Grenze International Journal of Engineering and Technology, June Issue

Real-Time Road Lane Detection for ADAS Vehicles

S.Towseef Ahmed[1], P Sai Charan[2], G Sunil Kumar[3] and G Sai Harshith[4]
[1-4]G.Pullaiah College of Engineering and Technology/Electronics and Communication Engineering, Kurnool, India
Email: towseef453@gmail.com, {patangecharan, gundamsunilkumar24, saiharshith765}@gmail.com

***Abstract*—One of the main motives that autonomous cars have been emerging recently is driver safety. Numerous accidents occur as a consequence of drivers' recklessness, including driving in the incorrect lane or attempting a hasty lane change. Autonomous vehicles are hitting the market in an attempt to resolve this problem. The passenger's safety is the primary concern in this scenario, so using lane detection as a safe driving alternative makes sense. Python programming and OpenCV can be used to build the Lane detection gadget. The technique involves a front camera that can be fitted on a vehicle. Lanes may simply be recognized by using the camera's vision and a few simple procedures. The transformation known as Hough can be used to extract the lanes by passing through two hyperbolas. Both colored and uncovered roadways can be equipped with the suggested approach. Any car can use this technique to drive safely in Roadways.**

***Index Terms*—ADAS, Open-cv, Lane detection, ROI (Region of Interest), Canny Edge, Hough Transformation, Autonomous Driving, Image Processing, LOG (Laplacian of Gaussian), Edge Detection**.

I. INTRODUCTION

With the swift development of society in the developing worldwide today, vehicles have become an essential mode of transportation for people to travel. In the intervening time, there are more and more vehicles of various types. Every year, the number of people who suffer from vehicle injuries rises in tandem with the number of cars on the road. The most revolutionary advancement in artificial intelligence is the autonomous driving car. They are always pushing beyond the boundaries of our culture and creating fresh possibilities in the field of transportation. A self-sustaining vehicle performs all of the functions of an autonomous human driver and can travel everywhere a conventional vehicle can. However, it's imperative that you teach it well. Lane detection is one of the various steps that are concerned at some point in the instruction of an autonomous vehicle; the first step is to do lane identification using clips of video. Autonomous vehicles benefit greatly from lane-detecting technologies since they restrict the vehicle's journey to designated lanes and prevent it from entering the incorrect one. This is incredibly helpful in preventing large-scale mishaps. This approach can be used with less effort and shows to be effective in locating lanes.[1][2][3]

A. Lane Detection for ADAS Vehicle

In the fields of computer vision and autonomous driving, lane detection is an essential element that improves road safety and vehicle navigation systems. Accurate lane recognition and monitoring on roads has become critical as cars get more and more automated and equipped with advanced driver assistance systems (ADAS). This

Grenze ID: 01.GIJET.10.2.31

technology detects lane borders and gives the vehicle's control system real-time feedback by combining advanced image processing techniques, machine learning algorithms, and sensor data. Vehicles that are capable of detecting lanes are able to stay on approved routes, maneuver through challenging traffic situations, and generally improve the safety and effectiveness of contemporary transportation systems. In light of this, creating strong lane detecting systems is an essential first step toward safer and more dependable autonomous driving.[4][5]

Fig 1: Lane Detection

B. ADAS

The term "Advanced Driver Assistance Systems" (ADAS) refers to a group of cutting-edge functions intended to improve overall vehicle performance and driver safety. These systems use a mix of cameras, sensors, and clever algorithms to deliver alerts and help in real time. Adaptive cruise control, which automatically modifies the speed to maintain a safe following distance, and lane-keeping assistance, which aids drivers in staying in their lanes, are examples of common ADAS features. Furthermore, sensors are used by collision avoidance systems to identify possible dangers and take action to avoid or lessen crashes. Other essential elements of an integrated suite of technologies that together make driving safer, more convenient, and more effective are parking assistance, blind-spot monitoring, and traffic sign recognition. The development of ADAS is a significant step toward the creation of autonomous vehicles, offering hope for a time when driving will be less stressful and more fun in addition to being safer.[6]

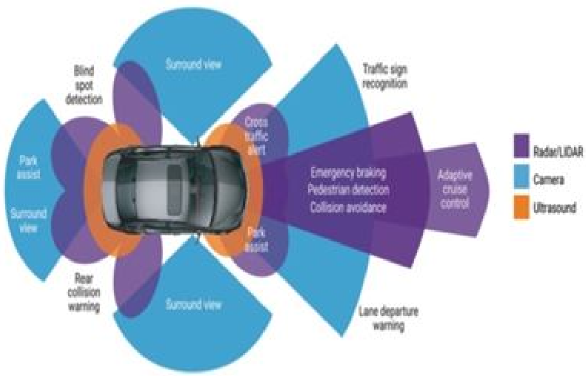


Fig 2: ADAS Vehicle

II. LITERATURE REVIEW

Grayscale Conversion: The video frames are in RGB format, however because processing a single channel image is quicker than processing a three-channel colored image, RGB is transformed to grayscale.
Reducing Noise: Before continuing, it is essential to do image smoothening because noise can produce misleading edges. This technique is carried out using a Gaussian filter. [7]
Canny Edge Detection: It calculates the gradient of our blurred image in all directions and tracks the edges with significant intensity variations. [8]
Region of Interest: Only the area covered by the road lane will be considered in this stage. Here, a mask is made that has the same dimensions as our image of the road. Additionally, between each pixel of our clever image and

this mask, a bitwise AND operation [9][10]is carried out. Finally, it hides the clever image and reveals the area of interest that was outlined by the polygonal contour of the mask.

Hough Line Transform: Straight lines can be found using the Hough Line Transform. Here, the probabilistic Hough Line Transform is utilized, and the output is represented by the identified lines' extremes.[11][12]

III. METHODOLOGY

A. Grayscale Image

A grayscale image is a single-channel image that represents the intensity of light in an image. It is created by converting a colored image (RGB) into a single-channel image where each pixel has a value between 0 and 255, with 0 representing black and 255 representing white. In the context of real-time road line detection for ADAS vehicles, grayscale images are used as an intermediate step in the image processing pipeline. The reason for converting colored images to grayscale is that it simplifies the image processing task by reducing the number of channels from three (RGB) to one. This reduction in complexity helps in improving the computational efficiency of the lane detection algorithm. Moreover, grayscale images are less sensitive to changes in lighting conditions compared to colored images. This is an important consideration for real-time road line detection as the lighting conditions can change rapidly, especially during night-time driving. By converting the image to grayscale, the lane detection algorithm becomes more robust to changes in lighting conditions. In summary, grayscale images are a crucial step in the real-time road line detection pipeline for ADAS vehicles. They simplify the image processing task, improve computational efficiency, and increase robustness to changes in lighting conditions.

B. Region of Interest

The region of selection refers to the area of the image where we want to detect the road lines. This is typically the lower half of the image, as that is where the road is located. By selecting a specific region of the image, we can reduce the amount of noise and other objects that might interfere with the lane detection algorithm. In the example code you provided, the region of selection is defined using a polygon with three vertices, which encloses the lower half of the image. The region of selection in image processing for real-time road line detection in ADAS vehicles is used to focus on the area of the image where the road lines are located. By selecting a specific region, the algorithm can reduce the amount of noise and other objects that might interfere with the lane detection process. This is typically the lower half of the image, where the road is located. By defining a region of selection, the algorithm can improve the accuracy and efficiency of the lane detection process.

C. Thresholding

Thresholding is a technique used to separate the pixels of an image into different categories based on their intensity values. In the context of road line detection, thresholding is used to separate the pixels of the image into two categories: road pixels and non-road pixels. This is typically done using a combination of color and gradient thresholding. For example, we might threshold the image based on the hue, saturation, and value (HSV) color space, as well as the gradient magnitude and direction. By applying these thresholding techniques, we can create a binary image where the road pixels are white and the non-road pixels are black.

D. Gaussian Blur

The purpose of Gaussian blur in image processing is to make the color transition from one side of an edge in an image to another smooth rather than sudden. It averages out rapid changes in pixel intensity and is often used as a pre-processing step before other tasks such as thresholding. Gaussian blurring is a type of low-pass filter, which reduces the high-frequency components of an image, and is implemented by convolving the image with a Gaussian function. This function gives more weight to the pixels nearest the center of the kernel and less weight to those further away, resulting in a blur that preserves boundaries and edges better than other, more uniform blurring filters.

E. Laplacian

In digital image processing, a Laplacian filter is frequently used for edge detection and image sharpening. Its foundation is the Laplacian operator, a second-order derivative operator that's employed to determine how quickly a physical quantity—like intensity—changes. In image processing, areas with abrupt changes in intensity—which typically line up with an image's edges—are highlighted by applying the Laplacian operator to an image's pixel values. Three-by-three convolution kernels are frequently used to represent the discrete form of the Laplacian operator. One definition of the Laplacian kernel is...

0	1	0
1	-4	1
0	1	0

Fig 3: Laplacian Filter

You convolve a picture using this kernel in order to apply the Laplacian filter to it. Sliding the kernel over the image and calculating the sum of element-wise products at each place constitute the convolution operation. The outcome is a fresh image with edges present indicated by high positive or negative values. But because the Laplacian filter is noise-sensitive, there could be a lot of high-frequency noise in the output. The Laplacian filter is frequently used in conjunction with a smoothing technique, like Gaussian blurring, to alleviate this problem. We refer to this combination method as the Laplacian of Gaussian (LOG) filter. The Laplacian filter is helpful for specific image processing applications since it may also be described in the frequency domain. Furthermore, depending on the particular requirements of the image processing task, Laplacian filter types such as the isotropic and anisotropic Laplacian filters may be used. In conclusion, the Laplacian filter is an effective tool for identifying image edges and improving aspects of images that are associated with changes in intensity. Applications related to computer vision and image processing make extensive use of it.

F. Sobel

One popular edge detection method in image processing, especially for computer vision applications, is the Sobel filter. This filter, which bears the name of its creator, Irwin Sobel, is intended to draw attention to sharp contrasts in an image, such as edges. The `cv2.Sobel()' function in OpenCV, a well-known computer vision library, implements the Sobel filter. In order to work, the Sobel filter convolves the image using two 3x3 kernels, one of which detects changes in the horizontal direction and the other in the vertical direction. These kernels highlight regions where intensity changes most quickly by computing the image's first-order gradient. The resulting gradient's direction and magnitude reveal important details about the image's edges. Because of its computing efficiency and simplicity, the Sobel filter works well for edge detection in real-world applications. It is an essential tool for computer vision pre-processing tasks like feature extraction, picture segmentation, and object recognition. The Sobel filter can also be used in conjunction with other methods, including thresholding, to improve pertinent image attributes and further refine edge maps. All things considered, the Sobel filter is an invaluable weapon in the arsenal of image processing algorithms, serving as a vital component in the extraction of significant data from digital images.

G. Canny Edge

The Canny edge detection algorithm is a popular method for detecting edges in images. It involves several steps, including noise reduction, gradient calculation, non-maximum suppression, and edge tracking. The algorithm first applies a Gaussian filter to the image to reduce noise, then calculates the gradient magnitude and direction for each pixel. Non-maximum suppression is then used to thin the edges, and hysteresis thresholding is used to identify the final set of edges. In the context of road line detection, the Canny edge algorithm is used to detect the edges of the road lines, which are then used to calculate the lane markings.

H. *Laplacian of Gaussian*

One method that is frequently used in digital image processing for edge detection is the Laplacian of Gaussian (LOG) filter. It combines second-order differentiation and smoothing, two essential image processing techniques. The procedure entails blurring an image using a Gaussian filter, then highlighting areas of fast intensity change—which usually correlate to edges—by using the Laplacian operator.

I. Gaussian Smoothing

The input image must first be subjected to a Gaussian filter. The original image is blurred, the noise is reduced, and the image is smoothed thanks to the Gaussian filter. A kernel, or matrix, that symbolizes the filter's form defines the Gaussian filter. The amount of smoothing the image receives depends on the size of the kernel. Stronger smoothing is produced by larger kernels. To create the smoothed image, the convolution process is applied in between the image and the Gaussian filter. The convolution operation for the Gaussian smoothing can be expressed mathematically as follows:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/(2\sigma^2)} \qquad (1)$$

J. Laplacian Operator

The second derivative of the image is calculated using the Laplacian operator following the use of Gaussian smoothing. This draws attention to areas that exhibit abrupt changes in intensity, which frequently line up with the image's edges. Usually, a 3x3 or 5x5 kernel is used to represent the Laplacian operator. The smoothed image and the Laplacian kernel are subjected to the convolution procedure. The Laplacian operator can be expressed mathematically as:

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \qquad (2)$$

K. Combining the Steps

Combining the Laplacian operator and Gaussian smoothing yields the Gaussian Laplacian. The Gaussian filter can be used to convolve the image, and the Laplacian operator can then be applied.
The Laplacian of Gaussian (LOG) operation can be expressed mathematically as follows:

$$LOG(x,y) = \nabla^2(G * I) \qquad (3)$$

where G∗I stands for the Gaussian filter's convolution with the input picture. Due to its ability to provide a response that is strong at edges and weak at uniform regions, the Laplacian of a Gaussian filter is helpful for edge detection. It aids in emphasizing the divisions that separate various areas of a picture.

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

Fig 4: Laplacian of Gaussian

L. *Hough Line Transform*

In computer vision, the Hough Line Transform is a method for identifying straight lines in a picture even when there are noise or other disturbances. It functions by converting a line's Cartesian representation (y = mx + c) into a parameter space, in which each line is a point. Possible lines are tracked using an accumulator array, and lines that have been detected are shown by peaks in this array. The transform can identify lines even when they are partially occluded or incomplete in the image, which makes it particularly valuable for applications like object recognition and lane detection in autonomous vehicles. For Python, the `cv2.HoughLines() ` function from the OpenCV package offers an implementation that may be used for a variety of image processing applications.

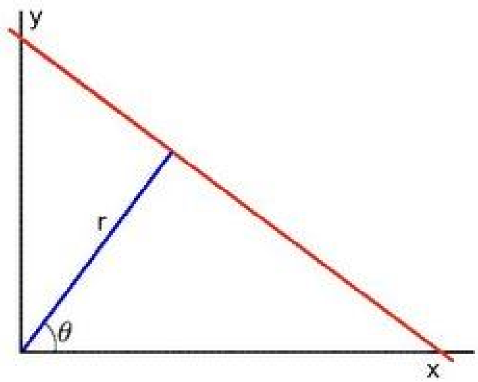
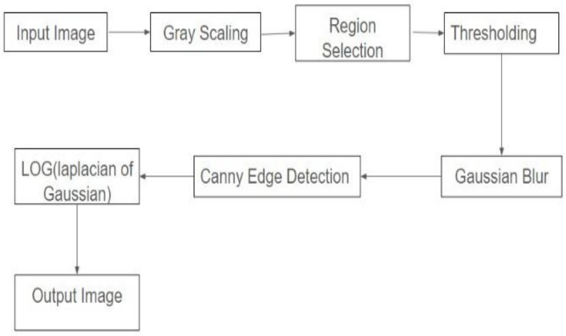


Fig 5: Hough Line Transform

IV. Flow Chart



V. Results And Discussion

The system is tested using a large number of photos of the road with various lanes, and the results show that it is above 95% accurate in identifying the lanes. When the road lanes were straight, the method performed well; however, it was less successful when the lanes were curved. The screenshots that were taken during the testing are as follows:

Fig 6: Input Image:

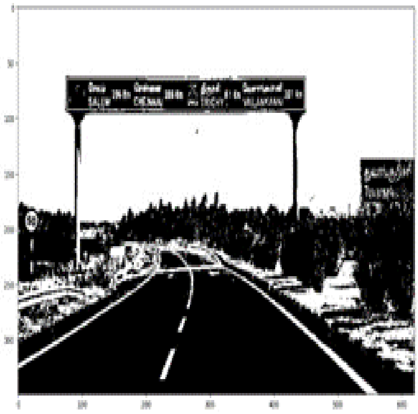


Fig 7: Threshold Image / Gray Scale Image

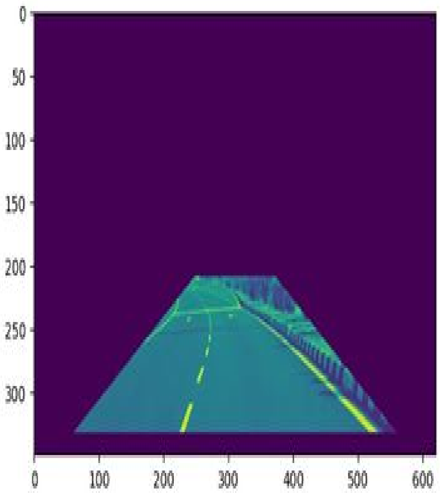


Fig 8: Region of Interest

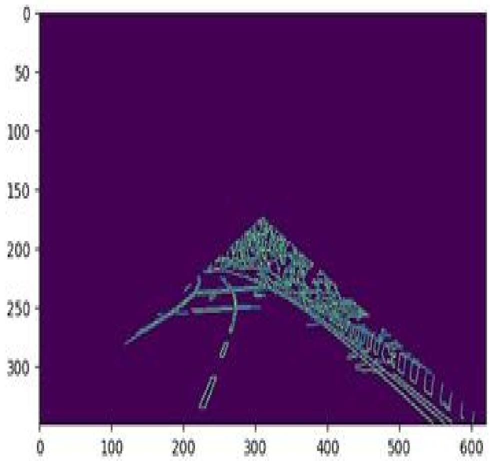


Fig 9: Lane Detection

canny Edge Detection

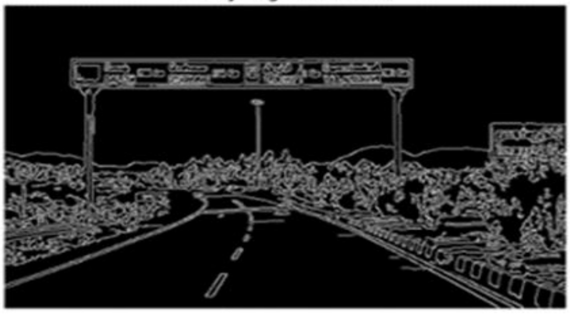


Fig 10: Canny Edge Detection

VI. CONCLUSION

In summary, OpenCV-based lane detection techniques for Advanced Driver Assistance Systems (ADAS) have proven to be very effective across a range of road circumstances. However, as technology develops, combining deep learning with convolutional neural networks (CNN) looks like a viable way to improve lane detection in the future. Even if they are strong, the conventional OpenCV-based methods could have trouble managing complex situations like complicated road geometries and changing lighting conditions. Using CNNs and deep learning can help overcome these obstacles by allowing complex spatial characteristics to be automatically extracted and temporal connections to be learned. These sophisticated models provide self-learning skills, adaptability to changing road conditions, and the possibility of continual improvement through iterative training on a variety of datasets. Real-time processing is possible despite the computing needs of deep learning because to continuous improvements in hardware and optimization methods. As technology advances, a new era of lane detecting systems with enhanced accuracy, resilience, and flexibility for ADAS vehicles operating in dynamic and unpredictable situations is about to begin with the merging of OpenCV techniques with CNN and deep learning approaches.

REFERENCES

[1] Mohammed Faisal1, Ganta Anurag Reddy1, Balusupati Anil Kumar1, Dr. D. Ajitha2 “Lane Line Detection for Autonomous Cars using Python and OpenCV” Accepted 24 June 2021 IOSR Journal of Engineering (IOSRJEN).

[2] Deepak Prasad, Vaishnavi BS, Soumya K N, “Real-Time Lane Detection for Autonomous Vehicles” 1st -International Conference on Recent Innovations in Computing, Science & Technology ISBN:978-81-963209-0-4.

[3] Aditya Singh Rathore B. Tech, J.K. Lakshmipat University, “Lane Detection for Autonomous Vehicles using OpenCV Library” (IRJET) Volume: 06 Issue: 01 | Jan 2019 e-ISSN: 2395-0056.

[4] Vishnu Balaji Eepari1, Tungala Sasi Pavan Kumar2, P Murali Krishna3, “ROAD LANE LINE DETECTION” (IJPREMS) Vol. 03, Issue 04, April 2023, pp: 510-514

[5] Abhishek Goyal, Mridula Singh, Anand Srivastava, “Lane Detection on Roads using Computer Vision” International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-9 Issue-1, October 2019.

[6] Shika Rao 1,† , Alexander Quezada 2,†, Seth Rodriguez 3,†, Cebastian Chinolla 3,†, Chan-Jin Chung 4 and Joshua Siegel 5, “Developing, Analyzing, and Evaluating Vehicular Lane Keeping Algorithms Using Electric Vehicles” Accepted: 23 September 2022 Published: 4 October 2022.MDPI.

[7] Prof. Asmeeta Mali Guide, Komal Shyam*1, Vedant Khule*2, Yukta More*3, Sangram Shinde*4 Students Department of Computer Engineering Dr. D. Y. Patil Institute of Technology Pimpri, Pune, India, “REAL TIME ROAD LANE DETECTION SYSTEM” (IJCRT) © 2023 IJCRT | Volume 11, Issue 4 April 2023 | ISSN: 2320-2882.

[8] Nidhi Lakhani1, Ritika Karande2, Deep Manek3, Vivek Ramakrishnan4, “LANE DETECTION USING IMAGE PROCESSING IN PYTHON” (IRJET) Volume: 09 Issue: 04 | Apr 2022 e-ISSN: 2395-0056.

[9] *Mr. Mustafa Surti, #Prof. (Dr.) Bharati Chourasia *PG Student, #Head-EC Department, SRK University, Bhopal, India. “Real time lane detection system using Python and OpenCV on Raspberry Pi” (IJREAM) ISSN: 2454-9150 Vol-05, Issue-06, Sep 2019.

[10] Fallah, A., Soliemani, A., Khosravi, H. (2021). Real-time Lane Detection Based on Image Edge Feature and Hough Transform. Journal of Electrical and Computer Engineering Innovations (JECEI), 9(2), 193 202. doi: 10.22061/jecei.2021.7659.418.

[11] Rong, Weibin, et al. "An improved CANNY edge detection algorithm." 2014 IEEE international conference on mechatronics and automation. IEEE, 2014.

[12] Robust lane detection and object tracking: Muhammad Shafique, Muhammad Fahim, Prashant Pydipogu - JUN 2013.

[13] ROAD LANE LINE DETECTION USING OPENCV: Dr. A. K. Madan, Divyansha Jharwal - 05/May-2022.

[14] K.C.T. Swamy, D. Venkata Ratnam and Towseef Ahmed Shaik, "Double difference method with zero and short base length carrier phase measurements for Navigation with Indian Constellation satellites L5 (1176.45 MHz) signal quality analysis", Int. J. Satell. Commun. Networking, vol. 40, no. 4, pp. 294-304, 2022.
[15] K.C.T Swamy, A. D Sarma, A Supraja Reddy, Analysis of Ionosphere scintillation of GPS and VHF/UHF signals over low latitude Indian region, Information and communi-cation Technologies (WICT), 2012 World Congress, IEEE, 2012