Journal of Informatics and Web Engineering

Vol. 4 No. 3 (October 2025) eISSN: 2821-370X

Robust Lane Detection under Varying Lighting Conditions Using Adaptive Vision-Based Techniques

Habibatelrahman Mosaad Abdelfattah Mohamed Abdelaal[1], Tee Connie[2*], Michael Kah Ong Goh[3]

[1,2,3]Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450, Melaka, Malaysia

**corresponding author: (tee.connie@mmu.edu.my; ORCiD: 0000-0002-0901-3831)*

Abstract - Reliable lane detection is crucial to autonomous driving but continues to be challenging with varying lighting conditions. Fluctuations in illuminations due to bright sunlight, shadows or low lighting at night can degrade the visual quality and adversely affect the accuracy of the lane detection results. This research proposes an adaptive approach for lane detection under different lighting scenarios. For daytime, a Region of Interest (ROI) masking and line averaging technique help in the stability and visibility of the lane markings. For nighttime conditions, a Probabilistic Hough Transform-based method improves lane detection in low-light environments. An evaluation tool has been developed to check if certain parameters correlate with day or night to enable dynamic selection of the most suitable detection technique. The proposed new method improves image preprocessing and combines several computer vision algorithms for accurate lane tracing. This new solution aids in shadow regions and faded marking areas, as well as improves precision for multi-lane roadways with varying lane widths. The approach adds accuracy for real-time lane recognition of autonomous vehicles on multi-lane highways with different degrees of illumination. This research also contributes toward the goal of improving safety and efficiency in autonomous driving by providing more effective methods of ensuring safe driving.

Keywords—Computer Vision, Lane Detection, Digital Image Processing, Kalman Filter, Autonomous Driving

Received: 28 March 2025; Accepted: 13 July 2025; Published: 16 October 2025

1. INTRODUCTION

Lane detection refers to a major task of self-driving cars where the vehicle detects and follows road lane markings to drive safely [1], [2], [3]. The self-driving vehicle's ability to accurately follow lanes is impaired by the external environment in the form of frequent light changes that can sometimes make proper lane detection impossible [4], [5], [6]. This study develops a robust lane detection system that adapts both daytime and nighttime lane detection strategies to dynamically achieve reliable performance in changing illumination conditions.

Recent advancements in lane detection have shown promise [7-10]. However, there are several limitations. For instance, [11] utilized hue, saturation, value (HSV) and grayscale spaces along with edge detection techniques. The method could detect the lane edges under normal circumstances, but it could not handle lanes that are severely

Journal of Informatics and Web Engineering

https://doi.org/10.33093/jiwe.2025.4.3.20

Published by MMU Press. URL: journals.mmupress.com/jiwe

occluded or faded lane scenarios. [12] introduced a CNN-RNN hybrid architecture that effectively handled temporal features in video sequences. Nevertheless, it incurred high computational costs making real-time deployment challenging. [13] proposed LaneNet, a deep-learning-based framework with efficient lane edge and line detection stages, but it was susceptible to false positives from road markings such as arrows or characters. These methods highlight the trade-offs between accuracy, robustness, and efficiency. Motivated by these challenges, our work focuses on developing a lane detection algorithm that balances real-time capability and robustness to challenging conditions such as poor lighting and occlusion while remaining computationally efficient.

Daytime lane detection is achieved through computer vision techniques, such as line averaging and ROI masking [14]. ROI masking provides the detection focus on the road area and filters out the noise from other areas as a background because they are considered irrelevant. The proposed method then implements edge detection and Hough Transform for the identification of lane lines. This is followed by line averaging for smoothening out detected lane edges and increasing stability. This approach provides robust lane detection under favourable lighting conditions.

On the other hand, for nighttime, the study employs Probabilistic Hough Transform [15] with sliding window for lane detection at night when visibility is poor and traditional techniques are less successful. By adding a probabilistic model that gives detected lane lines confidence values, this method improves lane detection. This makes the method more robust against noise and partial occlusions. Besides, to improve lane line contrast prior to detection, visibility enhancement techniques like HSV filtering and gamma correction are used.

To achieve stable lane following behaviour under various lighting conditions, the proposed method dynamically selects the appropriate method based on real-time brightness analysis. Under well-lit conditions, it prioritizes ROI masking and line averaging for precise lane identification. Conversely, when there is low light, the method choses the Probabilistic Hough Transform with improved preprocessing techniques. This one-size-fits-all method raises the precision of lane recognition to ensure uninterrupted operation even for weather changes.

The proposed lane detection method is designed to operate effectively in both daytime and nighttime conditions. The proposed method utilizes adaptive strategies such as dynamic ROI masking and an adaptive day/night switch to enhance lane detection under varying lighting and challenging road conditions. By combining computer vision techniques with real-time adaptability, the method improves detection accuracy and stability [16]. The following sections detail the proposed architecture, experimental setup, and results, demonstrating the effectiveness of the proposed approach in real-world scenarios.

1.1 Novelty and Contributions

While our method is mainly built on top of computer vision techniques like ROI masking and Kalman filter, its novelty lies in the adaptive orchestration of the components in a real-time lane detection framework.

- The dynamic ROI masking technique enables flexible adaption to different road geometries.
- Kalman filter is repurposed for both smoothing and validating predictions.
- A day/night adaptive switch improves the detection accuracy under varying lighting condition which is a gap inadequately addressed in the existing literature.

2. LITERATURE REVIEW

As presented by [11], their approach involved both the use of the HSV space to process the yellow lane lines and the utilization of the grayscale space in processing the white lane lines. This step improved the sensitivity in curvature-LCS identifier, as well as in LCS as a concept itself. The method involved the application of Canny edge detection, inverse perspective transformation, and a sliding window polynomial fitting algorithm to achieve real-time lane detection functionality. Such algorithms are typical in CV and its subfield, and they were successfully integrated in this research to determine the lane's location. The authors tested the traffic conditions in real time, after which it was shown that their algorithm could detect the curves and lane lines in varying light conditions. Furthermore, the algorithm processed the lane line detection findings and calculated the deviation distance.

[12] also came up with a scheme that considered the lane from a video sequence with consecutive images instead of just a single image. This method was created to accommodate heavy shades, bad marks, and vehicle obstructing during

traffic. This work came up with a hybrid architecture for deep learning consisting of a convolutional neural network (CNN) and recurrent neural network (RNN). In the first part, the information of each frame was transformed into CNN features by the CNN block, and the features learned from the continuous images along with the temporal space were related and sent into the RNN block for RNN feature training and lane identification. Extensive experimental evaluations using two datasets - one of the large scales and the other rather difficult - showed that the proposed method significantly outperformed the competing methods in terms of lane detection and were the best for situations that are difficult to handle. Lastly, the study established a new way to identify and locate vehicle lanes from driving onto the continuous scenes using neural networks. The proposed method illustrated the applicability of deep learning techniques in identifying lane areas in complex environment.

The LaneNet framework proposed by [13] employed a deep neural network strategy that divided lane detection into two key stages: lane edge proposal and lane line detection. Initially, lane edge proposition network was used, and it was used to provide detailed lane edge categorization at the pixel level. Following the first stage, the lane line identification network during the second stage exploited these propositions to be generated by the lane edge network to identify lane lines. The main task of LaneNet was concentrated on finding just lines of the lanes. Because of that, it was easy to generate false positives by similar markings, such as arrows or characters on the road. However, these difficulties are accounted for by the authors who showcased the effectiveness of their lane detection approach for both highway and urban driving via the use of precise image processing techniques that did not rely on defined number of lanes or lane configurations. The advantages of the LaneNet were operating speed and efficient computational requirements, making it feasible to be deployed inside a vehicle- based system. Experimental results showed LaneNet as a reliable and efficient solution for recognizing traffic lanes across different real situations.

A method coined as Vision-based Intelligent Lane Departure Warning System (VILDS) for Autonomous Vehicles was introduced by [17] to tackle issues like varying brightness, blur and occlusion in different locations. The Generative Adversarial Networks (GAN) of the VILDS exploited the most subtle features to create images that are perfect copies of the original but with better sharpness. The method incorporated Long Short-Term Memory (LSTM) to understand normal activities that occurred in the samples to forecast lanes live and considered processed images effectively, predicting the incomplete lanes and minimizing the traffic error. Also, the authors proposed an approach to boost the sensitivity of the prosed method by detecting the direction and angle of the deviation to find out when the AV was about to go over a lane. The overall assessment of the designed VILDS system showed that the lane detection sub-system and the lane departure warning sub-system could operate successfully with an accuracy of 98.2% and 96.5% respectively.

[18] offered a reliable method of camera (image)-based lane perception. The authors suggested a distinctive system that both traction filtering and detection mechanism possessed. The tracking filter was used to select a ROI for the tracing of lane segments at different distances. For each successful motion-detection, the information concerning lane position would be immediately stored and later used to update the lane geometry. The strategy aimed to optimize lane recognition and curve detection by introducing road labels and detection algorithm. The paper focused on lanes prediction accuracy as well as its accuracy under different conditions. The authors effectively implemented lane detection through Unet and Segnet models. The Tusimple data was used to confirm the accurate results. The Unet implementation showed that the Unet model gave better results compared to the Segnet model. A comparison among the methods is presented in Table 1.

3. PROPOSED SOLUTION

3.1 Overview

This study proposes an adaptive lane detection system that combines methods of automatic mode for both daytime and nighttime conditions. For the extended daytime period, we employ ROI masking and line averaging to boost lane visibility and, as a result, accuracy of detection. During the night, the system uses the Probabilistic Hough Transform combined with the gamma correction and HSV filtering to improve the contrast of lane lines in case of low illumination conditions. The switching between the two methods, made possible using a dynamic selection system, is based on real-time analysis of brightness, which ensures smooth insulation. Additionally, the Kalman filtering is also implemented in lane tracking for lane maintaining, abrupt occlusion compensation. In Figure 1, the block diagram shows the architecture of the method suggested.

Table 1. A comparison of existing methods

Study	Method	Real-Time Capability	Handling of Challenging Conditions	Accuracy
[11]	HSV (yellow), Grayscale (white), Canny, IPM, Sliding Window	Yes	Handles curves and varied lighting	93%
[12]	CNN + RNN for sequential video-based detection	Moderate	Robust under occlusion, blur, and heavy shade	96.4%
[13]	Lane Edge Proposal + Lane Line Detection via CNN	Yes	Highway, urban roads; issues with false positives	95.1%
[17]	GAN (image enhancement) + LSTM (prediction)	Yes	Blur, occlusion, brightness variations	98.2%
[18]	ROI Tracking + Motion Filtering + U-Net / SegNet	Moderate	Lane prediction under varying distances	U-Net: ~94%, SegNet: ~91%

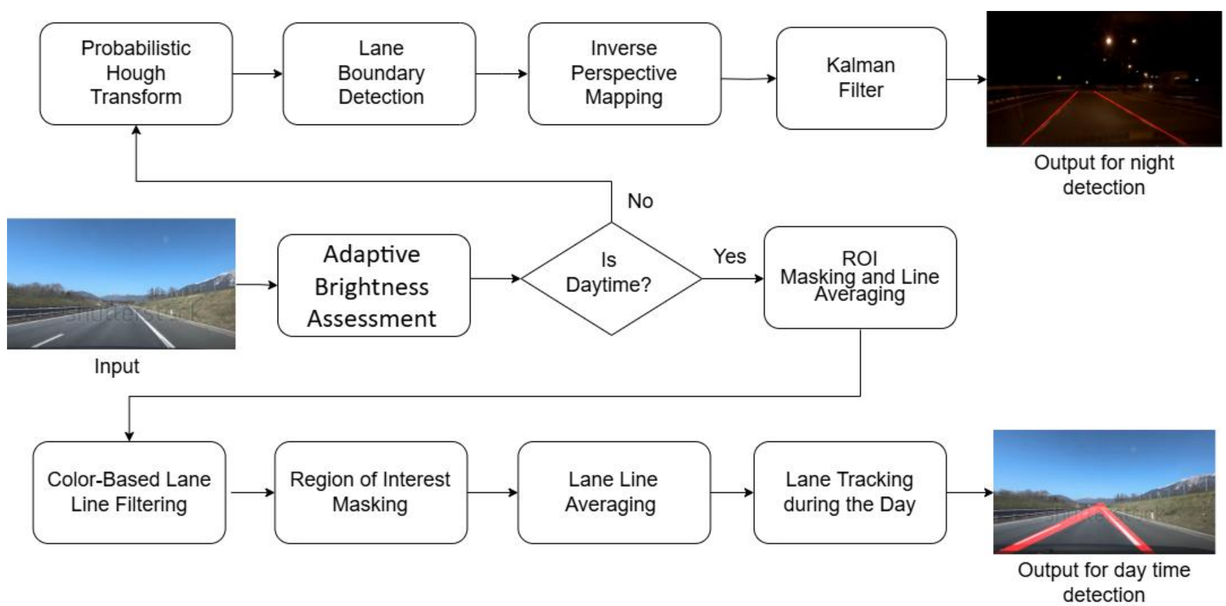

Figure 1. Block Diagram of the Proposed System

3.2 Lane Detection and Tracking during the Day

3.2.1 Day Time Lane Detection with ROI Masking and Line Averaging

Image preprocessing is the very first decisive step in lane detection, which guarantees high accuracy of the data collected. Starting from the image input, it is converted into grayscale and simplified, which lowers the computational burden. Gaussian filter is then utilized to attenuate noise and is further used to smooth the image. Following that, the ROI mask is applied to isolate the area where possibly the lane lines reside. Lane lines are identified after preprocessing by a Canny edge detector, with the subsequent application of the Hough Transform. To enhance the precision, this study adds Lane Line Averaging by which processed lanes' line outputs are averaged to provide uniform and adequately defined lane edges.

A) Colour-Based Lane Line Filtering

The colour selection process [19] includes filtering the image to obtain only the white and yellow colours, which are the typical colours for lane lines. This process is implemented using the HLS colour space [20]. The conversion from RGB to HLS is performed using:

$$H,L,S = \begin{cases} H = 60^{\circ} \times \frac{(G'-B')}{C} + 0^{\circ}, & \text{if } \max = R' \\ H = 60^{\circ} \times \frac{(B'-R')}{C} + 120^{\circ}, & \text{if } \max = G' \\ H = 60^{\circ} \times \frac{(R'-G')}{C} + 240^{\circ}, & \text{if } \max = B' \\ H = 0, & \text{if } C = 0 \end{cases} \tag{1}$$

$$L = \frac{\max(R',G',B') + (R',G',B')}{2} \tag{2}$$

$$S = \begin{cases} 0, & \text{if } C = 0 \\ \frac{C}{1-|2L-1|}, & \text{otherwise} \end{cases} \tag{3}$$

where:

$R' = \frac{R}{255}, G' = \frac{G}{255}, B' = \frac{B}{255}$, and $C = \max(R', G', B') - \min(R', G', B')$.

After colour selection, the image is converted to grayscale to simplify the image. Gaussian blur is further applied to reduce noise and smooth the image. Figure 2, Figure 3 and Figure 4 illustrate the results after applying colour selection, grayscale conversion and Gaussian blur, respectively.

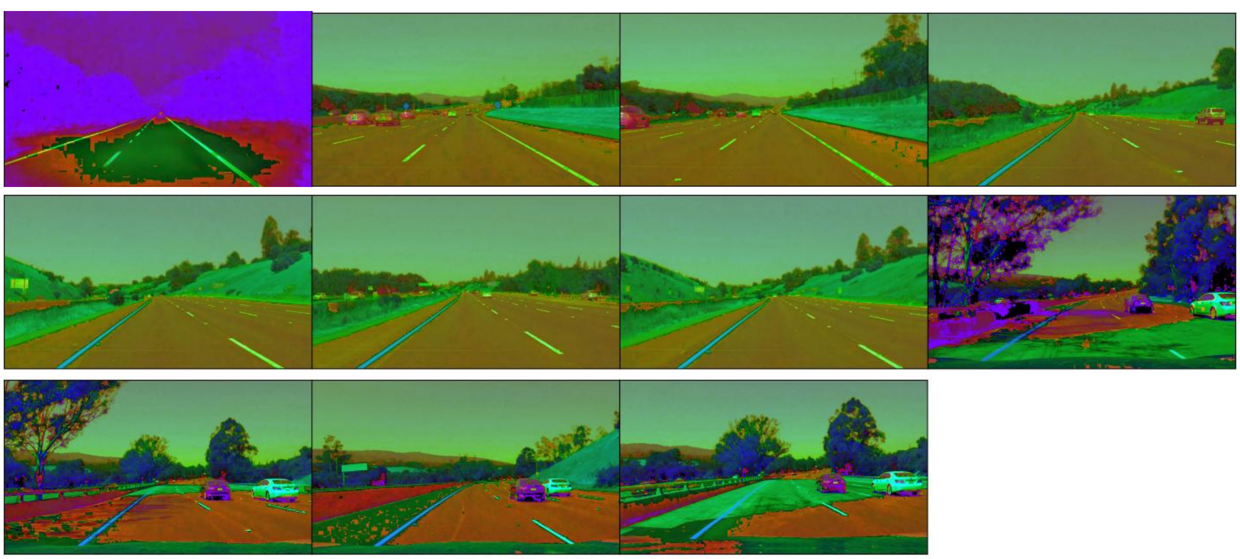

Figure 2. Colour Selection

B) ROI Masking

Masking by ROI is a precautionary method in lane detection to filter out the areas of an image where the lane lines are not likely to be found. Such a process, therefore, decreases noise arising from ignored background elements, such as other vehicles, trees, or buildings. The attention to this factor has encouraged the enhancement of lane detection accuracy.

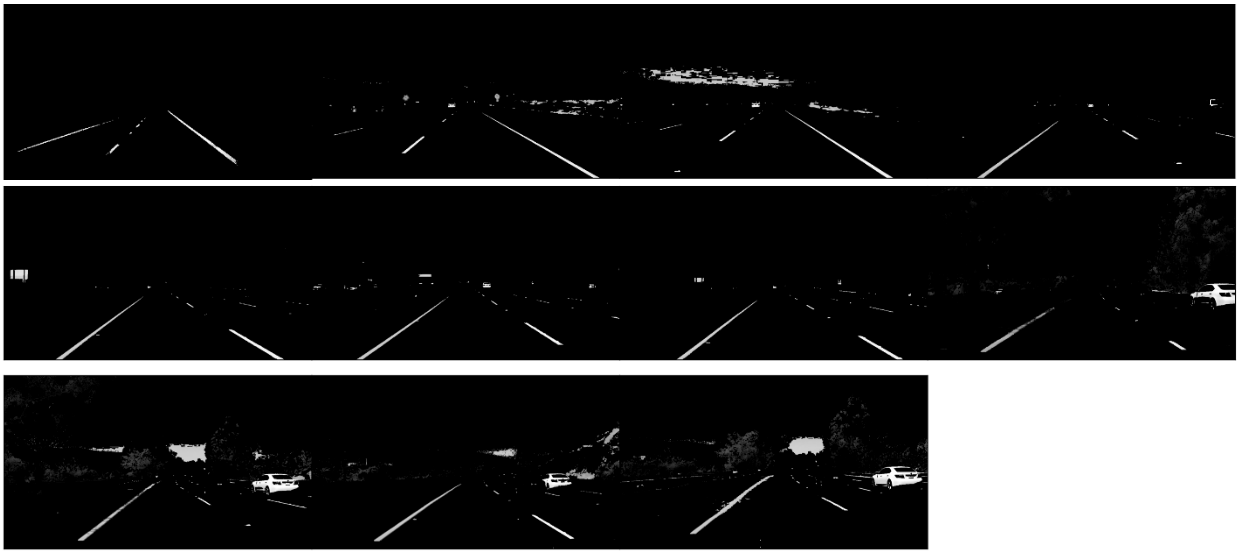

Figure 3. Grayscale Conversion

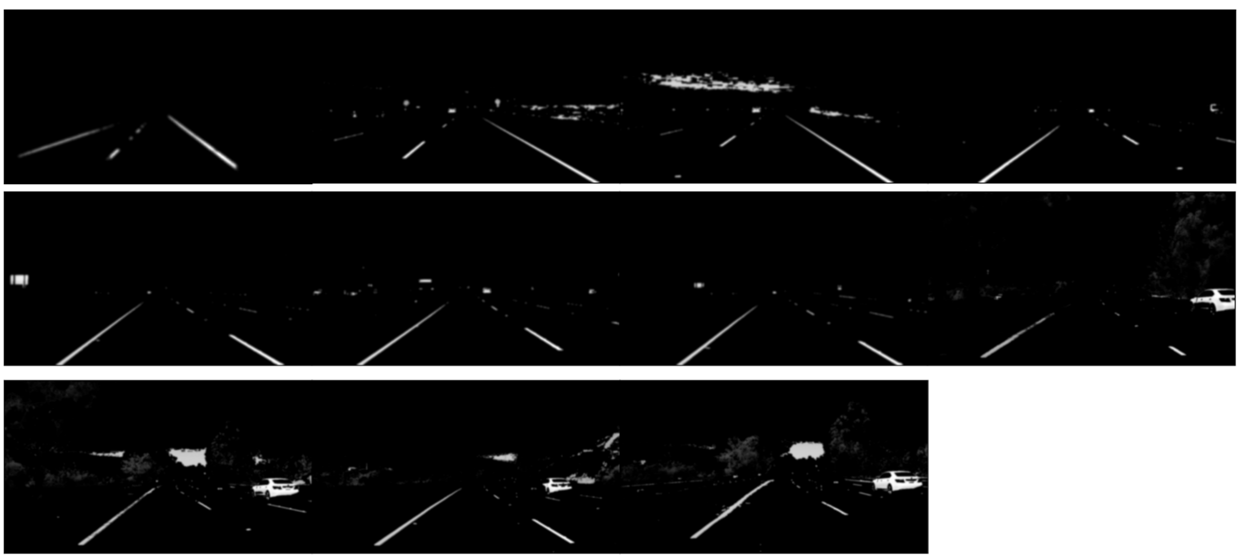

Figure 4. Gaussian Blur Applied

The ROI is specified as a polygon area, which is usually covering the facility surface while filtering the sky and other unneeded zones. The choice of this precise polygon is based on the angle at which the camera captures the lane markings, making sure that only the lane lines are included in the polygon and no other objects. First, a blank matrix, $M(x,y)$, of the same dimensions as the input image is created, initialized with zeros (black pixels). The polygon defining the ROI is then filled with white pixels (value 255), indicating the area of interest,

$$M(x,y) = \begin{cases} 255, & \text{if } (x,y) \in \text{polygon defined by vertices} \\ 0\,, & \text{otherwise} \end{cases} \tag{4}$$

The original image, $I(x,y)$, is then element-wise multiplied with the mask to ensure that only the defined ROI is retained, while the rest of the image is set to black,

$$I_{masked}(x, y) = I(x, y) \cdot M(x, y) \tag{5}$$

Figure 5 illustrates the results of images after applying the ROI mask. The results show that unnecessary regions are removed while leaving only the relevant road area for further processing.

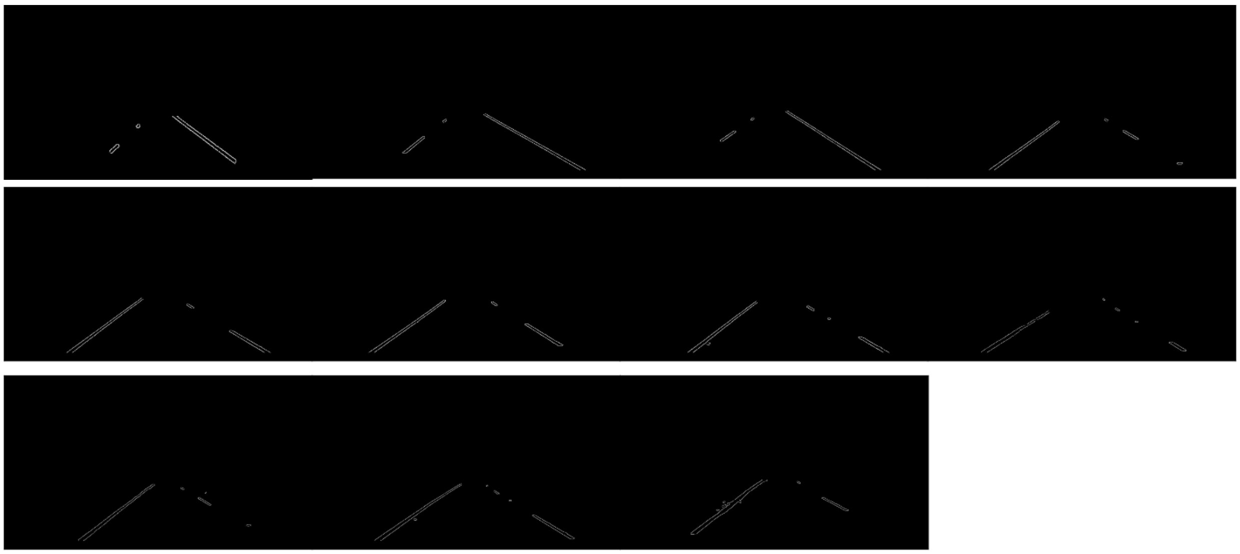

Figure 5. ROI Results

C) Lane Line Averaging

Lane line detection begins with edge detection using the Canny edge detector. It highlights edges in an image by analysing gradient magnitudes and directions. The binary edge map is given by:

$$I_{edges} = \begin{cases} 1, & if\ \nabla I_{masked} > high\ threshold \\ 0, & if\ \nabla I_{masked} < low\ threshold \\ weak\ edge, & if\ low\ threshold \leq \nabla I_{masked} \leq high\ threshold \end{cases} \tag{6}$$

where I_{edges} represents the binary edge map, and the gradient magnitude ∇I_{masked} is computed as:

$$\nabla I_{masked} = \sqrt{\left(\frac{\partial I_{masked}}{\partial x}\right)^2 + \left(\frac{\partial I_{masked}}{\partial y}\right)^2} \tag{7}$$

Once the edges are detected, Hough Transform is applied to detect lane lines. This method transforms edge points into a parameter space to identify straight-line segments. The equation of a line in Hough space is given by,

$$\rho = x\cos\theta + y\sin\theta \tag{8}$$

where ρ represents the perpendicular distance from the origin to the line, and θ is the angle between the *x*-axis and the perpendicular to the line. This transformation can detect straight lane lines even in the presence of noise or partial occlusions.

After detecting multiple line segments using the Hough Transform, Lane Line Averaging is applied to refine the final lane lines. The detected lane segments are averaged to obtain stable lane boundaries. The final lane lines are determined by calculating the average slope and intercept of the detected segments,

$$slope = \frac{y_2 - y_1}{x_2 - x_1} \quad (9)$$

$$intercept = y_1 - slope \cdot x_1 \quad (10)$$

This process results in final lane lines which are smooth and continuous (refer Figure 6). The proposed method improves the accuracy and robustness of the lane detection system, particularly on curved roads or in varying lighting conditions.

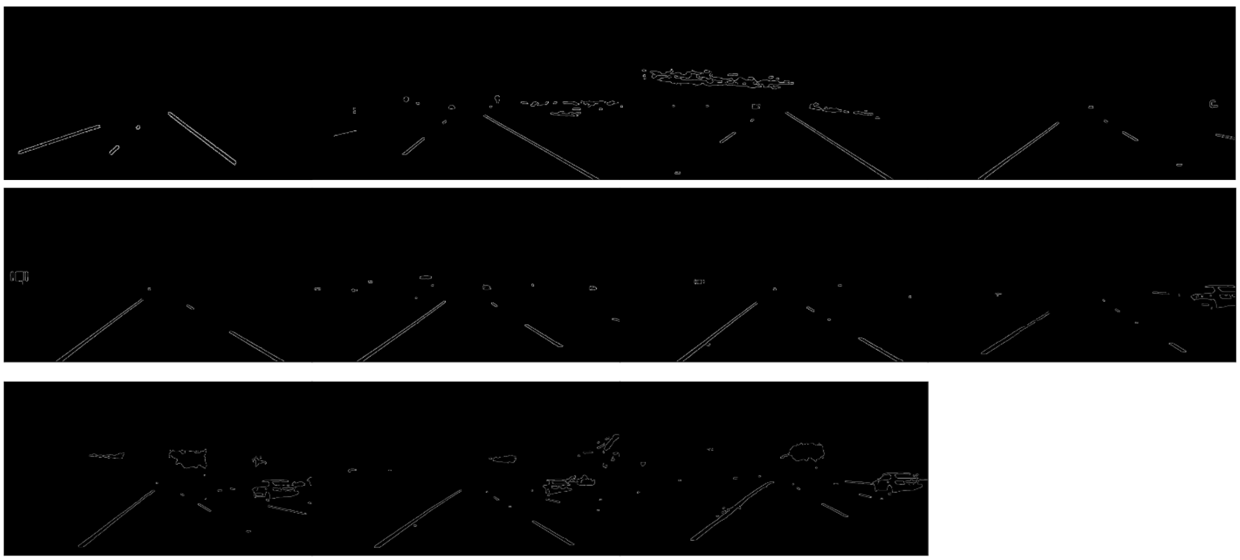

Figure 6. Edge detection

3.2.2 Lane Tracking during the Day

Daytime lane tracking relies on the detected lane lines. Lane lines are tracked continuously from frame to frame to ensure consistency and accuracy. By monitoring the detected lane lines within nearby frames, the method tracks a constant lane position so that the vehicle can stay on its designated path with the least amount of swaying and risk of collision. Some sample lane lines detected during the day are illustrated in Figure 7.

3.3 Lane Detection and Tracking at Night

3.3.1 Night Lane Detection with Probabilistic Hough Transform

Night lane detection is one of the important components in the proposed system. It incorporates a series of steps to enhance visibility and extract lane features effectively. These steps include gamma correction, HSV filtering, Probabilistic Hough Line Transform, lane boundary identification and line scaling.

Unlike the standard Hough Transform, Probabilistic Hough Transform (PHT) assigns a probability to each detected line. The transformation process accumulates votes for possible line parameters in a discretized Hough space,

$$H(\rho, \theta) = \sum I_{edges}(x, y) \quad (11)$$

where $H(\rho, \theta)$ stores the count of edge points supporting each line parameter. A predefined threshold determines the minimum number of votes required for a line to be considered valid.

Figure 7. Detected Lane Lines during the Day

A) Lane Boundary Detection

Lane boundaries are identified by selecting the two closest lines to the frame centre. The system calculates the orientation of each detected line relative to the horizon and filters out unwanted lines. The angle, θ, of each line is computed using the inverse tangent function,

$$\theta = \left|arctan\left(\frac{y_2-y_1}{x_2-x_1}\right)\right| \quad (12)$$

where (x_1, y_1) and (x_2, y_2) are the endpoints of a detected line. The absolute value ensures positive angle values. The horizontal distance *d* of each detected line from centre, x_c, is calculated as,

$$d = |x_m - x_c| \quad (13)$$

where x_m represents the line's midpoint:

$$x_m = \frac{x_1-x_2}{2} \quad (14)$$

The detected lane lines are scaled and adjusted to align with the road horizon using the following slope-intercept equations:

$$m = \frac{\mathrm{y}_2-\mathrm{y}_1}{\mathrm{x}_2-\mathrm{x}_1} \quad (15)$$

$$c = y_1 - mx_1 \quad (16)$$

$$base_{cross} = -\frac{c}{m} \quad (17)$$

To focus on relevant areas of the image, a masking process is applied by defining a polygon covering the region where lane lines are expected. A blank matrix is created, and the selected polygon is filled with white to isolate the lane detection area.

To enhance visibility under low-light conditions, gamma correction is applied. The gamma value is dynamically adjusted based on the average brightness of the frame:

$$\gamma = \frac{-0.3}{\log_{10}(Y_{\mathrm{average}}+\epsilon)} \quad (18)$$

$$\gamma = 0.7 - \gamma$$

where Y_{average} represents the mean brightness level of the frame. Besides, HSV filtering is used to extract lane markings by filtering yellow and white pixels:

$$HSV = f_{convert}(I_{RGB}) \tag{19}$$

$$mask_{yellow} = \begin{cases} 1, & if\ min\ \_val_y \leq HSV \leq max\_val_y \\ 0, & otherwise \end{cases} \tag{20}$$

$$mask_{white} = \begin{cases} 1, & if\ min\ \_val_y \leq HSV \leq max\_val_y \\ 0, & otherwise \end{cases} \tag{21}$$

$$mask = mask_{yellow} \vee mask_{white} \tag{22}$$

$$I_{filtered} = I_{RGB} \cdot mask \tag{23}$$

B) Inverse Perspective Mapping (IPM)

To provide a bird's-eye view of the road, IPM is performed. The transformation involves mapping ROI points from the original image to a new perspective:

$$P_{src} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \end{bmatrix} \tag{24}$$

$$P_{dst} = \begin{bmatrix} 0 & w-1 & w-1 & 0 \\ 0 & 0 & h-1 & h-1 \end{bmatrix} \tag{25}$$

where *w* and *h* represent the width and height of the output image. The homography matrix *H* maps points from the source plane to the destination plane.

$$H = \arg min \sum_{i=1}^{4} \left\| P_{dst}^{(i)} - H\, P_{src}^{(i)} \right\|^2 \tag{26}$$

$$s \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{27}$$

where s is a scaling factor, and (x', y') are the transformed coordinates. The final inverse perspective-mapped image is obtained as:

$$I_{IPM}(x', y') = I_{original}(H^{-1}(x', y', 1)^T) \tag{28}$$

3.3.2 Lane Tracking at Night

At night, lane tracking is performed using Kalman filter which predicts lane positions and corrects them based on new measurements. This ensures smooth and accurate tracking even in low-light conditions. The Kalman filter is initialized with the state size, measurement size, and transition matrices. The state vector includes lane line positions and velocities:

$$X_k = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} \tag{29}$$

The state transition matrix model's lane movement is given by:

$$F = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{30}$$

where Δt is the time step. The measurement matrix maps the predicted state to the observed lane line positions:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{31}$$

The Kalman Gain is computed to weigh new measurements:

$$K_k = P_k H_k^T (H_k P_k H_k^T + R_k)^{-1} \tag{32}$$

where P_k is the error covariance matrix, and R_k represents measurement noise covariance. The state is updated as follows:

$$x_k = x_k^- + K_k(z_k - H_k x_k^-) \tag{33}$$

$$P_k = (I - K_k H_k) P_k^- \tag{34}$$

The predicted state and covariance matrices are updated as,

$$x_k^- = F_k x_{k-1} \tag{35}$$

$$P_k^- = F_k P_{k-1} F_k^T + Q_k \tag{36}$$

where Q_k is the process noise covariance matrix. By integrating Probabilistic Hough Transform with Kalman filtering, the system ensures accurate lane detection and tracking, even in challenging nighttime conditions.

3.4 Adaptive Lane Detection and Tracking for Day and Night Conditions

The proposed lane detection system dynamically selects the appropriate method for lane detection and tracking based on the lighting conditions. The proposed method first determines whether the scene is during the day or night by calculating the average brightness of a frame and comparing it to a predefined threshold. The image is converted to grayscale, and the mean intensity is computed as follows,

$$B = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} I_{gray}(i,j) \tag{37}$$

where B is the average brightness, $I_{gray}(i,j)$ is the pixel intensity at position (i,j) in the grayscale image, and M and N are the image dimensions. A threshold T is used to classify the scene,

$$D = \begin{cases} 1, & if\ B > T (Daytime) \\ 0, & if\ B \leq T (Nighttime) \end{cases} \tag{38}$$

where $D = 1$ indicates daytime, $D = 0$ indicates nighttime, and $T = 50$ is the predefined threshold value. If daytime is detected, the method applies daytime lane detection, which includes ROI masking and line averaging to enhance lane visibility. If nighttime is detected, the method uses night lane detection which incorporates Probabilistic Hough Transform with Kalman filter for improved lane estimation in low-light conditions.

In the proposed method, the transition between daytime and nighttime modes is determined by the average pixel brightness of the frame. We plan to investigate further ways to handle extreme lighting transitions, such as tunnels or areas with frequent lighting changes, in future work.

4. EXPERIMENTAL RESULTS

4.1 Dataset

In making the evaluation of the new system, a video dataset was created, and a collection of videos with different driving conditions was assembled. The dataset comprises assorted factors, including lighting conditions (day and night), road types (straight as well as curved), and traffic density. This variability allows every aspect of the lane detection system, and under different conditions, its reliability can be confirmed.

Algorithm 1: Pseudocode for Adaptive Lane Detection and Tracking for Day and Night Conditions

```
BEGIN Lane_Detection_System
    FUNCTION determine_lighting(frame):
        gray_frame = convert_to_grayscale(frame)
        brightness = compute_average_brightness(gray_frame)
        IF brightness > DAYTIME_THRESHOLD:
            RETURN "day"
        ELSE:
            RETURN "night"

    FUNCTION compute_additional_measurements(frame):
        gamma_corrected = apply_gamma_correction(frame)
        hsv_filtered = apply_HSV_filtering(gamma_corrected)
        lane_edges = detect_edges(hsv_filtered)
        lane_lines = apply_Hough_Transform(lane_edges)
        lane_boundaries = detect_lane_boundaries(lane_lines)
        scaled_lines = scale_lines_to_horizon(lane_boundaries)
        masked_frame = apply_region_of_interest(scaled_lines)
        RETURN masked_frame, lane_lines

    FUNCTION choose_detection_method(frame):
        lighting_condition = determine_lighting(frame)
        masked_frame, lane_lines = compute_additional_measurements(frame)

        IF lighting_condition == "day":
            confidence = evaluate_CV_method(masked_frame, lane_lines)
            IF confidence is HIGH:
                RETURN "computer_vision"
            ELSE:
                RETURN "kalman_filter"

         ELSE:  // Nighttime
            confidence = evaluate_Kalman_Filter(masked_frame, lane_lines)
            IF confidence is HIGH:
                RETURN "kalman_filter"
            ELSE:
                RETURN "computer_vision"

    FUNCTION detect_lanes(frame, method):
        IF method == "computer_vision":
            grayscale = convert_to_grayscale(frame)
            blurred = apply_gaussian_blur(grayscale)
            edges = apply_canny_edge_detection(blurred)
            lines = apply_Hough_Transform(edges)
            averaged_lines = average_lane_lines(lines)
            final_lanes = draw_lane_lines(frame, averaged_lines)
            RETURN final_lanes

        ELSE IF method == "kalman_filter":
            initialize_Kalman_Filter()
            predicted_state = Kalman_Predict()
            updated_state = Kalman_Update(predicted_state, frame)
            final_lanes = draw_lane_lines(frame, updated_state)
            RETURN final_lanes

    FUNCTION process_video(video):
        FOR each frame in video:
            detection_method = choose_detection_method(frame)
            lane_lines = detect_lanes(frame, detection_method)
            display_lane_lines(frame, lane_lines)
END Lane_Detection_System
```

The dataset comprises ten videos (with a total of 36,000 frames) collected from publicly available platforms, such as the Udacity Self Driving Car Dataset, coupled with different video clips from various online initiatives and Shutterstock for additional differentiation. The front camera of the car filmed driving videos. To enhance the reliability of the evaluation, the videos were collected from separate sources, and they were shot at times when the day was busy or quiet.

To facilitate effective evaluation, each video in the dataset was annotated frame by frame. The annotations include metadata regarding the lighting condition (e.g., daytime, nighttime, low light), road type (e.g., straight, curved), and traffic density (e.g., high, medium, low). These contextual tags allow the evaluation of lane detection performance under various conditions. While annotations are not pixel-level ground truth, they may be used to systematically label environment factors and support qualitative and quantitative performance analysis.

4.2 Evaluation Metrics

The performance metrics used for lane detection include consistency, distance, slope, and detection of both lines.

- **Consistency** - Evaluates the stability of lane detection across consecutive frames. It is computed as,

$$Consistency = \frac{Number\ of\ consistent\ frames}{Total\ number\ of\ frames} \tag{39}$$

- **Distance** Measures the accuracy of detected lane positions by calculating the distance between the detected lane lines,

$$Distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{40}$$

- **Slope** - Assesses the correctness of lane angle detection by computing the slope of detected lane lines,

$$Slope = \frac{y_2 - y_1}{x_2 - x_1} \tag{41}$$

- **Detection of Both Lines** - Evaluates whether both lane lines are consistently detected,

$$Detection\ of\ Both\ Lines = \frac{Number\ of\ frames\ with\ both\ lines\ detected}{Total\ number\ of\ frames} \tag{42}$$

Table 2 lists the important symbols, parameters, and notations used in the manuscript.

Table 2. A summary of symbols and notations

Symbol/Parameter	Description	Unit
$\boldsymbol{x_1, y_1}$	Coordinates of the first point on the detected lane line	Pixels
$\boldsymbol{x_2, y_2}$	Coordinates of the second point on the detected lane line	Pixels
Consistency	Stability of detected lane lines across consecutive frames.	Ratio
Distance	Euclidean distance between detected left and right lane lines.	Pixels
Slope	Slope of the detected lane lines.	Gradient
Detection of lines	Checks if both the left and right lane lines are detected.	Ratio
N	Total number of frames evaluated	Count
ROI	ROI used for masking irrelevant parts of the image	-
FPS	Frames Per Second – processing speed of the algorithm	Frames/Second

4.3 Assessment of Lane Detection Methods

The numerical results for the evaluation metrics used to assess the performance of the lane detection methods are presented in Table 3. The results are averaged across multiple videos to provide a comprehensive evaluation.

Table 3. Performance of Lane Detection Methods

Metric	Day Time Lane Detection with ROI Masking and Line Averaging	Night Lane Detection with Probabilistic Hough Transform
Consistency	1.72	0.62
Distance	271.64	298.64
Slope	0.82	0.61
Detection of Lines	0.81	0.62

The consistency metric evaluates the stability of the identified lane lines across back-to-back frames of video. On the average, the nighttime lanes detection system gives a consistency score of 0.62 across the paths, while the daytime lanes detection system records a score of 1.72 along the loops. This indicates that the daytime system is a more consistent method of lane formation over the course of time.

The metric of distance investigates both the median distance between respective left and right lanes. The night lane detection technique shows an average distance between lanes of 298.64 pixels, while the daytime method gives us 271.64 pixels apart. Thus, we find that rather than the nighttime lanes being closely spaced, it is the daytime lanes that are spaced closely together.

The slope metric is to determine the angle detection of lanes. The night slope coefficient is 0.61, in the same way that the daytime Coefficient is 0.82. This means that the daytime method records lanes which are slightly steeper in slope.

The capacity of the system to detect the lane borders of left and right simultaneously via both lane lines is assessed through the detection of them. The daylight approach thus has a higher detection rate of 0.81, and the nighttime lane detection method is at 0.62. This indicates that both lane lines may be reliably detected using the daytime lane detection approach.

4.4 Comparison with Baseline Approaches

A comparative analysis between the proposed method under both day and nighttime conditions is presented in Table 4. The proposed method consistently outperforms the baselines across all metrics.

Table 4. Comparison with baseline approaches

Method	Scenario	Consistency	Distance	Slope	Detection of Lines
Proposed Method	Day-Straight	1.72	271.64	0.82	0.81
Hough + Static ROI	Day-Straight	1.20	285.30	0.65	0.70
Canny + Line Fit	Day-Straight	0.90	310.12	0.60	0.60
Proposed Method	Night-Curved	0.62	298.64	0.61	0.62
Hough + Static ROI	Night-Curved	0.40	320.00	0.55	0.50
Canny + Line Fit	Night-Curved	0.35	345.50	0.48	0.40

In the daytime-straight road scenario, it achieves the highest consistency score. It also records a lower average distance between detected lanes compared to the baselines. Besides, the proposed method shows a better slope estimation and a higher detection rate of both lane lines. In the more challenging night-curved road scenario, although all methods experience performance degradation due to poor lighting and road geometry, the proposed method still demonstrates promising results.

4.5 Visualization of Daytime Detection and Tracking Results

The proposed lane detection method was evaluated on videos captured during the daytime under various driving conditions. It was tested on both straight and curved roads, as well as under different lighting conditions. Figure 8 presents tracking results on a straight road with white lane lines, while Figure 9 illustrates tracking results on a straight road with yellow lane lines. The lane detection system accurately identified the lane lines in both scenarios. This aligns with the performance metrics which indicate high consistency and reasonable lane distance.

Besides, the method effectively detected lane lines on a curved road with shadows as shown in Figure 10. The performance metrics for this scenario also exhibited high consistency that validates the robustness of the daytime lane detection approach.

Figure 8. Tracking Results on a Straight Road Containing White Lane Lines during the Daytime

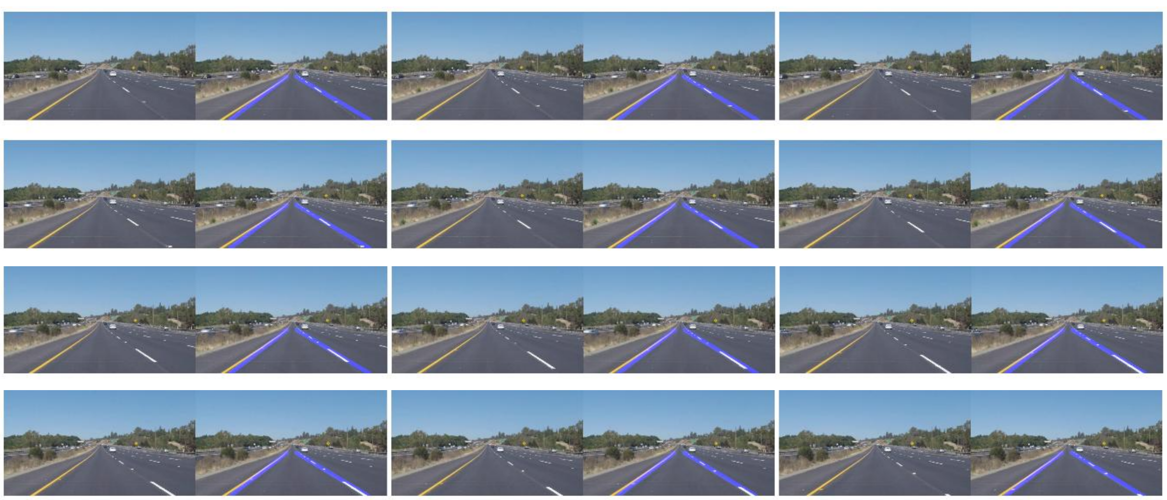

Figure 9. Lane Line Detection Results on a Straight Road Containing Yellow Lines during the Day

Figure 10. Tracking Results on a Curved Road with Shadows during the Daytime

4.6 Visualization of Nighttime Detection and Tracking Results

The lane detection method was also tested on videos captured at night. Figure 11 presents the detection results on a curved road at night. We observe that the proposed method successfully tracked both lane lines. Although the evaluation metrics were less favourable due to extremely dark lighting, the video results confirm successful lane detection using the proposed method.

Besides, the system effectively detected lane lines in a low-traffic nighttime scenario as shown in Figure 12. The performance results indicate accurate lane detection and tracking of both lines using the nighttime lane detection method. Despite the evaluation metrics not reflecting success due to poor lighting, the video results validate the system's effectiveness.

Besides, the system performed well on a road with moderate traffic at night as illustrated in Figure 13. The performance metrics demonstrate high consistency, reasonable distance, valid slope, and successful detection of both lane lines, confirming the reliability of the nighttime detection method. Lastly, Figure 14 showcases the system’s ability to detect lane lines in extremely low-light conditions where the proposed nighttime detection method alone achieved accurate lane detection and tracking of both lines.

4.7 Discussions

While the proposed system performs well for most of the challenging conditions, several limitations have been encountered. First, when large vehicles or shadows occlude a significant portion of lane lines, the Hough Transform may detect false positives or miss one line entirely. Besides, in some real-world cases with faded lane markings, the system cannot maintain consistency over successive frames. These limitations suggest the need for integrating deep learning-based semantic segmentation in the future work.

Regarding computational performance, the current system has a processing speed of 30 frames per second (fps) when executed on a standard desktop environment equipped with an Intel Core i7-9700K Central Processing Unit (CPU) @ 3.6 GHz and 16 GB RAM. This frame rate is adequate for real-time applications in simulation or PC-based environments. However, deploying this solution in embedded or automotive-grade hardware may require further optimization due to limited processing power and energy constraints. In future work, we plan to explore algorithm optimization techniques, such as model compression for real-time implementation.

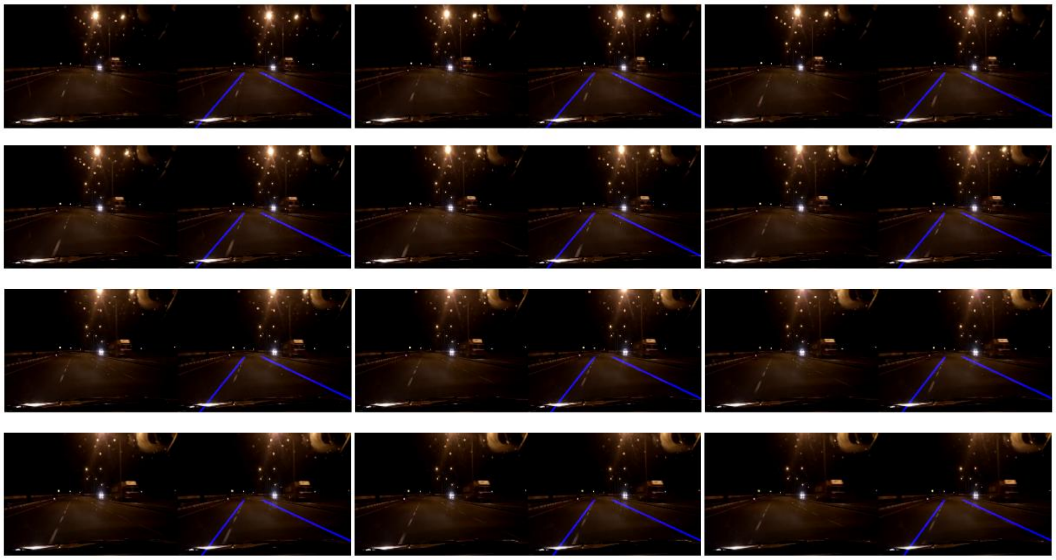

Figure 11. Detection Results on Lane Lines on a Curved Road during the Night

Figure 12. Detection Results of Lane Lines in a Low Traffic Condition during the Night

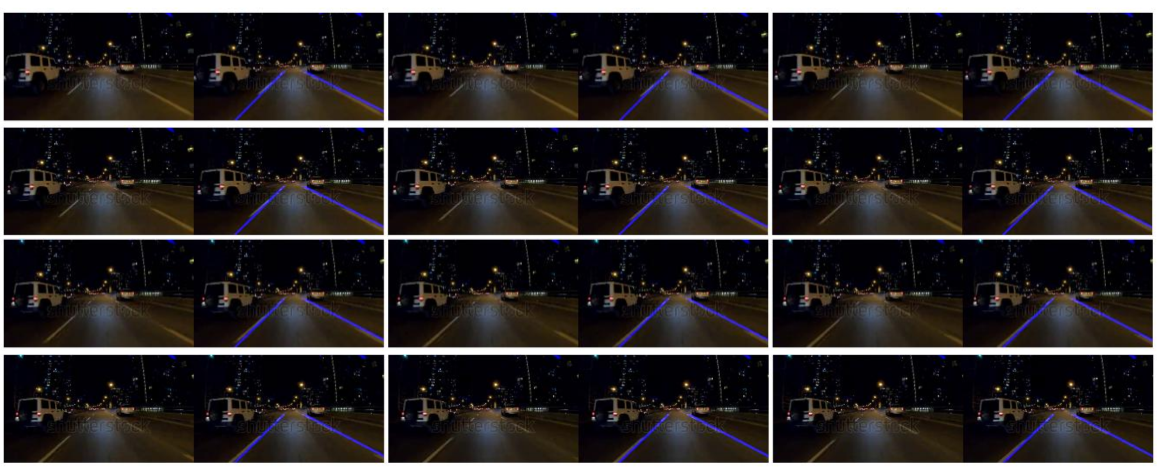

Figure 13. Detection Results of Lane Lines on a Road with Some Traffic Condition during the Night

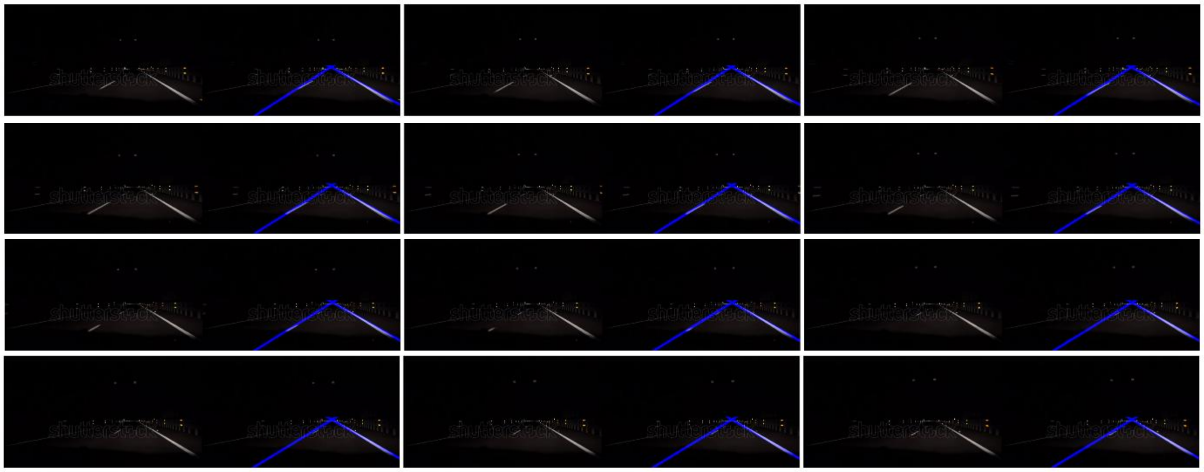

Figure 14. Detection Results of Lane Lines with Extreme Low Lighting

5. CONCLUSION

The proposed lane detection approach was experimented with videos captured under various driving conditions and with various performance metrics to evaluate the effectiveness of the system. Daytime lane detection algorithm performed well during daytime. Night-time lane detection algorithm had improved consistency, distance accuracy, slope stability, and detection of both lane lines, particularly in low-light conditions. The study comes up with an adaptive evaluation approach to select optimally the most appropriate detection method according to illumination conditions to better the overall system adaptability. The proposed lane detection method can potentially contribute to improving the safety of roads in autonomous vehicles. Future research will be directed towards the integration of state-of-the-art machine learning approaches such as deep learning models for even greater accuracy and robustness.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their valuable comments.

FUNDING STATEMENT

The authors received no funding from any party for the research and publication of this article.

AUTHOR CONTRIBUTIONS

Habibatelrahman Mosaad Abdelfattah Mohamed Abdelaal: Data Curation, Methodology, Validation, Writing – Original Draft Preparation;
Tee Connie: Conceptualization, Supervision, Writing – Review & Editing;
Michael Kah Ong Goh: Project Administration, Supervision, Writing – Review & Editing.

CONFLICT OF INTERESTS

No conflict of interests were disclosed.

ETHICS STATEMENTS

Our publication ethics follow The Committee of Publication Ethics (COPE) guideline. https://publicationethics.org/

REFERENCES

[1] N. Sukumar, and P. Sumathi, "An Improved Lane Detection and Lane Departure Warning Framework for ADAS," *IEEE Trans. Consum. Electron.*, vol. 70, no. 2, pp. 4793–4803, May 2024, doi: 10.1109/TCE.2024.3387708.

[2] M. Qiu, L. Christopher, S.Y.-P. Chien, and Y. Chen, "Intelligent Highway Adaptive Lane Learning System in Multiple ROIs of Surveillance Camera Video," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 8, pp. 8591–8601, Aug. 2024, doi: 10.1109/TITS.2024.3358732.

[3] J. Bi *et al.*, "Lane Detection for Autonomous Driving: Comprehensive Reviews, Current Challenges, and Future Predictions," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 5, pp. 5710–5746, May 2025, doi: 10.1109/TITS.2024.3524603.

[4] N. Pourjafari, A. Ghafari, and A. Ghaffari, "Navigating Unsignalized Intersections: A Predictive Approach for Safe and Cautious Autonomous Driving," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 269–278, Jan. 2024, doi: 10.1109/TIV.2023.3321275.

[5] D. Annunziata, D. Chiaro, P. Qi, and F. Piccialli, "On the Road to AIoT: A Framework for Vehicle Road Cooperation," *IEEE Internet Things J.*, vol. 12, no. 5, pp. 5783–5791, Mar. 2025, doi: 10.1109/JIOT.2024.3488855.

[6] J. Liu *et al.*, "Semantic Traffic Law Adaptive Decision-Making for Self-Driving Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 14858–14872, Dec. 2023, doi: 10.1109/TITS.2023.3294579.

[7] X. Luo, Y. Huang, J. Cui, and K. Zheng, "Deep learning-based lane detection for intelligent driving: A comprehensive survey of methods, datasets, challenges and outlooks," *Neurocomputing*, vol. 650, p. 130795, Oct. 2025, doi: 10.1016/j.neucom.2025.130795.

[8] F. You *et al.*, "Attention based network for real-time road drivable area, lane line detection and scene identification," *Eng. Appl. Artif. Intell.*, vol. 160, p. 111781, Nov. 2025, doi: 10.1016/j.engappai.2025.111781.

[9] I.-C. Sang, and W.R. Norris, "Improved generalizability of CNN based lane detection in challenging weather using adaptive preprocessing parameter tuning," *Expert Syst. Appl.*, vol. 275, p. 127055, May 2025, doi: 10.1016/j.eswa.2025.127055.

[10] L. Sun, H. Zhu, and W. Qin, "SP-Det: Anchor-based lane detection network with structural prior perception," *Pattern Recognit. Lett.*, vol. 188, pp. 60–66, Feb. 2025, doi: 10.1016/j.patrec.2024.11.030.

[11] Z. Sun, "Vision Based Lane Detection for Self-Driving Car," in *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA)*, Aug. 2020, pp. 635–638. doi: 10.1109/AEECA49918.2020.9213624.

[12] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, "Robust Lane Detection From Continuous Driving Scenes Using Deep Neural Networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 41–54, Jan. 2020, doi: 10.1109/TVT.2019.2949603.

[13] Z. Wang, W. Ren, and Q. Qiu, "LaneNet: Real-Time Lane Detection Networks for Autonomous Driving," Jul. 04, 2018, *arXiv*: arXiv:1807.01726. doi: 10.48550/arXiv.1807.01726.

[14] J.J. Ng, K.O.M. Goh, and C. Tee, "Traffic Impact Assessment System using Yolov5 and ByteTrack," *J. Inform. Web Eng.*, vol. 2, no. 2, Art. no. 2, Sep. 2023, doi: 10.33093/jiwe.2023.2.2.13.

[15] M. Rao, L. Paulino, V. Robila, I. Li, M. Zhu, and W. Wang, "Training-Free Lane Tracking for 1/10th Scale Autonomous Vehicle Using Inverse Perspective Mapping and Probabilistic Hough Transforms," in *2022 IEEE Integrated STEM Education Conference (ISEC)*, Mar. 2022, pp. 406–411. doi: 10.1109/ISEC54952.2022.10025098.

[16] J.H. Ang, and T.S. Min, "A Review on Sensor Technologies and Control Methods for Mobile Robot with Obstacle Detection System," *Int. J. Robot. Autom. Sci.*, vol. 6, no. 1, Art. no. 1, Apr. 2024, doi: 10.33093/ijoras.2024.6.1.11.

[17] S. Anbalagan, P. Srividya, B. Thilaksurya, S.G. Senthivel, G. Suganeshwari, and G. Raja, "Vision-Based Ingenious Lane Departure Warning System for Autonomous Vehicles," *Sustainability*, vol. 15, no. 4, Art. no. 4, Jan. 2023, doi: 10.3390/su15043535.

[18] M. Meuter, S. Muller-Schneiders, A. Mika, S. Hold, C. Nunn, and A. Kummert, "A novel approach to lane detection and tracking," in *2009 12th International IEEE Conference on Intelligent Transportation Systems*, Oct. 2009, pp. 1–6. doi: 10.1109/ITSC.2009.5309855.

[19] Z.Y. Lim, and K.S. Sim, "Multi-Color Code with High Data Capacity," *Int. J. Robot. Autom. Sci.*, vol. 4, pp. 35–45, Jul. 2022, doi: 10.33093/ijoras.2022.4.6.

[20] R.N. Grant, R.D. Green, and A.J. Clark, "HLS Distorted colour model for enhanced colour image segmentation," in *2008 23rd International Conference Image and Vision Computing New Zealand*, Nov. 2008, pp. 1–6. doi: 10.1109/IVCNZ.2008.4762139.

BIOGRAPHIES OF AUTHORS

	**Habibatelrahman Mosaad Abdelfattah Mohamed Abdelaal** is student at Multimedia University pursuing Bachelor's Degree in Computer Science (Artificial Intelligence). Her main research interests are in Artificial intelliegrnce, Computer Vision and IOT. She is also eager to learn new information from various fields specially computer science. Habiba can be contacted at email: habibamosaad2003@gmail.com.
	**Tee Connie** received her PhD (IT) degrees from Multimedia University in 2015. She is a Professor in the Faculty of Information Science and Technology Multimedia University. She is currently holding the position as the Dean of Institute for Postgraduate Studies in same the university. Her main research areas are pattern recognition, machine learning and computer vision. She can be contacted at email: tee.connie@mmu.edu.my.
	**Michael Kah Ong Goh** is an Associate Professor in the Faculty of Information Science & Technology (FIST) at Multimedia University (MMU), Malaysia. His current research interests encompass pattern recognition, image processing, data classification, video analytics, fusion, and computer vision, with a specific focus on automated hand-based multimodal biometrics. He had successfully published over 40 international journal articles and 35 conference papers. He can be contacted at email: michael.goh@mmu.edu.my.