Deep Learning for Line Road Detection in Smart Cars

Zohreh Dorrani
Department of Electrical Engineering, Payame Noor University, Tehran, Iran.
Email: dorrani.z@pnu.ac.ir (Corresponding author)

ABSTRACT:
In recent years, smart cars have advanced rapidly and use artificial intelligence technology to predict behavior, make decisions, and control goals. This technology significantly determines the knowledge level of vehicles. In the complex and dynamic environment of road traffic, negligence, and inattention can lead to irreparable damage. Real-time identification and positioning of road lines are key to improving the safety of driving cars. To improve the performance of safe driving assistance, this paper shows how to detect road lines using the YOLOV8 algorithm and then make decisions to continue driving straight, turn right, and turn left. Simulation results and accuracy comparison show that this approach can be used as a reliable source for creating driving assistance scenarios in natural road traffic environments. The use of artificial intelligence and the precise architecture of YOLOV8 promise high speed and accuracy in smart cars.

KEYWORDS: Artificial Intelligence, Deep Learning, Smart Cars, YOLOV8.

1. INTRODUCTION

Lane detection [1] is a crucial task in autonomous driving systems. Lane markings provide essential information about road structure and traffic flow, which is critical for navigation and automated vehicle control.

Various methods have been employed for lane detection. Traditional computer vision approaches utilize geometric features of lane markings to identify them. However, these methods may prove unreliable under poor visibility or complex road conditions.

Recently, deep learning methods [2, 3] have been adopted for lane detection. These methods employ deep neural networks [4, 5] to learn the characteristics of lane markings from images. The YOLOV8 architecture is a deep neural network architecture [6] designed for object detection in images and videos. This architecture is renowned for its fast and accurate object detection [7] [8] capabilities. Consequently, this paper utilizes this architecture for lane detection.

The YOLOV8 architecture can detect lane markings in real time, which is essential for autonomous driving systems. Additionally, it can detect lane markings with high accuracy under various road and visibility conditions. Moreover, the proposed method can be applied to detect lane markings on various types of roads and streets.

The paper begins by introducing the YOLOV8 architecture. Subsequently, the proposed method is elaborated. The proposed method is evaluated using simulations. The results are presented, and the conclusion follows.

2. RELATED WORK

Road lane detection [9] is one of the key tasks in advanced driver assistance and automated driving systems. This allows cars to detect their lane on the road and steer their movements safely in traffic. In recent years, deep learning has emerged as a powerful tool for solving various computer vision problems, including road line detection. A new method for real-time lane detection is presented, utilizing U-Net to extract road lane features from input images and classify pixels as lane or foreground [10]. This method achieves good accuracy in detecting road lines under various lighting and weather conditions. This method achieves good accuracy in detecting road lines in different light and weather conditions.

Paper type: Research paper
https://doi.org/10.30486/MJTD.2024.1107681
Received: 21 February 2024; revised: 1 May 2024; accepted:5 May 2024; published: 1 June 2024
How to cite this paper: Z. Dorrani, **"Deep Learning for Line Road detection in Smart Cars"**, *Majlesi Journal of Telecommunication Devices*, Vol. 13, No. 2, pp. 63-68, 2024.

A deep learning method for recognizing road features in advanced driver assistance systems is presented. This method uses two deep learning models, YOLOv7 and Faster R-CNN, to detect road types, traffic signs, and road lines [11].

Attention-based deep neural networks have also been used to detect road lines [12]. The attention mechanism allows the model to focus on important parts of the image that are likely to contain road lines. This method has improved the road lane detection performance compared to traditional methods that do not use attention.

It also uses deep reinforcement learning to recognize road contours [13]. Reinforcement learning is a type of machine learning that allows the agent to interact with its environment and learn through trial and error. In this research, a reinforcement learning agent has been used to learn how to recognize road lines from input images.

Another research has presented a new dataset of real roadway images for training and evaluating deep learning models for roadway detection [14]. This dataset contains high-resolution images and accurate labeling of road lines in different light and weather conditions.

3. YOLOV8 ARCHITECTURE

The YOLOv8 algorithm is a speedy object detection method that tackles the task in a single go. It's broken down into four main parts: the input segment, the backbone, the neck, and the output segment.

Prepping the Image (Input Segment): The first step involves some creative data manipulation on the input image. This is done through a technique called mosaic data augmentation, which combines information from several images into one. It also calculates anchors that best fit the objects in the image (adaptive anchor calculation) and adds grayscale information where needed (adaptive grayscale padding).

Extracting Features (Backbone and Neck): This is the core processing unit of YOLOv8. The image goes through a series of Conv and C2f modules. These C2f modules are an upgrade over the older C3 modules used in YOLOv7. They're the workhorses for extracting features from the image at various scales.

C2f modules are efficient thanks to the ELAN structure, which reduces the number of layers needed.

They also leverage Bottleneck modules to improve the flow of information within the network.

This approach keeps the model lightweight while allowing it to capture more intricate details.

Fusing the Features (Neck): After the feature extraction, the SPPF module comes into play. It uses pooling layers with different sizes to combine the extracted feature maps. These combined features are then passed to the neck layer.

The neck layer in YOLOv8 is a powerhouse for feature fusion. It borrows the FPN (Feature Pyramid Network) and PAN (Path Aggregation Network) structures to create a more robust model.

This structure merges high-level (global details) and low-level (local details) feature maps using upsampling and downsampling techniques. This allows the network to share information between features of different sizes, which is crucial for detecting objects of varying scales.

Making the Detection (Detection Head): The detection head follows a common approach, separating the classification head (identifies object type) from the detection head (locates the object). This stage involves calculating loss functions and filtering out irrelevant detection boxes.

A method called Task Aligned Assigner helps identify positive and negative samples during loss calculation. Positive samples are chosen based on a combination of classification and localization scores.

The loss calculation itself has two parts: classification loss and regression loss (excluding the Objectness branch from previous versions).

Classification loss uses Binary Cross-Entropy (BCE).

Regression loss employs a combination of Distribution Focal Loss (DFL) and CIoU loss functions.

YOLOv8 uses separate heads (decoupled heads) to predict classification scores (object type) and bounding box coordinates (object location) at the same time.

Classification scores are shown in a two-dimensional matrix, indicating the likelihood of an object existing in each pixel location.

Bounding box coordinates are represented by a four-dimensional matrix, specifying how far the object's center is from each pixel.

Finally, YOLOv8 uses a task-aligned assigner to evaluate how well these predictions align with the actual objects in the image. This metric considers both classification scores and Intersection over Union (IoU) values, allowing the network to optimize both object identification and localization simultaneously.

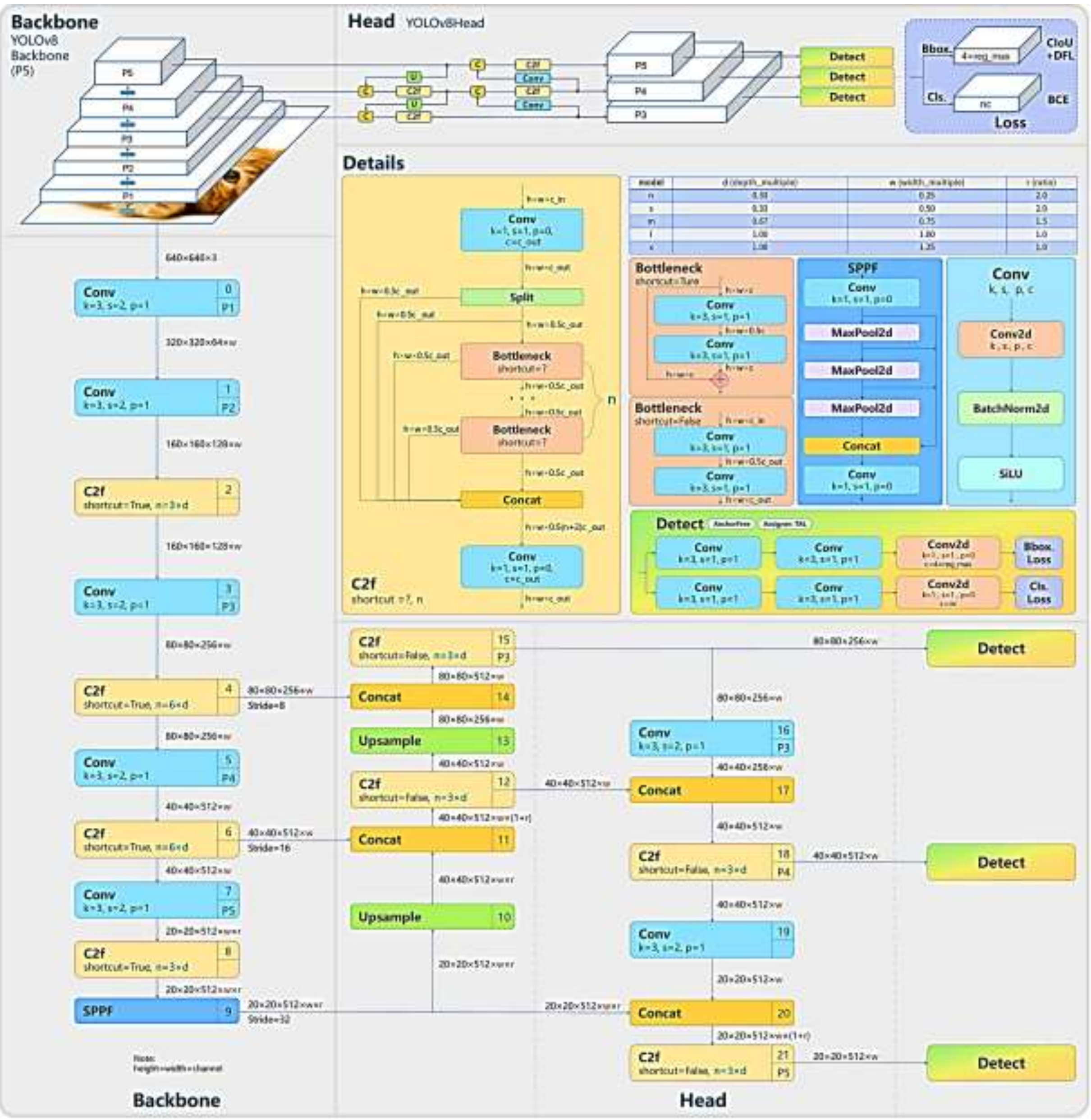

Fig. 1. YOLOv8 algorithm [15].

4. PROPOSED METHOD

Procedures for Extracting Road Line Features with YOLOV8:

A. Image Pre-processing: Car camera images undergo pre-processing before being fed into YOLOV8. This pre-processing includes resizing, noise removal [16], brightness enhancement, and removal of extraneous information.

B. Road Line Detection: YOLOV8 is trained to detect road lines within the pre-processed images. The model leverages deep learning algorithms to extract road line features such as position, direction, and lane type. By utilizing the backbone of this architecture, which consists of a series of convolutional layers, relevant features are extracted from the input image. SPPF processes features at different scales, allowing the model to detect road contours of varying sizes. C2f layers combine high-level features with background information, which improves detection accuracy. Finally, the final convolutional layer predicts the class and location of the lines.

C. Road Line Tracking: Following road line detection, YOLOV8 tracks these lines across consecutive images. This enables the automated driving system to comprehend the road geometry in real-time.

D. Post-processing: After tracking the road lines, the extracted information can be used to guide the car's behavior, such as making right or left turns, or continuing straight.

5. RESULT

The results of the simulation stage and how to extract the white and yellow lines are shown in Fig. 2.

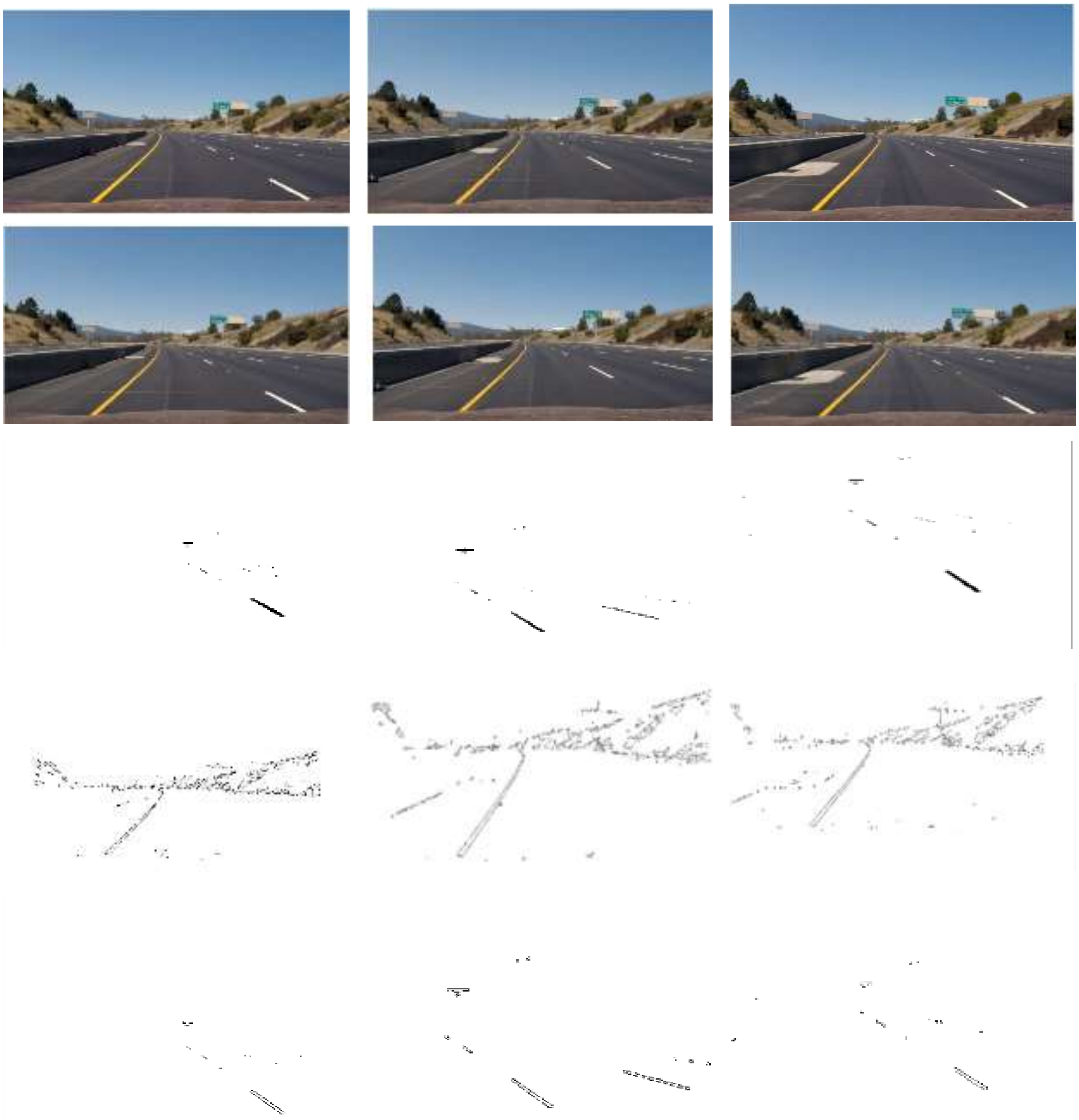

Fig. 2. Steps of the proposed method.

Lane markings play a vital role in identifying the designated path for vehicles on a road. They assist drivers in maintaining proper lane position, ensuring a safe distance from other vehicles, and determining the correct direction of travel.

Fig. 3 illustrates the results of road lane detection on dataset [17]. The red surface represents the detected lane area, and the green lines indicate the lane boundaries. However, achieving complete accuracy can be challenging due to uneven road surfaces and lane markings that may not be perfectly parallel.

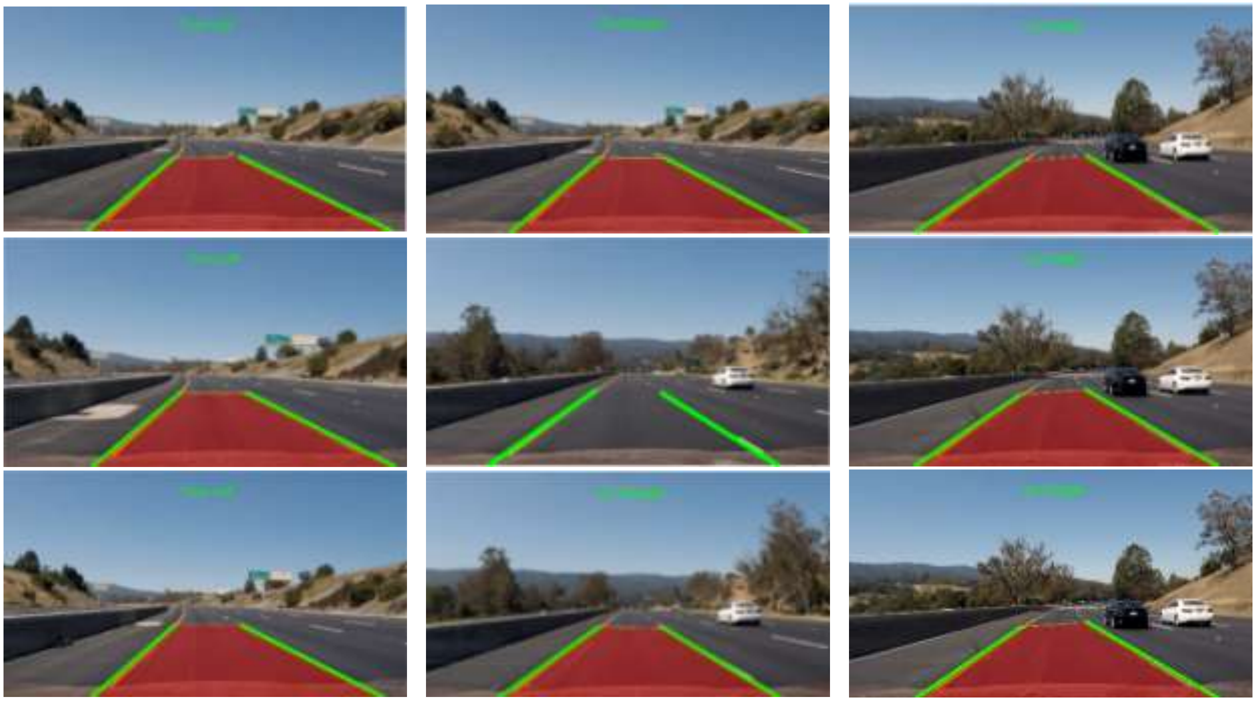

Fig. 3. lane line detection with proposed method.

To address this limitation, deep learning is employed. The YOLOv8 architecture is adopted, with adjustments made to the main channel and training process. It's important to note that if the weights of the initial layers are too small during loss propagation, the gradients may vanish within the network, hindering the learning process.

Line detection remains a crucial task for intelligent vehicles. However, environmental factors can sometimes obscure lane markings. These factors include shadows cast by vehicles or trees, poorly maintained lane markings, adverse weather conditions, and rough road surfaces. In such scenarios, the proposed scheme can be implemented to handle non-linear image features by extracting line segments.

The process involves pre-processing the image to remove noise and correct unsuitable background elements. Subsequently, line feature maps are extracted and sampled to classify the presence or absence of lane lines.

Table 1 compares the performance of the proposed method with several other methods in the field of detection accuracy. The first column shows the name of the method and the second column shows its accuracy.

Table 1. Performance evaluation of proposed method and comparison with some other methods.

Method	Accuracy
VGG16	89.91
VGG19	90.24
YOLO	91.14
YOLOV5	91.86
YOLOV8	92.85

Based on this table, the proposed method with 92.85% accuracy provides the best performance among the 5 methods for detecting objects in images. YOLOV8 with 92.85% accuracy, YOLOV5 with 91.86% accuracy, YOLO with 91.14% accuracy, and VGG19 with 90.24% accuracy are in the next ranks respectively. VGG16 has the lowest accuracy with 89.91% accuracy.

6. CONCLUSION

Road lane detection using the YOLOv8 architecture is a promising method for autonomous driving systems. This method offers several advantages, including high speed, high accuracy, and generalizability. This paper presents a novel approach for lane detection in smart cars using deep learning. The approach utilizes the YOLOv8 algorithm to identify road lines. Simulation results demonstrate that the approach can accurately detect road lanes under a wide range of conditions. This approach can be a reliable source for creating driving assistance scenarios in natural road traffic environments.

For future research, the approach could be tested in real-world conditions to assess its performance in low-light conditions and the presence of various objects on the road. Additionally, developing more computationally efficient approaches for road lane detection remains an important area of research.

REFERENCES

[1] D.-H. Lee and J.-L. Liu, "**End-to-end deep learning of lane detection and path prediction for real-time autonomous driving,"** Signal, Image and Video Processing, vol. 17, no. 1, pp. 199-205, 2023.

[2] A. Arshaghi and M. Norouzi, **"A Survey on Face Recognition Based on Deep Neural Networks,"** Majlesi Journal of Telecommunication Devices, 2023.

[3] A. A. Abed and M. Emadi, **"Detection and Segmentation of Breast Cancer Using Auto Encoder Deep Neural Networks,"** Majlesi Journal of Telecommunication Devices, vol. 12, no. 4, pp. 209-217, 2023.

[4] N. Habibi and S. Mousavi, **"A Survey on Applications of Machine Learning in Bioinformatics and Neuroscience,"** Majlesi Journal of Telecommunication Devices, vol. 11, no. 2, pp. 95-111, 2022.

[5] Z. Dorrani, "**Traffic scene analysis and classification using deep learning,"** International Journal of Engineering, 2023.

[6] M. Ghasemzade, **"Extracting Image Features Through Deep Learning,"** Majlesi Journal of Telecommunication Devices, vol. 9, no. 3, pp. 109-114, 2020.

[7] Z. Dorrani, H. Farsi, and S. Mohamadzadeh, **"Deep Learning in Vehicle Detection Using ResUNet-a Architecture,"** Jordan Journal of Electrical Engineering. All rights reserved-Volume, vol. 8, no. 2, p. 166, 2022.

[8] Z. Dorrani, H. Farsi, and S. Mohammadzadeh, **"Edge Detection and Identification using Deep Learning to Identify Vehicles,"** Journal of Information Systems and Telecommunication (JIST), vol. 3, no. 39, p. 201, 2022.

[9] Z. Dorrani, **"Road Detection with Deep Learning in Satellite Images,"** Majlesi Journal of Telecommunication Devices, vol. 12, no. 1, pp. 43-47, 2023.

[10] S.-W. Baek, M.-J. Kim, U. Suddamalla, A. Wong, B.-H. Lee, and J.-H. Kim, **"Real-time lane detection based on deep learning,"** Journal of Electrical Engineering & Technology, vol. 17, no. 1, pp. 655-664, 2022.

[11] H. Nadeem et al., "**Road feature detection for advance driver assistance system using deep learning,"** Sensors, vol. 23, no. 9, p. 4466, 2023.

[12] E. Oğuz, A. Küçükmanisa, R. Duvar, and O. Urhan, "**A deep learning based fast lane detection approach,"** Chaos, Solitons & Fractals, vol. 155, p. 111722, 2022.

[13] Z. Zhao, Q. Wang, and X. Li, "**Deep reinforcement learning based lane detection and localization,"** Neurocomputing, vol. 413, pp. 328-338, 2020.

[14] N. J. Zakaria, M. I. Shapiai, R. Abd Ghani, M. N. M. Yassin, M. Z. Ibrahim, and N. Wahid, **"Lane detection in autonomous vehicles: A systematic review,"** IEEE access, vol. 11, pp. 3729-3765, 2023.

[15] L. Zhang, G. Ding, C. Li, and D. Li, **"DCF-Yolov8: An Improved Algorithm for Aggregating Low-Level Features to Detect Agricultural Pests and Diseases,"** Agronomy, vol. 13, no. 8, p. 2012, 2023.

[16] Z. Dorrani and M. Mahmoodi, **"Noisy images edge detection: Ant colony optimization algorithm,"** Journal of AI and Data Mining, vol. 4, no. 1, pp. 77-83, 2016.

[17] Detection of lanes on a road and prediction of turns based on vanishing point. https://github.com/ysshah95/Lane-Detection-using-MATLAB.