

# Review Paper Real Time Road Lane Detection

Nupur Hadke<sup>1</sup>, Vaishnavi Kshirsagar<sup>2</sup>, Yashashri Sawarkar<sup>3</sup>, kshay Raut<sup>4</sup>,  
Gaurav Somnath<sup>5</sup>, Atharva Wekhande<sup>6</sup>

<sup>1,2,3,4,5,6</sup>Dept of computer Engineering, Bapurao Deshmukh College of Engineering,  
Sevagram, wardha  
Guided By: Prof.H.S.Belsare

**Abstract:** *In this research, a novel navigation algorithm for self-driving vehicles that avoids collisions with pedestrians and ad hoc obstacles is described. The proposed algorithm predicts the locations of ad hoc obstacles and wandering pedestrians by using an RGB-D depth sensor. Unique ad hoc-obstacle-aware mobility rules are presented considering those environmental uncertainties. A Deep Reinforcement Learning (DRL) algorithm is proposed as a decision-making technique (to steer the self-driving vehicle to reach the target without incident). The deep Q-network (DQN), double deep Q-network (DDQN), and dueling double deep Q-network (D3DQN) algorithms were compared, and the D3DQN had the fewest negative rewards. We tested the algorithms using the Carla simulation environment to examine the input values from the RGB-D and RGB-Lidar. The series of algorithms that make up the convoluted neural network D3DQN was consequently selected as the optimum DRL algorithm. In the modeling of slow-moving urban traffic, RGB-D and RGB-Lidar generated essentially the same results. A self-driving version of an updated child-ride-on-car was modified to demonstrate the real-time effectiveness of the proposed algorithm.*

**Index Term:** *Self-Driving Vehicle, Deep Reinforcement Learning, Ad Hoc-Obstacles, Obstacle Avoidance, Lane Detection, Object Detection.*

## 1. INTRODUCTION

Ad hoc, or temporary, obstacles such as road barricades, potholes, speed bumps, and roaming pedestrians are likely to provide a challenge for self-driving vehicles in India and similar countries. Alternatively, self-driving vehicles can communicate with a roadside unit or another moving vehicle using VANET (Vehicular Ad Hoc Network) to collect upto-date information about accidents, road obstacles, traffic congestion, and weather conditions. This type of information is also important for allowing the self-driving vehicles to travel safely, and thereby preventing road accidents. Researchers have mounted sensors on vehicles to identify ad hoc obstacles, and some researchers have used those sensors to guide self-driving vehicles. For example, some researchers developed a smartphone-based ad hoc-obstacle detection algorithm. At the same time, the information about the identified obstacles (kind, location) was recorded on a central server and used to alert other drivers traveling on the same road.

## 2. RELATED WORK

To accurately detect potholes, Salcedo et al. and Bučko et al. adopted a computer vision and deep learning based algorithm. They designed their algorithms to detect potholes even in low-light conditions and at night. Similar to pothole detection, speed breakers are a nightmare for human-driven and self-driving vehicles. Some speed bumps are clearly marked, while others are not, and running over them at speed can cause serious damage to vehicles. Arun priyan et al. designed a deep learning-based image segmentation algorithm to detect speed breakers, so using only relevant images would allow the steering and braking algorithms to work more quickly. Varma et al. developed a deep learning-based speed bump detection system and collected the depth value of the speed breakers using the stereovision ZED depth camera. The depth values were significant in reducing the speed of the vehicle. The Advanced Driver Assistance System (ADAS) and self-driving vehicles can greatly benefit from this speed breaker detection and depth information system. In the case of obstacle avoidance navigation systems, Wang et al. and Haris and Hou developed successful obstacle avoidance-based navigation algorithms for self driving vehicles. The authors of and each developed a self-driving vehicle that automatically maneuvered around obstacles such as boxes and cones that were placed randomly on the road. Manikandan et al developed a curve-aware navigation algorithm that extract curvature information on the path and then successfully navigates a self-driving vehicle over curvy roads

## 3. LITERATURE REVIEW

Mr. Mustafa Surti [1], used Hough transform for lane detection. Marked lane lines are detected by applying Canny edge detection on a custom Region of Interest. They demonstrated the lane detection system for lane departure warning on an embedded development board like RaspberryPi. Using Gaussian Based road model [2] lane could be detected using a multi stage algorithm. The processes involved in these stages are histogram generation, fitting and normalisation. Different ordered polynomial functions are used for lane fitting. In the approach used in [3], curvature is detected using image processing. The captured image is converted into binary space and segmented into three sections. The curvature direction is determined on the basis of the section having the most number of non-zero pixels. Approach I follows similar logic, but for calculating curve values average values [8] of the histogram of the lower part of the binary image are used instead of performing image segmentation. The approach [4] presents an efficient version of sliding window algorithm for lane curve fitting. One of the advantages of this improved version is that it allows sliding window to be applied even on uneven lane markings. In the second approach, this method is applied onto the road edges from the binary image. The coordinates of the road edges are then processed to obtain the curve value and the vehicle offset. Few of the popular algorithms includes GOLD system developed by Broggi which uses an edge-based lane boundary detection algorithm [9]. Remapping of acquired image is done to represent in bird's eye view of the road. Extraction of quasi vertical bright lines that concatenated into specific larger segments is done using specific adaptive filtering. The RALPH system [10], used to control the lateral position of the vehicle. Lane's curvature and lateral offsets are determined by a template to the averaged scan line intensity profile which is adaptively adjusted and aligned using a matching technique. The same university developed another system called AURORA which tracks the lane markers present on structured road using a color camera mounted on the side of a car pointed downwards

toward the road [11]. A. Benefits of Lane Detection ^ Lane Detection System can provide assistance and certain details to the drivers and as well as pedestrians. ^ Such systems can allow drivers to drive safely and also help them to stay on their particular lane by providing some kind of indicators. ^ According to research presented by WHO, [6] about 1.35 million human 5 deaths were caused due to road traffic injuries. Autonomous vehicles could bring this number down by some extent. □ According to McKinsey’s report in 2015, the mass adoption of autonomous vehicles could possibly reduce road traffic accidents by 90 ^ Since software will drive the car, [7] the modern vehicle can now be programmed to reduce emissions up to 60 according to Ohio University

#### 4. METHODOLOGY

- Convolution Neural Networks (DCNN) and Recurrent Neural Network (RNN), we present a hybrid deep neural network for road lane detection (RNN). The proposed model comprises of a DCNN from a wider perspective that incorporates different sequential images as a feedback and predicts the lane path in a svm classification manner in the current frame.
- In order to attain this segmentation objective, a fully deep convolution (DCNN) approach is used. It consists of a network of encoders and also a network of decoders, assuring that the final feature map is exactly the same size as the source images. From a local point of view, the encoder network’s summarised features of a Deep CNN are further explicated by a RNN
- We plan to build the network in the form of an encoder network -decoder network model in an effort to integrate CNN and RNN as an complete, end-to-end training network. The network architecture that is proposed .

#### 5. FLOW CHART

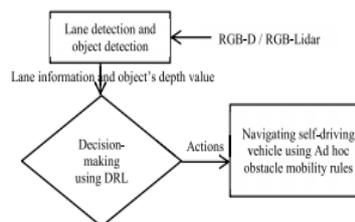


FIGURE 1. Proposed algorithm block diagram.

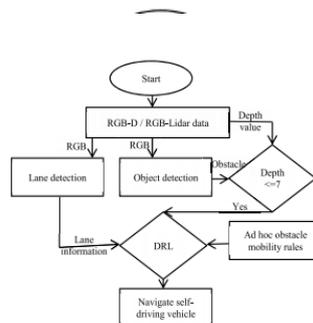


FIGURE 2. Flow chart of the proposed algorithm.

## 6. PROPOSED METHOD

Resizing: The image may be resized to a manageable size. In this paper, based on the previous preprocessing, we firstly extract the colour features based on the white colour and then extract the edge features based on the straight lane. Because the high-speed section is the traffic accident-prone section, the high-speed road section mostly is the straight line lane [18]. Therefore, in order to obtain a very high recognition rate, we successively carry on colour detection and edge detection to the lane. This paper combines colour features extraction and edge features extraction, and the experiment proves that the recognition rate and accuracy of lane detection are greatly improved. Our main contribution in this paper is to do a lot of work in the preprocessing stage. We proposed to perform colour transform of HSV in the preprocessing stage, then extract white, and then perform conventional preprocessing operations in sequence. Moreover, we selected an improved method proposed in the area of interest (ROI). In this paper, based on the proposed preprocessing method (after HSV colour transform, white feature extraction, and basic preprocessing), one-half part of the processed image is selected as the area of interest (ROI). In addition, we performed twice edge detection. The first is in the preprocessing stage, and the second is in the lane detection stage after the ROI is selected. The purpose of performing twice edge detection is to enhance the lane recognition rate.

1. Image Acquisition: - A camera mounted on a vehicle captures images or video frames of the road scene.
2. Preprocessing: - The acquired image is preprocessed to enhance lane visibility. This typically includes the following steps: - e resolution to reduce processing time. - Grayscale Conversion: Convert the image to grayscale to simplify processing and reduce the amount of data. - Noise Reduction: Apply filters, such as Gaussian blur, to reduce noise and smooth the image. - Contrast Enhancement: Adjust the contrast and brightness to highlight lane markings.
3. Edge Detection: - Edge detection algorithms, such as the Canny edge detector, are applied to identify potential lane markings. These algorithms highlight areas of high gradient in intensity, which often correspond to lane edges.
4. Region of Interest (ROI) Selection: - Define a region of interest (ROI) in the image. The ROI is typically a trapezoidal shape representing the road area ahead of the vehicle. This step helps narrow down the area for further lane detection.
5. Hough Transform: - Apply the Hough transform to detect lines in the edge-detected image. The Hough transform can identify lines based on their polar coordinates (slope and intercept).
6. Lane Fitting: - Detected lines from the Hough transform need to be grouped and fitted to represent the left and right lane lines. Here's how this is typically done: - Line Clustering: Group the detected lines by their slopes and proximity. - RANSAC (Random Sample Consensus): Use RANSAC to robustly estimate the parameters of the lanes (slope and intercept). RANSAC is effective in the presence of outliers and noise. - Lane Validation: Validate the identified lanes based on expected lane characteristics, such as width and curvature.
7. Lane Tracking: - For video streams, a tracking algorithm may be used to smooth and predict the position of the lanes over time. This helps to account for minor variations in lane markings.
8. Visualization: - Finally, the detected lane lines are overlaid on the original image, providing a visual representation of the detected lanes for the vehicle's driver or autonomous control system.

## 7. CONCLUSION AND FUTURE WORK

The developed algorithm successfully detected straight and slightly curved road lanes on motorways. The algorithm showed robustness under changing lighting conditions i.e. shadows and slightly worn lanes. The algorithm was successfully implemented on the SoC DSP

TMS320C6678 multicore processor. Also, it was optimised on four levels i.e. algorithm level, single core level, memory and multi-core level. These optimisations allow the complete system to achieve 81 fps speed and 94.1 resolution of 1242 Ö 375. The achieved high frame rate makes it easy to implement the other complicated systems in ADAS where multiple algorithms are executed, since it does not require a lot of computation time. Also, it leaves space for more computationally demanding algorithms like stereo vision. The OpenMP API used to parallelise the serial code made it flexible and processor independent, which gave a further increase of 76.7core. for future work

1. Improved Algorithm: Develop a more robust and accurate lane detection algorithm. This could involve exploring deep learning techniques like convolutional neural networks (CNNs) or recurrent neural networks (RNNs) to enhance performance.
2. Real-time Processing: Optimize the algorithm for real-time processing, as this is crucial for applications like self-driving cars. This could involve hardware acceleration or parallel computing techniques.
3. Adverse Conditions: Enhance the system's ability to detect lanes in adverse conditions, such as bad weather, low lighting, or complex road scenarios.
4. Lane Tracking: Implement lane tracking to predict lane positions in future frames, providing smoother and more reliable lane information.
5. Semantic Segmentation: Incorporate semantic segmentation to distinguish between different types of lanes, road markings, and objects on the road.
6. Data Augmentation: Expand the dataset used for training to include a wider range of road conditions and scenarios to improve generalization.
7. Human-Machine Interaction: Explore ways to incorporate driver-vehicle interaction, where the system can provide feedback or warnings to the driver based on lane detection.
8. Hardware Integration: Work on integrating the lane detection system with existing hardware platforms, such as onboard cameras and processors.
9. Regulatory Compliance: Ensure that the system complies with relevant safety and regulatory standards.
10. Open-source Contribution: Contribute to open-source projects related to lane detection, helping the community improve this technology.

## 8. REFERENCES

- [1] Mr.Mustafa Surti,Prof.(Dr.) Bharati Chourasia, "Real time lane detection system using Python and OpenCV on Raspberry Pi ",International Journal for Research in Engineering Application Management (IJREAM), Vol-05, issue-06, 2021.
- [2] Boon-Chin Yeo, Way-Soong Lim Heng-Siong Lim, 15 "Lane Detection In The Absence Of Lane Markings For Roadway Surveillance With Thermal Vision", ICIC International, Vol-12, Number-03, 2019.
- [3] Anuja Ihare, "Image Processing to Detect Curvature of Road", Article on medium.com, 2020.
- [4] Landon Bentley, Joe MacInnes, Rahul Bhadani Tamal Bose, "A PseudoDerivative Method for Sliding Window Path Mapping in Robotics-Based Image Processing", 2019.
- [5] GurjyotKaur and Gagandeeop Singh, "A Review Of Lane Detection Techniques",International Research Journal of Engineering and Technology (IRJET) , Vol-02, Issue-03, 2015.
- [6] World Health Organization, "Road traffic injuries", Article on who.int, 2020.
- [7] Ohio University, "The Future of Driving", Article on ohio.edu, 2021.
- [8] M. Hassan, "Self-Driving Car using Raspberry Pi", Article on computervision.zone, 2020. [9]Salna Joy,Mamta B.S,Tejas b.Mukesh,Mohammed Muzammil Ahmed,Uday Kiran,"Real time road lane detection using computer vision techniques in python",2022.
- [9] Pomerleau D. and Jochem,"Rapidly Adapting Machine Vision for Automated Vehicle Steering", IEEE,2020.

- [10] M. Chen., T. Jochem D. T. Pomerleau, “AURORA: A Vision-Based Roadway Departure Warning System”, In Proceeding of IEEE Conference on Intelligent Robots and Systems, 2004.
- [11] Sertap Kamc,1, Dogukan Aksu, Muhammed Ali Aydin,”Lane Detection For Prototype Autonomous Vehicle”, AIRCC Pulishing Corporation, Vol 09, Number 13, 2019.
- [12] Open Source Computer Vision,”Smoothing Images”, Article on docs.opencv.org, 2021.
- [13] Adrian Rosebrock,”4 Point OpenCV getPerspectiveTransform Example”, Article on pyimagesearch.com, 2014.
- [14] Aryan,”Implement Canny Ede Detector in Python using OpenCV”, Article on geeksforgeeks.org, 2020.