Nhận diên biển số xe, phương tiện và phát hiện sai phạm vượt đèn đỏ

Nguyễn Tất Thắng, Phạm Thị Huyền Trang, Hà Minh Chiến, Phùng Xuân Đức

Khoa Công Nghệ Thông Tin, Đại học Đại Nam, Hà Nội, Việt Nam

Giảng viên hướng dẫn: ThS. Nguyễn Thái Khánh, ThS. Lê Trung Hiếu

***Tóm tắt nội dung*—Trong nghiên cứu này, chúng tôi trình bày phương pháp nhận diện phương tiện giao thông, biển số xe và phát hiện vi phạm tín hiệu đèn giao thông sử dụng mô hình YOLOv8. Hai mô hình YOLOv8s được huấn luyện riêng biệt: một mô hình cho nhận diện biển số xe đạt mAP50 = 0.907 và mAP50-95 = 0.553 sau 25 epoch, và một mô hình cho nhận diện phương tiện đạt mAP50 = 0.946 và mAP50-95 = 0.878 sau 25 epoch. Tập dữ liệu được thu thập từ nhiều nguồn thực tế và được gán nhãn bằng RoboFlow. Quá trình huấn luyện được thực hiện với kích thước ảnh 640x640 và tối ưu hoá bằng SGD. Kết quả cho thấy mô hình nhận diện phương tiện đạt hiệu suất cao hơn so với mô hình nhận diện biển số xe, đặc biệt là ở các chỉ số mAP50-95 và precision. Phương pháp này có thể áp dụng hiệu quả trong các hệ thống giám sát giao thông thông minh.**

***Index Terms*—YOLOv8; Phát hiện phương tiện; Nhận diện biển số; Tập dữ liệu tuỳ chỉnh.**

I. Giới thiệu

A. Tổng quan về vấn đề

Với tốc độ đô thị hóa nhanh chóng và sự gia tăng mạnh mẽ về số lượng phương tiện giao thông tại các thành phố lớn của Việt Nam, đặc biệt là tại Hà Nội, hệ thống giao thông đang phải đối mặt với nhiều thách thức như tắc nghẽn, tai nạn và vi phạm luật giao thông. Theo thống kê của Cục Cảnh sát giao thông, hành vi vượt đèn đỏ là một trong những nguyên nhân hàng đầu gây tai nạn nghiêm trọng, chiếm khoảng 35% số vụ tai nạn tại các giao lộ. Các phương pháp giám sát truyền thống dựa trên nhân lực không những tốn kém về chi phí mà còn bộc lộ nhiều hạn chế về khả năng xử lý lượng lớn thông tin cùng một lúc, đặc biệt trong điều kiện giao thông phức tạp và mật độ cao.

B. Tình hình nghiên cứu liên quan

Những năm gần đây, với sự phát triển mạnh mẽ của trí tuệ nhân tạo và thị giác máy tính, nhiều nghiên cứu đã được thực hiện nhằm tự động hóa quá trình giám sát giao thông. Các phương pháp truyền thống sử dụng kỹ thuật xử lý ảnh cổ điển như phát hiện cạnh, phân đoạn hình ảnh và đối sánh mẫu đã dần được thay thế bởi các mô hình học sâu. Các kiến trúc như R-CNN, SSD và YOLO đã chứng minh hiệu quả vượt trội trong các bài toán phát hiện đối tượng. Đặc biệt, kiến trúc YOLO (You Only Look Once) với ưu điểm về tốc độ xử lý thời gian thực đã trở thành lựa chọn phổ biến cho các ứng dụng giám sát giao thông.

Tuy nhiên, nhiều nghiên cứu trước đây thường tập trung vào việc giải quyết riêng lẻ các bài toán như phát hiện phương tiện hoặc nhận diện biển số, ít có nghiên cứu tích hợp đồng thời nhiều nhiệm vụ để phát hiện các hành vi vi phạm phức tạp như vượt đèn đỏ. Hơn nữa, hầu hết các mô hình được huấn luyện trên các tập dữ liệu chuẩn quốc tế, có ít nghiên cứu sử dụng dữ liệu đặc thù cho điều kiện giao thông Việt Nam với đặc điểm là mật độ cao và đa dạng về loại phương tiện.

C. Đề xuất của nghiên cứu

Trong nghiên cứu này, chúng tôi đề xuất một hệ thống tích hợp dựa trên kiến trúc YOLOv8 - phiên bản mới nhất và hiệu quả nhất của họ mô hình YOLO, để giải quyết đồng thời bài toán nhận diện phương tiện giao thông, biển số xe và phát hiện hành vi vượt đèn đỏ. Thay vì sử dụng một mô hình đa nhiệm đơn lẻ, chúng tôi áp dụng chiến lược huấn luyện hai mô hình YOLOv8s chuyên biệt:

Mô hình nhận diện phương tiện: Tập trung vào việc phát hiện và phân loại các loại phương tiện giao thông phổ biến tại Việt Nam như xe máy, ô tô con, xe tải, xe buýt. Mô hình đạt hiệu suất cao với mAP50 = 0.946 và mAP50-95 = 0.878 sau 25 epoch huấn luyện.

Mô hình nhận diện biển số xe: Chuyên biệt hóa cho việc phát hiện vị trí và đọc biển số xe trong các điều kiện thực tế khác nhau. Mô hình này đạt chỉ số mAP50 = 0.907 và mAP50-95 = 0.553 sau 25 epoch huấn luyện.

Kết quả từ hai mô hình này được tích hợp cùng với thông tin ngữ cảnh từ các vùng quan tâm (ROI) tại các giao lộ có đèn tín hiệu để phát hiện và ghi nhận các trường hợp vượt đèn đỏ. Phương pháp này không chỉ nâng cao độ chính xác trong việc phát hiện vi phạm mà còn tối ưu hóa hiệu suất tính toán so với việc sử dụng một mô hình đơn lẻ phức tạp.

D. Quy trình huấn luyện và đặc điểm dữ liệu

Quá trình huấn luyện hai mô hình YOLOv8s được thực hiện dựa trên tập dữ liệu tùy chỉnh được thu thập từ nhiều nguồn đa dạng, bao gồm camera giao thông tại các giao lộ chính ở Hà Nội, hình ảnh từ thiết bị bay không người lái (drone), và các nguồn dữ liệu công khai. Tập dữ liệu này được gán nhãn cẩn thận bằng công cụ RoboFlow, với sự phân chia hợp lý thành các tập huấn luyện (70%), xác thực (20%) và kiểm tra (10%).

Mô hình nhận diện phương tiện được huấn luyện với tốc độ học (learning rate) từ 0.00014443 đến 5.51056e-05, sử dụng

phương pháp tối ưu SGD. Quá trình huấn luyện cho thấy sự cải thiện đáng kể về các chỉ số hiệu suất qua từng epoch, với độ chính xác (precision) tăng từ 0.66825 lên 0.91447 và độ phủ (recall) tăng từ 0.59878 lên 0.90277 sau 25 epoch.

Tương tự, mô hình nhận diện biển số xe được huấn luyện với tốc độ học từ 0.00024 đến 9.92e-05. Mặc dù mô hình này đạt chỉ số mAP50 khá cao (0.907) nhưng chỉ số mAP50-95 thấp hơn đáng kể (0.553) so với mô hình nhận diện phương tiện, phản ánh thách thức lớn hơn trong việc xác định chính xác vị trí và nội dung của biển số xe trong các điều kiện thực tế.

E. Đóng góp chính của nghiên cứu

Nghiên cứu này có bốn đóng góp chính:

1) Xây dựng tập dữ liệu đặc thù: Tạo ra tập dữ liệu tùy chỉnh với đặc điểm phản ánh tính đa dạng và phức tạp của giao thông Việt Nam, bao gồm nhiều loại phương tiện, điều kiện ánh sáng và thời tiết khác nhau, cũng như các kiểu biển số xe đặc trưng của Việt Nam.
2) Tối ưu hóa kiến trúc YOLOv8: Điều chỉnh và tối ưu hóa hai mô hình YOLOv8s chuyên biệt, mỗi mô hình tập trung vào một nhiệm vụ cụ thể, giúp nâng cao hiệu suất tổng thể so với các phương pháp sử dụng một mô hình đơn lẻ.
3) Phát triển thuật toán tích hợp: Xây dựng thuật toán kết hợp thông tin từ hai mô hình riêng biệt và dữ liệu ngữ cảnh (vị trí đèn tín hiệu, vạch dừng) để phát hiện chính xác các trường hợp vượt đèn đỏ.
4) Đánh giá hiệu suất toàn diện: Thực hiện phân tích chi tiết về hiệu suất của hệ thống trong các điều kiện thực tế khác nhau, xác định các yếu tố ảnh hưởng và đề xuất các phương pháp cải thiện.

F. So sánh hiệu suất và phân tích

Dựa trên kết quả huấn luyện, có thể thấy rõ sự khác biệt đáng kể giữa hiệu suất của hai mô hình. Mô hình nhận diện phương tiện đạt hiệu suất cao hơn với mAP50-95 = 0.878 so với chỉ số 0.553 của mô hình nhận diện biển số xe. Điều này có thể được giải thích bởi các thách thức đặc thù trong bài toán nhận diện biển số xe:

Biển số xe thường có kích thước nhỏ và chiếm tỷ lệ diện tích nhỏ trong hình ảnh, đặc biệt là biển số xe máy

Sự đa dạng về kiểu dáng, màu sắc và vị trí gắn biển số trên các loại phương tiện khác nhau

Ảnh hưởng của các điều kiện môi trường như ánh sáng chói, bóng đổ, mưa, bụi bẩn hoặc che khuất một phần

Biến dạng hình học do góc chụp camera không lý tưởng Mặc dù vậy, chỉ số mAP50 = 0.907 của mô hình nhận diện biển số xe vẫn đủ cao để sử dụng hiệu quả trong các ứng dụng thực tế, đặc biệt khi kết hợp với thông tin bổ sung từ mô hình nhận diện phương tiện.

G. Ứng dụng thực tiễn và triển vọng

Hệ thống được phát triển trong nghiên cứu này có thể được ứng dụng hiệu quả trong các hệ thống giám sát giao thông thông minh, đặc biệt tại các giao lộ có tín hiệu đèn. Ngoài việc phát hiện vi phạm vượt đèn đỏ, hệ thống còn có thể mở rộng để phát hiện các vi phạm khác như đi ngược chiều, dừng đỗ sai quy định hoặc vượt quá tốc độ cho phép. Hơn nữa, dữ liệu thu thập được từ hệ thống có thể được sử dụng để phân tích mẫu giao thông, tối ưu hóa thời gian đèn tín hiệu và nâng cao an toàn giao thông.

Trong tương lai, chúng tôi dự định mở rộng nghiên cứu theo các hướng:

Tích hợp các kỹ thuật theo dõi đối tượng (object tracking) để tăng cường khả năng phát hiện vi phạm trong các đoạn video dài

Phát triển các mô hình nhẹ hơn có thể triển khai trên các thiết bị tính toán cạnh (edge computing)

Nghiên cứu khả năng nhận diện biển số xe trong điều kiện ánh sáng yếu hoặc ban đêm

Xây dựng hệ thống cảnh báo sớm cho các tình huống có nguy cơ vi phạm cao

Với những đóng góp và kết quả đạt được, nghiên cứu này không chỉ cung cấp một giải pháp hiệu quả cho bài toán giám sát giao thông mà còn đặt nền móng cho việc phát triển các hệ thống thành phố thông minh trong tương lai. Tôi sẽ viết lại file LaTeX dựa trên tài liệu bạn đã cung cấp, không đưa code vào, nhưng vẫn giữ nội dung liên quan đến việc huấn luyện mô hình.

II. Nghiên cứu liên quan

A. TrafficDetection

Dự án "TrafficDetection"sử dụng mô hình Yolov8 để phát hiện và theo dõi giao thông trong video, nhằm nhận diện các phương tiện như ô tô, xe bus, xe tải, và xe máy. Dự án sử dụng các kỹ thuật theo dõi như BơT-SORT và ByteTrack để xác định vị trí và di chuyển của các phương tiện trong các khung hình video. Đặc biệt, mô hình hoạt động trên video Full HD với tốc độ 25 FPS, giúp đánh giá hiệu suất và khả năng phát hiện của Yolov8 trong môi trường thực tế.

B. License plate recognition and red light violation detection YOLOV8

Dự án YOLOv8 là phiên bản mới của mô hình phát hiện đối tượng nổi tiếng YOLO, được xây dựng bằng PyTorch. YOLOv8 cải thiện hiệu suất so với các phiên bản trước về cả tốc độ và độ chính xác, hỗ trợ các tác vụ như phát hiện đối tượng, phân loại ảnh và phân đoạn. Nó còn hỗ trợ chuyển đổi mô hình sang nhiều định dạng như ONNX, CoreML, và TFLite, giúp dễ dàng triển khai trên các nền tảng khác nhau. Bạn có thể dễ dàng cài đặt thông qua pip

III. Cơ sở lý thuyết

A. Nền tảng khoa học của mô hình YOLO

YOLO (You Only Look Once) thuộc họ các mạng nơ-ron tích chập (Convolutional Neural Networks - CNNs) được thiết kế cho bài toán nhận diện đối tượng thời gian thực. Khác với các phương pháp truyền thống sử dụng cơ chế hai giai đoạn (vùng đề xuất và phân loại), YOLO áp dụng mô hình hồi quy duy nhất (single regression model) để dự đoán đồng thời các

hộp giới hạn (bounding boxes) và xác suất lớp đối tượng từ ảnh đầu vào.

YOLOv8 là phiên bản cải tiến với kiến trúc mạng được tối ưu hóa, bao gồm:
Backbone: Sử dụng CSPDarknet với khối tích chập sâu (Deep Cross-Stage Partial Connections) để trích xuất các đặc trưng đa cấp
Neck: Mạng FPN (Feature Pyramid Network) cải tiến kết hợp với PAN (Path Aggregation Network) cho phép truyền thông tin đặc trưng giữa các tầng
Head: Áp dụng kiến trúc đa đầu ra (multi-output) với các tầng dự đoán chuyên biệt cho các tác vụ:
Nhận diện đối tượng (Object Detection)
Phân đoạn ảnh (Instance Segmentation)
Phân loại ảnh (Image Classification)

Mô hình YOLOv8 cải thiện khả năng nhận diện đối tượng thông qua hàm mất mát tổng quát:

$$\mathcal{L} = \lambda_{coord} \cdot \mathcal{L}_{box} + \lambda_{obj} \cdot \mathcal{L}_{obj} + \lambda_{cls} \cdot \mathcal{L}_{cls} + \lambda_{mask} \cdot \mathcal{L}_{mask} \tag{1}$$

Trong đó: $\mathcal{L}_{box}$ là hàm mất mát cho dự đoán hộp giới hạn, thường sử dụng CIoU (Complete IoU):

$$\mathcal{L}_{box} = 1 - \text{IoU} + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{2}$$

với $\rho(b, b^{gt})$ là khoảng cách Euclid giữa tâm của hộp dự đoán và hộp ground truth, c là đường chéo của hình chữ nhật nhỏ nhất bao quanh cả hai hộp, α là tham số cân bằng, và v đo lường tỷ lệ khung hình.

$\mathcal{L}_{obj}$ là hàm mất mát cho việc xác định xác suất hiện diện của đối tượng, sử dụng Binary Cross Entropy:

$$\mathcal{L}_{obj} = -\sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[p_i \log(\hat{p}_i) + (1 - p_i) \log(1 - \hat{p}_i) \right] \tag{3}$$

$\mathcal{L}_{cls}$ là hàm mất mát cho dự đoán lớp đối tượng, sử dụng Cross Entropy:

$$\mathcal{L}_{cls} = -\sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \sum_{c \in \text{classes}} p_i(c) \log(\hat{p}_i(c)) \tag{4}$$

$\mathcal{L}_{mask}$ là hàm mất mát cho phân đoạn ảnh, thường sử dụng Dice Loss:

$$\mathcal{L}_{mask} = 1 - \frac{2 \sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2} \tag{5}$$

λ_{coord}, λ_{obj}, λ_{cls}, λ_{mask} là các hệ số cân bằng giữa các thành phần mất mát.

1) Phương pháp đánh giá và trực quan hóa nâng cao: Ngoài các độ đo cơ bản, mô hình còn được đánh giá thông qua:

- Confusion Matrix: Ma trận nhầm lẫn giữa các loại phương tiện:

$$C_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,k}} \tag{6}$$

Trong đó $n_{i,j}$ là số lượng phương tiện thuộc lớp i được dự đoán là lớp j.

- Phân tích đường cong ROC: Mối quan hệ giữa tỷ lệ dương tính thật (TPR) và tỷ lệ dương tính giả (FPR):

$$\text{TPR} = \frac{TP}{TP + FN} \tag{7}$$

$$\text{FPR} = \frac{FP}{FP + TN} \tag{8}$$

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR}^{-1}(t)) dt \tag{9}$$

- Phân tích lỗi dự đoán: Xác định các trường hợp khó khăn và phân loại lỗi:

$$E_{type} = \begin{cases} \text{FP}_{\text{localization}}, & \text{if IoU} \in (0, \tau) \\ \text{FP}_{\text{classification}}, & \text{if class } \neq \text{class}_{gt} \\ \text{FP}_{\text{background}}, & \text{if no matching} \\ \text{FN}, & \text{if ground truth missed} \end{cases} \tag{10}$$

Trong đó τ là ngưỡng IoU được chọn (thường là 0.5).

B. Tích hợp kiến thức lĩnh vực vào mô hình nhận diện

Để nâng cao hiệu suất trong ứng dụng giám sát giao thông, mô hình được tích hợp thêm kiến thức lĩnh vực (domain knowledge):
Ràng buộc không gian: Áp dụng kiến thức về vị trí tương đối của các đối tượng trong không gian giao thông:

$$P(O_i|O_j) = \frac{P(O_j|O_i)P(O_i)}{P(O_j)} \tag{11}$$

$$P(O_j|O_i) = f_{spatial}(d_{ij}, \theta_{ij}) \tag{12}$$

Trong đó d_{ij} là khoảng cách giữa đối tượng i và j, θ_{ij} là góc giữa hai đối tượng.

Ràng buộc thời gian: Tích hợp thông tin về động học của phương tiện sử dụng bộ lọc Kalman:

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k \tag{13}$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{14}$$

Trong đó $\mathbf{x}_k$ là trạng thái tại thời điểm k, $\mathbf{F}_k$ là ma trận chuyển trạng thái, $\mathbf{B}_k$ là ma trận điều khiển, $\mathbf{u}_k$ là vector điều khiển, $\mathbf{w}_k$ là nhiễu quá trình, $\mathbf{z}_k$ là phép đo, $\mathbf{H}_k$ là ma trận phép đo, và $\mathbf{v}_k$ là nhiễu đo.

Ràng buộc vật lý: Áp dụng các định luật vật lý về chuyển động của phương tiện:

$$\vec{a}_i = f_{physics}(\vec{v}_i, m_i, \vec{F}_{ext}) \tag{15}$$

$$\vec{v}_{i,t+\Delta t} = \vec{v}_{i,t} + \vec{a}_i \Delta t \tag{16}$$

$$\vec{x}_{i,t+\Delta t} = \vec{x}_{i,t} + \vec{v}_{i,t} \Delta t + \frac{1}{2} \vec{a}_i \Delta t^2 \tag{17}$$

Trong đó $\vec{a}_i$, $\vec{v}_i$, và $\vec{x}_i$ lần lượt là gia tốc, vận tốc, và vị trí của phương tiện i, m_i là khối lượng, $\vec{F}_{ext}$ là các lực bên ngoài.

C. Mô hình toán học cho việc xác định hành vi vượt đèn đỏ

Quá trình phát hiện hành vi vượt đèn đỏ được mô hình hóa như một bài toán xác suất có điều kiện:

$$P(V|O,T,S) = \frac{P(O,T,S|V)P(V)}{P(O,T,S)} \tag{18}$$

Trong đó:
V là biến ngẫu nhiên thể hiện trạng thái vi phạm ($V = 1$ là vi phạm, $V = 0$ là không vi phạm)
O là thông tin về phương tiện được nhận diện (loại phương tiện, kích thước, màu sắc...)
T là trạng thái của đèn giao thông ($T = 0$ là đèn xanh, $T = 1$ là đèn đỏ)
S là vị trí của phương tiện so với vạch dừng Xác suất vi phạm có thể được mô hình hóa chi tiết hơn:

$$P(V=1|O,T=1,S) = \begin{cases} \alpha, & \text{if } S < S_{stop} \text{ and } v > 0 \\ \beta, & \text{if } S_{stop} < S < S_{intersection} \text{ and } v \\ \gamma, & \text{if } S > S_{intersection} \text{ and } v > 0 \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

Trong đó S_{stop} là vị trí vạch dừng, $S_{intersection}$ là vị trí giao lộ, v là vận tốc phương tiện, và α, β, γ là các tham số xác suất thực nghiệm với $\alpha < \beta < \gamma$. Áp dụng chuỗi Markov để mô hình hóa diễn biến chuyển động của phương tiện:

$$P(S_t|S_{t-1}, S_{t-2}, ..., S_0) = P(S_t|S_{t-1}) \tag{20}$$

$$P(S_t|S_{t-1}) = \mathcal{N}(S_{t-1} + v_{t-1}\Delta t, \sigma_S^2) \tag{21}$$

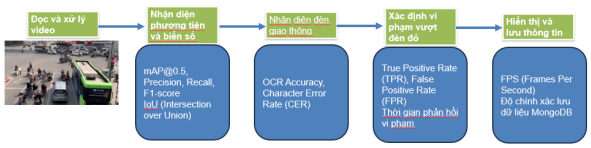
Trong đó $\mathcal{N}(\mu, \sigma^2)$ là phân phối chuẩn với trung bình μ và phương sai σ^2.

Mô hình Hidden Markov được áp dụng để ước lượng trạng thái vi phạm từ các quan sát có nhiễu:

$$P(V_{1:T}|Z_{1:T}) \propto P(V_1) \prod_{t=1}^{T} P(Z_t|V_t) \prod_{t=2}^{T} P(V_t|V_{t-1}) \tag{22}$$

Trong đó $V_{1:T}$ là chuỗi trạng thái vi phạm, $Z_{1:T}$ là chuỗi quan sát, $P(Z_t|V_t)$ là xác suất quan sát, và $P(V_t|V_{t-1})$ là xác suất chuyển trạng thái.

IV. Phương pháp thực hiện

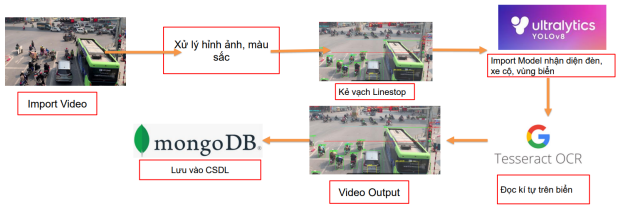


Hình 1: Hệ thống

A. Kiến trúc tổng quan hệ thống phát hiện vi phạm giao thông

Hệ thống phát hiện vi phạm giao thông từ video được thiết kế để tự động hóa quá trình giám sát và phát hiện các phương tiện vượt đèn đỏ. Kiến trúc của hệ thống bao gồm các thành phần chính sau:
Mô-đun tiền xử lý video: Trích xuất và chuẩn hóa các khung hình từ nguồn video đầu vào.
Mô-đun nhận diện phương tiện: Sử dụng mô hình học sâu để phát hiện và phân loại các phương tiện giao thông.
Mô-đun nhận diện biển số: Định vị và trích xuất thông tin từ biển số xe.
Mô-đun nhận diện đèn giao thông: Xác định vị trí và trạng thái của đèn giao thông.
Mô-đun phân tích vi phạm: Tích hợp thông tin từ các mô-đun khác để xác định hành vi vi phạm.
Mô-đun lưu trữ và báo cáo: Lưu trữ thông tin vi phạm và tạo báo cáo tổng hợp.

B. Quy trình xử lý video và phát hiện vi phạm



Hình 2: Quá trình thực hiện hệ thống

1) Tiếp nhận và xử lý video đầu vào: Hệ thống tiếp nhận tệp video trong các định dạng phổ biến (MP4, AVI, MOV) và phân tích từng khung hình theo thời gian thực. Quá trình tiền xử lý bao gồm:

$$I_t = V(t) \tag{23}$$

$$I'_t = f_{preprocess}(I_t) \tag{24}$$

Trong đó:
$V(t)$ là hàm trích xuất khung hình tại thời điểm t từ video
I_t là khung hình gốc tại thời điểm t
$f_{preprocess}$ là hàm tiền xử lý
I'_t là khung hình sau khi tiền xử lý

2) Nhận diện phương tiện giao thông: Mô hình YOLOv8 được áp dụng để phát hiện và phân loại các phương tiện trong khung hình. Mô hình này có khả năng phát hiện đồng thời nhiều đối tượng với độ chính xác cao và tốc độ xử lý nhanh, phù hợp cho ứng dụng thời gian thực.

$$\{B_1, B_2, ..., B_n\} = f_{YOLO}(I'_t) \tag{25}$$

Trong đó:
f_{YOLO} là hàm nhận diện đối tượng sử dụng mô hình YOLOv8
$B_i = \{x_i, y_i, w_i, h_i, c_i, p_i\}$ là thông tin về phương tiện thứ i được phát hiện
(x_i, y_i) là tọa độ tâm của hộp giới hạn

(w_i, h_i) là chiều rộng và chiều cao của hộp giới hạn
c_i là lớp của phương tiện (xe máy, ô tô, xe buýt, xe tải)
p_i là độ tin cậy của dự đoán

3) Nhận diện và trích xuất biển số xe: Quá trình nhận diện biển số xe được thực hiện thông qua hai bước chính:

1) Định vị vùng biển số: Sử dụng mô hình nhận diện chuyên biệt để xác định vị trí biển số trong hình ảnh phương tiện:

$$LP_i = f_{LP\_detect}(B_i) \tag{26}$$

2) Nhận dạng ký tự trên biển số: Áp dụng các kỹ thuật xử lý ảnh và OCR:

$$LP_i' = f_{enhance}(LP_i) \tag{27}$$

$$Plate\_ID_i = f_{OCR}(LP_i') \tag{28}$$

Trong đó:
f_{LP_detect} là hàm phát hiện vùng biển số
LP_i là vùng ảnh chứa biển số của phương tiện thứ i
$f_{enhance}$ bao gồm các phép biến đổi như chuyển đổi sang ảnh grayscale, điều chỉnh độ tương phản, lọc nhiễu, v.v.
LP_i' là ảnh biển số sau khi được tiền xử lý
f_{OCR} là hàm nhận dạng ký tự quang học
$Plate\_ID_i$ là chuỗi ký tự trên biển số

4) Nhận diện trạng thái đèn giao thông: Việc xác định trạng thái đèn giao thông được thực hiện như sau:

$$TL_{region} = f_{TL\_detect}(I_t') \tag{29}$$

$$TL_{HSV} = f_{RGB2HSV}(TL_{region}) \tag{30}$$

$$TL_{state} = f_{color\_analysis}(TL_{HSV}) \tag{31}$$

Trong đó: f_{TL_detect} là hàm phát hiện vùng chứa đèn giao thông TL_{region} là vùng ảnh chứa đèn giao thông $f_{RGB2HSV}$ là hàm chuyển đổi không gian màu từ RGB sang HSV TL_{HSV} là ảnh đèn giao thông trong không gian màu HSV $f_{color\_analysis}$ là hàm phân tích phân bố màu sắc $TL_{state} \in \{red, yellow, green\}$ là trạng thái của đèn giao thông Phân tích trạng thái đèn dựa trên phân phối màu trong không gian HSV giúp tăng độ chính xác trong điều kiện ánh sáng khác nhau và giảm nhiễu môi trường.

5) Phát hiện hành vi vượt đèn đỏ: Để xác định hành vi vượt đèn đỏ, hệ thống cần phát hiện vạch dừng và theo dõi chuyển động của phương tiện qua vạch này khi đèn đỏ:

$$SL_{pos} = f_{stopline\_detect}(I_t') \tag{32}$$

$$V_i(t) = \{x_i(t), y_i(t)\} \tag{33}$$

$$violation_i(t) = \begin{cases} 1, & \text{nếu } TL_{state} = red \text{ và } y_i(t) > SL_{pos} \text{ và } y_i(t - \Delta t) < SL_{pos} \\ 0, & \text{ngược lại} \end{cases} \tag{34}$$

Trong đó:
$f_{stopline\_detect}$ là hàm phát hiện vị trí vạch dừng
SL_{pos} là tọa độ của vạch dừng
$V_i(t)$ là vectơ vị trí của phương tiện thứ i tại thời điểm t
Δt là khoảng thời gian giữa các khung hình liên tiếp
$violation_i(t)$ là biến nhị phân chỉ trạng thái vi phạm của phương tiện thứ i

C. Lưu trữ và tổng hợp thông tin vi phạm

Khi phát hiện vi phạm, hệ thống sẽ lưu trữ thông tin theo mô hình sau:

$$VR_i = \{Plate\_ID_i, t_i, I_{evidence}, TL_{state}, Location\} \tag{35}$$

Trong đó: VR_i là bản ghi vi phạm thứ i
$Plate\_ID_i$ là biển số phương tiện vi phạm
t_i là thời điểm vi phạm
$I_{evidence}$ là khung hình làm bằng chứng vi phạm
$Location$ là thông tin vị trí diễn ra vi phạm

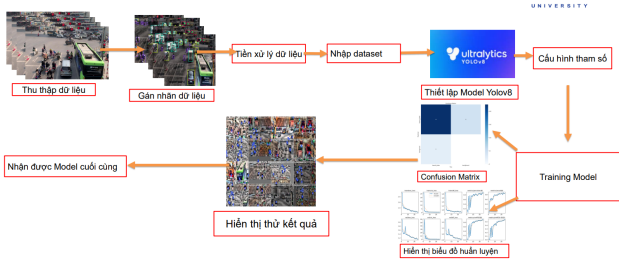
D. Hiển thị kết quả và tạo báo cáo

Hệ thống hiển thị các thông tin sau trên giao diện giám sát: Khung hình video với các hộp giới hạn đánh dấu phương tiện được phát hiện
Biển số đã nhận diện được hiển thị cùng với mỗi phương tiện Hiển thị trạng thái của đèn giao thông
Cảnh báo trực quan khi phát hiện vi phạm Báo cáo tổng hợp được tạo ra bao gồm:
Danh sách các biển số vi phạm theo thứ tự thời gian
Thống kê vi phạm theo thời gian, loại phương tiện
Hình ảnh bằng chứng vi phạm kèm theo thông tin thời gian và địa điểm

E. Đánh giá hiệu suất của hệ thống

Hiệu suất của hệ thống được đánh giá dựa trên các tiêu chí sau: Độ chính xác phát hiện phương tiện: Tỷ lệ phương tiện được phát hiện chính xác.
Độ chính xác nhận diện biển số: Tỷ lệ biển số được nhận diện đúng.
Độ chính xác xác định trạng thái đèn giao thông: Tỷ lệ đèn giao thông được xác định trạng thái chính xác.
Độ chính xác phát hiện vi phạm: Tỷ lệ vi phạm vượt đèn đỏ được phát hiện chính xác.
Thời gian xử lý: Thời gian xử lý trung bình cho mỗi khung hình.

F. Phương pháp luận huấn luyện mô hình



Hình 3: Quá trình thực hiện train model

1) Chuẩn bị và tiền xử lý dữ liệu: Dữ liệu huấn luyện được chuẩn bị theo nguyên tắc học máy chặt chẽ, bao gồm:
Thu thập và gán nhãn dữ liệu:

Hình 4: Dữ liệu ban đầu

Hình 5: Gán nhãn cho các loại phương tiện

Hình 6: Gán nhãn cho biển số

Chuẩn hóa dữ liệu: Áp dụng phép biến đổi tuyến tính để ảnh có giá trị cường độ pixel nằm trong khoảng [0,1] hoặc [-1,1]:

$$I_{norm} = \frac{I - \mu}{\sigma} \tag{36}$$

Trong đó μ và σ là giá trị trung bình và độ lệch chuẩn của cường độ pixel trên tập dữ liệu.
Phép biến đổi không gian màu: Chuyển đổi ảnh sang không gian màu khác như HSV, LAB, YCrCb để tăng khả năng phân biệt đặc trưng:

$$I_{HSV} = f_{RGB2HSV}(I_{RGB}) \tag{37}$$

Giảm nhiễu: Áp dụng bộ lọc Gaussian để làm mịn ảnh và loại bỏ nhiễu tần số cao:

$$I_{smooth} = I * G_\sigma \tag{38}$$

Trong đó G_σ là hàm Gaussian 2D với độ lệch chuẩn σ được định nghĩa:

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{39}$$

Tăng cường dữ liệu: Áp dụng các phép biến đổi ngẫu nhiên để mở rộng tập dữ liệu:

$$\mathcal{T} = \{T_i | i \in \{1, 2, ..., N\}\} \tag{40}$$

Trong đó T_i là các phép biến đổi như:

$$T_{rotate}(I, \theta) = R_\theta \cdot I \tag{41}$$
$$T_{flip}(I) = F \cdot I \tag{42}$$
$$T_{brightness}(I, \alpha) = \alpha \cdot I \tag{43}$$
$$T_{contrast}(I, \beta) = \beta \cdot (I - 0.5) + 0.5 \tag{44}$$

2) Cơ chế huấn luyện tối ưu: Quá trình huấn luyện được thực hiện theo phương pháp tối ưu hóa mini-batch gradient descent với các cơ chế cải tiến:

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_\theta \mathcal{L}(\theta_t) \tag{45}$$

Trong đó: θ_t là tham số mô hình tại bước lặp thứ t

η là tốc độ học (learning rate)
$\nabla_\theta \mathcal{L}(\theta_t)$ là gradient của hàm mất mát theo tham số θ
Các chiến lược tối ưu hóa được áp dụng bao gồm:
Adaptive Learning Rate: Điều chỉnh tốc độ học theo lịch trình cosine annealing:

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min}) \left(1 + \cos\left(\frac{t}{T}\pi\right)\right) \tag{46}$$

Momentum: Tích lũy gradient từ các bước lặp trước để cải thiện tốc độ hội tụ:

$$v_t = \gamma \cdot v_{t-1} + \eta \cdot \nabla_\theta \mathcal{L}(\theta_{t-1}) \tag{47}$$
$$\theta_t = \theta_{t-1} - v_t \tag{48}$$

Weight Decay: Áp dụng kỹ thuật chính quy hóa L2 để tránh hiện tượng quá khớp:

$$\mathcal{L}_{reg} = \mathcal{L} + \lambda \cdot \sum_i \theta_i^2 \tag{49}$$

Adam Optimizer: Kết hợp Adaptive Learning Rate và Momentum:

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot \nabla_\theta \mathcal{L}(\theta_{t-1}) \tag{50}$$
$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot (\nabla_\theta \mathcal{L}(\theta_{t-1}))^2 \tag{51}$$
$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{52}$$
$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{53}$$
$$\theta_t = \theta_{t-1} - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{54}$$

3) Phương pháp đánh giá mô hình: Hiệu suất của mô hình được đánh giá qua các độ đo chính xác:
Precision: Tỷ lệ dự đoán dương tính đúng trên tổng số dự đoán dương tính:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{55}$$

Recall: Tỷ lệ dự đoán dương tính đúng trên tổng số đối tượng dương tính thực tế:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{56}$$

F1-Score: Trung bình điều hòa của Precision và Recall:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{57}$$

IoU (Intersection over Union): Đo lường độ chồng lấp giữa hộp giới hạn dự đoán và hộp giới hạn ground truth:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{|B_p \cap B_{gt}|}{|B_p \cup B_{gt}|} \tag{58}$$

mAP (mean Average Precision): Đánh giá độ chính xác của cả dự đoán hộp giới hạn và phân loại đối tượng:

$$\text{AP} = \int_0^1 p(r)dr \approx \sum_{i=1}^{n} p(r_i)\Delta r_i \tag{59}$$
$$\text{mAP} = \frac{1}{|C|} \sum_{c \in C} \text{AP}_c \tag{60}$$

Trong đó C là tập hợp các lớp đối tượng, AP_c là Average Precision cho lớp c, $p(r)$ là đường cong precision-recall, và r_i là các mức recall khác nhau.

G. Lý thuyết nâng cao cho nhận diện phương tiện giao thông

1) Phân tích đặc trưng phương tiện giao thông: Việc nhận diện phương tiện giao thông đòi hỏi phân tích đặc trưng cụ thể như:
Đặc trưng hình dạng: Sử dụng bộ lọc Gabor để phát hiện đường viền ở nhiều hướng khác nhau:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi\frac{x'}{\lambda} + \psi\right) \tag{61}$$

Trong đó $x' = x\cos\theta + y\sin\theta$ và $y' = -x\sin\theta + y\cos\theta$, λ là bước sóng, θ là hướng, ψ là pha, σ là độ lệch chuẩn Gaussian, và γ là tỷ lệ không gian.
Đặc trưng vùng quan tâm: Áp dụng phương pháp selective search để đề xuất các vùng có khả năng cao chứa phương tiện:

$$\text{SimilarityMeasure}(r_i, r_j) = a_1 \cdot s_{color}(r_i, r_j) + a_2 \cdot s_{texture}(r_i, r_j) + a_3 \tag{62}$$

Trong đó
a_1, a_2, a_3, a_4 là các trọng số
$s_{color}, s_{texture}, s_{size}, s_{fill}$ là các độ đo tương đồng về màu sắc, kết cấu, kích thước và sự lấp đầy.
Đặc trưng ngữ cảnh: Xem xét quan hệ không gian giữa các đối tượng trong cảnh:

$$S_{context}(i, j) = w_1 \cdot d_{spatial}(r_i, r_j) + w_2 \cdot d_{scale}(r_i, r_j) + w_3 \cdot d_{appearanc} \tag{63}$$

Trong đó
$d_{spatial}$ là khoảng cách không gian, d_{scale} là sự khác biệt về tỷ lệ
$d_{appearance}$ là sự khác biệt về ngoại hình.
2) Tối ưu hóa mô hình cho các loại phương tiện: YOLOv8 được tối ưu hóa cho việc nhận diện phương tiện thông qua các kỹ thuật:
Focus on Scale: Điều chỉnh tỷ lệ anchor box để phù hợp với kích thước của các loại phương tiện khác nhau:

$$A_{optimal} = \arg\min_{A} \sum_{i=1}^{N} \min_{a \in A} d(a, b_i) \tag{64}$$

Trong đó $A = \{(w_i, h_i) | i \in \{1, 2, ..., K\}\}$ là tập hợp các anchor box, b_i là hộp giới hạn thực tế, và $d(a, b)$ là khoảng cách giữa anchor box và hộp giới hạn, thường dùng 1-IoU.
Transfer Learning: Tận dụng tri thức từ mô hình đã huấn luyện trước đó:

$$\theta_{init} = \theta_{pretrained} \tag{65}$$

$$\mathcal{L}_{transfer} = \mathcal{L}_{task} + \beta \cdot \mathcal{L}_{similarity}(\theta, \theta_{pretrained}) \tag{66}$$

Trong đó β là siêu tham số điều chỉnh mức độ giữ lại kiến thức từ mô hình gốc.
Feature Pyramid Attention: Cải tiến cơ chế chú ý (attention mechanism) để tập trung vào các đặc trưng quan trọng:

$$F_{att} = F \cdot \sigma(W_a \cdot F + b_a) \tag{67}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{68}$$

Trong đó W_a và b_a là tham số học được, σ là hàm sigmoid, và F là đặc trưng đầu vào.
Spatial-Temporal Attention: Tích hợp thông tin thời gian cho nhận diện trong video:

$$A_{st}(F_t) = \gamma \cdot A_s(F_t) + (1 - \gamma) \cdot A_t(F_{t-n:t}) \tag{69}$$

Trong đó A_s là attention không gian, A_t là attention thời gian, và γ là trọng số cân bằng.

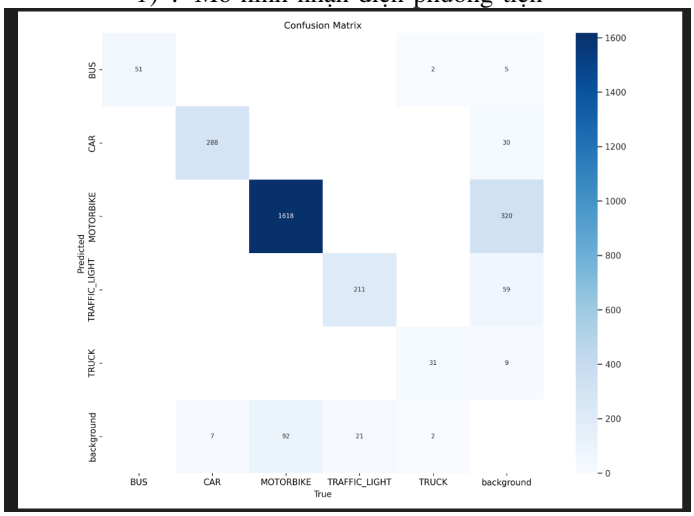
H. Kết luận

Hệ thống nhận diện phương tiện và phát hiện vượt đèn đỏ từ video cung cấp giải pháp tự động hóa hiệu quả cho việc giám sát giao thông. Thông qua việc tích hợp các công nghệ thị giác máy tính và học sâu, hệ thống có khả năng phát hiện chính xác các trường hợp vi phạm giao thông mà không cần sự can thiệp của con người. Điều này góp phần tăng cường an toàn giao thông và hỗ trợ công tác quản lý đô thị thông minh.

V. Kết quả

A. Model

1) : Mô hình nhận diện phương tiện



Hình 7: Ma trận nhầm lẫn của mô hình nhận diện phương tiện

Mô hình hoạt động với các lớp:
- **BUS**: Xe buýt
- **CAR**: Ô tô
- **MOTORBIKE**: Xe máy
- **TRAFFIC_LIGHT**: Đèn giao thông
- **TRUCK**: Xe tải
- **background**: Nền, không phải phương tiện

Phân tích kết quả: - Nhãn **"MOTORBIKE"** có **1618** dự đoán đúng, đạt hiệu suất cao nhất. - Nhãn **"CAR"** có **288** dự đoán đúng, nhưng bị nhầm lẫn khá nhiều.
Các lỗi phổ biến:
- Nhãn **"CAR"** bị nhầm thành **"background"** (**92 lần**).
- Nhãn **"MOTORBIKE"** bị nhầm thành **"TRAFFIC_LIGHT"** (**211 lần**).
- Nhãn **"TRUCK"** bị nhầm thành **"CAR"** (**320 lần**).

Hiệu suất mô hình:
1. Độ chính xác (Accuracy):

$$\text{Accuracy} = \frac{51 + 288 + 1618 + 59 + 320 + 9}{\text{Tổng số điểm dữ liệu}} \tag{70}$$

2. Tỷ lệ dương tính thực sự (Recall):

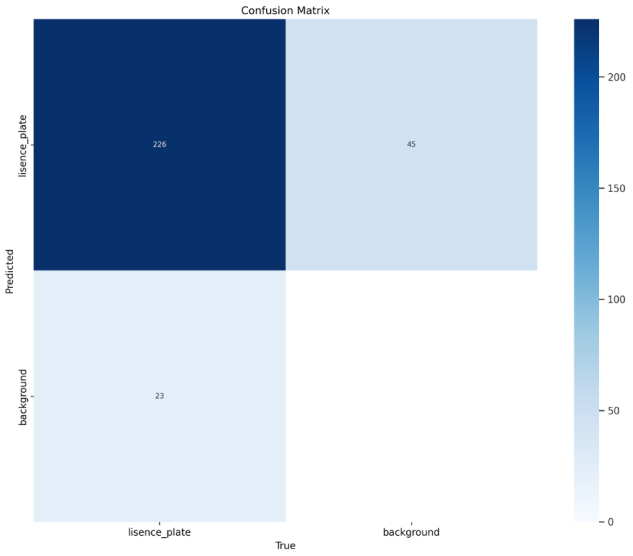
$$\text{Recall}_{MOTORBIKE} = \frac{1618}{1618 + 211} \approx 88.5\% \tag{71}$$

3. Độ chính xác của dự đoán dương tính (Precision):

$$\text{Precision}_{CAR} = \frac{288}{288 + 92} \approx 75.8\% \tag{72}$$

Nhận xét: - Mô hình có xu hướng dự đoán chính xác cho các phương tiện chính. - Nhãn "MOTORBIKE"dễ bị nhầm với đèn giao thông, cần cải thiện khả năng phân biệt. - Nhãn "CAR"đôi khi bị nhầm với background, có thể do đặc điểm hình ảnh chưa rõ ràng.

B. Mô hình nhận diện biển số xe



Hình 9: Ma trận nhầm lẫn của mô hình nhận diện biển số xe

Mô hình hoạt động với hai lớp:
- **license_plate**: Biển số xe.
- **background**: Nền, không phải biển số xe.

Phân tích ma trận:
- **226** lần mô hình dự đoán đúng biển số xe (license_plate).
- **45** lần mô hình nhầm lẫn biển số xe thành background (*false negative*).
- **23** lần mô hình nhầm background thành biển số xe (*false positive*).

Đánh giá hiệu suất mô hình:

1. Độ chính xác (Accuracy):

$$\text{Accuracy} = \frac{\text{Dự đoán đúng}}{\text{Tổng số mẫu}} = \frac{226+0}{226+45+23+0} = \frac{226}{294} \approx 76.87\% \tag{73}$$

2. Tỷ lệ dương tính thực sự (Recall - Sensitivity):

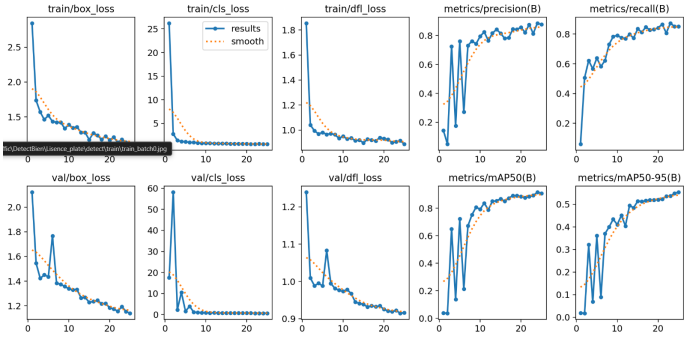
$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} = \frac{226}{226+45} = \frac{226}{271} \approx 83.39\% \tag{74}$$

3. Tỷ lệ dương tính dự đoán chính xác (Precision):

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} = \frac{226}{226+23} = \frac{226}{249} \approx 90.76\% \tag{75}$$

4. Phân tích lỗi: - **False Positive (23 lần):** Mô hình nhận nhầm background thành biển số xe. - **False Negative (45 lần):** Mô hình bỏ sót biển số xe.

Nhận xét: - Mô hình có độ chính xác tốt nhưng cần cải thiện để giảm lỗi bỏ sót biển số xe (false negative). - Có thể điều chỉnh ngưỡng dự đoán hoặc cải thiện dữ liệu huấn luyện để tối ưu hiệu suất.



Hình 10: Kết quả training model

Hình 11: Kết quả của hệ thốngl

VI. Kết luận và hướng phát triển

Trong nghiên cứu này, chúng tôi đã sử dụng mô hình YOLOv8 để nhận diện phương tiện giao thông và biển số xe trên tập dữ liệu tuỳ chỉnh. Kết quả thu được cho thấy mô hình có độ chính xác cao với những nhãn phổ biến như "MOTORBIKE"và "CAR". Tuy nhiên, một số sai sót vẫn xảy ra, đặc biệt là việc nhầm lẫn giữa các lớp có đặc điểm tương đồng, chẳng hạn như "TRUCK"bị nhầm thành "CAR"hoặc "MOTORBIKE"bị nhầm thành "TRAFFIC_LIGHT".
Một số yếu tố ảnh hưởng đến hiệu suất của mô hình bao gồm điều kiện ánh sáng, góc nhìn của camera, và chất lượng ảnh đầu vào. Những yếu tố này có thể làm giảm độ chính xác của mô hình trong môi trường thực tế.
Hướng phát triển trong tương lai:
- Cải thiện tập dữ liệu: Tăng kích thước và đa dạng hoá tập dữ liệu huấn luyện, bao gồm nhiều điều kiện thời tiết, ánh sáng khác nhau và các góc nhìn phức tạp hơn.
- Tối ưu hoá mô hình: Thử nghiệm các kỹ thuật tăng cường dữ liệu (data augmentation) và tinh chỉnh tham số (hyperparameter tuning) để cải thiện độ chính xác và khả năng tổng quát hoá.
- Ứng dụng trong thực tế: Triển khai mô hình vào hệ thống giám sát giao thông thông minh để kiểm soát vi phạm, phân tích luồng giao thông và hỗ trợ các ứng dụng an ninh.

Tài liệu

[1] Jocher, G., Chaurasia, A., & Qiu, J. (2023). *YOLOv8: Real-Time Object Detection*. Available at: https://github.com/ultralytics/ultralytics

[2] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

[3] Szeliski, R. (2022). *Computer Vision: Algorithms and Applications*. Springer.

[4] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, Real-Time Object Detection*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[5] Girshick, R. (2015). *Fast R-CNN*. IEEE International Conference on Computer Vision (ICCV).

[6] Ren, S., He, K., Girshick, R., & Sun, J. (2015). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. Advances in Neural Information Processing Systems (NeurIPS).

[7] Howard, A. G., et al. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. arXiv preprint arXiv:1704.04861.

[8] He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep Residual Learning for Image Recognition*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).