

THIẾT KẾ HỆ THỐNG CẢNH BÁO TẮC NGHẼN TRONG GIAO THÔNG ĐÔ THỊ SỬ DỤNG MÔ HÌNH HỌC SÂU

**Nguyễn Thành Lợi¹, Nguyễn Mạnh Hùng², Đỗ Phương Nhung²,
Đào Xuân Phúc², Đinh Khắc Long²
Email: thanhloi2001ym@gmail.com**

Ngày tòa soạn nhận được bài báo: 04/12/2024

Ngày phản biện đánh giá: 13/06/2025

Ngày bài báo được duyệt đăng: 25/06/2025

DOI: 10.59266/houjs.2025.585

Tóm tắt: Bài báo đề xuất thiết kế một hệ thống cảnh báo tắc nghẽn giao thông trong đô thị, nhằm cảnh báo trực tiếp tới các tài xế. Phương pháp bao gồm việc phát hiện các phương tiện bằng model YOLOv8 và theo dõi chúng bằng model deepSORT để đánh ID cho từng đối tượng. Từ đó, tính toán mật độ lưu lượng xe và vận tốc di chuyển trung bình. Hệ thống sử dụng các thông số này và kết quả thực nghiệm để xác định ngưỡng cảnh báo, sau đó thông báo tình trạng giao thông qua ứng dụng chat Telegram. Hệ thống đạt độ chính xác cao và có thể mở rộng để cung cấp thông tin giao thông chi tiết và điều tiết tín hiệu đèn giao thông.

Từ khóa: YOLOv8, deepSORT, Telegram, opencv, cảnh báo tắc nghẽn giao thông

I. Đặt vấn đề

Giao thông đô thị ngày càng phức tạp và tắc nghẽn là vấn đề đáng lo ngại, gây mất thời gian, tiêu tốn năng lượng, ảnh hưởng môi trường và kinh tế. Phát triển hệ thống cảnh báo tắc nghẽn thông minh là cần thiết để giúp người dân và cơ quan quản lý đưa ra quyết định tốt hơn về tuyến đường. Nghiên cứu này thiết kế hệ thống cảnh báo tắc nghẽn dựa trên model YOLOv8 và DeepSORT để phát hiện và theo dõi phương tiện. Model YOLOv8 giúp phát hiện nhanh và chính xác phương tiện, còn

DeepSORT giúp theo dõi vị trí phương tiện theo thời gian thực.

II. Các công trình liên quan

Hiện nay, các chủ đề cho việc xử lý các vấn đề về giao thông như cảnh báo tắc nghẽn... không phải là mới, dưới đây là một số mô hình giải quyết bài toán được trích dẫn từ các công trình nghiên cứu, bài báo khoa học mà được công bố trên IEEE.

Visual Traffic Jam Analysis Based on Trajectory Data (Wang, 2013). Bài báo đưa ra một hệ thống tương tác để phân

¹ Trường Điện - Điện tử, Đại học Bách khoa Hà Nội

² Trường Đại học Mở Hà Nội

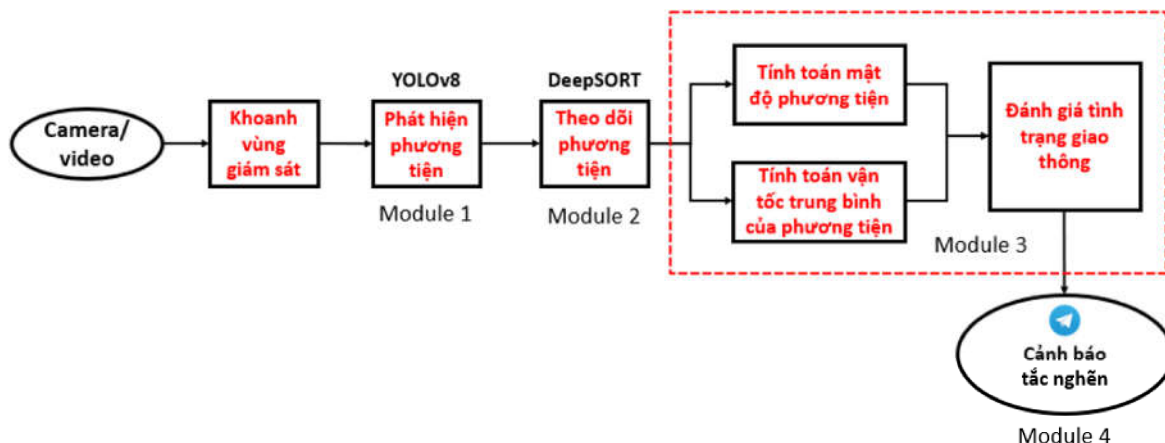
tích hình ảnh tắc nghẽn giao thông đô thị dựa trên các quỹ đạo GPS.

Automatic traffic diversion system using traffic signals (Sandhya, 2017). Bài báo xây dựng một kế hoạch chuyển hướng giao thông tự động trong đó nghiên cứu đến các phương pháp xử lý traffic signal lân cận rất linh hoạt và thông minh.

Traffic Surveillance System for Vehicle Flow Detection (Pan, 2010). Trong bài báo đề xuất một hệ thống giám sát giao thông để phát hiện và đếm phương tiện. Đầu tiên trích xuất ROI (vùng quan tâm) từ video giao thông bằng cách kết hợp phép trừ nền và phát hiện biên. Tiếp theo, sau khi phát hiện giao thông, sử dụng một phương pháp chia làn tự động để có được một số làn đường. Cuối cùng, dựa trên độ rộng của làn đường, thiết lập một số cửa sổ tự điều chỉnh cho việc đếm phương tiện thay vì phương pháp cửa sổ cố định truyền thống

III. Phương pháp áp dụng

3.1. Tổng quan hệ thống



Hình 1. Mô hình tổng quan

Hình 1 thể hiện mô hình tổng quan của từng khối trong hệ thống, đầu vào là dữ liệu được lấy từ camera theo thời gian thực hoặc video có sẵn đưa qua model YOLOv8, thuật toán giúp phát hiện các phương tiện, bao gồm có 5 loại chính là

Traffic Jam Detection Based on Corner Feature of Background Scene In Video-Based ITS (Lei, 2008). Một phương pháp để trích xuất sự kiện tắc nghẽn bằng cách phân tích đặc điểm nền đường đã được phát triển trong nghiên cứu này, trong đó các hình ảnh nền đa nền đã được sử dụng để trích xuất nền cuối cùng từ các phương tiện di chuyển chậm trong tình trạng tắc nghẽn. Thuật toán đề xuất bao gồm hai giai đoạn: trích xuất nền và phát hiện tắc nghẽn. Hình ảnh nền được trích xuất bằng cách thực hiện sự khác biệt trên ba khung hình liên tiếp. Dựa vào hình ảnh nền đã trích xuất, phân tích đặc điểm góc để phát hiện tắc nghẽn giao thông.

Các phương pháp độ chính xác còn thấp và chưa đảm bảo được khả năng xử lý trong thời gian thực vì vậy bài báo đề xuất sử dụng YOLOv8 và DeepSORT để giải quyết bài toán đầu ra cho mô hình

xe đạp, xe máy, ô tô, xe buýt, xe tải. Sau khi huấn luyện mô hình với bộ dữ liệu đã thu thập, kết quả cho ra độ chính xác khá cao. Ngoài ra, sử dụng thêm các file trọng số của YOLOv8 được huấn luyện từ tập dữ liệu của coco như file yolov8n.

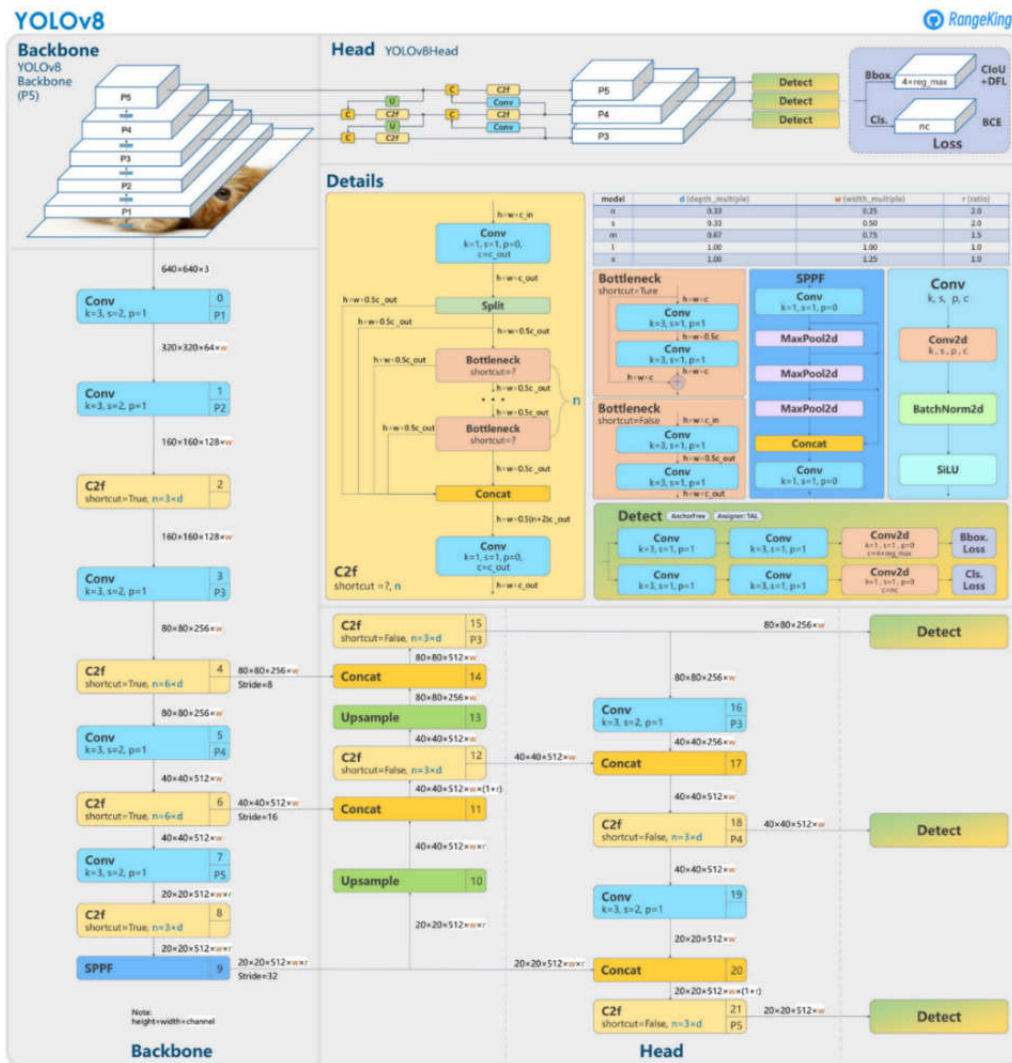
pt, yolov8s.pt, yolov8l.pt. Sau khi phát hiện và phân loại được các phương tiện, hệ thống sẽ sử dụng thuật toán Deepsort để theo dõi các phương tiện tham gia giao thông, đồng thời đánh nhãn ID cho từng đối tượng. Tiếp theo chúng tôi sẽ dựa vào mô hình phát hiện và theo dõi đối tượng để tính toán diện tích các vật thể là các phương tiện dựa trên giới hạn hình hộp chúng ngoài ra chúng tôi cũng xử lý phần khu vực khoanh vùng cho trước từ đó có thể tính được diện tích khu vực cần quan tâm cũng như tính toán được mật độ các phương tiện trên khu vực khoanh vùng đó. Tiếp theo để phân biệt các tình huống giao thông trong thực tế chúng tôi cũng xác

định ra vận tốc cho từng đối tượng và vận tốc trung bình của tất cả phương tiện giao thông trên khu vực đó. Dựa vào các hệ số được tính toán ở trên cũng như các kết quả mà chúng tôi thực nghiệm áp dụng hệ thống trong thực tế, chúng tôi sẽ xác định ra ngưỡng phù hợp để đưa ra đầu ra cho hệ thống là thông báo đến người dùng về tình trạng giao thông tại khu vực khoanh vùng bằng ứng dụng chat telegram.

3.2. Chi tiết hệ thống

3.2.1. YOLOv8

Khởi thứ nhất sẽ thực hiện bài toán classification các object là các phương tiện giao thông.



Hình 2. Kiến trúc YOLOv8 (Pereira, 2022)

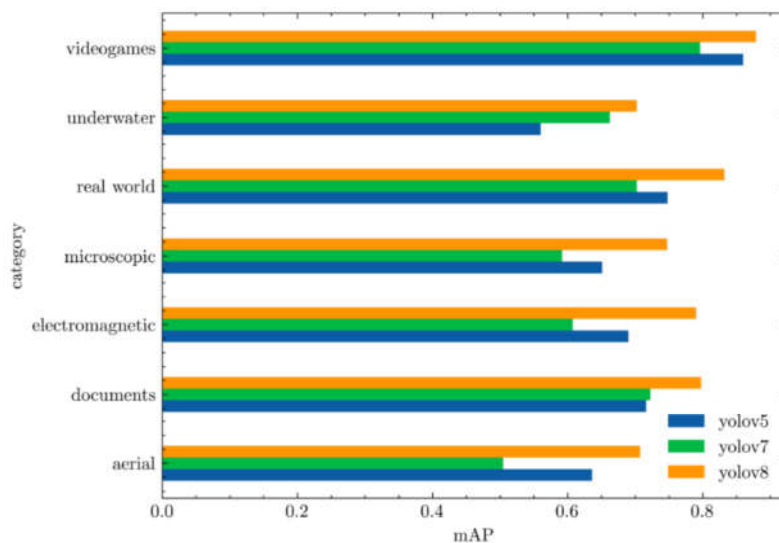
Anchor Free Detection

YOLOv8 là một mô hình không sử dụng anchor box, trái ngược với YOLOv3 truyền thống. Thay vì dựa vào anchor boxes để dự đoán vị trí đối tượng, YOLOv8 dự đoán trực tiếp tâm của đối tượng mà không cần thông tin về khoảng cách tới anchor boxes. Điều này giải quyết khó khăn của các phiên bản trước, khi anchor boxes không phản ánh chính xác phân bố của hình chữ nhật trong tập dữ liệu tùy chỉnh. Anchor-free detection giảm số lượng dự đoán hình chữ nhật, tăng tốc quá trình Non-Maximum Suppression (NMS), một bước xử lý sau suy luận để lọc các dự đoán trùng lặp. Với ít dự đoán hơn, NMS hoạt động nhanh và hiệu quả hơn, giữ lại các đối tượng duy nhất trong kết quả cuối cùng.

Độ chính xác của YOLOv8 trên tập dữ liệu kiểm tra COCO

Nghiên cứu YOLOv8 chủ yếu dựa trên đánh giá thực nghiệm trên tập dữ liệu COCO. Các thay đổi trong mạng và quá trình huấn luyện được xác minh bằng thí nghiệm trên COCO, một tập dữ liệu đa dạng và đại diện cho các tình huống thực tế. Cải thiện độ chính xác trên COCO là thước đo quan trọng của hiệu suất YOLOv8. Biểu đồ cột dưới đây cho thấy mAP@.50 cho mỗi danh mục trong benchmark RF100, với YOLOv8 vượt trội hơn các mô hình trước đó.

Hình 3 đã thể hiện đánh giá độ chính xác của các phiên bản YOLO trên các bộ dữ liệu khác nhau. Kết quả cho thấy YOLOv8 có độ chính xác tốt hơn các phiên bản khác vì vậy nhóm nghiên cứu quyết định sử dụng YOLOv8 làm mô hình phát hiện và nhận diện đối tượng.



Hình 3. Độ chính xác trung bình của YOLOv8 (mAP@.50) đối với các danh mục trong RF100

3.2.2. DeepSORT

➤ Trong đó thuật toán SORT (simple online real-time tracking) bao gồm bốn thành phần cốt lõi là:

- Phát hiện đối tượng: xác định vị trí và đặc điểm của các đối tượng trong khung hình.

- Đánh giá và dự đoán: ước lượng vị trí và tình trạng của các đối tượng trong các khung hình tiếp theo.

- Liên kết object: tìm ra mối quan hệ giữa của các object trong nhiều khung hình khác nhau.

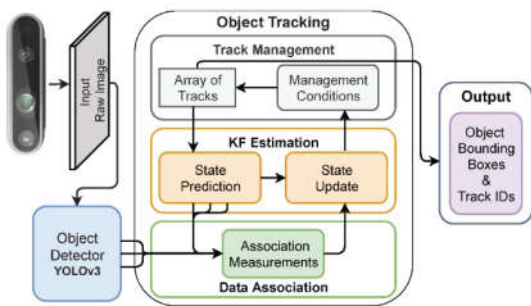
- Theo dõi và quản lý danh tính của đối tượng: theo dõi và cập nhật thông tin về các đối tượng theo thời gian.

➤ Tuy nhiên, thuật toán SORT cũng đối mặt với một số hạn chế::

- Giả định tuyến tính: sử dụng Linear Kalman Filter có thể không phù hợp trong một số tình huống thực tế.

- ID Switches: dựa vào độ đo IOU (Intersection over Union) gây chuyển đổi ID không mong muốn khi có che khuất hoặc quỹ đạo trùng lặp.

DeepSORT nâng cấp SORT bằng cách sử dụng thông tin xuất hiện của object và các đặc trưng hình dáng từ mạng CNN để tăng cường liên kết và xác định lại các track.



Hình 4. Tổng quan về thuật toán DeepSORT (Faisal, 2023)

Trong Deep-SORT, quá trình liên kết hộp giới hạn và track sử dụng thuật toán Hungary. Phần đầu tiên dùng chỉ số chuyển động và ngoại hình để liên kết track. Phần sau kết nối dữ liệu tương tự và tính khoảng cách Mahalanobis giữa dự đoán và detect của đối tượng. Sử dụng model CNN để trích xuất đặc điểm ngoại hình và tính toán khoảng cách cosine. DeepSORT cải thiện đáng kể so với SORT, giảm false positive và false negative trong quá trình theo dõi.

3.2.3. Các thuật toán triển khai sau khi detect và tracking đối tượng

Sau khi hệ thống đưa qua 2 model đầu tiên để detect và tracking được các phương

tiện giao thông thì bước tiếp theo để giải quyết đầu ra của bài toán là cảnh báo tắc nghẽn giao thông trong đô thị chúng ta cần quan tâm tới 4 thông số chính:

➤ Diện tích khu vực khoanh vùng.

➤ Tổng diện tích các phương tiện được detect và theo dõi trong khu vực khoanh vùng.

➤ Mật độ phương tiện giao thông được xác định dựa trên tỉ lệ của 2 thông số ban đầu.

➤ Trung bình của vận tốc của tất cả các phương tiện trong khu vực khoanh vùng đó.

Để xác định ra khu vực khoanh vùng cũng như diện tích của khu vực khoanh vùng cho trước ta thiết lập hàm bắt sự kiện click chuột trái trong đoạn mã code, hàm này cho phép ta lựa chọn các điểm cho trước rồi sau đó thiết lập thêm hàm để vẽ vùng bao quanh bởi các điểm vừa chọn lúc này ta đã có thể thiết lập được một khu vực khoanh vùng và từ đó ta cũng có thể tính được diện tích của khu vực trên dựa vào hàm thư viện trong opencv.

Để tính tổng diện tích các phương tiện được detect và tracking trong khu vực khoanh vùng thì ta sẽ dựa vào bounding box của từng phương tiện được detect và tracking đó, từ đó dựa trên giá trị chiều dài và chiều rộng của mỗi bounding box ứng với từng đối tượng ta sẽ xác định ra được diện tích hình chữ nhật của các bounding box đó, tiếp tục cộng tất cả diện tích của các phương tiện đó lại sẽ xác định được ra tổng diện tích các phương tiện có trong khu vực mà ta cần quan tâm.

Từ hai thông số diện tích khu vực khoanh vùng và tổng diện tích các phương tiện có trong khu vực khoanh vùng ta sẽ tính ra được mật độ lưu lượng

các phương tiện giao thông trên khu vực đó theo công thức:

$$\text{traffic density} = \frac{\text{total area of vehicles}}{\text{road area}}$$

Tiếp theo để hệ thống có thể phân biệt được giao thông đang tắc nghẽn hay vẫn lưu thông ta cần xác định được giá trị vận tốc trung bình của các phương tiện di chuyển trong khu vực khoanh vùng. Để làm được điều này ta sẽ sử dụng thuật toán dựa trên công thức tính vận tốc cơ bản:

$$v = \frac{s}{t}$$

Trong đó:

s là quãng đường di chuyển

t là thời gian di chuyển

v là vận tốc phương tiện di chuyển trong khoảng thời gian đó

Tiếp theo hệ thống sẽ thực hiện xác định quãng đường s theo công thức ta dựa vào khoảng cách Euclid từ vị trí của vật giữa 2 khung hình (frame) liên tiếp. Gọi điểm A là tâm của bounding box ứng với một đối tượng cụ thể ở khung hình trước có tọa độ A(x, y), điểm A' là tâm của bounding box đó ở khung hình kế tiếp có tọa độ A'(x', y'). Khi đó ta sẽ xác định khoảng cách giữa điểm A và A' theo công thức Euclid là:

$$d_{AA'(pixel)} = \sqrt{(x - x')^2 + (y - y')^2}$$

Vì khoảng cách này được tính theo đơn vị là pixel nên ta cần có sự quy đổi sang đơn vị đo chiều dài thực tế. Trong mô hình áp dụng các góc quay của camera chúng tôi thiết lập có giá trị ppm (pixel per meters) bằng 8 hay nói cách khác 1 mét = 8 pixel. Vì vậy để đổi từ pixel sang mét ta sẽ có công thức:

$$d_{AA'(m)} = \frac{d_{AA'(pixel)}}{ppm}$$

Tiếp theo để xác định thời gian phương tiện di chuyển giữa 2 khung hình liên tiếp ta cần xác định được tốc độ xử lý FPS (frame per second) của hệ thống, với hệ thống phần cứng mà chúng tôi triển khai có khả năng xử lý là 40FPS tức có thể tải được 40 khung hình trong 1 giây từ đó ta có thể suy luận ra thời gian cần xử lý giữa 2 khung hình liên tiếp nhau là $t = 1/40$ (giây).

Sau khi xác định được khoảng cách và thời gian di chuyển trên quãng đường đó ta sẽ xác định được vận tốc xe di chuyển giữa 2 khung hình liên tiếp theo công thức:

$$v = \frac{d_{AA'(m)}}{t} \text{ (m/s)}$$

$$v = \frac{d_{AA'(m)}}{t} * 3,6 \text{ (km / h)}$$

Với mỗi một phương tiện từ khi xuất hiện đến khi rời khỏi khu vực khoanh vùng ta sẽ lưu lại tất cả các giá trị vận tốc giữa các khung hình liên tiếp rồi từ đó ta có thể tính vận tốc trung bình của mỗi một phương tiện từ lúc xuất hiện trong khu vực khoanh vùng cho đến thời điểm đang xét theo công thức:

$$v_{each} = \frac{1}{n} \sum_{i=1}^n v_i$$

Trong đó:

v_{each} : là trung bình của vận tốc của một phương tiện từ lúc xuất hiện đến thời điểm xét

v_i : là vận tốc giữa 2 khung hình thứ i

n: là thời điểm xét đến giữa 2 khung hình thứ n

Sau khi xác định được vận tốc trung bình của mỗi phương tiện xuất hiện trong vùng quan tâm ta sẽ tiếp tục xác định trung bình của vận tốc của tất cả các phương tiện có trong khu vực khoanh vùng đó theo công thức:

$$v_{average_speed_total} = \frac{1}{n} \sum_{i=1}^n v_{each(i)}$$

Trong đó:

$v_{average_speed_total}$: là trung bình của vận tốc của toàn bộ các phương tiện xuất hiện trong khu vực khoanh vùng

$v_{each(i)}$: Trung bình của vận tốc của phương tiện thứ i trong khu vực khoanh vùng

n : là số lượng phương tiện giao thông có trong khu vực khoanh vùng đó.

Từ các thông số trên, ta sẽ đưa mô hình vào thực tế để chạy đánh giá thống kê dữ liệu, kết hợp giữa 2 thông số mật độ lưu lượng và vận tốc trung bình của tất cả phương tiện xuất hiện trong khu vực khoanh vùng để đưa phù hợp nhất để phân loại tình trạng giao thông tại khu vực là tắc nghẽn hay thông thoáng,... Khi có

được ngưỡng phù hợp ta sẽ gửi dữ liệu này đến cho người dùng nhờ vào ứng dụng Telegram, đây là một ứng dụng nhắn tin gọi điện miễn phí qua internet.

IV. Kết quả

Phần cứng sử dụng: Core i7 12700H, RAM 16GB, RTX 3060.

4.1. Xây dựng bộ dữ liệu huấn luyện

Bộ dữ liệu được xây dựng bằng nhiều cách, bao gồm dữ liệu được nhóm đi quay hình thực tế về giao thông Việt Nam, kết hợp cùng các tập dữ liệu có sẵn trên thế giới được đánh giá uy tín. Việc này giúp cho bộ dữ liệu được xây dựng có tính tổng quát hóa, phù hợp thực tế thực trạng của giao thông đô thị Việt Nam, Giúp đạt hiệu quả cao cho quá trình huấn luyện mô hình.

Bộ dữ liệu gồm 5 phương tiện: bicycle (xe đạp), motorcycle (xe máy, mô tô), car (xe hơi), truck (xe tải), bus (xe buýt).

Dữ liệu video đã được chia thành các khung hình, mỗi khung hình sẽ thực hiện đánh nhãn cho từng đối tượng phương tiện trong khung hình đó.



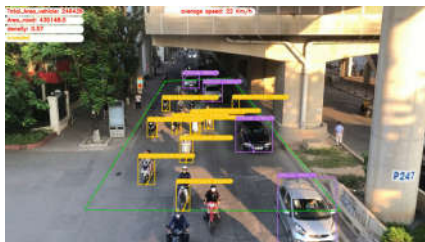
Hình 5. Minh họa bộ dữ liệu đào tạo

4.2. Kết quả

4.2.1. Kết quả của hệ thống cảnh báo tắc nghẽn giao thông



Hình 6. Thông báo giao thông “thông thoáng” Hình 7. Thông báo giao thông “tắc nghẽn”



Hình 8. Thông báo giao thông “đông đúc”

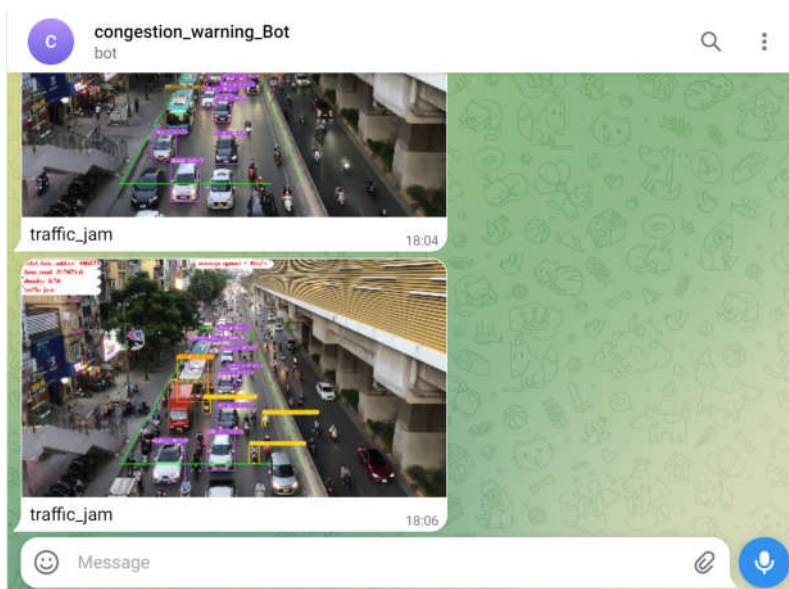
Hệ thống có khả năng phát hiện tắc nghẽn và đánh giá tình trạng giao thông dựa trên mật độ và vận tốc trung bình của

các phương tiện. Camera có thể đánh giá sự khác biệt giữa các làn đường trong khu vực khoanh vùng, như hình 6 (thông báo giao thông thông thoáng) và hình 7 (thông báo tắc nghẽn giao thông). Để phân biệt giữa đường đông nhưng lưu thông tốt và đường tắc nghẽn, hệ thống sử dụng yếu tố vận tốc. Hình 8 minh họa trường hợp mật độ phương tiện cao nhưng vận tốc tốt, nên thông báo là “đông đúc” thay vì “tắc nghẽn”.

Bảng 1. So sánh các phương pháp xử lý tắc nghẽn

Phương pháp	Giảm thời gian chờ do tắc nghẽn	Độ chính xác (%)	Tốc độ xử lý (FPS)	Dự đoán mỗi khung hình	Phần cứng
Sử dụng DxNAT (Sun, 2017)	N/A	90,7	N/A	N/A	GPU
Q-learning (Mansour, 2019)	60%	N/A	N/A	N/A	N/A
Đề xuất	N/A	96%	40	25ms	GPU: RTX 3060

4.2.2. Kết quả của hệ thống gửi thông báo đến người dùng bằng Telegram



Hình 9. Hệ thống gửi thông báo đến người dùng về tình trạng tắc nghẽn khu vực khoanh vùng

V. Kết luận

Nghiên cứu đã phát triển một hệ thống cảnh báo tắc nghẽn trong giao thông đô thị, kết hợp model YOLOv8 và DeepSORT cùng các thuật toán tính toán

mật độ và vận tốc của các phương tiện giao thông. Mục tiêu là cung cấp một giải pháp thông minh và hiệu quả để giảm tắc nghẽn và tăng an toàn cho người tham gia giao thông.

Chúng tôi đã đánh giá hiệu suất của hệ thống từ qua việc huấn luyện mô hình trên bộ dữ liệu đã xây dựng và kết quả cho thấy độ chính xác 96% cho kết quả classification và tracking các phương tiện di chuyển, cũng như cảnh báo tắc nghẽn một cách kịp thời và chính xác.

Ứng dụng của hệ thống này có thể đưa ra thông tin giúp hỗ trợ việc quản lý và điều phối giao thông đô thị, góp phần giảm tắc nghẽn và tăng an toàn. Tuy nhiên, có nhiều hướng phát triển tiềm năng, bao gồm cải thiện độ chính xác và tốc độ xử lý của các mô hình, cùng việc mở rộng hệ thống để áp dụng cho các mô hình lớn hơn và tích hợp các tính năng hỗ trợ quyết định quản lý giao thông đa dạng.

Tài liệu tham khảo

- [1] Faisal, M. M. (n.d.). *YOLOv8-DeepSORT-Object-Tracking [GitHub repository]*. GitHub. <https://github.com/MuhammadMoinFaisal/YOLOv8-DeepSORT-Object-Tracking>
- [2] King, R. (2023). *Brief summary of YOLOv8 model structure*. GitHub. <https://github.com/ultralytics/ultralytics/issues/189>
- [3] Lei, X., Qing, W., Xiumin, C., Jun, W., & Ping, C. (2008). Traffic jam detection based on corner feature of background scene in video-based ITS. In *Proceedings of the 2008 IEEE International Conference on Networking, Sensing and Control* (pp. 614-619). IEEE. <https://doi.org/10.1109/ICNSC.2008.4525291>
- [4] Mansour, A., & Rizk, M. R. M. (2019). Towards traffic congestion-free through intelligent traffic control system. In *2019 7th International Japan-Africa Conference on Electronics, Communications, and Computations (JACECC)* (pp. 5-8). IEEE. <https://doi.org/10.1109/JACECC49297.2019.9051234>
- [5] Pan, X., Guo, Y., & Men, A. (2010). Traffic surveillance system for vehicle flow detection. In *2010 Second International Conference on Computer Modeling and Simulation* (Vol. 1, pp. 314-318). IEEE. <https://doi.org/10.1109/ICCMS.2010.266>
- [6] Pereira, R., Carvalho, G., Garrote, L., & Nunes, U. J. (2022). SORT and DeepSORT based multi-object tracking for mobile robotics: Evaluation with new data association metrics. *Applied Sciences*, 12(3), Article 3. <https://doi.org/10.3390/app12031061>
- [7] Sandhya, K. S., & Karthikeyan, B. (2017). Automatic traffic diversion system using traffic signals. In *2017 International Conference on Nextgen Electronic Technologies: Silicon to Software (ICNETS2)* (pp. 131-136). IEEE. <https://doi.org/10.1109/ICNETS2.2017.8067940>
- [8] Sun, F., Dubey, A., & White, J. (2017). DxNAT deep neural networks for explaining non-recurring traffic congestion. In *2017 IEEE International Conference on Big Data (Big Data)* (pp. 2141-2150). IEEE. <https://doi.org/10.1109/BigData.2017.8258171>
- [9] Wang, Z., Lu, M., Yuan, X., Zhang, J., & Van De Wetering, H. (2013). Visual traffic jam analysis based on trajectory data. *IEEE Transactions on Visualization and Computer Graphics*, 19(12), 2159-2168. <https://doi.org/10.1109/TVCG.2013.228>

DESIGN OF AN URBAN TRAFFIC CONGESTION WARNING SYSTEM USING DEEP LEARNING MODELS

*Nguyen Thanh Loi³, Nguyen Manh Hung⁴, Do Phuong Nhung⁴,
Dao Xuan Phuc⁴, Dinh Khac Long⁴*

Abstract: *The article proposes an urban traffic congestion warning system designed to directly alert drivers. The method involves detecting vehicles using the yolov8 model and tracking them with the deepsort model to assign IDs to each object. Based on these data, the system calculates traffic density and average vehicle speed. Using these parameters and experimental results, the system determines a threshold for alerts and notifies drivers of traffic conditions via the Telegram messaging platform. The system achieves high accuracy and is expanded to provide detailed traffic information and control traffic light signals.*

Keywords: *YOLOv8, deepSORT, Telegram, opencv, traffic congestion warning*

³ School of Electrical and Electronic Engineering, Hanoi University of Science and Technology

⁴ Hanoi Open University