

# THIẾT KẾ HỆ THỐNG NHẬN DẠNG BIỂN BÁO GIAO THÔNG VỚI ỨNG DỤNG YOLO

**Trần Thịnh Mạnh Đức<sup>(1)</sup>, Đỗ Trí Nhựt<sup>(1)</sup>**

*(1) Trường Đại học Công nghệ Thông tin – VNU HCM*

*Ngày nhận bài 6/11/2023; Ngày gửi phản biện 6/12/2023; Chấp nhận đăng 02/3/2024*

*Liên hệ email: trinhutdo@gmail.com*

<https://doi.org/10.37550/tdmu.VJS/2024.02.533>

---

## **Tóm tắt**

Một hệ thống nhận dạng biển báo giao thông an toàn và đáng tin cậy hơn là nhu cầu của người lái xe và cũng là điểm nóng nghiên cứu của các nhà sản xuất ô tô hiện nay. Để giải quyết nhu cầu trên, các tác giả đề xuất trong bài báo này Hệ thống nhận dạng biển báo giao thông dựa trên kỹ thuật thị giác máy tính và thuật toán You Only Look Once (YOLO). Hệ thống được thiết kế 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. Để huấn luyện Hệ thống, 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.

**Từ khóa:** biển báo giao thông, thuật toán YOLO, thuật toán YOLT

## **Abstract**

### **DESIGN OF A TRAFFIC SIGN RECOGNITION SYSTEM WITH THE YOLO APPLICATION**

A safer and more reliable traffic sign recognition system is a need of drivers and is also a research hot spot for automobile manufacturers today. To address the above need, the authors propose in this article a traffic sign recognition system based on computer vision techniques and the You Only Look Once (YOLO) algorithm. The system is designed to recognize 9 different types of traffic signs including: No Left Turn, No Right Turn, No Left Turn and No Right Turn, No Stopping and Parking, No Parking, No Automobiles turning right, No Automobiles turning left, No U turns, and No Going straight- ahead. To train the proposed system, images were collected from roads in Ho Chi Minh City including 735 signs with name No Left Turn, 713 signs with name No Right Turn, 177

*images with No Left Turn and No Right Turn, 752 images with name No Stopping and Parking, 629 signs with name No Parking, 191 signs of prohibiting cars from turning right, 143 signs of prohibiting cars from turning left, 171 signs with name No U turns and 109 signs of No Going straight-ahead. Finally, the system is experimentally tested in the field for recognition accuracy according to the mAP@.5 measure.*

---

## 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) (Gudigar và cs., 2016). 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. Gudigar và cs. (2016) tổng hợp và đánh giá về đề tài này 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. Gudigar và cs. (2016) 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.

Swathi và cs. (2017) trình bày đánh giá về các phương pháp phát 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 viết 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. Sanyal và cs. (2020) 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. Liu và cs. (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. Zhu và cs. (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 (ví 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 (2023). 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. 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.

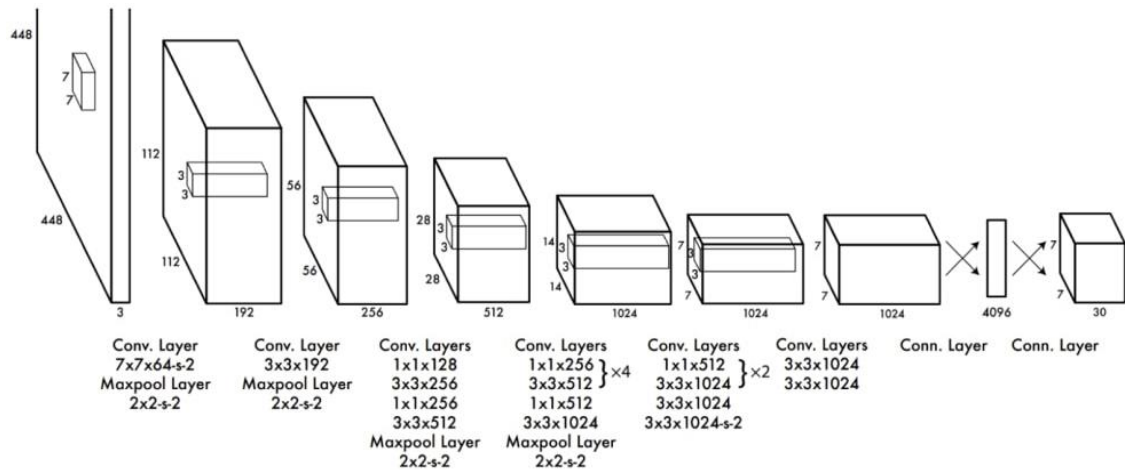
Trong bài báo này, các tác giả đề xuất thiết kế Hệ thống nhận dạng biển báo giao thông dựa trên kỹ thuật thị giác máy tính và thuật toán YOLO, phiên bản thứ 7. Hệ thống đề xuất được thiết kế để nhận dạng 9 loại biển báo giao thông ở Việt nam bao gồm: cấm rẽ trái, cấm rẽ phải, cấm rẽ trái-phải, cấm đỗ, 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 diện 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.

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

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 đồ. 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 triệt 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.

Model	#Param.	FLOPs	Size	AP <sup>val</sup>	AP <sup>val</sup> <sub>50</sub>	AP <sup>val</sup> <sub>75</sub>	AP <sup>val</sup> <sub>S</sub>	AP <sup>val</sup> <sub>M</sub>	AP <sup>val</sup> <sub>L</sub>
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	<b>51.2%</b>	<b>69.7%</b>	<b>55.5%</b>	<b>35.2%</b>	<b>56.0%</b>	<b>66.7%</b>
improvement	-43%	-15%	-	+0.4	+0.2	+0.2	+1.5	=	+1.3
YOLOv7-CSP-X [81]	96.9M	226.8G	640	52.7%	<b>71.3%</b>	57.4%	36.3%	57.5%	68.3%
YOLOv7-X	71.3M	189.9G	640	<b>52.9%</b>	71.1%	<b>57.5%</b>	<b>36.9%</b>	<b>57.7%</b>	<b>68.6%</b>
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	<b>35.2%</b>	<b>52.8%</b>	<b>37.3%</b>	<b>15.7%</b>	<b>38.0%</b>	<b>53.4%</b>
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%	<b>10.9%</b>	31.9%	51.5%
YOLOv7-tiny	6.2	3.5	320	<b>30.8%</b>	<b>47.3%</b>	<b>32.2%</b>	10.0%	<b>31.9%</b>	<b>52.2%</b>
improvement	-39%	-49%	-	=	=	=	-0.9	=	+0.7
YOLOv7-E6 [81]	115.8M	683.2G	1280	55.7%	73.2%	60.7%	40.1%	<b>60.4%</b>	69.2%
YOLOv7-E6	97.2M	515.2G	1280	<b>55.9%</b>	<b>73.5%</b>	<b>61.1%</b>	<b>40.6%</b>	60.3%	<b>70.0%</b>
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%	<b>42.4%</b>	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	<b>56.8%</b>	<b>74.4%</b>	<b>62.1%</b>	40.8%	<b>62.1%</b>	<b>70.6%</b>
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

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  $608 \times 608$  pixel, cao hơn độ phân giải  $416 \times 416$  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.

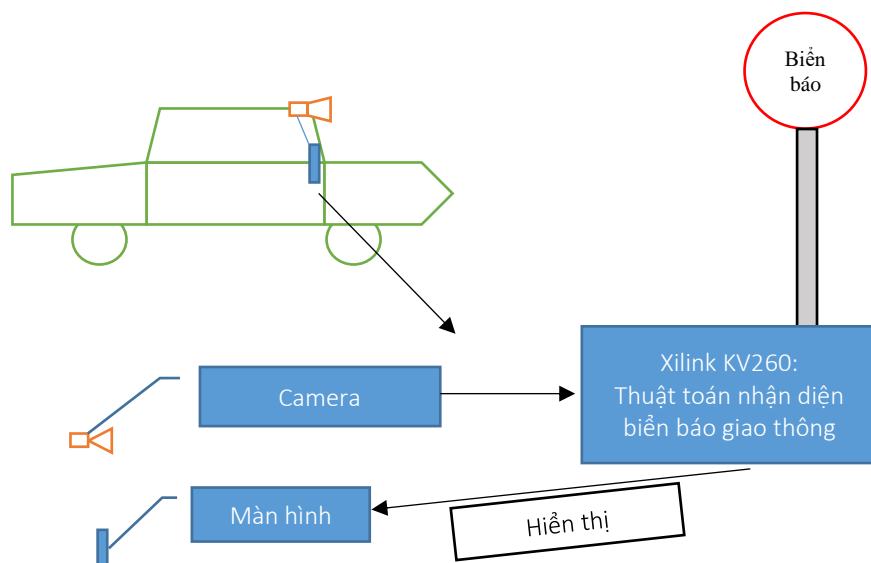
Về độ chính xác, YOLOv7 hoạt động tốt so với các thuật toán phát hiện đối tượng/vật thể khác. Nó đạt được độ chính xác trung bình là 37,2% ở ngưỡng IoU (intersection over union - giao điểm trên liên kết) là 0,5 trên bộ dữ liệu COCO phổ biến, có thể so sánh với các thuật toán phát hiện đối tượng tiên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

Hệ thống Nhận dạng biển báo giao thông được đề xuất thiết kế trong đề tài nghiên cứu này như hình 3, gồm 2 phần chính: Phần cứng gồm máy ảnh (camera) và máy tính;

và Phần mềm là thuật toán YOLOv7. 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.



**Hình 3.** Hệ thống Nhận diện biển báo giao thông.

### 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	100×25×50mm
Trọng lượng	115g
Chiều dài cáp	1.5m USB 2.0
Góc rộng	150 độ
Độ phân giải	HD 1920×1080
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

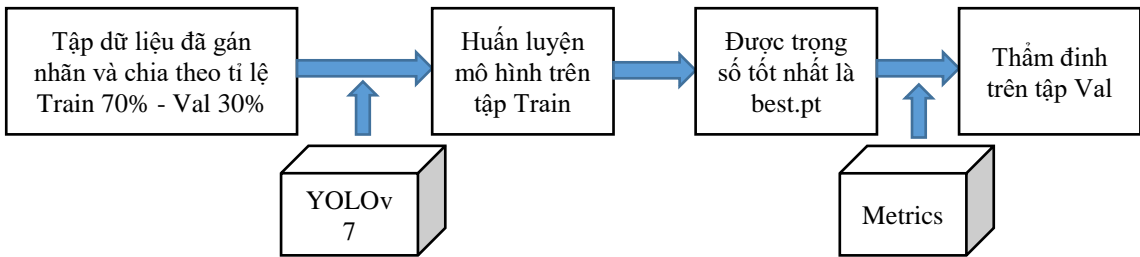
Tên sản phẩm	Xilinx Kria KV260 Vision AI
Kích thước	119mm × 140mm × 36mm
Giải pháp làm mát	Quạt + tản nhiệt
Các ô logic hệ thống	256K
Block RAM blocks	144

UltraRAM blocks	64
DSP slices	1.2K
Ethernet interface	One 10/100/1000 Mb/s
DDR memory	4GB (4 × 512Mb × 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

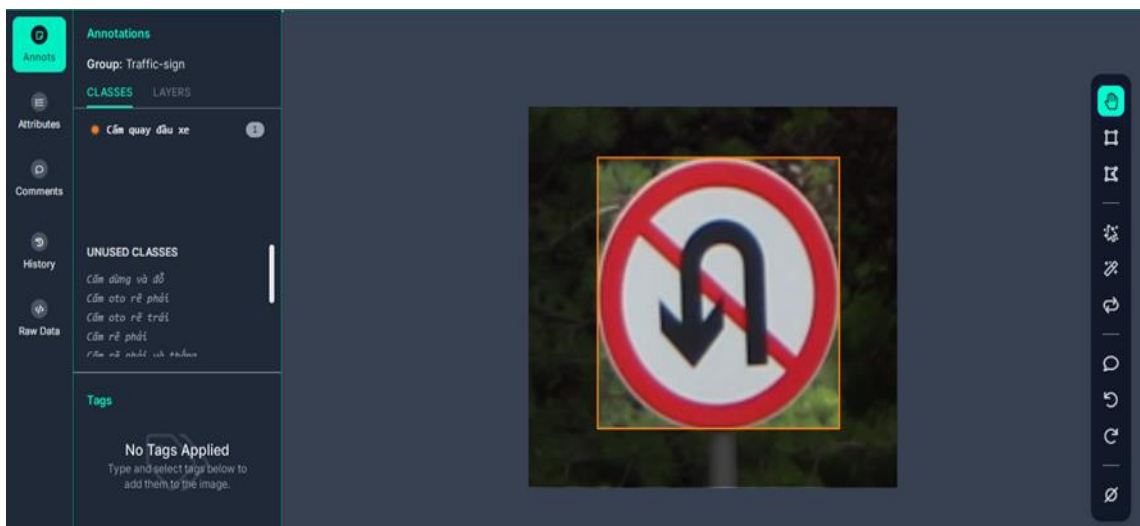
Hệ thống phân cứng được thiết kế như trong phần 3.1 được vận hành bởi phần mềm có thiết kế như lưu đồ giải thuật hình 4.



**Hình 4.** Lưu đồ giải thuật Hệ thống Nhận diện biển báo giao thông

#### 3.2.2. Phương pháp huấn luyện

Ta 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.



**Hình 5.** Mô tả dán nhãn ảnh

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 640×640.
- 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 diện, 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 hình 8.

```
!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
```

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95:
all	1001	1132	0.885	0.898	0.944	0.778
Cam di thang	1001	38	0.903	0.789	0.888	0.712
Cam do	1001	211	0.922	0.893	0.93	0.722
Cam dung va do	1001	267	0.926	0.941	0.961	0.846
Cam oto re phai	1001	56	0.873	0.893	0.932	0.743
Cam oto re trai	1001	39	0.784	0.949	0.944	0.772
Cam quay dau xe	1001	48	0.835	0.736	0.891	0.739
Cam re phai	1001	208	0.952	0.986	0.988	0.824
Cam re trai	1001	211	0.972	0.972	0.991	0.834
Cam re trai va phai	1001	54	0.802	0.926	0.969	0.807

**Hình 8.** Kết quả huấn luyện

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 cuối cùng đã được cải thiện như hình 9.

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95:
all	1001	1132	0.967	0.927	0.966	0.806
Cam di thang	1001	38	0.967	0.921	0.934	0.745
Cam do	1001	211	0.976	0.773	0.953	0.765
Cam dung va do	1001	267	0.965	0.925	0.956	0.846
Cam oto re phai	1001	56	0.961	0.877	0.937	0.764
Cam oto re trai	1001	39	0.921	0.949	0.945	0.749
Cam quay dau xe	1001	48	0.947	0.979	0.986	0.854
Cam re phai	1001	208	0.99	0.964	0.993	0.827
Cam re trai	1001	211	0.986	0.974	0.994	0.847
Cam re trai va phai	1001	54	0.989	0.981	0.995	0.853

**Hình 9.** Các trọng số đã được cải thiện

Hình 10 mô tả sự so sánh độ chính xác của các phiên bản YOLO khác nhau:

Model	Parameters	GPU_mem	P	R	mAP@.5	mAP@.5:.09	Ghi chú
YOLOv7	37248560	7.34G	0.885	0.898	0.944	0.778	25 epoch, batch size 8
YOLOv7	37248560	24.2G	0.948	<b>0.95</b>	<b>0.966</b>	0.781	thêm 10epoch, batch size 8-->16, loss_ota 1-->0
YOLOv7	37248560	11.9G	<b>0.967</b>	0.927	<b>0.966</b>	0.806	thêm 10epoch, batch size 8-->16, lr0 0.01-->0.001, lrf 0.1-->0.01
YOLOv7	37248560	24.2G	0.909	0.943	0.956	0.792	thêm 10epoch, batch size 8-->16, lr0 0.01-->0.001, iou_t 0.2-->0.4
YOLOv7	37248560	24.2G	0.907	0.905	0.945	0.78	thêm 10epoch, batch size 8-->16, lr0 0.01-->0.001, iou_t 0.2-->0.4, fl_gamma 0-->1.5
YOLOv7	37248560	24.2G	0.904	0.942	0.954	0.793	batch size 8-->16, lr0 0.01-->0.001, lrf 0.1-->0.001
YOLOv7	37248560	21.1G	0.947	0.934	0.965	<b>0.807</b>	thêm 25 epoch, batch size 32, lr0 0.01-->0.001, lrf 0.1-->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
YOLOv5-s	7039792	2.09G	0.935	0.921	0.948	0.765	25 epoch, batch size 8

**Hình 10.** So sánh độ chính xác

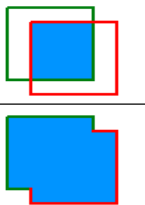
### 4.2.2. Phương pháp đánh giá

Phương pháp đánh giá đối với mỗi class sẽ được sử dụng thang đo F1, Precision và Recall và đối với toàn 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 thiết kế một hệ thống nhận diện biển báo giao thông trong môi trường giao thông đô thị tại thành phố Hồ Chí Minh, Việt nam. Hệ thống ứng dụng học sâu, trí tuệ nhân tạo với giải thuật YOLO, phiên bản thứ 7. 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 biển báo giao thông bao gồm biển báo cấm rẽ trái, biển báo cấm rẽ phải, biển 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 cuối cùng là biển 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%.

**Lời cảm ơn**

*Bài nghiên cứu này được tài trợ bởi Trường Đại học Công nghệ Thông tin – Đại học Quốc gia Thành phố Hồ Chí Minh.*

**TÀI LIỆU THAM KHẢO**

- [1] Gudigar, A., Chokkadi, S. & U, R. (2016). A review on automatic detection and recognition of traffic sign. *Multimed Tools Appl*, 75, 333-364. <https://doi.org/10.1007/s11042-014-2293-7>.
- [2] M. Swathi and K. V. Suresh (2017). *Automatic traffic sign detection and recognition: A review. International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET)*, Chennai, India, pp. 1-6, doi: 10.1109/ICAMMAET.2017.8186650.
- [3] B. Sanyal, R. K. Mohapatra and R. Dash (2020). *Traffic Sign Recognition: A Survey. International Conference on Artificial Intelligence and Signal Processing (AISP)*, Amaravati, India, pp. 1-6, doi: 10.1109/AISP48273.2020.9072976.
- [4] C. Liu, S. Li, F. Chang and Y. Wang (2019). Machine Vision Based Traffic Sign Detection Methods: Review Analyses and Perspectives. *IEEE Access*, vol. 7, pp. 86578-86596, 2019, doi: 10.1109/ACCESS.2019.2924947.
- [5] Zhu, Y., Yan, W.Q (2022). Traffic sign recognition based on deep learning. *Multimed Tools Appl*, 81, 17779–17791. <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.