ROI based real time straight lane line detection using Canny Edge Detector and masked bitwise operator.

Manan Doshi
Department of Information Technology
Dwarkadas J. Sanghvi College of Engineering
Mumbai, Maharashtra
manandoshi1301@gmail.com

Harsh Shah
Department of Information Technology
Dwarkadas J. Sanghvi College of Engineering
Mumbai, Maharashtra
harshshah7074@gmail.com

Neha Katre
Department of Information Technology
Dwarkadas J. Sanghvi College of Engineering
Mumbai, Maharashtra
neha.mendjoge@djsce.ac.in

***Abstract*—Research for autonomous cars has now been close to a decade and still it is not possible to employ these cars everywhere around the world, for one major reason being clear lane line detection. However, there is constant discovery to improve the method of lane detection, especially in real-time. For lane detection, various computer-vision techniques and deep learning models have been devised, but for practical use it is necessary to find an efficient solution in real-time. Our technique is based on the real-time efficient detection of straight lanes using a canny edge detector followed by finding a region of interest and Hough transformation. This method takes video as an input and gives outputs in the form of images with slopes and marked lines of lanes. For long highways with straight lanes, this algorithm can prove to be extremely efficient for detection, which can be easily employed in real-time using camera sensors that provide a video feed. Furthermore, there is no requirement for training the algorithm. Hence, this system works on most of the scenarios without any prior data training.**

***Index Terms*—Lane line detection, Canny Edge Detector, Region of Interest, Hough Transformation**

I. INTRODUCTION

According to WHO, road traffic injuries cause an estimated 1.35 million deaths worldwide each year, which equates to one fatality every 25 seconds on average [1]. Human error accounts for 98% of these accidents [2]. The young population of India, ages 18 to 45, accounts for 70% of all road accidents [3]. According to the 2016 Road Accidents Survey in India, the country reported at least 4,80,652 accidents resulting in 1,50,785 deaths, implying that at least 17 people died in road accidents every hour in 55 accidents [4]. One of the primary causes is abrupt lane changes, particularly on highways. A sudden lane change is one of the main reasons for traffic injuries, with distraction and driver weariness being the primary causes.

Driverless or autonomous cars are being researched and developed in order to eliminate or, at the very least, minimize the proportion of these incidents. The parallel growth of technology and the advent of high-end cars provide a doorway to autonomous vehicles [5]. They are autonomous in the sense that they do not require human involvement and can perform functions such as sensing the surrounding environment, avoiding collisions with other vehicles, recognizing street signs, automatic emergency braking, and LIDAR (Light Detection and Ranging), to name a few [2][5]. They are capable of becoming acquainted with their surroundings, making judgments, and operating independently without the assistance of humans [5]. The primary goal of such self-driving cars is to provide comfort and safety while also solving many problems such as traffic congestion, which will result in shorter commutes, parking, and reducing automobile accident injuries and fatalities [6]. However, driverless vehicles are still a relatively new phenomenon and are not completely immune to accidents. They are constantly being researched, developed, updated and improved and therefore different lane detection systems are required to assist minimize accidents. As a result, various ADAS (Advanced Driver Assistance Systems) have been developed specifically for this reason.

Lane detection is a key component of ADAS. Lane detection algorithms are primarily used to detect when a vehicle is about to leave the lane and notify the driverless system that it may collide with another vehicle, as well as for lane maintaining, vehicle overtaking, and automated cruise driving [6]. Lane detection algorithms are critical because without them, the safety of passengers and the automobile itself is jeopardized. These algorithms, however, must be robust, meaning they must work in various lighting settings and backgrounds and accurately recognize the lane markers. However, due to the intricacies of the real world, lane recognition remains a difficult task.

There are two different types of lane detecting approaches: sensor-based approaches and vision-based approaches. Sensor-based techniques, as the name implies, rely on external devices such as LIDAR sensors, which are essential components of autonomous vehicles, that provide a high-resolution 3D representation of their surroundings. LIDAR gives autonomous cars the ability to "see" by producing and measuring millions of data points in real time and building a detailed picture of their constantly changing environment for safe navigation. It also makes use of additional technologies such as laser sensors, radar, and GPS (Global Positioning System) [1]. Vision-based approaches, on the other hand, are further categorized as model-based approaches and feature-based approaches. Using the geometrical coordinates of the camera and the road as inputs, model-based techniques construct a mathematical model of the road to explain the lane structure [1][7]. While model-based approaches are less vulnerable to noise than feature-based methods, they are extremely difficult to implement since they demand extremely intensive calculations and previous knowledge of specific geometric parameters [7]. Feature-based techniques use road images as input and distinguish between lane markers and areas outside of the lane by recognizing characteristic road elements such as colour, gradient, and boundaries [1][7].

II. Objective

In this paper, we propose a feature-based lane detection system employing Canny Edge Detector and Hough Transformation to develop a lane detection system that will be a vital component of ADAS to help maintain lanes and minimize accidents due to sudden lane changes. The proposed algorithm has been designed to operate in real time, allowing it to be employed in autonomous vehicles for its intended purpose. The current approach is quite light-weight in terms of processing, allowing AI to use it in ADAS for long straight highways and use the computation power that is saved for other duties.

III. Related Work

The authors of paper [8] first used a Gabor filter for edge, which can extract edge information in multi-direction and multi-scale. Following that, the Hough transform is employed to find lanes by narrowing the traversal range for faster computations. Because the lane is usually white or yellow, the gray value changes significantly and the lane edge can be recognised using standard lane detection operators such as Sobel, Robert, Canny, and so on. The edge detection operator is chosen based on the lighting condition. Finally, in real-time conditions, the position of the previous image is used to identify the position of the next frame, and DROI (Dynamic Region of Interest) is defined to improve speed and robustness, and the lane location is derived using least square fitting in DROI.

The authors of paper [1] used an IPM (Inverse Perspective Mapping) on the ROI (Region of Interest) to make the lane lines that appeared to converge at the horizon vertical and parallel. The LSD (Line Segment Detection) technique is used for lane detection, and the line segment to be recognized is represented as a rectangle with pixels of varying orientations. To avoid the problem of no detection while utilising a single frame, multiple frames are employed, and lines detected in all of those frames are gathered and concatenated into a single frame.

The authors proposed a method in paper [7] that consists of three major steps: initialization, lane marking detection, and lane marking tracking. Instead of the traditional Canny algorithm, the EDLines approach is utilized to extract the edges from the grayscale image in this case. The rectangular ROI (Region of Interest) is then calculated by calculating IPs (Intersection Points) of numerous pairs of line segments from the image's LHR (Left Half Region) and RHR (Right Half Region), with the majority of these IPs found near the actual VP (Vanishing Point), and then these tentative VPs are tracked for a certain number of frames until the number of VPs exceed the threshold value. Following that, the line segments are extracted using the EDLines test. The HAC (Hierarchical Agglomerative Clustering) algorithm is then used to combine all of the line segments of a single lane marking into a single segment. The scan-line test is used to get a lane marking candidate for a specific number of frames and then clusters the candidates in comparable positions, which is known as inter-frame clustering.

To address the drawbacks of the Canny algorithm, the authors of paper [9] recommend using the Otsu-Canny algorithm. The Otsu algorithm is used first to determine the threshold for the Canny operator, and then Canny edge detection is utilized to retrieve the lane edge, from which the ROI (Region of Interest) is extracted. This approach is far more resilient than traditional edge detection algorithms, yielding a significantly clearer and sharper image.

The authors of paper [10] use an improved region growing method for unstructured curved-lane detection by using a block as the initial growth block instead of a single pixel, and if the pixel variance of the block is greater than the threshold value, it is considered a background block and the region growing process is stopped. This allows for more efficient and thorough extraction of the lane area.

The authors of paper [11] created a lane departure system that uses Euclidean Distance to alert the driver of changes in the lane ahead, and if the driver does not react to the changes, it issues a danger signal. When compared to comparable systems, the false warning rate is also low.

In order to eliminate the noise issues encountered by the traditional RANSAC algorithm, which led to false detection, the authors of paper [12] devised a system using Fuzzy Control to determine the difference between the actual lane boundary points and the points detected as lane edges due to interference noise and then to remove these noise points before using the RANSAC algorithm for lane boundary detection. The accuracy gained in effectively detecting lane boundaries was extremely high as a result of the Fuzzy system, and they were also successful in detecting inner lanes if they were present.

The authors of paper [13] used a Raspberry Pi to implement and evaluate the method for real-time lane detection that they devised using the Canny Edge Detector and the Hough Transform in real-world circumstances. Their experiment was successful since they were able to detect lanes on a moving car using a Raspberry Pi and a Pi camera.

IV. METHODOLOGY

We use video as our input for real-time detection. However, it is not possible to process an entire video. Thus, for processing, an image snapshot of the video obtained at a millisecond interval can be used. Detection in photos with the fewest computations possible can be an excellent technique to support the development of real-time lane detection systems. As a result, a single video is divided into a number of frames. Now, using our proposed method, each frame is processed in steps until an outcome is produced. There are five stages in our process. Figure 1 depicts our proposed approach in general.

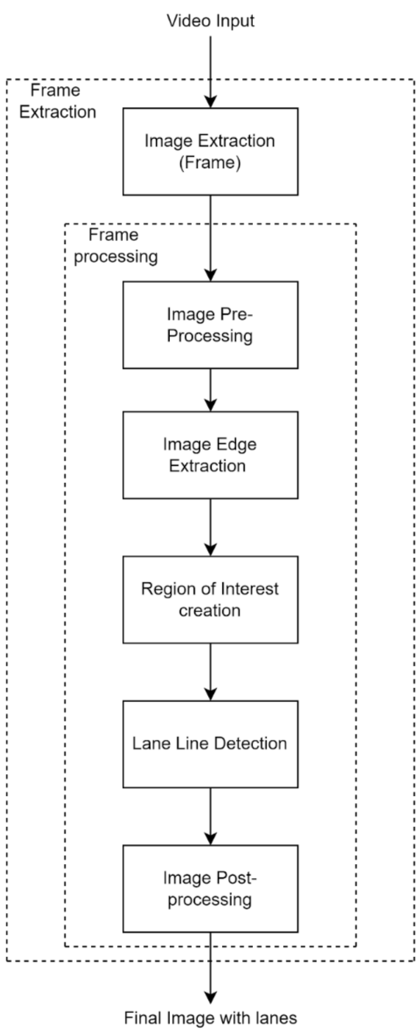


Fig. 1: Proposed algorithm for straight lane line detection

A. Image Pre-Processing

The first stage in most computer vision challenges is image pre-processing. To decrease the complexity of the image and make computations quicker, a coloured image is transformed to a grayscale image for lane detection. A coloured image employs the RGB model, which contains a large amount of information in the form of an 8-bit array for each colour (red, green, and blue) while gray-scale images, remove all of the superfluous information and provide us with an image that can be processed fewer computations. This can be done using three popular methods [9] mentioned in the table:

1) Average method: As the name suggests, the average of all three primary colours is taken:

$$Gray(x, y) = (R + G + B)/3$$

2) Weighted Average method: Green has the maximum sensitivity in human eyes, while blue has the lowest. As a result of the comparison:

$$Gray(x, y) = 0.29 * R + 0.578 * G + 0.225 * B$$

3) Maximum method: The strongest values of the three basic colours are compared here, and the value is assigned a gray value.[14]

$$Gray(x, y) = max(R, G, B)$$

The second phase in image pre-processing is image smoothening or noise reduction. Images contain a variety of signals and frequencies when captured with camera sensors under varied settings. These are variations among the group of pixels, which are nothing more than a high standard deviation in the brightness level of pixels in the image, which makes lane line marking detection difficult. Noise is reduced to remove this extra variability and to deal with the photos with less processing. As a result, we employ the Gaussian Blur technique to eliminate high passing noises, resulting in a smooth image [15].Given that a picture is represented in two dimensions (by the x and y axes, respectively), convolution is formed by combining two functions and two axes, respectively. The randomness among the pixels is reduced by using this third function, which creates a normal distribution.

B. Image Edge Extraction

Edge detection and extraction is one of the most important and widely utilised ideas in computer vision algorithms for a variety of applications. There are several edge detection algorithms of first and second-order derivatives like Sobel, Prewitt's, Roberts, Laplacian, etc. But Canny's Edge Detection Algorithm, is the most efficient and thus, commonly utilized. This algorithm is broken into five steps, which are as follows:

1) Noise Reduction: Noise was previously eliminated using Gaussian blur, thus, this step ensures that if any noise is present, it is subject to removal because Gradient Calculation is a noise-sensitive operation.
2) Gradient Calculation: Here edge intensity magnitude and direction are calculated using edge detection operators by calculating the gradient of the image.
3) Non-maximum Suppression: To achieve crisp edges, we must chop out our thick edges and provide accurate thin

edges. As a result, the edges are reduced here to make it sharp and clear.

4) Double Threshold: Following edge reduction, binary decisions are performed on the edges to brighten and make them visible for three categories of edge pixel identification: strong, weak, and non-relevant. The latter is unimportant, however in weak pixels, if it exceeds the threshold set, it is considered strong and is doubly brightened for clear edge recognition.
5) Edge tracking by hysteresis: This is the stage at which weak pixels are classified as strong or weak based on the threshold [16].

C. Region of Interest

We can recognize several lines in every image. Each line, however, cannot correspond to our own lane. As a result, we eliminate these lines by defining a zone of interest. The region of interest relates to the area of the image on which we wish to focus for detection rather than the entire image. It is also futile to process every line on the image in real time. As a result, it decreases the amount of processing required by deleting unneeded edges and shrinking our search space.

D. Lane-Line Detection

We can easily obtain the lines on the image by utilising the Canny edge detector and our region of interest. These are the lanes for which we need more information to feed our system. Using the Hough Transform approach, we can now obtain the slopes of the lane lines. The Hough approach is used to compute a global description of a feature given local data. In this line detection method, each input measurement, or coordinate point, explains how it contributes to a globally consistent result. In other words, Hough Transform converts the x-y plane to the m-c plane for the equation y = m * x + c, where m is the slope and c is the y-intercept, and then searches for lines that pass through some points using the point of intersection of all the lines in the m-c plane. Using this method, we can quickly detect collinear points in an image and obtain the line's coordinates. After lane detection, these coordinates are the most crucial in passing information about the lane to the next AI module of the autonomous cars. Hough transform can be implemented as:

$$\rho = x * \cos(\theta) + y * \sin(\theta)$$

Where θ = angle between normal line and x-axis which ranges from 0 to 90 degrees, ρ = Distance between x-axis and fitted line, (x, y) = co-ordinate value of the pixel [11].

E. Image Post-Processing

Once we have the coordinates, we must clip out all of the lines on the image, keep the image in scenic form for evaluation, and apply our line coordinates to make it stand out on the image. This is a three-step procedure in which we first retain the image, then arrange the lines, and then draw on the retained image for further processing or presentation.

V. Dataset

We have used three main datasets—CULane [17], Caltech lanes [18], and a test footage of lanes [19]—to understand the model's capacity for identifying lane lines. With varied daylight periods, about 5000 photographs from CULane and 700 images from Caltech were utilized to create the final model, which was then deployed on the video.

VI. Working

Each stage has its own relevance based on the supplied approach, and the entire algorithm is designed for real-time application, which means it can function on any pre-recorded video or real-life video segment. The algorithm in our proposed solution uses pre-recorded video from our dataset and outputs it as if each frame is a collected snapshot of the road at a specific time interval. The entire algorithm is now executed on each snapshot or frame.

The frame is first fed into the system for pre-processing, which reduces the image's complexity by transforming the RGB image to a grayscale image. This is accomplished by employing the weighted-average method, which is the most appropriate and efficient way for conversion. Because of the reduced computing demand, the resultant image, which has been reduced from 24-bit (RGB format) to an 8-bit (grayscale) image, is ideal for low-level operations in real-time applications. The grayscale image is smoothed by removing noise for the next level of pre-processing. This aids in the identification of the lane line, which appears in white on the grayscale image. Noise degrades the image and increases inaccuracy, which can lead to additional processing and an ineffective output. A captured image is typically made up of high and low-frequency components. We reduced some extremely high passing frequencies in these since our edge detector model is vulnerable to these weak edges. Our edge detector model is so sharp that it can detect high passing frequencies as edges if the noise in the image is not eliminated, which might lead to incorrect interpretation. As a result, we employ the Gaussian Blur approach, which is widely used to remove high-frequency components in the Fourier domain while allowing low frequencies to pass through the filter. Low pass filters (blurring techniques) include the Ideal Low Pass Filter and the Butterworth Low Pass Filter. The key advantage of Gaussian over these other filters, particularly Butterworth's, is that there is no ringing effect with any order of filter we choose to work with. Ringing effect is nothing but the rippling artefact around sharp edges caused by the loss or distortion of high-frequency information in the image, which can again misguide the edge detection process.

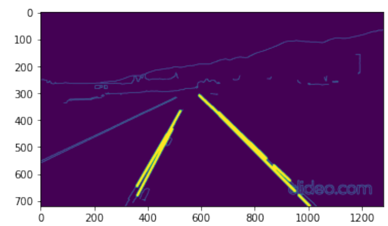
Following pre-processing, we arrive at edge detection, which is one of the most significant modules. We have numerous edge detection techniques, however Canny's Edge Detection algorithm is currently the best. Because the method is sensitive to weak edge frequencies, it consists of a sequence of phases that include blurring. However, because we've already used Gaussian Blur, this step ensures that any high frequency is removed if present. Furthermore, using gradient calculation, we can obtain all of the edge magnitudes and directions in the image. Canny Algorithm by itself identifies so many minor edges, that forces the algorithm to eliminate the weak edges in the remaining part of the algorithm. The algorithm then trims the thick margins of the discovered edges, formalising all of the image lines. This is followed by two stages in which it is chosen whether or not the edge should be eliminated based on the label they receive, i.e. weak or strong. Weak edges are identified and eliminated based on the threshold, whilst strong edges are brightened. This entire process yields the image's edges and constitutes Canny's Edge Detection Algorithm.

Because there are lane edges and other ambient objects, the Edge Detection algorithm produces edges throughout the entire image, which is ineffective for direct lane prediction. We create a region of interest for the system to identify lanes. The term "Region of Interest" refers to the acquisition of an area of interest on a picture. We hunt for specific coordinates across the image that can potentially identify lanes since we wish to identify lanes. We know that the lanes converge in such a way that the eventual lane structure looks like a triangle from the perspective of a front camera on the car responsible for road sensing. Using this context, we plot three points that potentially form a triangle on an image with no background, which is an array with all zero pixels except the three points which can be seen in Figure 2. On the image, these three points

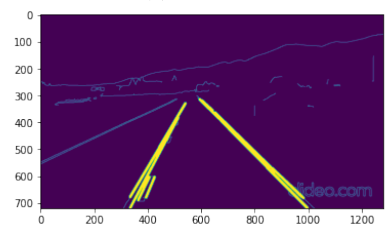
Fig. 2: Traingle mask

are then filled with white pixels to make the triangle. We will now utilize this image with only a triangle and mask it on our image with edges using a Bitwise AND. This eventually removes all background edges except the lane edges. Now we have the lane lines on the image, but it is just on the image and we don't have any further information about them. As a result, we attempt to gather information regarding the lane line. We want to get the slope of the intended lanes on both sides so that it may be supplied into the autonomous driving system for further processing. We apply the Hough Transformation to obtain the lane line slopes for this purpose. It operates on the m-c plane using the collinearity principle. The x-y plane gives rise to the m-c plane, and any collinear line on the x-y plane forms the point of intersection of many lines on the m-c plane that satisfy the equation of lines on the x-y plane. Thus, using the Hough Transform, we can calculate the slope of lines on an image using a parametric equation using sin and cos. Using the raw image, we can thus calculate the coordinates of the lane lines in each frame in real time. Figure 3 displays the results of applying the Hough Transformation to an image. These line details can now be used in any format, such as feeding the coordinates to the system or plotting the lane lines on an image for verification. The first four frames of the output are shown in Figure 4 after the lane line detection technique has been applied to an input video.

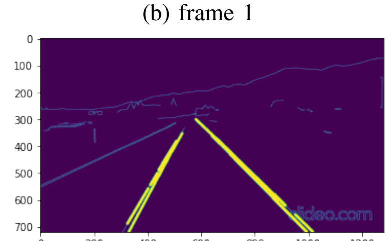
Fig. 3: Lane plotting based on Hough Transform co-ordinates



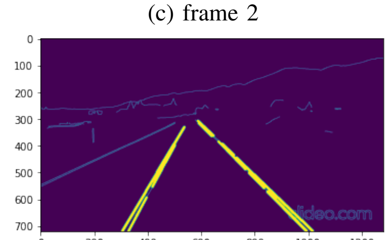
(a) frame 0



(b) frame 1



(c) frame 2



(d) frame 3

Fig. 4: Output frames in sequence

VII. Result

The approach of outcome identification uses observation rather than precision and recall. Actual output is lane co-ordinates which are being plotted and shown in the image. As a result, we can see that lane lines were precisely plotted on the pictures. The main goal is to obtain lane lines co-ordinates with the least amount of computing feasible, and in this case, we deal with the most fundamental bitwise operator to do so.

VIII. Conclusion

Lane detecting systems have grown increasingly vital as autonomous vehicles become more ubiquitous and commonplace in daily life. They serve a vital role in accident prevention. To avoid this, we presented a real-time straight-line lane recognition technique. To begin, we utilized the Gaussian Blur technique to eliminate high-frequency noise so that high-frequency components were not misinterpreted as edges, as edge detectors are susceptible to this type of noise. Another advantage is that it also helps to eliminate the rippling effect. Following the processing stage, the Canny edge detector is used to identify edges by distinguishing between strong and weak edges and removing weak ones. After the edges have been identified, a Region of Interest must be created to distinguish between the lane edges and all other edges by forming a triangle as the lane lines appear to converge at the horizon and then using this image containing the triangle formed by the lane lines and mask it on the image obtained after applying the Canny Edge Detector by applying a Bitwise AND operator. As a result, we obtain the image only with the lane edges. The Hough Transform is now employed for real-time lane detection to retrieve the slopes on both sides of the lane. After obtaining the slopes, the lane coordinates can be determined, and the lane detection process is complete; this information can then be sent to the autonomous system for real-time lane detection. We provided video inputs with varied illumination conditions to our system in order to test and evaluate the robustness of our system, and the experiment results indicate that the system has an excellent performance because it can reliably recognize the lane markings in all scenarios.

IX. Future Work

We designed the system for straight lane detection without utilizing any deep learning approaches; thus, our model is lightweight, which means it uses limited computational power and requires very little computational time, allowing us to detect straight lane boundaries in real time. As a result, future versions of this technique can be modified to be suitable for curve lane detection without the need for deep learning approaches, increasing its utility in real-world applications.

References

[1] A. Mahmoud et al., "Real-Time Lane Detection-Based Line Segment Detection," 2018 New Generation of CAS (NGCAS), 2018, pp. 57-61, doi: 10.1109/NGCAS.2018.8572124.

[2] S. Swetha and P. Sivakumar, "SSLA Based Traffic Sign and Lane Detection for Autonomous cars," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021, pp. 766-771, doi: 10.1109/ICAIS50930.2021.9396046.

[3] J. Patil, M. Prabhu, D. Walavalkar and V. B. Lobo, "Road Accident Analysis using Machine Learning," 2020 IEEE Pune Section International Conference (PuneCon), 2020, pp. 108-112, doi: 10.1109/PuneCon50868.2020.9362403.

[4] "Road accidents in India, 2016: 17 deaths on roads every hour, Chennai and Delhi most dangerous", The Indian Express, 2017. [Online]. Available: https://indianexpress.com/article/india/road-accidents-in-india-2016-17-deaths-on-roads-every-hour-chennai-and-delhi-most-dangerous-4837832/. [Accessed: 18- Sep- 2022].

[5] R. Hussain and S. Zeadally, "Autonomous Cars: Research Results, Issues, and Future Challenges," in IEEE Communications Surveys Tutorials, vol. 21, no. 2, pp. 1275-1313, Secondquarter 2019, doi: 10.1109/COMST.2018.2869360.

[6] X. Wei, Z. Zhang, Z. Chai and W. Feng, "Research on Lane Detection and Tracking Algorithm Based on Improved Hough Transform," 2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE), 2018, pp. 275-279, doi: 10.1109/IRCE.2018.8492932.

[7] C. Lee and J. -H. Moon, "Robust Lane Detection and Tracking for Real-Time Applications," in IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 12, pp. 4043-4048, Dec. 2018, doi: 10.1109/TITS.2018.2791572.

[8] Q. Gao, Y. Feng and L. Wang, "A real-time lane detection and tracking algorithm," 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2017, pp. 1230-1234, doi: 10.1109/ITNEC.2017.8284972.

[9] Y. Wang, L. Shi, J. Lausanne and D. Zhong, "Straight lane line detection based on the Otsu-Canny algorithm," 2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC), 2022, pp. 27-30, doi: 10.1109/ITOEC53115.2022.9734320.

[10] T. Huang, Z. Wang, X. Dai, D. Huang and H. Su, "Unstructured Lane Identification Based on Hough Transform and Improved Region Growing," 2019 Chinese Control Conference (CCC), 2019, pp. 7612-7617, doi: 10.23919/ChiCC.2019.8865870.

[11] P. N. Bhujbal and S. P. Narote, "Lane departure warning system based on Hough transform and Euclidean distance," 2015 Third International Conference on Image Information Processing (ICIIP), 2015, pp. 370-373, doi: 10.1109/ICIIP.2015.7414798.

[12] Y. Xu, X. Shan, B. Y. Chen, C. Chi, Z. F. Lu and Y. Q. Wang, "A lane detection method combined fuzzy control with RANSAC algorithm," 2017 7th International Conference on Power Electronics Systems and Applications - Smart Mobility, Power Transfer Security (PESA), 2017, pp. 1-6, doi: 10.1109/PESA.2017.8277759.

[13] Roshan Jahan, Preetam Suman, Deepak Kumar Singh, "Lane Detection using Canny Canny Edge Detection and Hough Transform on Raspberry Pi," International Journal of Advanced Research in Computer Science, vol. 9, issue 2, 2018, pp. 85-89, doi: https://doi.org/10.26483/ijarcs.v9i0.6143

[14] G. Yu and D. Qiu, "Research on Lane Detection Method of Intelligent Vehicle in Multi-road Condition," 2021 China Automation Congress (CAC), 2021, pp. 2779-2783, doi: 10.1109/CAC53003.2021.9728269.

[15] R. Muthalagu, A. Bolimera, and V. Kalaichelvi, "Lane detection technique based on perspective transformation and histogram analysis for self-driving cars," Computers amp; Electrical Engineering, vol. 85, p. 106653, 2020.

[16] X. Yan and Y. Li, "A method of lane edge detection based on Canny algorithm," 2017 Chinese Automation Congress (CAC), 2017, pp. 2120-2124, doi: 10.1109/CAC.2017.8243122.

[17] X. Tang, 'Spatial As Deep: Spatial CNN for Traffic Scene Understanding', AAAI Conference on Artificial Intelligence (AAAI), 2018.r.

[18] Mohamed Aly, Real time Detection of Lane Markers in Urban Streets, IEEE Intelligent Vehicles Symposium, Eindhoven, The Netherlands, June 2008.

[19] rslim087a, "rslim087a/road-image," GitHub, Aug. 26, 2018. https://github.com/rslim087a/road-image (accessed Nov. 20, 2022).