

NGHIÊN CỨU VÀ ỨNG DỤNG THUẬT TOÁN YOLOv11 ĐỂ NHẬN DIỆN BIỂN SỐ XE

Phạm Như Phẩm

Trường Cao đẳng Gia Lai

Email: phamnhupham@gmail.com

Ngày nhận bài: 7/12/2024; ngày hoàn thành phản biện: 16/12/2024; ngày duyệt đăng: 20/12/2024

TÓM TẮT

Trí tuệ nhân tạo, học sâu (deep learning), lấy ý tưởng từ mạng nơ ron thần kinh con người để xây dựng các thuật toán có thể suy nghĩ. Các mô hình học sâu của trí tuệ nhân tạo có khả năng học hỏi từ dữ liệu, xử lý và đưa ra các quyết định một cách tự động nhanh chóng. Bài báo này nghiên cứu và ứng dụng mô hình YOLOv11 trong nhận diện biển số xe. Thuật toán YOLOv11 được huấn luyện lại trên tập dữ liệu tùy chỉnh bao gồm hình ảnh biển số và kí tự, nhằm phát hiện và phân tích chính xác các đối tượng. Kết quả cho thấy mô hình đạt độ chính xác cao (mAP@50 trên 99%), thời gian xử lý nhanh, chứng minh tính khả thi khi triển khai trong các hệ thống thực tế như bãi giữ xe thông minh.

Từ khóa: học sâu, huấn luyện mô hình, nhận diện biển số xe, trí tuệ nhân tạo, YOLOv11.

1. GIỚI THIỆU

Phát hiện đối tượng theo thời gian thực là một chủ đề rất quan trọng trong hệ thống thị giác máy tính. Nó thường được ứng dụng trong các lĩnh vực như: theo dõi đối tượng, lái xe tự động, xử lý hình ảnh y khoa, giám sát giao thông thông minh [1, 2]. Nhận diện biển số xe bằng công nghệ xử lý ảnh là bài toán đã thu hút nhiều nhà nghiên cứu và cũng có nhiều công trình khoa học được công bố. Mặc dù đã có sự tiến bộ đáng kể trong công nghệ nhận diện xử lý hình ảnh, nhưng nhận diện biển số xe vẫn là một thách thức đối với các hệ thống giám sát giao thông, giám sát các bãi giữ xe thông minh, vì có nhiều yếu tố gây khó khăn trong quá trình nhận diện biển số, chẳng hạn như thay đổi về ánh sáng, góc nhìn, góc nghiêng, độ mờ, bóng tối và kích thước biển số xe [3, 4].

Phân tích hình ảnh chứa biển số xe và nhận diện biển số đòi hỏi phải có thuật toán phức tạp, đạt độ chính xác cao. Thuật toán phát hiện đối tượng YOLO (You Only

Look Once)[5], sử dụng trí tuệ nhân tạo học sâu để phát hiện, phân loại đối tượng một cách nhanh chóng, chính xác. Do đó YOLO đã được nhiều công trình khoa học [6, 7, 8] nghiên cứu đề xuất sử dụng mô hình YOLO để nhận diện biển số xe.

Nghiên cứu này tập trung vào tìm hiểu, nghiên cứu thuật toán YOLOv11 và tối ưu hóa thuật toán để huấn luyện lại mô hình theo bộ dữ liệu riêng của nhóm tác giả, từ đó có khả năng nhận diện biển số xe một cách chính xác, nhanh chóng và được áp dụng vào các bãi giữ xe tự động.

2. CÁC NGHIÊN CỨU LIÊN QUAN

Hiện nay, bài toán phát hiện và nhận diện biển số xe, có rất nhiều bài báo công bố kết quả nghiên cứu. Mỗi nghiên cứu dựa vào các thuật toán riêng, kết quả cho ra ưu điểm và một số hạn chế khác nhau. Nhận diện biển số xe đối với bài nghiên cứu [9] có hạn chế là độ chính xác của kết quả nhận diện phụ thuộc vào ánh sáng môi trường, lý do chương trình xây dựng chỉ dựa trên thuật toán xử lý ảnh bằng OpenCV, chưa sử dụng trí tuệ nhân tạo vào mô hình phát hiện biển số.

Đối với bài nghiên cứu [10], sử dụng thuật toán trí tuệ nhận tạo YOLOv5 để huấn luyện bộ dữ liệu gồm 12.500 bức ảnh biển số xe, để phát hiện biển số xe. Mô hình huấn luyện đạt kết quả khá cao, thể hiện bảng 1

Bảng 1. Kết quả huấn luyện mô hình dự đoán phát hiện biển số xe [10]

Algorithms	Recall	Precision	mAP*
Faster R-CNN	86.5	89.4	92.3
SSD300	88.3	89.6	93.3
RPnet	95.2	94.8	94.2
YOLOv5	93.5	93.4	94.6
YOLOv5-1	95.4	95.2	95.8
YOLOv5-LSE	96.5	97.4	97.1

*mAP(mean Average Precision): độ chính xác trung bình.

Trong bài nghiên cứu [11], cũng sử dụng thuật toán trí tuệ nhận tạo YOLOv8 để huấn luyện mô hình phát hiện biển số xe và đọc ký tự trên biển số xe. Bộ dữ liệu gồm 2.528 bức ảnh biển số xe oto Saudi, được chia theo tỉ lệ huấn luyện (70%, 1.769 ảnh), kiểm định (20%, 506 ảnh) và thử nghiệm (10%, 253 ảnh). Ảnh ký tự biển số gồm 1.849 ảnh, trong đó huấn luyện (70%, 1,295 ảnh), kiểm định (20%, 370 ảnh) và thử nghiệm

(10%, 184 ảnh). Các chỉ số huấn luyện của mô hình cũng đạt kết quả cao hơn so với các phiên bản YOLO trước, thể hiện bảng 2

Bảng 2. Kết quả dự đoán phát hiện biến số và nhận diện kí tự [11]

Kết quả dự đoán phát hiện biến số			Kết quả dự đoán nhận diện kí tự		
Model	mAP@0.5	mAP@0.95	Model	mAP@0.5	mAP@0.95
YOLOv8x	0.973	0.844	YOLOv8x	0.981	0.827
YOLOv7x	0.920	0.730	YOLOv7x	0.977	0.711
YOLOv5x	0.994	0.892	YOLOv5x	0.978	0.819

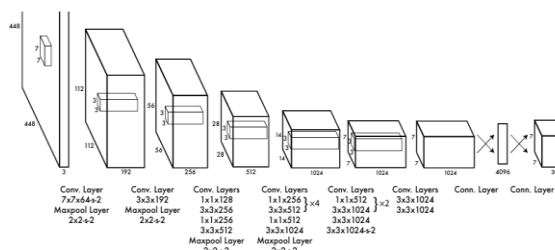
Qua 3 công trình nghiên cứu trên, nhóm tác giả nhận thấy sử dụng trí tuệ nhân tạo thuật toán YOLO để nhận diện biến số xe là chính xác và tối ưu nhất. Để thuật toán nhận diện chính xác thì bộ dữ liệu phải phong phú về số lượng và đa dạng về hình ảnh, góc nhìn, góc nghiêng, độ mờ và kích thước. Vì vậy nhóm tác giả tập trung nghiên cứu thuật toán YOLOv11 để ứng dụng trong việc nhận diện biến số xe.

3. ĐỐI TƯỢNG VÀ PHƯƠNG PHÁP NGHIÊN CỨU

3.1. Nhận diện vật thể đối với YOLO

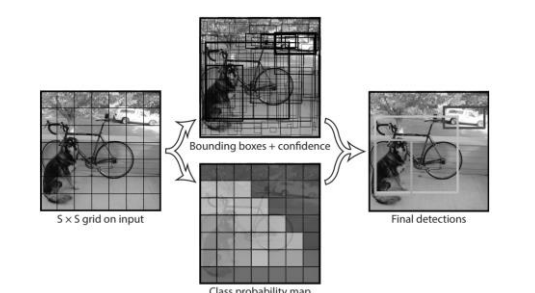
Kiến trúc YOLO (hình 1) là một mô hình mạng học sâu nơ-ron tích chập. Các lớp tích chập ban đầu của mạng trích xuất các đặc điểm từ hình ảnh, các lớp được kết nối dự đoán xác suất, tọa độ và cho kết quả đầu ra. Mạng YOLO có 24 lớp tích chập, theo sau là 2 lớp kết nối đầy đủ, thường sử dụng lớp 1x1 thay lớp tích chập 3x3. Với 9 lớp tích chập thay thế cho 24 lớp và một số bộ lọc trong các lớp đó, sẽ được mạng nơ-ron của Fast YOLO [5, 12].

Mạng YOLO được ứng dụng để phát hiện, nhận diện đối tượng vật thể, nó hoạt động như một bài toán hồi quy. Từ hình ảnh đầu vào, qua một mạng gồm các lớp tích chập, tổng hợp và kết nối đầy đủ có thể cho được kết quả đầu ra. Kiến trúc này có thể được tối ưu để chạy trên GPU (Graphics Processing Unit) với một lần



Hình 1. Kiến trúc của mạng YOLO [1, 5, 12, 13].

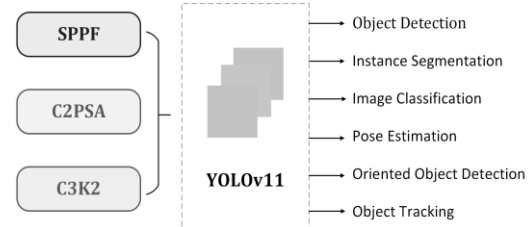
thường sử dụng lớp 1x1 thay lớp tích chập 3x3. Với 9 lớp tích chập thay thế cho 24 lớp và một số bộ lọc trong các lớp đó, sẽ được mạng nơ-ron của Fast YOLO [5, 12].



Hình 2. Mô hình tổng quát YOLO [1, 5, 13].

chuyên tiếp, vì thế nó đạt được tốc độ rất cao. Thuật toán YOLO có độ chính xác cao và nhanh cho những bài toán nhận diện đối tượng và thích hợp cho các ứng dụng thị giác máy [13].

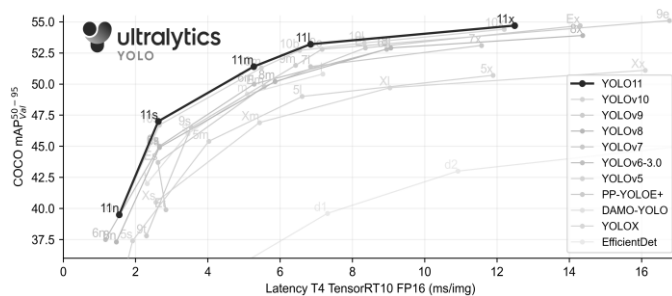
Đầu vào của mô hình YOLO là ảnh (hình 2), mô hình sẽ nhận dạng ảnh đó gồm có những đối tượng nào, sau đó xác định tọa độ của đối tượng trong bức ảnh. Ảnh đầu vào được chia thành thành $S \times S$ ô, thường thì sẽ là 3×3 , 7×7 , 9×9 Đầu ra mô hình là một ma trận 3 chiều có kích thước $S \times S \times (B \times 5 + C)$ với số lượng tham số mỗi ô là $(B \times 5 + C)$. Trong đó, B là số lượng bounding boxes + confidence và C là số class probabilities cho mỗi box cần phải dự đoán. Bounding box gồm 5 thành phần (x, y, w, h, confidence), trong đó (x, y) là tọa độ tâm của bounding box, (w, h) lần lượt là chiều rộng và chiều cao của bounding box và confidence là độ tự tin của dự đoán [1, 5, 13]



Hình 3. Các kiến trúc mô-đun chính của YOLOv11 [15].

3.2. Kiến trúc của YOLOv11

YOLOv11 là phiên bản mới của Ultralytics YOLO, được xây dựng trên nền tảng các phiên bản YOLO trước đó [14]. Được công bố tại hội nghị YOLO Vision vào tháng 9 năm 2024. YOLOv11 đánh dấu một bước tiến đáng kể trong công nghệ phát hiện đối tượng theo thời gian thực.



Hình 4. Biểu đồ so sánh hiệu năng của YOLOv11 với các phiên bản YOLO trước [14,15].

Phần lõi của kiến trúc YOLOv11 bao gồm ba thành phần cơ bản (hình 3), phần thứ nhất là Backbone đóng vai trò là một thành phần quan trọng của kiến trúc YOLO, chịu trách nhiệm trích xuất chính, sử dụng mạng nơ-ron tích chập để chuyển đổi dữ liệu hình ảnh thô đầu vào thành bản đồ tính năng đa tỷ lệ. Thứ hai là thành phần Neck hoạt động như một giai đoạn xử lý trung gian, sử dụng các lớp chuyên biệt để tổng hợp và tăng cường các tính năng biểu diễn trên các tỷ lệ khác nhau. Thứ ba, thành phần Head hoạt động như cơ chế dự đoán, tạo ra các đầu ra cuối cùng để định vị và phân loại đối tượng dựa trên các bản đồ tính năng đã tinh chỉnh [15]

Phiên bản mới cải tiến đáng kể cả về kiến trúc lẫn phương pháp huấn luyện, mở rộng giới hạn của độ chính xác và hiệu suất. Thiết kế đổi mới của YOLOv11 kết

hợp các kỹ thuật trích xuất đặc trưng tiên tiến, cho phép nắm bắt các chi tiết tinh tế hơn trong khi vẫn giữ số lượng tham số gọn nhẹ. Điều này giúp cải thiện độ chính xác trong thị giác máy tính từ phát hiện đối tượng đến phân loại. Đặc biệt YOLOv11 đạt được những cải tiến đáng kể về tốc độ xử lý, tăng cường đáng kể khả năng hoạt động theo thời gian thực [14].

Đồ thị so sánh các mô hình huấn luyện của YOLOv11 được mô tả ở hình 4 và bảng mô tả hiệu suất thể hiện ở bảng 3. Các mô hình YOLOv11 (11n, 11s, 11m và 11x) tạo thành một ranh giới hiệu suất riêng biệt, với mỗi mô hình đều đạt COCO mAP 50-95 cao hơn ở các mức trễ tương ứng các bản YOLO trước đó. Đáng chú ý YOLOv11x đạt được khoảng 54,5% mAP 50-95 ở độ trễ 13ms, vượt qua tất cả các mô hình YOLO trước đó. YOLOv11m thể hiện sự vượt trội hiệu quả, đạt độ chính xác hơn so với các thể hệ YOLO trước đó. Hiệu suất trong chế độ trễ thấp (2-6ms) thì mô hình YOLOv11s duy trì độ chính xác cao khoảng 47% mAP 50-95 [15].

Bảng 3. Hiệu suất của một số mô hình huấn luyện sẵn của YOLOv11 [14]

Model	size	mAP ^{val}	Speed	Speed	params	FLOPs
	(pixels)	(50-95)	CPU ONNX (ms)	T4 TensorRT10 (ms)	(M)	(B)
YOLO11n	640	39.5	56.1 ± 0.8	1.5 ± 0.0	2.6	6.5
YOLO11s	640	47.0	90.0 ± 1.2	2.5 ± 0.0	9.4	21.5
YOLO11m	640	51.5	183.2 ± 2.0	4.7 ± 0.1	20.1	68.0
YOLO11l	640	53.4	238.6 ± 1.4	6.2 ± 0.1	25.3	86.9
YOLO11x	640	54.7	462.8 ± 6.7	11.3 ± 0.2	56.9	194.9

3.3. Quy trình nhận diện biển số xe với YOLOv11

Ứng dụng YOLOv11, nhóm tác giả xây dựng mô hình bài toán nhận diện biển số xe, gồm các khối chức năng chính (hình 5) như sau:



Hình 5. Sơ đồ khối các bước nhận diện biển số xe.

- Khối thứ nhất: Là khối phát hiện biển số xe, làm nhiệm vụ đọc hình ảnh thông qua video, camera hoặc các ảnh về các phương tiện giao thông có chứa biển số xe, từ đó sử dụng mô hình huấn luyện YOLOv11 để phát hiện và khoanh vùng vị trí có chứa biển số trong những bức ảnh hoặc video đó, sau đó cắt vùng chứa biển số đưa vào khối tiếp theo để xử lý.

- Khối thứ hai: Là khối xoay biến số, làm nhiệm vụ đọc tọa độ của 2 ký tự ngoài cùng dòng dưới (hay dòng trên) của biển số, tính góc nghiêng của biển số so với phương ngang, sau đó xoay biến số theo chiều hợp lý để các ký tự của biển số xe vuông góc với phương ngang nhiều nhất có thể.

- Khối thứ ba: Là khối nhận dạng ký tự, sử dụng mô hình huấn luyện YOLOv11 để phân tách ký tự trong biển số, từ đó nhận dạng từng ký tự, cuối cùng trả ra kết quả là khoảng vùng ký tự và tên của ký tự đó trên biển số.

4. KẾT QUẢ VÀ THẢO LUẬN

4.1. Bộ dữ liệu huấn luyện

Bộ dữ liệu huấn luyện của YOLOv11 có hai thư mục lớn là images và labels. Trong mỗi thư mục lớn sẽ chứa 2 thư mục con là thư mục train và val. Dữ liệu được chia với quy tắc tỷ lệ dữ liệu trong thư mục train chiếm khoảng 80% và thư mục val chiếm khoảng 20%. Thư mục images chứa ảnh của dữ liệu, thư mục labels sẽ chứa các nhãn của các ảnh dữ liệu trên. Dữ liệu hình ảnh có kích thước theo quy định của YOLO là 640x640 pixels và được gán nhãn bằng phần mềm Makesense.

Trong mô hình nhận diện biển số xe này, tác giả đã xây dựng 2 bộ dữ liệu như hình 6. Một là bộ dữ liệu phát hiện biển số xe, gồm có 8.855 bức ảnh về biển số các loại xe, trong đó thư mục train chứa 7.050 ảnh và thư mục val chứa 1.805 ảnh. Hai là bộ dữ liệu nhận diện ký tự biển số gồm có 4.025 bức ảnh các ký tự của biển số, trong đó thư mục train chứa 3.210 ảnh và thư mục val chứa 815 ảnh. Để tăng cường sự đa dạng và số lượng hình ảnh của bộ dữ liệu phát hiện biển số tác giả sử dụng OpenCv để sao chép tạo thêm các ảnh (có đường biên, độ sáng, góc nghiêng, phóng to, thu nhỏ, dịch chuyển lệch tâm,...) khác với ảnh ban đầu.

4.2. Huấn luyện mô hình

Các mô hình huấn luyện sẵn của YOLOv11 (11n, 11s, 11m và 11x) không phát hiện được đối tượng trong nghiên cứu này là nhận diện vị trí biển số trong ảnh và nhận diện ký tự của biển số. Vì vậy, tác giả cần phải huấn luyện lại mô hình với dữ liệu đầu vào là tập hợp các hình ảnh của biển số và ký tự.



Hình 6. Ảnh đại diện của bộ dữ liệu phát hiện biển số và bộ dữ liệu đọc ký tự biển số.

Để việc huấn luyện mô hình diễn ra nhanh chóng, hiệu quả, đòi hỏi máy tính phải có cấu hình mạnh, đặc biệt có bộ phận xử lý đồ họa GPU để xử lý ảnh. Do đó chúng tôi sử dụng công cụ miễn phí Google Colab của Google và mô hình YOLOv11n của YOLOv11 để huấn luyện dữ liệu. Lệnh thực thi huấn luyện trên Google Colab có cấu trúc `!yolo train model=yolo11n.pt data=coco8.yaml epochs=90 imgsz=640`. Các tham số chính model, data, epochs, imgsz được thiết lập phù hợp với mục đích và yêu cầu của YOLO. Ngoài ra, còn có một số tham số khác đã được Google Colab thiết lập mặc định như learning rate (lr=0.01), batch size=16, optimizer=auto (SGD), momentum=0.937, box=7.5, cls=0.5, dfl=1.5, patience=100, device=None, weight_decay=0.0005, ...

Trong các tham số trên, epochs là siêu tham số quan trọng ảnh hưởng trực tiếp đến hiệu suất học máy. Nếu số epochs quá ít, mô hình có thể chưa học đủ các đặc trưng của dữ liệu, do đó độ chính xác thấp trên cả tập huấn luyện và tập kiểm định, dẫn đến hiện tượng underfitting. Ngược lại, nếu số epochs quá nhiều mà không có cơ chế kiểm soát, mô hình dễ bị overfitting, tức mô hình học quá kỹ vào dữ liệu huấn luyện và giảm khả năng tổng quát hóa trên dữ liệu kiểm định. Cơ chế dừng sớm (early stopping) là một kỹ thuật quan trọng giúp ngăn ngừa overfitting bằng cách theo dõi hiệu suất của mô hình trên tập kiểm định trong quá trình huấn luyện. Khi hiệu suất mô hình không còn cải thiện sau một số lượng epochs nhất định (patience), quá trình huấn luyện sẽ tự động dừng lại. Việc này giúp mô hình dừng lại ở thời điểm tối ưu, tiết kiệm thời gian huấn luyện, tránh việc tiếp tục học dẫn đến overfitting.

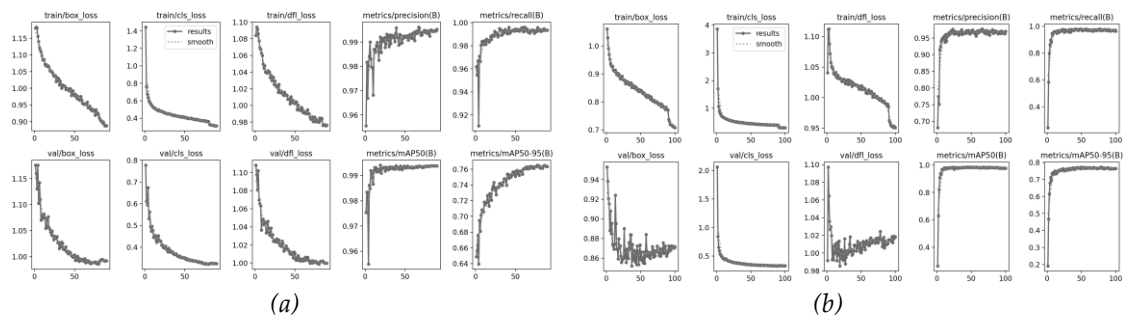
Thực hiện huấn luyện 90 epochs trên bộ dữ liệu phát hiện biển số và 100 epochs trên bộ dữ liệu nhận diện ký tự được trình bày ở trên hình 6, kết quả huấn luyện đạt được các thông số ở bảng 4.

Bảng 4. Kết quả huấn luyện của 02 bộ dữ liệu

Bộ dữ liệu	Precision	Recall	mAP@50	mAP@50-95
Phát hiện biển số	0.994	0.996	0.994	0.765
Nhận diện kí tự	0.97	0.964	0.977	0.764

Khi huấn luyện mô hình chỉ số độ chính xác trung bình mAP@50 luôn cao hơn độ chính xác trung bình mAP@50-95. Vì mAP@50 là trung bình của AP (Average Precision: diện tích dưới đường cong Precision – Recall của một lớp đối tượng) tại ngưỡng IoU=0.5, còn mAP@50-95 là trung bình của AP từ ngưỡng IoU=0.5 đến 0.95, do đó chỉ số mAP@50-95 thường đạt chỉ số thấp hơn mAP@50. Chỉ số mAP@50 dùng để đánh giá, kiểm tra về phát hiện đối tượng của mô hình, còn mAP@50-95 đánh giá toàn diện hơn về phát hiện đối tượng và xác định vị trí. Bảng 4 cho thấy độ chính xác trung bình mAP@50 rất cao, đặc biệt mAP@50 của bộ dữ liệu phát hiện biển số. Điều này khẳng định mô hình hoạt động cực kỳ ổn định, khả năng phát hiện biển số trong điều kiện bình thường là 99,4%. Tuy nhiên nếu trong điều kiện môi trường khó, phức tạp hơn như chồng chéo nhiều lớp, nhiễu nhiễu, che khuất, biển số nhỏ thì khả năng nhận của mô hình sẽ giảm xuống.

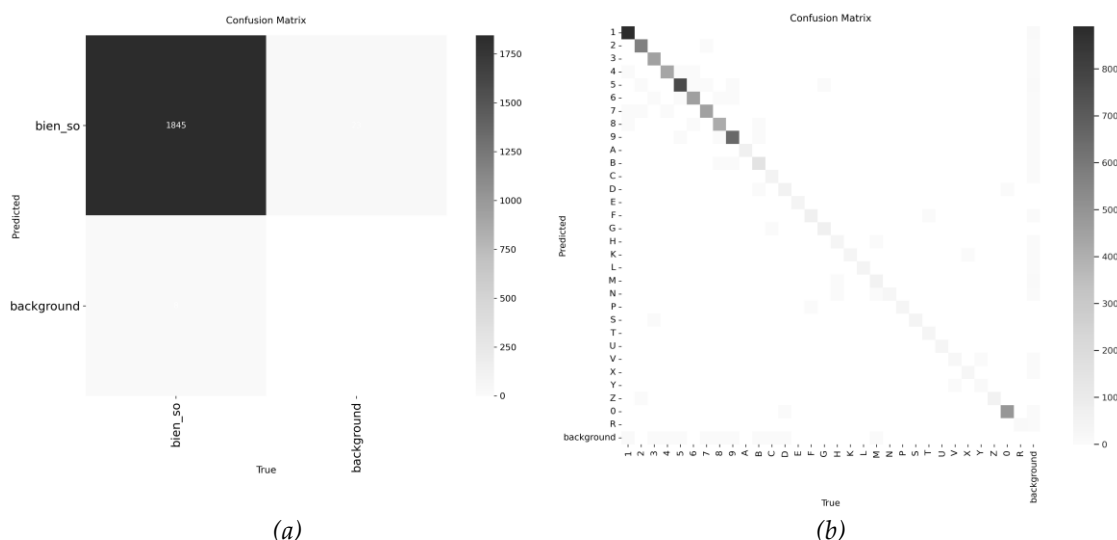
Đường cong hiệu suất huấn luyện mô hình (hình 7) cho thấy ba chỉ số mất mát: box_loss (mất mát hộp bao vật thể), cls_loss (mất mát phân lớp) và dfl_loss (mất mát tiêu điểm phân phối) của hai mô hình giảm đều, cho thấy kết quả huấn luyện mô hình đạt hiệu quả, cụ thể như sau:



Hình 7. Đường cong hiệu suất mô hình huấn luyện; đối với bộ dữ liệu phát hiện biển số (a), đối với bộ dữ liệu nhận diện kí tự (b).

- Đối với mô hình phát hiện biển số: Loss huấn luyện giảm đều, đến khoảng epochs 60 mô hình học đã hiệu quả. Loss kiểm định cũng giảm đều theo loss huấn luyện, không có dấu hiệu overfitting. Vậy mô hình hội tụ ở epochs 60, epochs huấn luyện hiệu quả là khoảng 60-65, do đó có thể đặt giá trị patience, để tự động dừng sớm khi hiệu suất không còn cải thiện.

- Đối với mô hình nhận diện kí tự: Loss huấn luyện giảm đều ổn định, đến khoảng epochs 60-70, sau đó hội tụ. Loss kiểm định cũng giảm nhưng dao động nhẹ sau epochs 70, mô hình có dấu hiệu overfitting nhẹ. Vậy mô hình hội tụ khoảng ở epochs 60-70, overfitting nhẹ sau epochs 70, do đó epochs huấn luyện hiệu quả là khoảng 65-70, nên có thể đặt giá trị patience để tự động dừng sớm khi hiệu suất không còn cải thiện.



Hình 8. Ma trận nhầm lẫn của mô hình huấn luyện; đối với bộ dữ liệu phát hiện biển số (a), đối với bộ dữ liệu nhận diện kí tự (b).

Đường chéo chính của ma trận nhầm lẫn (confusion matrix) hình 8, có màu đậm và rõ nét, chứng tỏ mô hình huấn luyện của hai bộ dữ liệu hoạt động tốt, độ chính xác cao. Tuy nhiên, đối bộ dữ liệu phát hiện biển số mô hình không nhận đúng được một mẫu background nào, do đó mô hình có thể thiên lệch về lớp “bien_so” nên dễ gây nhầm lẫn vùng ảnh là cùng biển số và có thể làm giảm hiệu quả cho mô hình nhận diện kí tự, cần phải tăng số lượng mẫu background. Đối với bộ dữ liệu nhận diện kí tự, ngoài đường chéo chính có màu đậm có một vài ô có màu nhạt, nghĩa là có một vài nhầm lẫn trong dự đoán kí tự, các kí tự dự đoán sai chủ yếu là các kí tự có hình dạng tương tự như: G và C, O và 0, l và 1, B và 8, Y và V,... , cần phải cải thiện tăng cường thêm mẫu học cho các lớp kí tự này.

Từ những kết quả trên, ta thấy mô hình huấn luyện cho kết quả khá tốt, đặc biệt là kết quả dự đoán precision và recall có độ chính xác rất cao.

4.3. Thực nghiệm mô hình

4.3.1. Lưu đồ thuật toán mô hình phát hiện biển số

Quy trình phát hiện biển số của mô hình (hình 9) gồm các bước như sau:

- Đọc và hiển thị ảnh gốc: Đọc ảnh đầu vào từ đường dẫn đã được chỉ định hoặc camera; Hiển thị ảnh gốc lên giao diện chương trình Python

- Dùng mô hình YOLOv11 nhận diện vùng chứa biển số: Sử dụng mô hình YOLOv11n đã được huấn luyện để phát hiện vùng chứa biển số; Vẽ các hộp giới hạn cho vùng chứa biển số trên ảnh gốc.

- Kiểm tra độ tin cậy về nhận diện vùng chứa biển số: Độ tin cậy chính xác phải lớn hơn 50%; Nếu thỏa mãn thì thực hiện bước tiếp theo, nếu không thì kết thúc chu trình.

- Xử lý tách vùng chứa biển số đã nhận diện được: Lặp qua từng vùng chứa biển số; Cắt vùng chứa biển số từ ảnh gốc.

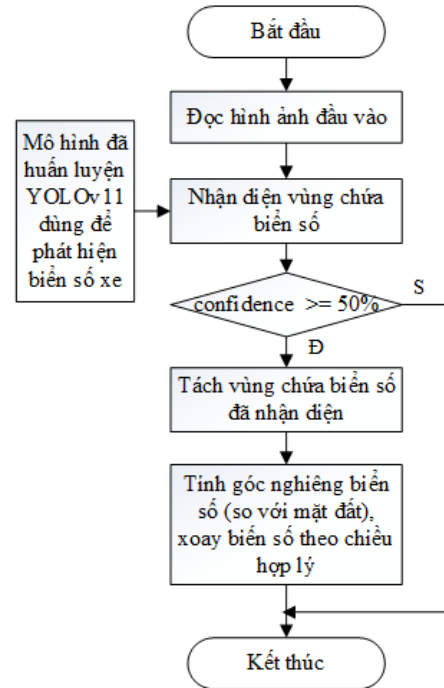
- Tính góc nghiêng và xoay biển số: Tìm tọa độ của 2 ký tự ngoài cùng dòng dưới (hay dòng trên) của biển số; Tìm hệ số góc của đường thẳng đi qua 2 điểm đó; Từ hệ số góc, tính được góc tạo bởi đường thẳng và trục hoành (góc nghiêng); Xoay biển số theo chiều của góc đó.

4.3.2. Lưu đồ thuật toán mô hình nhận diện ký tự

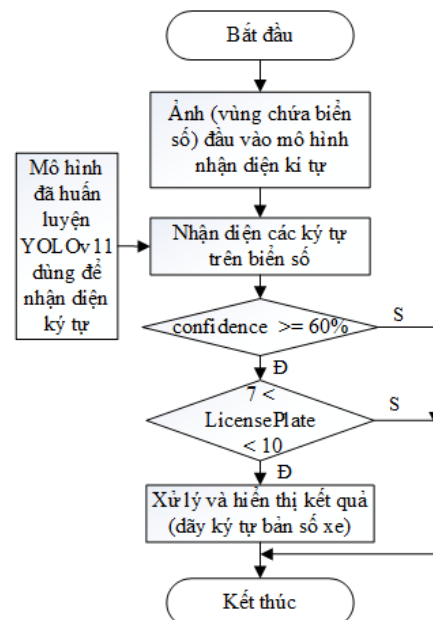
Quy trình nhận diện ký tự biển số của mô hình (hình 10) gồm các bước như sau:

- Sử dụng mô hình YOLOv11n đã huấn luyện để nhận diện các ký tự trong biển số.

- Kiểm tra độ tin cậy về các ký tự đã nhận diện: Độ tin cậy chính xác phải lớn hơn 60%; Nếu thỏa mãn thì thực hiện bước tiếp theo, nếu không



Hình 9. Lưu đồ thuật toán quy trình phát hiện biển số.



Hình 10. Lưu đồ thuật toán quy trình nhận diện ký tự.

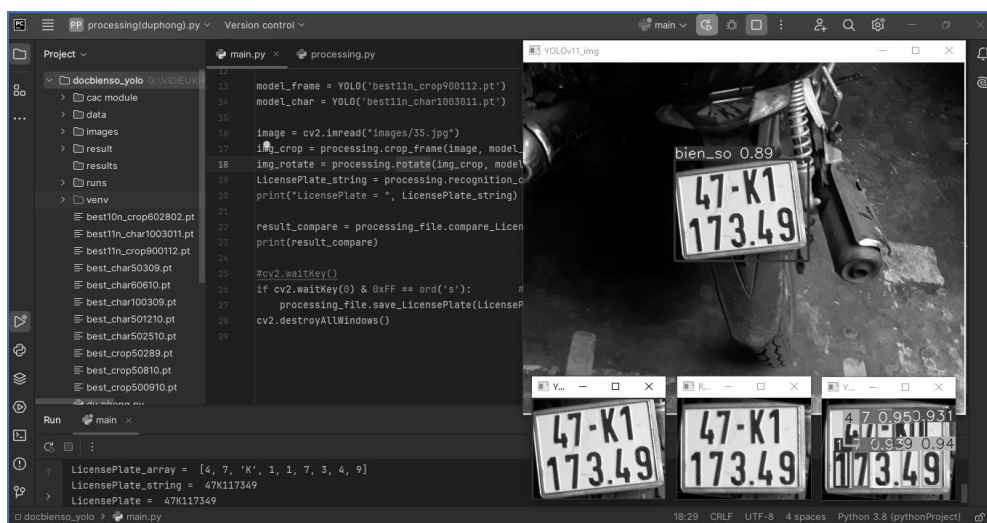
thì kết thúc chu trình.

- Kiểm tra số lượng các kí tự đã nhận diện phù hợp với biển số: Độ dài các kí tự trên biển số phải lớn hơn 7 kí tự và nhỏ hơn 10 kí tự; Nếu thỏa mãn thì thực hiện bước tiếp theo, nếu không thì kết thúc chu trình.

- Hiển thị kí tự biển số: Ghép các kí tự vừa nhận diện theo thứ tự để có kết quả biển số đúng với thực tế; Hiển thị và lưu kí tự.

4.3.3. Kết quả thực nghiệm mô hình

Ngôn ngữ sử dụng trong nghiên cứu này là Python phiên bản 3.8 và trình soạn thảo phát triển tích hợp (IDE) Pycharm. Dữ liệu đầu vào mô hình là hình ảnh phần sau của xe gắn máy có chứa biển số xe, dựa vào thuật toán đã huấn luyện, sau các bước thực hiện, chương trình đã phát hiện, khoanh vùng biển số với tỉ lệ dự đoán là 89%. Tính toán góc nghiêng so với mặt đất là 6.4 độ, xoay biển số vuông góc và nhận dạng từng kí tự với kết quả dự đoán rất cao trên 90% cho các kí tự. Kết quả cuối cùng chương trình nhận dạng, đọc biển số xe khá chính xác như hình 11.



Hình 11. Kết quả nhận diện biển số của mô hình.

Qua thử nghiệm nhiều loại biển số, nhóm tác giả thấy mô hình phát hiện và nhận diện được hầu hết các kí tự của biển số. Tuy nhiên, đối với một số biển số bị mờ nhiều, nhiễu cao hoặc góc chụp bị nghiêng nhiều, thì mô hình vẫn phát hiện được biển số nhưng nhận diện kí bị sai lệch hoặc không nhận diện như hình 12. Trường hợp trong điều kiện bình thường, biển số có độ nghiêng ít và có kí tự bị che khuất (thiếu kí tự biển số), quá trình đọc tọa độ hai kí tự ngoài cùng (trừ kí tự bị che khuất) của dòng biển số, để tính hệ số góc nghiêng vẫn theo quy trình bình thường, nên không ảnh hưởng đến quá trình xoay biển số, kết quả nhận diện kí tự vẫn đảm bảo hình 13.



Hình 12. Một số biển số bị mờ, nhiễu và nghiêng nhiều, mô hình vẫn phát hiện được vị trí biển số nhưng nhận diện ký tự bị sai lệch.



Hình 13. Quá trình đọc tọa độ, tính hệ số góc nghiêng đối với các trường hợp biển số có độ nghiêng ít và thiếu ký tự ngoài cùng.

Trong chương trình hiển thị ký tự biển số, tác giả lập trình chỉ cho hiển thị khi kết quả nhận diện chính xác, đầy đủ ký tự biển số. Nếu kết quả nhận diện ký tự bị thiếu thì chương trình không cho hiển thị và thực hiện thao tác nhận diện lại.

4.4. Đánh giá và bàn luận kết quả

Qua kết quả thực nghiệm mô hình nhận diện biển số xe đối với thuật toán YOLOv11, tác giả nhận thấy mô hình nhận diện khá tốt, khả năng dự đoán chính xác trên 90%. Nhóm tác giả kiểm chứng, sử dụng bộ dữ liệu như trên (hình 6), huấn luyện lại mô hình nhận diện với các phiên bản YOLO trước đó (YOLOv9 và YOLOv10). Tiến hành thực nghiệm với hình ảnh đầu vào là biển số xe (47K117348) hình 10, nhóm tác giả đã nhận thấy các thông số: thời gian nhận diện xử lý, tốc độ tiền xử lý, tốc độ suy

luận, tốc độ hậu xử lý cho mỗi bức ảnh và kết quả dự đoán có sự thay đổi tăng dần hiệu suất ở các phiên bản YOLO sau (bảng 5). Cụ thể thời gian nhận diện xử lý cũng như tốc độ suy luận của phát hiện biển số và nhận diện kí tự ở phiên bản YOLOv11 là nhanh nhất (177.9 ms, 164.7ms), trong khi đó ở phiên bản YOLOv10 (209.6 ms, 191.3ms) và YOLOv9 (357.4 ms, 331.1ms) thì chậm hơn.

Vậy phiên bản YOLOv11 với những kiến trúc thiết kế cải tiến, phương pháp huấn luyện được tối ưu đã mang lại tốc độ xử lý nhanh và độ chính xác cao nhất ở thời điểm hiện tại. Điều này được ứng dụng nhiều trong bài toán phát hiện đối tượng theo thời gian thực. Bên cạnh đó mô hình vẫn còn không nhận diện được hoặc nhận diện bị sai lệch, do kí tự biển số quá mờ, nhiễu nhiễu hoặc bị nghiêng nhiều.

Bảng 5. Thời gian xử lý (bức ảnh hình 9) của các phiên bản YOLO cho việc phát hiện biển số và nhận diện kí tự (ms)

Model	Thời gian xử lý phát hiện biển số			Thời gian xử lý nhận diện kí tự		
	preprocess	inference	postprocess	preprocess	inference	postprocess
YOLOv11	5.9	177.9	1.4	6.5	164.7	1.4
YOLOv10	6.0	209.6	0.5	3.3	191.3	0.4
YOLOv9	6.6	357.4	1.4	3.9	331.1	1.5

5. KẾT LUẬN

Trong bài báo này nhóm tác giả đã nghiên cứu, phân tích thuật toán YOLOv11, từ đó xây dựng mô hình bài toán nhận diện biển số xe. Cơ sở dữ liệu là 8.855 bức ảnh chứa biển số xe và 4.025 bức ảnh kí tự biển số. Được huấn luyện lại và chạy thực nghiệm với đầu vào là ảnh của biển số xe bất kỳ, cho kết quả đầu ra là dữ liệu kí tự của biển số. Kết quả thử nghiệm của mô hình tương đối chính xác, có thể chấp nhận được. Mô hình có thể nhận diện được hầu hết các biển số xe máy trong điều kiện thường. Tuy nhiên mô hình của chúng tôi vẫn còn một số sai sót nhỏ, cần phải cải thiện thêm để phát triển trong tương lai:

- Bổ sung thêm dữ liệu huấn luyện để kết quả dự đoán tăng thêm độ chính xác và huấn luyện lại mô hình với số epochs phù hợp, tối ưu hơn. Ứng dụng thêm các thuật toán khác trong xử lý ảnh, xoay biển số nghiêng, để tăng độ chính xác của kết quả thu được.
- Đề tài có thể áp dụng ở các bãi gửi xe, các trạm thu phí, các trạm gác cổng, ... hoặc đề tài có thể kết hợp với cảm biến vân tay, tạo thành cổng tự động nhà xe học sinh sinh viên, nhà trọ.

TÀI LIỆU THAM KHẢO

- [1]. Nguyễn Thái Cường và nhóm tác giả (2024). Nghiên cứu và ứng dụng thuật toán YOLOv7 để phân loại cà chua, *Tạp chí Khoa học Tài nguyên và Môi trường*.
- [2]. Đặng Thị Dung và nhóm tác giả (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*.
- [3]. Nguyễn Thành Lợi và nhóm tác giả (2023). Đề xuất Mô hình YOLOv5 ứng dụng trong nhận diện biển số xe, *Tạp chí Khoa học trường Đại học Mở Hà Nội*, số 104
- [4]. Shenghu Pan, Jian Liu and Dekun Chen (2022). Research on License Plate Detection and Recognition System based on YOLOv7 and LPRNet, *Academic Journal of Science and Technology*.
- [5]. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, RealTime Object Detection, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788.
- [6]. Ali Tourani, Asadollah Shahbahrami, Ching Yee Suen, Saeed Khazaee, Sajjad Soroori (2020). A Robust Deep Learning Approach for Automatic Iranian Vehicle License Plate Detection and Recognition for Surveillance Systems, *IEEE Access*.
- [7]. Yongjie Zou, Yongjun Zhang, Junyan, Xiaoxu Jiang, Tengjie Huang, Haisheng Fan, Zhongwei Cui (2020). A Robust License Plate Recognition Model Based on Bi-LSTM, *IEEE Access*.
- [8]. Trần Thị Hương, Ngô Thị Kiều Hằng (2020). Kỹ thuật nhận dạng biển số xe và ứng dụng vào bài toán quản lý bãi giữ xe tại Trường Đại học Hà Tĩnh, *Tạp chí Khoa học Đại học Đồng Tháp*.
- [9]. Hengliang Shi, Dongnan Zhao (2023). License Plate Recognition System Based on Improved YOLOv5 and GRU, *IEEE Access*.
- [10]. Mejdil Safran, Abdulmalik Alajmi, Sultan Alfarhood (2024). Efficient Multistage License Plate Detection and Recognition Using YOLOv8 and CNN for Smart Parking Systems, *Hindawi Journal of Sensors*.
- [11]. Gudala Lavanya, Sagar Dhanraj Pande (2023). Enhancing Real-time Object Detection with YOLO Algorithm, *EAI Endorsed Transactions on Internet of Things*.
- [12]. Trương Quốc Bảo và nhóm tác giả (2022). Nghiên cứu thuật toán thị giác máy tính để theo dõi việc thực hiện các quy định về phòng ngừa Covid19 sử dụng kỹ thuật học sâu, *Tạp chí Khoa học trường Đại học Cần Thơ*.
- [13]. <https://docs.ultralytics.com/vi/models/yolo11>.
- [14]. Rahima Khanam, Muhammad Hussain (2024). YOLOV11: An overview of the key Architectural Enhancements, *arXiv:2410.17725v1 [cs.CV] 23 Oct 2024*.
- [15]. Mujadded Al Rabbani Alif (2024). YOLOV11 For Vehicle Detection: Advancements, Performance and Applications in Intelligent Transportation Systems, *arXiv:2410.22898v1 [cs.CV] 30 Oct 2024*.

RESEARCH AND APPLICATION YOLOv11 ALGORITHM TO DETECTING LICENSE PLATE

Pham Nhu Pham

Gia Lai College

Email: phamnhupham@gmail.com

ABSTRACT

Artificial intelligence, deep learning, takes inspiration from human neural networks to build algorithms that can think. Artificial intelligence's deep learning models can learn from data, process it, and make decisions automatically and quickly. This paper presents the research and application of the YOLOv11 model for license plate recognition. The YOLOv11 algorithm was retrained using a customized dataset containing license plates and character images to detect and analyze objects accurately. The results demonstrate high accuracy (mAP@50 over 99%) and fast processing speed, confirming its feasibility for deployment in real-world systems such as intelligent parking facilities.

Keywords: deep learning, model training, license plate recognition, artificial intelligence, YOLOv11.



Phạm Như Phẩm sinh 02/6/1983 tại Bình Định. Ông tốt nghiệp cử nhân ngành vật lý tại trường Đại học Đà Lạt năm 2005, tốt nghiệp thạc sĩ chuyên ngành vật lý kỹ thuật tại trường Đại học Đà Lạt năm 2018. Ông công tác tại trường Cao đẳng Gia Lai từ năm 2008.

Lĩnh vực nghiên cứu: Vật lý kỹ thuật, Điện tử tự động, Ứng dụng công nghệ thông tin.