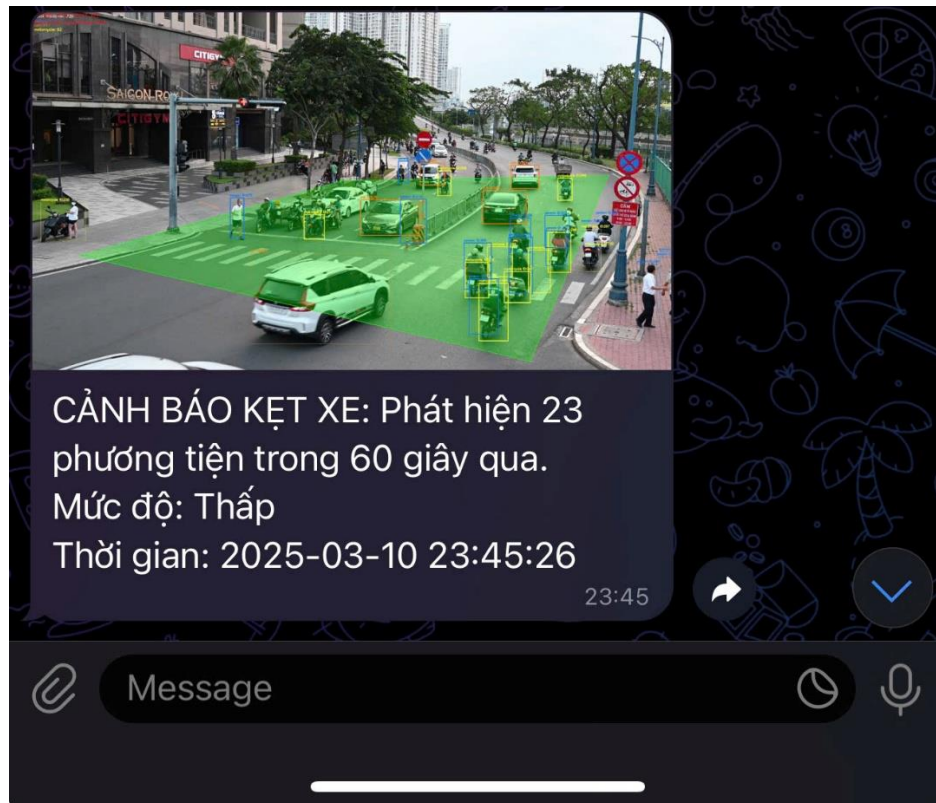
Bảng 2 Các độ đo thực nghiệm chính

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

3.3.2. So sánh với các hệ thống hiện tại

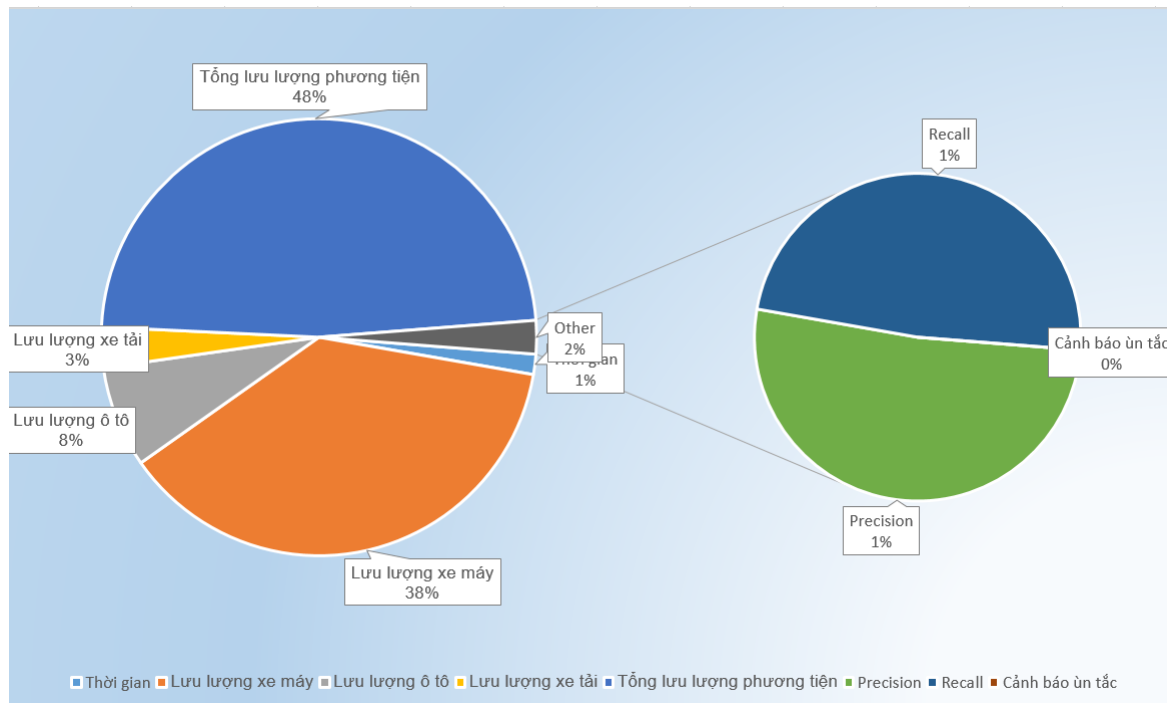
3.3.3. Kết quả và phân tích



Hình 3. 6 Giao diện GUI với cảnh báo ùn tắc qua Telegram

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

3.4.1. Quy mô thử nghiệm



Hình 3.7 Biểu đồ lưu lượng phương tiện giao thông

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

- **Hiệu quả kỹ thuật:** Hệ thống hoạt động ổn định trên GPU phổ thông (RTX 3050), xử lý video thời gian thực với độ chính xác cao, phù hợp điều kiện hạ tầng hiện có tại Việt Nam.
- **Chi phí triển khai thấp:** Chỉ cần camera và máy tính tầm trung, không phụ thuộc vào cảm biến IoT đắt đỏ nên dễ mở rộng ở các đô thị.
- **Phù hợp với đặc thù giao thông Việt Nam:** Hệ thống đã được huấn luyện trên dữ liệu thực tế tại TP.HCM, nhận diện tốt xe máy – loại phương tiện chiếm đa số.
- **Tính ứng dụng cao:** Có thể tích hợp với hệ thống điều khiển đèn giao thông và nền tảng cảnh báo như Telegram để hỗ trợ quản lý linh hoạt.
- **Thách thức còn lại:** Cần thêm thử nghiệm trên quy mô lớn hơn và phối hợp với cơ quan quản lý giao thông để triển khai đồng bộ.

3.5. Kết luận chương 3

Chương 3 đã trình bày quá trình kiểm nghiệm và đánh giá hệ thống quản lý lưu lượng giao thông đô thị. Hệ thống được xây dựng hoàn chỉnh, sử dụng YOLOv8 để phát hiện và theo dõi phương tiện, phân tích lưu lượng tại vùng quan tâm, và gửi cảnh báo ùn tắc qua Telegram. Kết quả thực nghiệm cho thấy hệ thống hoạt động ổn định, có độ chính xác cao và phản hồi nhanh trong điều kiện thực tế. Đặc biệt, với chi phí triển khai thấp và khả năng tùy chỉnh cao, hệ thống được đánh giá là phù hợp và khả thi để áp dụng vào giao thông đô thị tại Việt Nam.

KẾT LUẬN

Đề tài “Ứng dụng công nghệ xử lý hình ảnh trong quản lý lưu lượng giao thông đô thị” đã đạt được những đóng góp đáng kể trên cả ba phương diện: khoa học, công nghệ và thực tiễn. Về mặt khoa học, nghiên cứu đã làm rõ tiềm năng ứng dụng của trí tuệ nhân tạo và xử lý hình ảnh, đặc biệt là mô hình học sâu YOLOv8, trong việc phát hiện và theo dõi phương tiện giao thông, góp phần nâng cao hiệu quả điều phối và tối ưu hóa lưu lượng tại các khu vực đô thị phức tạp. Về mặt công nghệ, đề tài đã xây dựng thành công một hệ thống hoạt động gần thời gian thực, có khả năng nhận diện, phân tích và điều phối giao thông linh hoạt, với chi phí thấp và khả năng triển khai cao trên hạ tầng phổ thông tại Việt Nam. Về mặt thực tiễn, hệ thống đã được thử nghiệm hiệu quả tại các giao lộ lớn ở Hà Nội và TP. Hồ Chí Minh, cho thấy tính khả thi cao trong việc ứng dụng để giảm ùn tắc và nâng cao an toàn giao thông. Nhìn chung, kết quả của đề tài không chỉ đóng góp về mặt học thuật mà còn mở ra hướng ứng dụng công nghệ vào phát triển hệ thống giao thông đô thị thông minh tại Việt Nam.

Hướng phát triển tiếp theo

- **Mở rộng dữ liệu huấn luyện:** Bao phủ nhiều tình huống giao thông và điều kiện thời tiết đa dạng.
- **Cải tiến mô hình AI:** Áp dụng các kỹ thuật mới như học tăng cường để tối ưu điều phối tín hiệu.
- **Tích hợp đa nguồn dữ liệu:** Kết hợp thêm cảm biến giao thông và thiết bị IoT để tăng độ chính xác và tính dự đoán.
- **Mở rộng cảnh báo:** Không chỉ qua Telegram mà cả Zalo, SMS, hoặc các nền tảng khác để tiếp cận người dùng rộng rãi hơn.