Sakarya University Journal of Science

ISSN : 2147-835X
Publisher : Sakarya University

Vol. 28, No.2, 418-430, 2024
DOI: https://doi.org/10.16984/saufenbilder.1393307

Research Article

Computer Vision-Based Lane Detection and Detection of Vehicle, Traffic Sign, Pedestrian Using YOLOv5

Gülyeter Öztürk[1*], Osman Eldoğan[1], Raşit Köker[2]

[1] Sakarya University of Applied Sciences, Faculty of Technology, Mechatronics Engineering, Sakarya, Türkiye, gulyeterozturk@subu.edu.tr, eldogan@subu.edu.tr
[2] Sakarya University of Applied Sciences, Faculty of Technology, Electrical and Electronics Engineering, Sakarya, Türkiye, rkoker@subu.edu.tr
[*]Corresponding Author

ARTICLE INFO

Keywords:
YOLOv5
Object Detection
Lane Detection
Computer Vision
Deep Learning

Article History:
Received: 20.11.2023
Accepted: 08.01.2024
Online Available: 24.04.2024

ABSTRACT

There has been a global increase in the number of vehicles in use, resulting in a higher occurrence of traffic accidents. Advancements in computer vision and deep learning enable vehicles to independently perceive and navigate their environment, making decisions that enhance road safety and reduce traffic accidents. Worldwide accidents can be prevented in both driver-operated and autonomous vehicles by detecting living and inanimate objects such as vehicles, pedestrians, animals, and traffic signs in the environment, as well as identifying lanes and obstacles. In our proposed system, road images are captured using a camera positioned behind the front windshield of the vehicle. Computer vision techniques are employed to detect straight or curved lanes in the captured images. The right and left lanes within the driving area of the vehicle are identified, and the drivable area of the vehicle is highlighted with a different color. To detect traffic signs, pedestrians, cars, and bicycles around the vehicle, we utilize the YOLOv5 model, which is based on Convolutional Neural Networks. We use a combination of study-specific images and the GRAZ dataset in our research. In the object detection study, which involves 10 different objects, we evaluate the performance of five different versions of the YOLOv5 model. Our evaluation metrics include precision, recall, precision-recall curves, F1 score, and mean average precision. The experimental results clearly demonstrate the effectiveness of our proposed lane detection and object detection method.

1. Introduction

In recent years, although it varies from country to country, there has been a rapid increase in the number of vehicles worldwide, leading to a rise in accidents each year. Many people have been injured or killed in these accidents, where human mistakes play a major role [1]. According to a survey conducted by the World Health Organization in 2018, road traffic accidents affect approximately 1.25 million people each year [2]. To reduce the number of traffic accidents, it is crucial for drivers to prioritize their attention on the road, maintain focus while driving, and adhere to traffic rules. Human distraction while driving can occur for various reasons. In such cases, Advanced Driver Assistance Systems (ADAS) and smart transportation systems can play a vital role in ensuring the safety and comfort of drivers, passengers, and pedestrians. These systems can proactively develop strategies to prevent accidents before they occur. The rapid growth of China's automotive industry, combined with advancements in deep learning technology, has led to an increasing number of vehicles equipped with assisted driving and autonomous driving functions. By the year 2020, more than 50% of

Cite as: Öztürk G., Eldoğan O., Köker R. (2024). Computer Vision-Based Lane Detection and Detection of Vehicle, Traffic Sign, Pedestrian Using YOLOv5, *Sakarya University Journal of Science,* 28(2), 418-430. https://doi.org/10.16984/saufenbilder.1393307

vehicles in the market were equipped with these capabilities [3]. This trend underscores the growing attention and significance of ADAS in recent years. ADAS assists vehicles by helping them maintain their designated lanes, detecting objects in their surroundings, and continuously analyzing and interpreting the environment to ensure traffic safety. These systems encompass modules for detecting traffic signs, traffic lights, and other objects on the road, as well as lane detection, lane tracking, collision avoidance, and more. Companies such as Google, Tesla, Audi, Mercedes, General Motors, and Ford are utilizing deep learning infrastructure to develop self-driving vehicles without drivers. These companies rely on these multifunctional modules for various aspects of autonomous driving.

Computer vision and deep learning technologies enable vehicles to comprehend their surroundings by detecting lanes and objects. Detecting road lanes helps keep the vehicles within drivable areas, and accidents during lane-changing processes can be prevented with the assistance of lane departure warning systems. Studies on lane detection have been conducted using various methods throughout the years, from the past to the present. Lane detection studies can generally be classified into four categories: (i) lane detection based on extracting road features through machine learning or computer vision, (ii) lane detection through the creation of road models that capture features on roads with specific templates, (iii) lane detection achieved by employing a multi-sensor fusion detection method using GPS, radar, high-resolution cameras, and other fusion techniques, and (iv) lane detection enhanced by leveraging deep learning technologies [4].

Object detection is a task that involves predicting the locations of desired living or inanimate objects in an image or video and classifying them into different categories. In object detection studies, the features of the objects you want to detect are obtained through machine learning algorithms, which are then given to a neural network for classification. In deep learning methods, objects' features are acquired using convolutional kernels of various sizes in convolutional layers, and the classification process takes place in the final layer of the same model. Object detection models based on deep learning can be divided into two main categories: two-stage and one-stage object detection models. Region-based object detection models, such as Region-Based Convolutional Neural Network (R-CNN), Fast R-CNN, and Faster R-CNN, initially identify regions that have the potential to contain the object and then perform the classification process within these regions. Due to their two-stage detection process, this results in an increase in computation for the model and a decrease in frames per second (FPS). The You Only Look Once (YOLO), Single Shot MultiBox Detector (SSD), and Retina-Net models process the input image through CNN all at once to obtain the classes and coordinates of all objects in the image. In this paper, we present a study aimed at assisting drivers and self-driving vehicles in avoiding traffic mistakes. Our research focuses on lane detection and the identification of pedestrians, vehicles, and traffic signs on the road using five different versions of the YOLOv5 model, each trained with varying batch and input sizes. Our goal is to prevent accidents on the road and enhance overall road safety.

The remaining sections of the paper are organized as follows: Section 2 provides an overview of the literature related to lane detection and object detection. Section 3 explains the methodology used for detecting both straight and curved lanes, as well as objects on the road. Section 4 presents the experimental results of our proposed method. Finally, Section 5 concludes our study and offers suggestions for future research.

2. Related Work

Numerous studies have been conducted to enhance driving within a lane and ensure road safety by detecting objects on the road. These studies encompass lane detection, traffic sign detection, vehicle detection, and pedestrian detection. They hold significant importance in both computer vision and deep learning technologies. These studies have been conducted independently for traffic sign detection, pedestrian detection, vehicle detection, and lane detection. Additionally, research has been undertaken to combine multiple detection techniques, such as detecting both vehicles and pedestrians or detecting lanes and vehicles. In this section, we present a review of the existing literature on lane detection and detection of object on the road.

Lane markings, usually yellow or white in color, can take the form of straight or curved lines. In lane detection studies, common techniques include Gaussian filtering, Kalman filtering [5], Canny edge detection [6], and the Hough transform [7]. Recent years have witnessed a growing trend in lane detection studies utilizing deep learning approaches [8]. Furthermore, various camera-based lane detection studies have been conducted by researchers [9-11].

Kumar and his colleagues employed Hough transform optimization to enhance the accuracy of detecting both straight and curved lanes on the highway. They also utilized a Kalman filter to track the detected lanes. They compared the detection performance of straight and curved lanes based on specific metrics [12]. Dubey and Bhurchandi utilized the Hough transform and Gaussian filter for lane detection in their work [13]. Huang et al. employed inverse perspective transformation and the Kalman filter for lane detection and lane line tracking [14].

Muthalagu and his team developed a lane detection algorithm for autonomous cars, creating a Convolutional Neural Network (CNN)-based model that learns to drive from the driver's driving data [1]. This model acquires its learning by capturing data from the car's onboard cameras. The developed system underwent performance evaluation for an autonomous vehicle application capable of detecting stop signs and other vehicles. Ji and Zheng utilized the improved YOLOv3 algorithm for lane line detection, achieving better results in terms of detection accuracy and FPS compared to the original YOLOv3 algorithm [4].

Traffic signs can vary from country to country, consisting of various colors such as red, blue, white, yellow, black, and green, as well as simple shapes like triangles, rectangles, circles, polygons, and more. Shustanov and Yakimov conducted an analysis on a dataset of German traffic signs using several CNN architectures, comparing their performance in detecting and identifying traffic signs [15]. Kilic and Aydin carried out detection and recognition experiments on Turkey's 41 traffic signs using an Nvidia GTX 1080 Ti graphics card in their studies [16].

Wang et al. conducted research on object detection in the field of autonomous driving technology. In their object detection studies, they modified the structure of the YOLOv4 model to achieve high speed and accuracy in detecting both large and small objects. It was found that these modifications led to improvements in both models' performance [17]. While lane detection, pedestrian detection, traffic sign detection, and vehicle detection studies are often conducted separately, there are also studies that combine multiple detection tasks. Yang et al., using the KITTI dataset, performed pedestrian and vehicle detection with Faster R-CNN, YOLO v2, and their proposed YOLOv2-based model, providing a comparison of the accuracy and FPS of these detection models [18].

Ćorović et al. employed the YOLOv3 model for real-time detection of cars, trucks, pedestrians, traffic signs, and traffic lights. Their training, conducted up to 120 epochs, considered precision, recall, mean average precision (mAP), and average IoU (Intersection over Union) metrics [19]. Ozturk et al. conducted vehicle, pedestrian, and traffic sign detection using four different CNN models, comparing the mAP metrics of these models in detecting objects of various sizes [20]. Kemsaram and colleagues proposed a pipeline on the Nvidia Drive PX 2 platform, enabling real-time object detection, lane detection, and free space detection simultaneously [21].

3. Method and Experiment

In this study, with the aim of ensuring proper lane keeping, we conducted a computer vision-based detection process to identify straight or curved lanes within the vehicle's surroundings. We utilized the YOLOv5 model, renowned for its performance and speed in object detection studies, to analyze the surroundings by detecting traffic signs, pedestrians, and vehicles on the road.

3.1. Computer vision-based lane detection

The flow diagram of the lane detection algorithm, which displays the drivable areas in front of vehicles and helps prevent lane violations, ultimately aiming to reduce potential accidents, is depicted in Figure 1.

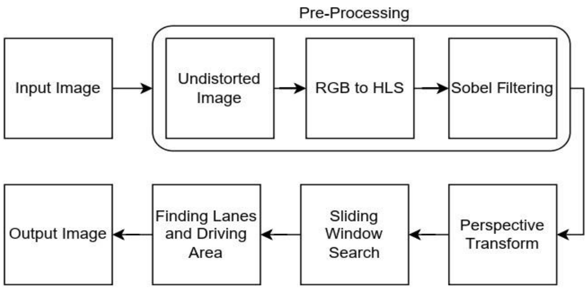

Figure 1. Flow diagram of the lane detection

Due to the camera lenses, objects in the raw image captured by the camera may appear closer or farther than they actually are. These distortions can lead to incorrect measurements in computer vision applications, affecting the size and shape of objects. To address this issue, camera calibration was performed using a chessboard pattern image.

When attempting color filtering on the raw image in the RGB color space to extract both yellow and white lanes, finding a suitable threshold value proved challenging. To reduce noise in both colors and enhance their visibility, thereby improving the detection of lane markings, we transformed the undistorted image from the RGB color space to the HLS color space. In the H (hue), L (lightness), and S (saturation) channels of the HLS color space, we determined lower and upper threshold values for both yellow and white colors.

After color filtering, the 3-channel, 24-bit color image was converted to a single-channel, 8-bit grayscale image. In the grayscale image, pixels take on values ranging from 0 to 255, representing different shades of gray from black (0) to white (255).

To detect edges in the grayscale image, we perform gradient detection to identify areas with high contrast. Figure 2 illustrates the changes in the x and y directions and shows low and high gradients. To detect high gradients, the Sobel filter was applied.

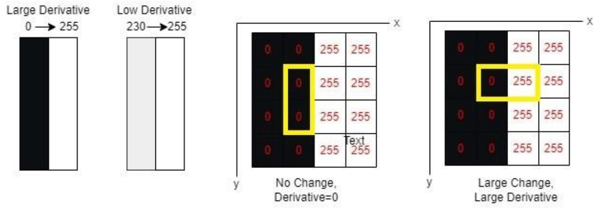

Figure 2. Gradient change in pixels

Using the Sobel filter, the gradient (Gx) on the horizontal axis and the gradient (Gy) on the vertical axis are obtained. Thus, for each pixel in the image, the edge gradient magnitude (G) and its direction (θ) are calculated as shown in Equations 1 and 2.

$$G = \sqrt{{G_x}^2 + {G_y}^2} \tag{1}$$

$$\theta = tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{2}$$

The 3x3 filters used in Sobel edge detection for detecting edges in the horizontal and vertical directions are shown in Figure 3.

-1	0	+1
-2	0	+2
-1	0	+1

G_x

+1	+2	+1
0	0	0
-1	-2	-1

G_y

Figure 3. Filter used in detecting horizontal and vertical edges

Detecting curved lanes in the camera field of view can be challenging. To resolve this difficulty, a bird's-eye view of the road obtained with the use of perspective transformation. For this purpose, coordinates are determined to obtain a bird's eye view of the lanes as shown in Figure 4.

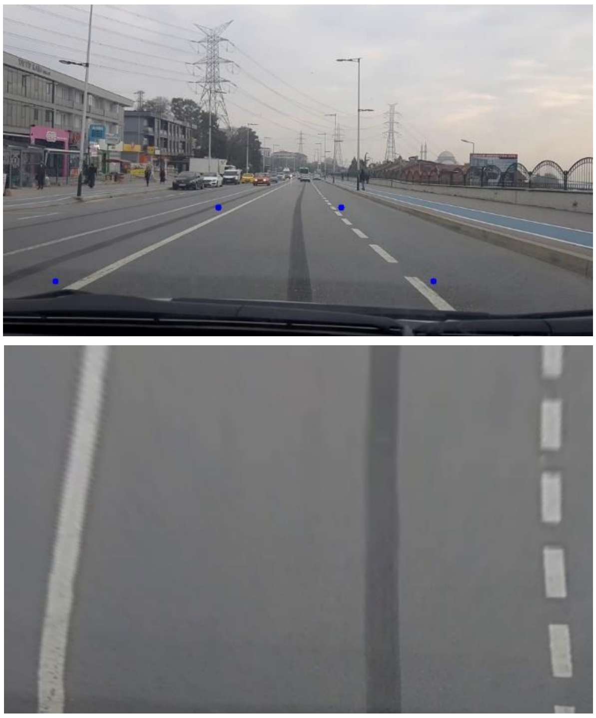

Figure 4. Obtaining bird's eye view from the image

To identify the starting points of the left and right lane markings in the bird's-eye view, we begin by calculating a histogram along the x-axis at the bottom of the image to find the two highest pixel values. We then create a window of defined width and height centered on these points. Subsequently, we employ a sliding window search method, using windows of the same size, to iteratively center on the highest pixel value from the bottom to the top of the image. The x and y coordinates of the highest value pixels detected using this method are used to generate a second-degree polynomial curve, as described in Equation 3. Figure 5 shows the sliding window search method and the polynomial curve.

$$f(y) = ay^2 + by + c = 0 \tag{3}$$

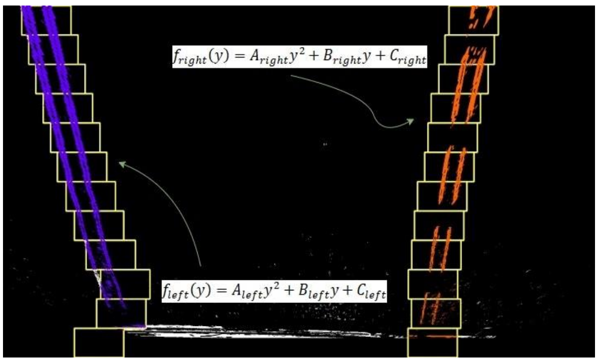

Figure 5. The sliding window search method and polynomial curve

3.2. YOLOv5 object detection model

YOLO treats object detection as a regression problem, reducing computational complexity while maintaining real-time detection with high accuracy. YOLO models with CNN structures process the input image just once to detect objects, determine bounding boxes, classifications, and confidence scores all in a single step. The first version of the YOLO family, YOLOv1, consisting of 24 convolution layers, was released by Joseph Redmon and his research team in May 2016. YOLOv1 takes raw input images sized at 448x448 pixels with three color channels.

The model divides the input image into SxS grids, and the grid containing the center point of detected objects is responsible for estimating the object's class probability, bounding box values, and confidence score. In December 2016, Redmon and Farhadi introduced YOLOv2, which employs the Darknet-19 network with 19 convolution layers for feature extraction. Subsequently, they released YOLOv3 in 2018, featuring the Darknet-53 base network with 53 convolution layers. YOLOv4 introduced in 2020 by Alexey Bochkovskiy and his team. Each new model modifies the structure of the previous versions to improve the performance of the model in object detection. Approximately two months after YOLOv4's release, Glenn Jocher introduced YOLOv5 [22]. Notably, YOLOv5 utilized PyTorch instead of Darknet, which had been used in previous YOLO versions.

YOLOv5 offers five different versions, each with varying network layer depths and processing density. These versions are denoted as YOLO v5n, YOLO v5s, YOLO v5m, YOLO v5l, and YOLO v5x. Figure 6 presents a comparison of the speed and AP values for these models. YOLOv5 versions have different depths, the computational load increases as the depth goes up. YOLO v5n stands out as the fastest version, thanks to its lower depth, although it exhibits the lowest AP value. Conversely, YOLO v5x boasts the highest depth and, as a result, performs more computationally intensive calculations.

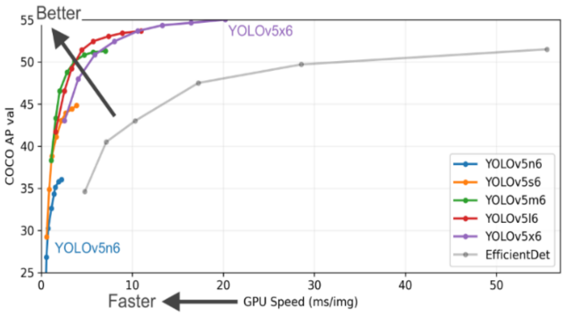

Figure 6. Speed and AP comparison of YOLOv5 versions [22]

The YOLOv5 architecture is divided into three sections: the backbone, the neck, and the head. The backbone primarily extracts key features from the input image. Feature pyramids obtained in the neck allow for the definition of objects at different sizes and scales. In the head part of the model, class probabilities, objectness scores, and bounding boxes of objects are obtained. As shown in Figure 7, YOLOv5 mainly consists of CBL, SPP (Spatial Pyramid Pooling), Upsample, and Concat modules used in YOLOv3 and YOLOv4 models, along with Focus and CSP (Cross Stage Partial) modules. Leaky rectified linear unit (Leaky ReLU) and Sigmoid are used as activation functions, while Stochastic Gradient Descent (SGD) or Adam is used as optimization functions.

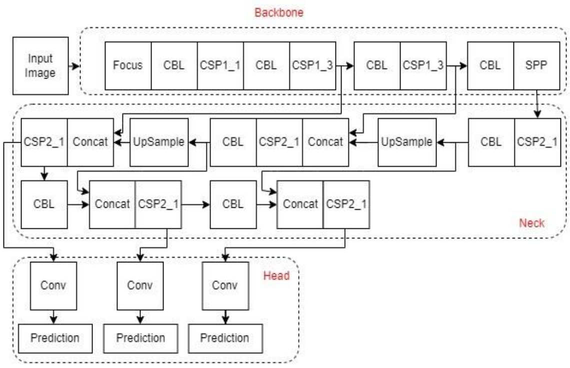

Figure 7. Network architecture of the YOLOv5

3.3. Dataset

Object detection models are significantly affected by the the dataset. It is important that the data to be given to the model is obtained from different aspects of daily life and under different conditions. Proper labeling also affects the training and validation.

The training and validation data were obtained from the GRAZ dataset and via a smartphone camera in different lighting and weather conditions and consist of a total of 497 images. Each image in the dataset may contain multiple objects. And after labelling, the dataset is obtained in txt format and it includes pedestrian, car, bicycle objects as well as stop, pedestrian crossing, give way, roundabout, uneven road, 20-speed limit, and 30-speed limit traffic sign objects.

The bounding box width and height values, and the coordinate values of the center points of the bounding boxes that cover each image's objects, are all contained in the txt file. The dataset is split into two parts: 80% for training and 20% for validation. Daily life videos were used for the testing process. Figure 8 displays some images from the dataset and Figure 9 illustrates the number of objects it contains. In this study, car class represents bus, automobile and minibus objects; and bicycle class represents motorcycle and bicycle objects.

Figure 8. Images from the dataset

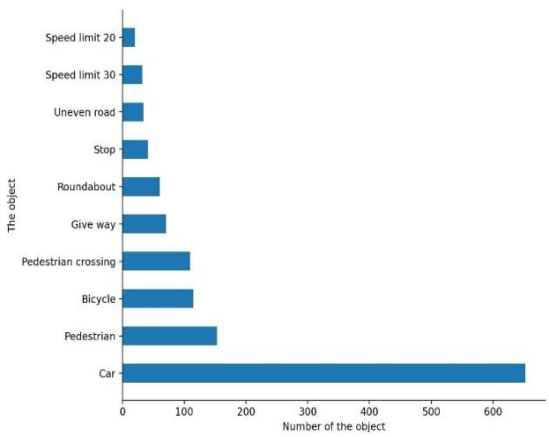

Figure 9. Number of the object in the dataset

4. Result and Discussion

On Google Collaboratory, the YOLOv5 model was trained for 1000 epochs using a Tesla T4 GPU. Increasing the number of epochs can lead to a higher risk of overfitting. This issue can be mitigated by increasing the number of images and diversifying them. The testing phase for object and lane detection was conducted on an Intel Core i5-9400 CPU. The variations in the loss values and performance metrics for different versions of YOLOv5 during the training process are shown in Figure 10 and Figure 11, respectively.

Each color in Figure 10 and Figure 11 denote the trained model. The names of the models in Table 1 represents the YOLOv5 version that has been used, the batch size value used in that version, and the size of the input image, respectively.

Table 1. The trained models

Model Name	Batch Size	Image Size	Epoch	Best Epoch
YOLO v5n_32_416	32	416	1000	792
YOLO v5n_32_640	32	640	1000	882
YOLO v5s_32_416	32	416	1000	847
YOLO v5s_32_640	32	640	1000	842
YOLO v5l_16_416	16	416	1000	804
YOLO v5l_32_416	32	416	1000	812
YOLO v5l_64_416	64	416	1000	614
YOLO v5x_16_416	16	416	1000	875
YOLO v5x_32_416	32	416	1000	738

Performance metric values for the trained models are presented in Table 2. From the table, it is evident that increasing the size of input images in the YOLO v5n and YOLO v5s models while keeping the batch size constant leads to improvements in performance metrics. However, there is no clear correlation between the results obtained when the input image size is held constant and the batch size is increased in the YOLO v5l and YOLO v5x models. In the input image size of 146x416, YOLO v5l exhibited better performance with a batch size of 64, while YOLO v5x performed better with a batch size of 16. Notably, in the YOLO v5l and YOLO v5x models, attempting to train with a larger input image size, such as 640, posed challenges due to insufficient memory resources. Consequently, training studies could not be conducted at these input image sizes or larger.

Table 2 shows that the YOLO v5x model, using a batch size of 16 and an input image size of 416, achieves the best results. The YOLO v5x model's deeper architecture and incorporation of a greater number of parameters enable it to capture more extensive and detailed features from various objects. As demonstrated in Figure 6, the unique attribute of YOLO v5x has enabled us to attain better outcomes when compared to alternative models. In this study, we will refer to this model as YOLO v5x in the subsequent process. Figure 12 depicts the variation in the loss values and performance metrics of the YOLO v5x model.

Figure 13 displays the confusion matrix obtained from the YOLO v5x model. The matrix indicates that the YOLO v5x model achieved a 100% detection rate for the stop traffic sign in the dataset. Generally, incorrect classifications were observed in relation to the background. Figure 14 presents the precision, recall, and mAP_0.5 performance values for each class in the YOLO v5x model. Precision measures how much of the objects that the model detects as positive are actually positive. It quantifies the rate at which the model detects the object as if it were present in the image, even when the object is not actually in the image. Recall measures the proportion of objects that should have been detected in the image but were not identified by the model. High precision and recall values indicate that the detection processes are performed very effectively. The F1 score, which aims to balance the performance of the model in both precision and recall in object detection, is calculated as the harmonic mean of these two metrics. The area under the PR curve, denoted as AP, represents how well the object detection model balances precision and recall. It visualizes how the model achieves a balance between precision and recall. The mean mAP is obtained by calculating the average of the AP values computed for each individual class. Considering Figure 14, it is generally observed that the precision value is close to 1. This situation indicates that the model's tendency to falsely detect an object as if it were present in the image, when there is actually no object, is very low. A low recall value negatively affected the F1 and mAP values. The low recall value is a result of the model's inability to detect an object in the image, even when the object is present. The lowest recall value was obtained in the speed limit 20 object. The limited number of images associated with the speed limit 20 object may be attributed to the model's inability to acquire sufficient features for detecting this object, hence resulting in a low recall value in detection. However, overall, the model achieved good results in detecting other classes.

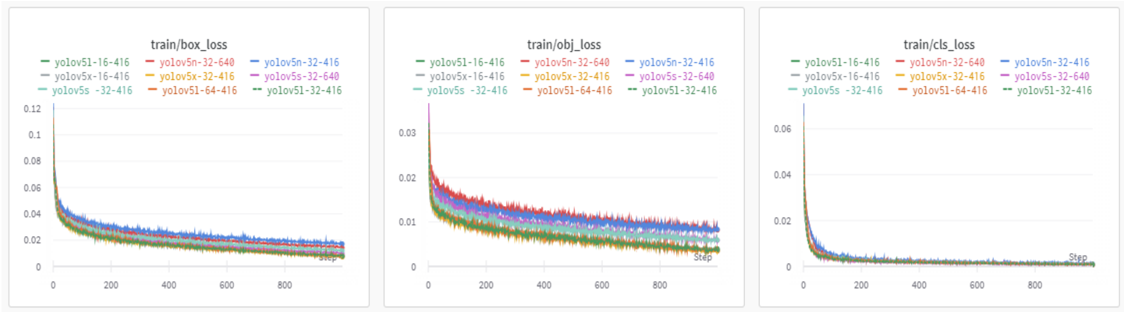

Figure 10. Graphs of losses of the models during the training processes

Figure 11. The variation of performance metrics of YOLOv5 versions during the training processes

Table 2. Comparison of the performance measures of the trained models

Model Name	P Curve	R Curve	PR Curve or mAP_0.5	F1 Curve	mAP_0.5:0.9
YOLO v5n_32_416	1.00	0.81	0.791	0.81	0.630
YOLO v5n_32_640	1.00	0.84	0.823	0.84	0.650
YOLO v5s_32_416	1.00	0.82	0.811	0.84	0.656
YOLO v5s_32_640	1.00	0.84	0.830	0.84	0.681
YOLO v5l_16_416	1.00	0.86	0.845	0.84	0.712
YOLO v5l_32_416	1.00	0.85	0.845	0.84	0.711
YOLO v5l_64_416	1.00	0.87	0.852	0.85	0.713
YOLO v5x_16_416	**1.00**	**0.88**	**0.869**	**0.86**	**0.726**
YOLO v5x_32_416	1.00	0.84	0.832	0.85	0.712

Figure 15 depicts the results of object detection tests performed on videos shot in various driving environments. Lane detection is shown in Figure 16. The image showing the object and lane detection together is given in Figure 17.

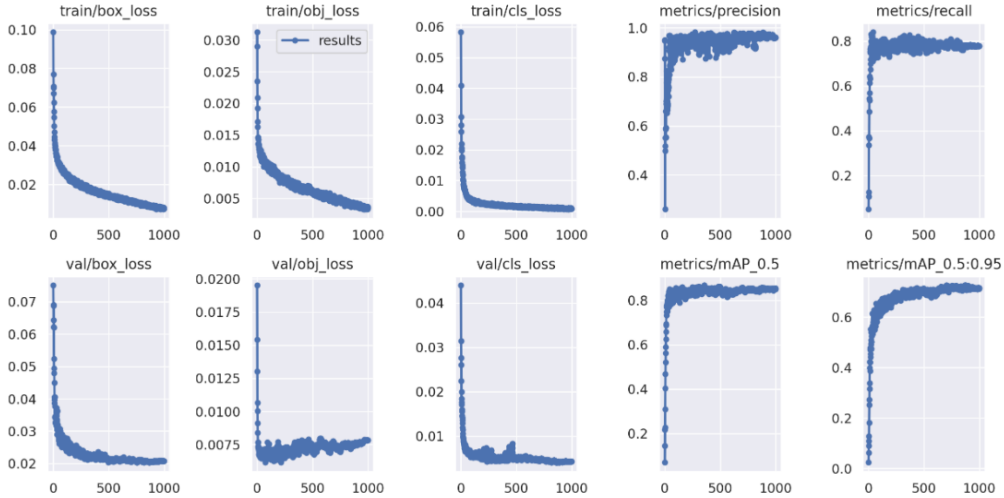

Figure 12. Observed changes in performance metrics during the training and the validation process

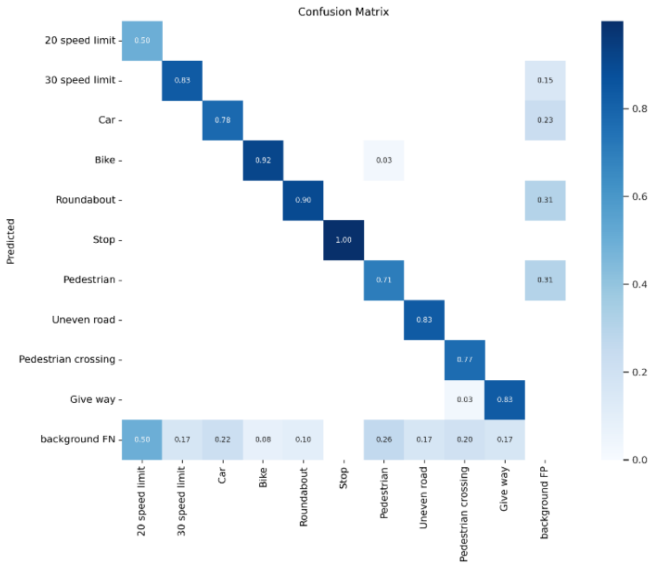

Figure 13. The confusion matrix obtained in the validation process

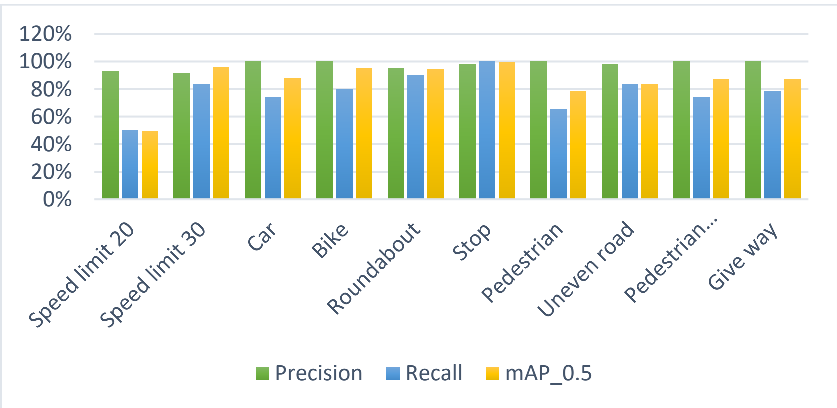

Figure 14. Comparison of classes based on precision, recall, and mAP

Figure 15. Detection of objects in various environments

Figure 16. Detection of the lane

Figure 17. Detection of the lane and the objects

5. Conclusion

This research aimed to study lane and surrounding object detection for cars, with the goal of preventing accidents in advanced driver support systems and autonomous driving vehicles. The proposed lane detection approach effectively identified both straight and curved lanes. However, lane detection may occasionally fail due to various road conditions, weather changes, and traffic congestion caused by other vehicles and objects.

Future studies could explore deep learning-based lane detection methods or other approaches to improve detection accuracy and address the challenges faced in lane detection. Additionally, a control system can be established for detected lanes. Furthermore, in situations where the vehicle approaches the boundary of any lane or deviates by a certain proportion from the center of the driving area, the control system can intervene.

The success of object detection models is determined by their accuracy in detecting objects and their detection time. In some cases, it may be necessary to use a memory-efficient model. The YOLOv5 model offers an advantage in this regard as it has a smaller file size compared to its previous versions, making it applicable in a wider range of scenarios. In this study, we trained 9 object detection models using five different versions of YOLOv5 on a Tesla T4 GPU. We compared the performance metrics of these models and identified the one that yielded the best results.

We utilized a set of 10 objects, encompassing traffic signs, pedestrians, and vehicles, commonly encountered in traffic. Thus, the study has been expanded by incorporating a greater variety of objects compared to literature studies that simultaneously perform lane and object detection. The images of these objects were obtained in rainy, sunny, and snowy weather conditions during both daytime and late afternoon hours. In future studies, the goal is to conduct detection using a broader range of objects encountered in traffic and to achieve detection in dark weather. Additionally, in future research, the aim is to develop systems that offer superior performance in terms of detection time, accuracy, and especially in the detection of small objects.

Article Information Form

Funding
The author (s) has no received any financial support for the research, authorship or publication of this study.

Authors' Contribution
Conceptualization, G.Ö., O.E. and R.K.; methodology, G.Ö.; software, G.Ö.; performing analysis, G.Ö.; data collection, G.Ö.; writing—original draft preparation, G.Ö.; writing—review and editing, G.Ö., O.E. and R.K. All authors have read and agreed to the published version of the manuscript.

The Declaration of Conflict of Interest/ Common Interest
No conflict of interest or common interest has been declared by the authors.

The Declaration of Ethics Committee Approval
This study does not require ethics committee permission or any special permission.

The Declaration of Research and Publication Ethics
The authors of the paper declare that they comply with the scientific, ethical and quotation rules of SAUJS in all processes of the paper and that they do not make any falsification on the data collected. In addition, they declare that Sakarya University Journal of Science and its editorial board have no responsibility for any ethical violations that may be encountered, and that this study has not been evaluated in any academic publication environment other than Sakarya University Journal of Science.

References

[1] R. Muthalagu, A. S. Bolimera, D. Duseja, S. Fernandes, "Object and Lane Detection Technique for Autonomous Car Using Machine Learning Approach," Transport and Telecommunication, vol. 22, no. 4, pp. 383–391, 2021.

[2] V. Nguyen, H. Kim, S. Jun, K. Boo, "A Study on Real-Time Detection Method of Lane and Vehicle for Lane Change Assistant System Using Vision System on Highway," Engineering Science and Technology, an International Journal, vol. 21, no. 5, pp. 822–833, 2018.

[3] H. G. Zhu, "An Efficient Lane Line Detection Method Based on Computer Vision," Journal of Physics: Conference Series, vol. 1802, no. 3, 2021, p. 032006.

[4] G. Ji, Y. Zheng, "Lane Line Detection System Based on Improved Yolo V3 Algorithm," In Review, preprint, 2021.

[5] B. Dorj, S. Hossain, D.-J. Lee, "Highly Curved Lane Detection Algorithms Based on Kalman Filter," Applied Sciences, vol. 10, no. 7:2372, 2020.

[6] X. Yan, Y. Li, "A method of lane edge detection based on Canny algorithm," in Chinese Automation Congress (CAC), Jinan, China, 2017, pp. 2120–2124.

[7] M. L. Talib, X. Rui, K. H. Ghazali, N. Mohd. Zainudin, S. Ramli, "Comparison of Edge Detection Technique for Lane Analysis by Improved Hough Transform," in Advances in Visual Informatics, H. B. Zaman, P. Robinson, P. Olivier, T. K. Shih, and S. Velastin, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2013, pp. 176–183.

[8] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, Q. Wang, "Robust Lane Detection from Continuous Driving Scenes Using Deep Neural Networks," IEEE Transactions on

Vehicular Technology, vol. 69, no. 1, pp. 41–54, 2020.

[9] T. M. Hoang, H. G. Hong, H. Vokhidov, K. R. Park, “Road Lane Detection by Discriminating Dashed and Solid Road Lanes Using a Visible Light Camera Sensor,” Sensors, vol. 16, no. 8, 2016.

[10] Y. Li, W. Zhang, X. Ji, C. Ren, J. Wu, “Research on Lane a Compensation Method Based on Multi-Sensor Fusion,” Sensors, vol. 19, no. 7, 2019.

[11] J. Wang, H. Ma, X. Zhang, X. Liu, “Detection of Lane Lines on Both Sides of Road Based on Monocular Camera,” in 2018 IEEE International Conference on Mechatronics and Automation (ICMA), 2018, pp. 1134–1139.

[12] S. Kumar, M. Jailia, S. Varshney, “An efficient approach for highway lane detection based on the Hough transform and Kalman filter,” Innovative Infrastructure Solutions, vol. 7, no. 5, p. 290, 2022.

[13] A. Dubey, K. M. Bhurchandi, “Robust and Real Time Detection of Curvy Lanes (Curves) with Desired Slopes for Driving Assistance and Autonomous Vehicles,” in International Conference on Signal and Image Processing (AIRCC), 2015.

[14] Y. Huang, Y. Li, X. Hu, W. Ci, “Lane Detection Based on Inverse Perspective Transformation and Kalman Filter,” KSII Transactions on Internet and Information Systems, TIIS, vol. 12, no. 2, pp. 643–661, 2018.

[15] A. Shustanov, P. Yakimov, “CNN Design for Real-Time Traffic Sign Recognition,” Procedia Engineering., vol. 201, pp. 718–725, 2017.

[16] I. Kilic, G. Aydin, “Traffic Sign Detection and Recognition Using TensorFlow’ s Object Detection API With A New Benchmark Dataset,” in 2020 International Conference on Electrical Engineering (ICEE), Istanbul, Turkey, 2020, pp. 1–5.

[17] R. Wang, Z. Wang, Z. Xu, C. Wang, Q. Li, Y. Zhang, H. Li, “A Real-Time Object Detector for Autonomous Vehicles Based on YOLOv4,” Computational Intelligence and Neuroscience, vol. 2021, p. e9218137, 2021.

[18] Z. Yang, J. Li, H. Li, “Real-time Pedestrian and Vehicle Detection for Autonomous Driving,” in 2018 IEEE Intelligent Vehicles Symposium (IV), Suzhou, China, 2018, pp. 179–184.

[19] A. Ćorović, V. Ilić, S. Ðurić, M. Marijan, B. Pavković, “The Real-Time Detection of Traffic Participants Using YOLO Algorithm,” in 2018 26th Telecommunications Forum (TELFOR), Belgrade, Serbia, 2018, pp. 1–4.

[20] G. Ozturk, R. Koker, O. Eldogan, D. Karayel, “Recognition of Vehicles, Pedestrians and Traffic Signs Using Convolutional Neural Networks,” in 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Istanbul, Turkey, 2020, pp. 1–8.

[21] N. Kemsaram, A. Das, G. Dubbelman, “An Integrated Framework for Autonomous Driving: Object Detection, Lane Detection, and Free Space Detection,” in 2019 Third World Conference on Smart Trends in Systems Security and Sustainablity (WorldS4), London, UK, 2019, pp. 260–265.

[22] G. Jocher, K. Nishimura, T. Mineeva, R. Vilariño. YOLOv5 Code Repository. June, 2020. [Online]. Available: https://github.com/ultralytics/yolov5