Scientific Insights
https://ojs.achpub.com/SI

ACADEMIC HORIZON PUBLISHER

Article

An Improved Canny–Hough Algorithm for Lane Detection

Xiang-sen Ning, Ai-juan Li *, Jia-qi Chen, Li-wen Wang, Qiang Zheng

School of Automotive Engineering, Shandong Jiaotong University, Jinan 250357, China
* Correspondence: liaijuan@sdjtu.edu.cn

Received: 4 March 2026; **Revised:** 7 April 2026; **Accepted:** 14 April 2026; **Published:** 8 June 2026

Abstract: Lane detection is a key perception task in autonomous driving. However, traditional lane detection methods based on edge extraction and Hough transform often struggle to balance detection accuracy and real-time performance under complex driving conditions. To address this issue, this study proposes a lightweight lane detection framework based on an improved Canny–Hough algorithm. The proposed method is designed as a coordinated pipeline that enhances edge extraction, adaptive thresholding, and constrained Hough voting in a unified manner. Specifically, a 3×3 multi-direction gradient operator, interpolation-based non-maximum suppression, and Otsu-based adaptive thresholding are introduced to improve the quality and stability of lane-edge extraction. In addition, region-of-interest masking and polar-angle restriction are incorporated into the Hough transform to reduce non-lane interference and computational redundancy. Experimental results on the TuSimple dataset show that the proposed method achieves a recognition rate of 91.72% and an average detection time of 32.45 ms. Compared with the traditional Hough-based method, the recognition rate is improved by 13.30 percentage points (16.95% relative to the baseline), while the detection time is reduced by 39.11%. Ablation and comparative experiments further verify that the proposed framework achieves a favorable trade-off between detection accuracy and efficiency. The proposed method provides an effective and lightweight solution for lane detection, with promising applicability under the tested TuSimple-based forward-view setting in CPU-based and resource-constrained deployment scenarios.

Keywords: image preprocessing; lane detection; Canny edge detection; Hough transform

1. Introduction

Lane detection is a core perception function for autonomous driving, providing geometric constraints for downstream planning and control. Traditional vision pipelines remain attractive for embedded deployment because they are lightweight and interpretable, yet their robustness often degrades under complex illumination, worn markings, and background clutter.

Traditional lane detection methods mainly rely on handcrafted geometric cues, edge extraction, Hough transform, and curve or line fitting [1]. These methods remain attractive for resource-limited platforms because they are lightweight, interpretable, and do not depend on large-scale annotated datasets or high-performance computing hardware. However, their performance is often sensitive to edge quality and tends to degrade in complex scenes with illumination variation, background clutter, worn lane markings, and noise interference.

Recent learning-based lane detection methods have achieved substantial progress and can generally be categorized into segmentation-based, anchor-based, and transformer-based architectures [2]. Representative studies include SCNN [3], which improves lane detection through spatial message passing in a segmentation framework, CLRNet [4], which enhances anchor-based lane detection by jointly exploiting high-level semantic information and low-level detailed features, and

Copyright ©2026 Xiang-sen Ning, *et al*.
DOI: https://doi.org/10.64897/si.2026v2i1.003
This is an open-access article distributed under a CC BY license
(Creative Commons Attribution 4.0 International License)
https://creativecommons.org/licenses/by/4.0/

Laneformer [5], which introduces a transformer-based architecture to better capture long-range lane-shape and contextual relationships. Although these methods can achieve high detection accuracy, they typically require extensive labeled data and considerable training and inference resources, which may limit their applicability in resource-constrained engineering scenarios. In related fields, transform-based geometric extraction and automatic surface-feature identification have also been investigated in LiDAR and remote-sensing applications, providing useful methodological insight for structured-feature detection under noisy and complex scene conditions [6–8].

However, existing approaches still face three practical limitations in real-world driving scenes. First, traditional Hough-based lane detection methods are highly dependent on the quality of edge extraction, and their performance tends to degrade in the presence of noise, background clutter, and illumination variation, often leading to fragmented lane boundaries and redundant detections. Second, many existing improvements focus on a single stage of the detection pipeline, such as edge extraction or line fitting, while lacking coordinated optimization across the full detection process; as a result, improvements in accuracy and efficiency are often not achieved simultaneously. Third, although deep-learning-based methods have demonstrated strong detection capability, they usually rely on large-scale annotated datasets and considerable computing resources, which may limit their applicability in resource-constrained engineering scenarios.

To address these issues, this study proposes a lightweight lane detection framework based on an improved Canny–Hough pipeline. The proposed method is designed as a coordinated full-pipeline optimization strategy for CPU-based and resource-constrained deployment scenarios. Rather than refining only a single stage of the detection process, the framework jointly enhances edge extraction, adaptive threshold selection, and constrained Hough voting within a unified classical lane-detection pipeline. Through task-oriented integration of complementary modules and system-level coordination across the full pipeline, the proposed design aims to improve both detection accuracy and computational efficiency under practical forward-view driving conditions.

The main contributions of this study are summarized as follows:

(1) A coordinated edge-enhancement module is developed by integrating a 3×3 multi-direction gradient operator, interpolation-based non-maximum suppression, and Otsu-based adaptive thresholding. This module improves gradient estimation, edge localization, and threshold robustness under varying scene conditions.

(2) A lightweight Hough voting strategy is constructed by combining region-of-interest masking with polar-angle restriction. This design reduces invalid voting, suppresses non-lane interference, and improves computational efficiency in the line-detection stage.

(3) A classical lane-detection pipeline is re-optimized for CPU-based and resource-limited scenarios, and its effectiveness is systematically validated on the TuSimple dataset through quantitative evaluation, ablation experiments, and comparative analysis. The results show that coordinated full-pipeline design can achieve a favorable trade-off between detection accuracy and runtime performance.

The remainder of this study is organized as follows. Section 2 presents the proposed lane detection approach based on the Canny–Hough algorithm and describes its main components. Section 3 reports the experimental results, including validation, ablation studies, and comparative evaluations. Section 4 discusses the main findings, limitations, and future research directions. Section 5 concludes the paper.

2. Methodology

First, the input image is preprocessed using grayscale conversion and Gaussian denoising. Next, edges are extracted using the improved Canny operator. Compared with the traditional Canny algorithm, the proposed method enlarges the gradient operator and increases the number of gradient directions when computing the gradient magnitude and orientation. Interpolation-based non-maximum suppression replaces the conventional discrete approximation. For threshold selection, an Otsu-based method is used instead of manually tuned thresholds. Finally, a polar angle constraint is incorporated into the Hough transform to restrict voting to a feasible angular range. The overall pipeline is illustrated in Figure 1.

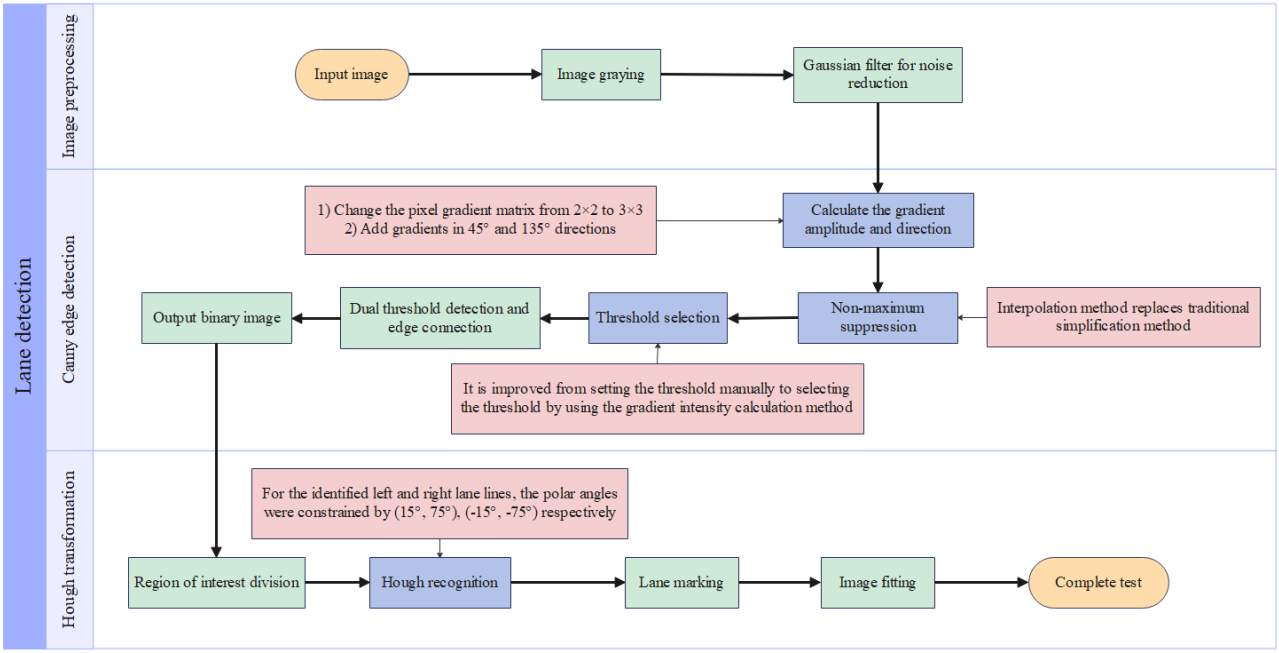


Figure 1. Lane detection process in this study

2.1 *Dataset Description*

The proposed method was evaluated on the TuSimple [9] lane detection dataset, a widely used public benchmark for lane detection. The dataset contains 6408 annotated road images collected in real driving environments and covers multiple scene categories relevant to lane detection tasks. In this study, 1923 valid images covering the major scene types defined in the dataset were selected for experiments to ensure sample diversity and representativeness. All images were processed at the original resolution to preserve lane-marking details and ensure experimental reproducibility.

The ground-truth data used in this study were obtained from the official TuSimple annotations rather than manually labeled by the authors. For each image, the annotated lane coordinates were connected sequentially and rasterized into binary lane masks with the same spatial resolution as the input image. Pixels belonging to rasterized lane annotations were assigned a value of 1, and all remaining pixels were assigned a value of 0. These masks were used as the reference standard for pixel-level evaluation.

From the 6408 annotated TuSimple images, 1923 samples were retained according to predefined screening criteria, including annotation completeness, annotation consistency for lane-mask generation, and suitability for stable forward-view lane evaluation. The same subset was used for all locally reproduced classical baseline methods to ensure fair comparison. These screened samples were selected before quantitative evaluation and were not adjusted according to the experimental results. The final subset was constructed to retain the major scene categories represented in the original TuSimple dataset.

2.2 *Image Preprocessing*

After acquisition, the road image is first preprocessed to suppress irrelevant information and preserve lane-related features, thereby improving the robustness and reliability of subsequent detection [10].

2.2.1 *Grayscale Conversion of Road Images*

The grayscale conversion is performed using the standard luminance-weighted model shown in Equation (1), which is widely adopted in image preprocessing.

$$Gray(x,y) = 0.299 \times R(x,y) + 0.587 \times G(x,y) + 0.114 \times B(x,y) \tag{1}$$

where $Gray(x,y)$ denotes the grayscale intensity at pixel (x,y), and $R(x,y)$, $G(x,y)$, $B(x,y)$ denote the red, green, and blue channel intensities, respectively. The coefficients 0.299, 0.587, and 0.114 are the standard perceptual weights for the RGB channels. Equation (1) follows the standard luminance-weighted RGB-to-grayscale formulation commonly used in image preprocessing [11].

2.2.2 *Gaussian Filtering and Denoising*

To enhance edge stability before gradient computation, we apply Gaussian smoothing to the grayscale image. A 2D Gaussian kernel of size $(2k+1, 2k+1)$ is used to suppress high-frequency noise while preserving lane boundaries; in this work, $k = 1$, and the kernel weights follow Equation (2).

$$M(i,j) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i-k-1)^2 + (j-k-1)^2}{2\sigma^2}\right) \tag{2}$$

where $M(i,j)$ denotes the coefficient of the two-dimensional Gaussian kernel at position (i,j), k represents the half-width of the Gaussian kernel, and σ denotes the variance along the x-direction. Equation (2) follows the standard discrete two-dimensional Gaussian-kernel formulation used for image smoothing in classical vision preprocessing [11].

2.3 *Improved Canny Edge Detection*

The traditional Canny [12] algorithm uses only a 2×2 pixel gradient matrix, which limits edge localization accuracy. In addition, when a simplified method is adopted for non-maximum suppression, the extracted edges are often insufficiently precise. Moreover, in the traditional Canny algorithm, the high and low thresholds must be set manually, which reduces adaptability under varying image conditions. In this study, the traditional Canny algorithm is improved to address these limitations.

2.3.1 *Improved Gradient Operator*

A 3×3 gradient operator is adopted in this study. In addition to the original 0° and 90° gradient directions, 45° and 135° directions are incorporated, resulting in four directional gradient operator matrices (denoted as S_0, S_{45}, S_{90}, and S_{135}). The corresponding directional pixel gradient magnitudes are G_0, G_{45}, G_{90}, and G_{135}, whose specific forms are given in Equations (3)–(6). Subsequently, the gradient magnitude and orientation of each pixel are computed via these directional gradients, as expressed in Equations (7) and (8).

$$G_0 = S_0 * I = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * I \tag{3}$$

$$G_{45} = S_{45} * I = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & +1 & +2 \end{bmatrix} * I \tag{4}$$

$$G_{90} = S_{90} * I = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * I \tag{5}$$

$$G_{135} = S_{135} * I = \begin{bmatrix} 0 & +1 & +2 \\ -1 & 0 & +1 \\ -2 & -1 & 0 \end{bmatrix} * I \tag{6}$$

where I is the grayscale image matrix, and $*$ denotes a cross-correlation operation.

To avoid mixing directional responses, the gradient magnitude is computed using the L2-norm over the four filter responses:

$$G_{xy}(i,j) = \sqrt{G_0^2(i,j) + G_{90}^2(i,j) + G_{45}^2(i,j) + G_{135}^2(i,j)} \tag{7}$$

$$\theta_{xy}(i,j) = \arctan\left(\frac{\sqrt{G_{45}^2(i,j) + G_{135}^2(i,j)}}{\sqrt{G_0^2(i,j) + G_{90}^2(i,j)}}\right) \tag{8}$$

where $G_{xy}(i,j)$ denotes the gradient magnitude of pixel (i,j), and $\theta_{xy}(i,j)$ is its gradient angle. $\theta_{xy}(i,j)$ ranges from 0° to 180°, representing the angle relative to the positive x-axis. Here, $G_0(i,j), G_{45}(i,j), G_{90}(i,j), G_{135}(i,j)$ correspond to the gradient magnitudes of pixel (i,j) in the 0°, 45°, 90°, and 135° directions, respectively.

By introducing diagonal derivative operators (45° and 135°), the gradient estimation better approximates the continuous distribution of edge orientations, reducing the angular quantization error inherent in using only horizontal and vertical derivatives. This improves robustness when lane markings appear with arbitrary orientations due to road curvature and camera pose changes.

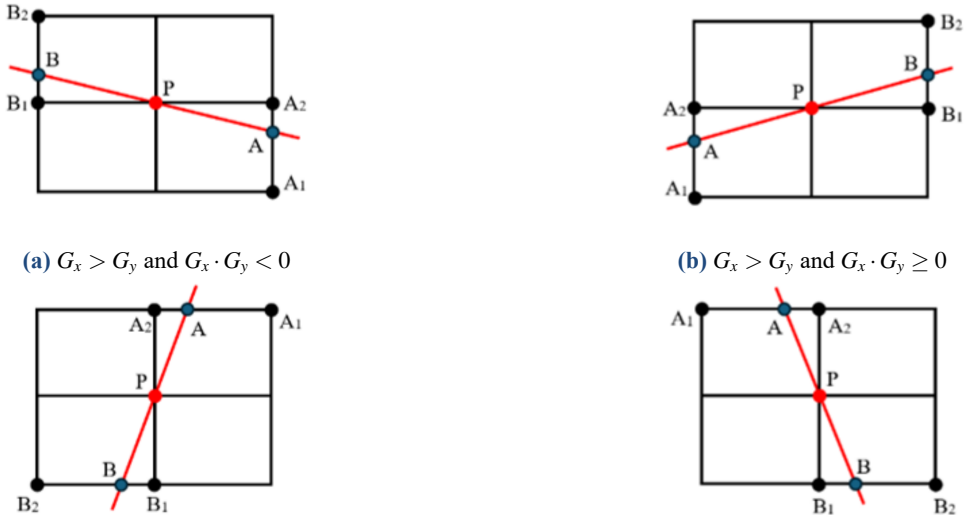
2.3.2 *Interpolation-Enhanced Non-Maximum Suppression*

To reduce interference from irrelevant boundary responses, non-maximum pixels are suppressed in the gradient map [13]. Instead of the conventional discrete non-maximum suppression, an interpolation-based method is adopted herein. The gradient calculations for the four cases are illustrated in Figure 2. The interpolated gradient magnitudes of two adjacent sub-pixel points are computed as shown in Equations (9) and (10). Compared with discrete neighbor selection, interpolation estimates the gradient magnitude at sub-pixel locations along the gradient direction, which reduces suppression errors caused by discretization. As a result, true lane boundaries are preserved as thinner and more continuous edges, while spurious responses are more effectively removed.

$$G_A = \omega G_{A1} + (1-\omega) G_{A2} \tag{9}$$

$$G_B = \omega G_{B1} + (1-\omega) G_{B2} \tag{10}$$

where ω denotes the weight. Specifically, ω is set to G_x/G_y when $G_x > G_y$, and to G_y/G_x when $G_x < G_y$. Here, G_x represents the gradient magnitude in the x-direction, while G_y denotes the gradient magnitude in the y-direction.



(a) $G_x > G_y$ and $G_x \cdot G_y < 0$ **(b)** $G_x > G_y$ and $G_x \cdot G_y \geq 0$

(c) $G_x < G_y$ and $G_x \cdot G_y < 0$ **(d)** $G_x > G_y$ and $G_x \cdot G_y \geq 0$

Figure 2. Gradient calculation in four cases

2.3.3 *Otsu-Based Double-Threshold Selection*

In this study, Otsu's method is used instead of manually selected thresholds. The inter-class variance is calculated using Otsu's criterion [14]. When the inter-class variance reaches its maximum, the corresponding gray level T is taken as the high threshold, and the low threshold is set to 0.4 times the high threshold. Otsu's criterion selects the threshold by maximizing inter-class variance, enabling automatic adaptation to global contrast changes. This reduces sensitivity to illumination variation and avoids over-detection and under-detection caused by manually tuned thresholds.

$$N = N_0 + N_1 \tag{11}$$

$$\mu = \mu_0 \frac{N_0}{N_0 + N_1} + \mu_1 \frac{N_1}{N_0 + N_1} = \frac{\mu_0 N_0 + \mu_1 N_1}{N} \tag{12}$$

$$\sigma_B^2 = \frac{N_0}{N_0 + N_1} \frac{N_1}{N_0 + N_1} (\mu_0 - \mu_1)^2 = \frac{N_0 N_1}{N^2} (\mu_0 - \mu_1)^2 \tag{13}$$

where N is the number of pixels in the entire image, μ is the average pixel gray value of the whole image, σ_B^2 is the between-class variance, N_0 is the number of pixels whose gray level is less than the gray level T, N_1 is the number of pixels whose gray level is greater than T, μ_0 is the average gray level of pixels with gray level less than T, and μ_1 is the average gray level of pixels with gray level greater than T. Equations (11)–(13) follow the standard Otsu threshold-selection criterion [15].

2.3.4 *Suppression of Isolated Weak Edges*

The value of each edge pixel in the image is examined. If the pixel value at a point is greater than the high threshold, it is marked as a strong edge; if it is less than the low threshold, it is suppressed. Pixels with values between the low and high thresholds are labeled as weak edge points. In the presence of noise, weak and strong edges may not be directly connected.

At this stage, each weak edge pixel is examined together with its eight neighboring pixels. If at least one neighboring pixel is a strong edge pixel, the weak edge pixel is retained as a true edge.

2.4 *Improved Hough Transform Detection*

The Hough transform is a feature extraction technique that can be used to detect objects with specific shapes, such as straight lines, circles, and ellipses [16]. From a geometric perspective, the equation of a straight line can be expressed as shown in Equation (14).

$$y = m_0 x + b_0 \tag{14}$$

where m_0 denotes the slope and b_0 denotes the intercept.

If the slope and intercept of each line are fixed and unchanging, then within the parameter space with m as the horizontal axis and b as the vertical axis, a unique point corresponding to each line can be found [17]. The transformation is as shown in Equation (15).

$$b_0 = -xm_0 + y \tag{15}$$

The parameter space defined by m and b constitutes the Hough space. To avoid the issue of undefined slopes for vertical lines in the Cartesian coordinate system, this study adopts a polar coordinate representation for lines. In polar coordinates, any point (p,q) on a line can be expressed as shown in Equation (16).

$$(p,q) = (\rho_0 \cos\theta_0, \rho_0 \sin\theta_0) \tag{16}$$

where ρ_0 denotes the perpendicular distance from the origin to the line, and θ_0 denotes the polar angle, defined as the angle between the normal and the positive x-axis.

If the angle between the original straight line and the x-axis is $90^\circ + \theta_0$, then its slope is shown in Equation (17).

$$\tan(90^\circ + \theta_0) = -\frac{1}{\tan\theta_0} = -\frac{\cos\theta_0}{\sin\theta_0} \tag{17}$$

If any point on the straight line is (x,y), then the slope is as shown in Equation (18).

$$\frac{y-p}{x-q} = \frac{y-\rho_0 \sin\theta_0}{x-\rho_0 \cos\theta_0} \tag{18}$$

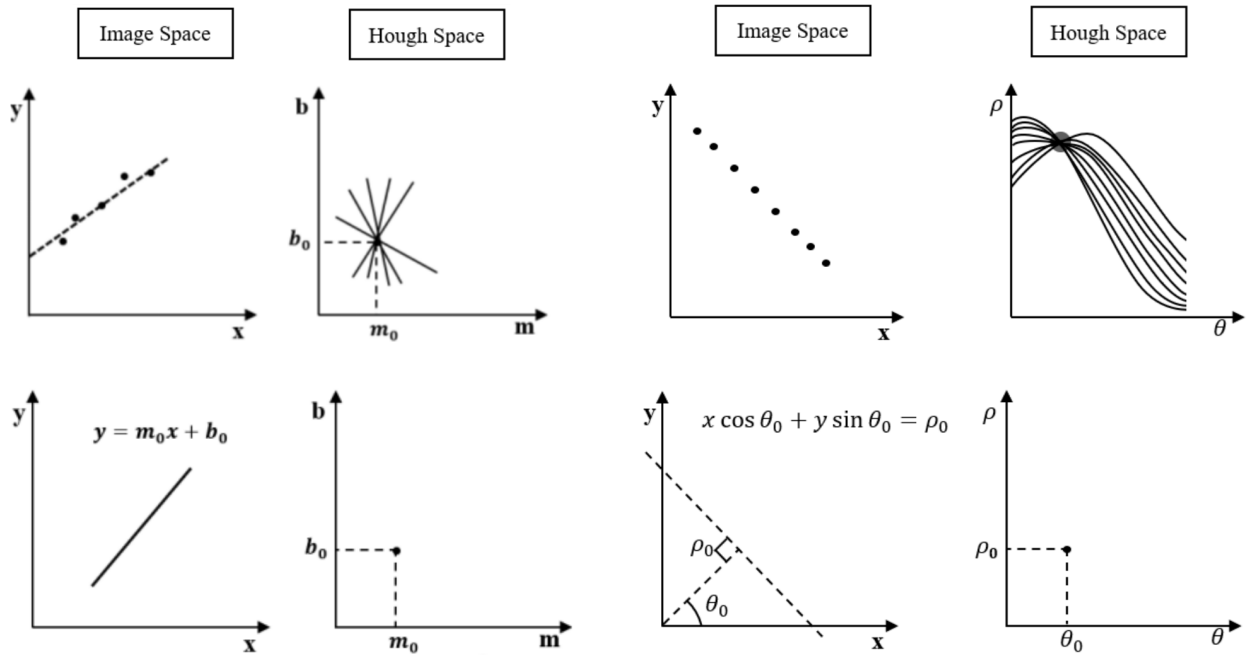
$$\rho_0 = x\cos\theta_0 + y\sin\theta_0, \quad \theta_0 \in [-90^\circ, 90^\circ] \tag{19}$$

To determine the polar-angle constraint ranges, adjacent annotated lane points in the selected TuSimple images were first connected to form local lane segments. The orientation of each segment was then transformed into the corresponding polar angle in Hough space. Statistical inspection of these angle distributions showed that most left-lane segments were concentrated in $[15^\circ, 75^\circ]$, while most right-lane segments mainly fell within $[-75^\circ, -15^\circ]$. Accordingly, these intervals were adopted as practical angular priors for the current experimental setting, with the aim of reducing invalid voting while preserving most valid lane candidates. It should be noted that these angular intervals are treated here as scene-dependent priors rather than universal constants, and their validity is associated with the forward-facing camera setup and highway-like viewpoint characteristics of the TuSimple dataset.

In the polar coordinate system, the steps of the Hough transform for identifying lane-lines are as follows. First, a two-dimensional accumulation array A is established in the parameter coordinate space. The range of the first dimension is $[-d, d]$, where d is the length of the diagonal of the image, and the range of the second dimension is $[-90^\circ, 90^\circ]$, which is initialized to 0. Each pixel point in the image coordinate space is then traversed, all corresponding (ρ_i, θ_i)

values are calculated using the Hough transform, and 1 is added to the array element $A(\rho_i, \theta_i)$ each time. Among all array elements, the two sets with the largest accumulated values are selected, and the corresponding line parameters are calculated. Corresponding mappings between image space and parameter space exist in both Cartesian and polar formulations of the Hough transform. The correspondence in plane Cartesian coordinates is shown in Figure 3(a), and the correspondence in polar coordinates is shown in Figure 3(b).

Equations (14)–(19) follow the standard Cartesian and polar representations used in the Hough transform.



(a) The correspondence in Cartesian coordinates **(b)** The correspondence in polar coordinates

Figure 3. The corresponding relationship between the original space and Hough space

Because the image contains a large number of edge points, the computational cost of the Hough transform is high. In the standard implementation, each edge point needs to be evaluated over all discretized angle values [18]. This leads not only to a high computational burden but also to increased storage requirements for the accumulator, which slows down the algorithm. To solve this problem, this study introduces the following improvements to the Hough transform:

RoI masking removes non-lane regions and retains only lane-related areas prior to Hough voting [19]. This not only improves detection speed but also effectively avoids interference from obstacles in non-lane areas. RoI masking encodes a geometric prior that lane markings mainly lie in the lower road region. By excluding unlikely areas, it reduces false positives from background structures and decreases the number of edge points participating in voting. To reduce computational cost, an RoI mask is applied prior to subsequent processing. Specifically, a binary mask M with the same size as the input image is initialized to zero and assigned a value of 255 within the predefined RoI region r_1. The mask is then used as a bitmap to gate pixel copying: pixels with $M \neq 0$ are retained, while pixels with $M = 0$ are suppressed, so the resulting image contains only the RoI information.

When a vehicle travels along lane-lines, the lane markings typically appear inclined in the captured image because of vehicle motion and viewpoint geometry. Consequently, it is necessary to filter out lines with excessively large or small slopes. In this study, a polar-angle constraint is introduced for lane detection, with separate constraint ranges applied to the left and right lane-lines. For the current TuSimple-based experimental setting, the polar angles of candidate lines in the left region of the image are constrained to $[15°, 75°]$, and those in the right region are constrained to $[-75°, -15°]$. These ranges are used as scene-dependent angular priors rather than universally optimal parameters. This method reduces

computational complexity to a certain extent and improves detection speed under the tested forward-view configuration. The implementation flowchart of the polar-angle constraint is shown in Figure 4. Restricting voting to feasible lane-angle intervals acts as a prior that suppresses accumulator peaks produced by non-lane edges, thereby reducing ambiguity in peak selection while also shrinking the search space.

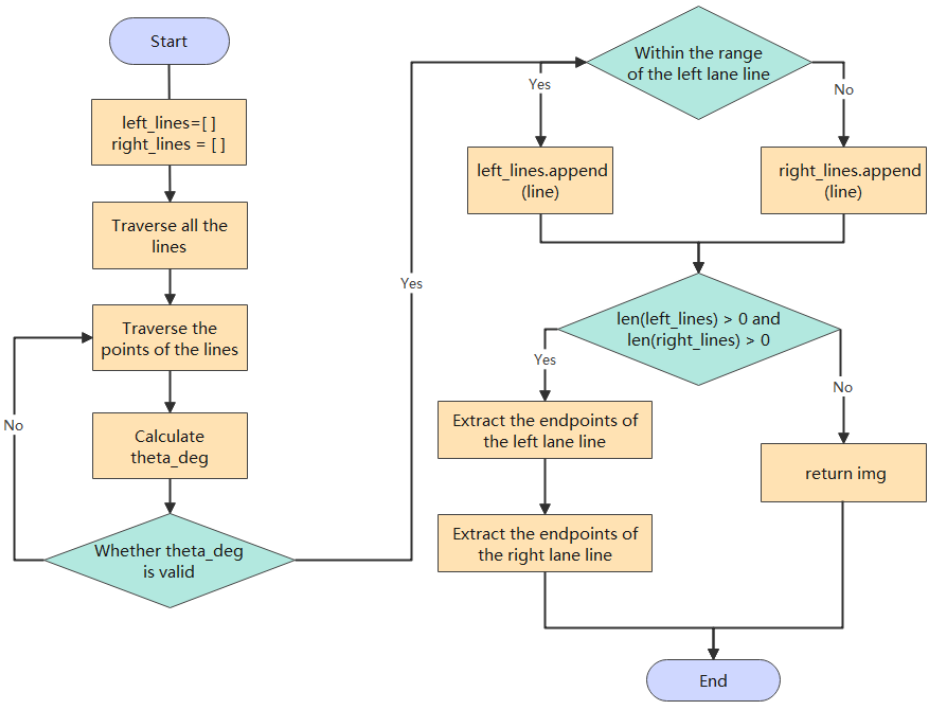


Figure 4. Polar angle constraint flowchart

Detected line segments are partitioned into left- and right-lane candidates according to their slope and spatial location. A representative lane-line for each side is then estimated by aggregating the parameters of the corresponding segment group, yielding a compact and stable representation for subsequent lane fitting and visualization.

Let $N = W \times H$ denote the total number of image pixels, E denote the number of edge pixels within the RoI, and T_{full} denote the number of discretized angle bins used in the standard Hough voting. Image preprocessing, including grayscale conversion and fixed-kernel Gaussian filtering, together with the improved Canny edge detection procedure, runs in $O(N)$ time. The dominant computational cost stems from Hough voting. For the proposed method, this cost is $O(E \cdot T_{\text{const}})$, where T_{const} denotes the number of angle bins within the constrained angular interval. Compared with the standard Hough transform that evaluates all T_{full} angle bins, the proposed RoI masking reduces E, as illustrated in Figure 5, and the polar angle constraint reduces the number of angle bins from T_{full} to T_{const}. This dual reduction decreases runtime in practice.

Figure 5. Illustration of the region of interest boundary.

3. Results

All experiments were conducted on a personal computer with an Intel Core i7-10700F processor and 24GB of RAM. The algorithm was implemented in Python using PyCharm as the development environment. Quantitative and qualitative comparisons were then conducted against the traditional lane detection method.

3.1 *Experimental Setup and Dataset Configuration*

To evaluate the proposed algorithm, all experiments were conducted under a consistent hardware and software setup to ensure reproducibility. The experiments were performed on 1923 selected TuSimple images at the original image resolution, and the official lane annotations were converted into binary ground-truth masks as described in Section 2.1. Accordingly, the hardware configuration, software environment, and parameter settings are summarized in Tables 1, 2, and 3, respectively.

Table 1. Experimental hardware configuration parameters.

Parameters	**Configuration**
CPU	Intel(R) Core(TM) i7-10700F CPU @ 2.90GHz
RAM	24.0 GB
GPU	NVIDIA Quadro P2200
Display Memory	5 GB

Table 2. Experimental software environment parameters.

Item	**Configuration**
OS	Windows 10 Professional 64-bit
PyCharm	2023.1.4
Python	3.9.16
OpenCV	4.7.0

Table 3. Experimental parameter settings.

Parameter	**Configuration**
Gaussian Filter Kernel Size	3×3
High-to-low Threshold Ratio	2.5
RoI Coordinates	$(0,720), (571,324), (792,321), (1280,720)$
Polar Angle Constraint Range	Left Lane: $[15^\circ \sim 75^\circ]$, Right Lane: $[-75^\circ \sim -15^\circ]$
Polar Radius Step Size	1 pixel
Image Weighting Coefficients	Original Image: 0.8, Lane-Line Image: 1.0

3.2 *Evaluation Metrics*

To objectively and comprehensively evaluate the performance of the proposed Canny–Hough lane detection algorithm, four key metrics are adopted: recognition rate, miss detection rate, false detection rate, and average detection time. These metrics collectively assess detection accuracy, localization reliability, robustness, and real-time performance, thereby enabling a quantitative evaluation of the effectiveness of the proposed improvements.

For metric calculation, the official TuSimple lane annotations were first transformed into pixel-level binary ground-truth masks, and the detected lane results produced by the proposed algorithm were compared with these masks on a pixel-by-pixel basis. To reduce ambiguity in metric calculation, the same rasterization rule was applied to the annotation-derived lane masks and the detected lane results. True-positive, false-negative, and false-positive lane pixels were counted under the same binary mask representation and image resolution. All accuracy-related metrics were computed on the same pixel-level evaluation basis for the proposed method and the reproduced classical baselines.

Recognition rate is a key metric for evaluating the detection performance of the algorithm [20]. It is defined as the ratio of the number of correctly detected lane-line pixels to the total number of ground-truth lane-line pixels in the image. The corresponding calculation is given as follows:

$$\text{Recognition Rate} = \frac{N_{\text{correct}}}{N_{\text{true}}} \times 100\% \tag{20}$$

where N_{correct} denotes the number of lane-line pixels correctly identified by the algorithm, and N_{true} denotes the total number of ground-truth lane-line pixels annotated in the dataset. A higher value of this metric indicates stronger capability of the algorithm in extracting and matching lane-line features, as well as a lower likelihood of false detections and missed detections.

The miss detection rate is defined as the ratio of ground-truth lane pixels that are not detected to the total number of ground-truth lane pixels. This metric reflects the continuity-preservation ability of the algorithm under complex scene conditions. A lower value indicates better retention of true lane structures.

$$\text{Miss Detection Rate} = \frac{N_{\text{miss}}}{N_{\text{true}}} \times 100\% \tag{21}$$

where N_{miss} denotes the number of ground-truth lane-line pixels that are not correctly identified by the algorithm. This metric is negatively correlated with the recognition rate and primarily reflects the algorithm's capability to preserve lane edge continuity in complex scenarios. A lower miss detection rate indicates stronger adaptability of the algorithm to varying scene conditions.

The false detection rate is defined as the ratio of non-lane pixels incorrectly classified as lane pixels to the total number of detected lane pixels. This metric reflects the ability of the algorithm to suppress interference from background structures and non-lane edges. A lower value indicates higher detection reliability.

$$\text{False Detection Rate} = \frac{N_{\text{false}}}{N_{\text{detect}}} \times 100\% \tag{22}$$

where N_{false} represents the number of non-lane pixels that are incorrectly classified as lane-lines, and N_{detect} denotes the total number of pixels detected as lane-lines by the algorithm. This metric directly reflects the algorithm's ability to suppress interference from non-lane regions. A lower false detection rate indicates more reliable detection results.

The average detection time is a key metric for evaluating the real-time performance of the algorithm. It is defined as the mean time required to process a single image from preprocessing to lane-line fitting and output. The statistic is computed over all experimental samples, and the unit is milliseconds. This metric directly determines whether the algorithm can satisfy the real-time requirements of applications such as autonomous driving. Because the present study focuses on a classical image-processing pipeline and reports pixel-level edge and line extraction behavior, a mask-based

pixel-level evaluation protocol was adopted to provide a direct and unified comparison of detection coverage, misses, and false responses.

3.3 *Ablation Study*

To isolate the individual contributions of the four core improvements, namely the 3×3 gradient operator, the interpolation-enhanced non-maximum suppression, the Otsu-based adaptive thresholding, and the angular constraint, a controlled ablation study was conducted. Each experimental group enables only one improvement while disabling the other three, with all remaining conditions kept identical. Since these components have been used individually in prior studies, the purpose of the ablation analysis here is not to claim component-level novelty, but to quantify their relative contributions and synergistic effects within the unified Canny–Hough framework proposed in this study. The baseline group G0 adopts the traditional Canny detector with a 2×2 gradient operator, manual thresholding, and simplified non-maximum suppression, followed by a Hough transform without any angular constraint. Group G1 activates only the 3×3 gradient operator. Group G2 activates only the optimized non-maximum suppression. Group G3 activates only the adaptive thresholding. Group G4 activates only the angular constraint within the Hough transform. Group G5 incorporates all four improvements and corresponds to the proposed Canny–Hough method.

Table 4. Definition of the Ablation Groups.

Label	3×3 Gradient Operator	Interpolation-Enhanced Non-Maximum Suppression	Otsu-Based Adaptive Thresholding	Angular Constraint
G0	×	×	×	×
G1	√	×	×	×
G2	×	√	×	×
G3	×	×	√	×
G4	×	×	×	√
G5	√	√	√	√

The independent contributions of the four core improvements were quantified via single-factor experiments. Otsu-based adaptive thresholding provides the largest gain in recognition rate, accounting for 42.9%, followed by the 3×3 gradient operator and interpolation-based optimization at 33.9% and 23.1%, respectively. The angular constraint is the main driver of runtime reduction, improving speed by 70.2%, while the other three modules yield moderate efficiency gains of 10% to 16.6%. More importantly, the four modules show clear complementarity under the unified experimental protocol: when integrated, they improve the recognition rate by 13.30 percentage points (16.95% relative to the baseline) and reduce the detection time by 39.11%. This result indicates that the main value of the proposed method lies in coordinated full-pipeline design rather than in any single component taken in isolation.

Table 5. Ablation study results of each module (G0–G5) in the proposed Canny–Hough pipeline.

Exp. Group	Recognition Rate (%)	Absolute Gain in Recognition Rate (percentage points)	Average Detection Time (ms)	Time Reduction Compared with Traditional Hough (ms)	Time Reduction Rate (%)
G0	78.42	–	53.31	–	–
G1	84.17	5.75	49.85	3.46	6.49
G2	82.33	3.91	51.22	2.09	3.92
G3	85.69	7.27	50.13	3.18	5.97
G4	79.05	0.63	38.67	14.64	27.46
G5	**91.72**	13.30	**32.45**	20.86	39.11

3.4 *Sensitivity Analysis of the Polar-Angle Constraint*

To examine whether the polar-angle constraint was overly tuned to the current TuSimple-based setting, a minimum sensitivity analysis was conducted by perturbing the default angular intervals in Table 3. The baseline setting used $[15°, 75°]$ for the left lane and $[-75°, -15°]$ for the right lane. Two additional settings were evaluated: a narrowed interval ($[25°, 65°]$ and $[-65°, -25°]$) and a widened interval ($[5°, 85°]$ and $[-85°, -5°]$). All other algorithmic components and evaluation conditions were kept unchanged.

The results are summarized in Table 6. The baseline configuration achieved the best overall balance between recognition performance and runtime efficiency, with a recognition rate of 91.72% and an average detection time of 32.45 ms. When the angular interval was narrowed, the recognition rate decreased to 90.16%, while the average detection time was reduced to 31.78 ms. When the interval was widened, the recognition rate remained relatively stable at 91.34%, whereas the average detection time increased to 33.27 ms. These results suggest that the selected angular priors represent a practical compromise between detection coverage and computational efficiency under the tested forward-view highway-like scenario.

Table 6. Minimum sensitivity analysis of the polar-angle constraint under ±10° interval perturbation.

Angular-Constraint Setting	Left-Lane Range	Right-Lane Range	Recognition Rate (%)	Average Detection Time (ms)
Narrowed interval (−10°)	$[25°, 65°]$	$[-65°, -25°]$	90.16	31.78
Baseline	$[15°, 75°]$	$[-75°, -15°]$	91.72	32.45
Widened interval (+10°)	$[5°, 85°]$	$[-85°, -5°]$	91.34	33.27

3.5 *Comparative Evaluation*

To provide a broader performance reference, the proposed method was compared with representative classical and deep-learning-based lane detection methods, including Canny + PHT, LaneNet, and SCNN. For classical baselines, experiments were conducted under the same local evaluation settings. For deep-learning baselines, the reported results are used as reference-level comparisons to illustrate the accuracy–efficiency trade-off rather than strict hardware-equivalent runtime competition. For brevity and cross-method comparability, Table 7 reports only recognition rate and runtime-related metrics, whereas miss detection rate and false detection rate are further discussed for the proposed method under challenging scenarios in Table 8.

Table 7. Performance comparison with additional baseline methods.

Algorithm	Recognition Rate (%)	Average Detection Time (ms)	FPS	Hardware Dependency	Data Dependency
Traditional Hough	78.42	53.31	18.76	CPU	–
Canny + PHT	83.56	41.27	24.23	CPU	–
LaneNet	93.15	58.62	17.06	GPU	Extensive annotation requirements
SCNN	**94.28**	65.37	15.30	GPU	Extensive annotation requirements
Proposed Algorithm	91.72	**32.45**	30.82	CPU	–

Note: FPS is computed as FPS = 1000/(average detection time in ms) under the same experimental settings.

As shown in Figure 6, the proposed algorithm achieves the shortest average detection time (32.45 ms) and the highest FPS (30.82) while maintaining a competitive recognition rate (91.72%), demonstrating a favorable accuracy–speed trade-off compared with both traditional pipelines and deep-learning baselines.

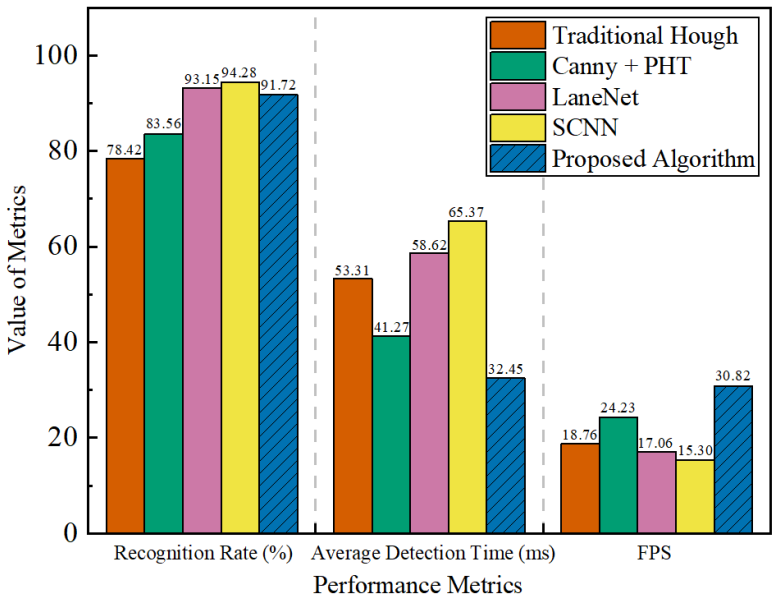


Figure 6. Comparison of Performance Metrics Among Different Lane Detection Algorithms

Compared with Traditional Hough, the proposed algorithm improves the pixel-level lane recall by 13.30 percentage points while reducing the average detection time by 20.86 ms, indicating that the coordinated improvements in thresholding, edge extraction, and constrained voting effectively enhance the classical pipeline. Relative to Canny + PHT, the proposed method achieves an additional 8.16 percentage-point gain in recall and reduces the detection time by 8.82 ms, further demonstrating the benefit of coordinated interaction between the improved Canny stage and the constrained Hough stage.

For the deep-learning baselines, the results in Table 7 are reported only as reference-level comparisons for positioning the proposed method relative to representative learning-based approaches, because the hardware platforms, software environments, and evaluation settings are not strictly identical. Therefore, these comparisons should not be interpreted as hardware-equivalent runtime competition. Under the current local CPU-based setting, the proposed method shows clear

efficiency advantages over the reproduced classical baselines while maintaining competitive detection accuracy relative to representative deep-learning methods.

Figure 7. Qualitative visualization of the proposed lane detection results.

4. Discussion

To address the challenge of simultaneously improving detection accuracy and real-time performance in classical lane detection, this study developed a lightweight Canny–Hough framework with coordinated improvements in edge extraction, adaptive thresholding, and constrained Hough voting. The results show that the proposed method improves both recognition performance and computational efficiency compared with the traditional Hough-based method, indicating that system-level coordination across the full detection pipeline is more effective than isolated refinement of individual modules. From an engineering perspective, the main value of the proposed framework lies in lightweight deployment-oriented redesign and practical full-pipeline re-optimization for resource-constrained scenarios. Accordingly, the contribution of this study lies primarily in deployment-oriented coordination of classical modules rather than in proposing a completely new primitive operator.

The performance gain of the proposed method results from the joint effect of multiple improvements. The enhanced gradient operator and interpolation-based non-maximum suppression improve the continuity and localization of lane-edge features, while Otsu-based thresholding increases adaptability under varying image conditions. In addition, RoI masking and polar-angle restriction reduce invalid voting and suppress interference from non-lane regions, thereby improving both robustness and runtime efficiency. These findings, together with the ablation results, confirm the value of coordinated integration within a unified classical lane-detection pipeline.

To further examine robustness in challenging environments, representative backlight, shadow, and low-light scenes were analyzed, and the quantitative results are summarized in Table 8. The proposed method still outperforms the traditional Hough-based approach in all three scenarios, but its performance decreases under severe illumination variation. This

decline is mainly associated with weakened lane–background contrast, shadow-induced pseudo-edges, and increased noise interference in low-light conditions. Specifically, the challenging-scene subset consisted of 72 backlight images, 69 shadow images, and 58 low-light images, which were categorized by visual inspection according to dominant illumination conditions. These categories were used only for descriptive robustness analysis and were not involved in model development, parameter tuning, or baseline selection. These challenging-scene samples were drawn from the 1923-image evaluation subset and were used only for robustness analysis rather than for parameter tuning.

Table 8. Quantitative performance analysis of the proposed method under representative challenging scenarios.

Scenario Type	Recognition Rate (%)	Miss Detection Rate (%)	False Detection Rate (%)	Average Detection Time (ms)	Recognition Rate of Traditional Hough (%)
Backlight Scenario	72.14	21.38	6.48	33.52	55.72
Shadow Scenario	76.82	17.56	5.62	33.01	60.18
Low-Light Scenario	77.68	18.24	6.08	32.89	58.36

Overall, the proposed method significantly improves the classical lane-detection pipeline, but its robustness under complex illumination conditions still requires further enhancement. Future work will focus on illumination-adaptive preprocessing, shadow suppression, and stronger contextual constraints. In addition, the polar-angle intervals adopted in this study were derived from the lane-angle distribution of the selected TuSimple images and shoulds therefore be regarded as scene-dependent priors rather than universally optimal parameters. Their applicability to road environments with sharp curvature, large elevation variation, urban intersections, or different camera installation settings remains to be further validated, and future work should examine whether adaptive or data-driven angle priors can improve generalizability. In addition, the minimum sensitivity analysis in Section 3.4 shows that moderate $\pm 10^{\circ}$ perturbations around the selected intervals do not change the overall conclusion of the study, although the balance between recognition performance and runtime is affected to some extent.

5. Conclusions

This study proposed a lightweight Canny–Hough lane detection framework for improving both detection accuracy and runtime efficiency in classical lane detection. Instead of optimizing only isolated stages, the proposed method jointly improves edge extraction, threshold selection, and constrained Hough voting within a unified pipeline. The results show that coordinated redesign of a classical pipeline can still provide meaningful practical value for CPU-based and resource-constrained deployment scenarios.

Specifically, a 3×3 multi-direction gradient operator, interpolation-based non-maximum suppression, and Otsu-based adaptive thresholding were integrated to improve the quality and stability of lane-edge extraction. In addition, region-of-interest masking and polar-angle restriction were incorporated into the Hough transform to reduce non-lane interference and computational redundancy. Experimental results on the TuSimple dataset show that the proposed method achieves a recognition rate of 91.72% and an average detection time of 32.45 ms. Compared with the traditional Hough-based method, the recognition rate is improved by 13.30 percentage points (16.95% relative to the baseline), while the detection time is reduced by 39.11%, demonstrating that coordinated full-pipeline optimization provides a favorable balance between accuracy and efficiency.

Overall, the proposed method provides an effective and lightweight solution for lane detection under the current TuSimple-based forward-view setting. However, the reported results should be interpreted within the current dataset, camera viewpoint, and angle-prior configuration. Further validation is still needed for more diverse road geometries,

illumination conditions, and camera setups. Future work will focus on improving illumination robustness and enhancing the adaptability of scene-dependent geometric priors.

Author Contributions

Methodology and writing—original draft preparation, X. N. ; formal analysis and investigation, A. L.; software, J. CH.; data curation, L. W.; validation, Q. ZH. . All authors have read and agreed to the published version of the manuscript.

Funding

This project is supported by Shandong Province Science and Technology oriented Small and Medium Enterprises Enhancement Project (Grant No. 2023 TSGC0288). Jinan 2023 talent development special fund research leader studio project (Grant No. 202333067). Foreign expert project (Grant No. G2023023002L). Shandong Provincial Higher Educational Youth Innovation Science and Technology Program (Grant No. 2019KJB019), Major science and technology innovation project in Shandong Province (Grant No. 2022CXGC020706), Ministry of industry and information technology manufacturing high quality development project, China (Grant No. 2023ZY02002).

Conflict of Interest

The authors declare no conflict of interests.

References

[1] Guo Z, Li S, Li Y et al. Automatic Lane line recognition algorithm based on improved Hough transform. *Journal of Changchun University of Technology*. 2024; 45(2): 130–137.

[2] Narote S P, Bhujbal P N, Narote A S, Dhane D M. A review of recent advances in lane detection and departure warning system. *Pattern Recognition*. 2018; 73: 216–234.

[3] Pan X, Shi J, Luo P, Wang X, Tang X. Spatial As Deep: Spatial CNN for Traffic Scene Understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 2018; 32(1): 7276–7283.

[4] Zheng T, Huang Y, Liu Y, Tang W, Yang Z, Cai D, He X. CLRNet: Cross Layer Refinement Network for Lane Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022: 898–907.

[5] Han J, Deng X, Cai X, Yang Z, Xu H, Xu C, Liang X. Laneformer: Object-Aware Row-Column Transformers for Lane Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 2022; 36(1): 799–807.

[6] Fayez Tarsha-Kurdi, Tania Landes, Pierre Grussenmeyer. Hough-Transform and Extended RANSAC Algorithms for Automatic Detection of 3D Building Roof Planes from Lidar Data. *ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007*, Sep 2007, Espoo, Finland. pp. 407–412.

[7] Gharineiat Z, Tarsha Kurdi F, Campbell G. Review of Automatic Processing of Topography and Surface Feature Identification LiDAR Data Using Machine Learning Techniques. *Remote Sensing*. 2022; 14(19): 4685.

[8] Ahmad Smeea, Fayez Tarsha Kurdi, Jwan Zoaa, Abbas Al-Haj-Ali Abbas. Remote Sensing Topographical Measurements and Damage Assessment in Disaster Areas. *Int J Environ Sci Nat Res*. 2025; 35(4): 556422.

[9] TuSimple. TuSimple Lane Detection Challenge. 2017.

[10] Cheng W, Wang X, Mao B. Research on lane line detection algorithm based on instance segmentation. *Sensors*. 2023; 23(2): 789.

[11] Gonzalez R C, Woods R E. *Digital Image Processing*. 3rd ed. Upper Saddle River, NJ: Prentice Hall. 2008.

[12] Canny J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1986; 8(6): 679–698.

[13] Li Z, Wang G. Research on lane line recognition based on traditional edge operators. *Modern Electronics Technique*. 2024; 47(7): 61–65.
[14] Otsu N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*. 1979; 9(1): 62–66.
[15] Yang Y. Lane detection method based on improved Hough transform. *International Journal of Simulation and Process Modelling*. 2023; 21(1): 14–21.
[16] Li J, Song L, Fan G et al. An improved probabilistic Hough line detection algorithm. *Journal of Naval University of Engineering*. 2024; 36(1): 62–68.
[17] Li D, Yang Z, Nai W, Xing Y, Chen Z. A road lane detection approach based on reformer model. *Egyptian Informatics Journal*. 2025; 29: 100625.
[18] Dong Y, Cao J, Sun F et al. Research on lane line recognition based on OpenCV image processing. *Instruments & Analytical Monitoring*. 2023; (3): 29–32.
[19] Liu Y, Li S, Lu T et al. AFLaneNet: An attention-fused instance segmentation network for real-time lane detection. *Signal, Image and Video Processing*. 2025; 19(3): 247.
[20] Feng Q, Chi C, Chen F et al. Grid anchor lane detection based on attribute correlation. *Applied Sciences*. 2025; 15(2): 699.