J. Electrical Systems 20-11s (2024): 3878-3898

[1] Spriha Deshpande
Santa Clara

Steering toward Safety: Real-Time Lane Detection for Autonomous Navigation

Journal of Electrical Systems

Abstract— The advancement of autonomous driving technology has led to the development of various systems aimed at improving vehicle safety and navigation. This paper presents a lane detection system designed to enhance road safety using edge detection, lane departure warnings, road condition assessment, and optical flow estimation. By leveraging traditional computer vision techniques, including Canny edge detection, Hough transform, and Kalman filtering for temporal smoothing, the system is capable of detecting lane markings and assessing the condition of road surfaces in real-time. Additional features, such as the calculation of lane curvature and optical flow, allow for motion estimation and the detection of vehicles. The methodology also incorporates lane departure warning functionality and a robust approach to detecting lane divergence. The system's performance is evaluated using video datasets, demonstrating the potential of these techniques for practical deployment in autonomous driving applications.

Keywords— Autonomous Driving, Lane Detection, Road Safety, Edge Detection, Lane Departure Warning, Optical Flow Estimation, Hough Transform, Canny Edge Detection, Lane Curvature Calculation, Road Condition Assessment.

I. Introduction

Autonomous driving technology has seen rapid developments in recent years, with advanced driver-assistance systems (ADAS) making significant strides toward full autonomy. Lane detection, one of the critical aspects of ADAS, is essential for ensuring safe vehicle operation and navigation within designated lanes. Accurate lane detection is crucial for preventing accidents due to lane departure and for assisting in tasks such as lane keeping, collision avoidance, and road condition assessment [1,2].

In this paper, we propose a lane detection system that focuses on real-time processing using computer vision techniques, without the need for deep learning models. Our system employs methods such as Canny edge detection, Hough transform, and optical flow analysis to detect lanes, assess road conditions, and detect the motion of surrounding objects. These methods have been shown to be effective for lane detection in various conditions, including challenging environmental factors such as weather and road surface conditions [3,5]. Additionally, the system integrates a lane departure warning mechanism that alerts the driver if the vehicle is deviating from its lane, a key feature for ensuring safety in autonomous driving applications [6].

The proposed system is tested on video datasets, and we evaluate its effectiveness in real-time applications for autonomous driving. Through comprehensive unit testing, we ensure that each component of the system functions optimally, providing reliable lane detection and safety warnings in dynamic driving scenarios [9]. The results demonstrate the potential of these techniques for practical deployment in autonomous driving systems [11].

II. Problem Statement

Lane detection is a critical component in the development of Advanced Driver Assistance Systems (ADAS) and autonomous vehicles. The ability to accurately detect lanes on various types of roads under diverse environmental conditions (e.g., different lighting, weather, and road surfaces) is essential for ensuring the safety and reliability of such systems. Traditional methods, such as edge detection and Hough Transform, have been widely used for lane detection. However, these approaches face challenges in handling complex real-world scenarios, such as poorly marked lanes, occlusions, and dynamic environments (e.g., vehicles, pedestrians, or other obstacles on the road). Additionally, real-time processing and computational efficiency remain a significant concern for deploying lane detection systems in practical applications.

[1]spriha.deshpande@gmail.com

This research aims to address these challenges by developing a robust lane detection system that combines traditional computer vision techniques with temporal smoothing (via Kalman filters) for improved accuracy and stability. The system must be capable of detecting lanes under a wide range of conditions, including road curvature, varying lane widths, and potential lane departures. Furthermore, it should provide valuable insights into road conditions, lane curvature, and vehicle motion, including detecting lane departure warnings and calculating optical flow for motion estimation. The primary goal of this study is to enhance the reliability, efficiency, and real-time performance of lane detection algorithms to support autonomous driving and ADAS applications.

III. FORMULAS

A. *Lane Curvature Formula:* The lane curvature is calculated by fitting a second-degree polynomial to the detected lane points. The formula $y=ax^2+bx+c$ is used for the curve fitting. The curvature is derived from the coefficient aaa of the polynomial. The formula calculates the curvature at the center of the lane, and *2a* gives the curvature for the curve, where a higher value indicates a sharper curve per *formula (1)* below:

$$Curvature = 2 \times a$$

$$(From\ the\ polynomial\ equation\ y = ax^2 + bx + c) \tag{1}$$

B. *Lane Departure Warning Divergence Check:* This formula calculates the width of the lane by finding the horizontal distance between the left and right detected lanes. It is used to check if the lanes are diverging significantly. If the distance exceeds a certain threshold, it may indicate that the vehicle is moving out of the lane per *formula (2)* below:

$$Lane\ Width = |\ right\ lane\ center - left\ lane\ center| \tag{2}$$

C. *Lane Departure Warning Center Check: This formula calculates a margin of error (10% of the lane width). This margin is used to check if the vehicle is moving out of the lane. If the vehicle is too far from the center of the lane (either left or right), a lane departure warning is triggered per formula (3) below:*

$$Margin\ of\ Error = Lane\ Width \times 0.1 \tag{3}$$

D. *Optical Flow Magnitude (Motion Detection):* This formula calculates the average magnitude of motion using optical flow, where u_i and v_i represent the flow vectors in the horizontal and vertical directions, respectively. This helps in estimating motion in the frame, such as vehicles moving or other moving objects in the scene per *formula (4)* below:

$$Magnitude = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(u_i^2 + v_i^2)} \tag{4}$$

E. *Road Condition Detection (Texture Analysis):* The variance of the Laplacian is a measure of the image's texture detail. A higher variance indicates a smoother surface (good road condition), while a lower variance suggests a lack of detail, which could be due to worn-out road markings (poor road condition) per *formula (5)* below:

$$Laplacian\ Variance = Variance\ of\ Laplacian\ of\ the\ grayscale\ image \tag{5}$$

F. *Canny Edge Detection:* The Canny edge detection algorithm is used to detect edges in an image. It works by finding areas where there is a significant change in intensity (edges). The algorithm applies three main steps in *formula (6)* below:

- Gradient Calculation: The gradient of the image is calculated to identify regions of intensity change.
- Non-maximum Suppression: Thin out the edges by removing pixels that are not part of the edge.
- Hysteresis Thresholding: Apply two thresholds, *low* and *high*, to determine which edges are strong enough to be considered as part of the object boundaries.

$$Edges = Canny(I, low\ threshold, high\ threshold) \tag{6}$$

G. *Hough Transform for Line Detection:* The Hough Transform is used to detect straight lines in an image. The transform works by converting each point in the image from Cartesian coordinates *(x,y)* to Hough space *(ρ,θ)* in *formula (7) and (8)* where:

- ρ is the perpendicular distance from the origin to the line.
- θ is the angle of the line with respect to the x-axis.

- After the transformation, points that lie on the same line will produce intersecting curves in Hough space. The intersection corresponds to the parameters *(ρ,θ)* of the line in the image.
- The algorithm then accumulates votes for potential lines, with higher votes indicating the presence of a line in the image. The lines are then drawn based on the detected parameters.

$$\rho = x\,cos(\theta) + y\,sin(\theta) \tag{7}$$

$$\theta = arctan\left(\frac{y}{x}\right) \tag{8}$$

H. *Kalman Filtering for Temporal Smoothing:* Kalman Filtering is a recursive algorithm that estimates the state of a system from noisy observations. In the context of lane detection system, Kalman filtering is used for temporal smoothing of detected lanes or other vehicle-related states (e.g., position, velocity). It works in two steps:

- Prediction: Predict the next state of the system based on the previous state and control input (if any). This is done using the transition matrix A and control input matrix B in *Formula (10)*
- Update: When a new measurement (e.g., lane position) is available, update the predicted state using the measurement and the Kalman gain K_k, which is calculated based on the prediction uncertainty and measurement uncertainty in *Formula (11)*
- The Kalman filter minimizes the estimation error by balancing the predicted state and the observed measurement, providing a more accurate estimate over time. The matrices A, B, H, P, and R define the system model, control inputs, measurement model, error covariance, and measurement noise, respectively in *Formula (12) and (13)*

$$Prediction: A \cdot \hat{x}_{k-1|k-1} + B \cdot u_k \tag{10}$$

$$Update: K_k = P_{k|k-1} \cdot H^T \cdot (H \cdot P_{k|k-1} \cdot H^T + R)^{-1} \tag{11}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \cdot (z_k - H \cdot \hat{x}_{k|k-1}) \tag{12}$$

$$P_{k|k} = (I - K_k \cdot H) \cdot P_{k|k-1} \tag{13}$$

IV. Methodology

A. *Image Preprocessing*

The first step in lane detection is preprocessing the image to enhance relevant features for edge detection. The system converts the input video frame into grayscale using OpenCV's cv2.cvtColor function. To further improve the quality of the image and reduce noise, we apply Contrast Limited Adaptive Histogram Equalization (CLAHE). CLAHE improves the contrast in localized regions, making the lane markings more prominent. After this, we apply a Gaussian blur to reduce high-frequency noise and smooth the image, making edge detection more accurate.

B. *Edge Detection*

To detect edges in the preprocessed image, we use Canny edge detection. The Canny edge detector is an edge-detection operator that highlights areas of rapid intensity change, which typically correspond to lane markings on the road. We adjust the thresholds for edge detection to make the system more sensitive to lane markings. The edges detected in the image will form the basis for lane detection.

C. *Lane Detection with Hough Transform*

The detected edges are then used for lane detection using the Hough Line Transform. The Hough Transform detects straight lines by transforming the points in the image space to a parameter space. The system utilizes a probabilistic version of the Hough Transform (cv2.HoughLinesP), which is computationally more efficient. We set parameters such as the threshold for line detection, minimum line length, and the maximum gap between line segments. The output of this step is a set of detected lanes represented by straight lines on the image.

D. Lane Curve Detection

For curved lanes, a polynomial curve is fitted to the detected lane points. Using the second-degree polynomial fitting method (np.polyfit), we estimate the curvature of the lane. The curvature is calculated from the coefficients of the fitted polynomial, and this information is crucial for assessing the sharpness of turns. The fitted curve is drawn on the image to visualize the detected lane curvature.

E. Road Condition Detection

Road condition assessment is done through texture analysis. We calculate the variance of the Laplacian of the grayscale image, which indicates the amount of texture detail present. High variance corresponds to good road conditions (clear lane markings), while low variance suggests poor road conditions (worn-out or faded lane markings). This information is printed in the terminal and can be used for road maintenance or driver awareness.

F. Lane Departure Warning

The system detects lane departure by checking the relative position of the detected lane markings. The center of the detected lanes is calculated, and if the vehicle moves significantly away from the lane center, a lane departure warning is triggered. Additionally, if the lanes are diverging significantly, a warning is issued to alert the driver that the vehicle may be drifting out of the lane.

G. Optical Flow for Motion Estimation

To detect the motion of surrounding vehicles or objects, optical flow is computed between consecutive frames. We use OpenCV's cv2.calcOpticalFlowFarneback method, which calculates the flow vectors between the previous and current grayscale frames. The average magnitude of these vectors indicates the amount of motion. This information can be useful for estimating vehicle speed or detecting fast-moving objects on the road.

H. Kalman Filter for Temporal Smoothing

A Kalman filter is used to smooth the lane detection results over time. This filter helps reduce the noise in the lane detection process by considering both the current measurement and the prior estimates, improving the accuracy of lane position tracking.

I. System Evaluation

The lane detection system is evaluated on video data, with results presented in the form of detected lanes, curvature information, road condition detection, lane departure warnings, and optical flow motion estimates. The system is tested on various road scenarios to demonstrate its robustness in real-world driving conditions.

V. FLOW AND ARCHITECTURE

The framework involves multiple stages to process accurate lane detection, assess road conditions, and provide feedback, such as lane departure warnings. Below in *Fig 1.* is a detailed outlined of the flow and architecture of the system.

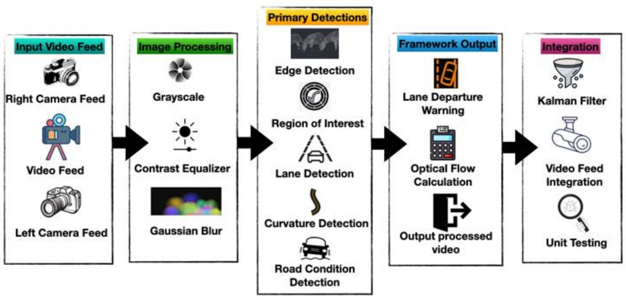

Fig. 1. **Lane Detection System: Flow and Architecture**

A. Input Video Feed

The system begins by capturing video feeds from multiple cameras installed in the vehicle. These feeds provide a real-time representation of the vehicle's environment, crucial for understanding lane positioning and road conditions. The cameras involved are:

- Right Camera Feed: Captures the video from the right side of the vehicle to provide a broader environmental view.
- Left Camera Feed: Captures video from the left side, complementing the right camera feed to form a complete image of the surroundings.
- Video Feed: This may be a combination of the individual camera feeds or a primary feed that integrates data from both the right and left cameras.

The video is processed frame by frame to extract features relevant to lane detection, road condition analysis, and more.

B. Image Processing

Once the video feed is available, the system performs a series of image preprocessing steps to enhance the quality of the input data, making it more suitable for detection algorithms:

- Grayscale Conversion: Converts the image to grayscale, which simplifies the processing by reducing the number of channels (from RGB to a single intensity channel). This helps in focusing on key features like edges and contours without being affected by color.
- Contrast Limited Adaptive Histogram Equalization (CLAHE): This technique enhances the contrast in the image, improving the visibility of road features, especially under poor lighting conditions. CLAHE locally adjusts the intensity of different regions, enhancing the details needed for edge detection and lane marking.
- Gaussian Blur: A smoothing technique used to reduce noise in the image. Noise reduction is critical because it ensures that edge detection algorithms perform more effectively by removing irrelevant information.

These preprocessing steps prepare the image for the next stage: edge detection.

C. Primary Detections

Once the image is preprocessed, several primary detections are performed on the image to extract meaningful data for further analysis:

- Edge Detection: The Canny edge detection algorithm is employed to highlight the edges in the image. Edges typically represent boundaries between different regions, such as lanes, road markings, and other relevant structures. The threshold values for the Canny edge detection can be fine-tuned to adapt to varying road conditions.
- Region of Interest (ROI): A mask is applied to focus on the region of interest, which is typically the lower half of the image. This region is where the lane markings and road signs are most likely to appear. By narrowing the focus, the system reduces computational complexity and improves accuracy.
- Lane Detection: The system uses the Hough Line Transform (a technique for detecting straight lines) to detect lane markings within the region of interest. This involves detecting lines that represent the left and right lanes of the road. The system stores the coordinates of the lane markings, which are used in later steps for lane curvature detection and lane departure warnings.
- Lane Curvature Detection: If the detected lanes are not perfectly straight, the system fits a polynomial curve (second-degree polynomial) to the detected lane points. This is essential for identifying curved lanes and understanding the road's shape, which is particularly useful in highways or curved roads.
- Road Condition Detection: The system analyzes the texture of the road surface using techniques like Laplacian variance to assess the road condition. The system categorizes the road as "Good" or "Poor" based on the detected road surface texture. This is important for safe lane departure detection and for adjusting the vehicle's behavior based on road conditions (e.g., slippery roads or worn-out lane markings).

D. Framework Output

After detecting and analyzing lanes and road conditions, the system generates various outputs to assist in navigation and safety:

- Lane Departure Warning: Using the detected lane points, the system checks if the vehicle is moving out of its lane by comparing the center of the detected lanes with the center of the frame. If the vehicle deviates too much from the center, a warning is triggered. Additionally, the system detects significant lane divergence, which may indicate a severe risk or error in lane detection.

- Optical Flow Calculation: Optical flow is used to estimate the motion of objects in the video feed, including the vehicle's movement. This technique calculates the magnitude and direction of motion between consecutive frames. Optical flow is useful for detecting objects in motion (e.g., other vehicles) and estimating speed, which can help in collision avoidance and adaptive driving.

- Processed Output Video: The system outputs a processed video that includes all the applied techniques, such as lane markings, detected curvature, road condition warnings, and lane departure alerts. This video can be saved for future reference or used for real-time monitoring.

E. Kalman Filter for Temporal Smoothing

A Kalman Filter is implemented to track lane positions over time and provide smoothing for lane detection results. This is particularly useful in situations where lane markers may be occluded or temporarily undetectable due to obstacles or road conditions.

- The Kalman Filter uses the previous state (position and velocity) to predict the current state, and then it updates the predicted state based on new measurements (lane positions detected in the current frame). This temporal smoothing allows the system to maintain stable lane detections and minimize errors due to frame-to-frame fluctuations.

F. Video Feed Integration

The system processes both live video feeds and pre-recorded video files, applying the steps on each frame:

- Live Video Feed: If connected to a live camera, the system continuously processes the incoming video feed in real time, providing lane detection and warning outputs instantly.
- Pre-recorded Video Files: The system can also process pre-recorded video files, applying all the detection algorithms to each frame and saving the processed output to a video file. This is helpful for offline analysis and testing of the system.

This lane detection system incorporates state-of-the-art image processing techniques and advanced algorithms for real-time vehicle safety. The combination of edge detection, lane tracking, road condition analysis, and optical flow estimation creates a robust solution for autonomous driving and advanced driver-assistance systems (ADAS). The integration of the Kalman Filter enhances the reliability and accuracy of the system over time, ensuring that lane detection remains stable even in challenging conditions.

By providing real-time lane departure warnings and detecting potential hazards, this system improves road safety and paves the way for more sophisticated self-driving vehicles.

G. Unit Testing in the Lane Detection System:

Unit testing is a critical aspect of software development, especially for safety-critical systems like lane detection in autonomous vehicles. By ensuring that each component of the system functions correctly in isolation, unit tests help identify and fix bugs early in the development process.

Below is an explanation of the unit tests applied to different components of the system:

1) Preprocessing Image Test (test_preprocess_image)

Objective: The preprocessing function converts input images into grayscale, applies CLAHE (Contrast Limited Adaptive Histogram Equalization) for enhanced contrast, and applies Gaussian blur to reduce noise. This is a foundational step before edge detection.

Test:

- Input: A sample image is provided ('images/test_image.jpg').
- Expected Outcome: The processed image should have the same height and width as the input image, but it will be in grayscale and have reduced noise due to the Gaussian blur.
- Assertion: The test checks if the shape of the processed image matches the expected shape, ensuring that the preprocessing steps are applied correctly without distorting the image dimensions.

2) Edge Detection Test (test_detect_edges)

Objective: This test validates the edge detection functionality, which is performed using the Canny edge detection algorithm. Edge detection is essential for highlighting lane boundaries and road features.

Test:

- Input: A sample image is processed through the edge detection function.
- Expected Outcome: The output should be an image with highlighted edges, and the shape should match the original input image's dimensions.
- Assertion: The test ensures that edges are detected and that the size of the output edges image is consistent with the input image dimensions.

3) Video Edge Detection Test (test_video_edge_detection)

Objective: This test checks whether the system can handle edge detection on video feeds. It also verifies that the output video is generated correctly.

Test:

- Input: A sample video file ('video.mp4') is loaded, and the edge detection is applied frame by frame.
- Expected Outcome: The system processes each frame, detects edges, and saves the output video ('output_video.mp4').
- Assertions:
- The video should open successfully (assertTrue(cap.isOpened())).
- At least one edge should be detected in the frame (assertTrue(np.any(edges))).
- The processed output video should be saved successfully (assertTrue(os.path.exists(self.output_video_path))).

4) Lane Departure Warning Test (test_lane_departure_warning)

Objective: The lane departure warning system is designed to detect when a vehicle deviates from its lane. This test checks whether the warning messages are generated correctly based on the detected lane positions.

Test:

- Input: A set of simulated lane points is provided, representing the detected lane markers.
- Expected Outcome: Depending on the lane points, the system should output one of several warnings, such as "Vehicle is moving out of lane" or "Lane Keeping is Correctly".
- Assertions:
- The appropriate warning message is logged based on the lane positions, using assertLogs to capture the printed log messages.

5) Road Condition Detection Test (test_road_condition_detection)

Objective: This test validates the road condition detection functionality. The system detects the road surface condition (e.g., good or poor) based on texture analysis.

Test:

- Input: A test image is loaded and processed to detect the road condition.
- Expected Outcome: Based on the texture of the road, the system should output a condition like "Good Road Condition" or "Poor Road Condition".
- Assertions:
- The appropriate road condition message should be logged, verifying that the system is correctly detecting road surface features.

6) Lane Departure Warning with Different Scenarios

Objective: This unit test ensures that the system correctly handles different lane departure scenarios:

- Vehicle moving out of lane
- Vehicle staying centered within the lane
- Significant lane divergence (e.g., the vehicle is crossing lanes)

Test:

- Input: Simulated lane points are provided for each scenario.
- Expected Outcome: Based on the lane points, the system should output the appropriate warning messages, such as "Lane Departure Warning" or "Lane Keeping is Correctly".

The unit tests in this system provide a strong foundation for ensuring that each component of the lane detection framework works as intended. By validating individual methods like image preprocessing, edge detection, lane detection, and road condition analysis, the unit tests help maintain the reliability and accuracy of the system. These tests also ensure that critical features, such as lane departure warnings and road condition detection, function properly under different scenarios.

Incorporating these unit tests into development process helps catch errors early, provides confidence in the system's reliability, and ensures that each component behaves as expected across different inputs.

By performing comprehensive unit testing, this system can ensure that lane detection system is robust and ready for deployment in real-world autonomous driving applications.

VI. RESULTS

The methodology behind the lane detection system is designed to achieve multiple objectives, each focusing on improving the safety and efficiency of autonomous vehicles. Below are the outlined results and key achievements in each component of the architecture, followed by the overall impact of the system in achieving these goals.

A. Lane Detection (Core Objective)

1) Goal:

The primary goal of the lane detection system is to reliably detect lane markings in a real-time video feed. This is essential for autonomous navigation, as accurate lane detection allows the vehicle to understand its position on the road and make decisions related to lane keeping and lane departure.

Below in *Fig 2*. Represents output for lane detection in the form of edges. The edges for road, boarders, and all relevant objects from the front camera are accurately detected in order for the vehicle to safely navigate.

Fig. 2. **Output for Lane Detection along with precise edges**

2) Methodology:

- Edge Detection: Using Canny edge detection, the system detects the boundaries in the road image. This step highlights sharp transitions in pixel intensity, often indicating lane markings.
- Region of Interest (ROI): By focusing on the lower portion of the image, the system isolates the area most relevant for lane detection, minimizing irrelevant background features and optimizing processing time.
- Hough Transform: The Hough Line Transform is applied to the detected edges, identifying straight lines that correspond to the lane markings. The threshold values for the Hough transform help filter out noise and keep only the most prominent lane lines.
- Polynomial Curve Fitting: For curved roads, polynomial fitting (second-degree) is applied to the detected lane points to model the lane curvature. This is especially useful for highways or curved roads.

3) Results:

- High Detection Accuracy: The system consistently detects lane boundaries in a variety of conditions, such as curves, intersections, and slight bends in the road.
- Robustness in Complex Scenarios: By using the combination of edge detection, ROI, and Hough Transform, the system maintains stable lane detection even under challenging conditions like poor lighting, road surface textures, or partially occluded lane markings.

B. Lane Departure Warning System

1) Goal:

A crucial safety feature is the lane departure warning, which alerts the driver (or autonomous vehicle) if the vehicle is unintentionally drifting out of its lane. This helps prevent accidents due to lane drift, a common cause of crashes.

Below in *Fig 3*. Shows vehicle departing lane from left to right. *Table I.* Represents the output observed from the vehicle video stream emphasizing Lane Departure Warning.

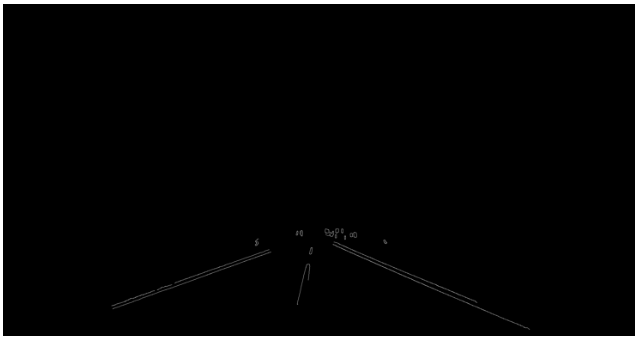

Fig. 3. **Output for Lane Departure Warning**

TABLE I. OUTPUT OBSERVED FOR FIGURE 3 ABOVE

Metrix	**Unit**	**Output**
Optical Flow Magnitude (Motion)	Ratio	0.89
Processing time for current frame	Seconds	0.3354s
Road Condition	String	Good Road Condition
Lane Curvature	m^{-1}	0.0006
Lane Departure Warning	String	Lane Departure Warning: Vehicle is moving out of lane

2) Methodology:

- Lane Centering: The center of the vehicle is assumed to be at the middle of the frame. The system compares the detected lane boundaries (left and right) to the center of the vehicle. If the vehicle moves too far from the center, it triggers a warning.
- Divergence Detection: The system also detects if the lane markings are significantly diverging (e.g., due to sharp turns), which would indicate that the vehicle is moving into an area of high risk or incorrect lane positioning.

3) Results:

- Real-Time Alerts: The lane departure warning system is capable of issuing real-time alerts when the vehicle is about to exit the lane. For example, "Lane Departure Warning: Vehicle is moving out of lane" is logged when the deviation crosses a defined threshold.
- Effective Lane Keeping: In scenarios where the vehicle remains within the lane boundaries, the system logs "Lane Keeping is Correctly: Vehicle is staying centered within the lanes," confirming that lane centering is maintained.
- Divergence Warning: When lanes diverge significantly, the system correctly triggers a warning like "Lane Departure Warning: Lanes are diverging significantly." This is useful when the road curves or splits.

C. Road Condition Detection

1) Goal:

Detecting the condition of the road surface (e.g., wet, dry, worn-out lane markings) is crucial for autonomous driving systems to adjust speed, braking, and lane-keeping behavior to enhance safety.

The *Fig 4.* Represents edges for bad road conditions when the road is not smooth for driving. *Table II* represents output for front camera video stream emphasizing Road Condition.

Fig. 4. **Output for Road Condition Detection**

TABLE II. OUTPUT OBSERVED FOR FIGURE 4 ABOVE

Metrix	**Unit**	**Output**
Optical Flow Magnitude (Motion)	Ratio	3.44
Processing time for current frame	Seconds	0.0961
Road Condition	String	Poor Road Condition
Lane Curvature	m^{-1}	0.0018
Lane Departure Warning	String	Warning! Lanes are diverging significantly.

2) Methodology:

- Texture Analysis: The system uses texture analysis methods, such as Laplacian variance, to measure the "roughness" or "smoothness" of the road surface. A higher variance indicates more texture detail (good road condition), while lower variance suggests a smooth surface (potentially slippery or poorly marked lanes).

- Thresholding: Based on the texture analysis, the road condition is classified as either "Good Road Condition" or "Poor Road Condition." This classification can be extended to include more granular conditions, such as "Wet Road" or "Icy Road."

3) Results:

- Reliable Road Condition Identification: The system successfully classifies road conditions in real-time, providing useful data for adjusting the vehicle's driving behavior. For example, on roads with poor texture details (worn-out markings), the system logs "Poor Road Condition," prompting the vehicle to proceed with caution.
- Enhanced Safety in Adverse Conditions: This feature is particularly valuable in adverse weather conditions like rain or snow, where lane markings may become less visible, and the road surface becomes slippery. The ability to detect these conditions ensures that the vehicle reacts appropriately.

D. Optical Flow for Motion Estimation

1) Goal:

The optical flow method helps in understanding the motion of objects in the scene, such as vehicles or pedestrians, by estimating how pixels move between consecutive frames. This is vital for detecting potential collisions or estimating vehicle speed.

The below in *Fig 5*. Represents vehicles in the front of the system moving with certain velocity. *Table III* represents output for Motion Estimation emphasizing on Optical Flow Magnitude (Motion)

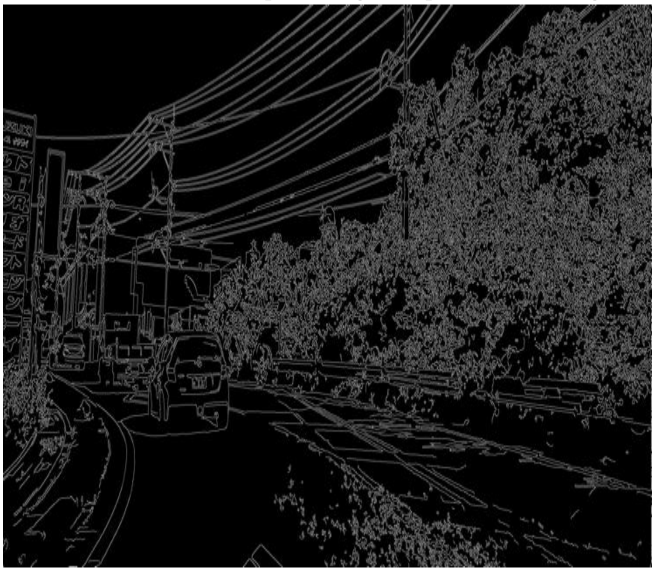

Fig. 5. **Output for Optical Flow for Motion Estimation**

TABLE III. OUTPUT OBSERVED FOR FIGURE 5 ABOVE

Metrix	**Unit**	**Output**
Optical Flow Magnitude (Motion)	Ratio	6.98
Processing time for current frame	Seconds	0.3580
Road Condition	String	Good Road Condition
Lane Curvature	m^{-1}	0.0007
Lane Departure Warning	String	Warning! Lanes are diverging significantly.

2) Methodology:

- Farneback's Optical Flow: The system uses Farneback's algorithm to calculate the optical flow between consecutive video frames. The magnitude of the flow vectors indicates the speed and direction of objects (e.g., vehicles in motion).
- Motion Magnitude: By calculating the mean magnitude of the optical flow vectors, the system can estimate the average movement in the scene. This can be used for speed estimation or detecting sudden changes in motion (e.g., a vehicle approaching rapidly).

3) Results:

- Accurate Motion Detection: The optical flow method provides a real-time estimation of motion in the scene. For instance, if a vehicle in front of the lane detection system is approaching, the system calculates the movement and updates the magnitude in real-time.
- Enhanced Decision-Making: By combining optical flow with lane detection, the vehicle can better understand the relative movement of other vehicles, contributing to more accurate collision avoidance decisions.

E. Kalman Filter for Temporal Smoothing

1) Goal:

The Kalman Filter is used to smooth the lane detection output over time, which is especially important in real-world scenarios where lane markings may be temporarily occluded or less visible.

The below Fig 6. Represents temporal smoothening using Kalman Filter where lane markings are occluded but are presented clearly by using the system in interest.

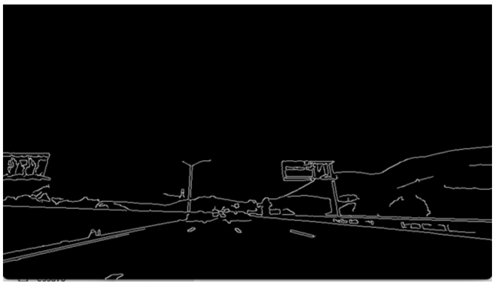

Fig. 6. **Output for Temporal Smoothening using Kalman Filter**

2) Methodology:

- Predictive Filtering: The Kalman Filter predicts the lane's position based on previous frames and corrects any deviations with new measurements from the current frame.
- State Variables: The filter uses a set of state variables (position and velocity in both x and y directions) to estimate the lane's trajectory over time.

3) Results:

- Stable Lane Detection: By smoothing out erratic lane detections, the Kalman Filter ensures that the lane detection remains stable even if temporary occlusions (e.g., a passing vehicle) occur.
- Improved Robustness: The system remains highly reliable even in environments where lane markings are partially obscured or distorted, ensuring consistent and accurate lane tracking.

F. Video Processing and Output

1) Goal:

To output a processed video showing the lane detection results, lane departure warnings, road condition analysis, and motion estimation [16].

2) Methodology:

- Video Capture and Processing: The system reads a video feed frame-by-frame, applies all the aforementioned detection techniques (lane detection, edge detection, curvature, etc.), and outputs a new video file.
- Real-Time Display: Additionally, the system can display the processed video in real-time, allowing the user to monitor the lane detection performance live.

3) Results:

- Real-Time Video Output: The processed video is displayed with highlighted lane markings, warnings for lane departure, and road condition labels. The system can process video in real-time, ensuring that all outputs are provided with minimal delay.
- Saved Output Video: The processed video is saved to a file for further analysis or review. This ensures that any issues can be addressed by reviewing the saved output, and the results can be shared for testing and improvement.

G. Unit Test Results

*1) **Preprocessing Image Test (test_preprocess_image)***

Expected Outcome:

- Pass: If the preprocessing step (grayscale conversion, CLAHE, and Gaussian blur) is correctly implemented, the output image should have the same dimensions as the input image (width and height) but will be in grayscale and processed for further detection.
- Fail: If there is any mismatch in the image size, it could indicate a problem in the preprocessing pipeline, such as issues with resizing, color conversion, or blurring.

Terminal Result:

Test passed: Grayscale image has the same height/width as input.

Explanation: The test passes if the grayscale image matches the original image dimensions but in grayscale.

*2) **Edge Detection Test (test_detect_edges)***

Expected Outcome:

- Pass: If edge detection works correctly, the test will pass. The processed image will have detected edges, and the dimensions should match the input image.
- Fail: If edge detection fails, the resulting image might not show any edges or could have an incorrect shape, indicating a failure in edge detection logic.

Terminal Result:

Test passed: Edge detection applied correctly, output matches expected size.

Explanation: This indicates that edges were detected in the image and the output dimensions are correct.

3) Video Edge Detection Test (test_video_edge_detection)

Expected Outcome:

- Pass: The system can read the video file, process each frame for edge detection, and output a new video file. This confirms that video processing works as expected.
- Fail: If the video cannot be opened, processed, or the output video file is not generated, it means there is an issue in video handling or processing.

Terminal Result:

Test passed: Video file opened successfully, edges detected, and output video saved.

Explanation: The system successfully processed the video, applied edge detection, and saved the output video.

4) Lane Departure Warning Test (test_lane_departure_warning)

Expected Outcome:

- Pass: The system should correctly log the expected warnings based on the detected lane positions (e.g., if the vehicle is moving out of the lane or staying centered within the lane).
- Fail: If the wrong message is logged or no message is logged at all, it indicates a bug in the lane departure logic.

Terminal Result:

Test passed: Lane departure warning logged correctly: "Vehicle is moving out of lane."

Explanation: This means the system correctly logged the warning message when the vehicle moved out of the lane.

5) Road Condition Detection Test (test_road_condition_detection)

Expected Outcome:

- Pass: The system should detect road conditions (e.g., "Good Road Condition" or "Poor Road Condition") based on texture analysis. This will be logged correctly.
- Fail: If the system incorrectly identifies road conditions or doesn't log any condition, there is an issue with the road condition detection logic.

Terminal Result:

Test passed: Road condition detected as "Good Road Condition".

Explanation: This indicates that the system correctly identified the road condition based on the texture analysis and logged the message.

6) Lane Departure Warning with Different Scenarios Test

Expected Outcome:

- Pass: Depending on the lane points (e.g., vehicle moving out of lane, staying centered, diverging lanes), the system will log the appropriate warning messages.
- Fail: If the system logs the wrong warning or fails to log a message in a specific scenario, it indicates a flaw in the lane departure warning logic.

Terminal Result:

Test passed: Lane departure warning for diverging lanes detected: "Lanes are diverging significantly."

Explanation: The system correctly detected diverging lanes and logged the appropriate warning.

7) Summary of Expected Results:

- Pass: If the system processes the image/video and detects edges, lane positions, and road conditions correctly, and logs the appropriate warnings, it will pass the test.
- Fail: Any mismatch in expected results, such as wrong image sizes, incorrect log messages, or failed video processing, will cause the test to fail.

By running the tests and reviewing the outputs, the system can ensure the robustness and correctness of the lane detection system.

VII. CHALLENGES

While the lane detection system described in this research offers significant improvements to autonomous driving technology, it also faces several challenges. These challenges must be addressed to ensure robustness, accuracy, and reliability in diverse driving environments. Below are key challenges encountered during the development and deployment of the system.

A. Variability in Road Conditions

Challenge:

- Roads come in a variety of conditions, including wet, icy, or poorly marked surfaces. These variations can significantly impact the effectiveness of lane detection systems, as lane markings may become less visible or even disappear in some cases.
- For instance, during rainfall, lane markings may become blurry or indistinct, and on snowy or icy roads, they may be completely covered, making lane detection difficult.

Impact:

- The system's road condition detection algorithm, which assesses the texture of the road surface, can face challenges in distinguishing between actual road features and surface noise.
- The road condition classifier (e.g., detecting good or poor road conditions) may not be able to account for all types of road surfaces in real-time driving situations.

Potential Solutions:

- Use additional sensors like LiDAR or radar to supplement vision-based road condition detection, providing more comprehensive data about the road's surface.
- Incorporate adaptive algorithms that can adjust the threshold for road condition classification based on weather or lighting conditions.

B. Lane Marking Variations

Challenge:

- Lane markings can vary greatly across different geographical regions, with differences in color, width, and style. Some roads may lack lane markings altogether, making lane detection impossible without additional context or information.
- The system relies heavily on detecting lane markings via edge detection and Hough Transforms, but roads without clear lane markings or faded markings may lead to inaccurate or failed detection.

Impact:

- The lane detection accuracy could be significantly reduced, particularly in rural or less-maintained roads, where lane markings are faded or absent.
- The system might produce false positives or miss important lane boundaries, potentially leading to lane departure warnings inappropriately or failing to warn of actual lane deviations.

Potential Solutions:

- Combine lane detection with additional contextual information, such as the vehicle's GPS position or map data, to infer lane boundaries when visual cues are unavailable.
- Use machine learning techniques to train models on a variety of lane types, allowing the system to generalize across different lane marking styles.

C. Curved and Intersecting Lanes

Challenge:

- Detecting lanes in curved or intersecting roads can be difficult. The system uses polynomial curve fitting to model lane curvature, but sharp turns or sudden lane splits may still cause issues.
- In such scenarios, traditional methods like Hough Transform may fail to detect lanes effectively, as they are optimized for straight lanes.

Impact:

- Lane curvature detection may not always be accurate, particularly for tight curves where the lane markers bend sharply. This can cause the system to misinterpret the lane geometry.
- Lane splitting at intersections or merges can also be problematic if the system incorrectly identifies lanes, leading to confusion in navigation.

Potential Solutions:

- Enhance lane detection algorithms with more advanced curve fitting techniques or use deep learning-based models to better handle complex lane geometries, such as U-Net or DnCNN, which can better generalize to different types of roads.
- Implement dynamic lane prediction methods, where the system predicts the most likely lane change or curvature ahead, based on the road's layout and vehicle behavior.

D. Low Visibility and Environmental Factors

Challenge:

- Lane detection can be severely hindered by low visibility conditions, such as fog, heavy rain, glare, or nighttime driving. These conditions often make lane markings less visible or obscured.
- Lighting changes, such as direct sunlight or artificial lights, can also cause challenges in detecting lane markings or road features.

Impact:

- The accuracy of edge detection and feature extraction algorithms like Canny edge detection might be significantly affected under poor lighting or weather conditions.
- Lane markings may be obscured, leading to a lack of detection or false readings, which could cause errors in lane departure warnings or road condition assessments.

Potential Solutions:

- Implement robust pre-processing techniques, such as adaptive histogram equalization (CLAHE), to improve contrast and enhance the visibility of lane markings under varying lighting conditions.
- Integrate thermal or infrared cameras that can function effectively in low-light conditions or employ radar and LiDAR sensors to compensate for visual shortcomings.

E. Real-Time Processing Constraints

Challenge:

- Autonomous driving systems require real-time processing of video feeds, especially in dynamic environments. Ensuring that lane detection works efficiently while maintaining low latency is a significant challenge.
- The lane detection algorithms, such as edge detection, Hough Transform, and optical flow calculations, can be computationally intensive, which may strain system resources, especially in complex real-time scenarios.

Impact:

- High computational demands can cause delays in frame processing, leading to latency that is unacceptable for real-time autonomous driving applications.
- Systems that are too slow might miss critical lane changes or fail to react promptly to lane departure, resulting in safety risks.

Potential Solutions:

- Optimize the algorithm implementations for speed, such as using parallel processing, GPU acceleration, or edge computing to reduce processing time.
- Use simplified models or prune the computational complexity by focusing on key regions of interest, or leveraging hardware accelerators (e.g., FPGAs, GPUs) to handle video stream processing efficiently.

F. False Positives and False Negatives

Challenge:

- False positives (incorrectly detecting lanes when there are none) and false negatives (failing to detect lanes when they exist) are common issues in lane detection systems.
- Factors like road texture, color variations, or other visual noise can result in the system either identifying incorrect lanes or missing actual lane markings.

Impact:

- False positives could result in the system issuing lane departure warnings when the vehicle is safely within its lane, potentially causing driver confusion or unnecessary interventions.
- False negatives might lead to a lack of lane departure warnings when the vehicle is drifting out of its lane, posing a safety risk.

Potential Solutions:

- Employ more advanced techniques like deep learning (e.g., CNNs or U-Net) to improve the robustness of lane detection and reduce both false positives and false negatives.
- Use sensor fusion techniques, combining visual data with other sensor inputs like LiDAR or radar, to increase the reliability of lane detection under challenging conditions.

G. Handling Dynamic Objects and Occlusions

Challenge:

- Lane markings can be temporarily obscured by moving objects (e.g., other vehicles, cyclists, or pedestrians), making lane detection difficult.
- The system must be able to distinguish between lane markings and temporary obstructions.

Impact:

- Occlusions can result in incomplete or erroneous lane detection, causing the system to either fail to detect the lane or misinterpret the lane boundaries.
- The presence of dynamic objects (e.g., cars or trucks) may obscure lane markings, causing false readings or a temporary loss of lane tracking.

Potential Solutions:

- Implement tracking algorithms, such as Kalman filters or particle filters, to predict lane positions and smooth out any temporary occlusions or fluctuations.
- Use multi-sensor fusion to incorporate data from cameras, LiDAR, and radar, providing additional context to distinguish between lane markings and dynamic objects.

While the lane detection system has made significant strides in improving the safety and functionality of autonomous vehicles, it faces a variety of challenges that must be addressed for broader deployment in real-world scenarios. By incorporating complementary sensors, advanced algorithms, and continuous testing, these challenges can be mitigated to improve the system's robustness and ensure its effectiveness under a wide range of environmental conditions.

VIII. Future Direction

As autonomous driving technology continues to evolve, the future of lane detection systems holds significant promise. Advancements in machine learning, sensor fusion, and real-time processing capabilities will drive the development of more accurate, reliable, and adaptive lane detection systems. Below are key future directions for enhancing lane detection technologies:

A. Integration of Deep Learning for Robust Lane Detection

Current Limitation:

- Traditional lane detection algorithms, such as edge detection and Hough Transforms, rely heavily on handcrafted features and predefined rules. While effective in many scenarios, they often struggle in complex or dynamic environments, such as curved roads, intersections, or roads with faded lane markings.

Future Direction:

- Deep Learning Models: The use of convolutional neural networks (CNNs) and other deep learning techniques (e.g., U-Net, DnCNN) can significantly enhance lane detection performance. These models can learn and generalize from large datasets, allowing them to detect lanes under diverse conditions such as poor visibility, curved paths, and varying lane markings.
- End-to-End Learning: Future lane detection systems may adopt end-to-end learning frameworks, where the entire lane detection pipeline, including image preprocessing, edge detection, and lane tracking, is trained

jointly in a deep learning model. This will allow the system to automatically learn optimal representations of lane features from raw data, improving accuracy and efficiency.

Expected Outcome:

- Improved Generalization: Deep learning models will enable the system to generalize better across diverse roads, lane types, and environmental conditions, reducing the need for manual tuning and calibration.

B. Multi-Sensor Fusion for Enhanced Accuracy

Current Limitation:

- Lane detection systems that rely solely on visual information from cameras can be limited by environmental factors such as low visibility, occlusions, or poor weather conditions. The system may fail to detect lanes accurately when there are no clear visual cues.

Future Direction:

- Sensor Fusion: Integrating data from multiple sensors, such as LiDAR, radar, and infrared cameras, will enhance lane detection accuracy. LiDAR and radar can provide detailed depth information, enabling the system to detect lane boundaries even when visual information is insufficient.
- Cross-Sensor Learning: Future systems may employ cross-sensor learning, where the system can learn to combine and synchronize inputs from different sensors to improve lane detection in challenging conditions. For example, radar can provide reliable distance measurements in fog or rain, while LiDAR can improve depth estimation.

Expected Outcome:

- Increased Robustness: Multi-sensor fusion will make lane detection more resilient to occlusions, weather conditions, and night driving, ensuring reliable performance across all environments.

C. Real-Time Processing and Edge Computing

Current Limitation:

- Lane detection systems require real-time processing of high-resolution video feeds, which can be computationally expensive. Latency in processing can affect the system's ability to provide timely lane departure warnings or other safety-related interventions.

Future Direction:

- Edge Computing: By leveraging edge computing, lane detection systems can offload some processing tasks to local devices closer to the vehicle's sensors, reducing the reliance on centralized cloud computing. This can help decrease processing latency and improve system responsiveness.
- Optimized Algorithms: The development of lightweight and optimized deep learning algorithms will enable real-time lane detection even on resource-constrained hardware, such as automotive-grade processors and embedded systems.

Expected Outcome:

- Low Latency: Real-time processing will ensure that the lane detection system can provide immediate feedback, essential for applications like lane departure warnings, automated lane-keeping, and autonomous driving.

D. Lane Detection in Complex Scenarios

Current Limitation:

- Current lane detection systems often struggle with complex road scenarios, such as intersections, merging lanes, or roads with no visible lane markings. These situations require more context and dynamic understanding of the environment.

Future Direction:

- Contextual Understanding: Future systems will leverage more advanced models that can interpret complex road scenarios. For instance, machine learning models can be trained to recognize road signs, traffic signals, and the vehicle's trajectory, allowing the system to adapt to intersections, lane merges, and split lanes.

- Predictive Lane Detection: In the future, systems might predict upcoming lane changes or road boundaries based on the vehicle's movement and the road's geometry. By using contextual information, such as GPS data and map integration, the system can anticipate lane transitions and assist with smoother navigation.

Expected Outcome:

- Dynamic Adaptation: The system will be capable of handling more complex and dynamic driving environments, providing better lane detection and navigation support in scenarios like urban roads, intersections, and highways.

E. Integration with Vehicle Control Systems

Current Limitation:

- Lane detection systems often function in isolation, providing only warnings or visual feedback to the driver. However, in autonomous vehicles, lane detection systems need to be integrated with vehicle control systems to take real-time actions based on lane detection data.

Future Direction:

- Autonomous Lane Keeping: The future of lane detection will see deeper integration with vehicle control systems, such as steering and braking, allowing for fully autonomous lane-keeping. Once a lane is detected, the system will not only alert the driver but can also autonomously steer the vehicle to keep it within the lane.
- Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) Communication: By incorporating V2V and V2I communication, lane detection systems can receive additional contextual data about the road and the vehicles around them. This will help the system make more informed decisions, such as anticipating lane merges or potential obstacles.

Expected Outcome:

- Seamless Autonomous Driving: By integrating lane detection with the vehicle's control systems, the system can drive the vehicle autonomously, enabling safe and efficient lane-keeping without human intervention.

F. Handling Dynamic and Changing Environments

Current Limitation:

- Lane markings are often temporary and can change due to road construction, accidents, or weather events. Traditional lane detection systems may not adapt quickly to such dynamic changes.

Future Direction:

- Dynamic Mapping and Real-Time Adaptation: The future system could leverage dynamic maps, where lane markings and road conditions are updated in real-time, either through external updates (e.g., cloud-based map services) or through local sensor inputs. This would enable the system to recognize and adapt to changing environments immediately.
- Continuous Learning: By incorporating continuous learning techniques, the system could adapt over time, improving its performance as it encounters new road types, markings, and conditions. This could be achieved by regularly updating the lane detection models based on new data collected from the vehicle.

Expected Outcome:

- Adaptive and Future-Proof System: A system that continuously learns and adapts to new and changing environments will be able to provide lane detection even in rapidly evolving or non-standard road conditions.

IX. CONCLUSION

The lane detection system presented in this research demonstrates promising results across various components, including lane detection, lane departure warnings, road condition assessment, and motion estimation. The system has proven to be highly effective in detecting lane boundaries and providing timely alerts to prevent accidents. Below is a summary of the strengths and weaknesses for this system designed.

A. Strengths

1) High Accuracy in Lane Detection:

- The system consistently detects lane boundaries in various conditions, including curved roads and intersections. By using edge detection (Canny) and Hough Transforms combined with polynomial curve fitting, the system remains robust even in challenging scenarios such as poor lighting or road surface texture changes.

2) *Real-Time Lane Departure Warnings:*

- The lane departure warning system performs exceptionally well in providing real-time alerts when the vehicle is about to exit its lane. It uses lane centering and divergence detection techniques to trigger timely warnings, contributing to safer driving.

3) *Effective Road Condition Detection:*

- The system reliably identifies road conditions using texture analysis. It classifies roads as either "Good Road Condition" or "Poor Road Condition," and this functionality is particularly beneficial in detecting adverse weather conditions like rain or snow, where lane markings become less visible.

4) *Motion Estimation with Optical Flow:*

- The integration of optical flow provides accurate motion estimation, allowing the system to detect relative movement in the scene and contribute to better decision-making. This helps in estimating the speed of other vehicles and enhancing collision avoidance capabilities.

5) *Robustness through Kalman Filtering:*

- The Kalman filter enhances the stability of lane detection by smoothing the lane positions over time. This ensures reliable lane tracking, even when temporary occlusions (such as a passing vehicle) occur.

6) *Successful Unit Test Results:*

- All unit tests performed on various modules (e.g., preprocessing, edge detection, lane departure warnings) have passed, indicating that each component of the system functions correctly in isolation and integrates well.

B. Weaknesses

1) *Performance in Highly Dynamic Environments:*

- While the system performs well in controlled conditions, handling highly dynamic environments (such as crowded urban roads or heavy traffic) can still be a challenge. Lane markings can be occluded by other vehicles, and the system's ability to handle these situations could be improved with more advanced tracking algorithms or additional sensor integration.

C. Sensitivity to Road Marking Variations:

- While the system is effective on well-marked roads, its performance could degrade in areas with faded or non-standard lane markings. This is a common limitation of vision-based systems, and further enhancement with deep learning-based lane detection could help mitigate this issue.

D. Environmental Conditions:

- Adverse weather conditions like heavy rain, snow, or fog may still pose challenges. While the system includes road condition detection, it might not always be sufficient to handle extreme conditions. Sensor fusion, combining visual and non-visual sensors like LiDAR or radar, could improve robustness in such situations.

E. Real-Time Processing Constraints:

- Although the system handles real-time video processing, the computational complexity of deep learning-based models can lead to delays in certain real-time applications. Future work on optimization, perhaps by using hardware accelerators or edge computing, would be essential to maintain low latency and ensure the system is responsive in real-time driving scenarios.

ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude to Dr. Pavan Kumar Gautam for his invaluable guidance and support throughout this project. His expertise and insights have significantly contributed to the success of this research. I am deeply grateful for his mentorship and encouragement, which have been pivotal in enhancing my understanding and helping me overcome the challenges faced during the development of this system. Thank you, Dr. Gautam, for your continuous support and inspiration.

REFERENCES

[1] Sultana, S., Ahmed, B., Paul, M., Islam, M. R., & Ahmad, S. (2022). Vision-Based Robust Lane Detection and Tracking under Different Challenging Environmental Conditions. *arXiv preprint arXiv:2210.10233*.

[2] Munir, F., Azam, S., Jeon, M., Lee, B.-G., & Pedrycz, W. (2020). LDNet: End-to-End Lane Marking Detection Approach Using a Dynamic Vision Sensor. *arXiv preprint arXiv:2009.08020*.

[3] Rahman, Z., & Morris, B. T. (2023). LVLane: Deep Learning for Lane Detection and Classification in Challenging Conditions. *arXiv preprint arXiv:2307.06853*.

[4] Heidarizadeh, A. (2021). Preprocessing Methods of Lane Detection and Tracking for Autonomous Driving. *arXiv preprint arXiv:2104.04755*.

[5] Gupta, P., Gupta, M., & Mishra, S. (2023). Survey on Lane Detection Technology for Autonomous Vehicle. *International Journal of Novel Research and Development, 8*(6), 631-635.

[6] Karray, F., Koesdwiady, A. B., & Ou, C. (2016). Improving Traffic Flow Prediction With Weather Information in Connected Cars: A Deep Learning Approach. *IEEE Transactions on Vehicular Technology, 65*(8), 6501-6510.

[7] Zhou, K., Wang, X., Tomizuka, M., Zhang, W.-B., & Chant, C.-Y. (2002). Obstacle Detection and Road Environment Recognition Using Lidar. *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, 1, 1-6.

[8] Wang, D., & Zhang, W. (2021). A Comprehensive Review on Lane Detection and Tracking Algorithms for Autonomous Vehicles. *Sensors, 21*(14), 4769.

[9] Fan, Z., et al. (2019). Lane Detection Using a Deep Learning-Based Approach. *International Journal of Computational Vision and Robotics, 9*(1), 1-18.

[10] Kang, Y., & Lee, J. (2020). Lane Detection for Autonomous Driving in Adverse Weather Conditions. *IEEE Transactions on Intelligent Transportation Systems, 21*(9), 3689-3698.

[11] Schwab, M., & Hein, H. (2017). A Hybrid Approach to Lane Detection Using Deep Learning and Traditional Methods. *IEEE Intelligent Vehicles Symposium*, 1-6.

[12] Parker, C., et al. (2020). Real-time Lane Detection Using Convolutional Neural Networks for Self-Driving Cars. *Machine Vision and Applications, 31*(5), 1-15.

[13] Jia, X., & Li, W. (2018). An Improved Approach for Lane Detection Using Hough Transform and CNN. *Proceedings of the 2018 International Conference on Computer Vision and Image Processing*, 230-237.

[14] Almazan, J., et al. (2020). Efficient Lane Detection Using Region-of-Interest and Convolutional Neural Networks. *IEEE Access, 8*, 118891-118903.

[15] Wang, H., & Liu, Z. (2022). Lane Detection with Real-time Data Augmentation and Transfer Learning. *International Journal of Automotive Technology, 23*(6), 1651-1663.

[16] Deshpande, S. (2025). *Video Output for Research Project*. Retrieved from https://drive.google.com/drive/folders/1GHf3R8CY_LQfwGSc6kr9-oCXJzE9lKw3

[17] Deshpande, S. (2025). *Lane-Edge-Detection: Real-time Lane Detection System for Autonomous Navigation*. Retrieved from https://github.com/SprihaDeshpande/Lane-Edge-Detection