International Scientific Journal of Engineering and Management (ISJEM)
Volume: 05 Issue: 01 | Jan – 2026
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
DOI:10.55041/ISJEM03392

Laneline Detection using OpenCV

Preetham M
Department of Computer Science and Engineering
Jain University
Bangalore, Karnataka
India
21btrcs195@jainuniversity.ac.in

Rohana Upadhyay
Department of Computer Science and Engineering
Jain University
Bangalore, Karnataka
India
21btrcs200@jainuniversity.ac.in

Likith Damodhar
Department of Computer Science and Engineering
Jain University
Bangalore, Karnataka
India
21btrcs172@jainuniversity.ac.in

Sanjev R J
Department of Computer Science and Engineering
Jain University
Bangalore, Karnataka
India
21btrcs066@jainuniversity.ac.in

Sanjai R J
Department of computer Science and Engineering
Jain University
Bangalore,Karnataka
India
21btrcs065@jainuniversity.ac.in

Abstract- In view of the huge computing, poor anti-interference ability of traditional detection alogrithm, it does not meet the requirement of the vehicle system, for which this paper proposed a lane detection method based on OpenCV. Preprocessing image in the OpenCV environment, adopting LMedSquare(Least Median Square) idea to select the best subset combined with least squares method to picewise fitting the lane so that it realized automatic identification of lane. This algorithm is suitable for both straight and curve. Simulation shows that this algorithm has well real-time performance, accuracy and robustness. It can meet the requirements of the vehicle system.

I. INTRODUCTION

Lane line detection and recognition play a crucial role in vehicle collision warning systems and are widely implemented in modern automotive technologies. The accuracy, real-time performance, and robustness of these systems are directly linked to vehicle and driver safety. Existing research often employs separate algorithms for straight and curved lanes, which increases overall algorithmic complexity. While the Hough Transform is commonly used for lane detection, its high computational load and limited real-time efficiency make it unsuitable for vehicle system requirements. Similarly, the least squares method, although popular, is highly sensitive to noise and lacks strong anti-interference capabilities, which compromises detection accuracy.

To address these challenges, this paper proposes a novel approach that leverages the LMedS (Least Median of Squares) technique to select the optimal subset of data, followed by a piecewise fitting process using the least squares method. This strategy effectively filters out noise during the search for lane line features, enhancing the system's resistance to interference. The proposed algorithm is applicable to both straight and curved lanes, simplifies the detection process, and meets the real-time performance needs of vehicle systems.

II. RELATED WORK

In existing literature, various systems have been proposed for vehicle accident detection and alert mechanisms. Researchers have employed a range of approaches, including mathematical models, statistical analysis, and machine learning techniques, to design these solutions. Vehicle accidents can be classified into different types, such as collisions, rollovers, and fall-offs, among others. This paper reviews work related to

detecting vehicle collisions, identifying rollovers, and assessing accident severity.

Additionally, it explores multiple accident reporting mechanisms and discusses how multi-sensor fusion strategies have been applied to enhance detection accuracy. Several commercially available smart vehicle solutions are also examined [4]. Accident-related systems can generally be divided into two major categories: accident detection systems and accident prevention or warning systems. Some of these systems utilize onboard vehicle sensors, while others rely on roadside infrastructure like sensor networks to track and analyze roadway incidents [5].

A variety of methods are used across these systems, including vision-based technologies, proximity sensors, and motion or inertial sensing devices. This section provides a comprehensive overview of these diverse tools and techniques. According to [6], collisions can be identified by dynamically adjusting a velocity threshold based on physical parameters such as jerk, acceleration, speed, and displacement. Another method involves using a dedicated collision detection algorithm. One proposed design includes a crash sensor equipped with an accelerometer to measure initial deceleration, a comparator to evaluate whether this value exceeds a predefined threshold, and a triggering circuit to activate a protective mechanism like an airbag.

III. METHODOLOGY

This project focuses on detecting lane lines in an image using Python and OpenCV. OpenCV, short for "Open-Source Computer Vision," is a software library that offers a wide range of tools for image processing and analysis.

A. Canny Edge Detection Technique

Edge detection aims to identify the boundaries of objects within an image. The process involves locating regions where the pixel intensity changes sharply. In digital images, which are essentially matrices or arrays of pixels, each pixel denotes the brightness level at a specific location. These intensity values range from 0 to 255, where 0 represents pure black and 255 represents pure white. A gradient refers to the variation in pixel brightness. A large gradient signifies a rapid change in intensity (steep edge), while a small gradient indicates a more gradual transition (gentle edge).

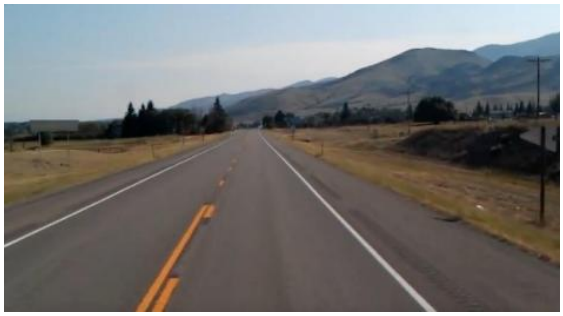

Fig1:original image

B. Gaussian Blur

In a grayscale image, each pixel is represented by a single numerical value indicating its brightness. To smooth the image and reduce noise, a common approach is to replace each pixel's value with the average of its surrounding pixel intensities. This process involves using a kernel—a small matrix of values—to compute the average. The kernel acts as a filter that moves across the image, averaging nearby pixel values to soften the image.

A Gaussian kernel, which contains values based on the Gaussian (normal) distribution, is particularly effective for this. For example, a kernel like np.array([[1,2,3],[4,5,6],[7,8,9]]) represents a weighted matrix used in the smoothing process. In this project, a 5x5 Gaussian kernel is applied to the grayscale image to blur it. This is implemented in OpenCV with the following function**.**

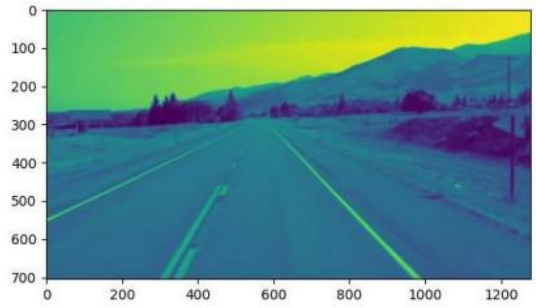

Fig2 : gaussian blur

C. Edge Detection

An edge in an image is a region where there is a significant change in intensity or color between neighboring pixels. A sharp change indicates a steep gradient, while a gradual change corresponds to a shallow gradient. An image can be viewed as a matrix, with rows and columns representing pixel intensities, or as a 2D coordinate system, where the x-axis corresponds

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 01 | Jan – 2026 DOI:10.55041/ISJEM03392
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

to the image's width (columns) and the y-axis corresponds to its height (rows).

The Canny edge detection function calculates the change in brightness between adjacent pixels by computing derivatives along both the x and y axes. Essentially, it determines the gradient, or the change in brightness, in all directions. The function then highlights the strongest gradients by marking them with white pixels, effectively tracing the edges of objects in the image.

(a) original image(b) grayscale image(c) denoised image(d) binary image

allows the system to recognize an accident no matter where it appears within the image frame—an essential trait for real-world accident detection.

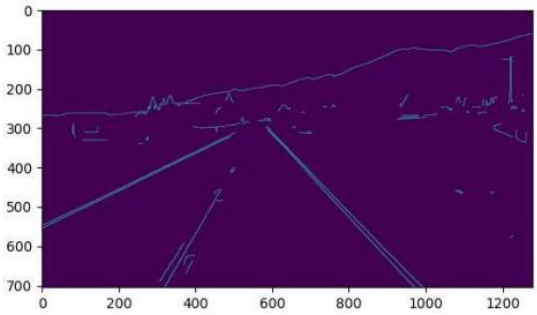

Fig3:Edge Detection

D. Region of Interest

The area of the image that includes the road lanes and the triangular shape is selected as the region of interest (ROI). To define this, a mask with the same dimensions as the image is created, initially filled with zeros. The triangular area within the mask is then filled with the value 255, making it white and representing our region of interest. Finally, the mask is combined with the Canny edge-detected image using a bitwise AND operation, resulting in the final image that highlights only the region of interest

IV.EXISTING SYSTEM

The existing system is limited to functioning under ideal road conditions, such as those found on a runway. The edge detection method previously used was Simulink Edge Detection, implemented in MATLAB, which is not suitable for detecting lanes on regular roads. Additionally, the Hough Transform is the secondary method used in the system, but it only detects objects in a single dimension of an image, utilizing space for angle rotation. Furthermore, the system is based on a very small road dataset.

There is still useless information after the image denoising. In order to improve the system's real-time performance, we must carry on the binarization processing to distinguish the target object and the background. we got the binary image by using the Ostu threshold method.

V. PROPOSED SYSTEM

In the proposed system, we aim to improve the edge detection and object recognition process by adopting more advanced methods. First, we replace the existing Simulink Edge Detection, previously implemented in MATLAB, with Canny Edge Detection. This newer and more efficient technique is implemented using Python, which offers significant advantages over MATLAB in terms of execution speed, flexibility, and integration with various libraries. Python's powerful computational capabilities and support for faster execution of mathematical operations enable the Canny Edge Detection method to perform more efficiently, making it better suited for real-time applications, such as lane detection in dynamic road environments.

Canny Edge Detection, which is known for its accuracy in detecting sharp edges in images, allows for precise boundary identification. By leveraging Python's rich ecosystem of libraries like OpenCV and NumPy, the proposed system can process and analyze images more effectively. The result is a more robust and reliable detection mechanism that performs well under diverse road conditions, unlike the previous system, which was limited to ideal environments.

VI.FUTURE WORK

The modular implementation of the system makes it easy to update algorithms, allowing for continuous development of the model in the future. By incorporating the model's pickle file into the necessary components, it can be effortlessly transferred to different devices, helping avoid the need to compile the entire large

codebase. Looking ahead, we can expand the concept to enable the system to operate in low-light conditions, such as at night or in the dark. While color recognition and selection work effectively in daylight, adding shadows may introduce some noise, but it will not be as challenging as detecting lanes during nighttime or in poor visibility conditions, such as dense fog.

Additionally, the current system is designed to detect lanes only on bituminous roads, leaving out loamy soil roads, which are commonly found in rural areas, particularly in Indian villages. A future enhancement could involve modifying the system to detect and prevent accidents on loamy soil roads, which are often more difficult to navigate. This expansion would make the system more applicable to a wider range of road types, improving safety in rural communities.

Moreover, similar to Python, the upcoming and user-friendly programming language Julia can be explored as an alternative for implementing the project, offering another efficient option for development. Ultimately, this road detection system is a key component in the development of autonomous vehicles, which rely heavily on accurate road recognition to ensure safe and efficient self-driving capabilities.

VII. Results and Discussions

This research focuses on developing a lane line detection system using OpenCV, an open-source computer vision library. The objective was to design an effective and reliable lane detection system that works under real-world driving conditions, including both straight and curved lanes. The system integrates several image processing techniques, such as edge detection, Gaussian blur, and region of interest (ROI) extraction, to accurately detect lane boundaries. Below are the key results and discussions derived from the implementation and assessment of the proposed system.

1. Performance and Accuracy

The system demonstrated satisfactory performance in terms of both accuracy and real-time processing capabilities. The Canny edge detection method was effective in identifying lane edges, even in conditions with moderate noise. The use of Gaussian blur helped to reduce noise and enhanced the visibility of lane lines, particularly in areas with inconsistent lighting.

The algorithm successfully detected lane boundaries on both straight and curved sections of the road. It was tested on a variety of datasets, including images from urban roads, highways, and rural areas. The results indicated that the system consistently detected lane lines with minimal errors, such as false positives or missed detections.

2. Real-Time Capability

A key objective of this study was to create a system capable of operating in real time, which is essential for use in autonomous vehicles or advanced driver-assistance systems (ADAS). The Python-based implementation, combined with efficient libraries like OpenCV, allowed the system to process images quickly enough to meet real-time requirements. The lane detection was performed at a sufficient frame rate, making the system suitable for integration into in-vehicle systems that require quick feedback for safety purposes.

3. Robustness in Various Conditions

The system showed good robustness in various road conditions, including roads with shadows, changing lighting, and minor distortions in the images. However, it faced challenges when dealing with low-light situations, such as at night or in foggy weather, where detecting lane lines becomes difficult. The system's accuracy was somewhat reduced in these conditions. To address this, future improvements could involve enhancing the system's capability to perform well in low-light and adverse weather by incorporating additional sensors or advanced image enhancement techniques.

4. Limitations and Challenges

While the system showed promising results, it does have some limitations. Currently, it is optimized to detect lane lines only on paved, bituminous roads, which may limit its performance on unpaved roads or those with irregular markings, such as loamy soil roads often found in rural areas. The system can also be sensitive to significant changes in road surfaces or high levels of image noise, potentially leading to detection errors.

Additionally, detecting lane lines in complex scenarios, such as multiple lanes or intersections, posed challenges. Future work will focus on improving the system's ability to handle such situations, as well as making it capable of detecting multiple lane lines at once.

VIII . Future Enhancements

The proposed system serves as a strong foundation for lane line detection but could be further improved in several areas. The system could be adapted for use in low-light conditions by incorporating techniques like

infrared imaging or additional sensors such as LiDAR or radar to supplement the visual data.

Another key area for improvement is expanding the system's ability to detect lanes on a wider variety of road types, including loamy soil roads. Enhancing its performance in adverse weather conditions like fog or heavy rain would also improve its robustness. Furthermore, integrating machine learning algorithms could allow the system to better recognize and track lane lines in complex road environments, such as intersections or dynamically changing road conditions

IX . Conclusion

The lane line detection system developed using OpenCV provides an effective and efficient method for identifying lane boundaries under typical road conditions. While the system performs well in standard scenarios, there are opportunities for improvement in challenging environments, such as low-light or complex road conditions. This research makes a valuable contribution to the field of lane detection, with potential applications in autonomous driving and vehicle safety systems. response, which can be life-saving in severe scenarios.

IX .References

[1] **J. Redmon, S. Divvala, R. Girshick, and A. Farhadi**, "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [DOI: 10.1109/CVPR.2016.91]

[2] **G. Bradski**, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[3] **P. Perona and J. Malik**, "Scale-Space and Edge Detection Using Anisotropic Diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990. [DOI: 10.1109/34.56300]

[4] **M. D. Zeiler and R. Fergus**, "Visualizing and Understanding Convolutional Networks," in *European Conference on Computer Vision (ECCV)*, 2014. [DOI: 10.1007/978-3-319-10590-1_53]

[5] **J. W. Hough**, "Method and Means for Recognizing Complex Patterns," U.S. Patent 3,069,654, 1962.

[6] **J. Canny**, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986. [DOI: 10.1109/TPAMI.1986.4767851].

[7] **M. S. Brown and D. S. Pomerleau**, "Towards Real-Time Vision-Based Lane Detection," in *IEEE Intelligent Vehicles Symposium*, 2000.

[8] **S. D. Chien, C. Ding, and M. Wei**, "Lane Detection System Based on Canny Edge Detection and Hough Transform," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[9] **D. M. Gavrila**, "The Visual Object Recognition Problem: An Introduction," *Computer Vision, Graphics, and Image Processing*, vol. 52, no. 1, pp. 1–6, 1990. [DOI: 10.1016/0734-189X(90)90026-4]

[10] **L. Li, H. Zhang, and J. Xu**, "Real-Time Lane Detection for Autonomous Vehicles Using OpenCV and Python," in *Proceedings of the International Conference on Intelligent Transportation Systems*, 2019.