

# Phát Hiện Lỗi Giao Thông Của Xe Máy

Mã Kim Phát and Đặng Chí Nguyên and Đường Thị Mộng Thúy  
Trường Đại Học Công Nghệ Thông Tin - Đại Học Quốc Gia TP.HCM, VietNam  
Khoa Khoa Học & Kỹ Thuật Thông tin  
{22521071, 22520963, 22521454}@gm.uit.edu.vn

## Abstract

Trong thành phố thông minh ngày nay, việc xây dựng một hệ thống tự động nhận diện các lỗi vi phạm là vô cùng cần thiết. Điều này sẽ giúp giảm thiểu gánh nặng lên những người quản lý giao thông và những người trong công tác giám sát. Trong báo cáo này, chúng tôi sẽ đề xuất một phương pháp để nhận diện lỗi đi ngược chiều, một lỗi vô cùng phổ biến ở Việt Nam. Hệ thống chúng tôi giới thiệu sử dụng mạng tích chập YOLOv5 để trích xuất hình ảnh của đối tượng, sau đó sẽ thu thập thông tin về vị trí của đối tượng được xác định qua thuật toán theo dõi và cuối cùng sẽ áp dụng thuật toán chúng tôi đề xuất để nhận biết hành vi đi ngược chiều của người tham gia giao thông. Hệ thống chúng tôi được thử nghiệm trên tập dữ liệu video giao thông ở giờ cao điểm trên những tuyến đường thường xuyên xảy ra kẹt xe cũng như đi ngược chiều trên thành phố Hồ Chí Minh. Độ chính xác khi phát hiện vật thể là ...

## 1 Giới Thiệu

Ở Việt Nam, giao thông đường bộ là loại hình giao thông phổ biến và phát triển nhất. Tính tới năm 2023, tổng số xe máy đăng ký của Việt Nam là khoảng 73 triệu chiếc. Với sự gia tăng số lượng phương tiện nhanh chóng, cần phải có một hệ thống phát hiện vi phạm giao thông giúp cho việc quản lý, giám sát giao thông một cách hiệu quả và giảm các trường hợp lái xe có khả năng gây ra nguy hiểm.

Việc xử lý các lỗi đi ngược chiều tốn một lượng thời gian tương tự với các lỗi khác. Chính vì vậy, có những trường hợp ngó lơ hành vi vi phạm này, thế nhưng hành vi này có thể gây ra khả năng tai nạn giao thông cho những người tham gia đúng. Để khắc phục vấn đề này, chúng tôi đề xuất một hệ thống giúp nhận diện lỗi đi ngược chiều có thể xử lý ngay trong thời gian thực.

- Input:** Một video quay lại tình hình giao thông.

- Output:** Bbox bao quanh đối tượng và tình trạng xe đi đúng chiều hay đi ngược chiều.

## 2 Bộ Dữ liệu

### 2.1 Tổng Quan Về Bộ Dữ Liệu

Trong đề tài này, chúng tôi đã sử dụng một phần của bộ dữ liệu (Ho Chi Minh Traffic Dataset). Đây là bộ dữ liệu bao gồm 8 Video giao thông tại các nút giao ở TPHCM được thu thập bởi Phòng thí nghiệm truyền thông đa phương tiện (Multimedia Communications Laboratory, hay MMLab) thuộc Trường Đại học Công nghệ thông tin, ĐHQG-Tp.HCM.<sup>1</sup> Trong đó, chúng tôi đã sử dụng 1 video trong bộ dữ liệu nhằm mục đích huấn luyện mô hình YOLOv5 nhận diện xe máy và 1 video khác cho việc phát hiện phương tiện đi ngược chiều.

### 2.2 Bộ Dữ Liệu Dùng Để fine-tuning Mô Hình YOLOv5

#### 2.2.1 Tổng quan

Nhằm phục vụ mục đích fine-tuning mô hình YOLOv5, chúng tôi đã cắt khung hình từ 1 video trong bộ dữ liệu.



Figure 1: Ảnh trong bộ dữ liệu tinh chỉnh mô hình

Bộ dữ liệu huấn luyện bao gồm tổng cộng 1.501 ảnh được chia thành 3 bộ train:dev:test với tỉ lệ lần

<sup>1</sup><https://mmlab.uit.edu.vn/dataset/2023/09/25/datasets9>

lượt là 7:1:2.<sup>2</sup>

## 2.2.2 Gán nhãn dữ liệu

Chúng tôi sử dụng công cụ labelimg cho việc gán nhãn dữ liệu vì nhiều mục đích:

- Giao diện sử dụng thân thiện, dễ sử dụng.

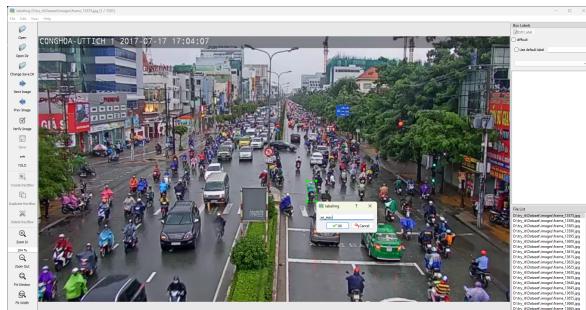


Figure 2: Giao diện sử dụng labelimg

- Định dạng nhãn phù hợp với việc huấn luyện mô hình YOLO.

Thực hiện gán nhãn với quy tắc:

- Chỉ gán nhãn với đối tượng là xe máy.
- Mỗi Bounding Box ôm trọn một và chỉ một đối tượng.

### ghi tiếp kết quả sau khi gán nhãn

## 2.3 Dữ liệu cho nhận diện hành vi vi phạm

Chúng tôi sử dụng một video với độ dài 20 giây và 15 fps để kiểm thử khả năng nhận diện hành vi đi ngược chiều của người tham gia giao thông. Trong video có sự xuất hiện dày đặc của các phương tiện tham gia giao thông như xe máy, ô tô, người đi bộ,... Video với mật độ dày đặc như vậy phù hợp và có thể đại diện được cho tình hình giao thông chung tại Việt Nam.

## 3 Hướng Tiếp Cận

### 3.1 Tổng Quan

Trong đề tài này, giải pháp được nhóm chúng tôi thực hiện được chia làm 3 giai đoạn:

- Giai đoạn 1: Sử dụng mô hình học sâu để nhận diện và phát hiện đối tượng.
- Giai đoạn 2: Sử dụng thuật toán theo dõi đối tượng (object tracking) để theo dấu và trích xuất thông tin vị trí của đối tượng qua nhiều khung hình liên tục.

<sup>2</sup><https://www.kaggle.com/datasets/nguyndc/traffic-data>

- Giai đoạn 3: Sử dụng thuật toán do nhóm đề xuất nhằm trích xuất vector di chuyển của từng vật thể từ danh sách vị trí của từng đối tượng. Từ vector hướng di chuyển trên có thể xác định phương tiện đi đúng hay ngược chiều.

## 3.2 YOLOv5 và lý do lựa chọn nó

### 3.2.1 Lý do lựa chọn YOLOv5

Hiện nay, nhóm họ mô hình phát hiện vật thể phổ biến nhất là nhóm single-stage detectors (mô hình phát hiện vật thể 1 giai đoạn) và multi-stage detectors (mô hình phát hiện vật thể nhiều giai đoạn). Trong đó họ YOLO là đại diện tiêu biểu nhất của single-stage detectors và Faster R-CNN là đại diện của multi-stage detectors. Trong bài báo [1], nhóm tác giả đã cho rằng đây là 2 mô hình tốt nhất có thể đại diện cho mỗi nhóm.

Cũng trong bài báo này, ở phần Testing, tác giả đã đưa ra được kết luận về độ chính xác và tốc độ xử lý của mô hình như sau:

- Về độ chính xác, nhóm tác giả sử dụng nhiều độ đo. Trong đó, độ đo mAP@.5:.95 của mô hình YOLOv5 và Faster R-CNN lần lượt là 31,725% và 41,03%
- Về tốc độ xử lý, tốc độ xử lý trên 1 khung hình của YOLOv5 và Faster R-CNN lần lượt là 0.417s/frame và 4.12s/frame.

Qua kết quả so sánh trên, ta có thể thấy rằng về độ chính xác, YOLOv5 thấp hơn khoảng 10% so với Faster R-CNN nhưng về tốc độ xử lý lại cao hơn gấp xấp xỉ 8 lần. Vì vậy, với một đề tài yêu cầu xử lý trên dữ liệu video như đề tài của chúng tôi, chúng tôi cần ưu tiên tốc độ xử lý hơn là độ chính xác. Vì lý do trên, chúng tôi quyết định sử dụng mô hình YOLOv5 để giải quyết giai đoạn 1 của đề tài.

### 3.2.2 Cấu trúc mạng YOLOv5

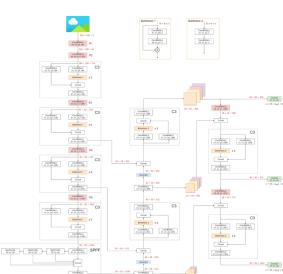


Figure 3: Cấu trúc YOLOv5

Cấu trúc của YOLOv5 gồm 3 phần chính:

- **Backbone:** Đây là thành phần chính của mạng. Với YOLOv5, phần Backbone được thiết kế bằng cách sử dụng cấu trúc **CSP-Darkness53** mới, được tinh chỉnh từ cấu trúc Darkness được sử dụng ở phiên bản trước.
- **Neck:** Phần này sẽ kết nối phần Backbone và Head. Trong YOLOv5, **SPPF** và cấu trúc **CSP-PAN** mới sẽ được tận dụng.
- **Head:** Phần này sẽ đóng vai trò tạo ra kết quả cuối cùng. YOLOv5 sử dụng lại Head của YOLOv3 ở phần này.

### 3.3 Thuật toán theo dõi đối tượng

Trong đề tài này, chúng tôi đã sử dụng thuật toán DeepSort(Simple Online and Realtime Tracker with Deep Learning) [2]. Đây là thuật toán theo dõi đa đối tượng (multi-object tracking - MOT) được giới thiệu lần đầu vào năm 2017.

Dựa trên tiền thân là thuật toán theo dõi SORT, DeepSort đã khắc phục nhược điểm của SORT bằng cách thêm vào 1 mô hình học sâu trích xuất đặc trưng giúp theo dõi chính xác hơn các vật thể trong các trường hợp vật thể bị che khuất, biến mất trong 1 vài khung hình,...

DeepSort hoạt động dựa trên 3 thành phần cốt lõi:

- **Kalman Filter:** Là thuật toán dự đoán vị trí tiếp theo của vật thể kể cả khi vật thể bị che khuất. Sau đó dựa vào những phát hiện mới để cập nhật trạng thái của vật thể.
- **Hungarian Algorithm:** Là thuật toán được đề xuất vào năm 1955. Đây là thuật toán dùng để giải bài toán phân chia công việc. Thuật toán được áp dụng trong việc liên kết các phát hiện mới với các đường dự đoán của Kalman Filter sao cho độ sai số là nhỏ nhất.
- **Deep Learning:** Như đã nói ở trên, Deep Learning cũng được áp dụng trong DeepSort nhằm trích xuất những vector đặc trưng về màu sắc, hình dạng,... của các vật thể nhằm phân biệt được các vật thể tương tự nhau.

### 3.4 Thuật toán nhóm đề xuất

#### 3.4.1 Hướng tiếp cận bán tự động

Nhóm chúng tôi sẽ vẽ quy định hai khu vực để xác định lỗi như sau:

Sau đó, ở mỗi vùng sẽ có 1 vector chiều để quy định hướng di chuyển của dòng xe

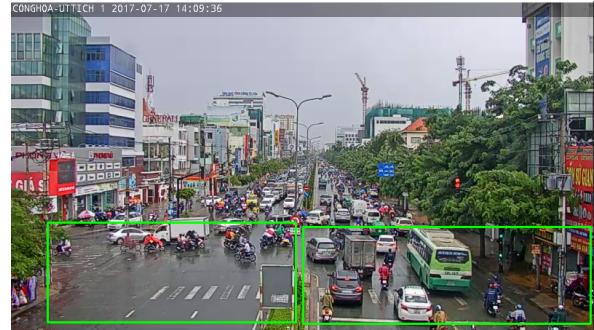


Figure 4: Quy định chiều di chuyển

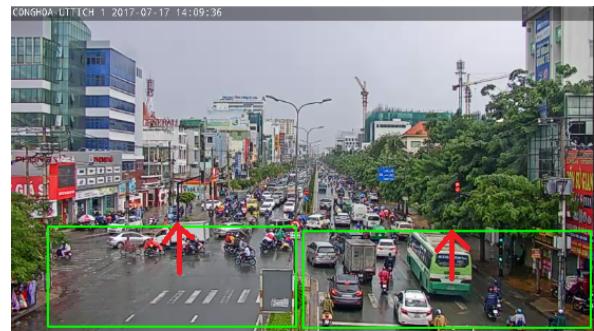


Figure 5: Vector hướng di chuyển của đường đi

Thuật toán theo dõi đối tượng sẽ trả về vị trí của mỗi đối tượng sau từng frame, khi đó ta sẽ có được vị trí của vật thể sau mỗi frame, từ đó chúng tôi sẽ tính được vector chiều của vật thể duy chuyển sau mỗi 2 frame với công thức tính vector như sau:

$$\overrightarrow{AB} = (x_B - x_A, y_B - y_A)$$

Với:

$x_A, y_A$  là tọa độ của vị trí trước đó của vật thể,  
 $x_B, y_B$  là tọa độ của vị trí sau đó của vật thể.



Figure 6: Hướng di chuyển của phương tiện

Sau khi có được vector chiều của vùng nhóm quy định và vector chiều di chuyển của phương tiện, nhóm sẽ tính góc giữa hai vector đó, **được gọi là  $\alpha$** .

Góc giữa hai vector  $a$  và  $b$  được tính bằng công thức:

$$\cos \alpha = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}$$

Với:

$$\mathbf{a} \cdot \mathbf{b} = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n,$$

$$\|\mathbf{a}\| = \sqrt{a_1^2 + a_2^2 + \cdots + a_n^2},$$

$$\|\mathbf{b}\| = \sqrt{b_1^2 + b_2^2 + \cdots + b_n^2}.$$

Với mỗi góc quay camera khác nhau, chúng tôi sẽ quy định **một góc  $\beta$  khác nhau** để phù hợp với thực tế.

Đối với vùng chỉ thấy được đuôi phương tiện. Giả sử  $\beta = 90^\circ$ , ta có các trường hợp sau:

- **Nếu  $\alpha < \beta$** : Xe đi đúng chiều so với vùng chỉ thấy đuôi xe.
- **Nếu  $\alpha \geq \beta$** : Xe đi ngược chiều so với vùng chỉ thấy đuôi xe.

Ngược lại, đối với vùng chỉ thấy được đầu phương tiện. Cũng với  $\beta = 90^\circ$ , ta có các trường hợp sau:

- **Nếu  $\alpha \leq \beta$** : Xe đi ngược chiều so với vùng chỉ thấy đầu xe.
- **Nếu  $\alpha > \beta$** : Xe đi đúng chiều so với vùng chỉ thấy đầu xe.

### 3.4.2 Hướng tiếp cận tự động

Với mỗi đối tượng theo dõi, chúng tôi sẽ tính khoảng cách của đối tượng đó tới những đối tượng đã bắt được còn lại trong khung hình, theo công thức như sau:

$$d(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\| = \sqrt{\sum_{i=1}^n (u_i - v_i)^2}$$

Sau đó chúng tôi sẽ chọn ra k vector gần với đối tượng theo dõi nhất để quan sát trạng thái của đối tượng di chuyển. Tiếp theo, chúng tôi sẽ tính góc giữa hai vector nhằm biết được hướng đi của đối tượng, theo công thức như sau và quy tắc như sau:

#### Định nghĩa:

- $n$ : Số vector trạng thái ( $n < k$ ).
- $m$ : Số vector trạng thái khác  $n$  ( $m < k - n$ ).
- $\mathbf{d}$ : Là vector đối tượng đang theo dõi
- Vector dùng để so sánh:  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ .

$$\cos \theta_i = \frac{\mathbf{v}_i \cdot \mathbf{d}}{\|\mathbf{v}_i\| \|\mathbf{d}\|}$$

#### Quy tắc:

- $\theta_i < 90^\circ$  đối tượng sẽ đi đúng chiều
- $\theta_i \geq 90^\circ$  đối tượng sẽ đi ngược chiều

Cuối cùng chúng tôi sẽ so sánh trạng thái của đối tượng với k vector và sẽ cho ra kết quả, theo quy tắc như sau:

#### Trạng thái của vector theo dõi:

$$\mathbf{v}_{\text{theo dõi}} = \begin{cases} \text{Theo chiều của } \mathbf{v}_i, & \text{nếu } \cos \theta_i > 0 \\ \text{Ngược chiều với } \mathbf{v}_i, & \text{nếu } \cos \theta_i < 0 \end{cases}$$

**Ví dụ**, chúng tôi theo dõi 5 vector gần nhất với đối tượng đang theo dõi, trong đó:

- Có 3 đối tượng là đi đúng chiều
- Có 2 đối tượng là đi ngược chiều

**Suy ra**, đối tượng chúng tôi đang theo dõi có trạng thái di chuyển là **đi đúng chiều**.

## 4 Thực Nghiệm và Kết Quả

### 4.1 mAP (mean Average Precision)

Để tính được giá trị của mAP, chúng ta cần phải tính các giá trị liên quan.

#### 4.1.1 IOU (Intersection over Union)

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

#### Trong đó:

- Overlap: Diện tích giao giữa khung dự đoán và khung thực.
- Union: Tổng diện tích của khung dự đoán và khung thực (trừ đi phần giao).

Với mỗi bài toán thường có IOU threshold nhất định (nhận giá trị từ 0 đến 1). Nếu  $\text{IOU} > \text{threshold}$  thì prediction được đánh giá là tốt. Trong đa số bài toán threshold thường được đặt bằng 0,5.

Các tiêu chí đánh giá với IOU threshold:

- True Positive (TP): Đối tượng được nhận dạng đúng với  $\text{IOU} \geq \text{threshold}$ .
- False Positive (FP): Đối tượng được nhận dạng sai với  $\text{IOU} < \text{threshold}$ .
- False Negative (FN): Đối tượng không được nhận dạng.

#### 4.1.2 Precision và Recall

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

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

#### 4.1.3 Precision Recall Curve và Average precision (AP)

Precision và Recall thay đổi với mỗi Confidence threshold. Để quan sát tất cả các precision và recall tương ứng các threshold ta sử dụng Precision Recall Curve – đường đi qua tất các điểm với giá trị (recall, precision) ứng với từng threshold.

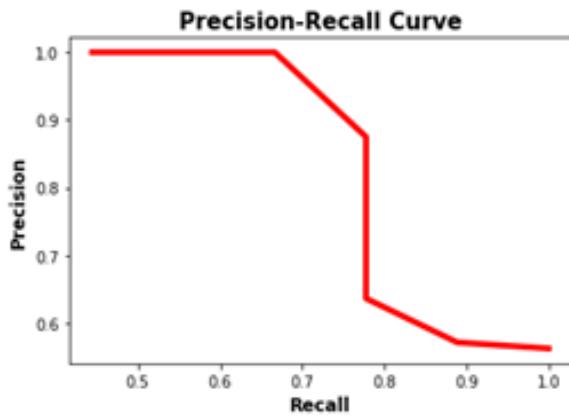


Figure 7: Precision-Recall Curve

AUC (Area Under the Curve ) - diện tích nằm dưới Curve giúp đánh giá model. Với Precision Recall Curve, Area Under the Curve (AUC) còn được gọi là Average precision (AP). AP được xác định bởi công thức:

$$AP = \sum_{k=1}^n (R_k - R_{k-1}) * P_k$$

Figure 8: Công thức tính AP

$R_k, P_k$  lần lượt là Recall và Precision ứng với threshold thứ k, n: số threshold

- AP lớn nếu vùng AUC này lớn, suy ra đường cong có xu hướng gần góc trên bên phải và có nghĩa là tại các threshold khác nhau thì Precision và Recall đều khá cao. Từ đó suy ra model tốt.
- AP nhỏ thì cả Precision và Recall đều khá thấp và model không tốt.

#### 4.1.4 mAP

Trong bài toán Object Detection nói chung hay YOLO nói riêng thì mAP được định nghĩa là trung bình cộng giá trị AP của tất cả các class. Trong đó:

$$mAP = \frac{1}{n} \sum_{i \in C} AP(i)$$

Figure 9: Công thức tính mAP

- C: là tập hợp tất cả các class
- n: là số class

mAP càng lớn thì đa số AP của từng class riêng biệt càng lớn dẫn đến model càng tốt. Từ đó việc train model sẽ có gắng train model có mAP lớn nhất có thể. Đây là lí do hoàn hảo sử dụng để mAP đánh giá model.

#### 4.2 Kết Quả Mô Hình YOLOv5

Kết quả sau khi tinh chỉnh mô hình YOLOv5 bằng bộ dữ liệu đã được đề cập ở phần 2.2:

Độ đo	Kết quả
Precision	81.48
Recall	81.45
mAP50	85.91
mAP50-95	48.32
fitness	52.08

Table 1: Kết quả tinh chỉnh mô hình YOLOv5

#### 4.3 Kết Quả Phát Hiện Lỗi Của Hai Phương Pháp

Chúng tôi so sánh dựa trên 1 video có độ dài 40 giây. so sánh giữa số lượng đối tượng đi ngược chiều do máy phát hiện và do con người phát hiện

	Thực tế	Module
Hướng tiếp cận bán tự động	6	12
Hướng tiếp cận tự động	6	19

Table 2: Kết quả nhận diện phương tiện đi ngược chiều

## 5 Hạn Chế và Hướng Phát Triển

### 5.1 Hạn Chế

Sau quá trình triển khai cả 2 ý tưởng, chúng tôi nhận thấy ý tưởng của chúng tôi vẫn còn tồn tại nhiều hạn chế:

- Mức độ chính xác vẫn còn thấp, vẫn còn nhiều đối tượng bị nhận diện nhầm
- Với ý tưởng bán tự động, độ chính xác và hiệu năng có thể được cải thiện nhưng tiềm năng trong việc ứng dụng là không lớn. Ngược lại, với ý tưởng tự động thì mặc dù độ chính xác và hiệu năng xử lý vẫn còn hạn chế nhưng tiềm năng phát triển lớn hơn.
- Hệ thống hiện nay chỉ phát hiện được với đối tượng vi phạm là xe máy cũng như chưa thể trích xuất thông tin đặc trưng của phương tiện và người điều khiển phương tiện.

### 5.2 Nhiệm Vụ Trong Tương Lai

Trong tương lai, chúng tôi tin rằng có thể cải thiện hiệu suất và độ chính xác bằng cách áp dụng các kỹ thuật tiên tiến hơn trong việc phát hiện và theo dõi đối tượng, cải thiện thuật toán hoạt động hiệu quả hơn với thiết bị có phần cứng hạn chế. Đồng thời, chúng tôi cũng cần phải xây dựng API và giao diện web thân thiện hơn với người dùng.

Chúng tôi cũng muôn phát triển đề tài theo hướng ứng dụng vào quá trình quản lý thành phố thông minh, xử lý theo thời gian thực dữ liệu được thu thập bởi những camera giao thông. Đồng thời cũng có thể phát triển để hệ thống để nhận diện nhiều hơn 1 lỗi của đa dạng các loại phương tiện.

## 6 Kết Luận

Tóm lại, trong đề tài này, chúng tôi đã thành công xây dựng hệ thống phát hiện đối tượng vi phạm lỗi đi ngược chiều trong khi tham gia giao thông. Dựa vào bộ dữ liệu từ camera giao thông đã được thu thập, chúng tôi đã tinh chỉnh mô hình YOLOv5 cho phù hợp với việc phát hiện người tham gia giao thông ở điều kiện giao thông đông đúc của Việt Nam. Sau đó sử dụng những phát hiện của YOLO kết hợp với thuật toán DeepSort nhằm theo dõi đối tượng trong đoạn video. Từ thông tin về vị trí của các đối tượng qua nhiều khung hình, chúng tôi có thể suy ra được hướng di chuyển của đối tượng và nhận diện đối tượng vi phạm lỗi đi ngược chiều. Tuy nhiên, đến thời điểm hiện tại, hệ thống vẫn tồn tại những vấn đề chưa được xử lý như không thể

hoạt động trên thiết bị có phần cứng hạn chế, độ chính xác vẫn chưa quá cao cũng như chỉ mới nhận diện được với đối tượng là người đi xe máy.

Trong tương lai, chúng tôi có thể mở rộng để hệ thống nhận diện được đa dạng lỗi vi phạm cũng như là đa dạng phương tiện vi phạm. Đồng thời cũng có thể kết hợp với các mô hình trích xuất thông tin từ biển số nhằm hỗ trợ cho quá trình giám sát giao thông công cộng trong việc quản lý thành phố thông minh.

## References

- [1] Trupti Mahendrakar et al. “Performance Study of YOLOv5 and Faster R-CNN for Autonomous Navigation around Non-Cooperative Targets”. In: *2022 IEEE Aerospace Conference (AERO)*. 2022, pp. 1–12. DOI: [10.1109/AERO53065.2022.9843537](https://doi.org/10.1109/AERO53065.2022.9843537).
- [2] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. “Simple online and realtime tracking with a deep association metric”. In: *2017 IEEE International Conference on Image Processing (ICIP)*. 2017, pp. 3645–3649. DOI: [10.1109/ICIP.2017.8296962](https://doi.org/10.1109/ICIP.2017.8296962).