Comprehensive Journal of Science
Volume (10), Issue (37), (NOV. 2025)
SICST2025, www.sicst.ly
ISSN: 3014-6266

مجلة العلوم الشاملة
عدد خاص بالمؤتمر الدولي الثالث للعلوم والتقنية
المجلد (10)، العدد (37)، (نوفمبر2025)
ردمد: 3014-6266

LANE LINES DETECTION ALGORITHM FOR SELF-DRIVING CARS

Abdulwahhab Aldoumani[1], and Muamer Hawej[2]*

[1] Electrical and Computer Engineering, Libyan Academy for Higher Studies, Tripoli, Libya,

[2]Department of Computer and Electrical Engineering, Elmergib University, Al-khomes, Libya,

Received: 30-09-2025; Revised: 10-10-2025; Accepted: 31-10-2025; Published: 25-11-2025

Abstract

A major reason for many traffic accidents is due to driver distraction and speeding. To assist drivers in their daily routine of operating an automobile, there has been significant research into developing Advanced Driver Assistance Systems (ADAS). ADAS systems will be able to observe and react to their surroundings so they can help prevent collisions from occurring and/or aid in navigating through difficult situations. One of the most important components of ADAS systems is the ability to recognize lanes on a roadway. Recognizing lanes is key to lane departure warnings, autonomous vehicle navigation, and collision avoidance.

The purpose of this paper is to present a robust lane detection algorithm for detecting lane lines that utilize an optimized version of the Canny edge detection process and the Hough Transform to accurately identify the location of lane boundaries in various types of environmental conditions. The proposed methodology consists of three main phases: pre-processing, feature extraction, and lane recognition. In the pre-processing phase, the raw input roadway image taken by a front facing camera is first converted to grayscale, and then processed using a series of Gaussian and Sobel filters to extract edge characteristics from the image. The Canny edge detection process is utilized to isolate the edges associated with the lane markings, and then a Region of Interest (ROI) is defined in order to limit the amount of processing time required to identify lane markings by eliminating unnecessary parts of the image, including the sky and roadside objects. In the final phase, the Hough Transform is used inside the ROI to locate and map the detected lane markings.

Keywords— Lane detection, Advanced Driver Assistance Systems (ADAS), Canny Edge Detection, Hough Transform, Computer Vision, Autonomous Vehicles.

1. Introduction

The rapid advancement of the automobile industry and the expansion of road transportation systems have significantly increased the demand for enhanced road safety. Numerous studies indicate that a large proportion of traffic accidents occur due to driver inattention, distraction, or delayed reaction at intersections. Among these incidents, those resulting in fatalities are often linked to excessive speed and inadequate situational awareness. To address these challenges, the development of Advanced Driver Assistance Systems (ADAS) has become a central focus of automotive research, aiming to improve driver awareness, reduce human error, and enhance overall traffic safety [1].

Over the past decade, ADAS technologies have evolved rapidly, with many modern vehicles now integrating systems such as Automated Emergency Braking (AEB), Lane Departure Warning (LDW), and Adaptive Cruise Control (ACC). These systems assist drivers by providing timely warnings and corrective actions to prevent collisions and maintain proper lane positioning [2]. A major challenge, however, lies in developing reliable and cost-effective ADAS solutions capable of functioning accurately in complex and dynamic driving environments.

Lane detection serves as a fundamental component of vision-based driver assistance systems and autonomous vehicles. It enables applications such as lane keeping, navigation, and forward collision avoidance. However, reliable lane detection remains challenging due to varying illumination, shadows, occlusions by other vehicles, and degraded lane markings [3]. Numerous methods have been proposed to overcome these issues using image processing and pattern recognition techniques [4]–[6].

This paper proposes a robust lane detection algorithm designed to identify left and right lane boundaries under diverse environmental conditions. The system employs optimized Canny edge detection and Hough Transform methods integrated into a three-stage framework consisting of preprocessing, feature extraction, and lane recognition. Real-time images captured by a forward-facing camera are processed to generate a reliable lane model. The proposed approach effectively detects lane markers on straight and curved roads with varying lane types—solid, dashed, white, or yellow—and performs well under challenging conditions such as shadows, glare, and road stains [7].

2. Background

2.1 Sensors

ADAS rely on multiple sensors to perceive the vehicle's environment. Ultrasonic sensors detect nearby obstacles, making them suitable for low-speed maneuvers such as parking. LiDAR emits laser pulses to measure distances based on the time of flight, enabling both short- and long-range detection. LiDAR performance can be affected by reflections and adverse weather. Radar sensors emit radio waves to detect object distance and speed using the Doppler effect. Compared to LiDAR, radar provides faster scanning for overall road monitoring but with lower resolution. Camera sensors offer high-resolution imagery at lower cost, though they require advanced post-processing for tasks like object detection and scene interpretation.

Because each sensor has unique strengths and limitations, sensor fusion is commonly employed to integrate data, improving reliability and robustness for ADAS, though at the cost of increased hardware and computational demands.

2.2Computer Vision

Computer vision enables vehicles to interpret surroundings from camera and sensor data, supporting tasks such as traffic light recognition, lane detection, and obstacle avoidance. A typical vision pipeline includes image acquisition, preprocessing, feature extraction, analysis, and decision-making, leading to vehicle control actions.

Fig.1 shows the general computer vision processing pipeline.

Fig. 1. General computer vision processing pipeline

Edge detection identifies points where image brightness changes sharply, forming curves that often correspond to object boundaries. It is fundamental for feature detection in lane detection, reducing irrelevant data and highlighting critical structures [8][9].

Region of Interest (ROI) focuses processing on areas likely to contain lanes, reducing computational load and improving detection accuracy [10]. Proper ROI selection is crucial, as poorly defined regions can lead to missed or inaccurate lane detection.

3. Theory

3.1 Image Pre-processing

Lane detection is affected by environmental noise, lighting variations, and image acquisition artifacts, which can blur lane edges and reduce detection accuracy. Image pre-processing enhances lane features and suppresses noise, providing a robust foundation for subsequent detection steps. The process of image pre-processing is shown in Fig. 2.

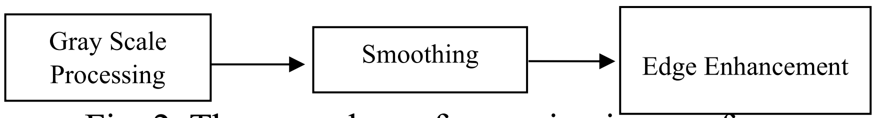
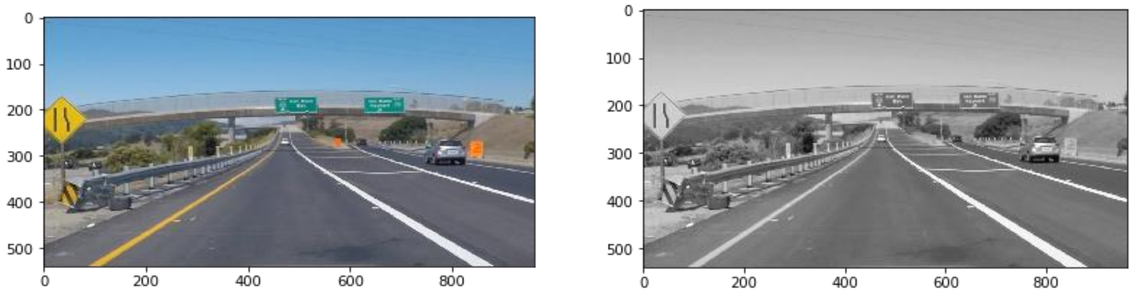


Fig. 2. The procedure of preparing images for processing

Color images consist of red (R), green (G), and blue (B) components, each ranging from 0 to 255. For real-time applications, images are converted to grayscale to reduce memory usage and computational load [11]. A weighted average accounts for human visual sensitivity:

$$g(x, y) = 0.299R + 0.587G + 0.114B \qquad (1)$$

Fig. 3 shows image conversion from RGB to gray scale color space.



(a) Original colored image (b) Grayscale image

Fig.3. Conversion from RGB to gray-scale color space

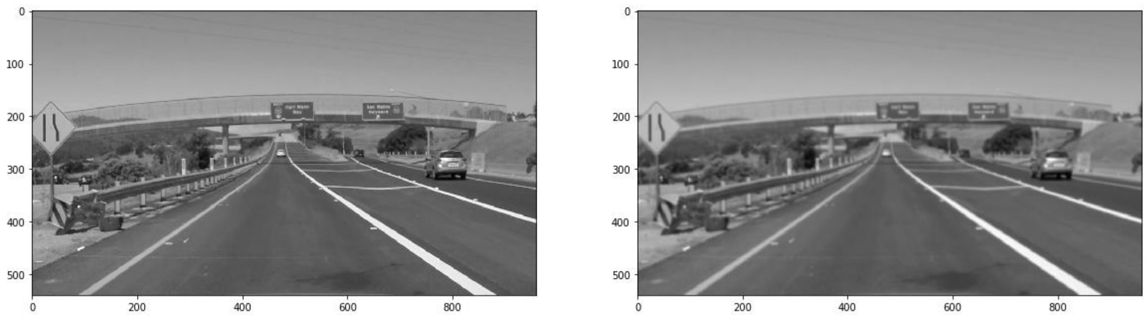
3.2 Canny Edge Detection

Edge detection reduces image data while preserving structural information for lane detection. The Canny algorithm [12] operates in five steps: smoothing, gradient computation, non-maximum suppression, double thresholding, and edge tracking by hysteresis.

A. Smoothing

There will always be some noise in any picture taken with a camera. Noise reduction is necessary to keep noise and edges from getting mixed up. So, the first step is to use a Gaussian filter to smooth the image. The kernel of a Gaussian filter with a standard deviation of σ=1.4 is calculated as shown in (2). Fig.4 shows how this filter affects the test image when it is used to smooth it out.

$$B = \frac{1}{159}\begin{pmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{pmatrix} \qquad (2)$$



(a) Gray-scale image image (b) Smoothed

Fig.4. The initial grayscale image is blurred with a Gaussian filter to reduce noise.

B. Gradients

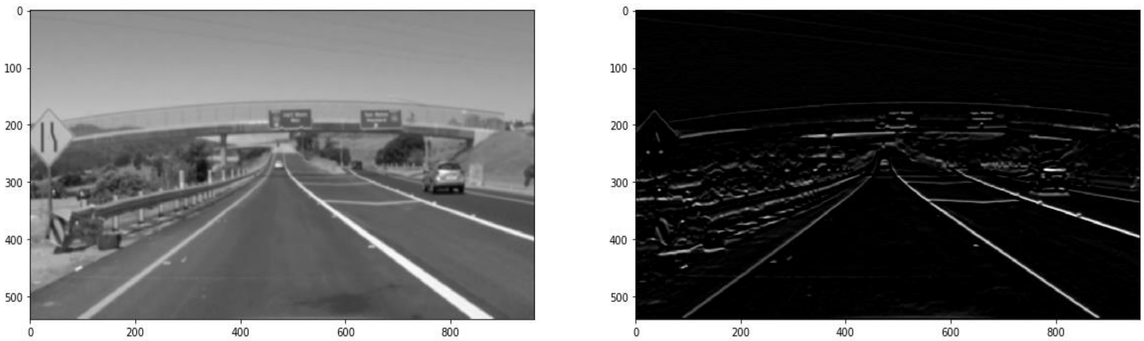
The Canny algorithm finds edges by looking for places where the grayscale intensity of the image changes the most. You can find these areas by looking at the image's gradients. The Sobel operator finds the gradients for each pixel in the smoothed image. The first step is to use the kernels shown in (3) to find the gradient in both the x- and y-directions.

$$\boldsymbol{S_{Gx}} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$
$$\boldsymbol{S_{Gy}} = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \tag{3}$$

Gradient magnitudes (also referred to as edge strengths) of images after smoothing can then be computed using the Pythagorean theorem for calculating the distance between two points in space (illustrated as (4) & (5)), as an Euclidean distance metric; the edge strengths calculated are compared to the smoothed image in Fig. 5.

$$|G| = \sqrt{{G_x}^2 + {G_y}^2} \tag{4}$$

$$\theta = \tan^{-1}\frac{|G_y|}{|G_x|} \tag{5}$$

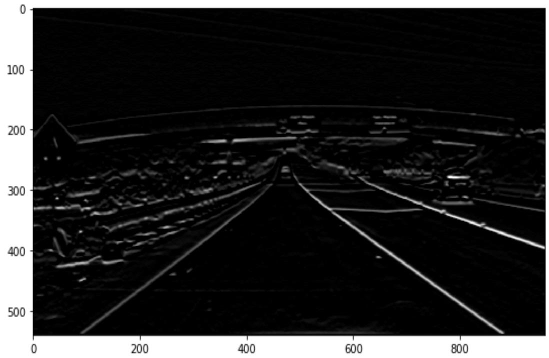


(a) Smoothed image (b) Gradient image

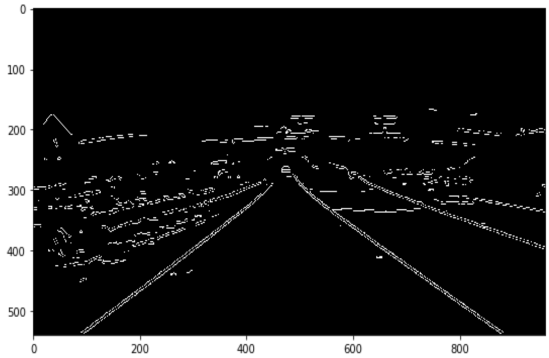
Fig.5. Gradient magnitudes in the smoothed image are determined by applying Sobel operator

C. Non-Maximum Suppression

The purpose of this step is to take the "fuzzy" edges from the gradient magnitude map and turn them into "sharp" edges. For each point, the value is checked to determine if it is a local maximum in the direction of the previously computed gradient as shown in Fig. 6.



(a) Gradient values



(b) Non-maximum suppression

Fig.6. Non-maximum suppression. Edge-pixels are only preserved

D. Edge tracking by hysteresis thresholding

Edges with high intensity are considered "strong" edges and can be added directly to the final edge image. Edges with low intensity are only added if those edges connect to a strong edge. The basis for this decision is that it is unlikely that the noise and minor variations present would have enough magnitude to provide a substantial benefit (assuming the proper setting of the threshold values), as shown in Fig. 7.

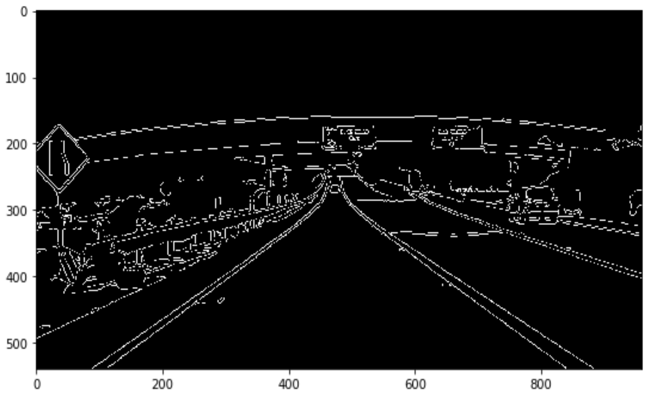


Fig.7. Edge tracking and final output, weak edges connected to strong edges.

3.3 ROI Extraction

The camera attached to the vehicle was located with a certain tilt angle at the middle of the top of the smart car, therefore the detected image included non-relevant background data such as the sky, trees, street signs and hillsides at either side of the road. As indicated by the fact that approximately two thirds of the detected image is represented as the ROI (Region Of Interest), the ROI will be able to remove unnecessary calculations and decrease the time required for the subsequent image processing phase. An example of an extracted ROI is shown in Fig. 8.

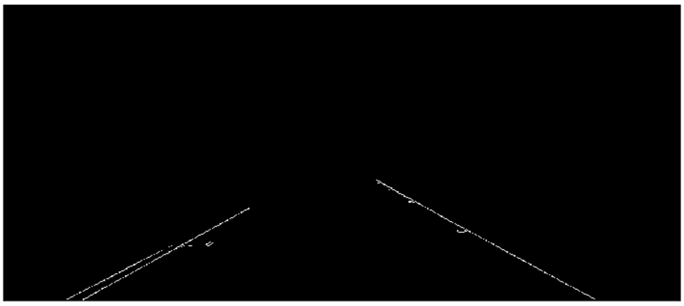


Fig.8. The extracted ROI.

3.4 Hough Transform for Line Detection

While edge detection reduces image data, representing lane lines using the Hough Transform further abstracts structural information. Lines can be expressed in polar coordinates:

$$r = x.cos\theta + y.sin\theta \qquad (7)$$

$$y = -\frac{cos\ \theta}{sin\ \theta}.x + \frac{r}{\sin\theta} \qquad (8)$$

Each edge point in the binary image maps to a sinusoidal curve in Hough space. The intersection of multiple curves indicates the presence of a line in the original image. This method enables reliable detection of lane lines even under partial occlusions or noise.

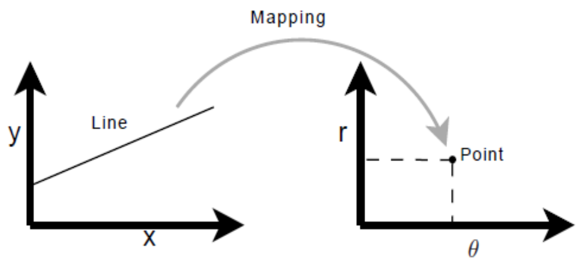
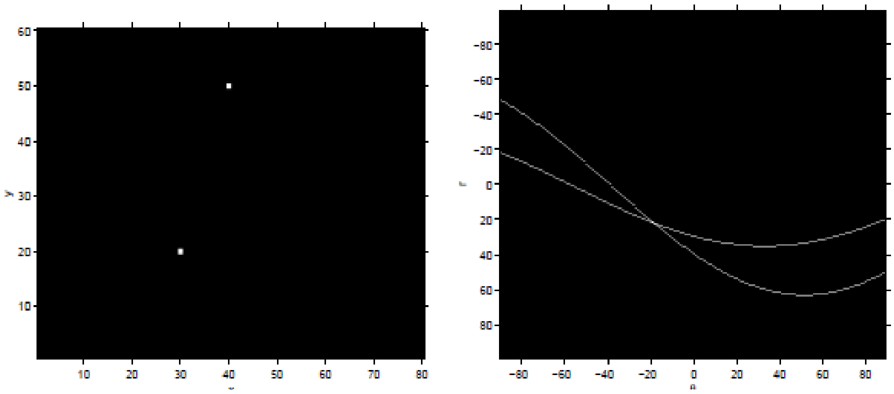


Fig. 9. Mapping of one unique line to the Hough space.

The Hough transform takes a binary edge map and identifies all edges from this edge map that are straight lines. In general, the idea behind the Hough transform is to convert each point of an edge in the edge map to every possible line that could go through that edge point.

Fig.10. illustrates this this for two points.



(a) Points p_0 *and* p_1 (b) All possible lines through p_0 and/or p_1

Fig. 10. Conversion of two points (p_0 and p_1) to two lines in the Hough space..

4. Modeling and Design

Fig.11 shows the Flow chart of the proposed algorithm:

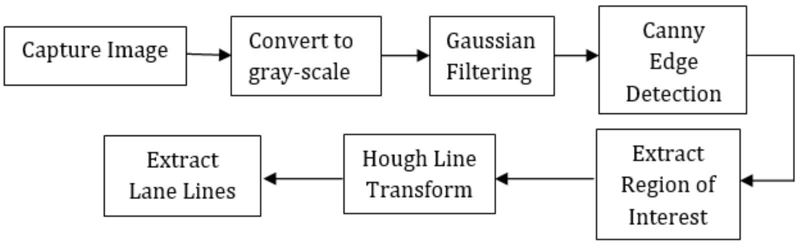


Fig. 11. Flow chart of the proposed algorithm

The region of interest is selected based on the probability of lines and lanes. Selecting an appropriate ROI diminishes process complexity and improves the overall speed and efficiency of the system. The ROI depends on the images and the presence of lanes within them; during the ROI selection process, extraneous elements such as the sky, buildings, bushes, and similar features are eliminated, concentrating exclusively on the roadways.

5. Experimental Results and Investigations

The performance of the proposed lane detection algorithm was evaluated on real-road video sequences captured by an on-board camera with a resolution of 960×540. The algorithm successfully detected and extracted both left and right lane markers. Edge detection effectively suppressed road noise caused by vehicles, signs, and other traffic elements, as shown in Fig. 12, while the Hough Transform accurately identified lane lines within a reduced region of interest to lower computational cost, as shown in Fig. 13. Adjustments to threshold values allowed robust detection under complex conditions. Overall, the results demonstrate that the proposed method provides reliable and efficient lane edge detection for real-time applications.

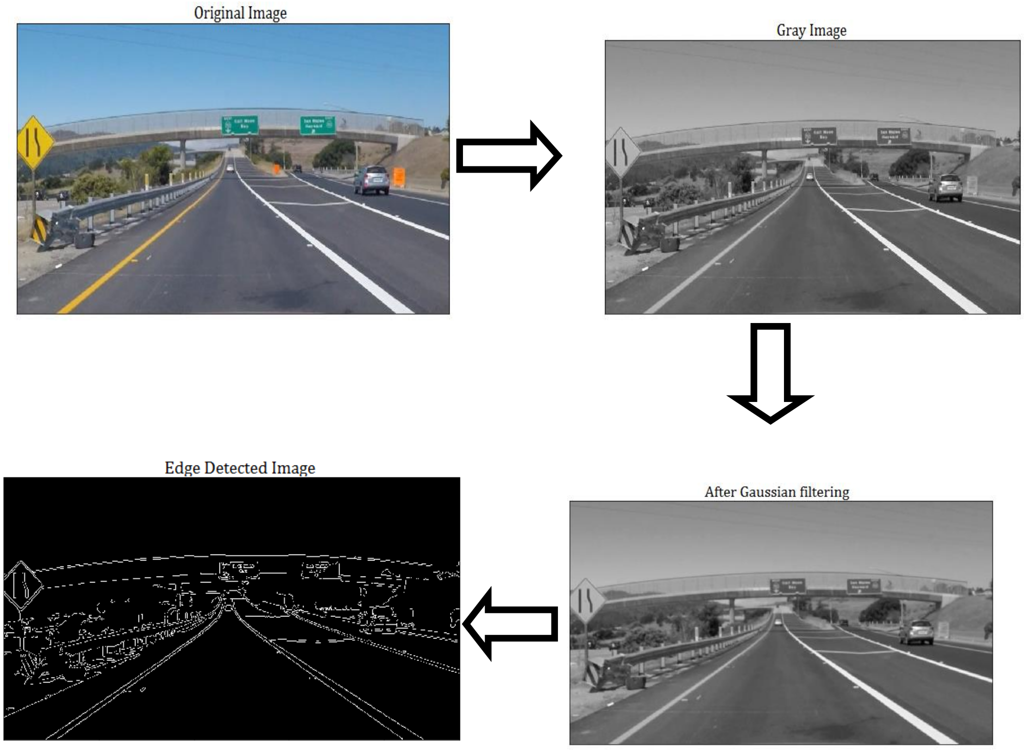


Fig. 11. The test result of edge detection algorithm

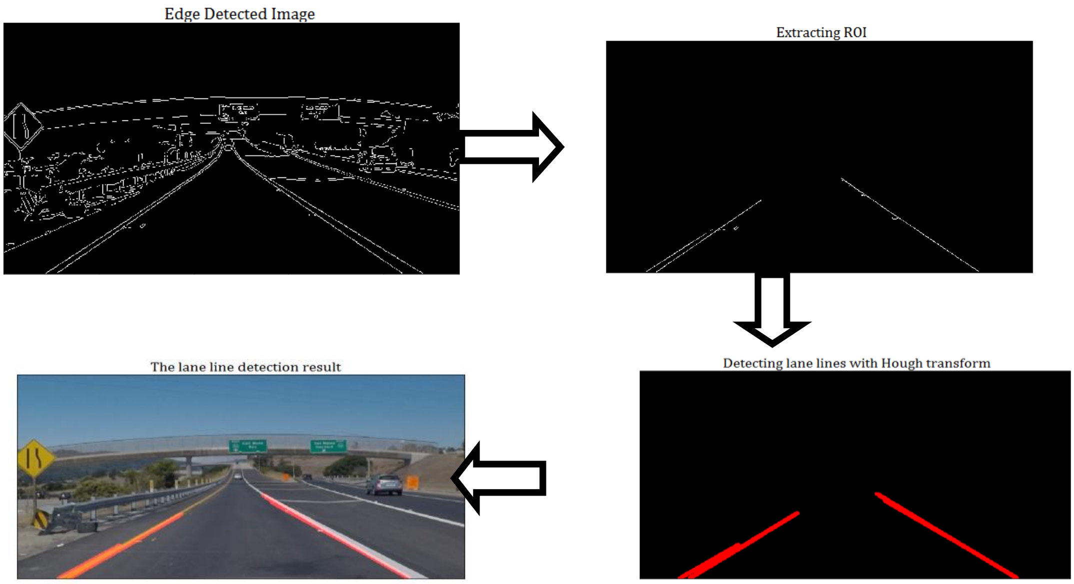


Fig. 12. The test result of lane lines detection algorithm based on Hough transform

6. Consclusion

This work provides a succinct investigation of a lane edge detection method that makes use of the Canny Algorithm. The proposed method reduces the number of lane parameters, lowering computational time and improving detection efficiency. Accuracy was evaluated using the correct detection rate on datasets of varying sizes (300–1500 images). Comparisons with basic preprocessing methods and ROI selection based solely on lane color demonstrate that the proposed preprocessing and adaptive ROI selection significantly enhance lane detection performance and robustness. We analyze the characteristics of the road image. The binary image offers essential data for lane detection. Edge detection is essential for lane identification, with the Canny algorithm acknowledged as a premier edge detection method. The method for detecting lane edges is presented through the integration of these features. The subsequent steps for executing lane edge detection are outlined. We detect linear lanes employing the Hough transform. The proposed methodology has been empirically validated. The experimental results demonstrate that the suggested method for lane edge detection is effective. The methodology of lane tracking will be examined in the imminent future. In future projects, the lane detection will use position tracking to make the computer's work even easier. By adding real-world positioning to the picture, you can figure out how far away you are and how long it will take to leave. Using the combined parameters, a new system for warning drivers when they leave their lane will be made.

References

[1] Y. Fan and W. Zhang, “Traffic sign detection and classification for Advanced Driver Assistant Systems,” in *Proc. 12th Int. Conf. Fuzzy Systems and Knowledge Discovery (FSKD)*, 2015.

[2] R. Sheri, N. Jadhav, R. Ravi, A. Shikhare, and S. Sannakki, “Object detection and classification for self-driving cars,” *Int. J. Eng. Techniques*, 2018.

[3] M. Bertozzi and A. Broggi, “GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection,” *IEEE Trans. Image Process.*, 1998.

[4] J. W. Lee and J. S. Cho, “Effective lane detection and tracking method using statistical modeling of color and lane edge-orientation,” *Adv. Inf. Sci. Serv. Sci.*, 2010.

[5] P. Foucher, Y. Sebsadji, J. P. Tarel, P. Charbonnier, and P. Nicolle, “Detection and recognition of urban road markings using images,” in *Proc. IEEE Int. Conf. Intelligent Transportation Systems*, 2011.

[6] D. J. Kang and M. H. Jung, “Road lane segmentation using dynamic programming for active safety vehicles,” *Pattern Recognit. Lett.*, 2003.

[7] Q. Li, N. Zheng, and H. Cheng, “Springrobot: A prototype autonomous vehicle and its algorithms for lane detection,” *Intell. Transp. Syst.*, 2004.

[8] S. E. Umbaugh, *Digital Image Processing and Analysis: Human and Computer Vision Applications with CVIPtools*. CRC Press, 2004.

[9] M. V. G. Aziz, A. S. Prihatmanto, and H. Hindersah, "Implementation of lane detection algorithm for self-driving car on Toll Road Cipularang using Python language," in *Proc. 4th Int. Conf. Electric Vehicular Technology (ICEVT)*, 2017.

[10] J. He, H. Rong, J. Gong, and W. Huang, "A lane detection method for lane departure warning system," in *Proc. Int. Conf. Optoelectronics and Image Processing*, 2010.

[11] Y. Li, L. Chen, H. Huang, X. Li, W. Xu, L. Zheng, and J. Huang, "Nighttime lane markings recognition based on Canny detection and Hough transform," in *Proc. IEEE Int. Conf. Real-time Computing and Robotics*, 2016.

[12] J. F. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, 1986.