Celal Bayar University Journal of Science

Detection of Red, Yellow, and Green Lights in Real-Time Traffic Lights with YOLO Architecture

Abdil Karakan[1]*

[1] Department, Dazkırı Vocational Scholl, Afyon Kocatepe University, 03204 Afyonkarahisar, Türkiye
* akarakan@aku.edu.tr
* Orcid No: 0000-0003-1651-7568

Received: 5 February 2024
Accepted: 2 May 2024
DOI: 10.18466/cbayarfbe.1432356

Abstract

In the study, red, yellow, and green lights at traffic lights were detected in real-world conditions and in real time. To adapt to real-world conditions, A data set was prepared from traffic lights in different locations, lighting conditions, and angles. A total of 5273 photographs of different traffic lights and different burning lamps were used in the data set. Additionally, grayscale, bevel, blur, variability, added noise, changed image brightness, changed color vibrancy, changed perspective, and resized and changed position have been added to photos. With these additions, the error that may occur due to any distortion from the camera is minimized. Four different YOLO architectures were used to achieve the highest accuracy rate on the dataset. As a result, the study obtained the highest accuracy at 98.3% in the YOLOV8 architecture, with an F1-Score of 0.939 and mAP@.5 value of 0.977. Since the work will be done in real time, the number of frames per second (FPS) must be the highest. The highest FPS number was 60 in the YOLOv8 architecture.

Keywords: Deep learning, traffic light, real-time detection, YOLOV8

1. Introduction

In recent years, electric vehicles have come to the fore again. The most important reason is that technology is advancing rapidly, and fossil fuels are losing their appeal [1]. Another reason is that electric vehicles are environmentally friendly [2]. Internal combustion engines have carbon emissions. Although efforts are made to reduce these rates, they remain high. Carbon emissions exceed the acceptable level with the increased number of vehicles. For this reason, it has accelerated the search for cleaner automobiles globally [3]. With the rapid development of technology, the concept of driverless vehicles has been added to electric vehicles. Electric vehicles have a very suitable structure for autonomous vehicles [4]. Mechanical control in an electric vehicle is minimal. It is mainly controlled electrically. In an electric vehicle, many controls, such as vehicle movement, braking, charging, and steering control, are performed by electrical signals. For this reason, it facilitates software-based control [5]. Artificial intelligence is used in many areas. One of these is electric vehicles. Electric vehicles can recognize their surroundings through sensors around the car. Thanks to sensors, vehicles can be developed as driverless [6]. The first way to get information from the environment of an electric and autonomous vehicle is through cameras and sensors. The use of sensors produced with the advanced technology used today in autonomous vehicles is also efficient and active [7-13]. The most important example of this is traffic lights. More than detecting traffic lights by cameras is required. The condition of the lights is also essential for autonomous drivers. There are many studies on the detection of traffic lights [14-20].

Many different deep-learning architectures are used to detect traffic lights. YOLO architecture provides good results in detecting traffic lights. They focused on specific features such as shape, color, and texture to detect traffic lights. In the study, hand-made features were added to the system to prevent sensor and atmospheric noise. Bosch small traffic light dataset was used as the dataset. As a result of the study, 55.7% mAP result was obtained. As a result of the study, only a traffic light was detected [21]. Multi Backbone Network (MBBNet) is a different architecture used for traffic light detection. MBBNet consists of three co-convolution modules. These are normal, residual, and DenseNet highway modules. With the study conducted, an accuracy of 0.94 was achieved. The size of the work is 1.35 MB. With this small size, it has achieved low power consumption, high resolution, and 14 FPS speed [22].

Using an accurate data set to detect traffic lights may be appropriate sometimes. Obtaining an actual data set is high in terms of time and cost. More accurate data is needed for this dataset. Somewhat unrealistic computer graphics were used. In this way, the data set was fast and cost-effective. As a result of the study, it was compared with the studies obtained with the actual data set; in the study in which the data set was prepared using computer graphics, a 4% higher accuracy rate was obtained [23]. Traffic lights are always working. For this reason, an effort was made to detect traffic lights during the day and at night. He used color features to detect traffic lights. The data set used two types of traffic lights: those illuminated by traditional bulbs and LED lights. Bayesian methane was used to filter the captured images. As a result of the study, an accuracy of 99.4% was achieved [24]. The regression method is a different method of detecting traffic lights. In this method, they used images taken from cameras mounted on vehicles. With this image, they detected small traffic lights with a focus regression deep learning architecture consisting of an encoder-decoder. In their study, they used the Bosch small traffic lights and LISA large traffic lights dataset. The study conducted with a small data set reached a maximum accuracy rate of 42%. The study was conducted with a large data set, reaching a maximum accuracy rate of 49%. The larger the data set, the higher the accuracy rate [25]. Traffic lights are only sometimes located in flat places. It is not easy to detect, especially at level crossings. Traffic flow is very intense at level crossings. They worked on a portable system to detect traffic lights at this density. In their study, they used DTDNet-Lite architecture to detect traffic lights. ResNet18 was used to increase the accuracy rate in their studies. VOC 2007 and a customized data set were used as the data set [26]. Convolutional Neural Networks and Deep Learning can be used to detect traffic lights. It used YOLOV5 and AlexNet architectures for traffic light detection. It used YOLOV5 architecture to detect and identify traffic lights. They then used AlexNet architecture to classify images. They created their own data sets instead of ready-made ones in their study. The data set they created contained low-light images. They used the ZeroDCE low light enhancement algorithm on the data set for this. As a result of their study, they achieved an accuracy of 87.75% [27].

In literature studies, only traffic lights are detected. More than detecting traffic lights is required in autonomous driving. In addition to the traffic light, it is also necessary to determine which light is on. Thus, autonomous driving takes place. In addition, Bosch's small traffic light dataset was mainly used in literature studies. The LISA ample traffic light dataset is mainly used. Studies carried out with this data remain only in simulation. It could be more successful in real applications. In real life, traffic lights' locations, sizes, shapes, and sizes vary. The data set created for autonomous driving should include traffic lights in real life. The data set created for this was obtained from traffic lights used in real life. While creating the data set, the traffic lights' location and height were considered. Photos were taken from different angles. In addition, a data set was created by taking photographs at different light intensities to detect day and night. Eight different YOLO architectures were used to achieve the highest accuracy rate in the created data set. In addition to accuracy, the FPS rate must also be very high. When the detection process is real-time, the FPS rate must be high as well as accurate. As a result of the study, FPS was 60.

2. Materials and Methods

The study was carried out in three stages. In the first stage, the data set was prepared. For this purpose, real-life traffic lights were used. Thus, when the study is implemented, it will provide higher compliance in real life. The highest accuracy was achieved in the second stage using the data set in four different architectures. Since the work will be real-time, the FPS rate must also be excellent. In the last stage, it has been implemented in real life. Figure 1 shows the general structure of the study carried out to detect traffic lights.

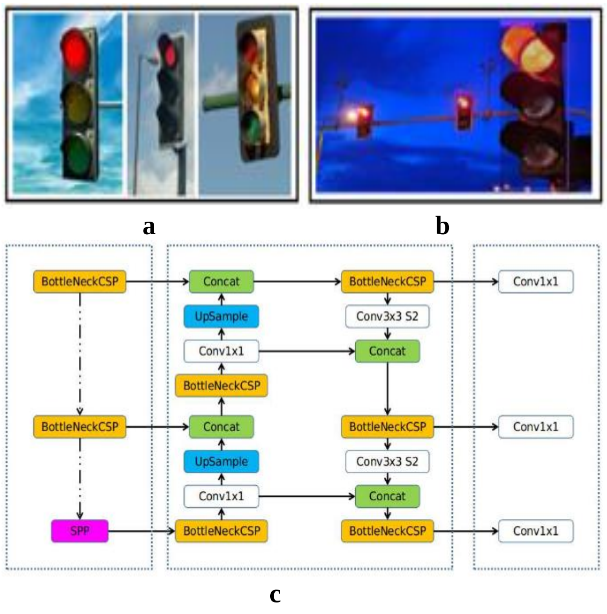

Figure 1. a) Preparation of the data set **b)** Study implementation in real life. **c)** Using the data set in eight different YOLO architectures [28].

2.1. Data Set

The first process to perform deep learning operations is to create the data set. Data set: Depending on the field in which it will be used, it may consist of mathematical expressions or visuals. Since traffic lights will be detected in real-time within the scope of this study, the data set consists of their images. It consists of an image matrix for digital processing. This matrix contains values between 0-255. With this data set, deep learning was carried out using the Yolo algorithm to find traffic lights in a photo, video, or real-time image. For this process, the data to be used must first be collected. Data collection was achieved through manual photography. There are three essential points to consider when collecting images.

Different locations: To find an object in a photo, video, or real-time image to be used in machine learning, the images of that object to be used in the data set must consist of images taken from different angles. This is important in minimizing erroneous object detection caused by an autonomous vehicle located on the right or left side of the road or at different heights when detecting traffic lights.

Different lighting conditions and levels: Machine learning, performed with a data set consisting of photographs taken into account, will facilitate object detection in images with light levels that will likely be encountered at different times of the day. Thanks to deep learning performed with a data set consisting of images taken at different distances, the vehicle can detect traffic lights from different distances.

Deep learning will be much more efficient because it addresses these issues. The data must be labeled after the collection process is completed. The labeling process enables marking the parts of the image containing meaningful pixels belonging to the desired object or person. In the study, red, yellow, and green lights of traffic lights were labeled. Many changes were made to the images that make up the data set, thus increasing their compatibility with the natural environment. The work carried out will be detected in real time. This will be done with images taken from the camera. An error that may occur in the camera will make detection difficult. For this purpose, all errors that may occur in the camera were applied to the photographs in the data set. Figure 2a, 25% variability has been added to the positioning and sizing of the images that comprise the database. Thus, this process was done with the help of the model being more durable, depending on the camera position. In Figure 2b, the images in the database are grayscale. In Figure 2c, +15% and -15% slope has been added to the images. In Figure 2d, random Gaussian blurring was performed to be more resistant to camera focus. In Figure 2e, +15% and -15% variability has been added to the rotations to be more resistant to camera roll. In Figure 2f, noise has been added to make it more robust against camera artifacts. In Figure 2g, +15% and -15% changes were made to the image brightness to make the model resistant to lighting and camera changes. In Figure 2h, the vividness of the colors in the images is randomly adjusted. In Figure 2i, perspective variability has been added to make it more resistant to camera, subject curtain, and aberration.

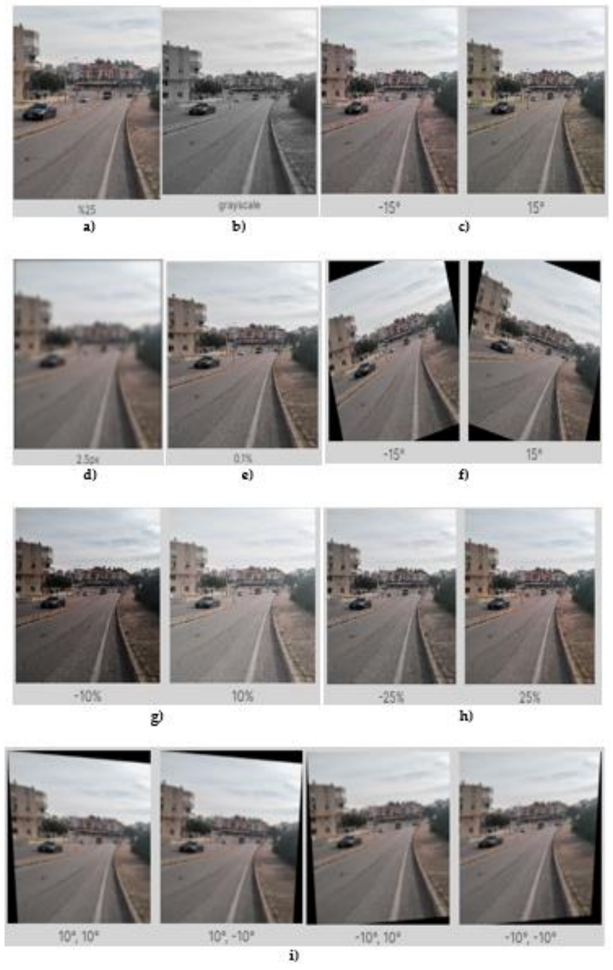

Figure 2a) Sizing and position change, **b)**Grayscale, **c)**Adding slope, **d)**Blurring, **e)**Adding variability, **f)** Adding noise, **g)**Image brightness change, **h)**Colour vividness change, **i)**Perspective change.

2.2. YOLO Architecture

As its working principle, YOLO transforms the detection process into a regression process. Unlike Fast R-CNN, a traditional method, YOLO does not require possible areas where the object can be found to detect an object. Regression allows bounding box coordinates and probabilities to be created simultaneously for each class in the image. Instead of processing the image separately for each class, the image is looked at only once. After viewing the image, bounding box coordinates and probabilities for all classes are created. In this way, learning which objects are in the image and where precisely the objects are in the image is achieved very quickly. This significantly shortens the detection time compared to traditional object detection systems. Figure 3 shows the general structure of the YOLO architecture.

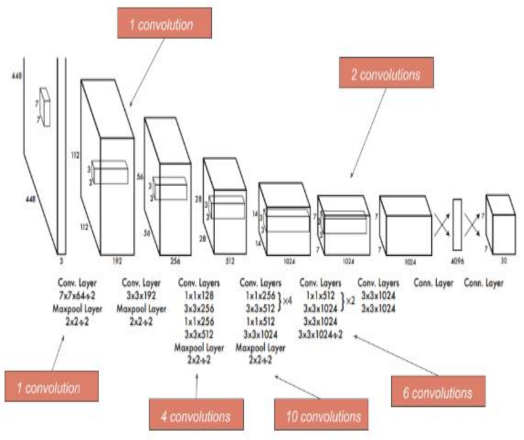
1 convolution
2 convolutions
Conv. Layer 7x7x64-s-2 Maxpool Layer 2x2-s-2
Conv. Layer 3x3x192 Maxpool Layer 2x2-s-2
Conv. Layers 1x1x128 3x3x256 1x1x256 3x3x512 Maxpool Layer 2x2-s-2
Conv. Layers 1x1x256 3x3x512 ×4 1x1x512 3x3x1024 Maxpool Layer 2x2-s-2
Conv. Layers 1x1x512 3x3x1024 ×2 3x3x1024 3x3x1024-s-2
Conv. Layers 3x3x1024 3x3x1024
Conn. Layer
Conn. Layer
1 convolution
4 convolutions
10 convolutions
6 convolutions

Figure 3. YOLO architecture

People can perform vision operations quickly and accurately. He can easily distinguish objects by looking at images. Using high algorithms makes the object detection process fast, and a high accuracy rate is achieved. In traditional object detection algorithms, an extra classification process is performed to classify the detected object. Algorithms first identify areas in an image where objects are likely to be found. Convolutional neural networks designed as classifiers in determined areas are run separately for each region to detect objects. Although the systems produced provide good results, the number of parameters and the required processing power increase because the image is processed in two separate processes. For this reason, it is impossible to use the systems in real-time operating systems. The R-CNN algorithm, one of the newest approaches designed in this way, first creates potential bounding boxes that can be found in an image. Then, region-based methods run a classifier on these proposed boxes. After classification, operations such as improving the confidence values of the generated bounding boxes, eliminating the detection of the same object multiple times, and reordering the boxes according to other boxes in the image are performed. Because each of these complex operations is trained separately, the system could be faster and easier to optimize.

YOLO has brought new insight into object detection. There are bounding boxes in YOLO. This method is not available in traditional object detection methods. Additionally, he treated all transactions as a single regression problem. YOLO detects objects by looking at the image once. In this way, detection occurs very quickly. Multiple bounding boxes are predicted simultaneously with a single convolutional network. Class probabilities are estimated for each class with a single convolutional network. It has various benefits over traditional object detection methods with a single convolutional network model.

3. Results and Discussion

Probability model systems are examined under four headings. These are true positive (TP), false positive (FP), true negative (TN), and false negative (FN) values. These occur depending on whether the tag value of the tagged data and the prediction result made by the system match or not. Assuming that the labeled data is positive, the prediction result is TP if it is positive and FN if it is negative. Assuming that the labeled data is negative, the prediction result is FP if it is positive and TN if it is negative.

Precision: Calculated as the ratio of predicted samples of true positive results to all predicted positive results. The formula for the precision value can be seen in equation 3.1.

$$\mathrm{P} = \frac{\mathrm{DP}}{\mathrm{DP+YP}} \qquad \textbf{(3.1)}$$

Sensitivity: Calculate the ratio of true positive results in predicted samples to all positive results that should occur. The formula for the sensitivity value is shown in equation 3.2.

$$\mathrm{P} = \frac{\mathrm{DP}}{\mathrm{DP+YP}} \qquad \textbf{(3.2)}$$

F1-Score: It is calculated as the harmonic mean of the values given above. The formula for the F-1 score is shown in equation 3.3.

$$\mathrm{F1\ Skoru} = 2 \times \frac{\mathrm{P} \times \mathrm{R}}{\mathrm{P+R}} \qquad \textbf{(3.3)}$$

Mean Average Precision (mAP): It is a metric system designed to evaluate values such as precision, sensitivity, F-1 score from a single point. The formula of the mAP value is shown in equation 3.4.

$$\sum_{i=0}^{S^2} 1_{ij}^{obj} \sum_{c\ E\ classes} \left(Ai\ (C) - \hat{a}\hat{i}(C)\right)^2 \qquad \textbf{(3.4)}$$

In the YOLO model, each cell produces more than one bounding box. In actual positive cases, the box with the highest IoU value among the generated bounding boxes should be selected to calculate the attenuation value. In this way, bounding boxes with low IoU values will be eliminated in future generations, and bounding boxes with higher IoU values will be produced. To calculate the attenuation value, YOLO takes the sum of the square errors of the class, localization, and confidence values between the bounding boxes, which are the predicted and the actual reference values. These values are called loss of classification, loss of localization, and loss of trust.

When an object is detected at classification loss, the square error of the conditional class probabilities is calculated for each class. It is possible to see the classification loss equation in Equation 3.5.

$$mAP = \int_0^1 P(R)dR \qquad \textbf{(3.5)}$$

1_i^{obj} = If there is an object in the cell, it is 1, otherwise it is 0.
$Ai\ (C)$ = It means the conditional class probability for class c in cell i.

The localization loss calculates the loss value related to the location and size of the estimated bounding box. It is calculated only for the box detecting the object. It is possible to see the localization loss equation in Equation 3.6.

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(Ei - \hat{E}i\right)^2 + (yi - \hat{y}i)^2 \right]$$

$$+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(\sqrt{wi} - \sqrt{\hat{w}i}\right)^2 + \left(\sqrt{hi} - \sqrt{\hat{h}i}\right)^2 \right] \qquad \textbf{(3.6)}$$

1_{ij}^{obj} = If the bounding box j in cell i is responsible for object detection, it is 1, otherwise it is 0.
λ_{coord} = It increases the weight of the loss value in the bounding box coordinates.

Confidence loss calculates the confidence value of the detecting box when the object is detected. It is possible to see the loss of trust equation in Equation 3.7 and 3.8.

$$\sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} (Ci - \hat{C}i)^2 \qquad \textbf{(3.7)}$$

$\hat{C}i$ = is the confidence value of box j in cell
1_{ij}^{obj} = If the bounding box j in cell i is responsible for object detection, it is 1, otherwise it is 0.

Loss of trust if no object is detected in the box:

$$\lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} (Ci - \hat{C}i)^2 \qquad \textbf{(3.8)}$$

$\hat{C}i$ = is the confidence value of box j in cell i.
1_{ij}^{noobj} = 1_{ij}^{obj} is the complement value of.
λ_{noobj} = It is used to reduce loss when background is detected.

Final loss value; It is the sum of classification, localization and loss of confidence values.

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(Ei - \hat{E}i\right)^2 + (yi - \hat{y}i)^2 \right] + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(\sqrt{wi} - \sqrt{\hat{w}i}\right)^2 + \left(\sqrt{hi} - \sqrt{\hat{h}i}\right)^2 \right] + \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left(Ci - \hat{C}i\right)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left(Ci - \hat{C}i\right)^2 + \sum_{i=0}^{S^2} 1_{ij}^{obj} \sum_{c\ E\ classes} (Ai\ (C) - \hat{a}i\hat{}(C))^2 \qquad \textbf{(3.9)}$$

3.1. YOLOv8 Architecture Results

Firstly, the dataset YOLOV8 architecture was used in the study. The training period for identifying the red, yellow, and green lights in traffic lights to the YOLOV8 architecture lasted 3 hours, 7 minutes, and 13 seconds. The highest accuracy rate for traffic lights was 99.1% in red lights, 98.6% in yellow lights, and 98.6% in green lights. The average accuracy rate was 98.3%. Table 1 shows the results occurring in the YOLOv8 architecture.

Table 1. Results of YOLOv8 architecture

Class	Precision	Recall	F1-Score	mAP@.5
All	0.983	0.903	0.939	0.977
Red	0.991	0.916	0.943	0.975
Yellow	0.986	0.893	0.937	0.980
Green	0.972	0.910	0.939	0.971

The training process was carried out using the data set consisting of red, yellow, and green lights in the YOLOV8 architecture. The performance values of YOLOV8 architecture are Precision 98.3%, Recall 90.3%, F1-Score 0.939, mAP@.5 0.977, mAP@.5:95:0.788. As a result of the YOLOV8 architecture, the complexity matrix consists of two parts: the predicted value and the actual value. Figure 4 shows the complexity matrix realized in the system.

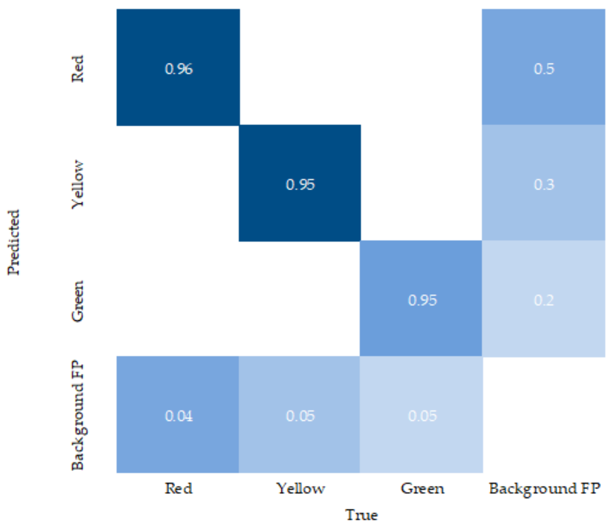

Figure 4. Complexity matrix resulting from YOLOV8 architecture

3.2. Other Architecture Results

In addition to the YOLOV8 architecture, YOLOV5, YOLOV6, and YOLOV7 architectures were also used in the study. The same data set was used in YOLOV5, YOLOV6, and YOLOV7 architectures. In this way, the error rate was tried to be minimized. Different mathematical algorithms were used in YOLOV5, YOLOV6, and YOLOV7 architectures. For this reason, the results are different. Table 2 shows the average values of YOLOV5, YOLOV6, and YOLOV7 architectures.

Table 2. Average results of YOLOV5, YOLOV6, and YOLOV7 architectures.

Class	Precision	Recall	F1-Score	mAP@.5
YOLOV5	0.920	0.939	0.932	0.971
YOLOV6	0.68	0.935	0.962	0.985
YOLOV7	0.911	0.899	0.911	0.949

Even though the same data set is used in YOLO architectures, different results are obtained. The biggest reason is that different mathematical algorithms are used in each architecture. This varies depending on the size or smallness of the data set. However, the most essential feature is the usage area.

When the study examined the accuracy, the highest accuracy was obtained at 98.30% in the YOLOv8 architecture. The lowest accuracy was in the YOLOv6 architecture. The second and third accuracy rates were achieved in YOLOv5 and YOLOv7. When the F1 scores are examined, the highest rate occurred in the YOLOv6 architecture. It was later implemented in YOLOv8, YOLOv5 and YOLOv7 architectures.

All architectures used in the system used the same number of photos and photo sizes. As a result of the study, the highest accuracy rate was 0.98 for the YOLOv8 architecture. These values are average values. There are different accuracy rates for red, yellow, and green traffic lights. The reason for this is the differences between the lights. Average values were taken in the study. An accuracy rate of 0.98 was achieved in the YOLOv8 architecture. The lowest accuracy rates were 0.68 in the YOLOv6 architecture. It was 0.920 and 0.911 in YOLOv5, and YOLOv7 architectures. The accuracy rates of YOLOv5 and YO-LOv8 architectures are close to each other. Figure 5 shows one result of the operation of the system.

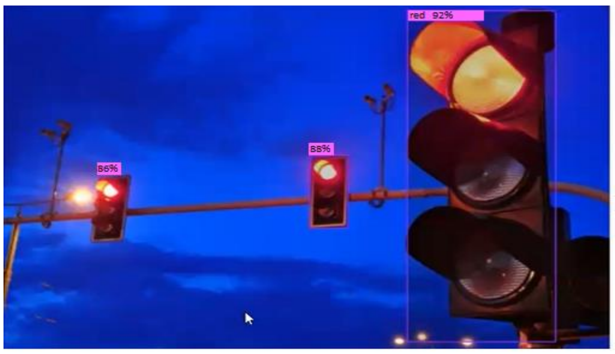

Figure 5. One result of the operation of the system.

Since the work will be real-time, the FPS speed must be sufficient. In the YOLOv8 architecture, which gives the highest accuracy rate, the speed has increased to 60 FPS. At this rate, it is sufficient for real-time detection. For this reason, a camera has been placed inside the vehicle. Real-time detection was carried out with images taken from the camera.

When the literature studies are examined, many studies have been done. In these studies, ready-made data sets were only used. In some studies, the most appropriate data set for the system was prepared. Table 3 shows the comparison of literature studies.

Table 3. Comparison of literature studies.

Writer	Number of Image	Architectural	Precision
Xie et all. [29]	7000	Faster R-CNN	%97.20
Boyuk et all. [30]	8726	Faster R-CNN	%75.16
Kamran et all. [31]	1586	Faster R-CNN	%60.72
Gupta et all. [32]	6772	YOLOv3-Tiny	%77.00
Kyrkou et all. [33]	350	Dronet	%95.00
Sun et all. [34]	32	YOLOv5	%93.30
Yong et all. [35]	755	GoogleNet	%73.01
Mansour et all. [36]	397	Faster R-CNN	%89.21
Bayram et all. [37]	3008	YOLOv6	%98.30
Galayol et all. [38]		YOLOv7	%93.42
Hassan et all. [39]		YOLOv6	%55.7
Ngoc et all. [40]		YOLOv8	%98.50
Gao et all. [41]		YOLOv4	%80.00
Sarhan et all. [42]	90	YOLOv3	%76.00
Omar et all. [43]		YOLOv3	%67.21
Nui et all. [44]		YOLOv5	%87.75

As a result of the literature review, it is understood that the use of previously made systems in daily life is quite difficult. These difficulties have low fps value. For this reason, the systems cannot be used in real-time. The systems only stopped at traffic lights in certain places. In this case, different traffic lights cause the systems to be detected at a very low rate or not at all. The data sets used in the studies consist of photographs taken at a certain time of the day. In real life, traffic lights are encountered at different times of the day.

Four of the latest versions of the YOLO architecture were used in the study. In this way, the highest accuracy and fps rates were determined by comparison. Since YOLO architectures only look at the image once and perform the detection process, their frame rates are excellent. A speed of 60 fps was achieved in the YOLOv8 algorithm. In this way, detection can be done very quickly in real-time.

Since the study will be used in real life, the data set must be prepared very well. For this purpose, the data set was prepared according to three important rules. These include different locations, lighting conditions, and levels, and finally, different distances. Examples were taken from traffic lights in different locations in real life. Images of traffic lights were taken under different lighting conditions and levels. Images were taken at different times of the day. Finally, images were taken from different distances. Thanks to these images, a data set was created. The more suitable the data set is to real life, the more accurate the detection process becomes.

4. Conclusion

Autonomous devices are increasing in our lives day by day. The most important of these is that electric vehicles are autonomous. For electric vehicles to be autonomous, detecting traffic lights will not be enough. In order for the vehicle to move, it is necessary to determine which traffic light is on. The study carried out the detection of red, yellow, and green lights of traffic lights. For this, YOLOv8, YOLOv7, YOLOv6, and YOLOv5 architectures were used. An accuracy of 98.3% was achieved in the YOLOv8 architecture. A speed of 60 fps was achieved in the YOLOv8 architecture. Thus, real-time detection was made. The data set was prepared with images obtained from real traffic lights. These images were obtained at traffic lights of different shapes. In this way, the detection of red, yellow, and green lights in real-life traffic lights has been achieved at a very high rate.

The locations, shapes, and scales of traffic lights are very different. Traffic lights are difficult to detect in real-world conditions due to their similarity to other objects. In addition to these features, detecting traffic lights is not sufficient in autonomous driving. It is necessary to determine which of the red, yellow, and green lights is on at traffic lights. In the study, red, yellow, and green lights on traffic lights were detected in real-time. With the study conducted, traffic lights were detected in many different ways in real life with high accuracy. For autonomous driving, detection alone is not enough. The detection process needs to be adapted to the autonomous system. First of all, a warning system should be added to the driver. The driver must be warned audibly and visually. In the next work, the vehicle should be enabled to switch to autonomous driving and brake automatically. In this way, human error will be minimized.

Ethics

There are no ethical issues after the publication of this manuscript.

References

[1]. Diaz-Cabrera, M, Cerri, P, Medici, P. 2015. Robust real-time traffic light detection and distance estimation using a single camera. *Expert. Syst. Appl;* 42, 3911–3923. https://doi.org/10.1016/j.eswa.2014.12.037

[2]. Hosseinyalamdary, S, Yilmaz, A. 2017. A Bayesian approach to traffic light detection and mapping, ISPRS J. Photograms. *Remote Sens;* 125, 184–192. https://doi.org/10.1016/j.isprsjprs.2017.01.008

[3]. Li, X, Ma, H, Wang, X, Zhang, X. 2018. Traffic light recognition for complex scene with fusion detections. *IEEE Trans. Intell. Transp. Syst;* 19, 199–208. https://doi.org/10.1109/TITS.2017.2749971

[4]. Boloor, A, Garimella, K, He, K, Gill, C, Vorobeychik, Y, Zhang, X. 2020. Attacking vi- sion-based perception in end-to-end autonomous driving models. *J. Syst. Archit;* 101766. . https://doi.org/10.1016/j.sysarc.2020.101766

[5]. Jensen, M.P, Philipsen, M.P, Møgelmose, A, Moeslund, T.B, Trivedi, M.M. 2016. Vision for looking at traffic lights: issues, survey, and perspectives. *IEEE Trans. Intell. Transp. Syst;* 17, 7, 1800–1815. https://doi.org/10.1109/TITS.2015.2509509

[6]. Ouyang, Z, Niu, J, Liu, Y, Guizani, M. 2019. Deep cnn-based real-time traffic light detector for self-driving vehicles. *IEEE Trans. Mob. Comput;* 19, 2, 300–313. https://doi.org/10.1109/TMC.2019.2892451

[7]. Kim, J, Cho, H, Hwangbo, M, Choi, J, Canny, J, Kwon, Y.P. Deep traffic light detection for self-driving cars from a large-scale dataset, in: 2018. *21st International Conference on Intelligent Transportation Systems (ITSC*), Maui, Hawaii, USA, 4-7 November 2018, pp. 280–285. https://doi.org/10.1109/ITSC.2018.8569575

[8]. Jensen, M, Philipsen, M, Møgelmose, A, Moeslund, T, Trivedi, M. 2016. Vision for looking at traffic lights: issues, survey, and perspectives. *IEEE Trans. ITS;* 17, 7, 1800–1815. https://doi.org/10.1109/TITS.2015.2509509

[9]. Behrendt, K, Novak, L, Botros, R. 2017. A deep learning approach to traffic lights: detection, tracking, and classification. *IEEE ICRA*; pp. 1370–1377. https://doi.org/10.1109/ICRA.2017.7989163

[10]. Sommer, L.W, Schuchert, T, Beyerer, J. Fast deep vehicle detection in aerial images. *IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE*; Santa Rose, California, USA. 24-31 March 2017. pp. 311–319. https://doi.org/10.1109/WACV.2017.41.

[11]. Zhang, X, Story, B, Rajan, D. 2021. Night time vehicle detection and tracking by fusing vehicle parts from multiple cameras. *IEEE Transa. Intelligent Transp. Syst*; pp. 258-265. http://dx.doi.org/10.1109/TITS.2021.3076406

[12]. Chen, C, Liu, B, Wan, S, Qiao, P, Pei, Q. 2021. An edge traffic flow detection scheme based on deep learning in an intelligent transportation system. *IEEE Trans. Intelligent Trans. Syst;* 22, 3, 1840–1852. https://doi.org/10.1155/2021/5583874

[13]. Zhang, Z, Zaman, A, Xu, J, Liu, X. 2022. Artificial intelligence-aided railroad trespassing detection and data analytics: methodology and a case study. *Accident Anal. Prevention*; 168, 106594. https://doi.org/10.1016/j.aap.2022.106594

[14]. Tsai, L.W, Hsieh, J.W, Fan, K.C. 2007. Vehicle detection using normalized color and edge map. *IEEE Trans. Image Process;* 16, 3, 850–864. http://dx.doi.org/10.1109/tip.2007.891147

[15]. Ehtesham, H, Yasser, K, Imtiaz, A. 2023. Learning deep feature fusion for traffic light detection. *Journal of Engineering Research;* 11, 94–99. http://dx.doi.org/10.1155/2020/7286187

[16]. Zhenchao, O, Jianwei, N, Tao, R, Yanqi, L, Jiahe, C, Jiyan, W. 2020. MBBNet: An edge IoT computing-based traffic light detection solution for autonomous bus. *Journal of Systems Architecture*; 109, 101835. http://dx.doi.org/10.1016/j.sysarc.2020.101835

[17]. Jean, P, V, M, Lucas, T, Rodrigo, F, B, Thiago, M, Alberto, F, S, Claudine, B, Nicu, S, Thiago, O, S. 2021. Deep traffic light detection by overlaying synthetic context on arbitrary natural images. *Computers & Graphics;* 94, 76–86. https://doi.org/10.48550/arXiv.2011.03841

[18]. Moises, D.C, Pietro, C, Paolo, M. 2015. Robust real-time traffic light detection and distance estimation using a single camera. *Expert Systems with Applications;* 42, 3911–3923. https://doi.org/10.1016/j.eswa.2014.12.037

[19]. Eunseop, L, Daijin, L. 2019 Accurate traffic light detection using deep neural network with focal regression loss. *Image and Vision Computing*, 87, 24–36. https://doi.org/10.1016/j.imavis.2019.04.003

[20]. Feng, G, Yi, W, Yu, Q. 2023. Real-time dense traffic detection using lightweight backbone and improved path aggregation feature pyramid network. *Journal of Industrial Information Integration;* 31, 100427. https://doi.org/10.1016/j.jii.2022.100427

[21]. Chuanxi, N, Kexin, L. 2022. Traffic Light Detection and Recognition Method Based on YOLOv5s and AlexNet. *Appl. Sci*; 12, 10808. https://doi.org/10.3390/app122110808

[22]. Lin, T,Y, Goyal, P, Girshick, R, He, K. Doll´ar, P. Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision*, Venice, Italy, 22-29 Oct. 2017, pp. 2980–2988. https://doi.org/10.1109/ICCV.2017.324

[23]. Liu, Y, Sun, P, Wergeles, N, Shang, Y. 2021. A survey and performance evaluation of deep learning methods for small object detection. *Expert Syst. Applied,* 172, 114602. https://doi.org/10.1016/j.eswa.2021.114602

[24]. Yu, W, Yang, T, Chen, C.Towards resolving the challenge of long-tail distribution in uav images for object detection, *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*; Waikoloa, HI, USA. 3-8 June 2021. pp. 3258–3267. https://doi.org/10.48550/arXiv.2011.03822

[25]. Sang, J, Wu, Z, Guo, P, Hu, H, Xiang, H, Zhang, Q, Cai, B. 2018. An improved YOLOv2 for vehicle detection. *Sensors;* 18, 12, 4272. https://doi.org/10.3390/s18124272

[26]. Zhang, F, Yang, F, Li, C, Yuan, G. 2019. CMNet: a connect-and-merge convolutional neural network for fast vehicle detection in urban traffic surveillance. *IEEE Access;* 7, 72660–72671. https://doi.org/10.1155/2021/5583874

[27]. Tan, M, Pang, R, Le, and Q.W. Efficientdet: scalable and efficient object detection. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*; Seattle, Washington, USA. 14-19 June 2020. pp. 10781–10790. https://doi.org/10.1109/CVPR42600.2020.01079

[28]. Xu, R, Lin, H, Lu, K, Cao, L, Liu, Y. 2021. A Forest Fire Detection System Based on Ensemble Learning. *MDPI Forests*, 12 (217), pp. 1568–1570. https://doi.org/10.3390/f12020217

[29]. Xie, X, He, C. Object detection of armored vehicles based on deep learning in battlefield environment. Proceedings - *2017 4th International Conference on Information Science and Control Engineering, ICISCE; Changsha, Chania, 21-23 July 2017.* 1568–1570. https://doi.org/10.1109/ICISCE.2017.327

[30]. Boyuk, M, Duvar, R, Urhan, O. Deep learning based vehicle detection with images taken from unmanned air vehicle. *Proceedings 2020 Innovations in Intelligent Systems and Applications Conference, ASYU; İstanbul, Türkiye, 15-17 October 2020. pp.175.* https://doi.org/10.1109/ASYU50717.2020.9259868

[31]. Kamran, F, Shahzad, M, Shafait, F. Automated military vehicle detection from low-altitude aerial images. *2018 International Conference on Digital Image Computing: Techniques and Applications, DICTA.* Canberra, Australia, 10-13 December 2018, 2 https://doi.org/10.1109/DICTA.2018.8615865

[32]. Gupta, P, Pareek, B, Singal, G, Rao, D. V. 2022. Edge device based military vehicle detection and classification from UAV. *Multimedia Tools and Applications*; 81(14), 19813–19834. https://doi.org/10.1007/S11042-021-11242-Y/FIGURES/12

[33]. Kyrkou, C, Plastiras, G, Theocharides, T, Venieris, S. I, Bouganis, C. S. 2018. DroNet: Efficient convolutional neural network detector for real-time UAV applications. *Proceedings of the 2018 Design, Automation and Test in Europe Conference and Exhibition*; 967–972. https://doi.org/10.23919/DATE.2018.8342149

[34]. Sun, Y, Wang, W, Zhang, Q, Ni, H, Zhang, X. 2022. Improved YOLOv5 with transformer for large scene military vehicle detection on SAR image. *2022 7th International Conference on Image, Vision and Computing, ICIVC;* 87–93. https://doi.org/10.1109/ICIVC55077.2022.9887095

[35]. Yong, S. P, Yeong, Y. C. 2018. Human object detection in forest with deep learning based on drone's vision. *2018 4th International Conference on Computer and Information Sciences: Revolutionizing Digital Landscape for Sustainable Smart Society, ICCOINS;* https://doi.org/10.1109/ICCOINS.2018.8510564

[36]. Mansour, A, Hassan, A, Hussein, W. M, Said, E. 2019. Automated vehicle detection in satellite images using deep learning. *IOP Conference Series: Materials Science and Engineering*; 610(1). https://doi.org/10.1088/1757-899X/610/1/012027

[37]. Bayram, A.F, Nabiyev, V. 2023. Detection of Hidden Camouflaged Tanks Based on Deep Learning: Comparative Analysis of the art YOLO Network. *Gümüşhane University of Journal of Science and Technology;* 182-193. https://doi.org/10.17714/gumusfenbil.1271208

[38]. Gelayol, G, Ignacio, M.A, Qi, W, Jose, M.A.C. 2023. Robust Real-Time Traffic Light Detection on Small-Form Platform for Autonomous Vehicles. *Journal of Intelligent Transportation System*; 1-11 https://doi.org/10.1080/15472450.2023.2205018

[39]. Hassan, E, Khalil, Y, Ahmad, I. 2023. Learning Deep Feature Fusion for Traffic Light Detection. *Journal of Engineering Research*; 11, 94-99. https://doi.org/10.1016/j.jer.2023.100128

[40]. Ngoc, H.T, Nguyen, K.H, Hua, H.K, Nguyen. H.V.N, Quach, L. 2023. Optimizing YOLO Performance for Traffic Light Detection and End-to-End Steering Control for Autonomous Vehicles in Gazebo-ROS2. *International Journal of Advanced Computer Science and Applications;* Vol.14. 7. https://doi.org/10.14569/IJACSA.2023.0140752

[41]. Gao, H, Wang, W, Yang, C, Jiao, W, Chen, Z, Zhang, T. 2021. Traffic Signal Image Detection Technology Based on YOLO. *Journal of Physics: Conference Series*; 1-21. https://doi.org/10.1088/1742-6596/1961/1/012012

[42]. Serhan, N.H, Olmary, A. Y. 2022. Traffic Light Detection Using Opencv and YOLO. *International Conference on Innovation and Intelligence for Informatics, Computing and Technologies*; 604-608. https://doi.org/ 10.1088/1742-6596/1961/1/012012

[43]. Omar, W, Lee, I., Lee, G, Park, K.M. 2020. Detection and Localization of Traffic Light Using YOLOv3 and Stereo Vision. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*; Vol.18, 1247-1252. https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-1247-2020

[44]. Nui, C, Li, K. 2022. Traffic Light Detection and Recognition Method Based on YOLOv5s and AlexNet, *Applied Science*; 12, 10808. https://doi.org/10.3390/app122110808