INTERNATIONAL JOURNAL OF
AUTOMOTIVE SCIENCE AND TECHNOLOGY

2023, VOL. 7, NO: 2, 125-140 www.ijastech.org e-ISSN: 2587-0963

Solution of Real-Time Traffic Signs Detection Problem for Autonomous Vehicles by Using YOLOV4 and Haar Cascade Algorithms

Fatma Nur Ortataş[1*], Emrah Çetin[1]

0000 0001 7897 9958[1], 0000 0002 7023 6604[1]

[1] *Yozgat Bozok University, Faculty of Engineering and Architecture, 66200, Yozgat, Turkey*

Abstract

Unmanned systems are increasingly used today to facilitate our daily lives and use time more efficiently. Therefore, this rapidly emerging and growing technology appears in every aspect of our lives with its various functions. Object recognition algorithms are one of the most important functions that we often encounter in these systems. Autonomous vehicle technologies are the latest and fastest-growing technology among unmanned systems. In this study, we investigate the success rates of two different algorithms for recognizing traffic signs and markings that can be used for partially or fully autonomous vehicles. In this study, two different solutions to the problem of recognizing the signs for fully autonomous and semi-autonomous vehicles, respectively, were presented and the correct identification of the markers was evaluated. The work was performed in real-time. Two different concepts were used for these products. An enclosed space where an ideal lighting environment is provided for the evaluation of models should be visualized. In addition, for the general recognition of the models, the test procedures were performed with a dataset obtained from the users and it was computed for the general recognition. In addition, this study aims to provide a better understanding of the basic working principles, the differences between machine learning and deep learning, and the contents of object recognition processes. Within the scope of the study, the detection success of real-time traffic markers belonging to YOLOv4, and Haar Cascade algorithm was measured. In the study performed with the original dataset, the overall correct detection success rate because of the tests of the YOLOv4 algorithm was 99% on average, and the correct detection success rate obtained because of the testing of the Haar Cascade algorithm was 61%.

Keywords: Autonomous vehicle, YoloV4, Haar cascade, Deep learning, Object detection, Machine learning

Research Article
https://doi.org/10.30939/ijastech..1231646

Received 09.01.2023
Revised 14.03.2023
Accepted 18.04.2023

* Corresponding author

Fatma Nur ORTATAŞ

f.nur.ortatas@yobu.edu.tr

Address: Yozgat Bozok University, Faculty of Engineering and Architture ,Computer Engineering66200, Yozgat, Turkey
Tel: +905066502394

1. Introduction

The human brain, with its perfect structure, is a faultless mechanism that can easily understand where the objects are and how far they are. For this reason, when it is desired to develop an unmanned system based on vision, this mechanism must be set up as perfectly as possible. This mechanism is an essential need in all autonomous vehicle technologies. An autonomous vehicle needs to perceive the environment just like a human. Vision cameras are used as efficient tools for recognizing and perceiving the environment for that. The process of distinguishing the object that is desired to be detected among many objects in the image from these cameras very a crucial task. Therefore, object detection algorithms are frequently used in these technologies. Various object detection algorithms are available in the literature. While algorithms such as Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) are used in machine learning, Region Convolutional Neural Networks (RCNN), Convolutional Neural Networks (CNN), and Fast- RCNN are more common in deep learning. Among all these algorithms, the You Only Look Once (YOLO) algorithm is one of the most preferred algorithms in terms of object detection success and speed [1].

The structure of the YOLO algorithm is to determine which objects are and where they are, by reading the image once. With this algorithm, the locations of all classes can be detected once the image is looked at. The algorithm's ability to detect objects at once directly affects its performance. For this reason, the model is advantageous compared to other cause detection algorithms. The first superiority is that the model is quite fast. The second one is that the traditionally known window-shifting or region estimation methods are not used while performing the classification. Instead, after reading the whole image, it splits the image into a grid. A confidence score is created from the information of the objects in the boxes created with the information extracted from the labels of the objects during the training and testing phase. According to these confidence scores, it can be determined which object the object is and where it is [2]. When the studies on autonomous vehicles with the YOLO algorithms are examined, it is seen that the YOLO algorithm is used for object detection firstly. In the study of Sarda et al., a trained YOLOv4 model was applied with data set in which a car, person, building, traffic light, traffic sign, and bicycle were defined. As a result of this study, it has been seen that it has an accuracy value of approximately %75 [1]. In another study, Kavitha R. and Nivetha S., aimed that an autonomous vehicle would reduce the risk of accidents in rainy or bad weather conditions. For achieving this, YOLOv3 model training was carried out. The detection of pothole or bad road conditions on the road had successful results in this research [3]. In another study, M. Gby luhakoviand ć, M. Herceg, the YOLO algorithm is used to detect vehicles around the autonomous vehicle. YOLOv2 model and Robot Operation Systems (ROS) are applied together. In this application, a car, a bus, a van, and a truck detection are provided with the YOLOv2 model. Thus, the distance and location information of the vehicle is obtained. As a result, possible accident or collision situations can be estimated around autonomous vehicles [4]. S. Chen and W. Lin were conducted to determine which version of the YOLO algorithms should be used in the embedded systems autonomous vehicles. In this paper, the YOLOv3 and YOLOv3 Tiny models were trained and tested by applying them for vehicle object detection. When the results were examined, it is observed that the YOLOv3 model is more successful than the YOLOV3 Tiny model [5]. Z. Xu et. Al. used the YOLO algorithm in unmanned aerial vehicles, which is one of the leading unmanned systems. In the study, the YOLOv2 algorithm was trained and tested for the detection of small objects in aerial images. According to the tests performed, the Mean Average Precision (mAP) yielded results of approximately 87%. [6].

The most basic task of autonomous vehicles is to be able to recognize traffic lane markers to be able to successfully move on the road. As seen from the literature, researchers focused on the detection of traffic signs by using the YOLO algorithm that has been carried out based on the applications encountered in autonomous vehicles. Successful results have been obtained in this study by Valeja et.al. by applying it to speed limit signs with a YOLO model trained on the CARLA simulator [7]. W. Yang and W. Zhang used a total of 15000 photographs dataset that includes Chinese warnings, speed limits, and prohibitive and instructing direction signs. They compared the YOLOv3 and YOLOv4 models' training with this dataset. According to the result of the study, the YOLOv4 model gave more successful results than the YOLOv3 model [8]. YOLOv3 model training with traffic signs is used by Mohd-Isa et.al. in Malaysia which resulted in a success of %90 [9]. In the same way, Novak et.al. achieved traffic sign detection with the YOLOv3 model in their research [10]. The importance of the dataset to be used in the detection of traffic signs is given in the study of Dewi et.al. The YOLOv4 model training was made with pictures of different resolution quality. And the synthetic images created to be added to the dataset. This has been more successful in detecting traffic signs [11]. The haar cascade is another object detection algorithm in the literature. With the haar cascade algorithms, there are basic processing steps for object detection operations. The first is to extract the features of the object to be detected in the image, called haar features. In this process, the image is divided into sub-frames by using the color distribution, contrast, and saturation level differences. The numerical information of the image is extracted from the differences in these frames, and they form the haar features. The haar features will be revealed by summing the pixel intensities of each Region and the difference between these sums. Attributes must be extracted for this process. For this process, information is collected from the image by using basic features such as edge, line, and center perimeter attributes [12]. To perform all these operations much faster, a method called integral image generation is applied in the study of Viola and Jones. The value of the integral image at the (x, y) position of the points belonging to this position. Then, the classifier needs to be trained according to the generated information. After scanning the whole image with haar feature extraction processes, it is necessary to separate the unrelated ones from the thousands of features obtained. For this, the Adaboost learning algorithm is used. Thus, combinations with weak classifiers are applied to create strong classifiers. Then, it is trained by using reinforcements to be made on weak learners. This process is called a cascade classifier. As a result of this process, a powerful classifier is obtained [13]. Haar cascade algorithm is one of the choices of the researchers used for autonomous vehicles in the literature. Irawan et.al. used the haar cascade algorithms in the alignment of the forklift by detecting the pulleys related to the use of mini heavy–loaded autonomous forklifts in the area operation [14]. The haar cascade algorithm was applied for vehicle detection and successful results are obtained in [15]. Therewithal, Arunmozhi et.al. have taken successful results obtained for the detection of stop signs [16]. Arunmozhi et.al. achieved the detection of traffic signs, in the evening time by using the haar cascade algorithm [17]. Recognition of 5 different road signs related to raspberry pi for autonomous vehicles demonstrated by Vinothini et. al. The process of recognizing road signs has been successfully completed in their research [18]. Another study is applied to the autonomous vehicle model by Mohit et.al. It has been depicted that the haar cascade algorithm performs it successfully for basic autonomous vehicle tasks such as stop sign and lane detection [19]. In another study, Dewi et al studied the detection of traffic markers using YoloV4 and YoloV4 tiny combined with Spatial Pyramid Pooling (SPP). As a result of the study, it was observed that it was detected with 99.4% accuracy [20]. In another study conducted in this area, Aysal et al., with the YoloV5 model, showed that the detection of traffic markers was performed with an accuracy of 97% [21].

In this study, the recognition success of the algorithms was observed by training the Haar Cascade and YOLOV4 models with

10 different traffic signs. Two different environments were prepared to test the models. The recognition success of the models in these environments was evaluated. Also, the recognition rates of the models belonging to these algorithms were calculated using the images of the traffic flow captured in real time. In addition, this study will provide a better understanding of the basic working principles, differences between machine learning and deep learning, and the contents of the processes for object recognition.

2. Material and Method

2.1 Model Definition

Zed2 camera was used in the tests of YOLOv4 and Haar cascade algorithms trained for detection of traffic signs. In addition, the computer features of the model trainings are given below.

- Msi z390m Motherboard
- Intel core i7 9700K
- Corsair 2x8 GB DDR4 Memory
- 750-Watt Power Supply
- Kingston 500 GB m2NVMe SSD
- Nvidia GTX 2060

While choosing the features of the computer, it is taken into consideration that real-time work will be done. For this reason, a Cuda core graphics card is preferred especially for the real-time performance quality of the YOLO algorithm. Other features, on the other hand, are preferred for their up-to-date and harmonious operation to increase the efficiency of the performance to be obtained from the graphics card.

In addition to all these features, Stereo Labs Zed 2, a stereo camera type, was preferred for the camera that is critical to detecting the traffic signs used in the vehicle. Stereo cameras are often preferred in the area such as autonomous vehicles and augmented reality technology. Stereo imaging is based on the logic of calculating depth by combining images taken from different angles with two cameras. Since this camera act like a human eye in detecting traffic signs and detecting the distance of the autonomous vehicle, it also provides depth information as well as the image. The specifications of the Stereo Labs Zed 2 camera are given in below.

Depth Information	0.2 to 20 meters
Frames Per Second	720p at 60 FPS
(FPS)	1080p at 30 FPS
Field of View	120°
Size	124 gr /175 x 30 x 33 mm

The name information of the traffic signs, which were tested within the scope of the study and whose model trainings were carried out, is given in Table m1.

Table 1. Traffic signs and names

Turn Left	Stop Sign	Station Sign	Straight Ahead or Turning Right	No Traffic Sign
				
Parking Sign	No Left Turn Sign	No Entry Sign	Straight Ahead or Turning Left	No Parking Sign
				

Model training was performed using Haar Cascade and YOLOv4 for the signals listed in Table 1. Two different study areas were prepared to test the trained models. For the study, an environment with ideal lighting was prepared. For the second test, an open area with natural light was prepared. This is also in a location that contains test-related information, and the models are clearly visible. The camera used in the test processes is placed on a fixed place on the vehicle. Within the scope of the study, it was worked with 30 FPS (Frames Per Second) imaging speed. It has been observed that a CNN model trained for object detection can successfully detect under 280 lumens of light [22]. The amount of illumination available in the indoor environment with ideal illumination of the study was measured as 300 lumens with a luxmeter. The test results obtained in these different areas are each listed in a separate table. Information on the success rates in training the models and the dataset is also included in 2.2. Haar Cascade algorithm and 2.3. YOLOv4 algorithm.

2.2. Haar Cascade Algorithm

As the basic working principle of the Haar Cascade algorithm, it performs the recognition process by dividing the image given as input for the object to be detected into sub-windows. This sub-windows separation is performed by utilizing the image properties. For these features, the so-called digital image features of the image, such as different color distributions, contrast ratios, and saturation levels, can be given as examples. The name given to all these features is the haar features. Separate sub-windows are opened for each different haar feature of the image, which is divided into sub-windows according to the haar features. Therefore, the information that each sub-window carries is different [12]. The necessary steps for training the haar cascade algorithm are given in Figure 1.

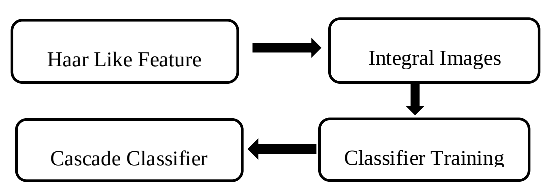
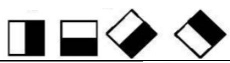


Fig. 1. Haar cascade process steps

According to the information given in Figure 1, firstly, there is the extraction of haar-like features. For this process, images are needed for model training. For this, images of the object to be detected, called positive, and images of places where this object is likely to be found, called negative, are needed. Afterward, feature extraction will be performed. For this, the features to be applied and each feature to be extracted are a single value obtained by subtracting the sum of the pixels under the white rectangle from the sum of the pixels under the black rectangle. By applying the features given in Table 2, feature extraction on the input image is performed [23].

Table 2. Some examples of haar-like features

Edge Features	
Line Features	
Center-surround Features	

Integral image is used for fast feature detection. The operation performed here is essentially a collection operation. Thus, the integral is formed by summing the pixel values above these points for the (x,y) position of the image. The image of this is given in Figure 2.

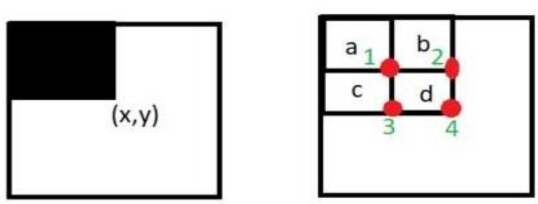


Fig. 2. Integral image

When the example given in Figure 3 is examined, for example, the value in the 4th position is calculated from the quadrilaterals a+b+c+d.

In the classifier training that will take place after the integral image formation, the feature extraction operations on the images are given as the input image with a sliding window. Most of the information obtained because of this scan is unrelated. For this reason, the Adaboost algorithm is used to create an effective classifier to be obtained from here. Thus, by distinguishing features with low error rates, it is ensured that objects are detected and classified correctly. In the cascade classifier to be implemented after this stage, the area of the detected object on the input image will be small, so this part is separated from the image. Thus, data on the location information of the object that is desired to be detected at high rates is collected. Thus, focusing on the areas where the object to be detected can be found on the image is ensured [12].

2.2.1. Haar Cascade Model Test

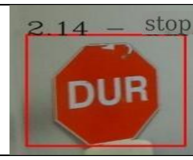
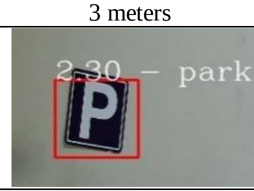
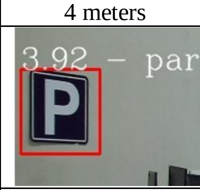
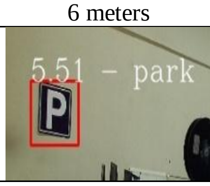
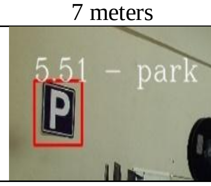
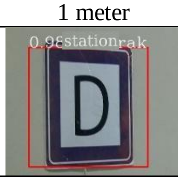
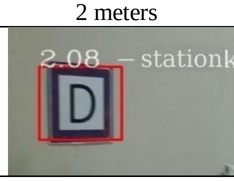
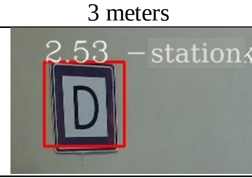
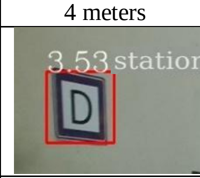
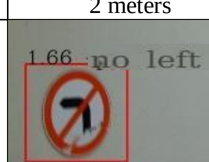
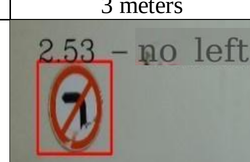
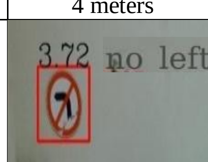
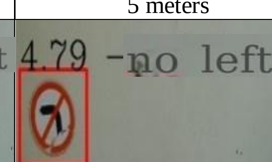
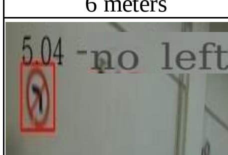
The dataset information used for model training of the haar cascade algorithm, which was trained in the study, is given in Table 3. The images that make up this whole dataset have been resized to vary between 95x95 and 110x110. In addition, for this model training, merging processes were carried out at different angles and sizes by using approximately 20.000 negative images for the positive dataset of each traffic sign. For the model training carried out, new images were created by superimposing positive images of different sizes on 20.000 negative photographs, as well as the dataset of each traffic sign specified in Table 3, with different angle and rotation processing.

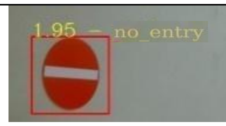
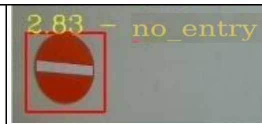
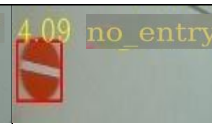
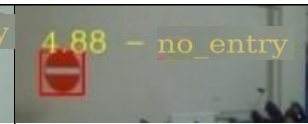
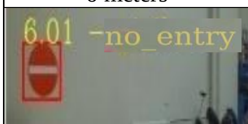
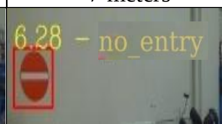
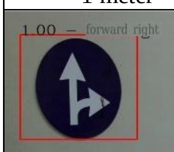
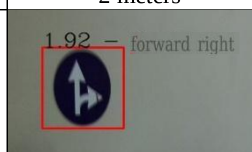
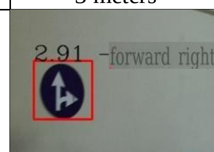
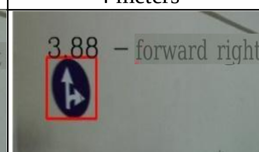
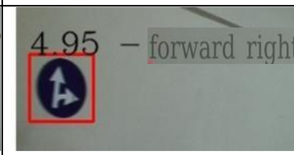
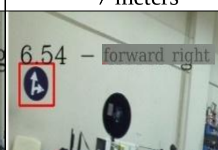
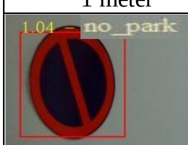
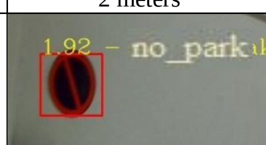
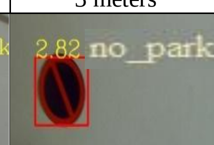
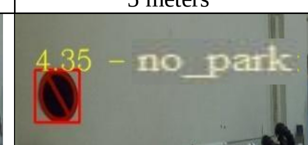
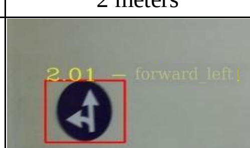
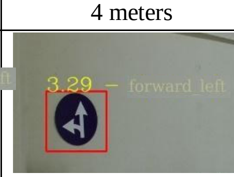
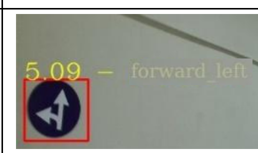
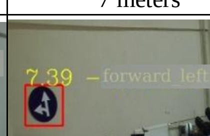
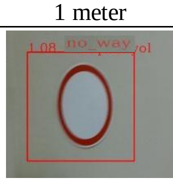
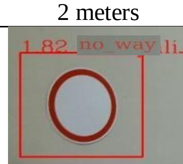
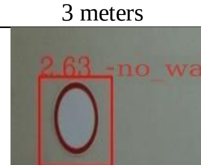
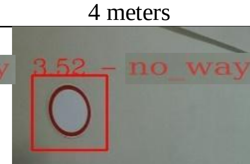
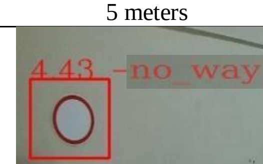
Table 3. Dataset Information for Haar Cascade Algorithm Training

Traffic Signs Name	Number of Images
Turn Left	90
Stop Sign	100
Parking Sign	110
Station Sign	125
No Left Turn Sign	100
No Entry Sign	118
Straight Ahead or Turning Right	100
No Parking Sign	90
Straight Ahead or Turning Left	95
No Traffic Sign	100

After the model's training has been completed, the test was carried out primarily in an enclosed space with the ideal lighting environment. For this environment, scaled distances are prepared to be 1 meter, 2 meters, 3 meters, 4 meters, 5 meters, 6 meters and 7 meters per traffic sign. This has observed the model's detection success at different distances. The data for this situation is given in Table 4. Testing up to 7 meters in an indoor with ideal lighting is due to the size of the area being tested. A pre-scaled outdoor test field is also prepared up to 20 meters with natural lighting. The results of the tests for the Haar Cascade algorithm, which is model trained in this test area, are provided in Table 5. The images given in Table 4 and Table 5 show 2 different values on the plates detected by the model. The first is the distance information of the plate identified to the camera, and the second value is the traffic sign name detected. There are cases where the traffic sign cannot be detected in the open area and indoor tests. The main reason for this is the amount of light coming to the traffic sign, shading or various features of the training data used in the dataset, such as distance, direction, angle and illumination. In addition, it gives -inf or -nan value in cases where it is detected but the distance cannot be determined by the Zed2 Camera used in the study. These values appear when the distance cannot be calculated or when the camera is outside the measuring range. The main reason why it is less undetected in indoor tests is that it is done in an ideal lighting environment and takes place in a smaller area. There is variation in detection status due to the use of natural lighting in the test processes performed outdoors.

Table 4. Traffic sign detection success of the haar cascade model for different distances at indoors and in an ideal lightning environment

Traffic Sign Name	Distance				
	1 meter	2 meters	3 meters	4 meters	5 meters
Stop Sign					
	6 meters	7 meters	8 meters	9 meters	10 meters
			Not detected	Not detected	Not detected
	1 meter	2 meters	3 meters	4 meters	5 meters
Parking Sign					
	6 meters	7 meters	8 meters	9 meters	10 meters
			Not detected	Not detected	Not detected
	1 meter	2 meters	3 meters	4 meters	5 meters
Station Sign					
	6 meters	7 meters	8 meters	9 meters	10 meters
			Not detected	Not detected	Not detected
	1 meter	2 meters	3 meters	4 meters	5 meters
No Left Turn Sign					
	6 meters	7 meters			
					
	1 meter	2 meters	3 meters	4 meters	5 meters

No Entry Sign					
	6 meters	7 meters			
					
Straight Ahead or Turning Right	1 meter	2 meters	3 meters	4 meters	5 meters
					
	6 meters	7 meters			
					
No Parking Sign	1 meter	2 meters	3 meters	4 meters	5 meters
					
	6 meters	7 meters			
					
Straight Ahead or Turning Left	1 meter	2 meters	3 meters	4 meters	5 meters
					
	6 meters	7 meters			
					
No Traffic Sign	1 meter	2 meters	3 meters	4 meters	5 meters
					
	6 meters	7 meters			

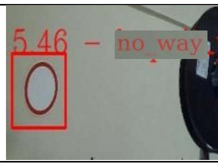
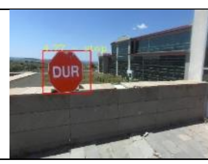
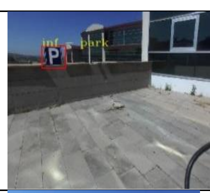
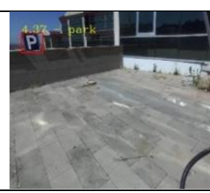
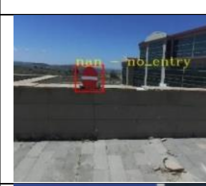
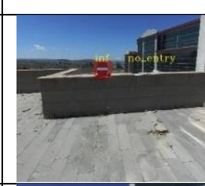
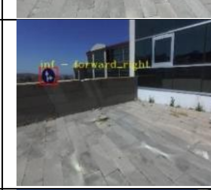
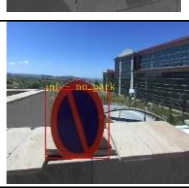
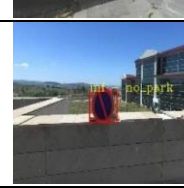
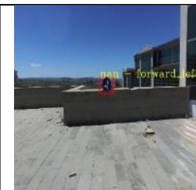
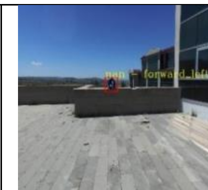
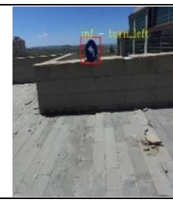
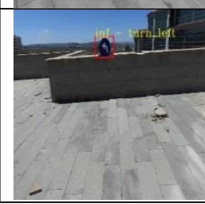
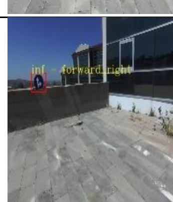
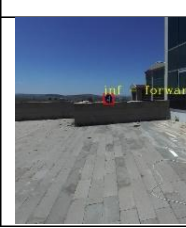
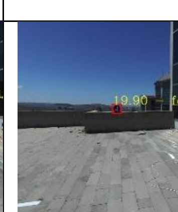
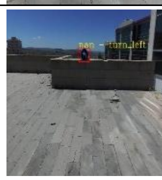
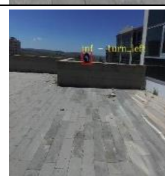
					
Turn Left	1 meter	2 meters	3 meters	4 meters	5 meters
					
	6 meters	7 meters			
					

Table 5. Traffic sign detection success of the haar cascade model for different distances at outdoors and in a natural lightning environment

Traffic Sign Name	Distance					
	1 meter	3 meters	5 meters	7 meters	9 meters	10 meters
Stop Sign						
Station Sign	Not detected	Not detected	Not detected	Not detected	Not detected	Not detected
Parking Sign						
No Left Turn Sign	Not detected		Not detected	Not detected	Not detected	Not detected
No Entry Sign						
Straight Ahead or Turning Right						
No Parking Sign						

Straight Ahead or Turning Left							
No Traffic Sign	Not detected	Not detected					
Turn Left							
Traffic Sign Name	Distance						
	11 meters	13 meters	15 meters	17 meters	18 meters	19 meters	20 meters
Parking Sign							
Station Sign	Not detected	Not detected	Not detected	Not detected	Not detected	Not detected	Not detected
No Entry Sign				Not detected	Not detected	Not detected	Not detected
Straight Ahead or Turning Right			Not detected	Not detected	Not detected	Not detected	Not detected
No Parking Sign				Not detected	Not detected	Not detected	Not detected
Straight Ahead or Turning Left							

No Traffic Sign							
Turn Left				Not detected	Not detected	Not detected	Not detected

The success of the Haar Cascade algorithm with model training on different photos was also measured as a percentage. The results for these measurements are given in Table 6.

Table 6. Outdoor general detection success of the model-trained haar cascade algorithm

Traffic Signs Name	Haar Cascade Model Accuracy Rates(%)
Turn Left	48
Stop Sign	60
Parking Sign	76
Station Sign	48
No Left Turn Sign	44
No Entry Sign	80
Straight Ahead or Turning Right	50
No Parking Sign	64
Straight Ahead or Turning Left	84
No Traffic Sign	56

2.3.YOLOv4 Algorithm

As the basic working principle of the Yolo algorithm, it divides the input image for the object to be detected into SxS grids at one. A scan is performed based on the labeling information of the object to be detected in these divided grids. As a result, a confidence score is created according to the status of the object to be detected belonging to each grid in that grid. According to the confidence scores, there are frames called bounding boxes around the objects. The main purpose of these bounding boxes is to determine how well they frame the desired object.

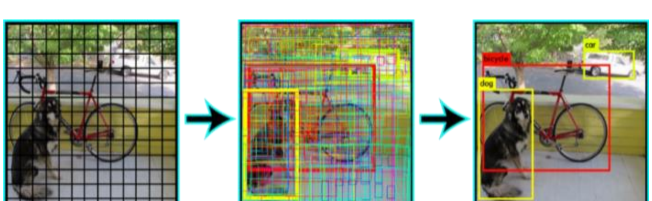


Fig. 3. YOLO working principle

A single-stage detector strategy is used for this very fast algorithm, where all these processes are carried out in one go. An object detector consists of input, backbone, neck, and head. The part that makes up each detector is different and consists of more than one structure [24]. The one-stage detector structure is shown in Figure 4.

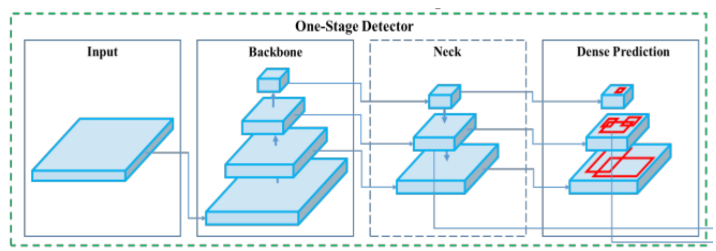


Fig. 4. One-stage detector

When YOLOv4 is examined in detail:

- Backbone: CSPDarknet53
- Neck: SSP, PAN
- Head: YOLOv3 is available.

The schematic structure of YOLOv4 is shown in Figure 5.

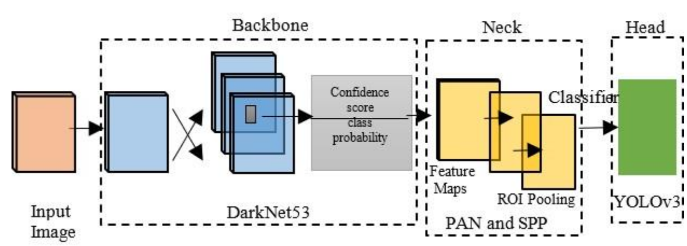


Fig. 5. Yolov4 structure

The backbone part usually consists of pre-trained networks over ImageNet. The CSPDarknet53 network is used for YOLOv4. This network is based on DenseNet. The main purpose of DenseNet is to connect the layers in convolutional neural networks. There are 53 layers in the DarkNet53 layer. In the neck part, a feature collection takes place. The main purpose is to mix and collect the features here by moving the neck components in the layers in convolutional neural networks. PAN structure is used for these operations. The SPP block is used to separate the most important features detected from the backbone. The head part of YOLOv4, on the other hand, performs the detection process by using the same YOLO head as the YOLOv3.

2.3.1. YOLOv4 Model Test

The dataset information used for model training of the YOLOv4 algorithm, which was trained in the study, is given in Table 7.

Table 7. Dataset information for YOLOv4 algorithm training

Traffic Signs Name	Number of Images
Turn Left	2168
Stop Sign	1279
Parking Sign	1554
Station Sign	1154
No Left Turn Sign	1842
No Entry Sign	1814
Straight Ahead or Turning Right	1478
No Parking Sign	1908
Straight Ahead or Turning Left	1611
No Traffic Sign	192

Within the scope of the study, Darknet installation was carried out for YOLOv4 model training. Later, the image and label to be used in the training were moved to the necessary places in the file. In the configuration file in the darknet file, the configuration file of the YOLOv4 model, which will be trained for the model, was changed and the batch value for model training was set to 64 and the subdivision value was set to 16. The training was carried out with 80.000 iterations. The learning rate is given as 0.002 in the configuration file of the model training. As a result of model training, the loss value decreased to 0.06. The loss graph obtained because of model training is given Figure 6.

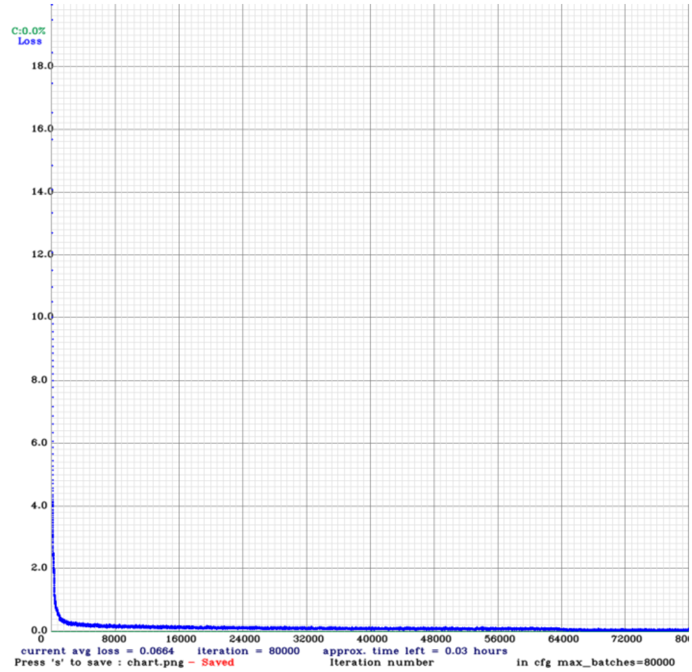


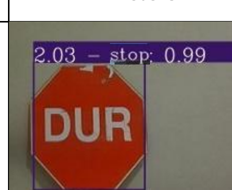
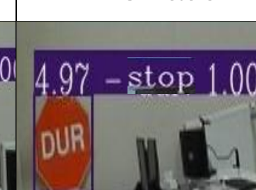
Fig.6. Loss graph of YOLOv4 model training

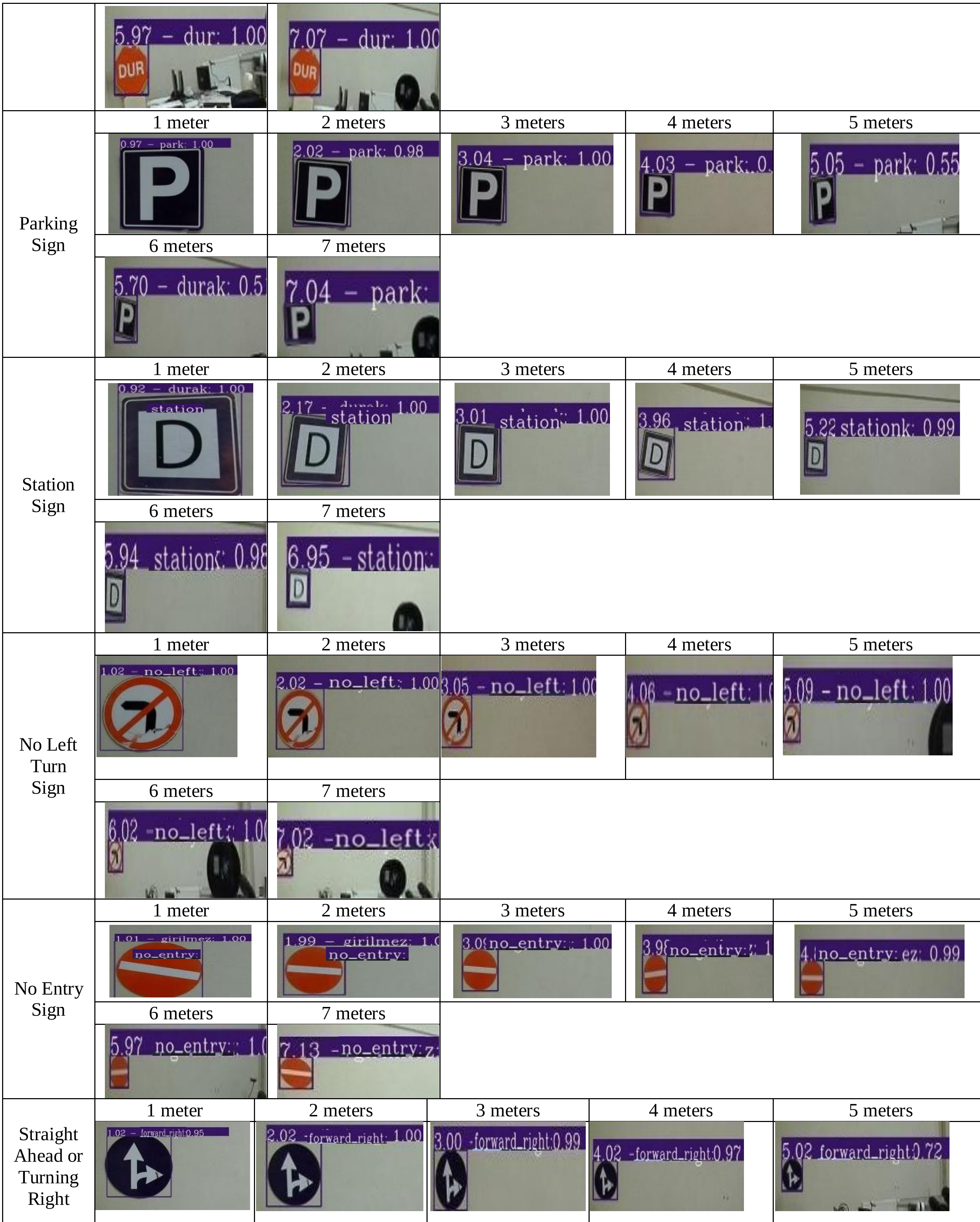
Within the scope of the study, the success of the model trained YOLOv4 algorithm on different photos was also measured as a percentage. The results for these measurements are given in Table 8.

Table 8. Outdoor general detection success of the model trained YOLOv4 algorithm

Traffic Signs Name	YOLOv4 Model Accuracy Rates (%)
Turn Left	100
Stop Sign	100
Parking Sign	100
Station Sign	98
No Left Turn Sign	92
No Entry Sign	100
Straight Ahead or Turning Right	100
No Parking Sign	98
Straight Ahead or Turning Left	98
No Traffic Sign	100

Table 9. Traffic sign detection success of the YOLOv4 model for different distances at indoor and in an ideal lightning environment

Traffic Sign Name	Distance				
	1 meter	2 meters	3 meters	4 meters	5 meters
Stop Sign					
	6 meters	7 meters			


Parking Sign
Station Sign
No Left Turn Sign
No Entry Sign
Straight Ahead or Turning Right
1 meter
2 meters
3 meters
4 meters
5 meters
6 meters
7 meters

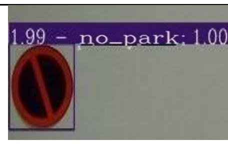
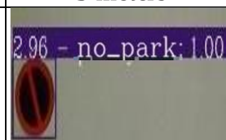
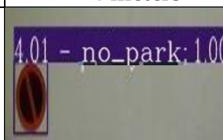
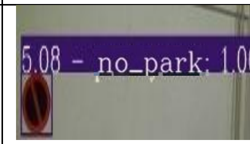
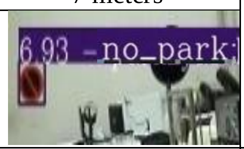
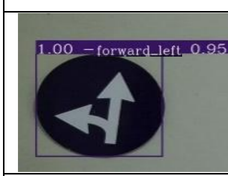
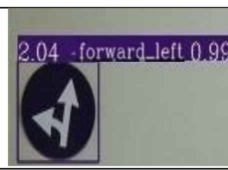
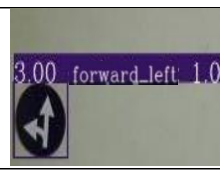
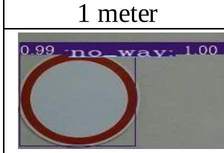
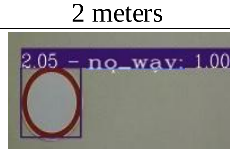
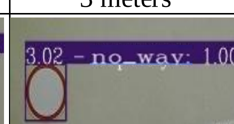
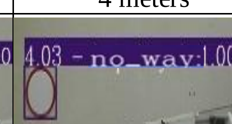
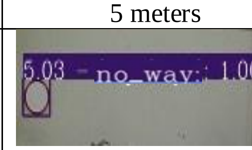
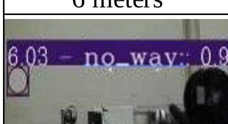
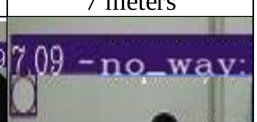
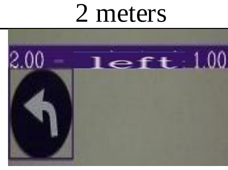
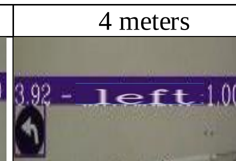
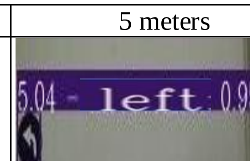
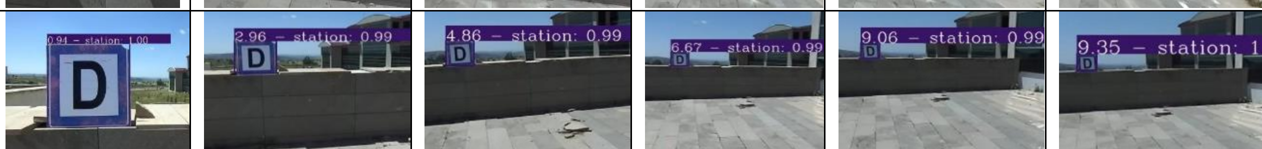
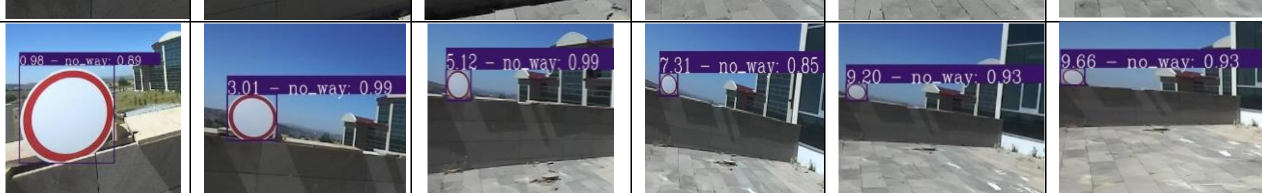
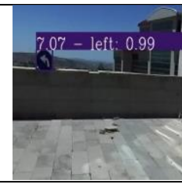
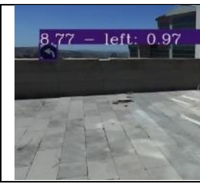
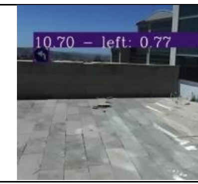
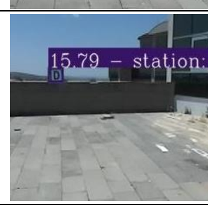
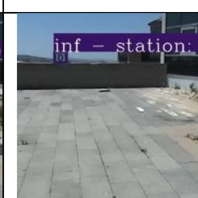
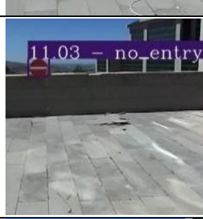
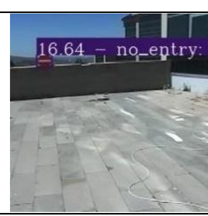
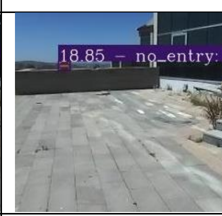
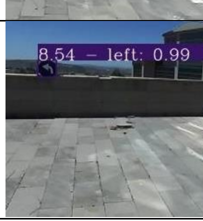
<table>
<tr><td rowspan="2"></td><td>6 meters</td><td>7 meters</td><td colspan="3" rowspan="2"></td></tr>
<tr><td></td><td>Not detected</td></tr>
<tr><td rowspan="4">No Parking Sign</td><td>1 meter</td><td>2 meters</td><td>3 meters</td><td>4 meters</td><td>5 meters</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>6 meters</td><td>7 meters</td><td colspan="3" rowspan="2"></td></tr>
<tr><td></td><td></td></tr>
<tr><td rowspan="4">Straight Ahead or Turning Left</td><td>1 meter</td><td>2 meters</td><td>3 meters</td><td>4 meters</td><td>5 meters</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>6 meters</td><td>7 meters</td><td colspan="3" rowspan="2"></td></tr>
<tr><td></td><td>Not detected</td></tr>
<tr><td rowspan="4">No Traffic Sign</td><td>1 meter</td><td>2 meters</td><td>3 meters</td><td>4 meters</td><td>5 meters</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>6 meters</td><td>7 meters</td><td colspan="3" rowspan="2"></td></tr>
<tr><td></td><td></td></tr>
<tr><td rowspan="4">Turn Left</td><td>1 meter</td><td>2 meters</td><td>3 meters</td><td>4 meters</td><td>5 meters</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>6 meters</td><td>7 meters</td><td colspan="3" rowspan="2"></td></tr>
<tr><td></td><td></td></tr>
</table>

Table 10. Traffic sign detection success of yolov4 model for different distances at outdoor and in a natural lightning environment

Traffic Sign Name	Distance					
	1 meter	3 meters	5 meters	7 meters	9 meters	10 meters
Stop Sign						
Parking Sign						
Station Sign						
No Left Turn Sign						
No En-try Sign						
Straight Ahead or Turn-ing Right					Not detected	Not detected
No Parking Sign						
Straight Ahead or Turn-ing Left						
No Traffic Sign						

Turn Left						
Traffic Sign Name	Distance					
	11 meters	13 meters	15 meters	17 meters	19 meters	20 meters
Stop Sign						
Parking Sign				Not detected	Not detected	Not detected
Station Sign						
No Left Turn Sign			Not detected	Not detected	Not detected	Not detected
No Entry Sign						
No Parking Sign		Not detected	Not detected	Not detected	Not detected	Not detected
Turn Left		Not detected	Not detected	Not detected	Not detected	Not detected

3.RESULTS

In this study, Haar Cascade and YOLOv4 model trainings used to figure out, to ensure autonomous movement of the vehicle on the model vehicle designed as an electric and autonomous vehicle. The features of the vehicle are specified in the Model Definition section. The model training is evaluated to test the performed training under the same conditions in both indoor and outdoor environments. Firstly, an indoor area with ideal conditions was prepared. Secondly, a test environment was prepared for both algorithms at outdoor with natural lighting during the day for the test procedures. With this study, the traffic sign detection problem was solved by two different technologies. So that, the differences in the basic structural and model training processes between machine learning and deep learning have been revealed.

For the test environment, the locations where the traffic signs have been placed and scaled to 7 meters at one-meter intervals for indoor use, while this scaling has been prepared to 20 meters for outdoor use. During the test process, the real-time detection processes were carried out. Zed2 Stereo camera was used in the test software of the trained models for both algorithms. Two separate datasets, consisting of the same traffic signs used in model training, with different numbers and different resolutions, were used. The test results and compatibility of the cameras with Haar Cascade and YOLOV4 models under all these conditions are given in Tables 4,5,9 and 10. The tests of the models of both algorithms indoor and outdoor areas were successfully performed. However, although both models were tested with the same stereo camera, it was found that the model of the YOLOV4 algorithm provided much more accurate results in terms of the distance of the detected traffic sign from the camera, i.e. the distance information.

In addition, a dataset of images of general traffic flow was created, except for the real-time test, which was performed in each test areas. This dataset was created for testing purposes to calculate the overall detection rates for the models of the two algorithms separately. While creating the dataset, the images were chosen as different images which are common in daily life, and they were taken at different times of the day. All datasets were prepared originally. The results of this test datasets, the overall detection rates were calculated. The values of the Haar-Cascade algorithm are shown in Table 6. Also, Table 8 shows the general detection rate of the YOLOv4 algorithm performed on the same test data. Some visual demonstrations for this study can be found in Appendix 1.

The solution to the traffic sign detection problem for autonomous vehicles is solved by two different algorithms in this study. The overall detection in real-time success rates of the two models were calculated. It has been proved that the success rate of object detection in the models of both algorithms is directly affected by the diversity of the images which compose the dataset. The differences in the ambient lighting ratios, and the resolution differences of the images that compose the dataset. These features need to be considered as they directly affect the object detection quality and efficiency of the model. Within the scope of the study, the detection success of real-time traffic markers belonging to YOLOv4, and Haar Cascade algorithm was measured. In the study performed with the original dataset, the overall correct detection accuracy rate because of the tests of the YOLOv4 algorithm was 99% on average, and the correct detection accuracy rate obtained because of the testing of the Haar Cascade algorithm was 61%.

Conflict of Interest Statement

No conflict of interest was declared by the authors.

CRediT Author Statement

Fatma Nur Ortataş: Conceptualization, Validation, Methodology, Analysis, Investigation
Emrah Çetin: Supervision, Conceptualization, Project administration

References

[1] Sarda A, Dixit S, Bhan A. Object Detection for Autonomous Driving Using Yolo [You Only Look Once] Algorithm. 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV). 2021;1370-1374.

[2] Redmon J, Divvala S, Girshick R, Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016;779-788.

[3] R K, NS. A. Pothole and Object Detection for An Autonomous Vehicle Using Yolo. 2021 5Th International Conference on Intelligent Computing and Control Systems (ICICCS). 2021;1585-1589.

[4] Gluhaković M, Herceg M, Popovic M, Kovačević J. Vehicle Detection in The Autonomous Vehicle Environment for Potential Collision Warning. 2020 Zooming Innovation in Consumer Technologies Conference (ZINC). 2020;178-183.

[5] Chen S, Lin W. Embedded System Real-Time Vehicle Detection Based on Improved Yolo Network. 2019 IEEE 3Rd Advanced Information Management Communicates, Electronic and Automation Control Conference (IMCEC). 2019;1400-1403.

[6] Xu Z, Shi H, Li N, Xiang C, Zhou H. Vehicle Detection Under Uav Based on Optimal Dense Yolo Method. 2018 5Th International Conference on Systems and Informatics (ICSAI). 2018;407-411.

[7] Valeja Y, Pathare S, Patel D, Pawar M. Traffic Sign Detection Using Clara and Yolo in Python. 2021 7Th International Conference on Advanced Computing and Communication Systems (ICACCS). 2021; 367-371.

[8] Yang W, Zhang W. Real-Time Traffic Signs Detection Based on Yolo Network Model. 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC). 2020;354-357.

[9] Mohd-Isa W, Abdullah M, Sarzil M, Abdullah J, Ali A, Hashim N. Detection of Malaysian Traffic Signs Via Modified Yolov3 Algorithm. 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI). 2020;1-5.

[10] Novak B, Ilić V, Pavković B. Yolov3 Algorithm with Additional Convolutional Neural Network Trained for Traffic Sign Recognition. 2020 Zooming Innovation in Consumer Technologies Conference (ZINC). 2020;165-168.

[11]Dewi C, Chen R, Liu Y, Jiang X, Hartomo KD. Yolov4 For Advanced Traffic Sign Recognition with Synthetic Training Data Generated by Various GAN. IEEE Access. 2021;9:97228-97242.

[12]Çetin E, Ortataş FN. Elektrikli Ve Otonom Araçlarda Makine Öğrenmesi Kullanarak Trafik Levhaları Tanıma ve Simülasyon Uygulaması. El-Cezeri. 2021;8(3): 1081 - 1092.

[13]Viola P, Jones M. Rapid Object Detection Using a Boosted Cascade of Simple Features. Proceedings Of The 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2001;I-I.

[14]Irawan A, Yaacob MA, Azman FA, Daud MR, Razali AR, Ali SNS. Vision-Based Alignment Control for Mini Forklift System in Confine Area Operation. 2018 International Symposium on Agent, Multi-Agent Systems and Robotics (ISAMSR). 2018;1-6.

[15]Arunmozhi A, Park J. Comparison of HOG, LBP And Haar-Like Features for On-Road Vehicle Detection. 2018 IEEE International Conference on Electro/Information Technology (EIT). 2018;0362-0367.

[16]Arunmozhi A, Gotadki S, Park J, Gosavi U. Stop Sign and Stop Line Detection and Distance Calculation for Autonomous Vehicle Control. 2018 IEEE International Conference on Electro/Information Technology (EIT). 2018;0356-0361.

[17]Caballero CUB, Beltrán ZZ. Detection Of Traffic Panels in Night Scenes Using Cascade Object Detector. 2018 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE). 2018;32-37.

[18]Vinothini K, Jayanthy S. Road Sign Recognition System for Autonomous Vehicle Using Raspberry Pi. 2019 5Th International Conference on Advanced Computing & Communication Systems (ICACCS). 2019;78-83.

[19]R.s. M, A SK, Vats M, S. S. Analyzing the Features of Self-Driving Cars Using Haar Classification Methodology. 2020 5Th International Conference on Communication and Electronics Systems (ICCES). 2020;1344-1350.

[20]Dewi C, Chen R, XJ, HY. Deep Convolutional Neural Network for Enhancing Traffic Sign Recognition Developed on YoloV4. Multimedia Tools and Applications. 2022;(81):37821–37845.

[21]Aysal FE, Yıldırım K, Cengiz E. Real-Time Application of Traffic Sign Recognition Algorithm with Deep Learning. Journal of Materials and Mechatronics: A. 2022;3(2):275 - 289.

[22]Carvalho S, Humphries J, Dunne N, Leahy S. Impact of Light Flickering on Object Detection Accuracy Using Convolutional Neural Networks. 2021 Telecoms Conference (ConfTELE). 2021;1-6.

[23]Sharifara A, Rahim MSM, Anisi Y. A General Review of Human Face Detection Including a Study of Neural Networks And Haar Feature-Based Cascade Classifier In Face Detection. 2014 International Symposium on Biometrics and Security Technologies (ISBAST). 2014;73-78.

[24]Bochkovskiy A, Wang C, Liao HM. Yolov4: Optimal Speed and Accuracy of Object Detection. Arxiv Preprint Arxiv:2004. 2020.

APPENDIX A

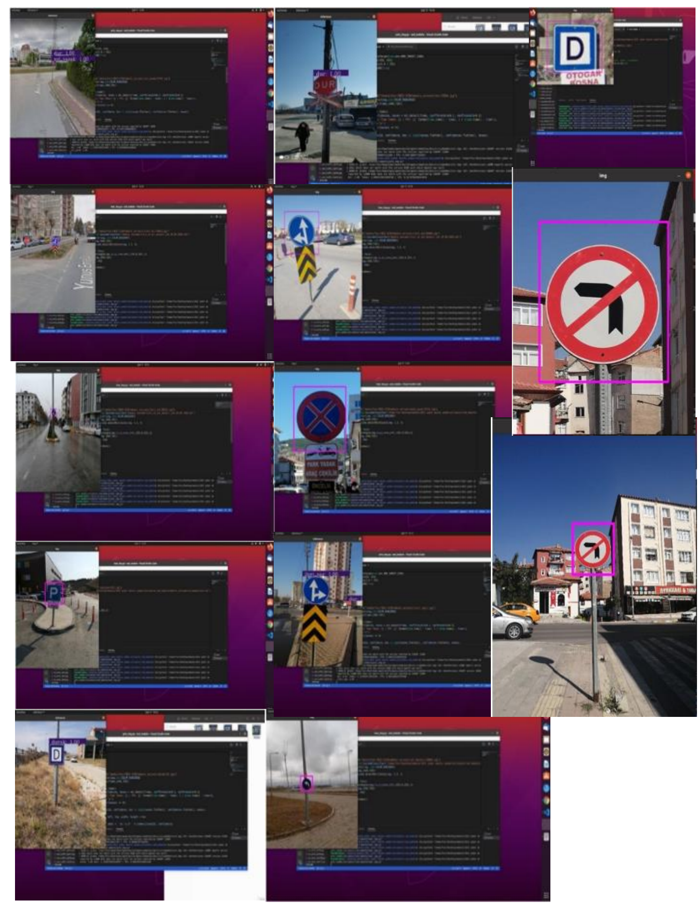


Fig. A1. Some images from the accuracy rate detection study