XÁC ĐỊNH MẬT ĐỘ PHƯƠNG TIỆN GIAO THÔNG
DETERMINING TRAFFIC DENSITY

LÝ PHƯƠNG TRANG[1,a], TRẦN HOÀNG VIỆT[2],
NGUYỄN QUAN KHÁNH[3], TỪ VĂN TRỌNG[3]
[1]*Trường Trung cấp Kỹ thuật - Nghiệp vụ Kiên Giang*
[2]*Trường Đại học Cần Thơ*
[3]*Trường Cao đẳng Kiên Giang*
[a]*Tác giả liên hệ: lyphuongtrang@tcktnvkg.edu.vn*

Nhận bài(Received): 17/11/2024; Phản biện (Reviewed): 11/12/2024; Chấp nhận (Accepted): 03/02/2025

TÓM TẮT

An toàn giao thông là vấn đề được xã hội đặc biệt quan tâm, với trung bình 30-35 ca tử vong hàng ngày, chủ yếu trên đường bộ. Nguyên nhân chính là do số lượng phương tiện giao thông ngày càng tăng, chủ yếu là các phương tiện cá nhân. Nghiên cứu, khảo sát, thống kê các thời điểm, tuyến đường có phương tiện giao thông cao, đề xuất ra các công cụ cảnh báo hỗ trợ người dân, cảnh sát giao thông và nhà quản lý giảm thiểu ùn tắc giao thông. Ứng dụng mô hình học sâu (deep learning) để phát triển giải pháp và sử dụng Yolov8 nhận dạng đối tượng. Tập dữ liệu bao gồm 1030 ảnh chia làm 2 thư mục: train (824 ảnh) và test (206 ảnh) là ảnh các loại xe ô tô, xe máy, xe tải, xe buýt, xe đạp. Kết quả thử nghiệm cho thấy hiệu suất nhận dạng là 88,5%. Nghiên cứu này đề xuất mô hình phát hiện và nhận dạng phương tiện dựa trên mô hình yolov8, kết hợp với phân tích thống kê các loại phương tiện giao thông. Đây là một phương pháp chính xác, dễ dàng và chi phí thấp để phát hiện các phương tiện giao thông.

Từ khóa: Yolov8, thị giác máy tính, nhận dạng đối tượng, phương tiện giao thông.

ABSTRACT

Traffic safety is an issue of particular concern to society, with an average of 30-35 fatalities daily, mainly on the roads. The main reason is the increasing number of vehicles, primarily personal vehicles. Research, survey, and statistically analyze the times and routes with high traffic, and propose warning tools to assist citizens, traffic police, and managers in reducing traffic congestion. Applying deep learning models to develop solutions and using Yolov8 for object detection. The dataset includes 1030 images divided into 2 folders: train (824 images) and test (206 images), which are images of various types of vehicles such as cars, motorcycles, trucks, buses, and bicycles. The experimental results show a recognition performance of 88.5%. This study proposes a vehicle detection and recognition model based on the yolov8 model, combined with statistical analysis of different types of vehicles. This is an accurate, easy, and cost-effective method for detecting vehicles.

Keywords: *Yolov8, Computer vision, Object recognition, Transportation.*

1. GIỚI THIỆU

Vấn đề ùn tắc giao thông xảy ra phức tạp tại các thành phố lớn, đô thị đông dân do lưu lượng phương tiện tăng đột biến, trong khi quy hoạch giao thông, tổ chức giao thông, hạ tầng giao thông chưa đáp ứng yêu cầu, gây tắc nghẽn trên diện rộng tại các đô thị lớn và trên các tuyến cao tốc, quốc lộ trọng điểm, gây ra những thiệt hại không nhỏ về kinh tế, ảnh hưởng xấu đến sức khỏe và đời sống của nhân dân, tác động không tốt đến môi trường du lịch, thu hút đầu tư nước ngoài và hình ảnh của Việt Nam đối với bạn bè quốc tế.

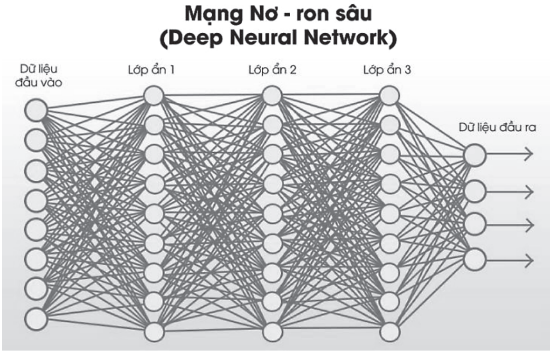
Hiện tại, các phương pháp truyền thống như thu thập dữ liệu thủ công hoặc sử dụng cảm biến tầng hạ tầng đang bộc lộ nhiều chế độ về chi phí, thời gian và độ chính xác. Ngược lại, các giải pháp dựa trên công nghệ hiện đại như phân tích hình ảnh và video, cùng với sự hỗ trợ của trí tuệ nhân tạo (AI), đang mở ra một hướng đi đầy hứa hẹn. Những công nghệ này không chỉ cung cấp thông tin phù hợp và chính xác mà còn có khả năng tự động hóa, giảm chi phí vận hành và sử dụng hiệu quả nguồn năng lượng.

Ứng dụng mô hình YOLOv8 trong phát hiện phương tiện giao thông là ô tô [1]. Kết quả cho thấy, mô hình YOLOv8 nhận diện đối tượng xuất hiện trên ảnh tốt hơn so với đối tượng xuất hiện trên video hay camera. Thuật toán ước lượng mật độ giao thông bằng cách sử dụng phương pháp học sâu và tiến hành tính toán diện tích mặt đường [2]. Trích xuất thông tin độ sâu ở Kinect và phân loại đặc tính của phôi dựa trên YOLO [3]. Kết quả thực nghiệm phản ánh sự hiệu quả và chính xác của mô hình. Nghiên cứu các phiên bản YOLOv8 và YOLO-NAS trong phát hiện biển số xe [16] đánh giá hiệu suất toàn diện của các kiến trúc YOLOv8 và YOLO-NAS để phát hiện biển số xe. Đối với các ứng dụng có cơ sở hạ tầng tốt và yêu cầu có độ chính xác cao thì YOLO-NAS-S là một mô hình phù hợp với độ chính xác là 83,92%, Precision là 0,9125; Recall là 0,9125; và F1 score là 0,9125. Đối với cơ sở hạ tầng kém phát triển hơn và yêu cầu về tốc độ thì có thể sử dụng YOLOv8n với số lượng tham số ít hơn nhưng độ chính xác lại khá ổn 81,4%; Precision là 0,9625; Recall 0,8415 và F1 score là 0,8979.

Tập trung nghiên cứu, tìm hiểu mạng nơron tích chập (CNN) kết hợp với mô hình nhận đối tượng (YOLO), đồng thời thực hiện cài đặt mô hình huấn luyện về nhận dạng hình ảnh nhằm đếm số lượng phương tiện xe lưu thông. Từ đó, hệ thống sẽ đưa ra những số liệu thống kê báo cáo tình hình mật độ giao thông theo từng thời điểm và từng đối tượng cụ thể.

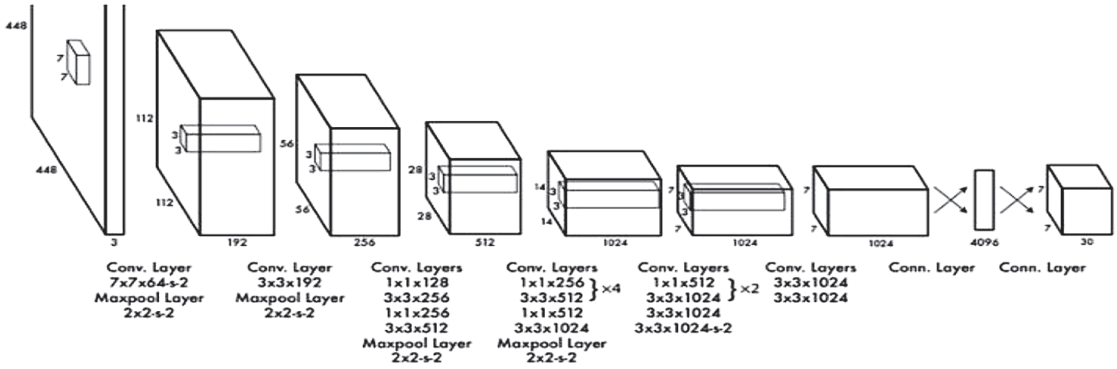
2. NỘI DUNG

Học sâu (Deep learning) là một bước tiến dài của Machine Learning, có thể xử lý dữ liệu theo cách tương tự như một bộ não con người, cho phép máy có thể tự đào tạo chính mình. Deep learning đã chứng minh rằng nó có thể đạt được hiệu suất cao hơn trên hầu hết các tác vụ trong thị giác máy tính [4]. Cho phép các mô hình tính toán gồm nhiều tầng xử lý để học biểu diễn dữ liệu với nhiều mức trừu tượng khác nhau.



Hình 1. Cách học sâu (Deep learning) hoạt động - Ảnh: Internet

YOLO là một mô hình mạng CNN cho việc phát hiện, nhận dạng, phân loại đối tượng. YOLO được tạo ra từ việc kết hợp giữa lớp tích chập và lớp kết nối đầy đủ [5]. Trong đó các lớp tích chập sẽ trích xuất ra các đặc trưng của ảnh, còn lớp kết nối đầy đủ sẽ dự đoán ra xác suất các đặc trưng đó và tọa độ của đối tượng.



Hình 2. Kiến trúc mạng YOLO - Ảnh Internet

Hình 2, sử dụng 24 lớp tích chập theo sau là 2 lớp được kết nối đầy đủ. Một số lớp tích chập sử dụng xen kẽ các lớp giảm 1 × 1 để giảm độ sâu của bản đồ các đặc trưng. Đối với lớp chập cuối cùng, nó tạo ra một tensor có hình dạng (7, 7, 1024). Tensor này sau đó được làm phẳng. Sử dụng 2 lớp được kết nối đầy đủ như một hình thức hồi quy tuyến tính, nó cho ra các $7 \times 7 \times 30$ tham số và sau đó định dạng thành (7, 7, 30), tức là 2 dự đoán hộp ranh giới cho mỗi vị trí.

Có rất nhiều phiên bản YOLO. Nhưng YOLOv8 đã mang lại nhiều điểm tích cực so với phiên bản tiền nhiệm, như phát hiện không dùng anchor, giới thiệu lớp tích chập C3 và tăng cường mosaic. Do đó, YOLOv8 có tỷ lệ chính xác cao hơn và khả năng nhận dạng các vật thể nhỏ tốt hơn.

2.1. Xây dựng cơ sở dữ liệu ảnh (Dataset)

Sử dụng, thu thập các ảnh xe làm bộ cơ sở dữ liệu phục vụ cho quá trình tính toán và huấn luyện. Dữ liệu đầu vào của hệ thống chính là ảnh chụp tự động được thực hiện trong nhiều môi trường có điều kiện khác nhau nhằm đảm bảo độ chân thực cho dữ liệu thực tế, những hình ảnh này được chụp từ nhiều góc độ khác nhau để tăng tính đa dạng.

Tập dữ liệu có 1.030 ảnh, trong đó 963 ảnh chụp và 67 ảnh sưu tập từ các nguồn khác. Trong đó có 289 ảnh ô tô (ảnh độc lập: 34, ảnh có cùng với các loại phương tiện khác hoặc vật thể khác: 255); 390 ảnh xe máy, 154 ảnh xe tải, 68 ảnh xe đạp, 129 ảnh xe khách.

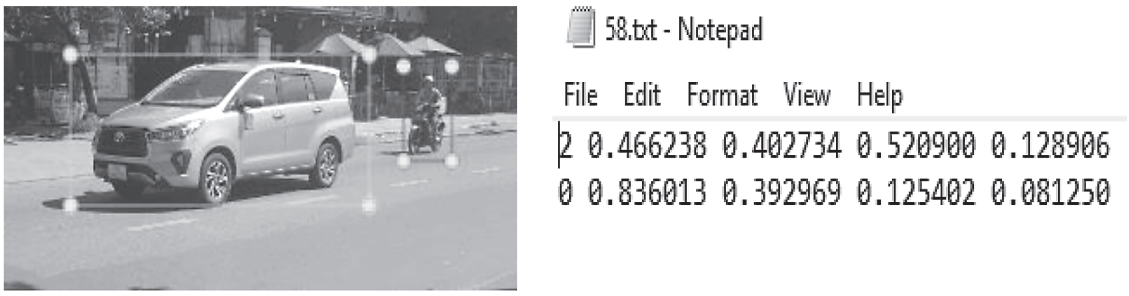
Tập dữ liệu ảnh này được phân chia làm 02 loại theo tỷ lệ 80% và 20% như sau: số ảnh huấn luyện là 824 ảnh và kiểm chứng là 206 ảnh.

2.2. Giai đoạn gán nhãn

Đây là một bước quan trọng trong quá trình xây dựng và huấn luyện các mô hình trí tuệ nhân tạo, đặc biệt trong các lĩnh vực như phân loại hình ảnh, phát hiện đối tượng, nhận dạng văn bản và nhiều ứng dụng khác.

Labelimg là một công cụ mã nguồn mở được sử dụng để gán nhãn (annotation) cho hình ảnh, giúp chuẩn

bị dữ liệu đầu vào cho các mô hình học máy, đặc biệt là trong các bài toán nhận diện đối tượng (object detection). Một số hình ảnh không thể đào tạo ngay lập tức được, vì có một số định dạng và kích thước khác nhau nên cần chuyển sang cùng một định dạng. Sau đó, hình ảnh được gán nhãn bằng công cụ Labelimg và xuất nhãn sang định dạng YOLO với một tập tin *.txt cho mỗi hình ảnh.



Hình 3. Hình ảnh sau khi được gán nhãn

Hình 3, mỗi ảnh sẽ có một tệp *.txt tương ứng, chứa thông tin về các đối tượng trong ảnh. Mỗi dòng trong tệp 58.txt biểu thị một đối tượng và bao gồm: [class_id] [x_center] [y_center] [width] [height]

- class_id: ID của đối tượng lớp (bắt đầu từ 0).
- x_center, y_center: tọa độ tâm của đối tượng, được chuẩn hóa theo kích thước ảnh (giá trị từ 0 đến 1).
- width, height: Kích thước của đối tượng được chuẩn hóa theo kích thước ảnh.

2.3. Giai đoạn huấn luyện

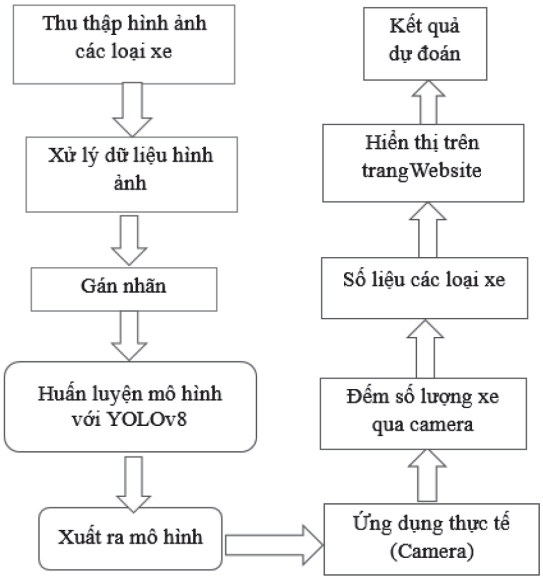
Trong giai đoạn này, chúng tôi tập trung vào các bước sau:

- Bước 1: Thu thập hình ảnh các phương tiện giao thông. Sau khi thu thập đủ số lượng hình ảnh, tiến hành thực hiện lấy mẫu phục vụ cho quá trình học.

- Bước 2: Gán nhãn các phương tiện giao thông: Sử dụng công cụ gán nhãn đã trình bày ở phần trên, nhằm xác định các loại phương tiện giao thông.

- Bước 3: Huấn luyện mô hình với Deep Learning: Sử dụng phương pháp học sâu: Mạng CNN, YOLOv8 để huấn luyện.

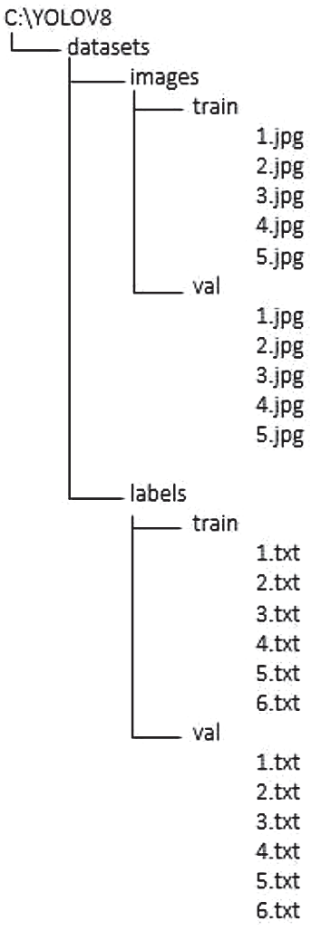
- Bước 4: Xây dựng mô hình phát hiện và nhận dạng các phương tiện lưu thông: Quá trình huấn luyện và nhận dạng mô hình được thực hiện như sau (*Hình 4*)



Hình 4. Mô hình huấn luyện và nhận dạng

Phương thức huấn luyện:

- Sau khi đã có ảnh và nhãn, tiến hành tổ chức thư mục datasets chứa ảnh và nhãn dùng cho việc huấn luyện mô hình.



Hình 5. Tổ chức thư mục chứa ảnh và nhãn

- Cấu hình file data.yaml chứa đường dẫn đến thư mục ảnh, nhãn và thông tin đối tượng. Tạo các tập tin dữ liệu đầu vào cho mô hình YOLO từ dữ liệu đã chuẩn bị. Mỗi tập tin dữ liệu đầu vào bao gồm đường dẫn tới ảnh và các thông tin về bounding boxes và nhãn tương ứng.

```
names:
- car
- motor
- bus
- truck
- bicycle
nc: 5
test: test/images
train: train/images
val: valid/images
```

Hình 6. Cấu hình file data.yaml

- Để huấn luyện mô hình, sử dụng câu lệnh sau:

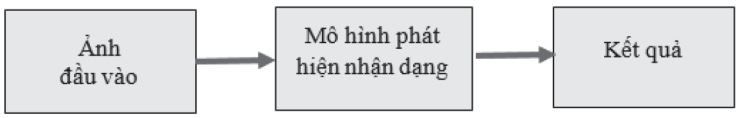
python train.py --img 640 --batch 24 --epochs 300 --data data.yaml --weights yolov8.pt

Trong đó:

- img: Kích thước ảnh
- batch: Số ảnh dùng để huấn luyện trong mỗi lượt
- epochs: Số lượt huấn luyện cho tất cả các ảnh trong tập dữ liệu huấn luyện
- data: Đường dẫn đến file cấu hình của tập dữ liệu
- weights: Đường dẫn đến file weight chứa độ liên kết giữa các neuron

2.4. Giai đoạn nhận dạng

Từ ảnh chụp ngẫu nhiên các loại xe được đưa vào hệ thống nhận dạng và ghi nhận các loại xe. Hệ thống sẽ đưa ra được kết quả chính xác từng loại xe.



Hình 7. Kiến trúc nhận dạng phương tiện giao thông

Sau khi đã huấn luyện mô hình ta tiến hành nhận dạng dựa trên câu lệnh sau:

python detect.py --source data/images --weights best.pt --conf 0.5

Trong đó:

- source: Đường dẫn đến thư mục ảnh muốn nhận dạng.
- weights: Mô hình đã được huấn luyện.
- conf: Độ tin cậy.

Kết quả huấn luyện mô hình có thể được đánh giá dựa trên một số metrics phù hợp với bài toán nhận dạng như mô hình YOLO với kiến trúc YOLOv8. Metrics thường được sử dụng để chỉ các độ đo hoặc chỉ số để đánh giá hiệu suất và độ chính xác của mô hình.

- Precision (độ chính xác dương tính): Đo lường tỷ lệ dự đoán đúng các mẫu dương tính (positive) so với tổng số mẫu được dự đoán là dương tính. Để hiểu precision, ta cần định nghĩa một số khái niệm:

+ True Positives (TP): Số lượng các mẫu dương tính được mô hình dự đoán đúng.

+ False Positives (FP): Số lượng các mẫu âm tính được mô hình dự đoán là dương tính sai.

Precision [6] được tính bằng tỷ lệ giữa True Positives và tổng số dự đoán dương tính (bao gồm True Positives và False Positives):

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FP}} \quad (1)$$

- mAP (Mean Average Precision): Đo lường độ chính xác trung bình của các bounding box được dự đoán bằng cách tính toán diện tích dưới đường cong Precision-Recall (PR curve). mAP thường được tính cho các ngưỡng (thresholds) khác nhau để đánh giá đa dạng và linh hoạt của mô hình.

- recall (Độ bao phủ) [23]: được tính bằng cách chia số lượng dự đoán đúng thuộc một lớp (true positive) cho tổng số lượng thực tế thuộc lớp đó. Công thức tính recall như sau:

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FN}} \quad (2)$$

+ True Positive (TP) là số lượng các mẫu thuộc lớp dự đoán đúng.

+ False Negative (FN) là số lượng các mẫu thuộc lớp thực tế nhưng bị dự đoán sai.

- Loss function (hàm mất mát): Là metrics được sử dụng trong quá trình huấn luyện để đo lường sai lệch giữa các dự đoán và giá trị thực tế. Một hàm loss function tốt sẽ giúp mô hình học tốt hơn và cải thiện hiệu suất đối với nhiệm vụ nhận dạng đối tượng. [7] Trong mô hình YOLO, thông thường sử dụng các hàm mất mát như sum squared error để đo lường hiệu suất huấn luyện. Hàm mất mát có dạng:

$$\begin{aligned}\mathcal{L} = {} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} \left[(\sqrt{\omega}_i - \sqrt{\hat{\omega}_i})^2 + (\sqrt{h}_i - \sqrt{\hat{h}_i})^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} \mathbb{I}_{ij}^{noobj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \end{aligned} \quad (3)$$

- F1-score là một metric tổng hợp sử dụng để kết hợp độ chính xác (precision) và độ bao phủ (recall) của mô hình phân loại. Nó được tính bằng cách lấy trung bình điều hòa (harmonic mean) của precision và recall:

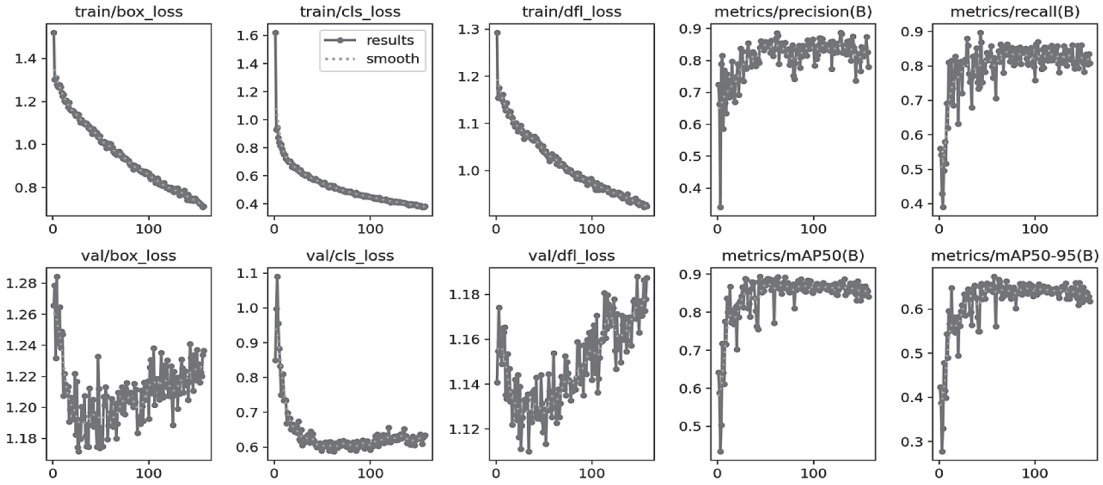
$$F1_score = \frac{2 * (precision * recall)}{precision + recall} \quad (4)$$

+ *Precision* là tỷ lệ giữa số lượng dự đoán đúng và tổng số dự đoán tích lũy cho một lớp cụ thể.

+ *Recall* là tỷ lệ giữa số lượng dự đoán đúng và tổng số đối tượng thực tế cho một lớp cụ thể.

2.5. Kết quả thực nghiệm

Trong bài báo này, chúng tôi đã sử dụng 1030 ảnh của phương tiện lưu thông làm ảnh gốc.



Hình 8. Kết quả dự đoán mô hình

Sau khi đã huấn luyện mô hình ta tiến hành nhận dạng bằng lệnh sau:

```
python detect.py --source data/images
    --weights best.pt --conf 0.5
```

Trong đó:

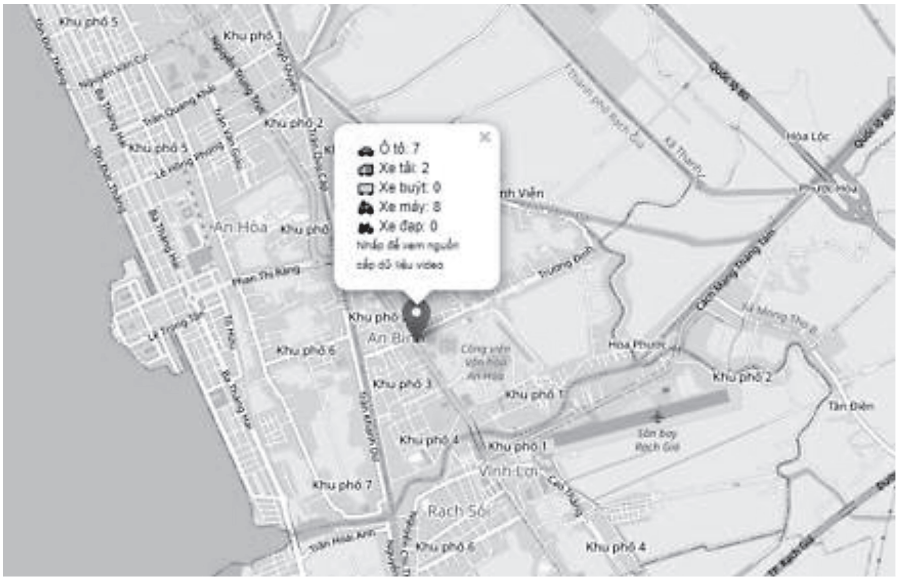
- source: Đường dẫn đến thư mục chứa ảnh muốn nhận dạng.
- weights: Mô hình đã được huấn luyện trước đó.
- conf: Độ tin cậy.

Hình 9. Kết quả nhận dạng các phương tiện giao thông

Việc nhận diện và phân loại phương tiện giao thông có ý nghĩa quan trọng quy hoạch đô thị và quản lý giao thông, ngày càng được nghiên cứu sâu trên thế giới. Giao thông tại Việt Nam có mật độ lưu thông lớn và phương tiện chủ yếu là xe máy nên vấn đề nhận diện phương tiện với độ chính xác cao càng trở nên phức tạp.

Số lượng các phương tiện lưu thông trên đường là một hoạt động quan trọng trong công việc quản lý và điều hành giao thông đô thị. Công việc này cung cấp dữ liệu chính xác về mật độ và chuyển luồng của các tiện ích đa phương tiện, giúp các nhà quản lý đưa ra quyết định tối ưu hóa hệ thống thông tin.

Để đếm các loại xe trong video, cần sử dụng video và hình ảnh thư viện xử lý như OpenCV và các mô hình học để phát hiện và đếm phương tiện giao thông. Sử dụng 10 video để thống kế số lượng các loại xe đang chạy từ đó cho biết đưa ra kết quả các phương tiện giao thông đang chạy trên các tuyến đường.



Hình 10. Thống kê số lượng xe trên đường tại thành phố Rạch Giá

Hình 11. Bảng thống kê các loại xe đang chạy

3. KẾT LUẬN

Nghiên cứu này xác định mật độ phương tiện giao thông qua hình ảnh và video bằng cách áp dụng phương pháp học sâu dựa trên mô hình YOLOv8. Kết quả, xây dựng mô hình nhận dạng các loại phương tiện lưu thông trên đường với độ chính xác cao 88,5% bằng phương pháp học sâu và đếm số lượng phương tiện giao thông tại một thời điểm trên một tuyến đường và phân loại phương tiện cụ thể (xe ô tô, xe máy, xe tải, xe đạp, xe buýt). Kết quả đạt được đã chứng minh tiềm năng và tính khả thi của mô hình này trong các

công việc nhận dạng, phân loại và thống kê số lượng các loại xe khác nhau.

Hướng phát triển, thu thập thêm nhiều ảnh của các phương tiện giao thông, mở rộng nhận dạng thêm nhiều loại xe chỉ cần thu thập hình ảnh và đào tạo lại mô hình, hiển thị các loại phương tiện giao thông, từ đó cung cấp số liệu chính xác về từng loại xe di chuyển. Ngoài ra, nghiên cứu còn mở ra hướng phát triển các hệ thống tự động hóa trong quản lý giao thông, bao gồm thống kê mật độ giao thông, dự đoán xu hướng lưu lượng xe và phát hiện tình huống khác.

TÀI LIỆU THAM KHẢO

[1] Phạm Thị Thanh Thủy, Tạp chí Khoa học Tài nguyên và Môi trường về Nghiên cứu và ứng dụng mô hình YOLOv8 trong phát hiện và thống kê số lượng vị trí còn trống trong bãi đỗ ô tô, DOI:https://doi.org/10.63064/khtnmt.2024.606.

[2] Vũ Lê Quỳnh Phương, Kỷ yếu hội thảo khoa học quốc gia 2019, CNTT và ứng dụng trong các lĩnh vực.

[3] Bùi Thanh Lâm, Nguyễn Đức Quang, Nguyễn Văn Trường, Phan Đình Hiếu, Vũ Tuấn Anh, Lưu Vũ Hải, Tạp chí Khoa học công nghệ về Nghiên cứu ứng dụng thuật toán học sâu kết hợp cảm biến Kinect trong phân loại vật thể, Tập 59 - Số 4 (8/2023) DOI: https://doi.org/10.57001/huih5804.2023.142

[4] https://www.elcom.com.vn/deep-learning-la-gi-ung-dung-cua-hoc-sau-trong-thuc-tien-1666582364

[5] G. Plastiras, C. Kyrkou, and T. Theocharides, "Efficient convnet-based object detection for unmanned aerial vehicles by selective tile processing," ACM Int. Conf. Proceeding Ser., 2018, doi: 10.1145/3243394.3243692.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-December, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[7] V. H. Sơn, N. V. Hiến, and V. Q. Vịnh, "Tóm tắt:," pp. 55–61, 2022.

[8] Đ. H. Quang, L. H. Minh, and T. D. Nguyên, "Nhận dạng khuôn mặt trong video bằng mạng nơ ron tích chập," vol. 62, no. 1, pp. 8–12, 2020

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–14, 2015.

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-December, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-December, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[13] Đoàn Phước Miền, Tống Thị Kim Quy, Trần Thế Vũ, Huỳnh Hữu Hưng, Kỷ yếu Hội nghị KHCN Quốc gia lần thứ XI về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR); Hà Nội, ngày 09-10/8/2018.

[14] Phan Anh Cang, Nguyễn Thị Mỹ Nga, Phan Thượng Cang, Kỷ yếu Hội nghị KHCN Quốc gia lần thứ XI về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR); Hà Nội, ngày 09-10/8/2018 DOI: 10.15625/vap.2018.00011

[15] D. A. Mulia, S. Safitri, and G. P. K. Negara, "YOLOv8 and Faster R-CNN Performance Evaluation with Super-resolution in License Plate Recognition," International Journal of Computing and Digital Systems, 2024, doi: 10.12785/ijcds/160129.

[16] Đặng Thị Dung, Hà Lê Ngọc Dung, Trương Lê Chương, Thái Chí Hào, Trần Văn Phúc, TNU Journal of Science and Technology về Nghiên cứu các phiên bản YOLOv8 và YOLO-NAS trong phát hiện biển số xe, DOI: https://doi.org/10.34238/tnu-jst.10336

[17] Trần Xuân Kiên, Đinh Xuân Trường, Phạm Tuấn Thành, Lê Minh Dương, Tạp chí Khoa học công nghệ về Ứng dụng theo dõi phương tiện, giám sát giao thông trong thời gian thực DOI: https://doi.org/10.57001/huih5804.2023.103