

PHƯƠNG PHÁP CẢI TIẾN HỆ THỐNG NHẬN DẠNG BIÊN GIAO THÔNG BẰNG ỨNG DỤNG YOLO

TRẦN THỊNH MẠNH ĐỨC, ĐỖ TRÍ NHỰT*

¹ Bộ môn Hệ thống Nhúng và Robotics, Khoa Kỹ thuật Máy tính, Trường Đại học Công nghệ Thông tin

*Tác giả liên hệ: trinhutdo@uit.edu.vn

DOIs: <https://doi.org/10.46242/jstiuh.v70i4.4912>

Tóm tắt. Hệ thống biển báo giao thông được nhận dạng chính xác và nhanh chóng đã góp phần cải thiện việc quản lý giao thông trong thời đại chuyển đổi số theo chủ trương của chính phủ hiện nay. Hệ thống này đã được nghiên cứu và phát triển mạnh trong những năm gần đây với các giải thuật kinh điển về xử lý ảnh đến các giải thuật với ứng dụng trí tuệ nhân tạo mà thuật toán You Only Look Once (YOLO) dựa trên kỹ thuật thị giác máy tính là một đại diện đáng để quan tâm. Trong bài báo này, một phương pháp nhằm cải tiến hệ thống nhận dạng biển báo giao thông bằng ứng dụng YOLO như vậy được nghiên cứu về phương diện số lượng biển báo nhận dạng được, tốc độ nhận dạng và độ chính xác. Hệ thống được cải tiến để nhận dạng 9 loại biển báo giao thông khác nhau bao gồm: biển báo cấm rẽ trái, biển báo cấm rẽ phải, biển báo cấm rẽ trái-phải, biển báo cấm dừng-cấm đỗ, biển báo cấm đỗ, biển báo cấm ô tô rẽ phải, biển báo cấm ô tô rẽ trái, biển báo cấm quay đầu và biển báo cấm đi thẳng. Hệ thống được cải tiến và huấn luyện lại với các hình ảnh được thu thập ở các con đường trên thành phố Hồ Chí Minh bao gồm 735 ảnh biển báo cấm rẽ trái, 713 ảnh biển báo cấm rẽ phải, 177 ảnh cấm rẽ trái-phải, 752 ảnh biển báo cấm dừng-cấm đỗ, 629 ảnh biển báo cấm đỗ, 191 ảnh cấm ô tô rẽ phải, 143 ảnh cấm ô tô rẽ trái, 171 cấm quay đầu và 109 cấm đi thẳng. Hệ thống sau đó được kiểm thử thực nghiệm trên thực địa cho độ chính xác nhận dạng theo độ đo mAP@.5 lần lượt như sau: 99.4% nhận dạng đúng biển báo cấm rẽ trái, 99.3% nhận dạng đúng biển báo cấm rẽ phải, 95.6% nhận dạng đúng biển báo cấm dừng-cấm đỗ, 95.3% nhận dạng đúng biển cấm đỗ, 98.6% nhận dạng đúng biển cấm quay đầu, 93.7% nhận dạng đúng biển cấm ô tô rẽ phải, 94.5% nhận dạng đúng biển cấm ô tô rẽ trái, 99.5% nhận dạng đúng biển cấm rẽ trái-phải và cuối cùng là 93.4% nhận dạng đúng biển cấm đi thẳng.

Từ khóa. phát hiện biển báo giao thông, nhận dạng biển báo giao thông, thuật toán YOLO, thuật toán YOLT.

1 GIỚI THIỆU

Nhận dạng biển báo giao thông là một lĩnh vực nghiên cứu quan trọng về thị giác máy tính và có nhiều ứng dụng thực tế trong hệ thống giao thông thông minh (Intelligent Transportation Systems - ITS) [1]. Nhìn chung, hệ thống nhận dạng biển báo giao thông có thể được chia thành hai loại: dựa trên thị giác máy tính và không dựa trên thị giác máy tính. Hệ thống dựa trên thị giác máy tính sử dụng camera để chụp ảnh đường rồi xử lý những hình ảnh này để phát hiện và nhận biết biển báo giao thông. Mặt khác, các hệ thống không dựa trên thị giác máy tính thì sử dụng các cảm biến như radar hoặc LIDAR để phát hiện biển báo giao thông. Bài báo [1] tổng hợp và đánh giá về đề tài này được xuất bản năm 2014 đã cung cấp đánh giá quan trọng về ba bước chính trong hệ thống Nhận dạng và Phát hiện Biển báo Giao thông Tự động (Automatic Traffic Sign Detection and Recognition - ATSDR), tức là phân đoạn, phát hiện và nhận dạng trong bối cảnh hệ thống hỗ trợ người lái xe dựa trên kỹ thuật thị giác máy tính. Bài đăng [1] cũng tập trung vào các thiết lập thử nghiệm khác nhau của hệ thống thu nhận hình ảnh và thảo luận về những thách thức nghiên cứu có thể có trong tương lai để làm cho ATSDR hiệu quả hơn.

Một bài đăng [2] đánh giá khác được xuất bản vào năm 2017 trình bày đánh giá về các phương pháp phát hiện hiện sự tồn tại như phương pháp phát hiện tồn tại dựa trên màu sắc, dựa trên hình dạng và dựa trên học tập (color-based, shape-based, and learning-based existing detection methods). Bài đăng [2] cũng thảo luận về các thuật toán so khớp tính năng và học máy được sử dụng trong giai đoạn nhận dạng biển báo giao thông. Một cuộc khảo sát được công bố vào năm 2020 cung cấp cái nhìn tổng quan về các công trình nghiên cứu về phát hiện và nhận dạng biển báo giao thông, bao gồm các phương pháp tiếp cận mới, mang

tính đột phá. Bài đăng [3] cũng thảo luận về cơ sở dữ liệu biển báo giao thông và các bước vốn có của nó: Tiền xử lý, Trích xuất và phát hiện tính năng, Xử lý hậu kỳ. Một bài báo [4] xuất bản năm 2019 trình bày đánh giá về các phương pháp phát hiện biển báo giao thông dựa trên thị giác máy tính; đồng thời nêu rõ các thảo luận về những thách thức liên quan đến việc Phát hiện biển báo giao thông (Traffic Sign Detection – TSD) như các loại khác nhau, kích thước nhỏ, bối cảnh lái xe phức tạp và tắc nghẽn. Cuối cùng, một bài báo [5] xuất bản vào năm 2022 có đánh giá hiệu suất của YOLOv5 dựa trên tập dữ liệu về Nhận dạng biển báo giao thông (Traffic Sign Recognition - TSR) thông qua so sánh toàn diện với SSD (thí dụ như là trình phát hiện nhiều hộp trên ảnh chụp một lần - single shot multibox detector).

YOLO được viết tắt từ cụm từ You Only Look Once, là thuật toán phát hiện đối tượng theo thời gian thực tiên tiến được giới thiệu vào năm 2015 bởi Joseph Redmon, Santosh Divvala, Ross Girshick và Ali Farhadi [6]. Nó được sử dụng trong thị giác máy tính để nhận dạng và bản địa hóa các đối tượng trong một hình ảnh hoặc video. YOLO nhanh hơn các mô hình phát hiện đối tượng khác và có thể xử lý hình ảnh ở tốc độ 45 khung hình mỗi giây (FPS). Nó có độ chính xác phát hiện cao với rất ít lỗi nền. YOLO cũng là tài nguyên mã nguồn mở và có tính khái quát tốt. Chính vì vậy, từ khi được giới thiệu, các phiên bản YOLO đã liên tục phát triển theo thời gian, với 15 mô hình từ YOLOv1 ban đầu đến YOLOv8 mới nhất [7]. Dưới đây là tổng quan ngắn gọn về các phiên bản khác nhau của YOLO:

- YOLOv1: Phiên bản đầu tiên của YOLO được giới thiệu vào năm 2015. Nó nhanh hơn các mô hình phát hiện đối tượng khác và có thể xử lý hình ảnh ở tốc độ 45 khung hình mỗi giây (FPS).
- YOLOv2: Được giới thiệu vào năm 2016, YOLOv2 đã cải thiện độ chính xác phát hiện của YOLOv1 bằng cách sử dụng kiến trúc phức tạp hơn và thêm hàng loạt chuẩn hóa.
- YOLOv3: Được phát hành vào năm 2018, YOLOv3 đã cải thiện hơn nữa độ chính xác của việc phát hiện đối tượng bằng cách giới thiệu một tính năng mới có tên là “Mạng kim tự tháp đặc trưng” (Feature Pyramid Networks - FPN).
- YOLOv4: Được phát hành vào năm 2020, YOLOv4 đã giới thiệu một số tính năng mới như “CSPDarknet53” và “SPP” để cải thiện độ chính xác và tốc độ phát hiện của mô hình.
- YOLOv5: Được phát hành vào năm 2020, YOLOv5 là một kiến trúc hoàn toàn mới dựa trên máy dò một giai đoạn. Nó nhanh hơn và chính xác hơn các phiên bản trước của YOLO.
- YOLOv6: Được phát hành vào năm 2021, YOLOv6 là phiên bản cải tiến của YOLOv5 sử dụng kiến trúc hiệu quả hơn để đạt được tốc độ nhanh hơn nữa trong khi vẫn duy trì độ chính xác cao.
- YOLOv7: Được phát hành vào năm 2022, YOLOv7 là phiên bản thậm chí còn nhanh hơn của YOLO, sử dụng kiến trúc mới có tên “YOLT” ((You Only Look Twice - Bạn chỉ nhìn hai lần) để đạt được hiệu suất tiên tiến trong các nhiệm vụ phát hiện đối tượng.
- YOLOv8: Được phát hành vào năm 2023, YOLOv8 là phiên bản mới nhất của gia đình YOLO. Nó giới thiệu một số tính năng mới như “Chuyển đổi động (Dynamic Convolution)” và “Chú ý không gian (Spatial Attention)” để cải thiện độ chính xác và tốc độ phát hiện của mô hình.

Trong bài báo này, các tác giả sau khi đã tìm hiểu các cơ sở dữ liệu về các loại biển báo giao thông đã có trên mạng và thấy không có cơ sở nào về biển báo giao thông phù hợp với thiết kế của biển báo ở Việt Nam nên đã tự thu thập dữ liệu hình ảnh biển báo giao thông của Việt Nam hơn ba ngàn tấm hình về chín loại biển báo đã được đề cập trong bài báo. Đối với mô hình Yolo, tác giả đã thay đổi chỉnh sửa các tham số trong mô hình để có được kết quả tốt nhất trong dự đoán biển báo giao thông. Ngoài ra cài đặt lên kit kv260 và camera để ứng dụng vào thực tế. Các tác giả đề xuất thiết kế Hệ thống nhận dạng biển báo giao thông dựa trên kỹ thuật thị giác máy tính và thuật toán YOLO, phiên bản thứ 7. Hệ thống đề xuất được thiết kế để nhận dạng 9 loại biển báo giao thông ở Việt nam bao gồm: cấm rẽ trái, cấm rẽ phải, cấm rẽ trái-phải, cấm đỗ, cấm dừng-đỗ, cấm ô tô rẽ phải, cấm ô tô rẽ trái, cấm quay đầu và cấm đi thẳng. Hình ảnh sẽ được chụp ở các góc độ có thể nhận dạng được từ camera khi đang chạy trên đường, hình ảnh cho huấn luyện sẽ không được quá mờ hoặc quá xa ngoài tầm nhìn của mắt người. Thực tế chứng minh qua các công bố khoa học mới nhất cho thấy các mô hình nhận dạng biển báo giao thông dựa trên YOLO cung cấp khả năng xử lý, hiệu quả và mạnh mẽ theo thời gian thực, khiến chúng trở thành lựa chọn hấp dẫn cho các ứng dụng trong xe tự hành và an toàn đường bộ [8, 9]. Các phân tích so sánh chi tiết về những ưu điểm của mô hình nhận dạng biển báo giao thông dựa trên YOLO so với các phương pháp khác được nhóm tác giả tổng hợp như sau:

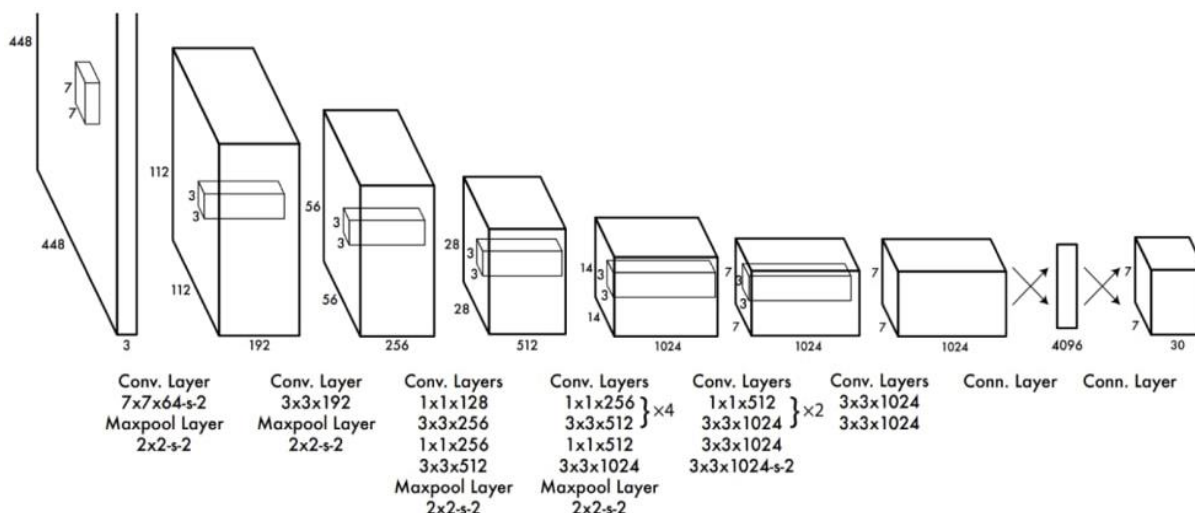
- **Xử lý thời gian thực:** YOLO xử lý hình ảnh theo thời gian thực, khiến nó phù hợp với các ứng dụng yêu cầu tốc độ cao, chẳng hạn như xe tự hành. Không giống như một số mô hình khác yêu cầu nhiều lần chuyển qua một hình ảnh, YOLO dự đoán các hộp giới hạn và xác suất của lớp trong một lần chuyển tiếp.
- **Độ chính xác và hiệu quả:** YOLO tạo ra sự cân bằng giữa độ chính xác và hiệu quả. Nó đạt được hiệu suất cạnh tranh trong khi vẫn hiệu quả về mặt tính toán. Các mô hình khác có thể chính xác nhưng tốn kém về mặt tính toán, khiến chúng ít thực tế hơn khi triển khai trong thế giới thực.
- **Phát hiện thống nhất:** YOLO phát hiện nhiều đối tượng trong một lần quét, bao gồm cả biển báo giao thông. Nó không dựa vào đề xuất khu vực hoặc cửa sổ trượt. Các mô hình khác có thể sử dụng các giai đoạn riêng biệt để định vị và phân loại đối tượng, việc này có thể kém hiệu quả hơn.
- **Sự khái quát:** Kiến trúc của YOLO cho phép nó khái quát hóa tốt các loại đối tượng khác nhau, bao gồm cả biển báo giao thông. Một số mô hình khác có thể gặp khó khăn với việc khái quát hóa do kiến trúc phức tạp hoặc trang bị quá mức.
- **Mạnh mẽ về quy mô và tỷ lệ khung hình:** YOLO xử lý các đối tượng có kích thước và tỷ lệ khung hình khác nhau một cách hiệu quả. Các mô hình khác có thể yêu cầu tiền xử lý bổ sung hoặc các kỹ thuật chuyên biệt để xử lý các biến thể tỷ lệ.
- **Kiến trúc mạng đơn:** YOLO sử dụng một mạng lưới thần kinh duy nhất cho cả việc định vị và phân loại đối tượng. Các mô hình khác có thể liên quan đến các mạng hoặc đường ống riêng biệt, dẫn đến độ phức tạp tăng lên.
- **Chuyển tiếp học tập:** YOLO có thể hưởng lợi từ việc học chuyên giao bằng cách đào tạo trước trên các tập dữ liệu lớn và tinh chỉnh cho các tác vụ cụ thể như nhận dạng biển báo giao thông. Các mô hình khác có thể không phù hợp với việc chuyển giao việc học tập.

Bài viết này được cấu trúc như sau. Phần 2 mô tả kiến thức cơ bản về nội dung đề tài nghiên cứu trong khi phần 3 là thiết kế hệ thống được đề xuất để đánh giá, bao gồm các thiết kế cho các bộ phận phần cứng và phần mềm. Phần 4 trình bày một số kết quả thực nghiệm chứng minh tính chính xác và ổn định của thuật toán sử dụng cho hệ thống đề xuất. Cuối cùng, phần 5 kết luận nội dung đã giới thiệu trong bài báo này và đề xuất một số hướng cải tiến trong tương lai.

2 KIẾN THỨC NỀN VỀ THUẬT TOÁN NHẬN DẠNG

Phát hiện đối tượng/vật thể là một nhiệm vụ phổ biến trong kỹ thuật thị giác máy tính. Nhiệm vụ đề cập đến việc khoanh vùng quan tâm trong một hình ảnh và phân loại vùng này giống như một bộ phân loại hình ảnh điển hình. Một hình ảnh có thể bao gồm một số vùng quan tâm trở đến các đối tượng khác nhau. YOLO (You Only Look Once) là một mô hình phát hiện đối tượng phổ biến được biết đến với tốc độ và độ chính xác. Trong bài viết này, chúng ta sẽ thảo luận về điều gì khiến YOLO v7 nổi bật và so sánh nó với các thuật toán phát hiện đối tượng khác như thế nào.

Thuật toán YOLO lấy hình ảnh làm đầu vào và sau đó sử dụng mạng nơ ron tích chập sâu đơn giản để phát hiện các đối tượng trong ảnh. Kiến trúc của mô hình CNN tạo thành xương sống của YOLO được trình bày như Hình 1 bên dưới.



Hình 1. Cấu trúc mô tả thuật toán YOLO. [10]

Hai mươi lớp chập đầu tiên của mô hình được huấn luyện trước bằng ImageNet bằng cách cắm vào lớp tổng hợp trung bình tạm thời và lớp được kết nối đầy đủ. Sau đó, mô hình được huấn luyện/đào tạo trước này được chuyển đổi để thực hiện phát hiện đối tượng/vật thể. Nhiều nghiên cứu trước đó đã chỉ ra rằng việc thêm các lớp tích chập và kết nối vào mạng được huấn luyện/đào tạo trước sẽ cải thiện hiệu suất. Lớp được kết nối đầy đủ cuối cùng của YOLO dự đoán cả xác suất của lớp và tọa độ hộp giới hạn.

YOLO chia hình ảnh đầu vào thành lưới $S \times S$. Nếu tâm của một đối tượng rơi vào một ô lưới thì ô lưới đó có nhiệm vụ phát hiện đối tượng đó. Mỗi ô lưới dự đoán B hộp biên và điểm tin cậy cho các hộp đó. Những điểm tin cậy này phản ánh mức độ tin cậy của mô hình và mức độ chính xác được dự đoán.

YOLO dự đoán nhiều hộp biên trên mỗi ô lưới. Tại thời điểm huấn luyện/đào tạo, ta chỉ muốn một bộ dự đoán hộp biên chịu trách nhiệm cho từng đối tượng. Điều này dẫn đến sự chuyên môn hóa giữa các yếu tố dự đoán hộp biên. Mỗi công cụ dự đoán sẽ hoạt động tốt hơn trong việc dự đoán các kích thước, tỷ lệ khung hình hoặc loại đối tượng nhất định, cải thiện điểm thu hồi tổng thể.

Một kỹ thuật quan trọng được sử dụng trong các mô hình YOLO là triết tiêu không tối đa (non-Maximum Suppression - NMS). NMS là bước xử lý hậu kỳ được sử dụng để cải thiện độ chính xác và hiệu quả của việc phát hiện đối tượng/vật thể. Trong phát hiện đối tượng, thông thường sẽ có nhiều khung giới hạn được tạo cho một đối tượng trong ảnh. Các khung giới hạn này có thể chồng lên nhau hoặc đặt ở các vị trí khác nhau nhưng chúng đều đại diện cho cùng một đối tượng. NMS được sử dụng để xác định và loại bỏ các hộp biên dư thừa hoặc không chính xác và xuất ra một hộp biên duy nhất cho từng đối tượng trong ảnh.

Như đã đề cập trong Phần 1, giới thiệu đề tài nghiên cứu, YOLO có nhiều phiên bản phát triển theo thời gian từ năm 2015. Trong phần này chúng ta sử dụng phiên bản thứ 7 (YOLOv7). Đây là phiên bản mới của YOLO, có một số cải tiến so với các phiên bản trước.

Một trong những cải tiến chính là việc sử dụng các hộp neo (anchor boxes). Hộp neo là một tập hợp các hộp được xác định trước với các tỷ lệ khung hình khác nhau được sử dụng để phát hiện các đối tượng có hình dạng khác nhau. YOLOv7 sử dụng chín hộp neo, cho phép phát hiện phạm vi hình dạng và kích thước đối tượng/vật thể rộng hơn so với các phiên bản trước, do đó giúp giảm số lượng kết quả dương tính giả.

THIẾT KẾ HỆ THỐNG NHẬN DẠNG BIÊN BẢO GIAO THÔNG VỚI ỨNG DỤNG YOLO

Model	#Param.	FLOPs	Size	AP ^{val}	AP ^{val} ₅₀	AP ^{val} ₇₅	AP ^{val} _S	AP ^{val} _M	AP ^{val} _L
YOLOv4 [3]	64.4M	142.8G	640	49.7%	68.2%	54.3%	32.9%	54.8%	63.7%
YOLOv4-u5 (r6.1) [81]	46.5M	109.1G	640	50.2%	68.7%	54.6%	33.2%	55.5%	63.7%
YOLOv4-CSP [79]	52.9M	120.4G	640	50.3%	68.6%	54.9%	34.2%	55.6%	65.1%
YOLOv4-CSP [81]	52.9M	120.4G	640	50.8%	69.5%	55.3%	33.7%	56.0%	65.4%
YOLOv7	36.9M	104.7G	640	51.2%	69.7%	55.5%	35.2%	56.0%	66.7%
improvement	-43%	-15%	-	+0.4	+0.2	+0.2	+1.5	=	+1.3
YOLOv7-X [81]	96.9M	226.8G	640	52.7%	71.3%	57.4%	36.3%	57.5%	68.3%
YOLOv7-X	71.3M	189.9G	640	52.9%	71.1%	57.5%	36.9%	57.7%	68.6%
improvement	-36%	-19%	-	+0.2	-0.2	+0.1	+0.6	+0.2	+0.3
YOLOv4-tiny [79]	6.1	6.9	416	24.9%	42.1%	25.7%	8.7%	28.4%	39.2%
YOLOv7-tiny	6.2	5.8	416	35.2%	52.8%	37.3%	15.7%	38.0%	53.4%
improvement	+2%	-19%	-	+10.3	+10.7	+11.6	+7.0	+9.6	+14.2
YOLOv4-tiny-3l [79]	8.7	5.2	320	30.8%	47.3%	32.2%	10.9%	31.9%	51.5%
YOLOv7-tiny	6.2	3.5	320	30.8%	47.3%	32.2%	10.0%	31.9%	52.2%
improvement	-39%	-49%	-	=	=	=	-0.9	=	+0.7
YOLOv7-E6 [81]	115.8M	683.2G	1280	55.7%	73.2%	60.7%	40.1%	60.4%	69.2%
YOLOv7-E6	97.2M	515.2G	1280	55.9%	73.5%	61.1%	40.6%	60.3%	70.0%
improvement	-19%	-33%	-	+0.2	+0.3	+0.4	+0.5	-0.1	+0.8
YOLOv7-D6 [81]	151.7M	935.6G	1280	56.1%	73.9%	61.2%	42.4%	60.5%	69.9%
YOLOv7-D6	154.7M	806.8G	1280	56.3%	73.8%	61.4%	41.3%	60.6%	70.1%
YOLOv7-E6E	151.7M	843.2G	1280	56.8%	74.4%	62.1%	40.8%	62.1%	70.6%
improvement	=	-11%	-	+0.7	+0.5	+0.9	-1.6	+1.6	+0.7

Hình 2. So sánh thuật toán YOLOv7 và các thuật toán khác trên cùng tập dữ liệu COCO. [10]

Một cải tiến quan trọng khác trong YOLOv7 là việc sử dụng hàm mất mát mới được gọi là “mất tiêu điểm” (focal loss). Các phiên bản trước của YOLO đã sử dụng hàm mất entropy chéo tiêu chuẩn, được biết là kém hiệu quả hơn trong việc phát hiện các đối tượng/vật thể nhỏ. Hàm mất tiêu điểm giải quyết vấn đề này bằng cách giảm trọng số mất mát đối với các mẫu được phân loại tốt và tập trung vào các mẫu khó (thí dụ như các đối tượng khó phát hiện).

YOLO v7 cũng có độ phân giải cao hơn các phiên bản trước. Nó xử lý hình ảnh ở độ phân giải 608 x 608 pixel, cao hơn độ phân giải 416 x 416 được sử dụng trong YOLOv3. Độ phân giải cao hơn này cho phép YOLOv7 phát hiện các vật thể nhỏ hơn và có độ chính xác tổng thể cao hơn.

Một trong những ưu điểm chính của YOLOv7 là tốc độ của nó. Nó có thể xử lý hình ảnh với tốc độ 155 khung hình/giây, nhanh hơn nhiều so với các thuật toán phát hiện đối tượng tiên tiến khác. Ngay cả mô hình YOLO cơ bản ban đầu cũng chỉ có khả năng xử lý ở tốc độ tối đa 45 khung hình mỗi giây. Điều này làm cho nó phù hợp với các ứng dụng thời gian thực nhạy cảm như giám sát và vận hành hệ thống lái cho xe tự lái, trong đó tốc độ xử lý cao hơn là rất quan trọng.

Về độ chính xác, YOLOv7 hoạt động tốt so với các thuật toán phát hiện đối tượng/vật thể khác. Nó đạt được độ chính xác trung bình là 37,2% ở ngưỡng IoU (intersection over union - giao điểm trên liên kết) là 0,5 trên bộ dữ liệu COCO phổ biến, có thể so sánh với các thuật toán phát hiện đối tượng tiên tiến khác. Sự so sánh định lượng của hiệu suất được hiển thị trong Hình 2 phía trên.

Cuối cùng, ta nên hiểu YOLO (You Only Look Once) là một thuật toán phát hiện đối tượng/vật thể phổ biến và đã cách mạng hóa lĩnh vực kỹ thuật thị giác máy tính. YOLO nhanh chóng và hiệu quả, khiến nó trở thành sự lựa chọn tuyệt vời cho các tác vụ phát hiện đối tượng/vật thể theo thời gian thực. Nó đã đạt được hiệu suất tiên tiến trên nhiều tiêu chuẩn khác nhau và đã được áp dụng rộng rãi trong nhiều ứng dụng thực tế khác nhau. Một trong những ưu điểm chính của YOLO là tốc độ suy luận nhanh, cho phép nó xử lý hình ảnh theo thời gian thực. YOLO rất phù hợp cho các ứng dụng như giám sát video, xe tự lái. Ngoài ra, YOLO có kiến trúc đơn giản và yêu cầu dữ liệu đào tạo tối thiểu, giúp dễ dàng triển khai và thích ứng với các nhiệm vụ mới. Bất chấp những hạn chế như khó nhận biết các vật thể nhỏ và không có khả năng thực hiện phân loại đối tượng chi tiết, YOLO đã chứng tỏ là một công cụ có giá trị để phát hiện đối tượng/vật thể và mở ra nhiều khả năng mới cho các nhà nghiên cứu và thực hành. Khi lĩnh vực Kỹ thuật Thị giác Máy

tính tiếp tục phát triển, sẽ rất thú vị khi xem YOLO và các thuật toán phát hiện đối tượng khác phát triển và cải tiến như thế nào.

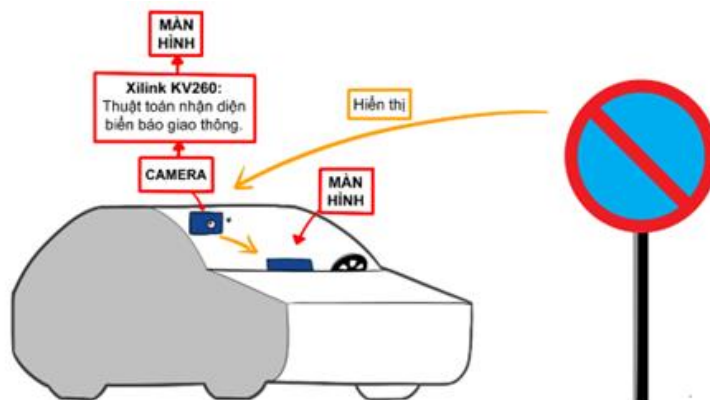
3 PHƯƠNG PHÁP THIẾT KẾ HỆ THỐNG NHẬN DẠNG

Nhìn chung tổng quan, chúng ta dễ dàng nhận ra các yếu tố và thách thức của nghiên cứu nhận dạng biển báo giao thông đã tồn tại cho đến nay, trong đó cần nêu bật yếu tố quan trọng nhất là yếu tố đầu vào bao gồm chất lượng ảnh huấn luyện, yếu tố môi trường, thời tiết, độ phân giải camera và độ rung lắc của thiết bị khi quan sát. Nhóm tác giả sau đây sẽ phân tích rõ ảnh hưởng của yếu tố đầu vào để khoanh vùng phạm vi của nghiên cứu liên quan đến những yếu tố trên.

Khi nghiên cứu về việc nhận dạng biển báo giao thông hoặc các ứng dụng khác, yếu tố đầu vào quan trọng bao gồm:

1. **Dữ liệu ảnh:** Dữ liệu ảnh là yếu tố quan trọng nhất trong quá trình nhận dạng biển báo giao thông. Dữ liệu cần phải đủ đa dạng với các loại biển báo khác nhau, trong điều kiện ánh sáng và thời tiết khác nhau.
2. **Định dạng biển báo:** Cần 9 loại biển báo giao thông cần nhận dạng: Biển báo cấm rẽ trái, biển báo cấm rẽ phải, biển báo cấm rẽ trái-phải, biển báo cấm dừng-cấm đỗ, biển báo cấm đỗ, biển báo cấm ô tô rẽ phải, biển báo cấm ô tô rẽ trái, biển báo cấm quay đầu và biển báo cấm đi thẳng.
3. **Kích thước và tỷ lệ biển báo:** Biển báo giao thông có thể xuất hiện ở nhiều kích thước và tỷ lệ khác nhau.
4. **Độ phân giải của camera:** Camera để lấy dữ liệu hình ảnh có ống quay chính 64 MP & Phụ 8 MP, 5 MP. Camera chính có khẩu độ f/1.79 giúp thu được nhiều ánh sáng hơn, giảm nhiễu mang lại chất lượng ảnh khi chụp đêm vô cùng ổn.
5. **Ngữ cảnh xung quanh:** Yếu tố ngữ cảnh xung quanh biển báo xung quanh gồm cây cối, xe cộ và các biển hiệu của các cửa hàng đôi khi sẽ gây trùng màu với biển báo cần nhận dạng.
6. **Tiền xử lý dữ liệu:** Đã thực hiện các bước tiền xử lý ảnh như chuẩn hóa, xử lý nhiễu, augmentations để cải thiện hiệu suất nhận dạng.

Hệ thống Nhận dạng biển báo giao thông được đề xuất thiết kế trong đề tài nghiên cứu này như Hình 3 phía dưới gồm 2 phần chính: Phần cứng gồm máy ảnh (camera) và máy tính; và Phần mềm là thuật toán YOLOv7.



Hình 3. Hệ thống Nhận dạng biển báo giao thông.

Máy ảnh được lắp trên xe để thực hiện chụp biển báo giao thông trên đường. Hình ảnh thu thập được sẽ được xử lý bằng thuật toán YOLOv7 chạy trên máy tính cá nhân. Hệ thống được thiết kế sao cho phải nhận dạng được chính xác 9 loại biển báo giao thông bao gồm cấm rẽ trái, cấm rẽ phải, cấm rẽ trái-phải, cấm đỗ, cấm dừng-đỗ, cấm ô tô rẽ trái, cấm ô tô rẽ phải, và cấm đi thẳng. Hệ thống nhận dạng với thời gian tính toán thấp và đạt được độ chính xác tốt được đề xuất.

3.1 Thiết Kế Phần Cứng

Phần cứng của Hệ thống bao gồm máy ảnh (camera) và máy tính Nhúng (development kit KV260 Vision). Các phần cứng có các thông số kỹ thuật lần lượt được mô tả liệt kê trong Bảng 1 cho máy ảnh và Bảng 2 cho máy tính Nhúng.

Bảng 1. Thông số kỹ thuật của máy ảnh.

Tên sản phẩm	Webcam Xiaomi Xiaovv HD USB 6320S
Kích thước	100x25x50mm
Trọng lượng	115g
Chiều dài cáp	1.5m USB 2.0
Góc rộng	150 độ
Độ phân giải	HD 1920x1080
Tốc độ khung hình	30 fps
Hệ điều hành hỗ trợ	Windows7 / 8/10, Linux2.4.6 trở lên, MacOS10.5 trở lên
Định dạng video	H.264 H.265 MJPG NV12 YUY2

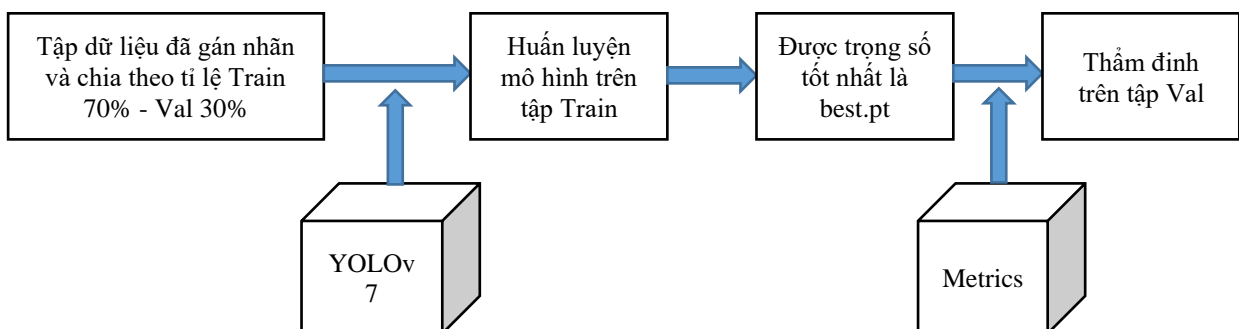
Bảng 2. Thông số kỹ thuật của máy tính Nhúng.

Tên sản phẩm	Xilinx Kria KV260 Vision AI
Kích thước	119mm x 140mm x 36mm
Giải pháp làm mát	Quạt + tản nhiệt
Các ô logic hệ thống	256K
Block RAM blocks	144
UltraRAM blocks	64
DSP slices	1.2K
Ethernet interface	One 10/100/1000 Mb/s
DDR memory	4GB (4 x 512Mb x 16 bit) [non-ECC]
Primary boot memory	512Mb QSPI
Secondary boot memory	SDHC card
Image sensor processor	OnSemi AP1302 ISP

3.2 Thiết Kế Giải Thuật

3.2.1. Lưu đồ giải thuật

Hệ thống phần cứng được thiết kế như trong Phần 3.1 được vận hành bởi phần mềm có thiết kế như lưu đồ giải thuật Hình 4.



Hình 4. Lưu đồ giải thuật Hệ thống Nhận dạng biên báo giao thông.

Như lưu đồ giải thuật Hình 4 mô tả, mô hình YOLOv7 được chọn trong nghiên cứu này sau khi nhóm tác giả đã phân tích so sánh với các mô hình khác. Kết quả phân tích cho thấy sự khác biệt ưu việt của mô hình YOLOv7 so với các mô hình khác như sau:

- **Tinh gọn và hiệu quả:** YOLOv7 được thiết kế để làm việc nhanh và hiệu quả, giúp nhận dạng đối tượng trong thời gian thực. Mô hình này thiên về tốc độ và hiệu suất vượt trội trên các mô hình khác.

- **Độ chính xác:** Mặc dù tốc độ là yếu tố quan trọng, YOLOv7 cũng đạt độ chính xác cao trong việc nhận dạng đối tượng.
- **Cấu trúc mô hình:** YOLOv7 sử dụng kiến trúc mạng nơ-ron sâu (deep neural network) với các lớp convolutional và lớp pooling để trích xuất đặc trưng của ảnh.
- **Multi-scale feature fusion:** YOLOv7 sử dụng kỹ thuật multi-scale feature fusion để kết hợp các đặc trưng từ các tầng khác nhau của mạng nơ-ron, giúp cải thiện khả năng nhận dạng các vật thể ở nhiều kích cỡ khác nhau trong ảnh.
- **Regularization techniques:** YOLOv7 sử dụng các kỹ thuật regularization như dropout, batch normalization để giúp chống overfitting và cải thiện khả năng tổng quát hóa của mô hình.
- **Training techniques:** YOLOv7 sử dụng các kỹ thuật huấn luyện tiên tiến như augmentation, layer-wise pre-training, để cải thiện hiệu suất huấn luyện và khả năng tổng quát hóa của mô hình.

Hơn nữa, mô hình YOLOv7 còn là một mô hình định vị vật thể và nhận dạng vật thể trong ảnh. Dưới đây là các thành phần chính trong mô hình YOLOv7:

- **Feature Extractor (Trích xuất đặc trưng):** YOLOv7 sử dụng một kiến trúc mạng CNN (Convolutional Neural Network) để trích xuất đặc trưng từ ảnh đầu vào. Các lớp convolutional và pooling đóng vai trò quan trọng trong việc học và trích xuất thông tin quan trọng về ảnh.
- **Backbone Network:** YOLOv7 sử dụng một mạng lưới nơ-ron (neural network) làm cơ sở để trích xuất đặc trưng từ ảnh. Các phần của mô hình có thể bao gồm ResNet, DarkNet hoặc một loại backbone network khác.
- **Multi-scale Feature Fusion:** YOLOv7 sử dụng kỹ thuật multi-scale feature fusion để kết hợp các đặc trưng từ các tầng khác nhau của mạng backbone, giúp cải thiện khả năng nhận dạng vật thể ở nhiều kích cỡ khác nhau trong ảnh.
- **Detection Head (Đầu nhận dạng):** Phần này của mô hình YOLOv7 chịu trách nhiệm dự đoán các bounding box và lớp tương ứng với các vật thể trong ảnh. Đối với mỗi grid cell, mô hình dự đoán một số lượng predefined bounding box và xác suất của các lớp vật thể.
- **Loss Function (Hàm mất mát):** YOLOv7 sử dụng một hàm mất mát đặc biệt để đánh giá hiệu suất của mô hình. Hàm mất mát này bao gồm các thành phần như loss của bounding box prediction, loss của class prediction, loss của object confidence và loss của no-object confidence.
- **Post-processing (Xử lý kết quả):** Sau khi mô hình dự đoán bounding box và xác suất của các lớp, một công đoạn xử lý kết quả được thực hiện để lọc ra các vật thể có độ chắc chắn cao và loại bỏ các vật thể dự đoán không chính xác.

Đây là các thành phần chính trong mô hình YOLOv7, mỗi phần có vai trò quan trọng trong quá trình nhận dạng và xác định vật thể trong ảnh một cách nhanh chóng và chính xác.

3.2.2. Phương pháp huấn luyện

Phần này của bài báo, nhóm tác giả sẽ mô tả chi tiết cơ sở để huấn luyện mô hình bao gồm: cách thu thập dữ liệu như thế nào, dữ liệu được chia sẻ hay được thu thập bởi tác giả, chất lượng ảnh thu thập, và yếu tố môi trường/thời tiết.

- **Dữ liệu huấn luyện được thu thập bởi chính nhóm tác giả:** thu thập các biển báo giao thông khi đi trên các tuyến đường ở thành phố Hồ Chí Minh. Dữ liệu được chia sẻ công khai trên RoboFlow cụ thể ở đường link <https://universe.roboflow.com/c-trn-thnh-mnh/traffic-sign-9cl>.
- **Chất lượng hình ảnh:** có độ sắc nét cao với size 1024x1024 trở lên chiếm 72% tổng số hình ảnh thu thập được. Các hình ảnh được chụp với các góc độ phù hợp khi di chuyển bằng xe.
- **Yếu tố môi trường:** khi thu thập đa dạng gồm chụp lúc trời trưa, chiều và tối, thời tiết bình thường. Với điều kiện chụp sáng thì ảnh đầu vào hiển thị rõ và tốt, ảnh chụp buổi tối thì biển báo có một lớp phản quang nên khi chiếu đèn xe thì vẫn nhận dạng rõ.
- **Số lượng hình ảnh:** trong tập huấn luyện được thu thập không đồng nhất (khác nhau) bởi tần suất xuất hiện khi tìm và chụp khá ít, một số biển báo xuất hiện ở các khu vực nhất định. Ngoài ra khi chạy qua mô hình thì độ chính xác các biển báo có nhiều hình ảnh không chênh lệch quá nhiều so với các biển báo có ít hình hơn.

THIẾT KẾ HỆ THỐNG NHẬN DẠNG BIÊN BẢO GIAO THÔNG VỚI ỨNG DỤNG YOLO



Hình 5. Mô tả dán nhãn ảnh.

Sau đó, nhóm tác giả sẽ sử dụng ứng dụng Roboflow để thực hiện việc upload cũng như gán nhãn các ảnh cho quá trình huấn luyện. Sau khi upload ảnh lên, Roboflow sẽ loại bỏ các ảnh trùng lặp và hiển thị tổng số ảnh đã được upload lên màn hình. Sau đó chúng ta sẽ “assign” để tiến hành thực hiện việc gán nhãn ảnh như mô tả trong Hình 5.

Sau khi việc gán nhãn các hình ảnh đã hoàn thành thì chúng ta có thể kiểm tra lại các ảnh đã gán nhãn và sửa lại nếu sai. Tiếp theo ta sẽ chia dữ liệu thành 70% cho tập huấn luyện và 30% cho tập thực nghiệm kiểm chứng. Chúng ta chuẩn bị hình ảnh và dữ liệu cho việc huấn luyện Hệ thống bằng cách biên dịch chúng thành một phiên bản mới. Thử nghiệm với các cấu hình khác nhau để đạt kết quả tập luyện tốt hơn. Các bước tiến hành như sau:

- Thay đổi kích thước (Resize) ảnh thành 640x640.
- Tăng cường hình ảnh bằng cách tạo thêm mẫu huấn luyện cho mô hình. Ở đây ta sử dụng 5 tùy chọn tăng cường như hình dưới gồm: Cắt ảnh, kéo ảnh qua trái phải, độ sáng, làm mờ và thêm nhiễu.
- Khởi tạo (Generate): Xem lại các lựa chọn và chọn kích cỡ phiên bản để tạo ảnh chụp nhanh cho bộ dữ liệu với các phép biến đổi đã áp dụng. Phiên bản lớn sẽ huấn luyện lâu hơn nhưng sẽ cho ra kết quả tốt hơn.
- Sau khi khởi tạo thì chúng ta đã có 7946 tấm ảnh trong đó gồm 6945 hình ảnh cho tập huấn luyện và 1001 hình ảnh cho tập thực nghiệm kiểm tra. Số lượng tấm ảnh dùng cho huấn luyện được liệt kê như Hình 6.

Cấm dừng và đỗ	752
Cấm rẽ trái	735
Cấm rẽ phải	713
Cấm đỗ	629
Cấm oto rẽ phải	191
Cấm rẽ trái và phải	177
Cấm quay đầu xe	171
Cấm oto rẽ trái	143
Cấm đi thẳng	109

Hình 6. Số lượng các loại ảnh huấn luyện.

- Xuất hình ảnh về máy tính để huấn luyện với mô hình YOLOv7. Chọn format cho hình ảnh là YOLOv7 Pytorch cho mô hình.

Việc huấn luyện Hệ thống được thực hiện trên Google Colab Pro vì có thể sử dụng được GPU cao cấp. Bạn có thể nâng cấp chế độ cài đặt GPU của sổ tay trong mục Thời gian chạy > Thay đổi loại thời gian

chạy của trình đơn để bật Trình tăng tốc cao cấp. Tùy theo tình trạng sẵn có, khi chọn GPU cao cấp, bạn sẽ có quyền dùng GPU V100 hoặc A100 của Nvidia. Trong bài này sẽ dùng A100 để có RAM nhiều hơn, cụ thể là 40GB cho mỗi epoch huấn luyện. Các bước tuần tự như sau:

- Clone YOLOv7 repo.
- Truy cập GG Colab và cài đặt các requirements. Ở trong thư mục yolov7, ta sẽ thấy thư mục tên là “requirements.txt”. Mở thư mục lên và remove dòng 11 và 12 ghi là torch và torchvision. Tạo một thư mục mới với tên là “requirements_gpu.txt” trong thư mục yolov7.
 - from google.colab import drive
 - drive.mount('/content/drive')
- Chuẩn bị dữ liệu: Tạo 2 thư mục: “images” và “labels”. Thư mục “images” chứa các hình ảnh và thư mục “labels” chứa các nhãn như trên.
- Thiết lập thư mục config: Mở file “coco.yaml” trong thư mục data và xóa 4 dòng đầu tiên.
 - Thiết lập “train: data/train”
 - Thiết lập “val: data/train”
 - Thiết lập “nc: 9” (số lớp)
 - Thiết lập “names: ['Cam do', 'Cam dung va do', 'Cam re phai', 'Cam ô tô re trai']”
- Tiếp theo mở yolov7/cfg/training và mở thư mục “yolov7.yaml”. Thay đổi ở dòng thứ 2; thay đổi nc: 9. “nc” là viết tắt của từ number of classes.
- Tải trọng số pre-trained của mô hình YOLOv7. Chúng ta sẽ sử dụng trọng số pre-trained yolov7-.pt và để ở thư mục yolov7.
- Bắt đầu huấn luyện trên mô hình YOLOv7 với tập dữ liệu đã tạo ở trên. Chạy các câu lệnh để cài đặt các thư viện yêu cầu.

4 KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

4.1 Thiết lập điều kiện thực nghiệm

Hệ thống được đề xuất thiết kế đã được tiến hành kiểm thử thực tế trong giao thông tại thành phố Hồ Chí Minh, Việt nam như mô tả trong Hình 7. Các hình ảnh có chứa 9 loại biển báo giao thông, mà hệ thống được đề xuất thiết kế để nhận dạng, được trích xuất từ các đoạn video dài từ 240 giây đến 360 giây.



Hình 7. Hình ảnh kiểm thử thực nghiệm.

Dùng câu lệnh để kiểm tra hình ảnh test đã được up lên drive. Ta sẽ sử dụng trọng số best.pt sau khi đã huấn luyện (đưa best.pt ra thư mục yolov7) như sau:

```
!python detect.py --weights best.pt --conf 0.4 --img-size 640 --source File_test_5.jpg
```

4.2 Kết quả thực nghiệm và đánh giá kết quả

4.2.1. Kết quả thực nghiệm

Sau khi chạy huấn luyện với phiên bản yolov7.pt thì ta thu được kết quả đầu tiên như Bảng 3.

```
!python train.py --workers 8 --device 0 --batch-size 8 --epochs 25 --img 640 640 --hyp data/hyp.scratch.custom.yaml --name yolov7-custom --weights yolov7.pt
```

Bảng 3. Kết quả huấn luyện thô.

Class	Images	Labels	Precision	Recall	mAP@.5	mAP@.5:.95
All	1001	1132	0.885	0.898	0.944	0.778
Cấm đi thẳng	1001	38	0.903	0.789	0.888	0.712
Cấm đỗ	1001	211	0.922	0.893	0.930	0.722
Cấm dừng và đỗ	1001	267	0.926	0.941	0.961	0.846
Cấm oto rẽ phải	1001	56	0.873	0.793	0.932	0.743
Cấm oto rẽ trái	1001	39	0.784	0.949	0.944	0.772
Cấm quay đầu xe	1001	48	0.835	0.736	0.891	0.739
Cấm rẽ phải	1001	208	0.952	0.986	0.988	0.824
Cấm rẽ trái	1001	211	0.972	0.972	0.991	0.834
Cấm rẽ trái và phải	1001	54	0.802	0.926	0.969	0.807

Sau khi chạy huấn luyện với phiên bản yolov7.pt thì ta thu được trọng số tốt nhất và tiếp tục tinh chỉnh các tham số và tăng epoch để cho ra kết quả huấn luyện tinh cuối cùng đã được cải thiện như trong Bảng 4.

Bảng 4. Kết quả huấn luyện đã cải thiện với các tham số đã được tinh chỉnh.

Lớp	Ảnh	Nhãn	Độ chính xác	Thu hồi	mAP@.5	mAP@.5:.95
All	1001	1132	0.967	0.927	0.966	0.806
Cắm đi thẳng	1001	38	0.967	0.921	0.934	0.745
Cắm đỗ	1001	211	0.976	0.773	0.953	0.765
Cắm dừng và đỗ	1001	267	0.965	0.925	0.956	0.846
Cắm oto rẽ phải	1001	56	0.961	0.877	0.937	0.764
Cắm oto rẽ trái	1001	39	0.921	0.949	0.945	0.749
Cắm quay đầu xe	1001	48	0.947	0.979	0.986	0.854
Cắm rẽ phải	1001	208	0.990	0.964	0.993	0.827
Cắm rẽ trái	1001	211	0.986	0.974	0.994	0.847
Cắm rẽ trái và phải	1001	54	0.989	0.981	0.995	0.853

Việc huấn luyện trên liên quan đến một số yếu tố ảnh hưởng đến kết quả với các khía cạnh sau:

- **Chất lượng và tính đa dạng của bộ dữ liệu:** Chất lượng và tính đa dạng của tập dữ liệu huấn luyện ảnh hưởng đáng kể đến hiệu suất của mô hình. Bộ dữ liệu chất lượng cao với hình ảnh biển báo giao thông được dán nhãn chính xác là rất cần thiết. Các bộ dữ liệu đa dạng sẽ bao gồm các điều kiện ánh sáng, thời tiết, quan điểm và biến thể biển báo khác nhau.
- **Độ chính xác của nhãn:** Chú thích hộp giới hạn chính xác cho biển báo giao thông là rất quan trọng. Lỗi trong chú thích có thể dẫn đến việc đào tạo không chính xác và ảnh hưởng đến khả năng khái quát hóa của mô hình.
- **Kích thước tập dữ liệu:** Kích thước tập dữ liệu đầy đủ là điều cần thiết cho việc đào tạo mạnh mẽ. Dữ liệu không đủ có thể gây ra hiện tượng overfitting, trong khi dữ liệu overfitting có thể làm chậm quá trình đào tạo. Vì thế mỗi loại biển báo cần có nhiều hình ảnh dưới các góc độ khác nhau để tăng sự chính xác cũng như tránh các tình trạng overfitting có thể xảy ra.
- **Sự ảnh hưởng của từng loại biển báo:** Các biển báo đều có sự tương đồng về hình dáng nên khi lấy dữ liệu và gán nhãn cần lấy hình ảnh thể hiện sự khác biệt giữa các biển báo, gán nhãn chính xác để tránh bị sai sót làm ảnh hưởng tới kết quả huấn luyện. Từ đó phải cần nhiều dữ liệu để mô hình học tốt hơn.

Phương pháp đề xuất được hiện nhiều lần trên các phiên bản phụ của giải thuật Yolov7 khác nhau và kết quả so sánh độ chính xác được tổng hợp trong Bảng 5.

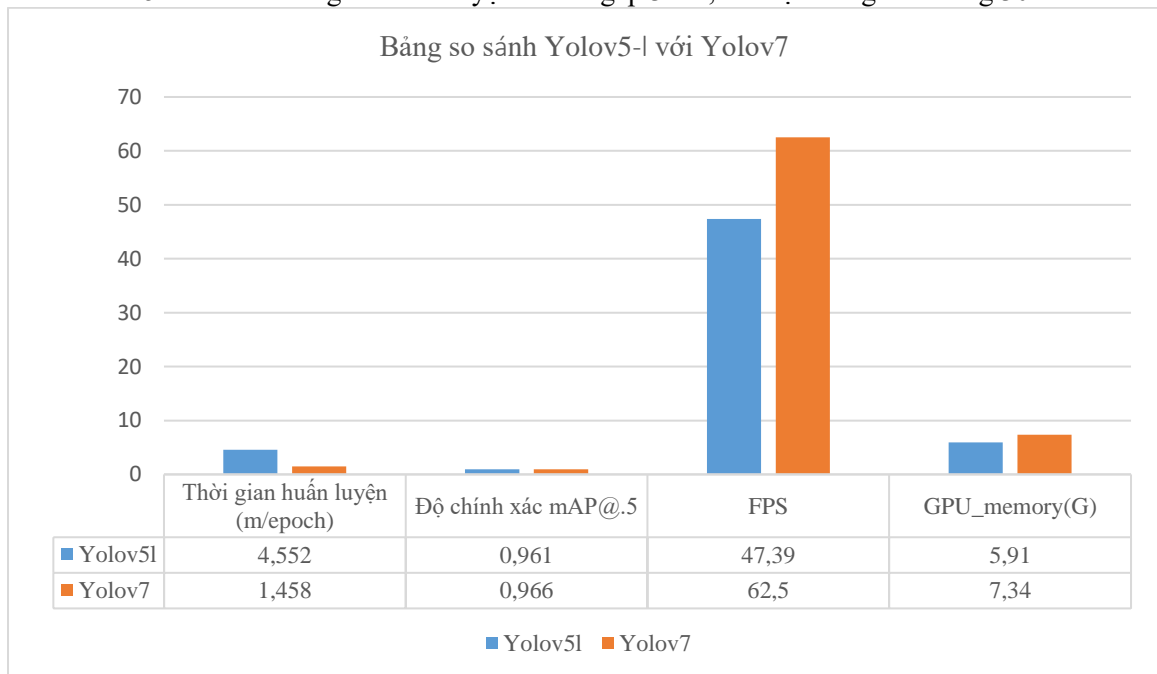
Bảng 5. So sánh độ chính xác các phiên bản Yolov7 khác nhau.

Model	Parameters	GPU mem	Precision	Recall	mAP@.5	mAP@.5:.95	Các thay đổi
Yolov7	37248560	7.34G	0.885	0.898	0.944	0.778	25 epoch, batch size: 8
Yolov7	37248560	24.2G	0.948	0.950	0.966	0.781	add 10 epoch, batch size: 16, loss_ota: 0
Yolov7	37248560	11.9G	0.967	0.927	0.966	0.806	add 10 epoch, batch size: 16, lr0: 0.001, lrf: 0.01
Yolov7	37248560	24.2G	0.909	0.943	0.956	0.792	add 10 epoch, batch size: 16, lr0: 0.001, iou_t: 0.4
Yolov7	37248560	24.2G	0.907	0.905	0.945	0.780	add 10 epoch, batch size: 16, lr0: 0.001,

THIẾT KẾ HỆ THỐNG NHẬN DẠNG BIÊN BẢO GIAO THÔNG VỚI ỨNG DỤNG YOLO

							iou_t: 0.4, fl_gamma: 1.5
Yolov7	37248560	24.2G	0.904	0.942	0.965	0.793	add 10 epoch, batch size: 16, lr0: 0.001, lrf: 0.001
Yolov7	37248560	21.2G	0.947	0.935	0.965	0.807	add 25 epoch, batch size: 32, lr0: 0.001, lrf: 0.01
Yolov7-x	70866630	9.33G	0.933	0.948	0.954	0.774	25 epoch, batch size: 8
Yolov7-w6	69880104	5.16G	0.906	0.899	0.829	0.731	25 epoch, batch size: 8

Phương pháp đề xuất trong bài báo này đã được tiến hành thực nghiệm trên phiên bản Yolov5-1 và kết quả so sánh tổng hợp được mô tả như Hình 8. Kết quả cho thấy phiên bản Yolov7 có những ưu điểm vượt trội hơn Yolov5 như sau: thời gian huấn luyện nhanh gấp 3 lần, tốc độ khung hình tăng 30%.



Hình 8. So sánh kết quả 2 phiên bản Yolov5-1 và Yolov7.

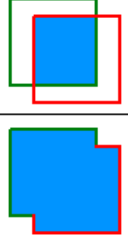
4.2.2. Phương pháp đánh giá

Phương pháp đánh giá đối với mỗi class sẽ được sử dụng thang đo F1, Độ chính xác (Precision) và Thu hồi (Recall) và đối với toàn Lớp (Class) sẽ được sử dụng thang đo mAP.5 và mAP.5:.95.

- Để phát hiện, một cách phổ biến để xác định xem một đề xuất của đối tượng có đúng hay không là Intersection over Union (IoU, IU). Việc này lấy tập A gồm các pixel đối tượng được đề xuất và tập hợp các pixel đối tượng thực B và tính toán:

$$IoU(A,B) = \frac{area(A \cap B)}{area(A \cup B)}$$

- Hình ảnh bên dưới chỉ rõ IOU giữa hộp giới hạn thực tế cơ bản (màu xanh lá cây) và hộp giới hạn được phát hiện (màu đỏ).

$$IOU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{area of overlap}}{\text{area of union}}$$


- Một số khái niệm cơ bản được sử dụng bởi các số liệu:
 - True Positive (TP) : Phát hiện chính xác. *Phát hiện với $IOU \geq threshold$*
 - False Positive (FP) : Phát hiện sai. *Phát hiện với $IOU < threshold$*
 - False Negative (FN) : Không phát hiện được ground truths
 - True Negative (TN) : Không áp dụng. Nó sẽ đại diện cho một phát hiện sai đã được sửa chữa. Trong tác vụ phát hiện đối tượng, có nhiều hộp giới hạn có thể không được phát hiện trong ảnh. Do đó, TN sẽ là tất cả các hộp giới hạn có thể không được phát hiện chính xác (rất nhiều hộp có thể có trong một hình ảnh). Đó là lý do tại sao nó không được sử dụng bởi các số liệu.
 - *thres(ngưỡng)* : tùy thuộc vào số liệu, nó thường được đặt thành 50%, 75% hoặc 95%.
- Độ chính xác (Precision): Độ chính xác là khả năng của một mô hình chỉ xác định được các đối tượng có liên quan. Đó là tỷ lệ phần trăm dự đoán tích cực chính xác và được đưa ra bởi:

$$\text{Độ chính xác} = \frac{TP}{TP+FP} = \frac{TP}{\text{Tất cả các mẫu nhận diện được}}$$

- Thu hồi (Recall): Thu hồi là khả năng của một mô hình để tìm tất cả các trường hợp có liên quan (tất cả các hộp giới hạn trong ground truths). Đó là tỷ lệ phần trăm của kết quả True positive thực sự được phát hiện trong số tất cả các ground truths có liên quan và được đưa ra bởi:

$$\text{Thu hồi} = \frac{TP}{TP+FN} = \frac{TP}{\text{Tất cả ground truths}}$$

Đường cong (**Precision x Recall**) là một cách hay để đánh giá hiệu suất của bộ phát hiện đối tượng vì độ tin cậy được thay đổi bằng cách vẽ đường cong cho từng lớp đối tượng. Trình phát hiện đối tượng thuộc một lớp cụ thể được coi là tốt nếu độ chính xác của nó vẫn cao khi mức thu hồi tăng, điều đó có nghĩa là nếu bạn thay đổi ngưỡng tin cậy thì độ chính xác và mức thu hồi sẽ vẫn cao. Một cách khác để xác định một trình phát hiện đối tượng tốt là tìm kiếm một trình phát hiện chỉ có thể xác định các đối tượng có liên quan (0 False Positives = độ chính xác cao), tìm tất cả các đối tượng thực tế cơ bản (0 False Negatives = thu hồi cao).

Trình phát hiện đối tượng kém cần tăng số lượng đối tượng được phát hiện (tăng False Positives = độ chính xác thấp hơn) để truy xuất tất cả các đối tượng ground truths (có khả năng thu hồi cao). Đó là lý do tại sao đường cong Precision x Recall thường bắt đầu với các giá trị có độ chính xác cao và giảm dần khi mức thu hồi tăng.

Một cách khác để so sánh hiệu suất của máy dò đối tượng là tính **diện tích dưới đường cong** của đường cong Precision x Recall. Vì các đường cong thường là những đường cong ngoằn ngoèo đi lên và đi xuống, nên việc so sánh các đường cong khác nhau (các bộ dò khác nhau) trong cùng một đồ thị thường không phải là một nhiệm vụ dễ dàng - bởi vì các đường cong có xu hướng giao nhau rất thường xuyên. Đó là lý do tại sao **Độ chính xác Trung bình (Average Precision – AP)**, một thước đo bằng số, cũng có thể giúp chúng ta so sánh các máy dò khác nhau. Trong thực tế AP là độ chính xác được tính trung bình trên tất cả các giá trị thu hồi trong khoảng từ 0 đến 1.

Thông thường, IoU > 0,5 có nghĩa là đã thành công, nếu không thì là thất bại. Đối với mỗi lớp, người ta có thể tính toán:

- True Positive (**TP(c)**): một đề xuất được đưa ra cho lớp **c** và thực sự có một đối tượng thuộc lớp **c**
- False Positive (**FP(c)**): một đề xuất được đưa ra cho lớp **c**, nhưng không có đối tượng nào thuộc lớp **c**

- Độ chính xác trung bình (Average Precision-AP) cho lớp c : $\frac{\#TP(c)}{\#TP(c) + \#FP(c)}$

Vì thế ta có MAP (độ chính xác trung bình trung bình) = $\frac{1}{|class|} \sum_{c \in class} \frac{\#TP(c)}{\#TP(c) + \#FP(c)}$

Điểm mAP@.5:.95 có nghĩa là mAP trung bình trên các ngưỡng IoU khác nhau, từ 0,5 đến 0,95, bước 0,05 (0,5, 0,55, 0,6, 0,65, 0,7, 0,75, 0,8, 0,85, 0,9, 0,95).

5 KẾT LUẬN

Bài báo này đề cập đến việc đề xuất một phương pháp nhằm cải tiến một hệ thống nhận dạng biên báo giao thông trong môi trường giao thông đô thị tại thành phố Hồ Chí Minh, Việt nam. Thông qua các kết quả thực nghiệm, hệ thống được đề xuất thiết kế trong bài báo này đã chứng tỏ khả năng nhận biết được 9 loại biên báo giao thông bao gồm biên báo cấm rẽ trái, biên báo cấm rẽ phải, biên cấm rẽ trái-phải, biên báo cấm dừng-cấm đỗ, biên báo cấm đỗ, biên báo cấm ô tô rẽ phải, biên báo cấm ô tô rẽ trái, biên báo cấm quay đầu và cuối cùng là biên báo cấm đi thẳng; với độ chính xác lần lượt là 99.4%, 99.3%, 99.5%, 95.6%, 95.3%, 93.7%, 94.5%, 98.6% và 93.4%.

ACKNOWLEDGMENT

Nghiên cứu được tài trợ bởi Trường Đại học Công nghệ Thông tin – ĐHQG-HCM.

REFERENCES

- [1] Gudigar, A., Chokkadi, S. & U, R. A review on automatic detection and recognition of traffic sign. *Multimed Tools Appl* 75, 333–364 (2016). <https://doi.org/10.1007/s11042-014-2293-7>.
- [2] M. Swathi and K. V. Suresh, "Automatic traffic sign detection and recognition: A review," 2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET), Chennai, India, 2017, pp. 1-6, doi: 10.1109/ICAMMAET.2017.8186650.
- [3] B. Sanyal, R. K. Mohapatra and R. Dash, "Traffic Sign Recognition: A Survey," 2020 International Conference on Artificial Intelligence and Signal Processing (AISP), Amaravati, India, 2020, pp. 1-6, doi: 10.1109/AISP48273.2020.9072976.
- [4] C. Liu, S. Li, F. Chang and Y. Wang, "Machine Vision Based Traffic Sign Detection Methods: Review, Analyses and Perspectives," in *IEEE Access*, vol. 7, pp. 86578-86596, 2019, doi: 10.1109/ACCESS.2019.2924947.
- [5] Zhu, Y., Yan, W.Q. Traffic sign recognition based on deep learning. *Multimed Tools Appl* 81, 17779–17791 (2022). <https://doi.org/10.1007/s11042-022-12163-0>.
- [6] <https://www.datacamp.com/blog/yolo-object-detection-explained>, last accessed 24 Oct. 23.
- [7] <https://deci.ai/blog/history-yolo-object-detection-models-from-yolov1-yolov8/>, last accessed 24 Oct. 23.
- [8] Soylu, E., Soylu, T. A performance comparison of YOLOv8 models for traffic sign detection in the Robotaxi-full scale autonomous vehicle competition. *Multimed Tools Appl* 83, 25005–25035 (2024). <https://doi.org/10.1007/s11042-023-16451-1>.
- [9] Han T, Sun L, Dong Q. An Improved YOLO Model for Traffic Signs Small Target Image Detection. *Applied Sciences*. 2023; 13(15):8754. <https://doi.org/10.3390/app13158754>.
- [10] Qureshi, Rizwan , Mohammed Gamal Ragab, Said Jadid Abdulkader, Amgad Muneer, Alawi Alqushaib, Ebrahim Hamid Sumiea, and Hitham Alhussian, 2023. "A comprehensive systematic review of yolo for medical object detection (2018 to 2023)". <https://doi.org/10.36227/techrxiv.23681679.v1>

METHOD TO IMPROVE THE YOLO-BASED TRAFFIC SIGN IDENTIFICATION SYSTEM

TRẦN THỊNH MẠNH ĐỨC^{1,2}, ĐỖ TRÍ NHỰT^{1,2,*}

¹ Department of Embedded Systems and Robotics, Faculty of Computer Engineering, University of Information Technology, Linh Trung Ward, Thu Duc City 71308, Ho Chi Minh City, Việt nam

² Vietnam National University in Ho Chi Minh City, Linh Trung Ward, Thu Duc City 71308, Ho Chi Minh City, Việt nam

*Tác giả liên hệ: trinhutdo@gmail.com, trinhutdo@uit.edu.vn

Abstract. The system of accurately and quickly identified traffic signs has contributed to improving traffic management in the era of digital transformation according to the current government policy. This system has been researched and developed strongly in recent years from classic algorithms on image processing to algorithms with artificial intelligence applications such as the You Only Look Once (YOLO) algorithm based on the technology. Computer vision technique is a representative worthy of attention. In this article, a method to improve the traffic sign recognition system using the YOLO application is studied in terms of the number of recognized signs, format, recognition speed and accuracy. The system has been improved to recognize 9 different types of traffic signs including: no turning left sign, no turning right sign, no left-right turn sign, no stopping-no parking sign, no stopping sign, parking, signs prohibiting cars from turning right, signs prohibiting cars from turning left, signs prohibiting turning around, and signs prohibiting cars from going straight. The system was improved and retrained with images collected from roads in Ho Chi Minh City including 735 images of signs prohibiting left turns, 713 images of signs prohibiting right turns, 177 images prohibiting left turns - right, 752 photos of no-stop-no-parking signs, 629 photos of no-parking signs, 191 photos of prohibiting cars from turning right, 143 photos of prohibiting cars from turning left, 171 prohibiting turning around and 109 prohibiting going straight.

Keywords. traffic sign detection, traffic sign recognition, YOLO algorithm, YOLT algorithm.

Received on October 14 – 2023

Revised on April 12 – 2024