

## ESTIMATING THE TRAFFIC FLOW SPEED ON TRAFFIC VIDEOS USING YOLOV8 AND BYTETRACK

Vu Le Quynh Phuong<sup>1\*</sup>, Pham Nguyen Khang<sup>2</sup>, Tran Nguyen Minh Thu<sup>2</sup>

<sup>1</sup>Kien Giang Teachers Training College, <sup>2</sup>Can Tho University

ARTICLE INFO	ABSTRACT
<b>Received:</b> 17/01/2024	Traffic flow is a critical aspect of economic, social, and environmental development. To assess traffic flow, estimating the speed of traffic is crucial. In this research, we propose a model for estimating traffic speed based on data collected from traffic surveillance cameras. The main objective is to count and track vehicles to estimate traffic flow by combining the YOLOv8 and ByteTrack models, then calculating the average speed of vehicles. To train and evaluate the model's performance, data collected from the Vinh Thanh Van Police - Rach Gia City, including 10,092 images and over 96,024 labeled objects in various conditions, were used. The study experimented and compared the performance of our model with models combining YOLOv8 and DeepSort. The results indicate that the proposed model has the lowest execution time and the capability to estimate traffic flow close to reality, with an accuracy of 91.39%. The dataset used in this research can be explored and utilized as a benchmark for similar problems.
<b>Revised:</b> 28/3/2024	
<b>Published:</b> 29/3/2024	
<b>KEYWORDS</b>	
Traffic flow speed	
YOLOv8	
ByteTrack	
Object recognition	
Object tracking	

## ƯỚC LƯỢNG TỐC ĐỘ LUỒNG GIAO THÔNG TRÊN VIDEO GIAO THÔNG SỬ DỤNG YOLOV8 VÀ BYTETRACK

Vũ Lê Quỳnh Phương<sup>1\*</sup>, Phạm Nguyên Khang<sup>2</sup>, Trần Nguyễn Minh Thu<sup>2</sup>

<sup>1</sup>Trường Cao đẳng Sư phạm Kiên Giang, <sup>2</sup>Trường Đại học Cần Thơ

THÔNG TIN BÀI BÁO	TÓM TẮT
<b>Ngày nhận bài:</b> 17/01/2024	Lưu lượng giao thông là một lĩnh vực quan trọng trong sự phát triển của kinh tế, xã hội và môi trường. Để đánh giá lưu lượng giao thông, việc ước lượng tốc độ luồng giao thông là quan trọng. Trong bài nghiên cứu này, chúng tôi đề xuất một mô hình ước lượng tốc độ luồng giao thông dựa trên dữ liệu thu thập từ các camera giám sát giao thông. Mục tiêu chính là đếm và theo dõi các phương tiện để ước lượng lưu lượng giao thông bằng cách kết hợp mô hình YOLOv8 và ByteTrack, sau đó tính toán tốc độ trung bình của các phương tiện. Để huấn luyện và đánh giá hiệu suất của mô hình, dữ liệu thu thập từ Công an phường Vĩnh Thanh Vân – Thành phố Rạch Giá, bao gồm 10 092 ảnh và hơn 96 024 đối tượng được gán nhãn trong nhiều điều kiện khác nhau được sử dụng. Trong nghiên cứu đã thử nghiệm và so sánh hiệu suất của mô hình của mình với các mô hình kết hợp YOLOv8 và DeepSort. Kết quả thực nghiệm cho thấy rằng mô hình đề xuất có thời gian thực thi thấp nhất và có khả năng ước lượng lưu lượng giao thông gần với thực tế, với độ chính xác là 91,39%. Bộ dữ liệu sử dụng trong nghiên cứu này có thể được nghiên cứu sử dụng như một tập kiểm thử đối với các bài toán tương tự.
<b>Ngày hoàn thiện:</b> 28/3/2024	
<b>Ngày đăng:</b> 29/3/2024	
<b>TỪ KHÓA</b>	
Tốc độ luồng giao thông	
YOLOv8	
ByteTrack	
Nhận dạng đối tượng	
Theo dõi đối tượng	

DOI: <https://doi.org/10.34238/tnu-jst.9604>

\* Corresponding author. Email: vlqphuong@cdsphkg.edu.vn

## 1. Giới thiệu

Tắc nghẽn giao thông đã trở thành một trong những thách thức lớn trong thời kỳ hiện đại, do nhu cầu di chuyển nhanh chóng giữa các địa điểm khác nhau. Đồng thời, với sự gia tăng đô thị hóa, tăng dân số và số lượng phương tiện trong mạng lưới đường đô thị, vấn đề này ngày càng trở nên nghiêm trọng. Để giải quyết vấn đề này, việc ước lượng tốc độ luồng giao thông trở thành quan trọng để đánh giá tình trạng giao thông đường bộ, hỗ trợ quy hoạch đô thị và đóng góp quan trọng cho phát triển kinh tế và xã hội. Việc này không chỉ giúp doanh nghiệp đưa ra quyết định chính xác về quảng cáo ngoài trời và lựa chọn vị trí văn phòng mà còn giúp tiết kiệm thời gian và nhiên liệu khi lên lịch trình di chuyển mà không bị gián đoạn.

Hiện nay, có nhiều phương pháp để đánh giá tình trạng tắc nghẽn, bao gồm cả các phương pháp dựa vào cảm biến, dựa vào người tham gia giao thông sử dụng Internet và phương pháp sử dụng thị giác máy tính. Dữ liệu giao thông thường được thu thập bằng cách sử dụng cảm biến và máy chiếu hồng ngoại, tuy nhiên những phương pháp này thường tốn nhiều công sức, chi phí và không luôn đảm bảo độ chính xác [1]. Trong bối cảnh này, sự phát triển của Internet và dịch vụ bản đồ giao thông của Google mang lại một nguồn thông tin phong phú về tình trạng giao thông, giúp phân loại lưu lượng giao thông dựa trên tốc độ di chuyển [2]. Dựa vào tốc độ luồng giao thông để phân loại lưu lượng giao thông, sự giúp đỡ của dịch vụ bản đồ giao thông Google là không thể phủ nhận. Tuy nhiên, đây là một phương pháp phụ thuộc hoàn toàn vào người tham gia giao thông sử dụng điện thoại thông minh kết nối Internet, vì vậy kết quả có thể bị nhiễu hoặc chịu ảnh hưởng lớn từ sự không nhất quán trong dữ liệu.

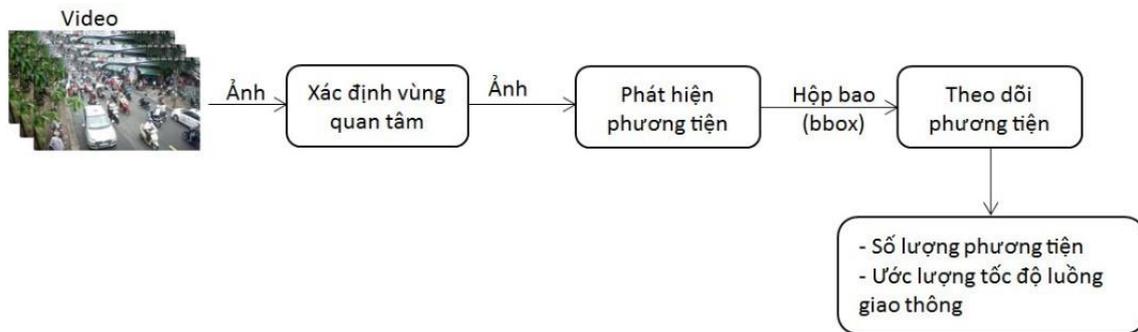
Ngoài ra, nhiều đường trên cả nước đã được trang bị hệ thống giám sát video (CCTV), tạo ra một lượng lớn dữ liệu. Dữ liệu này có thể phục vụ tốt trong lĩnh vực thị giác máy tính có hiệu quả. Nghiên cứu trước đó đã chứng minh rằng dữ liệu từ CCTV có thể giải quyết nhiều vấn đề trong hệ thống giao thông thông minh, như ước lượng mật độ giao thông [3] – [5] và nhận dạng hành vi không phù hợp. Điều này mở ra nhiều khả năng để ứng dụng thị giác máy tính và học sâu để giải quyết các thách thức trong lĩnh vực này, như được thể hiện trong nghiên cứu [6] về việc sử dụng Yolov3 và thuật toán theo dõi đối tượng SORT để tính toán lưu lượng giao thông. Những tiến bộ trong lĩnh vực này giúp nâng cao hiệu quả và chính xác của quá trình đánh giá tình trạng giao thông.

Mục tiêu của nghiên cứu này là xây dựng một mô hình để ước lượng tốc độ của luồng giao thông. Dữ liệu được thu thập chủ yếu tập trung vào các khu vực có mật độ giao thông cao nhất tại tỉnh Kiên Giang, đặc biệt là vào các giờ cao điểm. Để đạt được mục tiêu này, mô hình YOLOv8 [7] đã được huấn luyện để trích xuất đặc trưng của các phương tiện giao thông. Tiếp theo, mô hình ByteTrack [8] được sử dụng để theo dõi và đếm các đối tượng trong thời gian thực. Cuối cùng, thông qua quá trình này, tốc độ của các phương tiện tham gia giao thông trên đường được ước lượng. Hệ thống được xây dựng có khả năng đếm và ước lượng tốc độ luồng giao thông với độ sai số tương đối trung bình và thời gian thực thi thấp, đáp ứng đầy đủ yêu cầu về xử lý và tính toán trong thời gian thực. Các phần tiếp theo của bài báo sẽ trình bày chi tiết các phương pháp nghiên cứu trong Phần 2, sau đó là kết quả đạt được khi triển khai và đánh giá mô hình trên bộ dữ liệu kiểm thử trong Phần 3. Cuối cùng, Phần 4 sẽ đưa ra nội dung kết luận của nghiên cứu.

## 2. Phương pháp nghiên cứu

### 2.1. Kiến trúc tổng quan

Nghiên cứu này sử dụng thuật toán YOLOv8 để xác định các phương tiện giao thông. Tiếp theo, thuật toán theo dõi đối tượng ByteTrack được tích hợp để xác định các phương tiện và gán địa chỉ nhận dạng (ID) cho mỗi xe. Sau đó, số lượng xe được đếm trong từng khoảng thời gian cụ thể và trong vùng quan tâm. Bước cuối cùng của quy trình là ước lượng tốc độ của 5 phương tiện gần trung tâm khung hình để tính toán tốc độ trung bình của luồng giao thông. Lưu đồ thuật toán ước lượng tốc độ luồng giao thông được minh họa trong Hình 1.



**Hình 1.** Sơ đồ tổng quan của hệ thống

## 2.2. Phát hiện phương tiện giao thông với giải thuật YOLOv8

Phát hiện đối tượng là bài toán cơ bản nhưng đóng vai trò dùng để phân loại và xác định vị trí các đối tượng vật thể có trong ảnh hoặc video. Nhờ sự phát triển nhanh chóng về mặt dữ liệu cũng như sự xuất hiện thêm nhiều giải thuật mới, bài toán phát hiện đối tượng đã đạt được nhiều bước tiến đáng kể và được ứng dụng rất nhiều trong thực tế. Đặc biệt, sự xuất hiện của các bộ phát hiện đối tượng một giai đoạn đã thay đổi cách tiếp cận vấn đề phát hiện đối tượng. Các mô hình trước đó chia thành từng giai đoạn riêng biệt để tạo ra các đề xuất đối tượng, thì trong các mô hình một giai đoạn xem xét toàn bộ hình ảnh và thực hiện cả phân loại và hồi quy trực tiếp trong một mạng nơ-ron tổng thể. Các giải thuật này không phải tạo ra các đề xuất cố định trước, mà thay vào đó xem xét tất cả các vị trí tiềm năng trên ảnh và dự đoán xem chúng có chứa đối tượng hay không.

Các bộ phát hiện dựa trên hộp "anchor" đã được xác định trước, với các tỷ lệ khung hình khác nhau, nhằm phục vụ cho việc phát hiện các đối tượng có hình dạng và kích thước đa dạng. Mô hình nổi bật về phương pháp này là "You Only Look Once" (YOLO) [9] – [11]. YOLO hoạt động bằng cách biểu diễn hình ảnh đầu vào như một lưới các ô, trong đó mỗi ô có trách nhiệm dự đoán một hộp giới hạn nếu trung tâm của hộp nằm trong ô đó. Mỗi ô lưới dự đoán nhiều hộp giới hạn và đầu ra vị trí và nhãn lớp cùng với điểm tin cậy (confidence score). YOLO được đánh giá tốt về tốc độ và đơn giản nhưng có nhược điểm là tỷ lệ thu hẹp tương đối thấp. Để khắc phục những hạn chế này, YOLOv7 đã đưa ra nhiều cải tiến quan trọng so với các phiên bản trước đó, mang lại khả năng phát hiện đối tượng trong ảnh một cách hiệu quả hơn [12]. Một trong những cải tiến quan trọng đó là việc sử dụng các hộp "anchor". Tuy nhiên, YOLOv7 gặp khó khăn trong việc phát hiện các đối tượng nhỏ, ở các tỷ lệ khác nhau, dưới sự thay đổi về ánh sáng hoặc các điều kiện môi trường khác.

YOLOv8 đã giải quyết thách thức này bằng cách áp dụng phương pháp phát hiện đối tượng dưới dạng bài toán hồi quy thay vì phân loại. Điều này mang lại ưu điểm cho mô hình YOLOv8 khi có khả năng cung cấp dự đoán chính xác cho các lớp đã được huấn luyện và dễ dàng học các lớp mới. Bên cạnh đó, mô hình YOLOv8 cải tiến so với các mô hình trước đó bao gồm việc loại bỏ việc sử dụng hộp "anchor" và thêm vào đó kỹ thuật tăng cường dữ liệu "mosaic".

Hộp "anchor" trước đây được sử dụng để xác định hình chữ nhật bao quanh đối tượng, nhưng trong một số trường hợp, nhất là khi đối tượng trong ảnh không đồng đều, hộp "anchor" hình đa giác không thể ánh xạ một cách rõ ràng. YOLOv8 đã loại bỏ hộp "anchor" để cải thiện khả năng phát hiện đối tượng trong các tình huống phức tạp.

Kỹ thuật tăng cường dữ liệu "mosaic" là một cải tiến đơn giản, kết hợp bốn hình ảnh khác nhau và đưa chúng vào mô hình như đầu vào. Điều này giúp mô hình học được đối tượng từ nhiều góc độ và trong các trạng thái bị che khuất. Tuy nhiên, để giảm thiểu ảnh hưởng tiêu cực đối với hiệu suất, việc thực hiện tăng cường dữ liệu "mosaic" chỉ được áp dụng trong 10 vòng lặp

cuối cùng của quá trình huấn luyện. Những thay đổi này đã cải thiện độ chính xác và giảm thời gian suy luận của mô hình YOLOv8, đồng thời tăng cường khả năng học và phát hiện đối tượng trong các bối cảnh đa dạng và phức tạp.

### 2.3. Theo dõi phương tiện giao thông với mô hình ByteTrack

Sau khi phương tiện được xác định, bước tiếp theo của hệ thống là sử dụng mô hình ByteTrack để thực hiện theo dõi các đối tượng. ByteTrack lựa chọn những đối tượng có điểm tin cậy (confidence score) cao để bắt đầu quá trình theo dõi. Sau khi mô hình YOLOv8 đã dự đoán các hộp giới hạn và điểm tin cậy tương ứng cho mỗi đối tượng, mô hình ByteTrack tiếp tục xử lý thông tin này. Ý tưởng chính của ByteTrack là lựa chọn những đối tượng có điểm tin cậy cao xử lý trước. Lúc này, các hộp giới hạn được phân loại thành hai nhóm: một nhóm chứa các hộp có độ chắc chắn cao ( $\geq 0,6$ ) và một nhóm chứa các hộp có độ chắc chắn thấp ( $0,1 \leq s \leq 0,6$ ). Thuật toán lọc Kalman được áp dụng để dự đoán vị trí mới của các đối tượng theo dõi.

Đối với nhóm các hộp giới hạn có điểm tin cậy cao, ByteTrack sử dụng sự tương đồng giữa các hộp và các đối tượng theo dõi để thực hiện quá trình liên kết. Quá trình này có thể sử dụng độ chồng chéo (IoU) hoặc khoảng cách đặc trưng Re-ID. Trong khi đó, đối với nhóm các hộp có điểm tin cậy thấp, chỉ sự tương đồng IoU được sử dụng trong quá trình liên kết.

Tiếp theo, ByteTrack loại bỏ các đối tượng theo dõi không kết hợp sau quá trình liên kết thứ hai và đưa chúng vào danh sách các đối tượng mất hoặc che khuất. Đồng thời, nếu có hộp phát hiện có điểm tin cậy cao không kết hợp, ByteTrack sẽ khởi tạo các đối tượng theo dõi mới từ chúng. Kết quả cuối cùng là mỗi khung hình đầu ra chứa thông tin về vị trí và nhãn của các đối tượng theo dõi.

### 2.4. Ước lượng tốc độ luồng giao thông

Trong mỗi khung hình, vùng thực hiện ước tính tốc độ được giới hạn bằng vùng quan tâm như Hình 2. Giả định 2 vùng này sẽ có độ rộng là 50px (có thể thay đổi cho trường hợp khác), chương trình sẽ chỉ ghi lại vị trí gần nhất so với đoạn thẳng, do đó cũng tránh được việc thông tin về khung hình bị sai lệch.



**Hình 2.** Mô tả vùng giới hạn giả định trong khung hình

Ước lượng tốc độ được thực hiện đối với mỗi đối tượng được theo dõi một cách độc lập, sử dụng công thức (1):

$$V = \frac{S}{\frac{(t_2 - t_1)}{FS}} \times H \quad (1)$$

$S$  là khoảng cách phương tiện giao thông di chuyển giữa đoạn thẳng 1 và đoạn thẳng 2 như trong mô tả Hình 2 (đơn vị tính là pixel hoặc mét);  $t_1$  là thời điểm xe bắt đầu vào đoạn thẳng 1,  $t_2$  là thời điểm xe ra khỏi đoạn thẳng 2.  $FS$  là tốc độ khung hình trên giây của video.  $H$  là hệ số chuyển đổi được sử dụng để chuyển đổi khoảng cách từ đơn vị pixel sang đơn vị thực tế (ví dụ: từ pixel sang mét). Quá trình này giúp theo dõi và cập nhật thông tin về tốc độ của các đối tượng theo thời gian trong quá trình theo dõi. Trong Hình 2, 2 đoạn thẳng giới hạn có độ rộng là 50px. Tốc độ khung hình là 10 Fps. Thời điểm xe vào là khung hình thứ 78, thời điểm xe ra là khung hình thứ 103. Vậy, tốc độ của phương tiện tính được là:

$$V = \frac{50}{\frac{(103 - 78)}{10}} = 20 \text{ (pixel / s)}$$

Trong thực tế, chiều dài giữa 2 đoạn thẳng là 18 m. Vậy, vận tốc của xe, có thể được tính là:

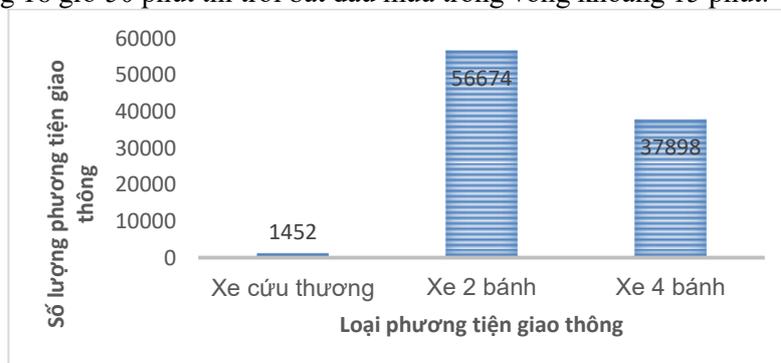
$$V = \frac{18}{\frac{(103 - 78)}{10}} = 7,2 \text{ (m/s)} = 25,92 \text{ (km/h)}$$

### 3. Kết quả thực nghiệm

Trong bài báo này, các mô hình được xử lý trên một máy tính đơn sử dụng nền tảng Google Colab với CPU Intel Xeon Processor with two cores 2.30 GHz đi kèm với 13GB ram cho phép huấn luyện với batch size lớn hơn, GPU có hiệu suất cao như GPU Nvidia Tesla T4 15102MiB giúp rút ngắn đáng kể thời gian tính toán. Các mô hình được huấn luyện với kích thước lô (batch size) được thiết lập là 8, thuật toán tối ưu ADAM với động lượng (momentum) 0,9 và tốc độ học ban đầu là 0,001. Quá trình đào tạo được thực hiện trong 25 vòng lặp. Tỷ lệ học tập sẽ giảm 10 lần nếu sau 10 vòng lặp mà độ chính xác trên tập dữ liệu xác thực không được cải thiện.

#### 3.1. Dữ liệu thực nghiệm

Dữ liệu được cung cấp bởi công an phường Vĩnh Thanh Vân – TP. Rạch Giá – tỉnh Kiên Giang. Gồm có 03 góc camera (CAM1, CAM3, CAM5). Vị trí đặt camera đường Lê Lợi (đối diện bệnh viện tỉnh Kiên Giang). Các video clip dữ liệu đều có thời lượng gần 1 tiếng. Độ phân giải ở góc CAM1 và CAM5 là 1920x1080, FPS lần lượt là: 15 khung hình/giây và 12 khung hình/giây. Riêng độ phân giải ở góc CAM3 là 1280x720, FPS: 10 khung hình/giây. Cảnh quay ở video clip góc CAM1 là vào ban ngày, trời nắng, thời gian từ 10 giờ 13 phút sáng ngày 01-11-2020 đến 11 giờ 13 phút sáng cùng ngày. Cảnh quay ở video clip góc CAM3 cũng vào ban ngày, trời nắng, thời gian từ 10 giờ tới 11 giờ sáng ngày 09-09-2019. Video clip góc CAM5 có thời gian từ 16 giờ 23 phút tới 17 giờ 23 phút chiều ngày 12-11-2020, cảnh quay ban đầu trời nắng nhưng tới khoảng 16 giờ 50 phút thì trời bắt đầu mưa trong vòng khoảng 15 phút.



Hình 3. Số lượng các phương tiện giao thông trên tập dữ liệu

Tập dữ liệu bao gồm 10082 ảnh, được gán nhãn: “0” là xe ưu tiên; “1” là xe 2 bánh bao gồm xe máy, xe đạp; “2” là nhãn xe 4 bánh ứng với các loại xe ô tô 4 chỗ, 7 chỗ, xe bus.... Số lượng các mẫu đối tượng trên 3 nhãn (0, 1, 2) lần lượt là (1 452; 56 674; 37 898) được minh họa trong Hình 3. Trung bình mỗi ảnh có 9,5 phương tiện giao thông. Xét trên từng video gốc, tập dữ liệu đã gán nhãn được chia thành 3 tập con là “train”, “test”, “valid”, tất cả các ảnh sẽ được sắp xếp theo thứ tự dòng thời gian như trong video gốc, 80% ảnh ở đầu video sẽ được sử dụng cho tập “train” và 20% đoạn sau được chia đều làm 2 phần cho tập “test” và “valid”. Thực hiện như vậy trên từng video gốc sau đó tổng hợp lại ta thu được 3 tập dữ liệu “train”, “valid”, “test” tổng hợp.

### 3.2. Phát hiện phương tiện giao thông với giải thuật YOLOv8

Sau giai đoạn thực nghiệm là quá trình đánh giá mô hình YOLOv8 trên tập dữ liệu thực nghiệm, các độ đo đánh giá IoU và độ đo giá trị trung bình của độ chính xác trung bình (mAP) được sử dụng để đánh giá trong nghiên cứu.

Mô hình được huấn luyện với kích thước lô (batch size) 8, thuật toán tối ưu SGD, tốc độ học (learning rate) ban đầu là 0,01. Quá trình đào tạo được thực hiện trong 20 vòng lặp (epochs). Huấn luyện tập dữ liệu với mô hình YOLOv8 để thực hiện việc phát hiện phương tiện giao thông đã được gán nhãn. Sau quá trình huấn luyện, độ chính xác trung bình dự đoán của huấn luyện đạt 94%, độ mất mát (loss) 1%, kích thước mô hình 22,5Mb, thời gian huấn luyện 3,326 giờ.

**Bảng 1.** Kết quả mAP trên tập kiểm tra và tập huấn luyện

Lớp	mAP 50 (%)	
	Tập huấn luyện	Tập kiểm tra
Tất cả	94	94,1
Xe ưu tiên (0)	98,1	98,2
Xe 2 bánh (1)	92,5	92,6
Xe 4 bánh (2)	91,3	91,4

Kết quả thực nghiệm cho thấy, xe ưu tiên có mAP cao nhất (98,1%) do dữ liệu huấn luyện có kích thước lớn và chất lượng. Xe 2 bánh và 4 bánh có mAP lần lượt là 92,5% và 91,3% được thể hiện trong Bảng 1. Đây là kết quả tốt, đặc biệt trong ngữ cảnh phát hiện và theo dõi các phương tiện giao thông. Mô hình có khả năng dự đoán được nhiều đối tượng khác nhau trong cùng một ảnh được minh họa trong Hình 4.



**Hình 4.** Kết quả minh họa dự đoán các đối tượng được nhận dạng và phân loại

### 3.3. Theo dõi phương tiện giao thông với mô hình ByteTrack

Đối với mô hình ước lượng giao thông, video giao thông thu thập được cắt trong 15 phút với chương trình Kdenlive để làm tập kiểm thử. Video có các thông số: 1280x720, 10FPS, tổng số frame là 9005. Dữ liệu được đếm và cập nhật mỗi phút một lần để làm tập kiểm tra cho mô hình ước lượng lưu lượng giao thông. Để đánh giá độ chính xác cũng như thời gian thực thi của

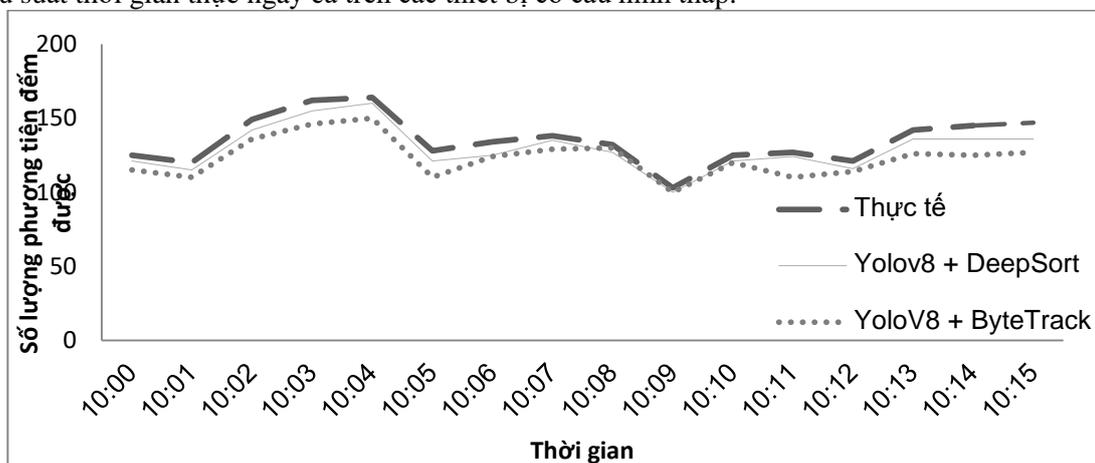
ByteTrack, nhóm nghiên cứu so sánh với giải thuật DeepSort. Kết quả trình bày trong Hình 5 và Bảng 2.

Từ kết quả thực nghiệm cho thấy, mô hình Yolov8 kết hợp DeepSort có độ chính xác của hệ thống theo dõi tốt hơn Yolov8 và ByteTrack được trình bày trong Hình 5. DeepSort có khả năng xử lý các hiện tượng che khuất và xuất hiện đối tượng phức tạp tốt hơn so với ByteTrack. Kết quả tương tự như trong nghiên cứu được thực hiện trên các tập dữ liệu MOT17 và MOT20 [13].

**Bảng 2.** So sánh thời gian và độ chính xác của mô hình DeepSort và ByteTrack

Mô hình	Độ chính xác (%)	Thời gian thực thi (s)
DeepSort	95,78	697,83
ByteTrack	91,39	536,79

Thời gian thực thi của mô hình ByteTrack vượt trội hơn DeepSort vì có kiến trúc đơn giản hơn, số lượng tham số ít hơn. ByteTrack có kích thước mô hình nhỏ hơn so với DeepSort, nên phù hợp hơn cho triển khai trong môi trường có tài nguyên hạn chế. ByteTrack có thể đạt được hiệu suất thời gian thực ngay cả trên các thiết bị có cấu hình thấp.



**Hình 5.** Kết quả minh họa dự đoán số lượng phương tiện trong mỗi phút

### 3.4. Ước lượng vận tốc luồng giao thông



**Hình 6.** Video kết quả được lưu lại

Hệ thống ước lượng vận tốc luồng giao thông được xây dựng thông qua thư viện PyQt5, mô hình phát hiện phương tiện giao thông YOLOv8 đã được huấn luyện, thuật toán theo dõi phương tiện giao thông là ByteTrack và phương pháp tính vận tốc đã được trình bày ở phần 2. Vùng xác định giới hạn là tùy chỉnh. Sau đó chương trình tính ước lượng lưu lượng luồng giao thông bằng cách tính vận tốc trung bình của tất cả các phương tiện được phát hiện. Hình 6 là video kết quả thu được sau khi hệ thống chạy hết video.

#### 4. Kết luận

Trong nghiên cứu này, nhóm tác giả đã thành công trong việc xây dựng một mô hình ước lượng lưu lượng giao thông bằng cách kết hợp mô hình phát hiện đối tượng sử dụng YOLOv8 và mô hình theo dõi đối tượng ByteTrack. Từ đó, họ có thể tính toán tốc độ của luồng giao thông bằng cách tính tốc độ trung bình của các phương tiện giao thông được phát hiện. Hệ thống đã được thử nghiệm trên các video thu thập từ môi trường thực tế, đặc biệt là ở khu vực đông đúc nhất của tỉnh Kiên Giang, Việt Nam. Nhiều góc quay và khoảng thời gian khác nhau đã được sử dụng cho các video thử nghiệm. Tốc độ thực thi của hệ thống khi thử nghiệm là 536,79 giây trên tập kiểm thử, với sai số trung bình là 13,45%. Đây là một mô hình có thể áp dụng trong thực tế. Tuy nhiên, do mục đích ban đầu của việc cài đặt camera an ninh, chưa có sự tối ưu hóa góc quay cho bài toán liên quan đến giao thông. Điều này đã ảnh hưởng đến độ chính xác của hệ thống. Mặc dù hệ thống đạt được kết quả tích cực, nhưng vẫn còn nhiều thách thức cần giải quyết. Trong hệ thống này, việc không phân loại các loại phương tiện mà chỉ tính toán chung tất cả chúng để ước lượng lưu lượng giao thông cũng là một điểm hạn chế. Tác giả coi những vấn đề này là những hướng nghiên cứu khả thi cho tương lai.

#### TÀI LIỆU THAM KHẢO/ REFERENCES

- [1] A. Fedorov, K. Nikolskaia, S. Ivanov, V. Shepelev, and A. Minbaleev, "Traffic flow estimation with data from a video surveillance camera," *Journal of Big Data*, vol. 6, pp. 1-15, 2019.
- [2] M. M. Rahman and N. Nower, "Attention based Deep Hybrid Networks for Traffic Flow Prediction using Google Maps Data," in *Proceedings of the 2023 8th International Conference on Machine Learning Technologies*, 2023, pp. 74-81.
- [3] L. Q. P. Vu, N. T. Bui, K. H. Nguyen, N. M. T. Tran, and N. K. Pham, "Estimating the traffic density from traffic cameras," in *Future Data and Security Engineering. Big Data, Security and Privacy, Smart City and Industry 4.0 Applications*, Springer, 2021, pp. 248-263.
- [4] T. Abirami, C. Nivas, R. Naveen, and G. Nithishkumar, "Deep Learning based Traffic Analysis of Motor Cycles in Urban City," in *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2022, pp. 1304-1309.
- [5] F. Zhang, C. Li, and F. Yang, "Vehicle Detection in Urban Traffic Surveillance Images Based on Convolutional Neural Networks with Feature Concatenation," *Sensors*, vol. 19, pp. 549-570, 2019.
- [6] T. Abirami, C. Nivas, R. Naveen, and T. G. Nithishkumar, "Deep Learning based Traffic Analysis of Motor Cycles in Urban City," in *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, 2022, pp. 1304-1309.
- [7] A. Vats and D. C. Anastasiu, "Enhancing Retail Checkout Through Video Inpainting, YOLOv8 Detection, and DeepSort Tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5529-5536.
- [8] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box," in *European Conference on Computer Vision*, Cham: Springer Nature Switzerland, 2022, pp. 1-21.
- [9] N. D. T. Yung, W. K. Wong, F. H. Juwono, and Z. A. Sim, "Safety helmet detection using deep learning: Implementation and comparative study using YOLOv5, YOLOv6, and YOLOv7," in *2022 International Conference on Green Energy, Computing and Sustainable Technology (GECOST) IEEE*, 2022, pp. 164-170.
- [10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *2016 IEEE conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 779-788.

- [11] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.
- [12] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7464-7475.
- [13] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng, "Strongsort: Make deepsort great again," *IEEE Transactions on Multimed*, vol. 25, pp. 8725-8737, 2023.