

# Phát hiện lỗi đi sai làn đường của phương tiện giao thông dựa trên deep learning

■ TS. NGUYỄN DUY TRƯỜNG GIANG; THS. NGUYỄN HẠNH PHÚC

Trường Đại học Hàng hải Việt Nam

**TÓM TẮT:** Mật độ người và phương tiện tham gia giao thông trong đô thị ngày càng tăng đang gây áp lực rất lớn cho hạ tầng cũng như công tác quản lý của các cơ quan chức năng, đặc biệt là lĩnh vực giao thông. Cuộc cách mạng công nghiệp 4.0 với sự phát triển nhanh chóng của công nghệ xử lý hình ảnh dựa trên trí tuệ nhân tạo AI đã mang đến giải pháp mới trong công tác quản lý giao thông. Việc ứng dụng công nghệ AI Camera trong công tác quản lý giao thông hứa hẹn mang lại thay đổi tích cực, sâu sắc trong phương thức quản lý, nâng cao ý thức, văn hóa tham gia giao thông của toàn xã hội. Các trung tâm điều hành đô thị thông minh đang được các tỉnh, thành phố triển khai với việc tích hợp camera phục vụ giám sát, điều tiết, xử phạt giao thông. Các lỗi vi phạm như: phương tiện đè lên vạch kẻ đường (hoặc đi qua vạch kẻ đường khi đèn tín hiệu đã chuyển màu đỏ); đi không đúng phần đường hoặc làn đường quy định (đi sai làn) rất phổ biến. Bài báo đề xuất giải pháp ứng dụng các thuật toán deep learning phát hiện lỗi đi sai làn đường của phương tiện tham gia giao thông.

**TỪ KHÓA:** Học sâu, nhận dạng mẫu, phát hiện đối tượng, mạng nơ-ron tích hợp.

**ABSTRACT:** The increasing density of traffic participants (including people and vehicles) is putting great pressure on the infrastructure as well as on the traffic controlling and traffic management. The 4.0 technology revolution with the rapid development of AI-based image processing technology has brought a new solution to traffic management. The application of AI-Camera technology in traffic management promises to bring positive and profound changes in traffic management methods, raising awareness and improving behavior of traffic participants. Smart-city management centers are being deployed by provinces and cities with the integration of cameras for traffic monitoring, traffic regulation and sanctioning. Violations such as: vehicles moving over the road markings (or crossing the stop lines when the signal light has turned red), vehicles going on the wrong side of the road or lane (wrong-lane driving) are very common. The article proposes a solution, in

which deep learning algorithms are used to detect wrong-lane driving traffic participants.

**KEYWORDS:** Deep learning, pattern recognition, object detection, CNN network.

## 1. ĐẶT VẤN ĐỀ

Việc xử phạt vi phạm giao thông hiện tại đang theo phương pháp thủ công. CSGT quan sát (có thể có thêm thiết bị ghi hình làm bằng chứng). Việc này tốn nhiều nguồn lực, tuy nhiên không đem lại hiệu quả cao vì xử phạt không hết; gây mất an toàn khi người vi phạm chống đối hoặc bỏ chạy... Do đó, trong nghiên cứu này, nhóm tác giả tập trung vào đề xuất giải pháp phát hiện lỗi phương tiện đi sai làn đường dựa trên nhận dạng phương tiện giao thông bằng phương pháp học sâu CNN [1]. Nguyên lý hoạt động của hệ thống dựa trên một quá trình nhận diện phương tiện khi tham gia giao thông thông qua camera giám sát và công nghệ xử lý thông tin. Hiện nay trên thế giới cũng đã có nhiều nước quan tâm nghiên cứu và ứng dụng công nghệ này [1-6], tuy nhiên ở Việt Nam áp dụng vẫn chưa nhiều. Nội dung chính của nghiên cứu được trình bày chi tiết trong các mục 2,3, kết luận nghiên cứu trình bày trong mục 4.

## 2. MÔ HÌNH XỬ PHẠT GIAO THÔNG



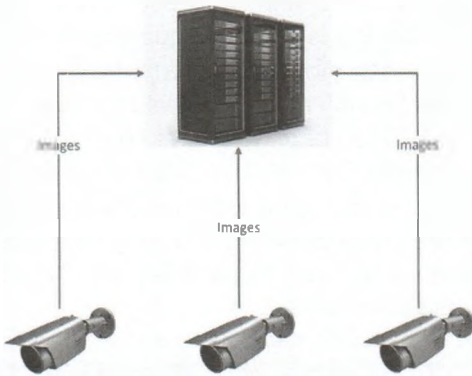
Hình 2.1: Phương tiện vi phạm lỗi đi sai làn

Hiện nay, trên thế giới, nhiều nước đã áp dụng AI vào quản lý xử lý vi phạm giao thông. Giải pháp này mang lại nhiều lợi ích cho xã hội, giúp nâng cao ý thức tuân thủ các quy định về tham gia giao thông.

Các hệ thống AI xử lý hình ảnh hiện nay được xây dựng theo hai phương án sau:

**\* Phương án xử lý tập trung:**

Tất cả các camera truyền hình ảnh trực tiếp về một server để xử lý.



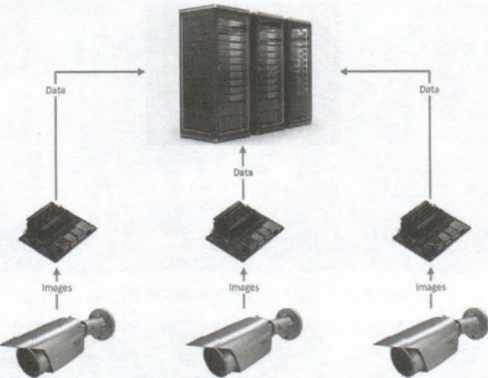
Hình 2.2: Mô hình xử lý tập trung

- Ưu điểm của phương án này: Giúp quản lý và bảo trì dễ dàng, nhiều tài nguyên tính toán được dùng để thực thi hiệu thuật toán phức tạp đem lại độ chính xác cao.

- Nhược điểm: Chi phí đầu tư ban đầu lớn, khó mở rộng số lượng camera, hệ thống đã được thiết kế cho số lượng camera nhất định, server xử lý là nút thắt cổ chai của hệ thống.

**\* Phương án xử lý tại biên:**

Hình ảnh của camera được xử lý tại chỗ rồi chỉ gửi kết quả về cho server.



Hình 2.3: Phương án xử lý tại biên

- Ưu điểm của phương án này: Chi phí đầu tư ban đầu nhỏ, có thể mở rộng số lượng camera trong hệ thống, server chỉ đóng vai trò tổng hợp kết quả ko còn là nút thắt cổ chai của hệ thống

- Nhược điểm: Công tác quản lý và bảo trì khó khăn do thiết bị xử lý nằm rải rác ở nhiều nơi, yêu cầu nhiều kĩ thuật vận hành cao do phần cứng ở các thiết bị biên có hạn.

Dựa trên những phân tích trên nhóm tác giả lựa chọn phương án xử lý tại biên để thực hiện. Về thuật toán xử lý, ngày nay đã có rất nhiều model deeplearning được phát triển để chạy trên những thiết bị tại biên, ở đây nhóm tác giả sử dụng dòng YOLO.

**2.1. Tổng quan hệ thống**

Hệ thống phát hiện lỗi vi phạm giao thông qua hình ảnh camera là một hệ thống xử lý hình ảnh theo thời gian thực yêu cầu cao về tốc độ và độ chính xác. Một hệ thống xử lý hình ảnh theo thời gian thực bao gồm các bước sau:

- Trích xuất thông tin: Xây dựng định nghĩa giúp máy tính xác định được những vật thể cần quan tâm trong một

bức hình, kĩ thuật học sâu (deeplearning) chủ yếu được áp dụng ở đây.

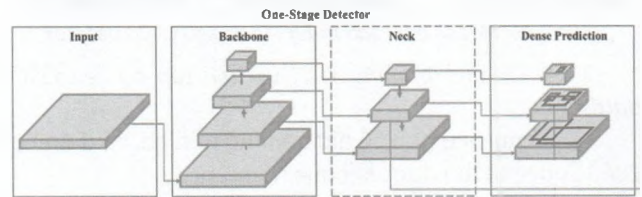
- Xử lý ngữ cảnh: Xây dựng định nghĩa trạng thái của đối tượng, dựa trên dữ liệu trích xuất ở bước 1 ta xác định xem trạng thái đối tượng đang như thế nào.

- Kết luận và đưa ra quyết định: Xây dựng mô-đun đưa ra kết luận (ví dụ: Kết luận một chiếc ô tô có lấn làn hay không).

**2.2. Mô hình phát hiện và nhận diện phương tiện**

Kỹ thuật học sâu (deeplearning) vượt trội hơn hẳn các phương pháp xử lý ảnh truyền thống. Ngày nay, bài toán xác định vị trí đối tượng (Object detection) đã phát triển thành một nhánh riêng trong deeplearning, mô hình nổi tiếng nhất trong bài toán này là các mô hình YOLO (You Only Look Once), nhóm tác giả lựa chọn model YOLOv4.

**2.2.1. Cấu trúc model YOLOv4**



Hình 2.4: Cấu trúc One-Stage Detector

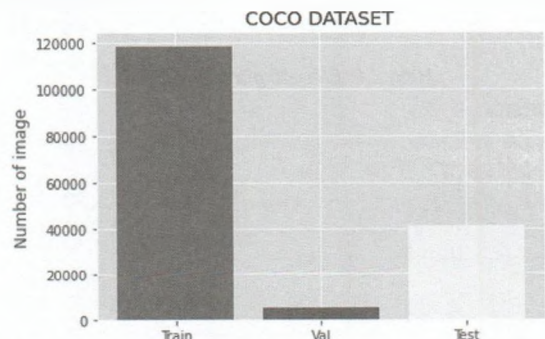
Model YOLOv4 thuộc kiểu One-Stage Detector, cấu trúc của model này bao gồm 3 thành phần sau: Backbone: CSPDarknet53, mạng này có độ chính xác cao trong bài toán object detection so với ResNet. Neck: Tổng hợp đặc trưng sẽ thực hiện trộn và kết hợp quá trình nhận dạng (YOLOv4 gọi là Dense prediction) với các Features Map đã được trích xuất từ backbone (CSPDarknet53). Head: Sử dụng Anchor box cho quá trình nhận dạng.

**2.2.2. Các kĩ thuật trong quá trình training**

Tăng cường dữ liệu: Giúp cải thiện kết quả inference mà không làm thay đổi tốc độ thực thi của model, những phương pháp thông thường: flip, rotation, random Contrast... Ta sử dụng thêm các phép biến đổi sau: Mixup dùng để trộn lẫn hai đối tượng vào nhau, cutout dùng để loại bỏ đi một phần đối tượng, CutMix dùng cắt ghép đối tượng...

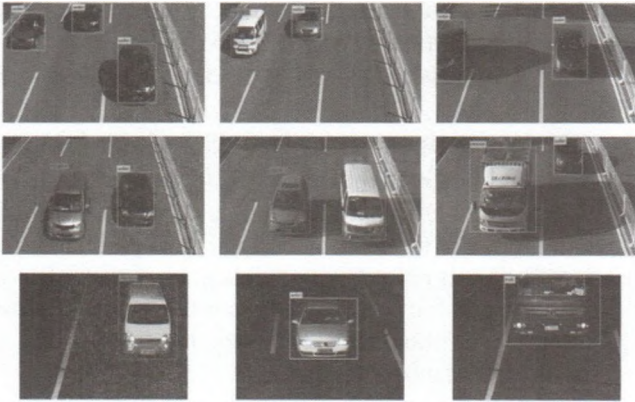
**2.2.3. Dữ liệu training và validation: COCO Dataset**

COCO dataset là tập dữ liệu rất nổi tiếng được công bố bởi Microsoft. Không chỉ sở hữu số lượng ảnh khổng lồ: 163.957 bức hình, mà số lượng đối tượng trong COCO Dataset cũng rất phong phú: 80 đối tượng (xe con, xe tải, xe máy, xe buýt...)



Hình 2.5: Thống kê số lượng ảnh trong các tập train, val, test của COCO Dataset

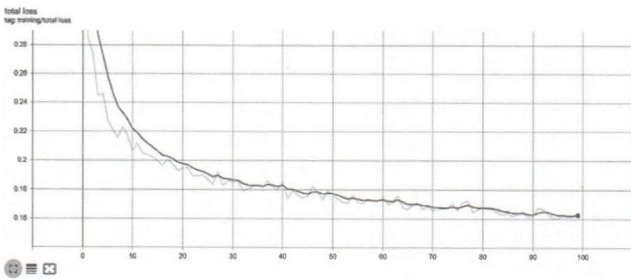
Tập dữ liệu COCO dataset được chia thành 3 phần: Train - 118.287 ảnh, Val - 5.000 ảnh, Test - 40.670 ảnh. Một số hình ảnh phương tiện giao thông trong COCO dataset.



Hình 2.6: Một số hình ảnh trong tập dữ liệu COCO Dataset

2.2.4. Kết quả training YOLOv4 trên tập dữ liệu COCO dataset

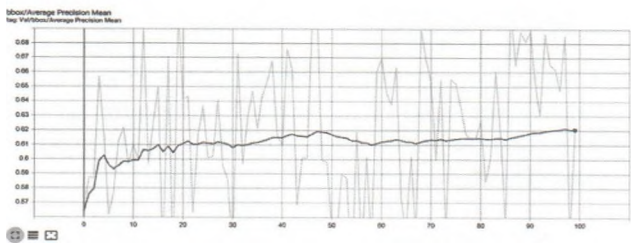
- Training: Sau hơn 50 giờ training trên hai card đồ họa 2060 Super, ta thu được kết quả như sau:



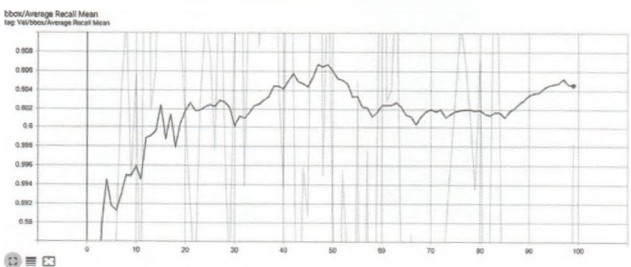
Hình 2.7: Biểu đồ giá trị loss

Quan sát biểu đồ biểu diễn chỉ số Loss, ta thấy sau 100 epochs training model học khá tốt trên tập dữ liệu COCO dataset. Giá trị loss giảm đều trong 90 epochs đầu, 10 epochs cuối giá trị loss có xu hướng nhích lên và đi ngang - đây là biểu hiện model đã fit với tập dữ liệu training.

- Validate:



Hình 2.8: Biểu đồ giá trị Precision

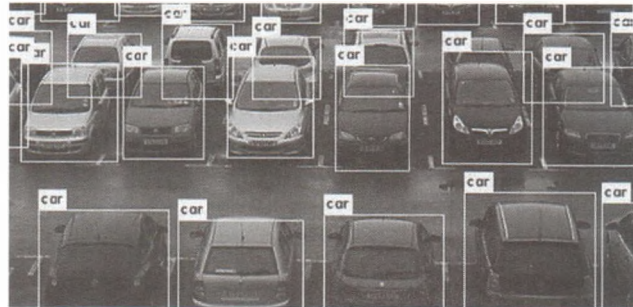


Hình 2.9: Biểu đồ giá trị Recall

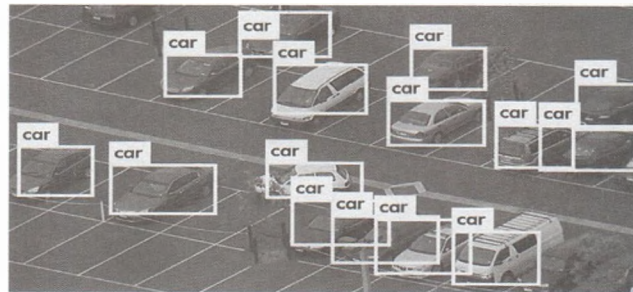
Tiếp đến là hai biểu đồ biểu diễn chỉ số Precision và Recall trên tập val, ta thấy kết quả không được cải thiện nhiều kể từ 50 epochs trở đi. Chỉ số Precision có xu hướng giảm ở giữa và lập lại đỉnh cũ ở epoch thứ 97, còn chỉ số Recall giảm dần. Đây là lúc ngừng quá trình training, vì nếu tiếp tục training thì model sẽ bị overfitting.

- Testing: So sánh model YOLOv4 trên tập dữ liệu Test với các model khác ta thấy YOLOv4 có kết quả kém hơn các model EfficientDet (D0 ~ D4), ASFF và CenterMask. Đây đều là những model thuộc kiểu Two-State, những model này được phát triển để hướng đến nâng cao độ chính xác và không quan tâm đến tốc độ inference. Còn về tốc độ YOLOv4 vượt trội hơn tất cả.

- Một vài hình ảnh inference:



Hình 2.10: Kết quả inference ở góc gần



Hình 2.11: Kết quả inference ở góc xa



Hình 2.12: Kết quả inference ở đường phố thực tế

2.3. Mô-đun theo dõi hướng di chuyển của phương tiện



Hình 2.13: Kết quả dự đoán hướng di chuyển của các phương tiện giao thông

Bài toán theo dõi hướng di chuyển của phương tiện hướng đến giải quyết vấn đề: “Đồng thời phát hiện và theo dõi tất cả đối tượng trong tầm nhìn, bao gồm cả các đối tượng mới xuất hiện và đã từng xuất hiện”.

Thuật toán nhóm tác giả đề xuất để xây dựng mô-đun theo dõi hướng di chuyển của phương tiện là Simple Online realtime Object Tracking (SORT).

SORT là một thuật toán thuộc dạng Tracking-by-detection (hay Detection based Tracking). Đặc điểm của bài toán này là tách phần Object detection thành bài toán riêng biệt nên đầu vào là kết quả của model detector. SORT sẽ tìm cách liên kết các bounding box ở mỗi frame và gán ID cho mỗi đối tượng. Do đó, quá trình xử lý mỗi frame sẽ diễn ra tự qua mỗi bước sau:

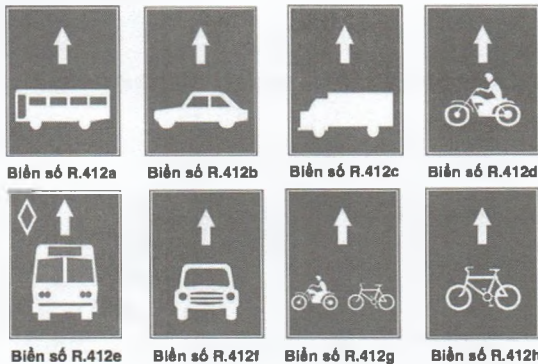
- Detect: Phát hiện vị trí các đối tượng trong frame (sử dụng detector).
- Predict: Dựa vào các frame trước đó dự đoán vị trí mới của các đối tượng.
- Associate: Liên kết các vị trí dự đoán với các vị trí detected được để gán ID tương ứng.

Hai thuật toán cốt lõi được sử dụng trong SORT là bộ lọc Kalman Filter và giải thuật Hungary.

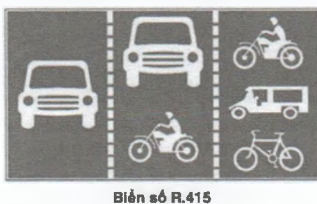
Áp dụng hai thuật toán trên, các bước xử lý của SORT như sau:

- Bước 1: Sử dụng bộ lọc Kalman Filter dự đoán các trạng thái (track) mới dựa trên các trạng thái (track) trong quá khứ.
- Bước 2: Kết hợp kết quả detection với các track vừa thu được, xây dựng ma trận chi phí. Chi phí ở đây được tính dựa vào giá trị IOU giữa các bounding box của detection và track.
- Bước 3: Áp dụng giải thuật Hungary trên ma trận chi phí vừa lập.
- Bước 4: Phân loại, xử lý các bounding box detection.
- Bước 5: Cập nhật bounding box của detection đã được liên kết với track bằng bộ lọc Kalman filter.

#### 2.4. Mô-đun tổng hợp thông tin và phát hiện lỗi vi phạm giao thông



Hình 2.14: Biển báo làn xe di chuyển riêng



Biển số R.415

Hình 2.15: Biển báo làn xe di chuyển hỗn hợp

- Coi làn đường là một đa giác khép kín ta xác định được tập hợp các điểm thuộc khu vực làn đường đó, thu hẹp lại sẽ được tập hợp các điểm nằm trên cạnh của đa giác tức lề của làn đường và gọi đây là tập hợp điểm xác định.

- Dựa vào công thức ta sẽ tính được hiệu khoảng cách tương đối giữa vật thể (ở đây là phương tiện tham gia giao thông) với các lề của làn đường cần đối chiếu.

- Mỗi một làn đường sẽ được đặt luật dựa theo vạch kẻ đường.

- Chia hướng di chuyển làm 4 hướng chính (các hướng còn lại sẽ kết hợp của 4 hướng này thành các tổ hợp di chuyển): sang phải, sang trái, xuống dưới, lên trên.

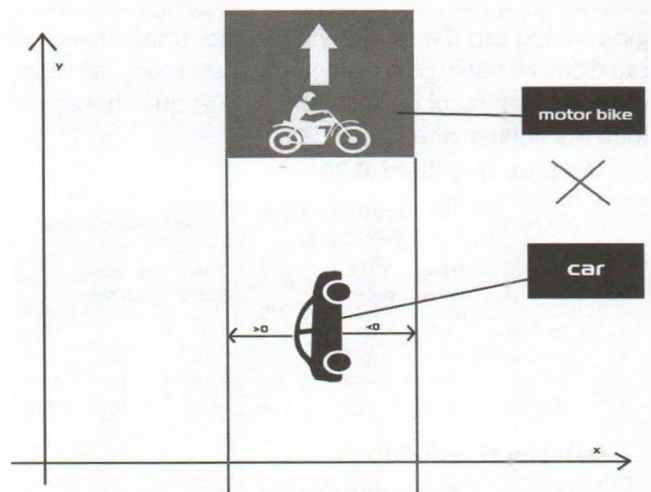
+ Để xác định lỗi lấn làn - sai làn: Ta xét khoảng cách tương đối giữa vật thể cần nhận diện lỗi với tập hợp điểm xác định, bằng cách xác định vật thể có vị trí tương đối tập các điểm xác định theo hướng di chuyển. Ở đây chia ra làm các trường hợp:

1) Xe chạm vạch: Hiệu vị trí tương đối của vật thể với tập điểm xác định của một vạch = 0.

2) Xe vượt vạch: Hiệu vị trí tương đối của vật thể với lề của làn nếu hướng di chuyển chính là sang trái hoặc từ trên xuống bằng 0 sau đó  $< 0 \Rightarrow$  xe vượt vạch, ngược lại với hiệu vị trí tương đối hướng di chuyển sang phải và lên trên ngày vượt qua ngưỡng 0 và dương thì xe vượt vạch. Trường hợp này chỉ áp dụng với vạch có luật trắng nét liền, vàng nét liền.

3) Xe đi sai làn (không vượt vạch): Bắt đầu từ điểm xuất phát của làn nếu hiệu vị trí tương đối của vật thể có kết quả như sau: lề trái (hoặc lề dưới)  $<$  vật thể  $<$  lề phải (hoặc trên), đồng thời tên phương tiện theo biển R412 hoặc R415 được gán luật cho làn này khác với nhãn phương tiện trả về thì đánh dấu phương tiện sai làn.

4) Xe đi sai làn (có vượt vạch): Kết hợp thuật toán b và c sẽ được thuật toán xử lý của lỗi này.



Hình 2.16: Xác định xe đi sai làn

+ Xác định đi ngược chiều: Xét vị trí tương đối của vật thể với điểm xuất phát của làn. Nếu vị trí tương đối giảm  $\Rightarrow$  xe đi ngược chiều, vị trí tương đối tăng  $\Rightarrow$  xe đi đúng chiều. Kết hợp với thuật toán xác định hướng di chuyển để làm tăng tính chính xác.



Hình 2.17: Minh họa xe đi sai làn thực tế

### 3. CÀI ĐẶT THỬ NGHIỆM MÔ HÌNH

Mô hình được cài đặt thử nghiệm bằng ngôn ngữ lập trình python và các thư viện hỗ trợ: Opencv, Pytorch, Tensorflow... Mô hình chạy trên máy tính ASUS core i5-9400F 6 luồng 2.9Ghz turbo 3.8Ghz, card đồ họa NVIDIA GeForce GTX 1660, 16G DDR4, hệ điều hành Win 10 - Win 11.

Kết quả thử nghiệm: Quá trình chạy thử nghiệm trên 2 video và 1 Rtp streaming camera cho thấy hệ thống detect và tracking các loại phương tiện khá tốt. Trong trường hợp nhiều ô tô có vị trí sát gần nhau vẫn phân biệt được từng đối tượng tỷ lệ bị dính đối tượng xảy ra rất thấp. Đối với các đối tượng ở xa, hệ thống vẫn đảm bảo nhận diện được loại phương tiện chính xác. Các đối tượng bị che chắn bởi vật thể như cây cối hay cột điện cũng được trích xuất. Xử lý việc vật thể bị phình to bởi bóng trong khung giờ có nắng cao đạt tỷ lệ chính xác cao. Trong khung giờ cao điểm, có nhiều phương tiện đang lưu thông, hệ thống vẫn phát hiện ra lỗi đề vạch, lấn làn. Kết quả chi tiết quá trình thử nghiệm như sau:

#### Detector threshold: 0,25

STT	Tên video	Số lượng phương tiện giao thông			Số lượng vi phạm		
		Trong video	YOLOv4 detected	Độ chính xác	Trong video	Hệ thống phát hiện	Độ chính xác
1	Video1.mp4	345	344	99%	2	2	100%
2	Video2.mp4	652	645	94%	18	19	98%
3	Cam RTSP	745	754	97%	16	16	100%
	Tổng số	2978	1679	<b>97%</b>	36	40	<b>98%</b>

#### Detector threshold: 0,3

STT	Tên video	Số lượng phương tiện giao thông			Số lượng vi phạm		
		Trong video	YOLOv4 detected	Độ chính xác	Trong video	Hệ thống phát hiện	Độ chính xác
1	Video1.mp4	345	443	93%	2	3	91%
2	Video2.mp4	652	747	95%	18	20	91%
3	Cam RTSP	745	759	97%	16	18	91%
	Tổng số	2978	1699	<b>95%</b>	36	41	<b>91%</b>

#### Detector threshold: 0,4

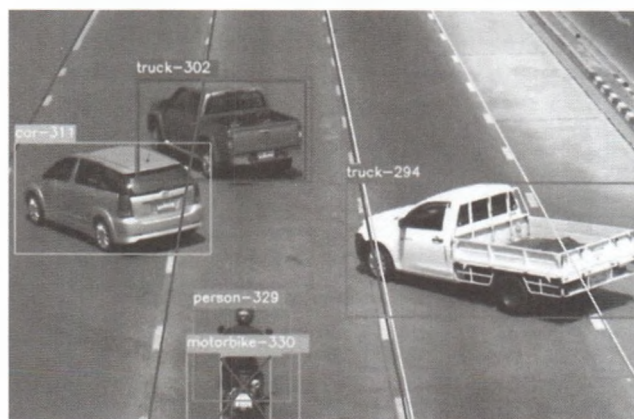
STT	Tên video	Số lượng phương tiện giao thông			Số lượng vi phạm		
		Trong video	YOLOv4 detected	Độ chính xác	Trong video	Hệ thống phát hiện	Độ chính xác
1	Video1.mp4	345	440	92%	2	4	80%
2	Video2.mp4	652	740	91%	18	24	88%
3	Cam RTSP	745	991	85%	16	24	82%
	Tổng số	2978	1671	<b>89%</b>	36	52	<b>83%</b>



Hình 3.1: Hệ thống nhận diện vật thể trả kết quả trực tiếp trên video



Hình 3.2: Nhận diện với khung giờ có ánh nắng cường độ cao



Hình 3.3: Hệ thống nhận diện xe máy đi vào làn đường chỉ dành cho xe 4 bánh

Hệ thống phát hiện lỗi vi phạm giao thông dựa trên thuật học sâu nên tham số ảnh hưởng nhiều đến độ chính

c của hệ thống là giá trị ngưỡng của YOLOv4 (Score threshold).

Ở giá trị ngưỡng thấp (Score threshold = 0,25). Hệ thống nhận diện tốt các vật thể, phương tiện được lọc chính xác với ngưỡng 97 - 98%.

Ở giá trị ngưỡng 0,3 (Score threshold = 0,3). Hệ thống phát hiện được nhiều phương tiện giao thông hơn nhưng nhiều phương tiện bị nhận diện trùng - giảm độ chính xác của YOLOv4 từ 98% xuống 95%, nhưng kết quả phát hiện vi phạm lỗi giao thông vẫn không thay đổi.

Ở giá trị ngưỡng cao hơn: 0,4 (Score threshold = 0,4). YOLOv4 nhiều vật thể bị nhận lên khiến kết quả bắt vi phạm bị sai số và trùng lặp nhiều.

Model YOLOv4 hoạt động tốt, nhận diện được các loại xe khác nhau, tiêu biểu: xe con - car, xe tải xe bán tải - truck, xe buýt - bus, xe máy nói chung - motorbike, xe đạp - bicycle. Hệ thống nhận diện được cả người tham gia giao thông đối với các phương tiện 2 bánh hay người đi trên vỉa hè.

#### 4. KẾT LUẬN

Thông qua nghiên cứu, nhóm tác giả đã xây dựng thành công mô hình thử nghiệm phát hiện vi phạm giao thông đi sai làn đường. Qua quá trình chạy thử nghiệm, ta thấy hệ thống chạy ổn định, xử lý nhanh và chính xác các trường hợp vi phạm giao thông đạt độ chính xác 91%.

Ngoài ra, sản phẩm của nghiên cứu cũng hướng đến xử lý phát lỗi vượt đèn đỏ. Một số hạn chế liên quan tới điều kiện môi trường ánh sáng thu thập ảnh mẫu cho huấn luyện sẽ cần được tìm hiểu kỹ hơn để tăng độ chính xác của quá trình nhận dạng.

**Lời cảm ơn:** Nghiên cứu này được tài trợ bởi Trường Đại học Hàng hải Việt Nam trong Đề tài mã số DT21-22.56.

#### Tài liệu tham khảo

- [1]. Cezanne Camacho (2018), *Convolutional Neural Networks*, ([https://cezannec.github.io/Convolutional\\_Neural\\_Networks/](https://cezannec.github.io/Convolutional_Neural_Networks/)).
- [2]. J. Yu, Y. Jiang, Z. Wang, Z. Cao and T. Huang (2016), *Initbox: An advanced object detection network*, in ACM MM.
- [3]. S. Ren, K. He, R. Girshick and J. Sun (2015), *Faster r-cnn: Towards realtime object detection with region proposal networks*, in NIPS, pp.91-99.
- [4]. Bewley A, Ge Z, Ott L, Ramos F, Upcroft B. (Aug. 2016), *Simple online and realtime tracking*, In: 2016 IEEE international conference on image processing (ICIP), pp.3464-8.
- [5]. Brooks J. COCO annotator (2019), <https://github.com/jsbroks/coco-annotator/>.
- [6]. Chu P, Ling H. FAMNet (2019), *Joint learning of feature, affinity and multidimensional assignment for online multiple object tracking*.

**Ngày nhận bài: 01/6/2022**

**Ngày chấp nhận đăng: 27/6/2022**

**Người phản biện: TS. Nguyễn Trung Đức  
TS. Trần Thị Hương**