Konya Mühendislik Bilimleri Dergisi, c. 13, s. 1, 220-237, 2025
Konya Journal of Engineering Sciences, v. 13, n. 1, 220-237, 2025
ISSN: 2667-8055 (Electronic)
DOI: 10.36306/konjes.1531208

REAL-TIME DETECTION OF TRAFFIC SIGNS WITH YOLO ALGORITHMS

*** Abdil KARAKAN**

Afyon Kocatepe University, Dazkırı Vocational Scholl, Electrical Department, Afyonkarahisar, TÜRKİYE
akarakan@aku.edu.tr

Highlights

- Customized Turkish Traffic Signs dataset consisting of 78895 images
- Preprocessing methods have been added to increase the efficiency of the model.
- The accuracy of the proposed YOLOv8 model is 99.60%.
- Detection was made in different weather conditions and at different speeds.

***Corresponding Author:** Abdil KARAKAN, akarakan@aku.edu.tr

Konya Mühendislik Bilimleri Dergisi, c. 13, s. 1, 220-237, 2025
Konya Journal of Engineering Sciences, v. 13, n. 1, 220-237, 2025
ISSN: 2667-8055 (Electronic)
DOI: 10.36306/konjes.1531208

REAL-TIME DETECTION OF TRAFFIC SIGNS WITH YOLO ALGORITHMS

*** Abdil KARAKAN**

Afyon Kocatepe University, Dazkırı Vocational Scholl, Electrical Department, Afyonkarahisar, TÜRKİYE
akarakan@aku.edu.tr

(Received: 10.08.2024; Accepted in Revised Form: 10.02.2025)

ABSTRACT: The number of electric vehicles is increasing day by day. The biggest reason for the increase in electric vehicles is their autonomous or semi-autonomous use feature. Autonomous or semi-autonomous driving; It is the movement of the vehicle with the data coming from the sensors, cameras, and sensors around the vehicle. The majority of traffic accidents are caused by driver errors. The most important of these mistakes is not obeying traffic rules. Autonomous or semi-autonomous driving largely prevents driver-related traffic accidents. The biggest problem of autonomous vehicles is the difficulties in detecting traffic signs in real-time. The locations, shapes, and scales of traffic signs are very different. Traffic signs are difficult to detect in real-world conditions due to their similarity to other objects. The study carried out real-time detection of traffic signs. For this purpose, images were taken from the camera placed inside the vehicle. A data set was created with these images. The more real environment images the data set consists of, the more accurate the real-time detection process increases. In this study, 8931 traffic sign images were taken from real environments. These images were taken from different locations, different lighting levels, and different distances. In addition, the number of data was increased to 78895 by adding grayscale, adding slope, blurring, adding variability, adding noise, changing image brightness, changing colour vividness, changing perspective, resizing, and positioning the images. With this study, the data set was adapted to the real environment. The created data set was used in 3 different versions of YOLOv5 architecture, YOLOv6, YOLOv7 and YOLOv8 architectures. As a result of the study, the highest accuracy was found to be 99.60%, F1-Score was 0.962 and mAP@.5 value was 0.993 in YOLOv8 architecture.

Keywords: *Deep Learning, Real-Time Detection, Traffic Signs, YOLO*

1. INTRODUCTION

As a result of increasing air and environmental pollution, approximately 90% of the world's population breathes polluted air. More than five million people die due to air and environmental pollution every year. Our fossil fuel-powered private cars, which are used extensively in our daily lives, also have a large share in this pollution. In addition to air pollution, exhaust gas emissions also cause climate change. When we look at today's energy consumption, fossil fuels are used as the primary source and to the greatest extent. However, the amount of fuel used in transportation and transportation accounts for an average of 14% of the greenhouse gases and 23% of the CO^2 emissions worldwide [1-3].

It is predicted that electric vehicles can help reduce pollution caused by fossil fuel vehicles. However, considering that autonomous vehicles will both reduce air pollution and contribute to traffic safety, it has also focused on this area.

The idea of giving vehicles autonomous features dates back to the 1920s. In 1939, as a result of the joint research and development activities of General Motors and the American Sarnoff Laboratory, the idea of autonomous vehicles was presented to the public for the first time at the New York World's Fair. The autonomous vehicle idea exhibited also included a highway system that would assist the vehicle. In 1958, testing of this type of vehicle was carried out for the first time, and a self-driving autonomous car was tested. Later, much work was done to develop autonomous buses, trucks, super smart vehicle systems, and driving video image processing [4].

The biggest problem for autonomous vehicles is the detection of traffic signs. Traffic signs must be

***Corresponding Author:** Abdil KARAKAN, akarakan@aku.edu.tr

identified so that vehicles can move or direct their movement. For this, traffic signs must be detected with high accuracy in all kinds of environments. Many methods are applied to detect traffic signs and markers. These detection structures have many variants with different data structures, training methods, and detection functions. One approach uses colour and shape, two important physical characteristics of traffic signs. Colour-based detection method uses blue, red, and yellow colours present in traffic signs and markers and intensity detection technique. With this method, the image is separated into colour channels. The largest area of these separated colour pixels is created to be connected to each other. Intensity information is obtained by converting the image into colour space. In shape-based detection, triangles, circles, quadrilaterals, and octagons are used as common standard shapes of traffic signs. In shape-based detection, detection is made by highlighting these features. For these features, border lines, structures and textures, and key points of standard shapes are used. The Hough transform is usually applied to detect these features. In its most basic sense, the Hough transform is used to define lines in the image. In recent years, significant developments have been experienced in machine learning, and machine learning-based detection methods have been developed. Machine learning methods such as AdaBoost, Support Vector Machine (SVM), and Neural Networks (NN) are used to detect traffic signs and markers. These methods have different training methods and different detection processes. In these detection processes, there are many input derivatives with different properties. AdaBoost and Cascade-based detection structures have been successful in many object identification problems, such as face recognition, vehicle recognition, and license plate identification. This structure has also been successfully applied to detect traffic signs and markers [5-8].

Ortataş used the Haar cascade algorithm to identify traffic signs and markers. Although there are examples of this structure in the literature for face recognition, pedestrian recognition, and license plate recognition, there is no application for plate recognition. In this context, firstly, a data set consisting of traffic signs and markers was prepared. Then, machine learning was performed on this data set with the Haarcascade algorithm. A simulation environment has been prepared to prove the accuracy of this learning. With the Vector Zero Roadrunner program, a road with traffic signs, and markers was defined on a three-dimensional track. Possible signs that a driver may encounter in traffic flow and which are learned with the Haarcascade structure have been placed in the simulation environment. This created track was added to the Unreal Engine platform, and an autonomous vehicle with a camera that can use this track was defined. Optimization studies were also carried out according to simulation outputs [9].

Convolutional Neural Network (CNN)-based detection methods learn features through a convolutional network. With the development of deep learning methods, CNN was used to detect traffic signs and markers. A CNN classifier is used to identify these objects. Most CNN-based detection networks are quite slow to detect objects. Among these networks, You Only Look One (YOLO) has very fast performance [10]. YOLO neural network basically starts by extracting a single image from a video. It resizes this image extracted in the second step. These images represent entry into the YOLO network. The YOLO network consists of layers. There are different structures in each layer. YOLO network takes an image as input. Then, it gets the coordinate information by drawing a boundary box. Evaluates the probability of there being an object inside the box. Finally, it checks whether the objects in the box belong to any class [11].

In the study conducted by Shao et al., the Faster R-CNN model was used for traffic sign detection. The recognition speed was increased by including Gabor wavelets in the detection process. They especially worked on the detection of small targets. For this, it made changes to the layers of the VGG16 architecture. It used Chinese and German traffic sign datasets as data sets. In experimental results, it achieved high detection, especially on small targets. As a result of the study, a high accuracy of 99.01% was achieved with the proposed method [12]. Zhang used YOLOv2 architecture for traffic sign detection. It used the Chinese traffic sign dataset as the data set. This data set contains 9176 samples from 25 different classifications. It divides the input entry images into grids to detect small traffic signs. As a result of the experimental study, it achieved an accuracy rate of 89.7% [13]. In their study, Lui and his colleagues worked on detecting small traffic signs. The data set used the Chinese traffic sign data set. As a result of their study, they achieved a 6% increase in performance in detecting small traffic signs [14]. Belghaouti and his colleagues used the Ghost-YOLO architecture, a lightweight model, to detect traffic signs. With this model, they focused on features for detecting small signals. In the experimental study, it was determined that mAP50 was 92.71%. It shows that the number of parameters and calculations of the study decreased to 91.4% and 50.29% of the original, respectively [15]. Tabernik and Skocaj used the MASK R-CNN algorithm in their study. In their studies, they worked on data sets. They have improved the overall performance of the datasets. He applied his studies to the detection of 200 traffic sign categories. With the proposed approach, they found an error rate below 3%, which is sufficient for distribution in practical applications of traffic sign inventory management [16]. Stallkamp and his colleagues used Convolutional Neural Networks to recognize traffic signs using CNN. In their study, they used a dataset containing images of more than 50,000 German road signs from 43 different classes. As a result of their experimental studies, a high classification accuracy of 99.46% was achieved [17]. Kim and Lee used the Consensus (RANSAC) algorithm and CNN networks for lane tracking. In their study, they achieved superior performance compared to other lane detection algorithms by combining RANSAC with CNN [18]. Qian used the CNN method for traffic sign detection to achieve high performance in terms of traffic sign detection speed and recognition accuracy. Chinese traffic signs and GTSDB dataset were used in their study. They worked on detecting 96 different traffic signs. As a result of their studies, an average accuracy rate of 97.56% was obtained [19]. Acros and his colleagues used the comparative analysis method to detect traffic signs. They used the German traffic signs dataset as the data set. They worked with different architectures. It achieved the highest accuracy of 95.77% in the Resnet V2 architecture, and 95.15% in the Resnet 101 architecture. The difficulties in detecting small traffic signs were noted in their models and it was determined that MobileNet-based models were suitable for mobile and embedded devices [20]. Hussian used Fast Branch architecture from CNN models to recognize traffic signs. They achieved a high accuracy rate of 98.52% in their study [21].

When the literature studies are examined, most of them use ready-made data sets. Experimental studies give good results in studies using ready-made data sets. When experimental results are applied in real life, traffic signs are either not detected or detected very little. The biggest reason for this is that in real life, traffic signs are located in different locations, at different lighting levels and the surrounding objects make it difficult to detect the event. In the study, photographs of real traffic signs were taken with the camera placed in the vehicle. This was done from different positions, different distances, different lighting conditions, and levels to ensure adaptation to the real environment. The resulting data set was used in 3 different algorithms in the YOLO5 architecture. As a result of the study, the highest accuracy was 98.80% in the YOLOV5 architecture, F1-Score was 0.950, and mAP@.5 value was 0.981.

2. MATERIAL AND METHODS

Real-time detection of traffic lights with the YOLO algorithm was carried out in three steps. In the first step, sub-processes related to the preparation of the data set were carried out. These transactions; include data preparation, pre-processing, training data, and test data processes. The second step is to create and adjust the convolutional neural network to be used for deep learning. These transactions; Creating a deep learning neural network model, adjusting deep learning neural network parameters, and uploading training and test data to the system. The last step is to run the system. In this step; training the model, retraining the system based on the results, and finally getting the results. Figure 1 shows the general working principle of real-time detection of traffic lights with the YOLO algorithm.

2.1. Data Set

Traffic signs differ in shape and size. At the same time, the places where traffic signs are placed are different. Especially the surroundings of the places where traffic signs are used are very different. For these reasons, it is difficult to detect traffic signs. To overcome these difficulties, the photographs that make up the data set must be very similar to the real environment. To achieve this, the data set was created from images obtained with a camera placed in the vehicle. In this way, images very close to the real environment were obtained.

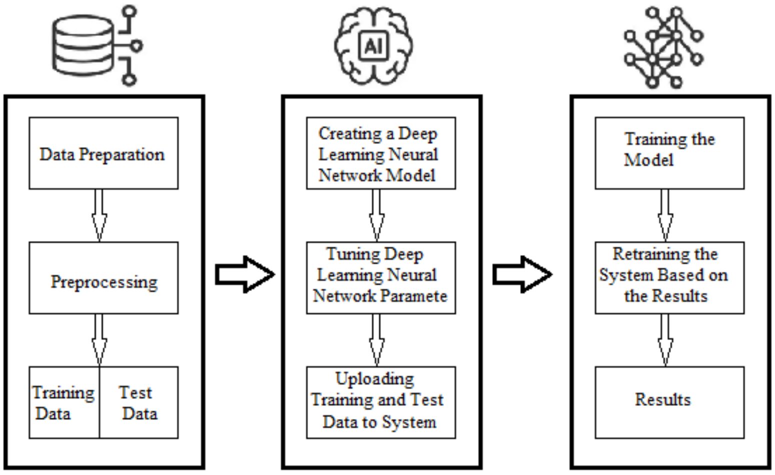

Figure 1. General working principle of real-time detection of traffic lights with the YOLO algorithm.

Some issues were taken into consideration when creating the data set. The first of these is the height of traffic signs. Not all traffic signs are equal in height. For this purpose, photographs of traffic signs at different heights were taken and added to the data set. Traffic signs are not always on one side of the road. For this purpose, photographs of the traffic signs on the right and left of the road were taken. Vehicles move at all hours of the day. In this case, traffic signs are encountered at different illumination levels. To minimize error, photographs of traffic signs in different lighting conditions and levels were added to the data set. Traffic lights do not always approach at the same distance. Especially at traffic lights on winding roads, the approach distance can be very short. To prevent this, images of traffic signs were taken from different distances and added to the data set. Figure 2 shows 9 different examples that make up the data set.

Figure 2. Some sample photos from the dataset

The weather is not always nice. Sometimes there is wind, rain and snow. In the study, photographs of traffic signs taken in different weather conditions were added. In this way, the study was adapted to real life. Figure 3 shows traffic signs taken in three different weather conditions.

Figure 3. Traffic signs taken in three different weather conditions.

25 of them were used in traffic signs in the data set. Table 1 shows the names and sampling numbers of the traffic signs used in the data set.

Table 1. The names and sampling numbers of the traffic signs used in the data set are shown.

Traffic Sign	Number of Samples	Traffic Sign	Number of Samples
Traffic Lamp	305	50km/h	400
Stop	700	60 Km/h	105
Park	251	70 Km/h	500
Left Turn Prohibited	378	80 Km/h	435
Turn Right	450	Main Road	409
No entry	378	Risk of Ice	201
Parking Prohibited	402	No Turn Right	178
Road Closed to Traffic	301	No Turn Left	202
Forced Direction Forward Left	205	Stop Giving Way	329
Forward Right Forced Direction	322	No Parking	394
Red light	708	School Crossing	217
Green Light	725	Level Crossing with Gate	221
30 Km/h	215		

Many changes to the images that make up the dataset. With these changes, two important features were added to the data set. Firstly, the data set must be very large for real-time detection. With these operations performed on the data set, the size of the data set was increased. A second important feature minimizes errors that may occur from cameras in real-time detection. In real-time detection, cameras do not always capture proper images. The distortions that these images will be exposed to make the detection process difficult. With this process, the photographs in the data set are introduced to the model in advance. Thus, higher efficiency is achieved in real-time detection operations.

With these changes made from the photographs that make up the data set, the error rate in real-time traffic sign detection has been minimized. A camera is placed inside the vehicle for real-time detection. The detection process was carried out using the images taken from this camera. Distortions occur in the photos taken due to camera and vehicle movement. The changes made to the data set and the errors that occurred during the detection process were downloaded.

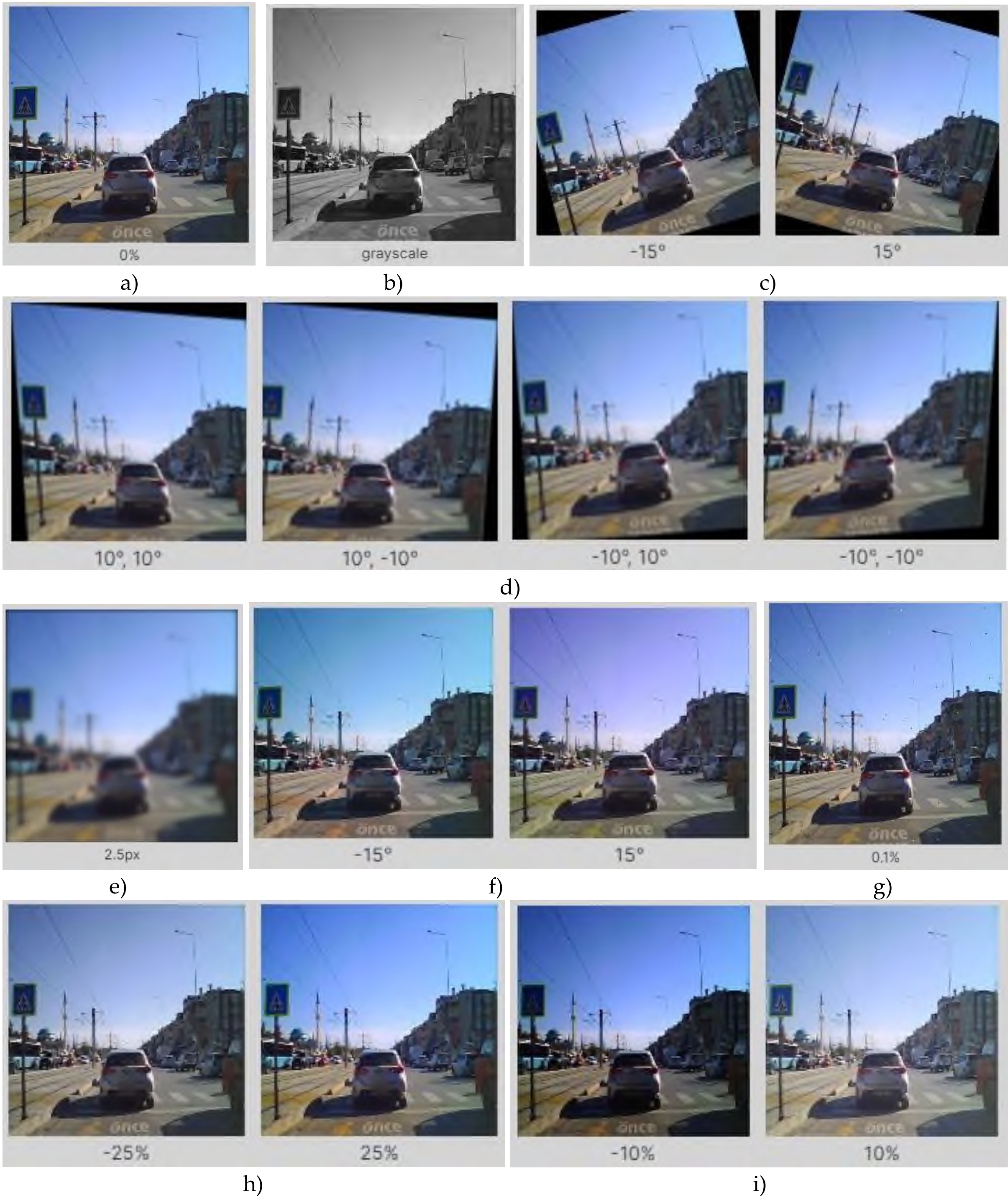

Figure 4 a) Sizing and position change, **b)**Grayscale, **c)**Adding slope, **d)**Blurring, **e)**Adding variability, **f)** Adding noise, **g)**Image brightness change, **h)**Colour vividness change, **i)**Perspective change.

2.2. YOLO Algorithm

YOLO architecture was introduced by Joseph Redmon in 2015. YOLO is used for real-time object detection. This algorithm detects box-shaped objects in the image with forward propagation. The YOLO algorithm operates extremely quickly. It sees the entire image during the training and testing of the data set. Thus, it encodes information about classes and their views. YOLO learns generalizable representations of objects. Thus, the algorithm performs better when trained and tested on natural images. Figure 5 shows the YOLO architecture.

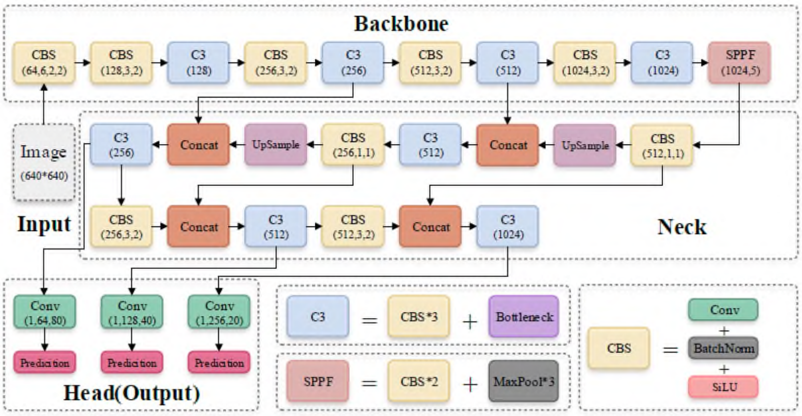

Figure 5. YOLO architecture [22].

2.2.1 YOLOv5

YOLOv5 is a model in the You Only Look Only (YOLO) family of computer vision models. YOLOv5 is widely used to detect objects. YOLOv5 comes in four main versions: small (s), medium (m), large (l), and extra-large (x), each offering increasingly higher accuracy rates. Each variant also takes a different amount of time to train. In the graph, the goal is to create a very performance object detection model (Y-axis) based on extraction time (X-axis). Preliminary results show that YOLOv5 performs very well compared to other state-of-the-art techniques for this purpose. Figure 6 shows the features of four different YOLOV5 architectures.

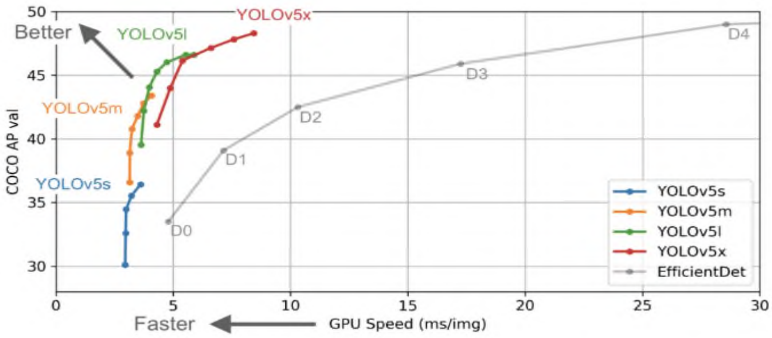

Figure 6. YOLOv5 architecture [23].

There are four algorithms in the YOLOv5 architecture: YOLOv5x, YOLOv5l, YOLOv5m, and YOLOv5s. The highest accuracy rate in these algorithms is realized in the YOLOv5x algorithm. The lowest accuracy rate is in the YOLOv5s algorithm. There is a contrast between accuracy rates and detection times. In the study conducted, both the high accuracy rate and the detection speed should be very low.

2.2.2 YOLOv6

The main goal of YOLOv6 is to provide superior performance, especially in real-time applications, by operating at high speed and low latency. The architecture of the model includes various improvements and optimizations to achieve better overall accuracy. These improvements include optimized convolutional layers for more effective feature extraction, new activation functions for better overall performance, and advanced data augmentation techniques. YOLOv6 has also managed to create lighter structures by reducing model sizes, thus becoming a system that can run efficiently even on low-power hardware. This model is quite successful in detecting objects of different sizes and scenes with various levels of complexity. The basis of this success is the large datasets and optimized loss functions used in the training process of the model. With its multi-scale detection capabilities, YOLOv6 can effectively detect both small and large objects, and thus can be used in a wide range of applications. For example, it can be used effectively in tasks such as pedestrian detection, other vehicles, and traffic signs in autonomous vehicles, in recognizing suspicious objects or people in security systems, and in tasks such as shelf tracking and inventory management in the retail sector. In addition, thanks to its versions optimized for mobile devices, fast and accurate object detection is also possible on portable devices. YOLOv6 is open to continuous development and optimization thanks to its open source and wide community support. At the same time, it can be adapted to different needs by offering a customizable structure for different datasets and application scenarios. The high accuracy, low latency, and optimized performance offered by YOLOv6 are important models that make it stand out in today's computer vision applications that require speed and accuracy. Figure 7 shows the features of four different YOLOv6 architectures.

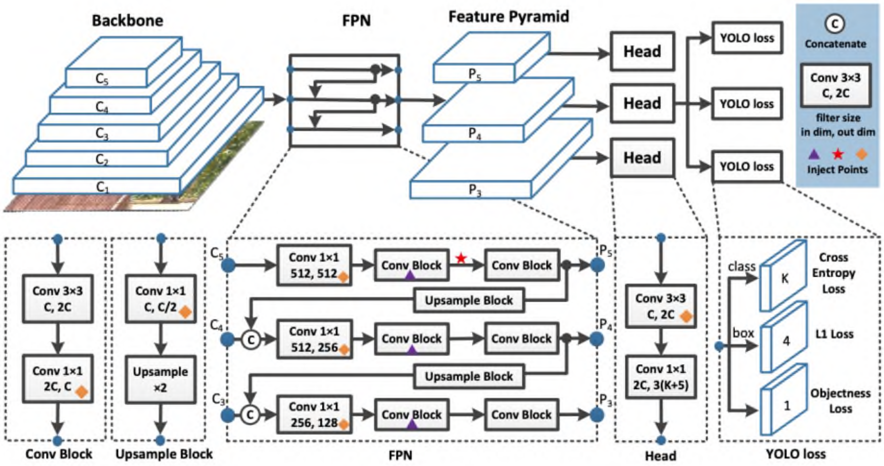

Figure 6. YOLOv6 architecture [24].

2.2.3 YOLOv7

YOLOv7 has made significant progress in the field of object detection, improving the strengths of previous versions while also introducing innovations that increase performance and efficiency. Unlike previous versions, YOLOv7 is equipped with optimized architectural changes and new training techniques. This provides superior performance in terms of both accuracy and speed. One of the most significant improvements of the model is its ability to strike a perfect balance between computational efficiency and detection accuracy. This feature makes YOLOv7 ideal for both high-performance servers and devices with limited hardware resources. The model uses advanced model scaling strategies that offer flexible scalability depending on the hardware. In this way, it can run on different platforms without losing quality. In addition, thanks to its improved feature extraction mechanisms and attention modules, it can detect objects of different sizes and complexity with high accuracy even in challenging environments. One of the most striking features of YOLOv7 is its high performance in real-time applications. The fact that it does not compromise on accuracy while maintaining high frame rates provides a great advantage, especially in areas where fast and precise object recognition is critical, such as autonomous driving. In addition, it can be effectively used in various fields such as security systems, robotic applications, and augmented reality. YOLOv7 integrates new techniques for better generalization on different datasets, providing consistent performance despite the variability in the input data. The flexibility of the model allows it to adapt to a wide range of deployment environments, from cloud-based infrastructures to endpoint devices such as mobile devices and embedded systems. The open-source nature of YOLOv7 provides a foundation for continuous development and community-supported improvements. The model supports advanced training optimizations such as dynamic label assignment and adaptive loss functions, resulting in faster training times and higher overall performance. Figure 8 shows the features of four different YOLOv7 architectures.

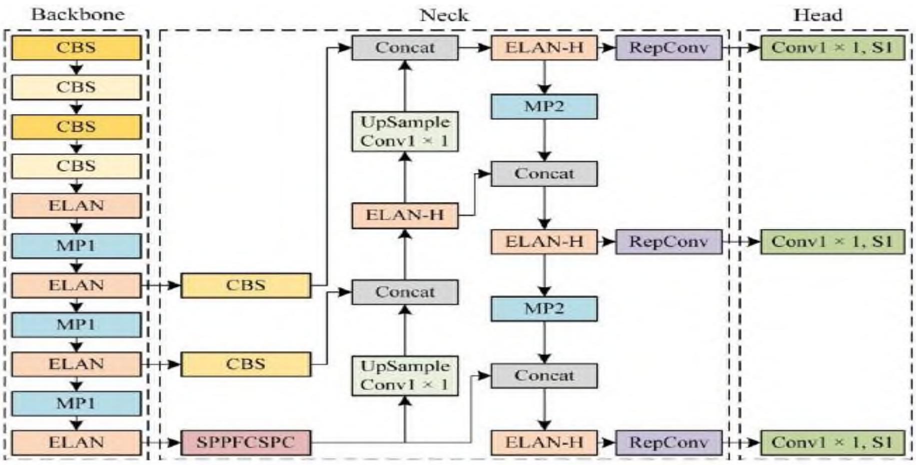

Figure 8. YOLOv7 architecture [25].

2.2.4 YOLOv8

YOLOv8 presents a lighter and more powerful structure thanks to the radical changes made in terms of architecture and advanced training techniques. YOLOv8 draws attention with the new layer arrangements and parameter optimizations developed to increase the efficiency of the model. These improvements allow faster and more accurate results to be obtained, while providing high performance even on low-power devices. The model's attention mechanisms and advanced feature extraction layers help it detect objects of different sizes and difficulty levels more precisely. In this way, YOLOv8 can detect and classify a large number of objects with high accuracy even in complex scenes. Another important innovation is the versatility of YOLOv8. The model is designed to be used in different computer vision tasks, such as not only object detection but also image segmentation and pose estimation. This makes it usable in a wide range of applications, from autonomous driving to security systems, from the healthcare sector to robotics. YOLOv8's improved training processes are supported by techniques such as dynamic label assignment and adaptive loss functions. This enabled the model to reach higher accuracy rates in a shorter time. YOLOv8's modular structure and flexible architecture allow users to customize it according to their own datasets and specific needs. YOLOv8's comprehensive documentation and user-friendly interface provide an easy integration process for both beginners and advanced users. The model's high accuracy, low latency, and optimized computational efficiency make it one of the most advanced solutions in object detection and computer vision. Figure 9 shows the features of four different YOLOv8 architectures.

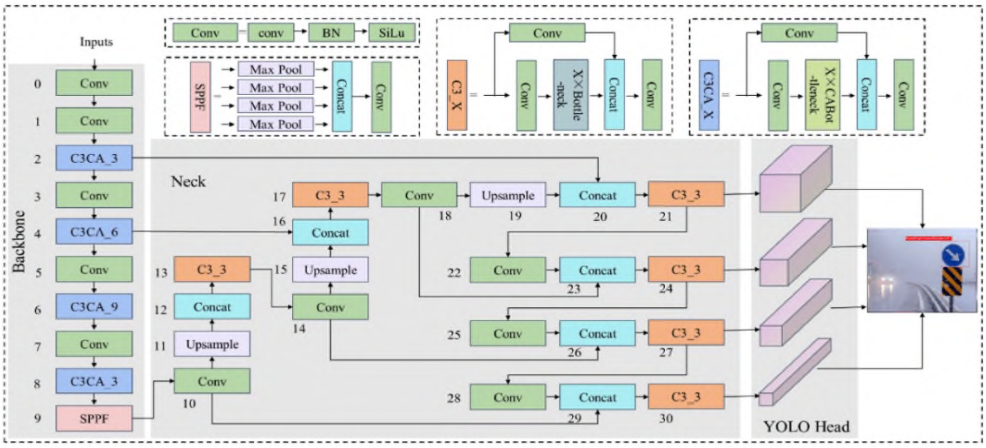

Figure 9. YOLOv8 architecture [26].

3. RESULTS AND DISCUSSION

In the study, traffic signs were detected in real time. For this purpose, a camera is placed inside the vehicle. Detection is carried out using images taken from the in-car camera. Figure 10 shows the working algorithm prepared for the operation of the system.

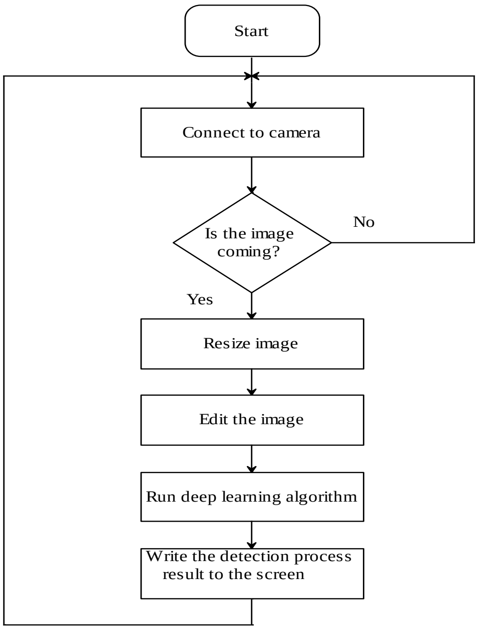

Figure 10. Operating algorithm of the system

The study was used in YOLOv5x, YOLOv6, YOLOv7, and YOLOv8 models. The same data set was used in each architecture. In this way, errors that may occur due to the data set are minimized. In the YOLOv5x model, the confusion matrix achieved a success rate of 97.20%. In the study, F1-Score was 0.98. Figure 11 shows the results after training on the YOLOv5x architecture.

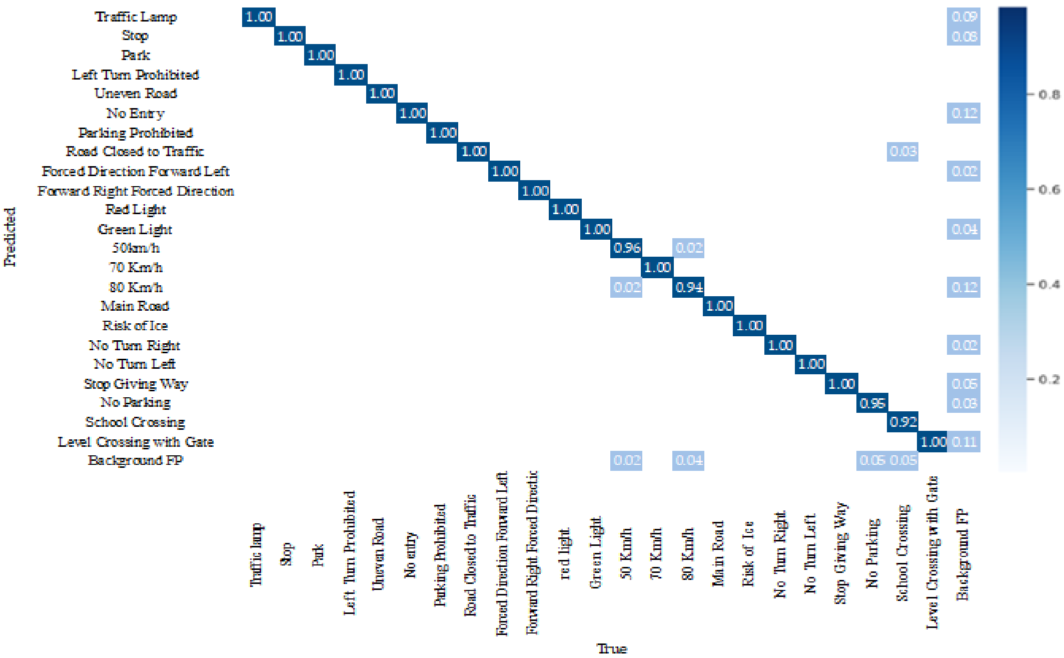

Figure 11. Results after training on YOLOv5x architecture.

The YOLOv6 model has a less complex structure. In the YOLOv6 model, the confusion matrix achieved a success rate of 99.1%. In the study, F1-Score was 0.95. Figure 12 shows the results after training on the YOLOv6 architecture.

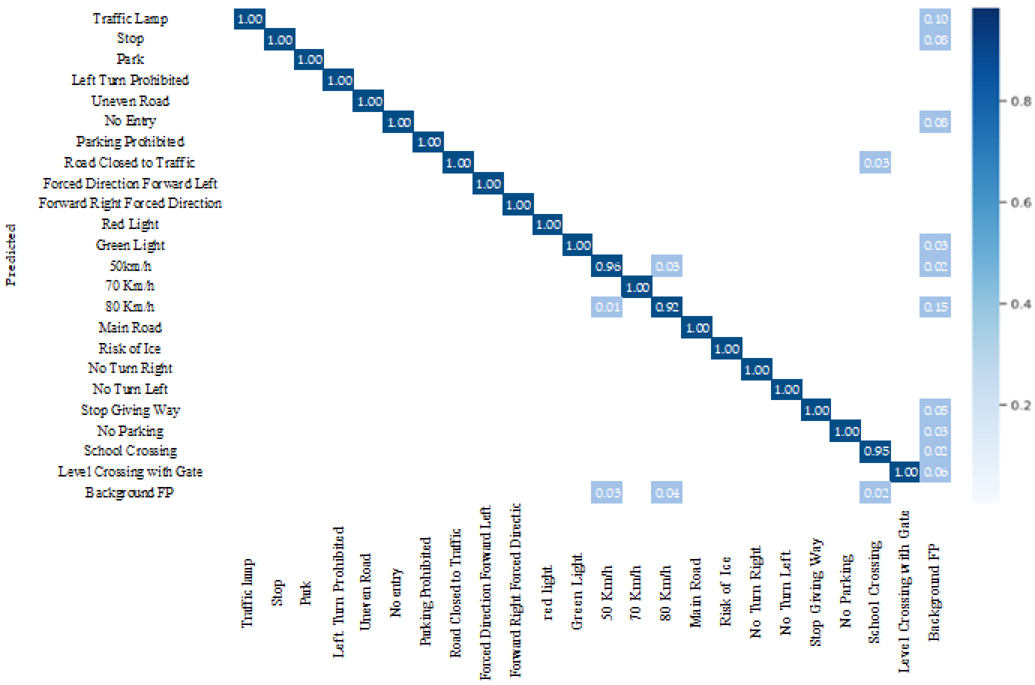

Figure 12. Results after training on YOLOv6 architecture.

In the YOLOv7 model, the confusion matrix achieved a success rate of 99.50%. In the study, F1-Score was 0.99. Figure 13 shows the results after training on the YOLOv7 architecture.

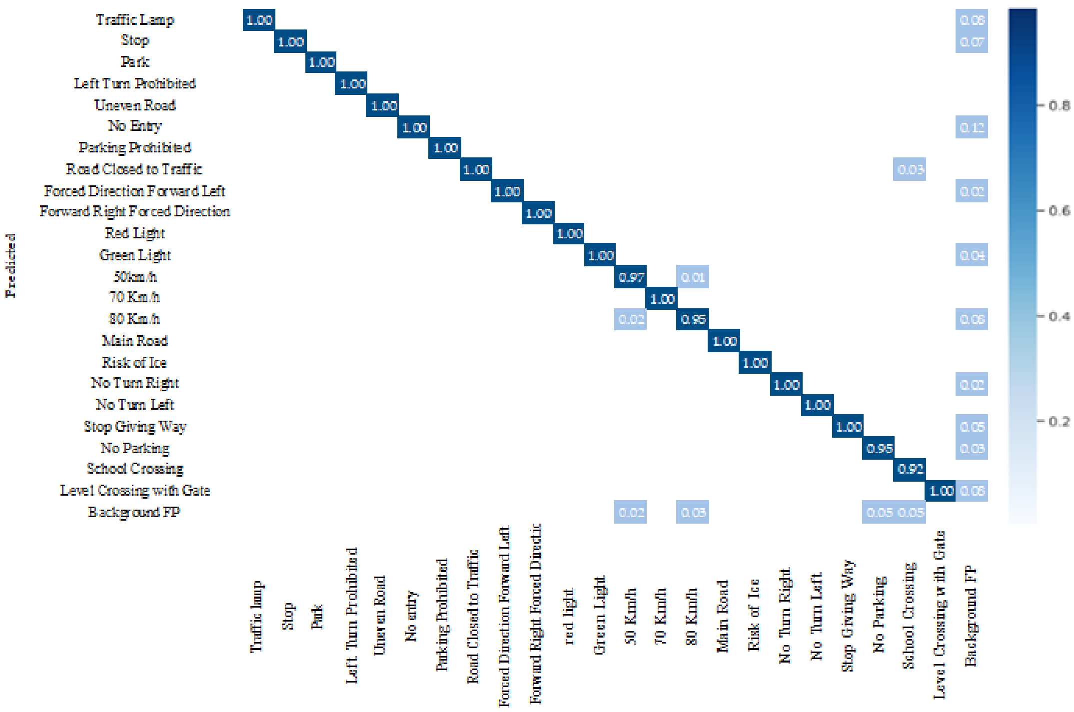

Figure 13. Results after training on YOLOv7 architecture.

In the YOLOv8 model, the confusion matrix achieved a success rate of 99.60%. In the study, F1-Score was 0.99. Figure 14 shows the results after training on the YOLOv8 architecture.

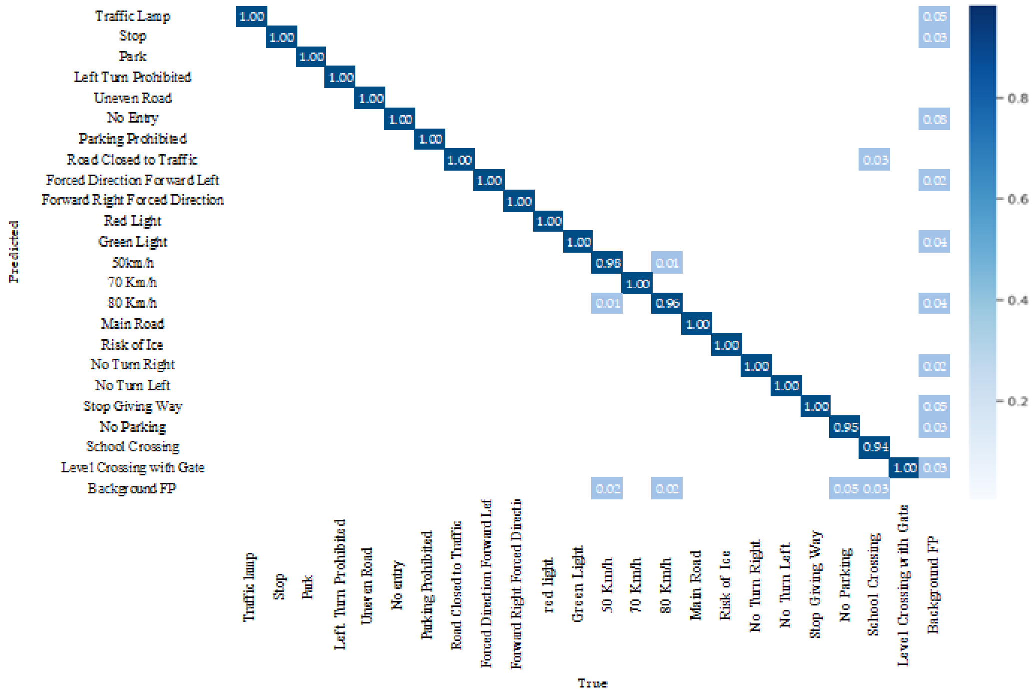

Figure 14. Results after training on YOLOv8 architecture.

In the study, 25 traffic signs were identified. For this, YOLOv5m, YOLOv5s, YOLOv5x, YOLOv6, YOLOv7, and YOLOv8 architectures were used. The internal structure of each architecture is different. For this reason, different results occur. Table 2 shows the average values of the YOLOv5m, YOLOv5s, YOLOv5x, YOLOv6, YOLOv7, and YOLOv8 models.

Table 2. Average values of YOLOv5m, YOLOv5s and YOLOv5x models shown.

Architectural Name	Precision	Recall	mAP50
YOLOv5m	0.985	0.926	0.974
YOLOv5s	0.988	0.945	0.981
YOLOv5x	0.994	0.939	0.969
YOLOv6	0.991	0.951	0.985
YOLOv7	0.995	0.955	0.991
YOLOv8	0.996	0.962	0.993

When the training results of YOLOv5m, YOLOv5s, YOLOv5x, YOLOv6, YOLOv7 and YOLOv8 models are examined, the model with the highest accuracy rate is the YOLOv8 model. The model with the lowest accuracy rate is the YOLOv5m model with 98.50%. When the mAP50 results of the models are examined, the highest YOLOv8 model is calculated as 0.993 and the lowest YOLOv5x model is calculated as 0.969.

In real-time traffic sign detection, the selection is not made only based on the highest accuracy rate. The frames per second (FPS) of the architecture to be selected, the weight file size of the architecture, training, and detection time were taken into consideration. It was chosen among six different architectures, taking into account the results of YOLO. YOLOv8's architecture showed better performance in real-time traffic sign detection.

Raspberry Pi4B 4GB and Raspberry Pi4B 8GB models were used in the study. In the study, different sizes such as 1024 × 1024, 800 × 600, 600 × 360, 416 × 416 and 320 × 320 were used. The best performance was at 320 × 320. In the study conducted with Raspberry Pi4B 4GB and photo size 320 × 320, a speed of 0.6 FPS was achieved. In the study conducted with Raspberry Pi4B 8GB and photo size 320 × 320, a speed of

2,5 FPS was achieved.

Real-time traffic sign detection was tested at 4 different speeds. Testing was carried out at speeds of 30 km/h, 40 km/h, 50 km/h and 60 km/h, taking into account the city speed limits. Table 3 shows the number of traffic signs and signals and different speed limits read by the model from a moving vehicle during the testing process.

Table 3. The rate of detecting traffic signs of the system at different speed

	60 km/h		50 km/h		40 km/h		30 km/h	
Traffic Sign	Sign	Read Sign	Sign	Read Sign	Sign	Read Sign	Sign	Read Sign
Stop	60	51	60	53	60	54	60	55
Green Light	57	50	57	51	57	53	57	53
Red Light	50	45	50	45	50	46	50	47
Crosswalk	35	31	35	31	35	33	35	33
Turn Right	33	29	33	30	33	30	33	31
Total Sign	235	206	235	210	235	216	235	219
Total Rate	% 87.65		% 89.36		% 91.91		% 93.19	

An accuracy of 87.65% was achieved in real-time traffic sign detection at a vehicle speed of 60 km/h. The second highest accuracy was achieved at a speed of 50 km/h with 89.36% and then at a speed of 40 km/h with an accuracy rate of 91.91%. The best result was achieved with an accuracy rate of 93.19% at a speed of 30 km/h. The main reasons for this are due to the processing speeds of the camera and graphics card used.

When we look at the literature studies, many studies have been done on traffic signs. Ready-made data sets were used in most of their studies. High accuracy rates were achieved in these studies. However, in real-time detection applications, traffic signs were detected at a very low rate or not at all. Table 4 shows literature studies.

Table 4. Literature studies.

Authors	**Data Set**	**Architectural**	**Precision**
Zhu et al. [27]	Custom	YOLOv5	0.97
Wan et al. [28]	GTSDB	TSYOLO	0.83
Lu et al. [29]	Custom	YOLOV5	0.88
Yi et al. [30]	TT100K	YOLOv4	0.92
Jiang et al. [31]	TT100K	YOLOv5	0.87
Aggar et al. [32]	IQTSDB	YOLOv5	0.90
Çınarel et al. [33]	GTDRB	YOLOv5	0.99
Azimjanov et al. [34]	own data set	YOLO	0.95
Rodriguez et al. [35]	own data set	YOLOv3	0.95
Ren et al. [36]	TT100K	YOLOv7	0.92
Hsieh et al. [37]	OCR	YOLOv7	0.97
Kuppusamy et al. [38]	GTSRB	YOLOv7	0,98
Yu et al. [39]	KITTI	YOLOv7-Tiny	0.96
Zhao et al. [40]	TT100K	YOLO-MAM	0.84
Ji et al. [41]	TT100K	YOLOv8	0.98
Zhang et al. [42]	TT100K	TST-DETR	0.99
Yaamini et al. [43]	own data set	YOLOv8	0.90
Han et al. [44]	GTSRB	EDN-YOLO	0.93
Mercoldo et al. [45]	own data set	YOLO	0.97
Sun et al. [46]	TT100K	ERF-YOLO	0.84

The majority of literature studies have been conducted experimentally. It has not been implemented in a real environment. The study was carried out both experimentally and practically. A very high level of accuracy has been achieved in experimental studies. Such high accuracy has not been achieved in real applications. Detection studies were carried out at different speeds in real applications. Thus, the accuracy rates of the study at different speeds were compared. Accuracy rates were high at low speeds. As speed increased, accuracy rates decreased. There are two important reasons for this. Firstly; The frame per second rate (FPS) of the work is not high enough. The higher the FPS speed, the higher the detection rate of traffic signs at high speeds. Since the FPS rate is sufficient at low speeds, the detection rate has increased to high levels. The second important reason is the speed of the computer used. Raspberry Pi4B 8GB was used in the study. Increasing the graphics card and processor speed of the computer used will affect the accuracy rate in the studies. Figure 11 shows the traffic signs detected in the study.

Figure 11. Traffic signs identified in the study.

4. CONCLUSIONS

In the study, real-time detection of traffic signs was achieved. A camera is placed inside the vehicle for this process. The data set was prepared with images taken from the real environment. Grayscale, tilt, blurring, variability, noise, image brightness change, colour vividness change, perspective change, sizing, and position change were made on the photographs that make up the data set, to ensure their harmony with the real environment. The dataset was used in 3 different versions of the YOLOv5 architecture and YOLOv6, YOLOv7 and YOLOv8 architectures. As a result of the study, 99.60% accuracy was achieved in the YOLOv8 architecture. The study was implemented in a real environment. For the application, the car was detected at speeds of 60, 50, 40, and 30 km/h. An accuracy of 87.65% was achieved in real-time traffic sign detection at a vehicle speed of 60 km/h. The second highest accuracy was achieved at a speed of 50 km/h with 89.36% and then at a speed of 40 km/h with an accuracy rate of 91.91%. The best result was achieved with an accuracy rate of 93.19% at a speed of 30 km/h. The main reasons for this are due to the processing speeds of the camera and graphics card used.

Declaration of Ethical Standards

As the author of this study, he declares that he complies with all ethical standards.

Declaration of Competing Interest

The authors declared that they have no conflict of interest.

Funding / Acknowledgements

No funding is available for this research.

Data Availability

Data available on request from the author.

REFERENCES

[1] J. Webb, C. Wilson, and T. Kularatne, "Will people accept shared autonomous electric vehicles. A survey before and after receipt of the costs and benefits," *Economic Analysis and Policy*, vol. 61, pp. 118–135, 2019. https://doi.org/10.1016/j.eap.2018.12.004.

[2] A. Moorthy, R. De Kleine, G. Keoleian, J. Good, and G. Lewis, "Shared autonomous vehicles as a sustainable solution to the last mile problem: A case study of Ann Arbor-Detroit area," *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, vol. 10(2), 2017. https://doi.org/10.4271/2017-01-1276.

[3] A. Faisal, T. Yigitcanlar, G. Currie, S. Journal, and L. Use, "Understanding autonomous vehicles Linked references are available on JSTOR for this article: Understanding autonomous vehicles: A systematic literature re- view on capability, impact, planning and policy," *Journal of Transport and Land Use*, vol. 12(1), pp. 45–72, 2019. https://doi.org/10.5198/jtlu.2019.1405

[4] H. Fleyeh, "Color detection and segmentation for road and traffic signs," *IEEE Conference on Cybernetics and Intelligent Systems,* vol. 2, pp. 809-814, 2004. https://doi.org/0000-0002-1429-2345.

[5] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in realworld images: The German traffic sign detection benchmark," *The 2013 International Joint Conference on Neural Networks,* pp. 1-8, 2013. https://doi.org/10.3390/app11073061.

[6] J. Liu, and X. Wang, "Tomato diseases and pests detection based on improved Yolov3 convolutional neural network," *Frontiers in Plant Science*, vol. 11, pp. 898, 2020. https://doi.org/10.3389/fpls.2020.00898.

[7] S. Tan, G. Lu, Z. Jiang, and L. Huang, "Improved Yolov5 network model and application in safety helmet detection," *2021 IEEE International Conference on Intelligence and Safety for Robotics*, pp. 330-333, 2021. https://doi.org/10.1109/ISR50024.2021.9419561

[8] F. Zhou, H. Zhao, and Z. Nie, "Safety helmet detection based on Yolov5," *IEEE International Conference on Power Electronics Computer Applications*, pp. 6-11, 2021. https://doi.org/10.3390/s22249843.

[9] F. N. Ortataş, and E. Çetin, "Traffic Sign Recognition and the Application of Simulation Using Machine Learning in Electric and Autonomous Vehicles," *El-Cezerî Journal of Science and Engineering*, vol. 8 (3), pp. 1081-1092, 2021. https://doi.org/10.31202/ecjse.867733.

[10] F. Mehdi, and G. Sedigheh, "Traffic Road Sign and Classification," *Majlesi Journal of Electrical Engineering*, vol. 4 (23), pp. 54- 62, 2021.

[11] L. Zeng, B. Sun, and D. Zhu, "Underwater target detection based on Faster R-CNN and adversarial occlusion network," *Engineering Applications of Artificial Intelligence*, vol. 100, pp. 104190, 2021. https://doi.org/10.1016/j.engappai.2021.104190

[12] F. Shao, X. Wang, F. Meng, J. Zhu, D. Wang, and J. Dai, "Improved faster R-CNN traffic sign detection based on a second region of interest and highly possible regions proposal network," *Sensors*, vol. 19 (10), pp. 2288, 2019. https://doi.org/10.3390/s19102288.

[13] J. Zhang, M. Huang, X. Jin, and X. Li, "A real-time Chinese traffic sign detection algorithm based on modified Yolov2," *Algorithms*, vol. 10 (4), pp. 127, 2017. https://doi.org/10.3390/a10040127.

[14] C. Liu, S. Li, F. Chang, and Y. Wang, "Machine Vision Based Traffic Sign Detection Methods: Review," Analyses and Perspectives, *IEEE Access*, vol. 7(1), pp. 86578-86596, 2019. https://doi.org/10.1109/Access.2019.2924947

[15] O. Belghaouti, W. Handouzi, and M. Tabaa, "Improved traffic sign recognition using deep ConvNet architecture," *Procedia Computer Science*, vol. 177, pp. 468–473, 2020. https://doi.org/10.1016/j.procs.2020.10.064

[16] D. Tabernik, and D. Skočaj, "Deep learning for large-scale traffic-sign detection and recognition," *IEEE transactions on intelligent transportation systems*, vol. 21 (4), pp. 1427-1440, 2019. https://doi.org/10.1109/TITS.2019.2913588

[17] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural networks*, vol. 32, pp. 323-332, 2022. https://doi.org/10.1016/j.neunet.2012.02.016.

[18] J. Kim, and M. Lee, "Robust Lane Detection Based On Convolutional Neural Network and Random Sample Consensus," *Neural Information Processing*. 2014. https://doi.org/10.1007/978-3-319-12637-1_3.

[19] R. Qian, B. Zhang, Y. Yue, Z. Wang, an F. Coenen, "Robust Chinese traffic sign detection and recognition with deep convolutional neural network," *11th International Conference on Natural Computation*, pp. 791-796, 2015. https://doi.org/10.1109/ICNC.2015.7378104.

[20] A. Arcos-Garcia, J. A. Alvarez-Garcia, and L. M. Soria-Morillo, "Evaluation of deep neural networks for traffic sign detection systems," *Neurocomputing*, vol. 316, pp. 332-344, 2018. https://doi.org/10.1016/j.neucom.2018.07.072.

[21] S. Hussain, M. Abualkibash, and S. Tout, "A survey of traffic sign recognition systems based on convolutional neural networks," *IEEE International Conference on Electro/Information Technology*, pp. 0570-0573, 2018. https://doi.org/10.1109/EIT.2018.8399772.

[22] M. E. I. Malaainine, H. Lechgar, H. Rhinane, "YOLOv2 Deep Learning Model and GIS Based Algorithms for Vehicle Tracking," *Journal of Geographic Information System*, vol. 13, pp. 395-409, 2021. https://doi.org/0.4236/jgis.2021.134022.

[23] D. Dluznevskij, P. Stefanovic, S. Ramanauskaite, "Investigation of YOLOv5 Efficiency in iPhone Supported Systems," *Baltic J. Modern Computing*, vol. 9, pp. 333–344, 2021. https://doi.org/10.22364/bjmc.2021.9.3.07

[24] B. R. Chang, H. Tsai, C. Hsieh, "Accelerating the Response of Self-Driving Control by Using Rapid Object Detection and Steering Angle Prediction," *Electronics*, vol. 12, pp. 2161, 2023. https://doi.org/10.3390/electronics12102161

[25] H. Herfandi, O. S. Sitanggang, M. R. A. Nasution, H. N, Y. M. Jang, "Real-Time Patient Indoor Health Monitoring and Location Tracking with Optical Camera Communications on the Internet of Medical Things," *Appl. Sci.* vol. 14, pp. 1153, 2024. https://doi.org/10.3390/app14031153

[26] B. Xing, W. Wang, J. Qian, C. Pan, Q. Le,"A Lightweight Model for Real-Time Monitoring of Ships," *Electronics*, vol. 12, pp. 3804, 2023. https://doi.org/10.3390/electronics12183804.

[27] Y. Zhu, and W. Yan, "Traffic sign recognition based on deep learning," *Multimedia Tools and Applications*, vol. 81(13), pp. 17779-17791, 2022. https://doi.org/10.1007/s11042-022-12163-0

[28] H. Wan, L. Gao, M. Su, Q. You, H. Qu, and Q. Sun, "A novel neural network model for traffic sign detection and recognition under extreme conditions," *Journal of Sensors*, 2021. https://doi.org/10.1155/2021/9984787.

[29] E. Lu, H. C. M. Gozdzikiewicz, K. H. Chang, and J. M. Ciou, "A hierarchical approach for traffic sign recognition based on shape detection and image classification," *Sensors*, vol. 22 (13), pp. 4768, 2022. https://doi.org/10.3390/s22134768.

[30] L. Yi, L. Jinguo, Z. Yongjie, and M. Ping, "Detection of self-explosive insulators in aerial images based on improved Yolov4," *In Journal of Physics: Conference Series*, vol. 2320 (1), pp. 012025, 2022. https://doi.org/10.1088/1742-6596/2320/1/012025

[31] L. Jiang, H. Liu, H. Zhu, and G. Zhang, "Improved Yolov5 with balanced feature pyramid and attention module for traffic sign detection," *MATEC Web of Conferences*, pp. 355, 2022. https://doi.org/10.1051/matecconf/202235503023

[32] A. Aggar, M. Zaiter, "Iraqi traffic signs detection based on Yolov5," *International Conference on Advanced Computer Applications,* pp. 5-9, 2021. https://doi.org/10.2991/978-94-6463-034-3_72

[33] G. Çınarer, "Deep Learning Based Traffic Sign Recognition Using YOLO Algorithm," D*üzce University Journal of Science & Technology*, vol. 12, pp. 219-229, 2022. https://doi.org/10.29130/dubited.1214901.

[34] A. Jahongir, and Ö. A. Ahmet, "A real-time vehicle detection and a novel vehicle tracking systems for estimating and monitoring traffic flow on highways," *Advanced Engineering Informatics*, vol. 50, pp. 101393, 2021. https://doi.org/10.1016/j.aei.2021.101393 ,

[35] R. B. Rodriguez, C. M. Carlos, O. O. V. Villegas, V. G. C. Sanchez, and H. J. O. Dominguez, "Mexican traffic sign detection and classification using deep learning," *Expert Systems With Applications,* vol. 202, pp. 117247, 2022. https://doi.org/10.1016/j.eswa.2022.117247.

[36] B. Ren, J. Zhang, T. Wang "A Hybrid Feature Fusion Traffic Sign Detection Algorithm Based on YOLOv7," *Computer, Materials and Continua,* vol. 80, pp. 1425-1440, 2024. https://doi.org/10.32604/cmc.2024.052667

[37] C. Hsieh, C. Hsu, W. Huang, "A two-stage road design detection and text recognition system based on YOLOv7," *Internet of Things*, vol. 27, pp. 101330, 2024. https://doi.org/10.1016/j.iot.2024.101330

[38] P. Kappusamy, M. Sanjay, P. V. Deepshree, C. Iwendi, "Traffic Sign Reorganization for Autonomous Vehicle Using Optimized YOLOv7 and Convolutional Block Attention Module," *Computer, Materials and Continua,* vol. 77, pp. 445-466, 2023. https://doi.org/10.32604/cmc.2023.042675

[39] Q. Yu, X. Xu, P. Xia, S. Xu, H. Wang, A. Rodic, P. B. Petrovic, "YOLOv7-Tiny road target detection algorithm based on attention mechanism," *Procedia Computer Science,* vol. 250, pp. 95-110, 2024. https://doi.org/10.1016/j.procs.2024.11.014

[40] R. Zhao, S. H. Tang, J. Shen, E. E. B. Supeni, S. A. Rahim "Enhancing autonomous driving safety: A robust traffic sign detection and recognition model TSD-YOLO," *Signal Processing,* vol. 225, pp. 109619, 2024. https://doi.org/10.1016/j.sigpro.2024.109619

[41] B. Ji, J. Xu, Y. Liu, P. Fan, M. Wang, "Improved YOLOv8 for small traffic sign detection under complex environmental conditions," *Franklin Open,* vol. 8, pp. 100167, 2024. https://doi.org/10.1016/j.fraope.2024.100167

[42] L. Zhang, K. Yang, Y. Han, J. Li, W. Wei, H. Tan, P. Yu, K. Zhang, X. Yang, "TSD-EDR: A lightweight real-time detection transformer of traffic sign detection for long-range perception of autonomous driving," *Engineering Applications of Artificial Intelligence,* vol. 139, pp. 109536, 2025. https://doi.org/10.1016/j.engappai.2024.109536

[43] H. S. G. Yaamini, K. J. Swathi, N. Manohor, G. A. Kumar, "Lane and traffic sign detection for autonomous vehicle: addressing challenges on Indian road conditions," *MethodsX,* vol. 14, pp. 103178, 2025. https://doi.org/10.1016/j.mex.2025.103178

[44] Y. Han, F. Wang, W. Wang, X. Zhang, X. Li, "EDN-YOLO: Multi-scale traffic sign detection method in complex scenes," *Digital Signal Processing,* vol. 153, pp. 104615, 2024 https://doi.org/10.1016/j.dsp.2024.104615.

[45] F. Mercaldoa, F. Martinellib, A. Santonea, "Real-Time Road Sign Localisation through Object Detection," *Procedia Computer Science,* vol. 246, pp. 30–37. https://doi.org/10.1016/j.procs.2024.09.225.

[46] Y. Sun, X. Li, D. Zhao, Q. Wang, "Evolving traffic sign detection via multi-scale feature enhancement, reconstruction and fusion," *Digital Signal Processing*, vol. 160, pp. 105028, 2025 https://doi.org/10.1016/j.dsp.2025.105028.