Energy-Efficient Lane Identification Module with Sliding-Based Parallel Segment Recognition Accelerator for FPGA

T. KARPAGALAKSHMI
Department of Electronics and Communication Engineering
Adhiparasakthi Engineering College
Melmaruvathur, India
karpagalakshmipraba@gmail.com

M. MALATHI
Department of Electronics and Communication Engineering
Adhiparasakthi Engineering College
Melmaruvathur, India
ecemalathi@gmail.com

2025 International Conference on Computing and Communication Technologies (ICCCT) | 979-8-3315-3757-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICCCT63501.2025.11020556

***Abstract*— Lane detection is an emerging technology integrated into vehicles to facilitate autonomous navigation. Most existing Lane recognition systems are developed for well-structured roads, trusting on the presence of lane patterns. To address this limitation, we propose a sliding-based K-means parallel segment image dispensation approach. Since most onboard systems in autonomous vehicles are lightweight, the method is optimized to minimize computational load and power consumption. Lane identification is achieved by applying a Hough transform to an image where edge finding has been performed concurrently. The primary goal of edge detection is to recognize and pinpoint abrupt changes in pixel intensity, which define object boundaries and distinct elements within a scene. Various implementations of the Canny algorithm exist, as boundary detection serves as a fundamental step in many computer vision applications, such as edge-based obstacle recognition, facial recognition, target detection, and image compression. These computational techniques help distinguish vertical, flat, step, and corner edges. The effectiveness of edge detection depends on factors like lighting conditions, noise levels, edge density, and intensity variations within the image. Consequently, achieving precise boundary localization is crucial. Therefore, computational methods must account for these intensity fluctuations to establish accurate edge detection correlations. Ultimately, the process ensures lane detection with reduced noise, improved contrast boundaries, and optimized processing time.**

Keywords*—*Self-driving technology, lane recognition, Canny edge detection, Hough transformation, FPGA optimization, energy-efficient design, K-means clustering.

I. INTRODUCTION

The increasing number of vehicles on the roads and the growing demand for harmless and more well-organized conveyance systems has driven the development of driver assistance and autonomous driving technologies. Among these technologies, accurate lane detection plays a crucial role in ensuring that Vehicles stay within assigned lanes, directly contributing to road safety.

However, lane detection presents significant challenges in dynamic environments and under various lighting, weather, and lane wear conditions. These factors complicate the task and demand robust, real-time solutions. Computer vision-based approaches have shown promise in addressing these difficulties by providing algorithms capable of interpreting the environment in a manner similar to human vision.

In this context, the present work aims to develop a lane detection system using traditional computer vision techniques. Unlike recent approaches based on machine learning, the proposed method seeks to explore techniques that are less dependent on large volumes of data and have lower computational costs, making them suitable for devices with limited capabilities, such as cameras embedded in passenger vehicles. The developed system aims to overcome obstacles such as adverse weather conditions, lane wear, and different types of roads, offering an efficient solution for real-time detection. The methodology adopted for lane detection includes steps such as camera calibration, distortion correction, perspective transformation to obtain a bird's-eye view, lane segmentation through limiting techniques, and the application of sliding window methods for tracking detected lanes. Eric Hsueh-Chan Lu and Wei-Chih Chiu [1], introduced a CycleGAN based lane detection method by integrating CycleGAN and image segmentation method and also loss function for outcome differences [1]. Yingying Xing, et al. [2] suggested a innovative technique by utilizing The Mexican hat Wavelet (MHW) for precise detection of key time points [2]. Lu Zhang, et al. [3], presented a novel lane uncovering technique called as TSA-LNet, combined two directional separation attention (TSA) and lightweight network (LNet) to improve the detection ability of vehicles [3]. Yuxuan Chai et al. [4] tackle these challenges by improving the row-anchor-based lane identification approach. They utilize a Transformer encoder-decoder erection, where row arrangement strengthens the model's capability to capture universal topographies and accurately identify lane Boundaries in intricate surroundings [4].

Yuxuan Chai makes significant contributions to LiDAR-based lane detection in three key aspects. First, the BoostedDim Attention method is introduced to improve the conventional Multi-Head Self-Attention (MHA) computations within the shallow Vision Transformers-based K-Lane baseline model. This approach proves highly effective, especially in challenging conditions such as unfamiliar and nighttime environments at short distances (0–30 m) and daytime settings for extended ranges (30–50 m) [5]. Seung-Hwan Lee et al. [6] implemented a technique by using a A U-Net-driven framework for learning, integrating extra lane-specific details into a four-channel input data structure that accounts for lane features. The fourth channel functions as an edge attention map (E-attention map), enabling the modules to attain more dedicated erudition focused on lane detection [6].

Sunil Kumar et al. [7], present an efficient lane detection approach using semantic division to recognize road outlines in a high-dimensional dataset by incorporating perpendicular longitudinal attributes and appropriate lashing data. This study introduces two designed modules—the feature integration chunk and the information exchange block—to enhance the finding of uncertain and congested lane lines [7]. Libo Sun, et al, proposed an innovative anchor-based lane detection network (SP-Det) that integrates the distinct mechanical properties and pixel delivery of lane lines. Exactly, they introduce a Semantic-Guided Feature Calibration Unit (SG-FCU) to semantically adjust and improve topographies across dissimilar coatings, effectively reducing the semantic break throughout union [8].

II. METHODOLOGY

The aim of the suggested theory is to create an effectual, low computational cost solution capable of operating in real-time and under a variety of road conditions. The adopted methodology, based on camera calibration, perspective transformation, and colour and gradient segmentation, proved effective in accurately detecting lanes under normal driving conditions, achieving an accuracy rate of over 90% in daytime scenarios.

Although the system demonstrated robust performance in most tests, limitations were observed by using Clustering algorithm under adverse conditions, such as worn lanes, severe weather variations, and sharp curves. These limitations highlight the need for future improvements, particularly in adapting segmentation limits for scenarios with high environmental variability and improving perspective projection for roads with more complex geometries.

The modularity of the implemented pipeline allows for iterative enhancements, making it possible to integrate more advanced computer vision or deep learning techniques to handle more challenging scenarios. Additionally, fusion with other sensors, such as LiDAR or radar, can increase the system's robustness, making it more adaptable to different traffic and environmental conditions.

The proposed system represented in figure 1, shows significant potential for application in driver assistance systems and autonomous vehicles, especially in contexts that require efficient, low- cost solutions. Despite the observed limitations, the traditional computer vision approach provides a solid foundation for future extensions and refinements, with possibilities for application in various areas of assisted driving and vehicle automation.

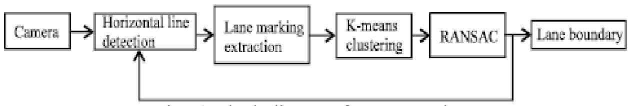

Fig : 1 Block diagram for proposed system

A. *Different Variants and Methods of the Hough Transform*

Over time, several modifications and methods have enhanced the efficiency and versatility of the Hough Transform:

1) The original concept introduced by Paul Hough for detecting lines is referred to as the Standard Hough Transform (SHT). This approach involves voting for all possible parameter combinations and discretizing the parameter space.

2) Probabilistic Hough Transform (PHT): The PHT enhances efficiency by randomly selecting a subset of edge points and performing line detection only on those specific locations. This approach reduces computational complexity while preserving accuracy, making it suitable for real-time applications.

3) Accumulator Space Dimensionality: While the traditional Hough Transform identifies lines in two dimensions, it can also be extended to higher dimensions to detect more complex shapes, such as ellipses or circles. Each additional dimension corresponds to an extra parameter of the detected shape. To diminish clatter in the image, a 5×5 Gaussian filter is used. The Sobel kernel is then used to determine the image's intensity gradient, detecting edges in straight, perpendicular, and slanting directions. A full scan of the image is conducted, shadowed by the removal of unnecessary pixels that do not represent edges, a process known as Non-Maximum Suppression. The final step, Hysteresis Limiting, determines which edges are valid by using two limit values—minimum and maximum—to differentiate real edges from noise.

4) Edge Classification: Edges with an intensity gradient surpassing the extreme limit are confirmed as valid edges, while those below the minimum limit are eliminated. The remaining edges, which fall Among these two values, they are categorized according to their connectivity.

B. *Image Expansion:*

The output of the Canny edge finding role is an image containing only the boundary shape of the route, with all other image features detached. This processed image must then be conceded to a penetrating procedure to detect the lane geometry. However, before doing so, image dilation is applied to enhance edge clarity, making them more continuous and reducing breaks. Dilation enlarges the pixel regions and adds pixels along object boundaries.

The image is then divided into horizontal slices, and a search window—a fixed-width and height rectangular region—is slid across each slice to identify areas with the highest frequency of edge pixels. For this purpose, a window search analyzes the histogram of the lower half of the binary image, helping determine the approximate location where the lane lines start. After both the left and right edges are detected, a suitable function is employed to generate a best-fit curve along the lines.

Since Window scanning can be computationally demanding intensive, a margin search can be used in subsequent frames to optimize processing time. Instead of scanning the entire image, margin search restricts the search to a narrow region around the previously detected lane line, improving efficiency while maintaining accuracy.

C. *Perspective Transformation:*

After distortion correction, the image is subjected to a perspective transformation to simulate a bird's-eye view. This technique is widely used in lane detection, as the orthogonal projection of the road facilitates the identification of lanes as approximately parallel lines, simplifying the detection and tracking process.

The perspective transformation was performed using the cv2.getPerspectiveTransform function, which requires defining four source points in original image and their consistent coordinates in transformed image.

These points were selected empirically, considering that lanes in the original images may appear distorted depending on the camera position and road geometry. The perspective transformation aims to align the lanes so they can be detected more precisely, especially on straight road sections.

However, this technique presents limitations on roads with sharp curves, where the orthogonal projection may not accurately represent the actual lane trajectory. This issue is addressed in the subsequent step of lane detection and tracking, with the introduction of algorithms that handle variations in road geometry.

D. Image Segmentation:

The third step of the pipeline is image segmentation, a crucial process to isolate the pixels corresponding to the lane markings. For this, a combination of limiting techniques based on gradients and colour channels was employed. The segmentation aims to highlight the lane markings, differentiating them from the rest of the road and the surrounding scene.

The approach used explores different colour spaces, such as RGB, HLS, HSV, and LAB, to maximize the system's robustness under various lighting and surface conditions. Each of these colour spaces provides distinct characteristics that help isolate the white and yellow lane markings commonly found on roads.

In the HLS space, for example, yellow and white lanes are detected using adaptive limiting. One of the challenges encountered was the adaptation of the detection limit according to road conditions, especially in cases of wear or highly reflective surfaces. The cv2.inRange function was applied to create binary masks that highlight the lanes of interest, combining them to generate the final binary image.

E. Lane Detection:

After generating the binary image, the next step is lane detection. For this, the sliding windows method was used, a technique widely employed in line segmentation in images. The sliding windows algorithm works by identifying regions in the image with a high density of pixels corresponding to the lanes. Once the initial points are identified, the algorithm expands these windows along the image, tracking the pixels associated with the lanes and fitting them to a second- degree polynomial that describes the trajectory of the lanes along the road.

Lane detection method shown in figure 2, offers the advantage of being efficient in real-time and operating robustly on straight roads. However, on curved roads or where lanes are partially worn, detection may be less precise. To mitigate this issue, an adaptive search technique was introduced, allowing the algorithm to automatically adjust the detection windows based on the geometric characteristics of the lanes detected in previous frames. Additionally, lane detection was refined by combining different color and gradient segmentation methods, ensuring system robustness under various lighting and road texture conditions. The pixels corresponding to the lanes were then organized into a logical sequence, enabling continuous lane tracking across subsequent frames during video processing.

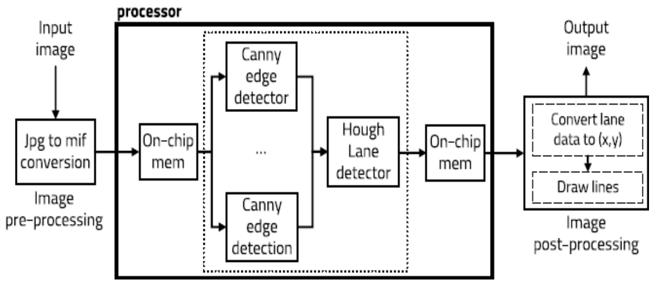

Fig 2: Lane detection processing

F. Lane Detection Accuracy:

The algorithm demonstrated robust performance in identifying lane markings under a variety of road conditions. When tested on videos from the CU Lane and Tu Simple datasets, the system was able to detect and track lanes accurately, both on straight sections and on slightly curved roads. The perspective transformation technique, combined with colour and gradient segmentation, proved effective in separating the lanes from the rest of the image, facilitating continuous tracking throughout the processed frames.

Specifically, the white and yellow lane markings, predominant on the roads in the evaluated videos, were consistently identified, even under variable lighting conditions such as direct sunlight and partial shadows. In night time lighting scenarios, colour segmentation was also effective, provided the lanes were well- defined and exhibited sufficient contrast relative to the road surface.

G. Performance in Adverse Conditions

Although the system exhibited satisfactory accuracy in normal driving conditions, some challenges were identified in adverse scenarios. On roads with heavily worn lane markings or in extreme weather conditions, such as fog or intense sunlight reflections, the algorithm showed a reduction in detection accuracy. This occurred mainly due to the low distinction between the lane markings and the rest of the road, which made it difficult to precisely segment the pixels corresponding to the lanes.

Additionally, on sharp curved sections, the perspective transformation caused distortions in the lane projections, compromising tracking accuracy. This phenomenon is due to the orthogonal projection used in the perspective transformation, which tends to inadequately represent the curved road geometries.

H. K-means and RANSAC Method for Finding Boundary

The suggested approach identifies the left and right lanes separately. That an edge point goes to whether left lane or right lane is unknown, so K-means clustering procedure is used to classify the edge points into two groups: left lane and right lane shown in figure 3.

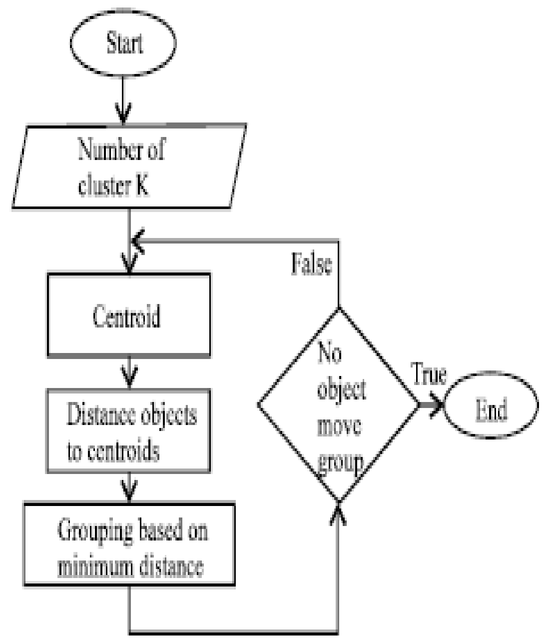

Fig 3: K-cluster Flow model

K-means clustering is a straightforward yet efficient clustering algorithm that requires specifying the number of desired clusters before classification. The primary goal of K-means clustering is to minimize an objective function, which is defined as follows:

Here, ||xi(j) - cj||² represents the chosen distance metric between a data point xi(j) and the cluster center cj, indicating the proximity of the n data points to their respective cluster centers. This process summarizes the key stages of the K-means clustering algorithm.

In the current study, we set k = 2 as the number of clusters. Initially, two edge points are randomly selected as the starting cluster centers for each lane. The remaining edge points are then assigned to their nearest cluster center based on the measured distance and orientation.

In the next stage, the average values of each cluster are computed and considered as the updated cluster centers. Subsequently, the entire procedure is iterated with these new cluster centers until there is no further alteration in their positions. As a result, we obtain the left lane group and the right lane.

To model a line for each K-means clustering dataset, which contains numerous anomalies, both the Hough Transform and Least Squares Fitting techniques proved ineffective. Consequently, we employed the Random Sample Consensus (RANSAC) algorithm. The number of iterations, NNN, was set sufficiently high to guarantee a 99% likelihood that at least one randomly selected sample set would be free of anomalies. In this research, NNN was set to 300, leading to the final outcome of the proposed method. Once the left and right lane boundaries were identified, their intersection point was determined. This point was then considered as the horizontal line position for the subsequent frame in the sequence. This strategy enhances the stability of the entire process and minimizes computational effort by omitting the repeated detection of the horizontal line.

I. Post Processing of K-Means output:

A straightforward approach to utilizing the K-Means output is to consider the midpoints of the cluster-mean-line segments. These midpoints serve as control points, allowing the entire lane to be represented as a spline constructed from these points. As mentioned earlier, curved lanes contain more mean points compared to straight lanes. This ensures that the algorithm effectively captures any curvature, providing a more detailed representation for highly curved lanes while using fewer data points for straight lanes. Preprocessing edge detection shown in figure 4.

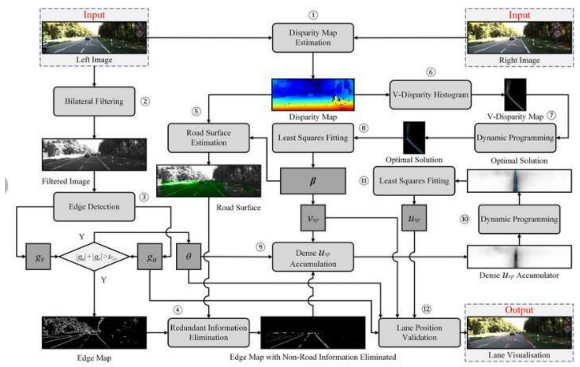

Fig 4 : Pre-processing Edge Detection program

Lanes are typically visualized as splines, but one challenge remains. Due to the randomized initialization of clustering, these data points are not arranged sequentially. As a result, the lane has not yet been fully reconstructed in its proper form. It is essential to recognize that these points may not accurately represent the lane as required. If the points are not in the correct order, plotting the spline properly becomes impossible. To make the data more useful for visualization, the points must be arranged in a structured order suitable for spline plotting.

Lanes are generally long, narrow strips regardless of their colour. The contours outlining these lanes in an image typically form elongated, thin polygons. These polygons often have distinct tips that mark the starting point of the lanes. The lane data point closest to this tip serves as the lane's starting point. These sharp "tips" or "corners" are usually near the initial point of the spline that accurately represents the lane. Therefore, among the set of points obtained from K-Means clustering, it is necessary to identify the one closest to this "tip" as the starting point of the spline. Once this point is determined, the remaining points can be sequentially arranged by iteratively selecting the closest point from the remaining set. If the output is corrupted due to data nonuniformity, the lane fitting techniques is employed for correct the output.

III. RESULT & DISCUSSION

To evaluate the presentation of lane finding and trailing procedures, the results are associated with a ground truth dataset to determine in terms of performance metrics. A true positive arises once a lane marker exists in the ground truth and is correctly noticed by the algorithm, while a false positive happens when the process detects a lane marker that does not exist. A false negative is recorded when the algorithm fails to detect a lane marker present in the ground truth, whereas a true negative occurs when no lane marker exists in the ground truth, and the algorithm correctly refrains from detecting any markers. The model view in represented in figure 5.

Fig 5: Model view

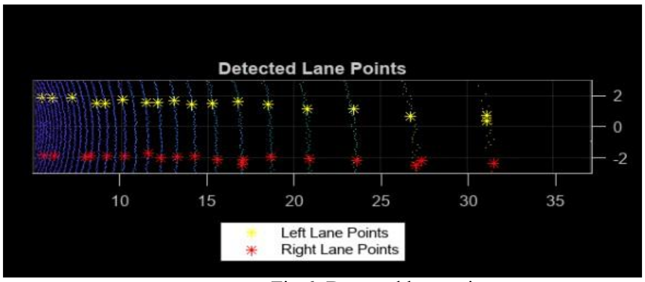

Fig 6: Detected lane point

Figure 6 representing the detected lane point. As a result, the overall algorithm maintains a polynomial time intricacy, creating it appropriate for real-time claims while ensuring high accuracy. In the denoising phase, selecting the appropriate size for the examining window and determining the unstable detachment are crucial to maintaining reliable performance. Consequently, the values of RL and RU were determined to be 0.07 and 0.55, respectively. Adjustments were made to the window sliding approach in the line detection step. Specifically, SC was set to 7×30, and KC was set to 10. The limit parameters utilized in the algorithm are detailed in Table 1.

TABLE 1: LIMIT PARAMETERS

Parameter name(s)	**Values(s)**
TE — TH	12-50
TG1 — TG2	20-15
T_1 — T_2	0.15-0.25
TC	0.5

The final output was transformed back onto the Ground plane utilizing the inverse of matrix W, with the entire process requiring just 50 ms per image. Digital image processing involves designing a digital system that executes operations on a digital image.

An image is essentially a two-dimensional signal, represented mathematically as f(x, y), where x and y correspond to the horizontal and vertical coordinates. The largeness off at any coordinate pair (x, y) is known as The brightness or grayscale level of the image at that specific point. When x, y, and the amplitude values of f are all limited and take discrete values, the image is termed a digital image. A digital image consists of a finite set of elements, each with a distinct position and value, commonly referred to as picture elements, image elements, pels, or pixels. The region of intrest and output shown in figure 7 and 8.

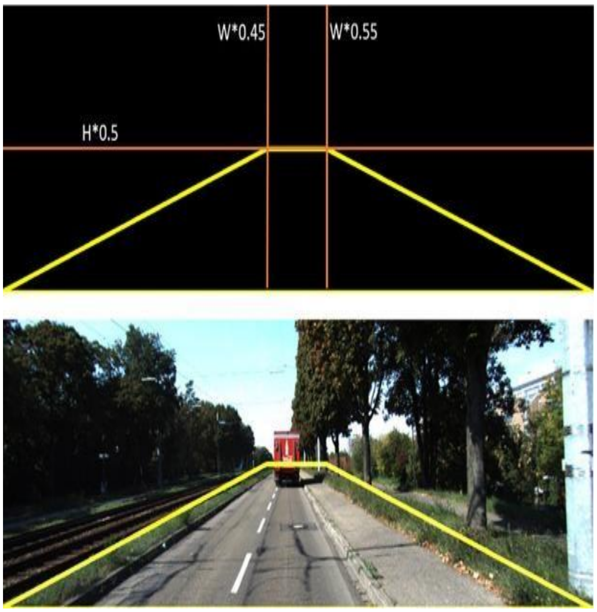

Fig : 7 The Region of interest (ROI) structure

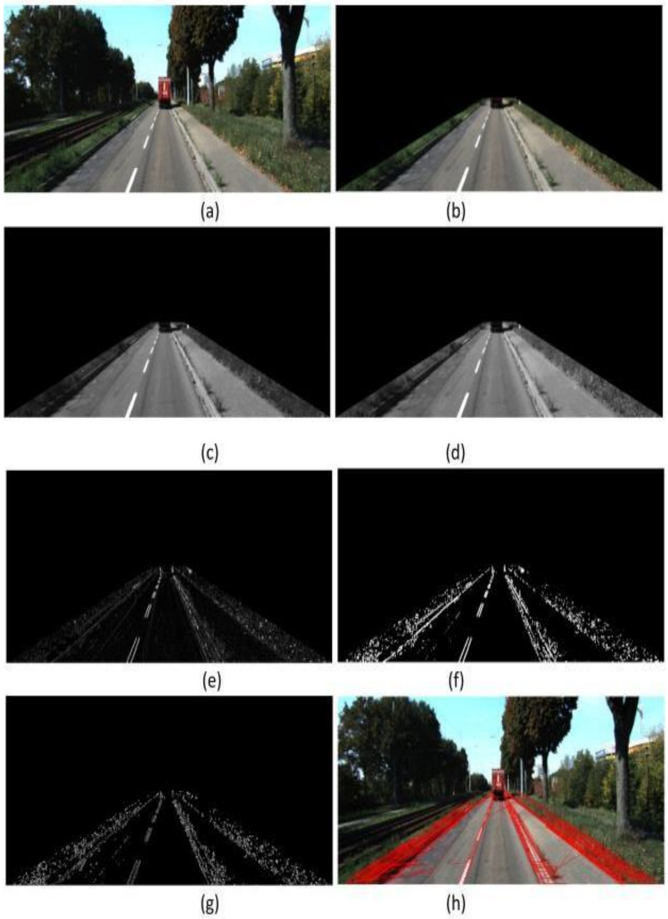

Fig 8. output

IV. CONCLUSION

The algorithm was designed to identify lane borders and transform their statistics into a suitable structured arrangement. For additional processing in self-driving systems. This objective has been successfully accomplished, making the algorithm viable for integration into autonomous vehicle systems for various applications. Future enhancements may focus on refining K-Means clustering with more precise mathematical approximations, improving illumination normalization methods, and implementing more effective noise reduction techniques. This work developed and implemented a lane detection system using traditional computer vision techniques. This is to create an efficient, low computational cost solution capable of operating in real-time and under a variety of road conditions. The adopted methodology, based on camera calibration, perspective transformation, and color and gradient segmentation, proved effective in accurately detecting lanes under normal driving conditions, achieving an accuracy rate of over 90% in daytime scenarios. Lane parting cautioning is a crucial A key element of advanced driver assistance. systems. Over the past decade, Significant growth has been achieved in lane detection. and tracking. A vision-based approach remains one of the simplest methods for lane detection. However, despite considerable progress in this field, there is still room for improvement due to the vast variability in lane environments.

REFERENCES

[1] Eric Hsueh-Chan Lu and Wei-Chih Chiu, “Lane Detection Based on CycleGAN and Feature Fusion in Challenging Scenes”, ehicles, 2025, volume: 7(1), https://doi.org/10.3390/vehicles7010002

[2] Yingying Xing, Yutong Wu, Hongwei Wang, Ling Wang, Lun Li, Yichuan Peng, “Failed lane-changing detection and prediction using naturalistic vehicle trajectories”, Transportation Research Part C: Emerging Technologies, 2025, volume:170, https://doi.org/10.1016/j.trc.2024.104939

[3] Lu Zhang, Fengling Jiang, Jing Yang, Bin Kong, Amir Hussain, “A real-time lane detection network using two-directional separation attention”, computer-aided civil and infrastructure engineering, 2024, volume:39(1), pages: 86-101, https://doi.org/10.1111/mice.13051

[4] Yuxuan Chai, Shixian Wang, and Zhijia Zhang, “A Fast and Accurate Lane Detection Method Based on Row Anchor and Transformer Structure”, Sensors 2024, 24(7), 2116; https://doi.org/10.3390/s24072116

[5] Omkar Patil, Binoy B. Nair, Rajat Soni, Arunkrishna Thayyilravi, C.R. Manoj, “BoostedDim attention: A novel data-driven approach to improving LiDAR-based lane detection”, Ain Shams Engineering Journal, 2024, volume: 15(9), https://doi.org/10.1016/j.asej.2024.102887

[6] Seung-Hwan Lee and Sung-Hak Lee, “U-Net-Based Learning Using Enhanced Lane Detection with Directional Lane Attention Maps for Various Driving Environments”, Mathematics 2024, 12(8), 1206; https://doi.org/10.3390/math12081206.

[7] Sunil Kumar, Ankur Pandey & Sudeep Varshney, “Exploring the Impact of Deep Learning Models on Lane Detection Through Semantic Segmentation”, SN Computer Science, 2024, volume:5, https://doi.org/10.1007/s42979-023-02328-5

[8] Libo Sun, Hangyu Zhu, Wenhu Qin, “SP-Det: Anchor-based lane detection network with structural prior perception”, Pattern Recognition Letters, 2025, Volume 188, Pages 60-66, https://doi.org/10.1016/j.patrec.2024.11.030