Tạp chí điện tử
Khoa học và Công nghệ Giao thông
Trang website: https://jstt.vn/index.php/vn

Proposing a Model for License Plate Recognition of Non-Helmeted Traffic Participants Based on YOLOv11 Deep Learning Network

Pham Nhu Pham*, Pham Van Dieu, Nguyen Minh Nhut, Nguyen Thanh Minh
Gia Lai College, Tran Nhat Duat Street, Dien Hong Ward, Gia Lai Province, Vietnam

Article info
Type of article:
Original research paper

DOI:
https://doi.org/10.58845/jstt.utt.2026.vn.6.5.34-53

***Corresponding author:**
Email address:
phamnhupham@gmail.com

Received: 23/10/2025
Received in Revised Form: 11/06/2026
Accepted: 10/07/2026

Abstract: Deep learning is a field of artificial intelligence, and deep learning algorithms are attracting significant research interest from scientists to improve lightweight architecture, speed, and accuracy. Deep learning models are built from intelligent algorithms, enabling computers to learn from existing data and make predictions and process tasks quickly and accurately. This paper presents research and application of the YOLOv11 deep learning network to detect and recognize license plates of motorcyclists not wearing helmets. The model was trained on a specific dataset including images of motorcyclists without helmets, license plates, and characters. The results show that the training model achieved high evaluation scores with Precision = 0.889 - 0.981, Recall = 0.867 - 0.976, mAP@50 = 0.918 - 0.984 and mAP@50-95 = 0.607 - 0.774, processing speed of approximately FPS = 1.0, which demonstrates the feasibility and potential for practical application of the research.

Keywords: Deep learning, YOLOv11, detect objects, helmet, license plate recognition.

Tạp chí điện tử
Khoa học và Công nghệ Giao thông
Trang website: https://jstt.vn/index.php/vn

Nghiên cứu mạng học sâu YOLOv11, đề xuất mô hình nhận diện biển số xe người tham gia giao thông không đội mũ bảo hiểm

Phạm Như Phẩm*, Phạm Văn Điều, Nguyễn Minh Nhựt, Nguyễn Thanh Minh
Trường Cao đẳng Gia Lai, Đường Trần Nhật Duật, Phường Diên Hồng, tỉnh Gia Lai, Việt Nam

Thông tin bài viết
Dạng bài viết:
Bài báo nghiên cứu

DOI:
https://doi.org/10.58845/jstt.utt.2026.vn.6.5.34-53

***Tác giả liên hệ:**
Địa chỉ Email:
phamnhupham@gmail.com

Ngày nộp bài: 23/10/2025
Ngày nộp bài sửa: 11/06/2026
Ngày chấp nhận: 10/07/2026

Tóm tắt: Học sâu là một lĩnh vực của trí tuệ nhân tạo, các thuật toán học sâu đang được các nhà khoa học quan tâm nghiên cứu, để cải tiến kiến trúc gọn nhẹ, nâng cao tốc độ và độ chính xác. Các mô hình học sâu được xây dựng từ các thuật toán thông minh, giúp máy tính có khả năng học từ dữ liệu sẵn có và thực hiện các dự đoán, xử lý công việc một cách nhanh chóng và chính xác. Bài báo trình bày về nghiên cứu, ứng dụng mạng học sâu YOLOv11 để phát hiện, nhận dạng biển số xe người tham gia giao thông không đội mũ bảo hiểm. Mô hình được huấn luyện trên tập dữ liệu riêng gồm hình ảnh người tham gia giao thông không đội mũ bảo hiểm, biển số xe và ký tự. Kết quả cho thấy, mô hình huấn luyện đạt được các chỉ số đánh giá cao, với Precision = 0.889 - 0.981, Recall = 0.867 - 0.976, mAP@50 = 0.918 - 0.984 và mAP@50-95 = 0.607 - 0.774, tốc độ xử lý khoảng FPS = 1.0. Điều này chứng minh nghiên cứu có tính khả thi và tiềm năng ứng dụng trong thực tế.

Từ khóa: Học sâu, YOLOv11, phát hiện đối tượng, mũ bảo hiểm, nhận diện biển số.

1. Giới thiệu

Phát hiện đối tượng là tác vụ phân loại và định vị trí các đối tượng trong hình ảnh hoặc video, là một lĩnh vực trọng tâm của thị giác máy tính. Nó đã trở nên nổi bật trong những năm gần đây nhờ những ứng dụng rộng rãi trong các bài toán trong thực tế như phát hiện theo dõi đối tượng, giám sát giao thông, xe tự lái, robot thông minh [1, 2]. Hiện nay, có nhiều mô hình học sâu tiên tiến, dùng để nhận diện phát hiện đối tượng như CNN (Convolutional Neural Networks), R-CNN (Region based Convolutional Neural Networks), Fast R-CNN, Faster R-CNN, Mask R-CNN, YOLO (You Only Look Once) [3]. Ứng dụng trí tuệ nhân tạo để phát hiện người lao động ở các công trình [4] hay người tham gia giao thông mà không đội mũ bảo hiểm [5, 6] là vấn đề cần thiết của các nhà quản lý, nhằm giảm bớt những chấn thương phần đầu khi xảy ra tai nạn. Việc phân tích hình ảnh chứa đối tượng người đi xe máy không đội mũ bảo hiểm, biển số xe đòi hỏi thuật toán phải phức tạp và chính xác.

YOLO là thuật toán sử dụng mạng học sâu của trí tuệ nhân tạo để phát hiện và nhận diện đối tượng, rất phù hợp đối với các bài toán về phát hiện mục tiêu theo thời gian thực và đang được các nhà khoa học tiếp tục nghiên cứu, phát triển, hướng tới cho ra những phiên bản mạnh mẽ hơn

về tốc độ, độ chính xác [7]. Phát hiện người đi xe máy không đội mũ bảo hiểm và nhận diện biển số xe bằng thuật toán YOLO đã có nhiều công trình nghiên cứu và công bố, như các bài báo [3-6], [8-12]. Bên cạnh đó, cũng có những công trình nghiên cứu tổng hợp hai nội dung trên, nghĩa là phát hiện người đi xe máy không đội mũ bảo hiểm thì đồng thời nhận diện trích xuất ký tự biển số xe, như các nghiên cứu [9-11]. Mỗi nghiên cứu dựa vào các thuật toán, thư viện, phần mềm riêng và kết quả cuối cùng là đọc được chính xác biển số. Bài báo [9] của nhóm tác giả Swapna Rani Kura sử dụng công nghệ camera "CCTV footage" để phát hiện nhận diện mũ bảo hiểm và biển số xe. Bài báo đề cập về cài đặt thư viện học sâu TensorFlow, thư viện OpenCV, tải dữ liệu mô hình YOLO hoặc SSD và phần mềm sử dụng thiết bị, kết quả nhận diện có độ chính xác cao, nhưng chưa nói rõ về học thuật phát hiện mũ bảo hiểm và nhận diện trích xuất biển số. Hoặc bài báo [10], nhóm tác giả dùng TensorFlow, EasyOCR và YOLOv8 để nhận diện xử lý ảnh. Sử dụng thư viện Ultralytics và OpenCV để huấn luyện dữ liệu và gán nhãn mũ bảo hiểm, không mũ bảo hiểm, biển số cho tập dữ liệu bằng công cụ LabelImg. Quy trình thực hiện theo thứ tự là mô hình nhận diện, phân loại được người đi xe máy đội mũ hoặc không đội mũ bảo hiểm và sau đó sử dụng thư viện EasyOCR để trích xuất biển số xe. Trong bài nghiên cứu [11], nhóm tác giả giới thiệu YOLOv3 để phát hiện mũ bảo hiểm và trích xuất thông tin biển số xe.

Dữ liệu hình ảnh đường phố Ấn Độ được thu thập thủ công, gán nhãn bằng Label Studio cho các đối tượng: xe máy, biển số, mũ bảo hiểm, không mũ bảo hiểm, và phân chia theo tỷ lệ 80:20 lần lượt cho tập huấn luyện và kiểm định. Mô hình được huấn luyện trên cấu hình phần cứng PC core i9 Windows OS, Ram 16GB, GPU RTX 3060. Hiệu suất huấn luyện cao, độ chính xác trung bình mAP50 là 92.44% và các thông số huấn luyện khác được thể hiện ở Bảng 1.

Bảng 1. Độ chính xác trung bình cho các lớp đối tượng của bài nghiên cứu [11]

Lớp	Độ chính xác trung bình	Dự đoán đúng	Dự đoán sai
Helmet	91.54%	16	4
No_Helmet	90.72%	36	2
Number_plate	89.18%	13	3
Motorcycle	94.88%	75	2

Qua các công trình nghiên cứu [1–11], có thể nhận thấy phần lớn tác giả lựa chọn kiến trúc YOLO để xây dựng và triển khai hệ thống nhận dạng đối tượng, trong đó có bài toán phát hiện nhận diện và trích xuất thông tin người tham gia giao thông không đội mũ bảo hiểm. Bởi vì YOLO đạt được sự cân bằng hiệu quả giữa độ chính xác và tốc độ xử lý, đặc biệt phù hợp với các ứng dụng yêu cầu xử lý gần thời gian thực. So với các mô hình phát hiện đối tượng hai giai đoạn như Faster R-CNN, mặc dù có độ chính xác cao nhưng đòi hỏi chi phí tính toán lớn, YOLO thuộc nhóm mô hình một giai đoạn, cho phép phát hiện trực tiếp đối tượng trên ảnh đầu vào với tốc độ suy luận nhanh hơn đáng kể trong khi vẫn duy trì độ chính xác ở mức cao.

Vì vậy, nhóm tác giả tiếp tục lựa chọn YOLOv11 để phát hiện, nhận dạng và trích xuất thông tin biển số của người tham gia giao thông không đội mũ bảo hiểm theo một hướng tiếp cận khác. Bài báo này tập trung vào tìm hiểu, nghiên cứu thuật toán YOLOv11 và huấn luyện tinh chỉnh theo mô hình YOLOv11n với các bộ dữ liệu riêng của nhóm tác giả. Các bộ dữ liệu gồm: bộ dữ liệu phát hiện người không đội mũ bảo hiểm, bộ dữ liệu phát hiện người đi xe máy, bộ dữ liệu khoanh vùng chứa biển số và bộ dữ liệu đọc ký tự. Từ đó có khả năng phát hiện người đi xe máy không đội mũ bảo hiểm và nhận diện biển số xe người vi phạm giao thông một cách kịp thời và chính xác.

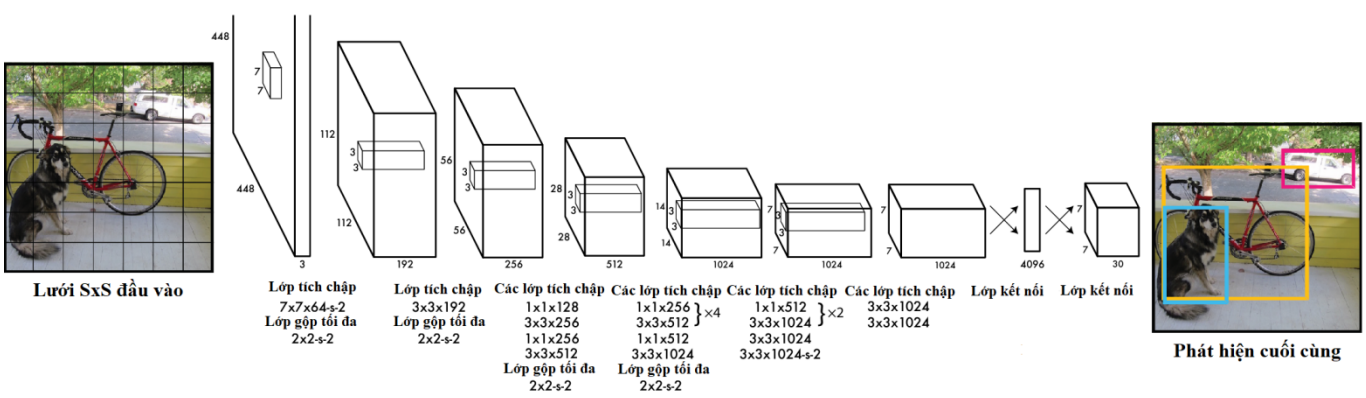
2. Cơ sở lý thuyết và phương pháp nghiên cứu

2.1. Sơ lược về phát hiện đối tượng đối với mô hình YOLO

YOLO [12] là một trong những mô hình phát hiện và phân loại đối tượng đầu tiên sử dụng mạng nơ-ron học sâu CNN, được xây dựng từ sự kết hợp giữa các lớp tích chập (convolutional layers) và các lớp kết nối đầy đủ (fully-connected layers) được thể hiện ở hình 1. Trong đó các lớp tích chập sẽ trích xuất các đặc trưng (feature) của ảnh, các lớp kết nối đầy đủ sẽ dự đoán tọa độ của đối tượng và cho kết quả dữ liệu ở đầu ra [1, 2, 12, 13].

Khi nhận ảnh đầu vào, mô hình sẽ chuẩn hóa ảnh về dạng kích thước cố định và chia nhỏ ảnh đó thành các ô lưới có kích thước S×S. Nhờ khả năng tối ưu hóa cho phần cứng, xử lý song song GPU (Graphics Processing Unit) nên mạng CNN có khả năng dự đoán chỉ một lần truyền dữ liệu duy nhất và đầu ra là ma trận ba chiều S×S×(5×B+C), với mỗi ô dự đoán chứa (5×B+C) tham số. Trong đó tham số B biểu thị số lượng hộp giới hạn (bounding boxes) và độ tin cậy (confidence), C là số lớp đối tượng cần phân loại. Mỗi hộp giới hạn có 5 thông số là x, y, w, h và độ tin cậy. Thông số x, y là tọa độ tâm hộp, w là chiều rộng và h là chiều cao của hộp, còn độ tin cậy thể hiện độ tự tin của dự đoán.

Sau khi tất cả các hộp giới hạn được dự đoán, YOLO sử dụng phương pháp triệt tiêu không cực đại (Non-Maximum Suppression) để loại bỏ các hộp giới hạn trùng lặp hoặc giao nhau quá lớn, chỉ giữ lại những hộp giới hạn có độ tin cậy cao nhất cho mỗi đối tượng. Do đó, đầu ra cuối cùng của YOLO gồm tập hợp các hộp giới hạn duy nhất, kèm theo nhãn lớp và độ tin cậy tương ứng [1, 3, 12, 13].



Hình 1. Mô hình kiến trúc tổng quát của mạng YOLO [1, 2, 12, 13]

2.2. Mô hình YOLOv11

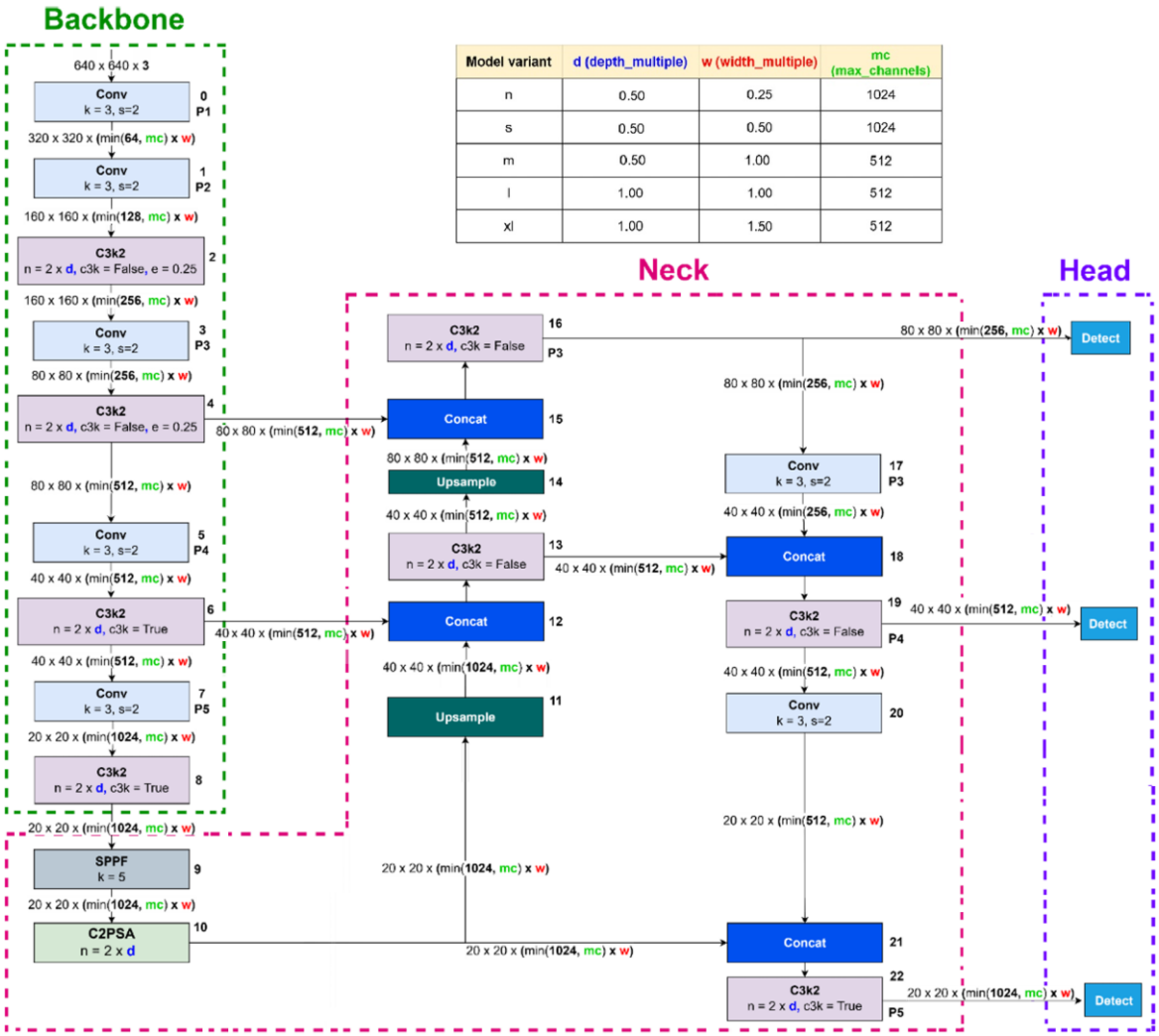
YOLOv11 là phiên bản mới trong họ mô hình phát hiện đối tượng theo thời gian thực của Ultralytics, được công bố tại hội nghị YOLO Vision năm 2024 [15]. Mô hình này được phát triển dựa trên các phiên bản YOLO trước đó nhằm cải thiện độ chính xác, tốc độ xử lý và hiệu quả tính toán. YOLOv11 tích hợp một số cải tiến quan trọng như các khối C3k2 (Cross - Stage Partial Bottleneck with 2 stacked: Khối thắt cổ chai bán phần liên tầng với 2 lớp xếp chồng), SPPF (Spatial Pyramid Pooling – Fast: Khối gộp kim tự tháp không gian tốc độ cao) và C2PSA (Cross - Stage Partial Spatial Attention: Khối chú ý không gian bán phần liên tầng). Những thành phần này giúp tăng cường khả năng trích xuất đặc trưng ở nhiều mức không gian khác nhau, đồng thời vẫn đảm bảo số lượng tham số nhỏ gọn và hiệu suất tính toán cao. Nhờ đó, mô hình có thể đạt hiệu quả tốt trong các nhiệm vụ như phát hiện, phân loại và theo dõi đối tượng trong thời gian thực [14, 15]. Kiến trúc của YOLOv11 được tổ chức thành ba thành phần chính gồm Backbone, Neck và Head (hình 2), mỗi phần đảm nhận một chức năng riêng trong quá trình xử lý ảnh và dự đoán đối tượng.

Backbone là bộ phận thực hiện trích xuất đặc trưng từ ảnh đầu vào bằng các lớp tích chập của mạng nơ-ron CNN. Ảnh đầu vào có kích thước 640×640×3 được chuyển đổi thành các bản đồ đặc trưng với kích thước giảm dần như 320×320×64, 160×160×128, 80×80×256, 40×40×512 và 20×20×1024. Trong quá trình này, khối C3k2 giúp cải thiện khả năng học đặc trưng ở nhiều mức độ phân giải, đặc biệt tại các tầng trung gian. Kết quả của Backbone là các bản đồ đặc trưng đa tỷ lệ, được chuyển sang phần Neck để tiếp tục xử lý [15, 16].

Neck đóng vai trò kết nối và tổng hợp các đặc trưng từ nhiều tầng khác nhau của Backbone. Thành phần này sử dụng các kỹ thuật như SPPF để thu nhận thông tin đa tỷ lệ và C2PSA để tăng cường khả năng chú ý không gian của mô hình. Bên cạnh đó, các thao tác Upsample và Concat được sử dụng để tăng độ phân giải của bản đồ đặc trưng và kết hợp thông tin từ nhiều tầng khác nhau. Nhờ vậy, mô hình có thể biểu diễn tốt hơn các đối tượng có kích thước đa dạng trong ảnh [14-16].

Head là phần cuối của mô hình, chịu trách nhiệm đưa ra dự đoán về vị trí và loại đối tượng. Thành phần này thường được chia thành hai nhánh chính: nhánh định vị (bounding box) để xác định tọa độ và kích thước khung bao quanh đối tượng, và nhánh phân loại (class prediction) để xác định nhãn lớp của đối tượng kèm theo giá trị độ tin cậy (confidence score).

Cơ chế dự đoán hai nhánh giúp YOLOv11 vừa xác định chính xác vị trí vừa nhận diện đúng loại đối tượng, từ đó nâng cao hiệu quả trong các bài toán phát hiện đối tượng theo thời gian thực [14-16].



Model variant	d (depth_multiple)	w (width_multiple)	mc (max_channels)
n	0.50	0.25	1024
s	0.50	0.50	1024
m	0.50	1.00	512
l	1.00	1.00	512
xl	1.00	1.50	512

Hình 2. Kiến trúc của YOLOv11 [15]

2.3. Quy trình phát hiện và trích xuất biển số xe người tham gia giao thông

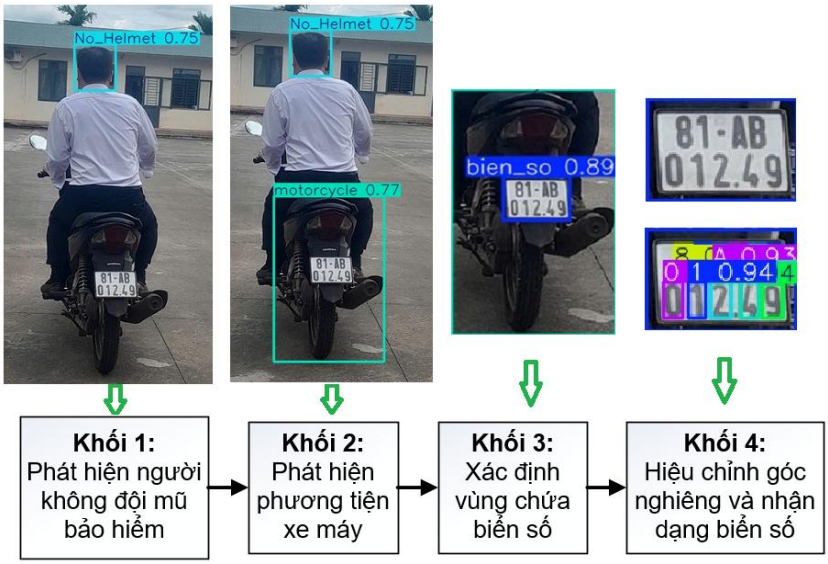
Để phát hiện và trích xuất biển số xe của người tham gia giao thông không đội mũ bảo hiểm bằng thuật toán nhận diện vật thể YOLOv11, tác giả đề xuất mô hình tổng thể gồm bốn khối chức năng như thể hiện ở hình 3, trong đó mỗi khối đảm nhiệm một nhiệm vụ cụ thể như sau:

- Khối 1: Phát hiện người không đội mũ bảo hiểm: Khối này nhận dữ liệu hình ảnh từ video hoặc camera giao thông, sau đó sử dụng mô hình YOLOv11 đã được huấn luyện để phát hiện người không đội mũ bảo hiểm.

- Khối 2: Phát hiện phương tiện xe máy: Để tiếp nhận đầu vào từ khối thứ nhất, nhóm tác giả dùng phiên bản YOLOv11n của YOLOv11 để xác định đối tượng là người điều khiển xe máy.

- Khối 3: Xác định vùng chứa biển số: Dựa trên mô hình YOLOv11n đã tinh chỉnh, nhóm tác giả thực hiện việc định vị và cắt vùng ảnh có chứa biển số.

- Khối 4: Hiệu chỉnh góc nghiêng và nhận dạng ký tự biển số: Đầu tiên xác định tọa độ của hai ký tự ở rìa ngoài cùng của dòng trên (hoặc dòng dưới) trên biển số, sau đó tính toán góc nghiêng và xoay biển số sao cho các ký tự thẳng đứng nhất và cuối cùng dùng mô hình tinh chỉnh YOLOv11n để tách riêng và nhận diện từng ký tự trên biển số, đồng thời hiển thị vùng chứa ký tự cùng tên ký tự tương ứng.



Hình 3. Sơ đồ khối các bước phát hiện, nhận diện và trích xuất biển số xe máy của người tham gia giao thông không đội mũ bảo hiểm

2.4. Phương thức đánh giá mô hình

Để đánh giá mức độ hiệu quả của mô hình phát hiện đối tượng YOLO, bốn chỉ số định lượng chính được sử dụng phổ biến, bao gồm: mAP (Mean Average Precision – Giá trị chính xác trung bình); Precision (độ chính xác); Recall (độ nhạy); và F1-Score (điểm F1). Các chỉ số này phản ánh toàn diện khả năng nhận diện đúng, bao phủ đầy đủ, cân bằng giữa độ chính xác và độ nhạy của mô hình.

- Chỉ số mAP biểu thị độ chính xác trung bình của mô hình trên toàn bộ các lớp đối tượng.

Với mỗi lớp, giá trị AP (Average Precision – Độ chính xác trung bình) được xác định bằng diện tích dưới đường cong Precision-Recall, sau đó mAP được tính bằng trung bình cộng của các AP trên tất cả các lớp, thể hiện ở công thức (1) như sau:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \text{ [16]} \qquad (1)$$

Trong đó: N là tổng số lớp đối tượng; mAP thường được tính ở các ngưỡng IoU (Intersection over Union – Mức độ chồng lấp giữa hộp giới hạn dự đoán và hộp giới hạn thực tế) khác nhau, như

mAP@0.5: chỉ tính khi IoU ≥ 0.5, còn mAP@[0.5:0.95]: tính trung bình các giá trị mAP từ 0.5 đến 0.95 với bước 0.05 theo bộ dữ liệu chuẩn COCO (Common Objects in Context – Các đối tượng phổ biến trong bối cảnh).

Giá trị mAP càng cao thể hiện mô hình phát hiện đối tượng càng chính xác và ổn định trên toàn bộ tập dữ liệu.

- Precision thể hiện tỷ lệ dự đoán đúng trong tổng số dự đoán mà mô hình nhận định là có đối tượng, được tính theo công thức (2):

$$\text{precision}=\frac{TP}{TP+FP} \text{ [16]} \quad (2)$$

Trong đó: TP (True Positive - Dương tính đúng) là số dự đoán đúng, và FP (False Positive – Dương tính giả) là số dự đoán sai. Precision cao cho thấy mô hình có khả năng phân biệt rõ ràng giữa đối tượng thật và nhiễu nền, hạn chế báo sai.

- Recall phản ánh khả năng mô hình phát hiện đầy đủ các đối tượng thực sự có trong ảnh, được tính theo công thức (3):

$$\text{recall}=\frac{TP}{TP+FN} \text{ [16]} \quad (3)$$

Với FN (False Negative - Âm tính giả) là số đối tượng thật mà mô hình không phát hiện được. Giá trị Recall cao cho thấy mô hình ít bỏ sót đối tượng, đặc biệt quan trọng trong các ứng dụng yêu cầu độ bao phủ cao như giám sát an ninh hay y tế.

F1-Score là chỉ số tổng hợp giữa Precision và Recall, thể hiện mức cân bằng giữa khả năng phát hiện chính xác và bao phủ của mô hình. Giá trị F1 càng gần 1 cho thấy mô hình đạt hiệu năng ổn định, tránh tình trạng lệch về một trong hai chỉ số Precision hoặc Recall.

3. Thực nghiệm và đánh giá kết quả

Để xây dựng tổng quát mô hình phát hiện và nhận diện biển số xe của người tham gia giao thông không đội mũ bảo hiểm dựa trên thuật toán YOLOv11, nhóm tác giả cần phải thực hiện qua các bước sau: Thu thập dữ liệu hình ảnh, tiền xử lý gán nhãn dữ liệu, huấn luyện mô hình, thực nghiệm mô hình và đánh giá mô hình.

3.1. Bộ dữ liệu

Thu thập dữ liệu và chuẩn hóa là một bước quan trọng của mô hình học sâu. Bộ dữ liệu COCO8 trong YOLOv11 được tổ chức theo cấu trúc thư mục phân cấp rõ ràng, dễ quản lý, tuân theo chuẩn chung của họ YOLO, gồm ba thành phần chính: Thư mục chứa ảnh, thư mục chứa nhãn và tệp cấu hình dữ liệu.

Thư mục chứa ảnh lưu trữ toàn bộ ảnh đầu vào của mô hình, có định dạng phổ biến là JPG hoặc PNG và kích thước là 640x640 pixels. Các ảnh trong thư mục này được chia thành hai tập con là thư mục huấn luyện chứa các ảnh dùng cho giai đoạn huấn luyện mô hình, chiếm khoảng 80% và thư mục kiểm thử chứa các ảnh dùng để kiểm thử và đánh giá hiệu suất mô hình sau huấn luyện, chiếm khoảng 20%.

Bảng 2. Cấu trúc các bộ dữ liệu

Bộ dữ liệu	Tổng số ảnh	Thư mục huấn luyện	Thư mục kiểm thử
Phát hiện người không đội mũ bảo hiểm	19214	15214	4000
Phát hiện vùng chứa biển số	9038	7225	1813
Nhận diện ký tự	4100	3275	825

Thư mục chứa nhãn lưu các tệp tin văn bản dạng (.txt). Mỗi tệp nhãn có tên trùng khớp với ảnh tương ứng trong thư mục chứa ảnh. Các tệp này chứa thông tin về vị trí và loại đối tượng, gồm năm giá trị: class_id, x_center, y_center, width, height. Các giá trị tọa độ và kích thước này được tác giả sử dụng phần mềm MakeSense để gán nhãn tạo ra và được chuẩn hóa theo kích thước ảnh. Tương tự như thư mục chứa ảnh, thư mục chứa nhãn cũng được chia thành hai phần là thư mục huấn luyện và thư mục kiểm thử, đảm bảo tính đồng bộ giữa dữ liệu huấn luyện và kiểm thử.

Tệp cấu hình dữ liệu nằm tại thư mục gốc của bộ dữ liệu, đóng vai trò là tệp cấu hình chính, được sử dụng để chỉ định đường dẫn đến thư mục train và val, đồng thời khai báo danh sách

các lớp đối tượng trong bộ dữ liệu.

Trong mô hình phát hiện nhận diện này, nhóm tác giả sử dụng bốn bộ dữ liệu. Trong đó bộ dữ liệu phát hiện xe máy sử dụng của YOLO đã huấn luyện sẵn trên tập COCO và 3 bộ dữ liệu còn lại do nhóm tự xây dựng. Số lượng hình ảnh và tỷ lệ chia ảnh ở thư mục train và thư mục val của các bộ dữ liệu được trình bày ở Bảng 2. Bộ dữ liệu phát hiện không đội mũ bảo hiểm được tác giả thu thập từ nguồn Roboflow Universe [17], trong đó dữ liệu giao thông Việt Nam chiếm khoảng 50%. Ngoài ra, do đặc điểm vùng đầu của con người (đội mũ hoặc không đội mũ) tương đối tương đồng giữa các quốc gia nên mô hình có khả năng học và khái quát tốt trong quá trình huấn luyện. Bộ dữ liệu khoanh vùng chứa biển số và nhận dạng ký tự, được xây dựng từ hai nguồn chính là một phần do tác giả sưu tầm và tổng hợp từ các dự án nghiên cứu trong nước, phần còn lại được thu thập trực tiếp thông qua việc chụp ảnh biển số. Bên cạnh đó, để tăng mức độ phức tạp và khả năng tổng quát mô hình khoanh vùng biển số, tác giả dùng thư viện OpenCV (Open Source Computer Vision Library – Thư viện thị giác máy tính mã nguồn mở) để mở rộng dữ liệu bằng cách tạo thêm các biến thể ảnh như thay đổi về độ mờ, sáng, đường biên, góc nghiêng, thu nhỏ và phóng to, giúp mô hình có thể nhận diện một cách ổn định và chính xác hơn trong điều kiện thực tế.

3.2. Huấn luyện mô hình

Huấn luyện mô hình YOLOv11 là quá trình giúp mạng học sâu học được các đặc trưng của đối tượng trong ảnh để thực hiện phát hiện và phân loại chính xác ảnh thực tế. Thuật toán YOLOv11 đã có các mô hình được huấn luyện sẵn như YOLOv11n, YOLOv11s, YOLOv11m và YOLOv11x. Tuy nhiên, các mô hình này chưa được tối ưu để phát hiện và nhận diện đối tượng đặc thù trong nghiên cứu này, cụ thể là trích xuất thông tin biển số của người điều khiển xe máy không đội mũ bảo hiểm. Vì vậy, nhóm tác giả cần tiến hành huấn luyện tinh chỉnh lại mô hình với bộ dữ liệu gồm các hình ảnh liên quan đến mũ bảo hiểm, biển số xe và ký tự trên biển số.

Bảng 3. Các tham số huấn luyện chính của mô hình YOLOv11n

Tham số	Giá trị	Ý nghĩa
Epochs	48/80/90	Số vòng huấn luyện
Imgsz	640	Kích thước ảnh đầu vào pixel
Batch size	16	Số lượng ảnh xử lý trong mỗi lô (batch)
Optimizer	SGD	Tối ưu giảm dần độ dốc ngẫu nhiên
Learning rate (lr0)	0.01	Tốc độ học ban đầu
Momentum	0.937	Tham số quán tính cho SGD
box	7.5	Mất mát cho hộp giới hạn
cls	0.5	Mất mát cho phân loại
dfl	1.5	Mất mát cho tiêu điểm phân phối

Để quá trình huấn luyện đạt hiệu quả và rút ngắn thời gian xử lý, hệ thống cần máy tính có cấu hình mạnh, đặc biệt là GPU để thực hiện xử lý ảnh nhanh chóng. Do đó, nhóm nghiên cứu đã lựa chọn nền tảng Google Colab và phiên bản YOLOv11n để tiến hành huấn luyện các mô hình với các tham số huấn luyện chính được trình bày ở Bảng 3. Việc lựa chọn các tham số ở Bảng 3 dựa trên cơ sở sự cân bằng giữa khả năng hội tụ, độ chính xác và tài nguyên tính toán. Cụ thể, số vòng huấn luyện (48–90) đảm bảo mô hình học đủ sâu nhưng tránh hiện tượng quá khớp, kích thước ảnh đầu vào (640), tối ưu hóa giữa độ chính xác phát hiện và tốc độ xử lý. Lô ảnh xử lý có kích thước (16) cùng với bộ tối ưu hóa giảm dần độ dốc ngẫu nhiên và tốc độ học ban đầu (0.01) giúp quá trình huấn luyện ổn định và tăng khả năng tổng quát hóa. Tham số quán tính

(0.937) cải thiện tốc độ hội tụ và giảm dao động gradient. Ngoài ra, các hệ số mất mát như mất mát hộp giới hạn (7.5), mất mát phân loại (0.5) và mất mát tiêu điểm phân phối (1.5) được điều chỉnh nhằm ưu tiên độ chính xác định vị và cải thiện chất lượng dự đoán bounding box, từ đó nâng cao hiệu năng tổng thể của mô hình.

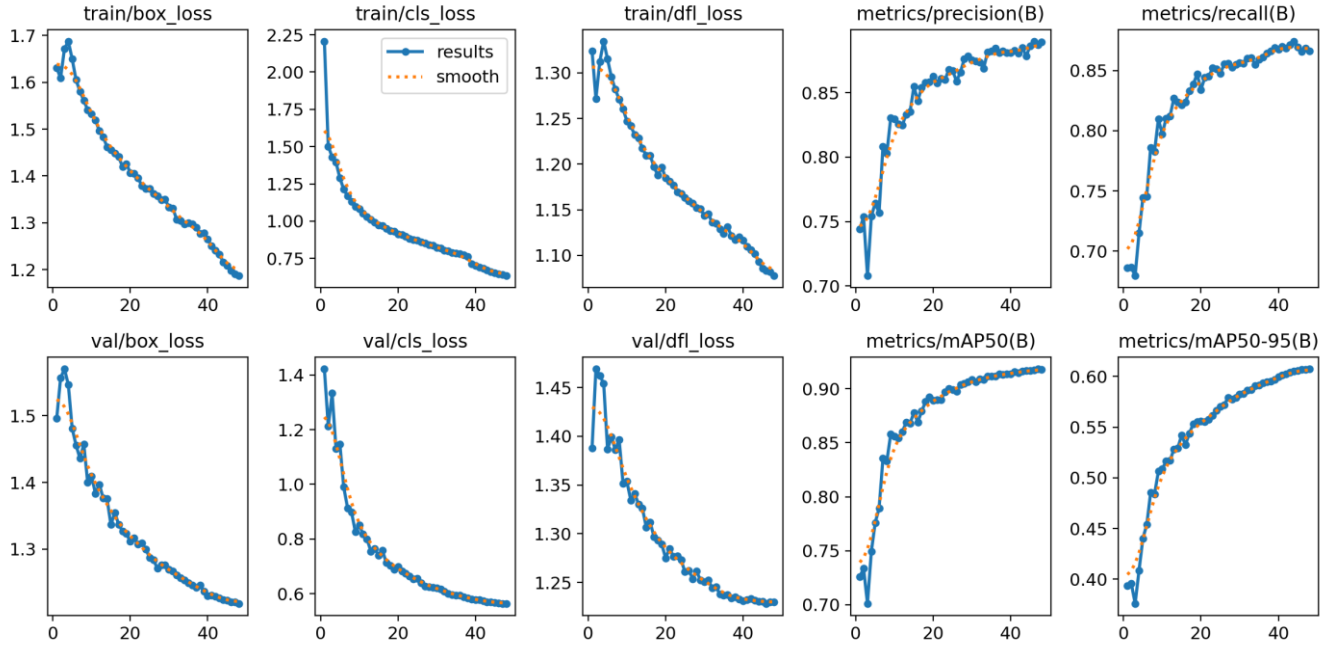
Nghiên cứu đã thực hiện 48 vòng huấn luyện cho bộ dữ liệu phát hiện không đội mũ bảo hiểm, 80 vòng huấn luyện cho bộ dữ liệu khoanh vùng biển số và 90 vòng huấn luyện cho bộ dữ liệu nhận dạng ký tự, các chỉ số của quá trình huấn luyện được trình bày ở Bảng 4 và các đồ thị hiệu suất thể hiện ở các hình 4, 5, 6.

Bảng 4. Kết quả các chỉ số huấn luyện của 3 bộ dữ liệu

Bộ dữ liệu	Precision	Recall	mAP 50	mAP 50-95
Phát hiện người không đội mũ bảo hiểm	0.889	0.867	0.918	0.607
Phát hiện vùng chứa biển số	0.981	0.961	0.982	0.703
Nhận diện ký tự	0.965	0.976	0.984	0.774

Hình 4a là đồ thị thể hiện đường cong hiệu suất huấn luyện của bộ dữ liệu phát hiện người không đội mũ bảo hiểm. Các đồ thị train/val box_loss, cls_loss, dfl_loss đều giảm dần theo số epoch và hội tụ ổn định, chứng tỏ quá trình huấn luyện diễn ra hiệu quả, không xảy ra hiện tượng quá khớp. Đường cong của tập huấn luyện và tập kiểm thử gần như trùng nhau, thể hiện khả năng khái quát hóa tốt của mô hình. Các chỉ số đánh giá hiệu năng như Precision, Recall và mAP đều tăng dần theo thời gian huấn luyện và đạt giá trị cao ở cuối quá trình.

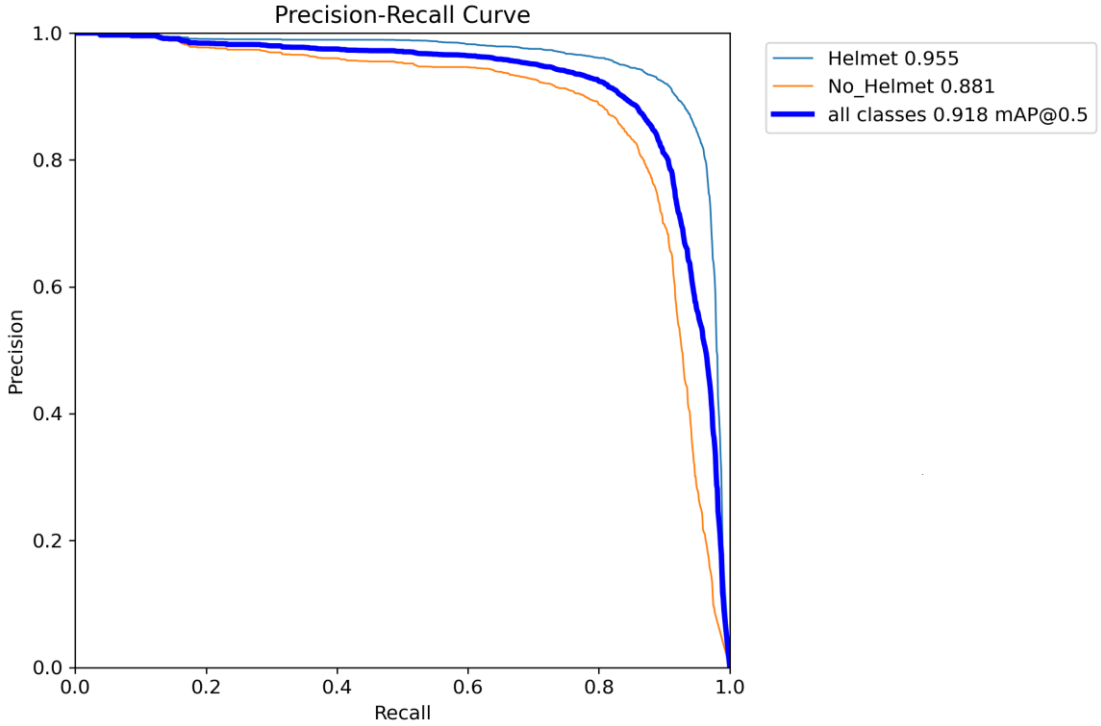
Cụ thể, Precision và Recall của lớp No_Helmet lần lượt 0.889 và 0.867, trong khi mAP@0.5 đạt 0.981 và mAP@0.5:0.95 duy trì ở mức cao, phản ánh mô hình có khả năng phát hiện đối tượng ổn định ở nhiều ngưỡng IoU khác nhau.



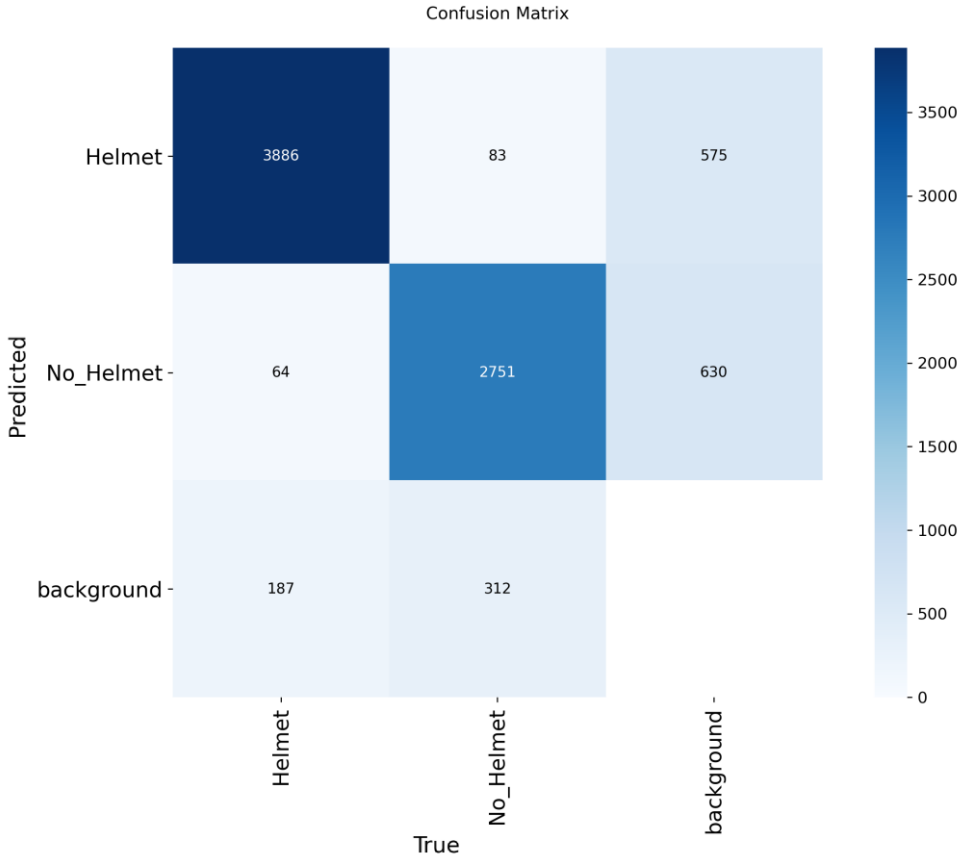
Hình 4a. Đường cong đánh giá hiệu suất huấn luyện của mô hình phát hiện không đội mũ bảo hiểm

Hình 4b thể hiện đường cong Precision-Recall của hai lớp "Helmet" và "No_Helmet". Đường cong của cả hai lớp đều nằm gần góc trên bên phải, chứng tỏ mô hình duy trì độ chính xác cao ngay cả khi tỷ lệ phát hiện lớn. Cụ thể, lớp Helmet đạt mAP@0.5 = 0.955, lớp No_Helmet đạt 0.981, cho thấy mô hình phát hiện hiệu quả cả hai loại đối tượng với sai số thấp. Tuy nhiên dựa vào ma trận nhầm lẫn hình 4c, cho thấy FP = 694, FN = 395, nghĩa là có 694 trường hợp mô hình nhận

nhầm lẫn, trong đó nhận nhầm có mũ thành không mũ là 64 trường hợp và nhầm lẫn với nền (background) là 630. Bỏ sót không nhận được 395 trường hợp không đội mũ, trong đó cũng nhầm với nền là 312 trường hợp. Điều này có nghĩa mô hình chưa học tốt đặc trưng của nền.

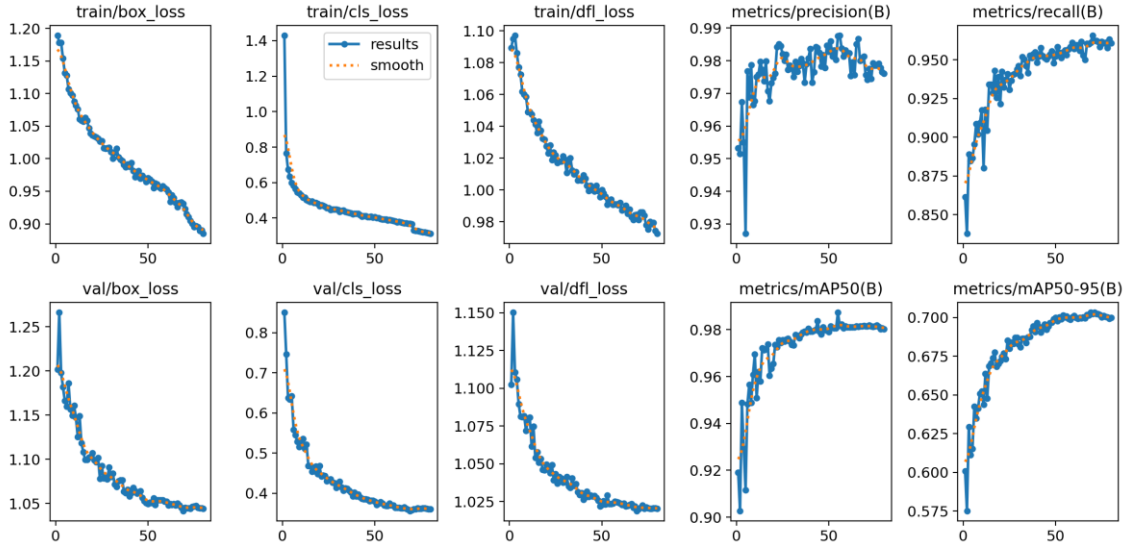


Hình 4b. Đường cong huấn luyện Precision–Recall của mô hình phát hiện không đội mũ bảo hiểm

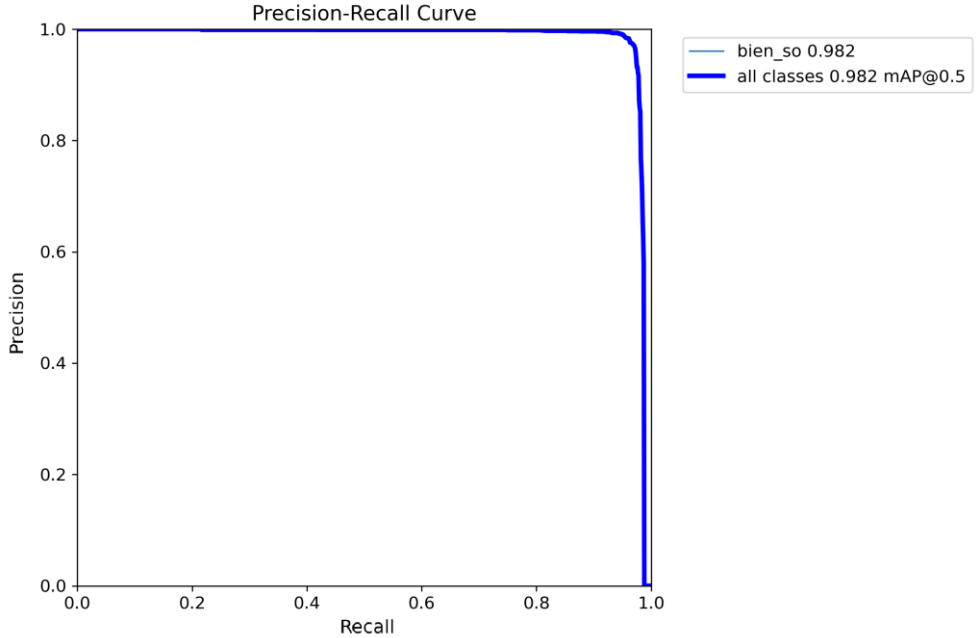


Hình 4c. Ma trận nhầm lẫn của mô hình phát hiện không đội mũ bảo hiểm

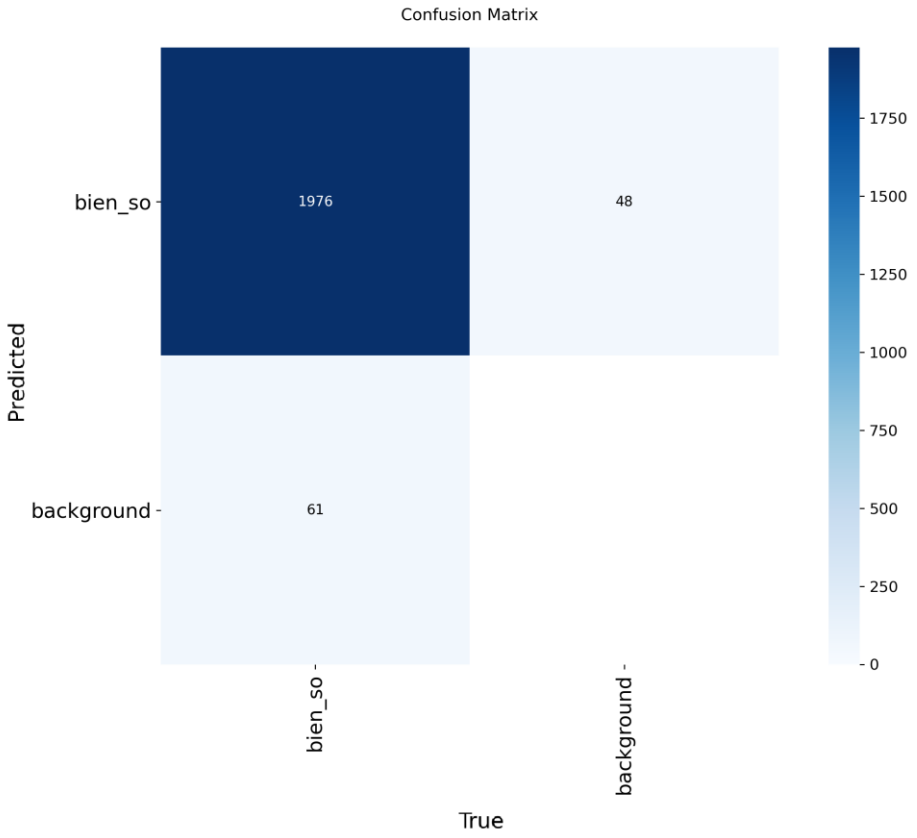
Hình 5 trình bày kết quả huấn luyện mô hình phát hiện vùng chứa biển số. Các đường cong hàm mất mát của tập huấn luyện và tập kiểm thử đều giảm dần theo số vòng huấn luyện và dần hội tụ, quá trình học ổn định và không xuất hiện hiện tượng quá khớp (Hình 5a). Đường cong huấn luyện và kiểm thử phản ánh khả năng khái quát hóa tốt của mô hình. Các chỉ số đánh giá hiệu năng đạt giá trị cao, Precision xấp xỉ 0.98 và Recall đạt khoảng 0.96, cho thấy mô hình vừa đảm bảo độ chính xác cao trong dự đoán, vừa có khả năng phát hiện phần lớn các đối tượng thực. Giá trị mAP@0.5 đạt khoảng 0.982 và mAP@0.5:0.95 đạt xấp xỉ 0.70, phản ánh hiệu suất nhận dạng ổn định ở nhiều ngưỡng IoU khác nhau. Đường cong Precision-Recall có dạng gần vuông tại vùng góc trên bên phải, cho thấy mô hình duy trì độ chính xác cao ngay cả khi Recall lớn (Hình 5b). Ma trận nhầm lẫn (Hình 5c) cho thấy số lượng dự đoán đúng chiếm ưu thế (1976 mẫu đúng), số lượng nhầm lẫn tương đối nhỏ (48 mẫu nhầm sang nền và 61 mẫu nền bị nhận nhầm là biển số). Tuy nhiên, số lượng mẫu nền trong tập đánh giá còn hạn chế, nên dẫn đến việc phân biệt đối tượng và nền chưa thể đánh giá đầy đủ.



Hình 5a. Đường cong đánh giá hiệu suất huấn luyện của mô hình phát hiện vùng chứa biển số

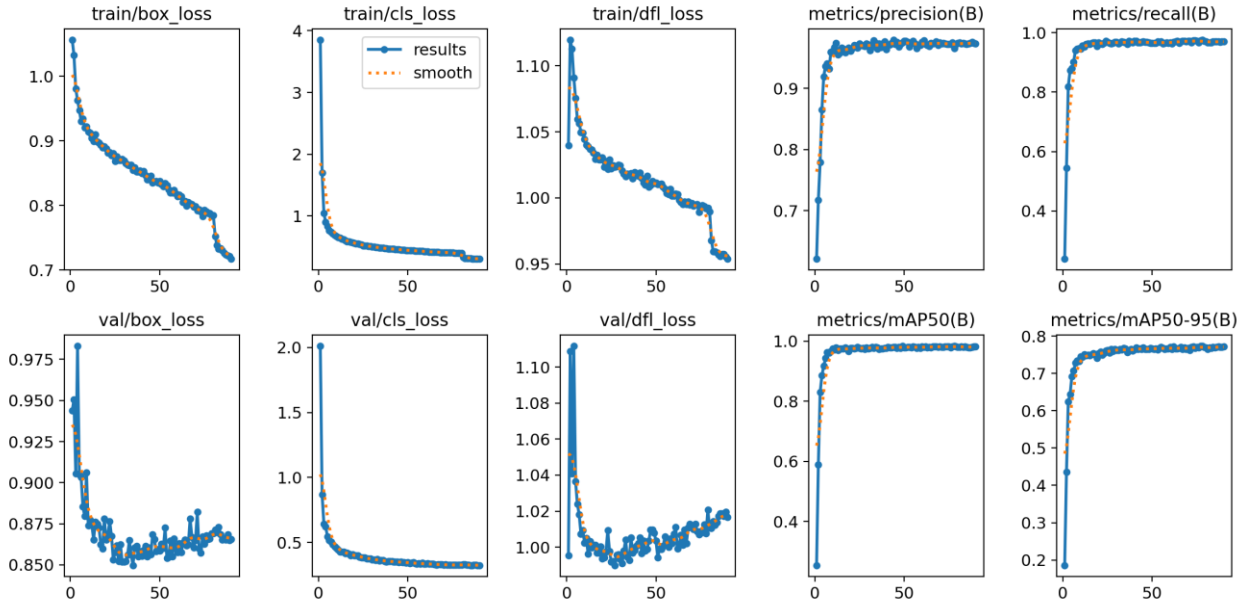


Hình 5b. Đường cong huấn luyện Precision–Recall của mô hình phát hiện vùng chứa biển số



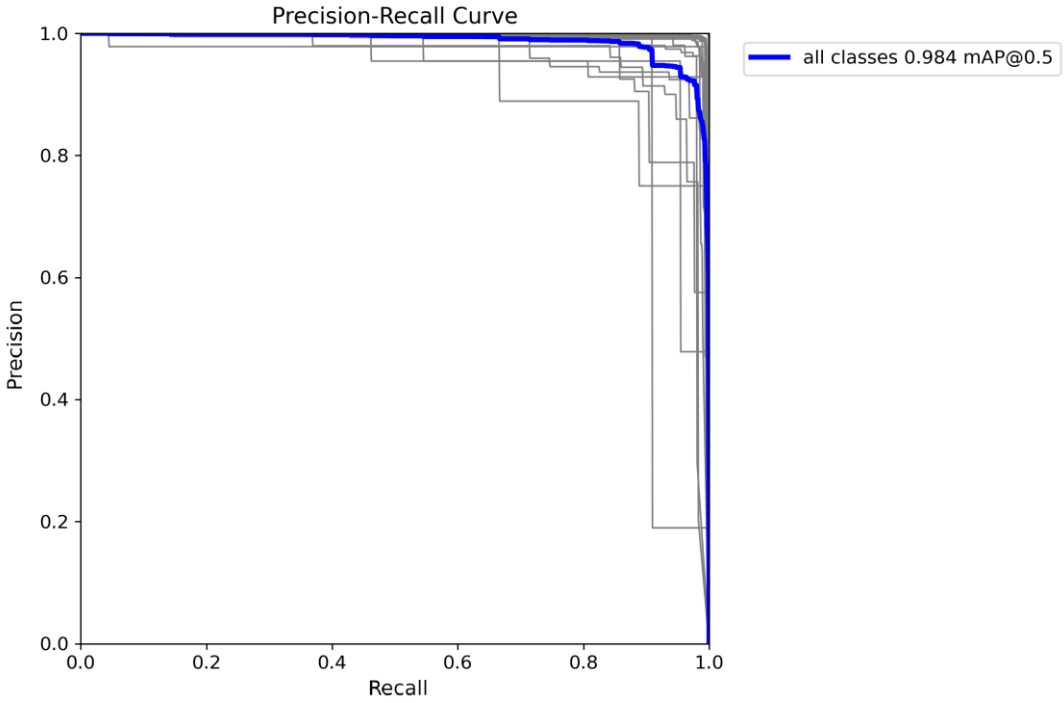
Hình 5c. Ma trận nhầm lẫn của mô hình phát hiện vùng chứa biển số

Hình 6 là đồ thị thể hiện kết quả huấn luyện của bộ dữ liệu nhận dạng ký tự. Các giá trị hàm mất mát của tập huấn luyện và kiểm thử đều giảm ổn định theo số vòng huấn luyện, chứng tỏ mô hình hội tụ tốt và không xuất hiện hiện tượng quá khớp. Các chỉ số đánh giá hiệu năng đạt kết quả cao, với Precision và Recall đều tiến gần tới giá trị 1.0, thể hiện khả năng nhận dạng chính xác và phát hiện đầy đủ các đối tượng trong ảnh. Giá trị mAP@0.5 đạt 0.984, và mAP@0.5:0.95 duy trì ở mức cao 0.774, phản ánh hiệu quả của mô hình ở nhiều ngưỡng IoU khác nhau.

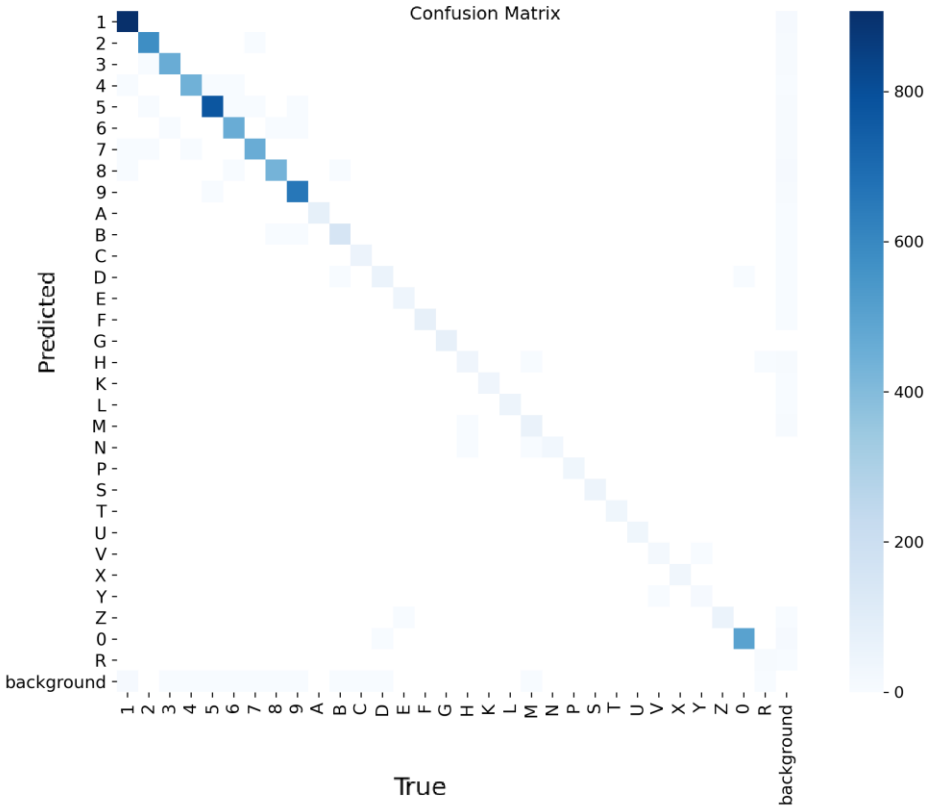


Hình 6a. Đường cong đánh giá hiệu suất huấn luyện của mô hình nhận diện ký tự

Đường cong Precision–Recall được thể hiện ở hình 6b có dạng gần vuông góc ở góc trên bên phải, cho thấy mô hình duy trì được độ chính xác cao ngay cả khi tỷ lệ phát hiện lớn. Ma trận nhầm lẫn như được thể hiện trong hình 6c cho thấy mô hình nhận dạng ký tự đạt độ chính xác cao với phần lớn giá trị tập trung trên đường chéo chính. Tuy nhiên, vẫn tồn tại một số nhầm lẫn giữa các ký tự có hình dạng tương đồng như “0” và “O”, “1” và “I”.



Hình 6b. Đường cong huấn luyện Precision–Recall của mô hình nhận diện ký tự

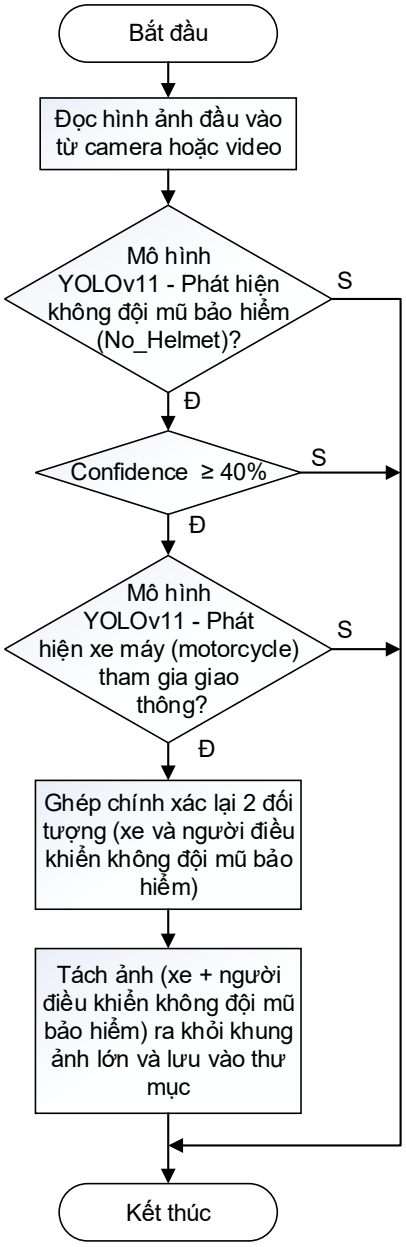


Hình 6c. Ma trận nhầm lẫn của mô hình nhận diện ký tự

Từ những kết quả trên, ta có thể thấy mô hình huấn luyện ứng với các tập dữ liệu trên cho kết quả khá tốt, hiệu suất vượt trội, đảm bảo tính ổn định và độ tin cậy khi ứng dụng trong nhận dạng đối tượng.

3.3. Thực nghiệm mô hình

Trong nghiên cứu này, nhóm nghiên cứu sử dụng ngôn ngữ lập trình Python phiên bản 3.8 và môi trường phát triển tích hợp IDE PyCharm để xây dựng triển khai và kiểm thử mã nguồn, đồng thời PyCharm còn hỗ trợ quản lý dự án, gỡ lỗi và trực quan hóa mã lệnh. Các chương trình điều khiển và cấu trúc thuật toán được thiết kế dựa trên các lưu đồ ở hình 7, hình 8. Kết quả thực nghiệm thể hiện qua ảnh chụp màn hình, như được trình bày ở hình 9.



Hình 7. Lưu đồ thuật toán chương trình phát hiện người không đội mũ bảo hiểm

3.3.1. Lưu đồ thuật toán phát hiện người không đội mũ bảo hiểm

Quy trình phát hiện người đi xe máy không đội mũ bảo hiểm (hình 7) gồm các bước sau:

- Đọc hình ảnh trực tiếp đầu vào từ camera hoặc video từ đường dẫn.

- Sử dụng mô hình YOLOv11 đã được huấn luyện để phát hiện những người tham gia giao thông không đội mũ bảo hiểm. Mô hình này được huấn luyện với hai lớp đối tượng, gồm “Helmet”

và “No_Helmet”. Trong quá trình nhận diện, thuật toán sẽ lựa chọn lớp “No_Helmet” và vẽ khung giới hạn (bounding box) lên vùng đầu của đối tượng không đội mũ bảo hiểm trên ảnh gốc.

- Kiểm tra độ tin cậy về nhận diện không đội mũ bảo hiểm, nếu độ tin cậy bằng hoặc lớn hơn 40% thì thực hiện bước tiếp theo, nếu không thì kết thúc quy trình.

- Sử dụng mô hình YOLOv11n của YOLO để phát hiện xe gắn máy đang tham gia giao thông. Mô hình YOLOv11 được huấn luyện với 80 lớp đối tượng khác nhau, trong đó lớp “xe gắn máy” là lớp số 03, nằm ở vị trí thứ tư trong danh sách các lớp. Khi phát hiện được đối tượng, mô hình sẽ vẽ khung giới hạn bao quanh các xe gắn máy trên ảnh gốc.

- Sau khi chọn được đối tượng xe máy và người điều khiển không đội mũ bảo hiểm từ 2 mô hình riêng; thuật toán ghép chính xác 2 đối tượng lại thành một ảnh hoàn chỉnh người điều khiển xe máy không đội mũ bảo hiểm.

- Cắt vùng hình ảnh người điều khiển xe máy không đội mũ bảo hiểm ra khỏi khung hình lớn và lưu vào thư mục chỉ định, chuẩn bị thực hiện bước tiếp theo là tách biển số và nhận diện ký tự.

Hình 8. Lưu đồ thuật toán chương trình phát hiện biển số và nhận diện ký tự

3.3.2. Lưu đồ thuật toán phát hiện vùng chứa biển số và nhận diện ký tự

Quy trình phát hiện vùng chứa biển số và nhận diện ký tự (hình 8) gồm các bước như sau:

- Sử dụng mô hình đã huấn luyện, tìm vùng giới hạn có chứa biển số và vẽ các khung hình lên ảnh gốc.

- Kiểm tra độ chính xác các vùng chứa biển số vừa phát hiện, nếu bằng hoặc lớn hơn 50% thì tiếp tục thực hiện; ngược lại thì kết thúc quy trình.

- Xác định được vùng chứa biển số có độ chính xác cao nhất, sau đó cắt ra khỏi ảnh gốc.

- Xác định góc nghiêng và điều chỉnh biển số sao cho mô hình nhận diện ký tự hiệu quả nhất.

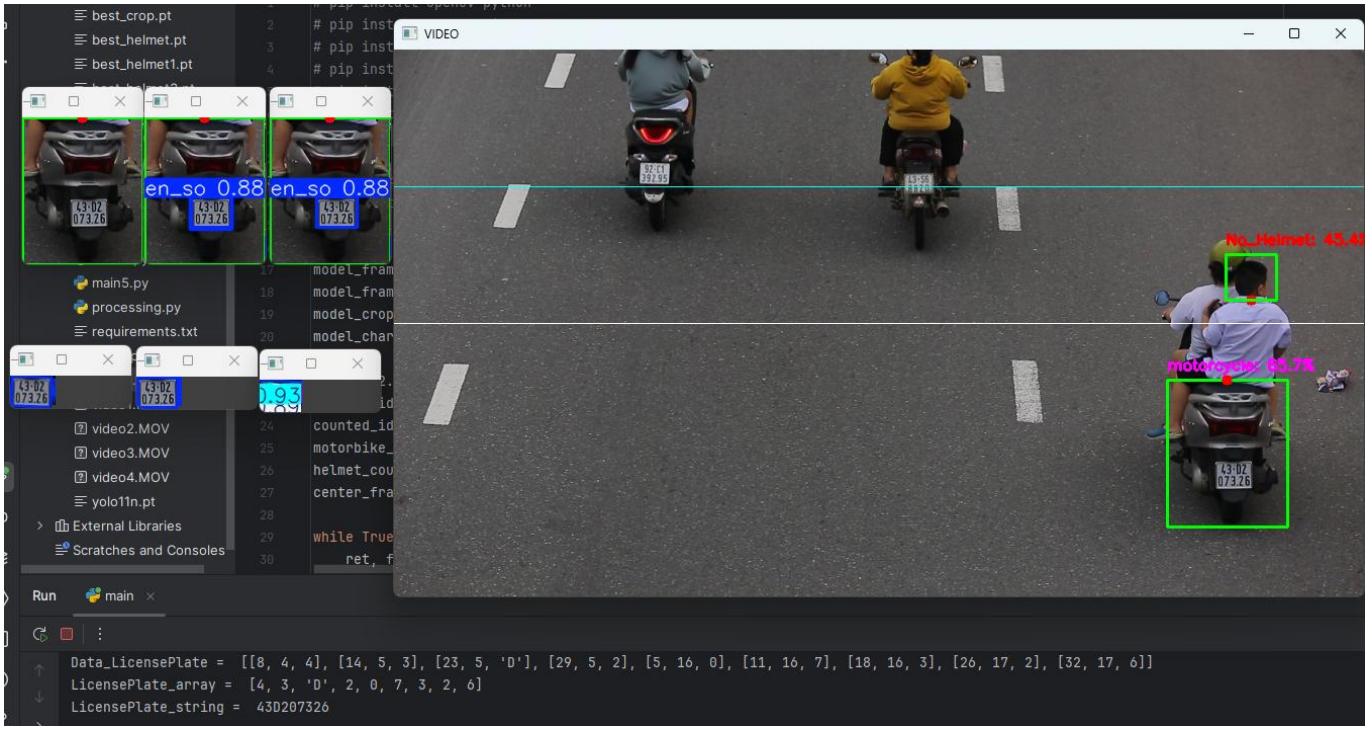
- Nhận dạng từng ký tự trong biển số bằng cách dùng mô hình YOLOv11 đã huấn luyện.

- Kiểm tra độ chính xác các ký tự vừa nhận dạng, nếu độ chính xác bằng hoặc lớn hơn 60% thì tiếp tục thực hiện; nếu không thì quy trình dừng lại.

- Kiểm tra số lượng ký tự hợp lệ, nếu tổng số ký tự lớn hơn 7 và nhỏ hơn 10 thì hợp lệ, tiếp tục bước tiếp theo; nếu không thì kết thúc quy trình.

- Dựa vào tọa độ của các ký tự được phát hiện, chương trình tiến hành ghép nối để tái tạo chuỗi biển số theo đúng thứ tự thực tế.

Đầu tiên, chương trình tính giá trị tọa độ y trung bình của toàn bộ ký tự, từ đó phân chia thành hai nhóm, bao gồm nhóm các ký tự có tọa độ y nhỏ hơn giá trị trung bình được xác định thuộc dòng trên và các ký tự có tọa độ y lớn hơn được xác định thuộc dòng dưới của biển số. Sau khi hoàn thành việc phân tách nhóm trên nhóm dưới, các ký tự trong từng nhóm được sắp xếp theo thứ tự tọa độ x tăng dần, phục hồi chính xác thứ tự các ký tự trên biển số thực và hiển thị kết quả.



Hình 9. Kết quả thực nghiệm quá trình phát hiện, nhận diện và trích xuất biển số xe người tham gia giao thông không đội mũ bảo hiểm

3.3.3. Kết quả thực nghiệm

Mã nguồn được nhóm tác giả chạy trên máy tính cá nhân (personal computer) với cấu hình bao gồm: Bộ vi xử lý Intel(R) Core(TM) i5-3570 CPU @ 3.40GHz 3.80 GHz; Bộ nhớ RAM 8,00 GB; Card đồ họa Intel(R) HD Graphics (32 MB); Hệ điều hành 64-bit, bộ xử lý kiến trúc x64. Đầu vào mô hình là đoạn video giao thông của các xe

máy được quay từ điện thoại Samsung. Dựa vào mô hình YOLOv11 đã huấn luyện và mã code xử lý dữ liệu bằng ngôn ngữ Python, chương trình đã phát hiện, khoanh vùng được người điều khiển xe máy không đội mũ bảo hiểm với tỷ lệ dự đoán lần lượt là 65.7% và 45.5%. Sau đó tiếp tục tìm, khoanh vùng chứa biển số với tỷ lệ dự đoán 88%, tính toán hiệu chỉnh góc nghiêng, xoay biển số, nhận dạng ký tự (tỷ lệ dự đoán trên 90%). Kết quả cuối cùng hệ thống nhận dạng, đọc được biển số xe vi phạm giao thông khá chính xác như hình 9, với thời gian xử lý trung bình khoảng 990.3ms, tương ứng tốc độ xử lý 1.0 FPS.

3.4. Đánh giá và bình luận kết quả

Thử nghiệm mô hình với nhiều đoạn video giao thông của xe máy với tốc độ trung bình khoảng 50km/h. Nhóm tác giả nhận thấy mô hình phát hiện, nhận diện được hầu hết biển số xe máy mà người điều khiển xe không đội mũ bảo hiểm với độ chính xác trên 90%.

Đối với một số trường hợp mà người điều khiển xe đội mũ bảo hiểm màu đen có hình dạng như mũ lưỡi trai thì mô hình nhận diện sai lệch, nhầm tưởng là không đội mũ bảo hiểm như thể hiện ở hình 10. Nguyên nhân chủ yếu là do sự tương đồng về đặc trưng thị giác giữa mũ bảo hiểm màu đen và mũ lưỡi trai trong điều kiện góc quan sát không thuận lợi. Bên cạnh đó, số lượng mẫu mũ bảo hiểm màu đen hoặc các trường hợp dễ gây nhầm lẫn trong tập huấn luyện còn hạn chế, khiến mô hình chưa học được đầy đủ các đặc trưng nhận dạng đặc thù. Trường hợp xe chạy với vận tốc lớn dẫn đến quá trình bắt biển số không rõ như bị mờ nhiều, hoặc nhiễu cao thì mô hình vẫn khoanh vùng được biển số nhưng nhận diện ký tự có thể bị sai hoặc nhận diện không được. Do đó, cần phải bổ sung thêm dữ liệu huấn luyện cho mô hình, để mô hình nhận diện chính xác hơn.

Tốc độ xử lý trung bình đạt 1.0 FPS (Bảng 5) là do hệ thống thực nghiệm đang chạy nối tiếp trên thiết bị cá nhân có cấu hình hạn chế. Nếu triển khai cơ chế xử lý song song, sử dụng các thiết bị có GPU chuyên dụng hoặc tối ưu hóa mô hình trên nền tảng ONNX Runtime thì tốc độ xử lý có thể được cải thiện đáng kể và có khả năng đáp ứng yêu cầu thời gian thực.

Qua các chỉ số hiệu suất huấn luyện mô hình ở các phiên bản YOLO khác nhau (Bảng 6) cho thấy YOLOv11n cải thiện hơn YOLOv8n và YOLOv10n. Cụ thể là các chỉ số (Precision, Recall, mAP@50, mAP@50-95) bằng YOLOv8n và cao hơn YOLOv10n, còn tốc độ xử lý FPS (Frames Per Second – Số khung hình trên giây) thì lớn hơn YOLOv8n và nhỏ hơn YOLOv10n. Điều này cho thấy phiên bản YOLOv11 mang lại sự tối ưu và cân bằng hiệu quả hơn nhờ vào cải tiến trong kiến trúc backbone, cơ chế head phân tách, cùng việc tích hợp các kỹ thuật chú ý hiện đại.

Hình 10. Mô hình nhận nhầm một số mũ bảo hiểm màu đen như mũ lưỡi trai

Bảng 5. Thời gian (ms) thực thi của từng khối và tổng thể của hệ thống

Thời gian xử lý của các khối (ms)	Tiền xử lý	Suy luận	Hậu xử lý	FPS
Phát hiện không đội mũ bảo hiểm	3.0	130.9	1.0	7.4
Xác định phương tiện xe máy	2.0	151.9	2.0	6.4
Phát hiện vùng biển số	5.0	253.8	1.0	3.8
Nhận diện ký tự	2.0	435.7	2.0	2.3
Tổng thời gian	12.0	972.3	6.0	1.0

Bảng 6. Kết quả huấn luyện mô hình phát hiện không đội mũ bảo hiểm, phát hiện biển số và nhận diện ký tự ở các phiên bản YOLO

Mô hình		Precision	Recall	mAP50	mAP50-95	FPS
YOLOv11n	Phát hiện không đội mũ bảo hiểm	0.889	0.867	0.918	0.607	227.3
	Phát hiện vùng biển số	0.981	0.961	0.982	0.703	303.0
	Nhận diện ký tự	0.965	0.976	0.984	0.774	285.7
YOLOv10n	Phát hiện không đội mũ bảo hiểm	0.857	0.858	0.897	0.586	312.5
	Phát hiện vùng biển số	0.972	0.951	0.980	0.698	454.5
	Nhận diện ký tự	0.953	0.950	0.979	0.774	434.8
YOLOv8n	Phát hiện không đội mũ bảo hiểm	0.883	0.874	0.921	0.614	212.7
	Phát hiện vùng biển số	0.977	0.961	0.985	0.700	270.3
	Nhận diện ký tự	0.979	0.963	0.984	0.776	270.3

4. Kết luận

Trong bài nghiên cứu này, nhóm tác giả đã xây dựng và thử nghiệm thành công mô hình phát hiện nhận diện biển số xe người tham gia giao thông không đội mũ bảo hiểm. Dữ liệu được thu thập cho mô hình gồm 19214 hình ảnh về phát hiện người không đội mũ bảo hiểm, 9038 hình ảnh về biển số xe và 4100 hình ảnh về ký tự, sử dụng công cụ Google Colab để huấn luyện tập dữ liệu trên theo phiên bản YOLOv11n của YOLOv11, kết quả huấn luyện đạt các chỉ số cao như: Precision = 0.889 - 0.981, Recall = 0.867 - 0.976, mAP@50 = 0.918 - 0.984, mAP@50-95 = 0.607 - 0.774. Thử nghiệm mô hình với đoạn video giao thông của các loại xe máy trong điều kiện bình thường (vận tốc dưới 50km/h) làm dữ liệu đầu vào, kết quả mô hình nhận diện khá chính xác (gồm phát hiện người tham gia giao thông không đội mũ bảo hiểm, phát hiện phương tiện giao thông xe máy, xác định vùng chứa biển số và phân loại nhận dạng được ký tự), với tỷ lệ dự đoán đúng ký tự trên 90% và thời gian xử lý chấp nhận được, phù hợp với ứng dụng giám sát phát hiện vi phạm giao thông, nơi mà độ chính xác được ưu tiên hơn tốc độ xử lý tức thời. Tuy nhiên, mô hình vẫn nhận diện nhầm lẫn đối với các mũ bảo hiểm màu đen (giống như mũ lưỡi trai) là không đội mũ, nhận diện không được đối với ký tự biển số bị mờ nhiều. Do đó, trong thời gian tới, nghiên cứu sẽ tiếp tục phát triển một số nội dung nhằm nâng cao độ chính xác nhận diện và cải thiện tốc độ xử lý của mô hình, cụ thể là:

- Mở rộng bộ dữ liệu huấn luyện cho bài toán phát hiện không đội mũ bảo hiểm theo hướng đa dạng về kiểu dáng, màu sắc mũ và bổ sung các trường hợp dễ gây nhầm lẫn; đồng thời áp dụng kỹ thuật tăng cường dữ liệu, fine-tuning trên các mẫu khó và tích hợp cơ chế chú ý nhằm nâng cao khả năng phân biệt và giảm dự đoán sai. Đối với bài toán nhận diện ký tự, cần bổ sung thêm dữ liệu cho các trường hợp ký tự nhỏ, mờ hoặc khó phân biệt để cải thiện độ chính xác dự đoán. Bên cạnh đó, cần nghiên cứu tích hợp nhiều mô hình thành kiến trúc đa nhiệm, triển khai

quy trình xử lý song song hoặc tối ưu trên nền tảng ONNX Runtime để cải thiện tốc độ xử lý, nâng cao chỉ số FPS. Ngoài ra, nghiên cứu sẽ tiếp tục thực hiện các thực nghiệm cắt lớp trên các mô-đun cải tiến của YOLOv11 như C3k2 và C2PSA để đánh giá định lượng mức độ đóng góp của từng thành phần đối với hiệu năng phát hiện cho bộ dữ liệu. Kết quả này sẽ giúp hoàn thiện cơ sở thực nghiệm và định hướng tối ưu kiến trúc cho các nghiên cứu tiếp theo.

- Nghiên cứu có thể triển khai, áp dụng ở các trường trung học phổ thông, trường trung cấp, cao đẳng,… để có thể giám sát, theo dõi và chấn chỉnh kịp thời những học sinh, sinh viên vi phạm, tham gia giao thông không đội mũ bảo hiểm.

Tài liệu tham khảo

[1] N.V. Mạnh, L.V. An, T.Đ. Lương, N.T. Tú, N.T.T. Huyền, N.T.B. Thúy, N.T. Cường. (2024). Nghiên cứu và ứng dụng thuật toán YOLOv7 để phân loại cà chua. *Tạp chí Khoa học Tài nguyên và Môi trường*, 50, 102-112. https://doi.org/10.63064/khtnmt.2024.569

[2] G. Lavanya, S.D. Pande. (2023). Enhancing Real-time Object Detection with YOLO Algorithm. *EAI Endorsed Transactions on Internet of Things*, 10(2024), 1-9. https://doi.org/10.4108/eetiot.4541

[3] Đ.T. Dung, H.L.N. Dung, T.L. Chương, T.C. Hào, T.V. Phúc. (2024). Nghiên cứu các phiên bản YOLOv8 và YOLO-NAS trong phát hiện biển số xe. *Tạp chí Khoa học và Công nghệ Đại học Thái Nguyên*, 229(07), 156-167. https://doi.org/10.34238/tnu-jst.10336

[4] K. Patel, V. Patel, V. Prajapati, D. Chauhan, A. Haji, S. Degadwala. (2023). Safety Helmet Detection using YOLO V8. *2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN),* pp. 22-26. https://doi.org/10.1109/ICPCSN 58827.2023.00012

[5] C. Srusti, V. Deo, R.C. Jaiswal. (2022). Helmet Detection Using Machine Learning. *Journal of Emerging Technologies and Innovative Research (JETIR)*, 9(10), d10-d17. https://www.jetir.org/view?paper=JETIR2210312

[6] A. Aboah, B. Wang, U. Bagci, Y. Adu-Gyamfi. (2023). Real-time Multi-Class Helmet Violation Detection Using Few-Shot Data Sampling Technique and YOLOv8. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),* pp. 5350-5358. https://doi.org/10.1109/CVPRW59228.2023.00564

[7] X. Cong, S. Li, F. Chen, C. Liu, Y. Meng. (2023). A Review of YOLO Object Detection Algorithms based on Deep Learning. *Frontiers in Computing and Intelligent Systems*, 4(2), 17-20. https://doi.org/10.54097/fcis.v4i2.9730

[8] E. Soltanikazemi, A. Dhakal, B.K. Hatuwal, I.E. Toubal, A. Aboah, K. Palaniappan. (2023). Real-Time Helmet Violation Detection in AI City Challenge 2023 with Genetic Algorithm-Enhanced YOLOv5. *2023 IEEE Applied Imagery Pattern Recognition Workshop (AIPR),* pp.69-78. https://doi.org/10.1109/AIPR60534.2023.10440713

[9] S.R. Kura, S. Rathlavath, B.S.R. Lingannagari, S.K.R. Golla, A. Vislavath, H. Kanakadurga, N. Singh. (2024). Helmet and number plate detection. *MATEC Web of Conferences,* 392, 1-6. https://doi.org/10.1051/matecconf/202439201096

[10] D. Prajapati, S. Sabat, S. Bhilare, R. Vishe, S. Bhujbal. (2024). Helmet Detection and Number Plate Recognition Using YOLOv8 and TensorFlow Algorithm in Machine Learning. *International Journal of Innovative Research in Computer Science and Technology (IJIRCST),* 12(2), 91-95. https://doi.org/10.55524/ijircst.2024.12.2.16

[11] K. Paul, K. Singh, S. Sharma. (2024). Efficient Helmet and Number Plate Detection using YOLOv3: A Road Safety Perspective. *International Journal for Research in Applied Science and Engineering Technology*, 12(IV),

4208-4214. https://doi.org/10.22214/ijraset.2024.60872

[12] J. Redmon, S. Divvala, R. Girshick, A. Farhadi. (2016). You Only Look Once: Unified, Real-Time Object Detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 779–788. https://doi.org/10.1109/CVPR .2016.91

[13] N.Đ. Thiện, L.T. Thắng, N.V. Thặng, T.Q. Bảo. (2022). Nghiên cứu thuật toán thị giác máy tính để theo dõi việc thực hiện các quy định về phòng ngừa Covid-19 sử dụng kỹ thuật học sâu. *Tạp chí Khoa học trường Đại học Cần Thơ,* 58(5), 1-13. https://doi.org/10.22144/ ctu.jvn.2022.217

[14] R. Khanam, M. Hussain. (2024). YOLOv11: An overview of the key Architectural Enhancements. arXiv preprint. arXiv:2410.17725. https://doi.org/10.48550/arXiv.2410.17725

[15] P. Hidayatullah, N. Syakrani, M.R. Sholahuddin, T. Gelar, R. Tubagus. (2026). YOLOv8 to YOLO11: A Comprehensive Architecture In-depth Comparative Review. arXiv preprint arXiv:2501.13400 [cs.CV]. https://doi.org/10.48550/arXiv.2501.13400

[16] M.C. Hiếu, T.Q. Trường, L.T. Anh. (2025). Mô hình học sâu phát hiện và nhận diện mã container áp dụng trong vận hành cảng thông minh. *Tạp chí Vật liệu & Xây dựng*, 15(3), 191-197. https://doi.org/10.54772/jomc.03.2025.995

[17] Roboflow Universe, Helmet Detection, Version 1. https://universe.roboflow.com/bchamomoyolo/helmet_detection-mq83z/dataset/1, 2025 (accessed 20 August 2025)