iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.70867

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Lane Detection for Autonomous Vehicles

Ayushmaan Kamboj[1], Dr. Archana Kumar[2]
Artificial Intelligence & Data Science, ADGIPS, GGSIPU

Abstract: ***Nowadays, autonomous vehicles are developing rapidly toward facilitating human car driving. One of the main issues is road lane detection for a suitable guidance direction and car accident prevention. This paper aims to improve and optimize road line detection based on a combination of camera calibration, the Hough transform, and Canny edge detection. The video processing is implemented using the Open CV library with the novelty of having a scale able region masking. The aim of the study is to introduce automatic road lane detection techniques with the user's minimum manual intervention.***

I. INTRODUCTION

In this modern era, cars have become an inseparable part of our lives and the most important issue regarding cars is human safety. Without a doubt, in this context, the topic of driving safely and road safety is a challenging task. In order to improve drivers' safety and reduce accidents caused by human error, road lane detection-based techniques are found to be useful. In the last decades, image processing and afterward deep learning and artificial intelligence have been employed to improve the accuracy of such a task. The image processing techniques are found to be easy to be implemented and relatively non-costly in comparison with machine learning techniques. The most common image processing method related to this task uses a combination of Canny edge detection and Hough transform. In this study, we optimize the existing methods and improve the results of the previous works by using camera calibra-tion, optimized Canny edge window selection, and automatic masking. The results are tested on video images viral on the internet related to lane detection of dangerous roads, as well as, videos taken by cameras placed in front of a car to capture road videos. The proposed approach uses minimum manual intervention and is easy to be used on either microchips or computers, either on real-time or offline videos, and gives on-time notice to the driver in case of crossing the lines or detecting any object entering the road on the lane.

II. PROPOSED FRAMEWORK

In this section, we discuss the method, improvements, and evaluation meters. The one view of the lane detection approach can be seen in Fig. 1. In the following, we give further details.

A. Camera Calibration

Many cameras have the problem of lens distortion. Dis- tortion, or frequently named camera sectioning, leads to the process of estimating the parameters of a lents image sensor,as we can see an example in Fig. 2. In this study, camera calibration was used to solve this problem [3].

B. Gray Scale

As a first step, a grayscale filter is used to prepare the data for Canny edge detection. This step computes the magnitude (norm) of the gradient for each pixel, according to Sobel derivatives in x and y directions. Using Canny edges for each channel would lead to a slow algorithm and delay the image processing algorithm. In Fig.3 we show one of the grayscale images of the test data. Readers can refer to [4] for more details on the grayscale process by averaging the values in Red, Green, and Blue channels.

C. Noise Removal and optimized region masking

Due to the fact that the main image processing elements used for the line detection involve Canny edges and Hough transform, mathematical operations such as derivatives are involved. Such operators lead to high sensitivity to image noise. To avoid such a problem it is crucial to deal with the noise problem before we further proceed with lane detection. In our work, we use a simple Gaussian filter for noise removal. As the process is sensitive, as it can lead to image blurring and loss of significant data, selecting a proper window size of the Gaussian filter is another important issue. Through experiments, it was noticed that for different cameras the size of the Gaussian filter varies. To cope with such a problem we propose an optimization method using a loop for 5 different sizes of the kernel.

©IJRASET: All Rights are Reserved

The kernels are taken of sizes 3X3, 5 X 5, 9 X 9, 15 X 15, and 20 X 20. In the setup of the camera onthe first run, we suggest two options. The first option is forthe scenario where test data and its correct line detection is

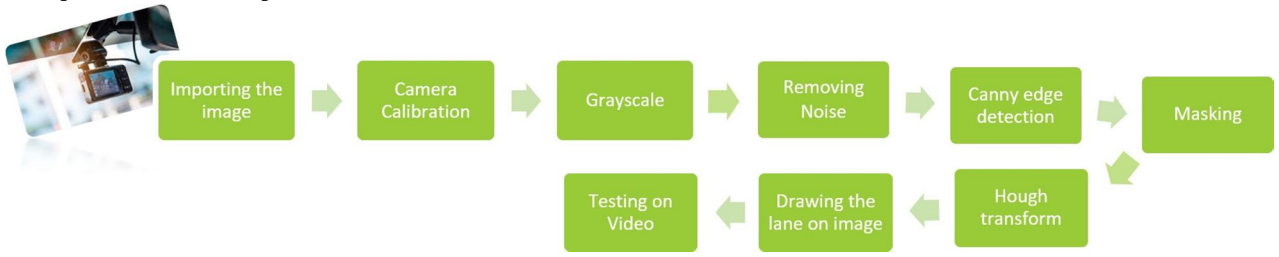
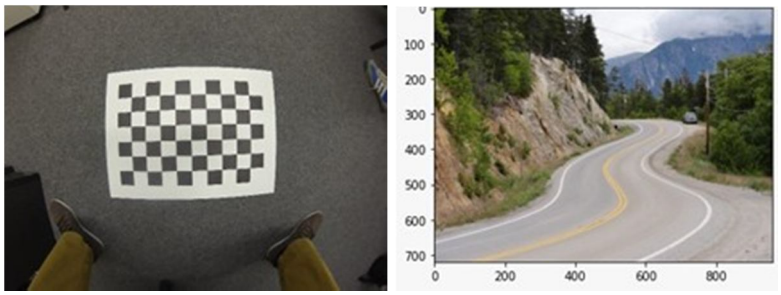


Fig. 1: Approach of Lane Detection in one view



(a) Camera calibration (b) Test image using the camera calibration

Fig. 2: The left image shows the camera calibration process before setting it in front of the car [3], whereas the right imageis an image taken from one of the most dangerous roads in US [5].

provided and the second scenario has a deficiency of such data. In this regard, we work on having a completely automatic process. Fig. 3 shows the result of the Gaussian filter already applied in this research for one of the processed images.

Considering the fact that images and videos come from a camera placed on the forehead of the car, only a certain part ofthe images are in the interest of the lane detection approach. This relates to the triangle defined by the parameters of the lens of the camera [6] as shown in Fig. 4. One of the main contributions of this work relates to figuring out the mask sizebased on the image size of the camera using the width and length of the photo as x and y. The mask coordinates formula is given in Eq. 1. The mask makes the choice of area of interest automatically. This means that this method is looking for the lanes on the Approximate location of the road and will not search for the lanes in other areas of the picture like the sky which is above the masking area. Fig. 4 shows the output of such a mask.

$$[0, 0], ([x/2, y/2] - x/10), ([x/2, y/2] + x/10), [0, max(x)] \quad (1)$$

D. Canny edge detection

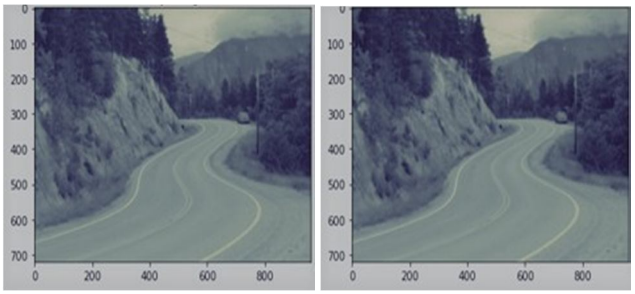
Canny edge detection is a well-known edge detection method that is mostly used to detect boundaries based onderivative. In the method proposed by John F. Canny, the gradient of pixels had been used to detect dense regions wherethe intensity changes are occurring. The result of the Canny edge detector [7] [8] on the test image is shown in Fig. 5.Here Canny edge detection is applied in the standard way where the derivative of the image are computed in x and$y$ directions using a Sobel kernel. Then the magnitude has been computed as shown in Eq. 2. Furthermore, the gradient orientation information can be computed through Eq. 3. The result of such operations is visually shown in Fig.7.

$$magnitude(G) = \sqrt{(G_x^2 + G_y^2)} \quad (2)$$

$$Angle(G) = \tan^{-1}\frac{G_y}{G_x} \quad (3)$$

Non-maximum suppression: After calculating the values of gradients and their directions, by checking the directions ofall pixels, every pixel in another direction of the maximum in the neighborhood will be deleted.

Edge tracking by Hysteresis: After non-maximum suppres- sion, for detecting the lanes on the road, a minimum value andmaximum values are chosen for the intensity gradient in the way that every edges below the minimum value and above themaximum value had been removed from the edges. For any values between this minimum and maximum, the decision hadbeen made based on their connectivity [9].



(a) Gray scale (b) Gaussian blur

Fig.3: Removing the noise by Gaussiankernel blur filter for the gray scale image on the left.

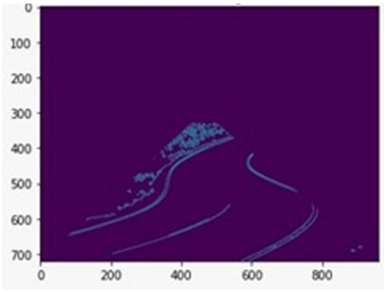


Fig. 4: Automatic masking the region of interest. Size of the camera using the *x* and *y* coordinates of the image as detailed in Eq. 1

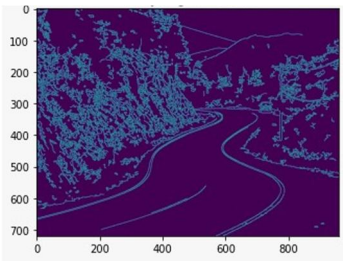


Fig. 5: The result of Canny edge detector on our test image

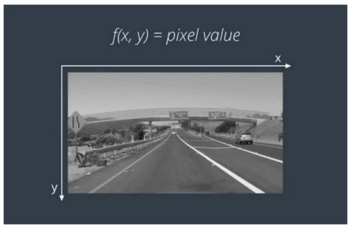


Fig. 6: Road Lane detection with function of a Gray scale

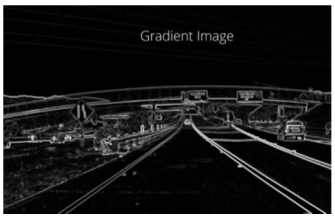


Fig. 7: Gradient image of figure 6

E. Hough probabilistic transform

Hough transform is a well-known algorithm to detect lines and had been used in many autonomous driving projects andit is one of the most important factors to achieve the maingoal of this project. This method was developed by Paul Hough in 1962 to find the presence, position, and angles of shapes in three different forms linear, circular, and general [9].The shape detection using the Hough transform can generally be summarized with the following steps. Firstly, edges are determined in the source image, then the image is made binary (black, white) using a threshold method and finally,for each edge pixel, the possible shapes of each edge pixelare graded by increasing the polar coordinate values of the possible geometric shapes that the point may be on, one by one on a matrix used and then the shapes with the highest matrix value are the ones with the most votes, they are likely to be found in the image [10]. In our approach, Hough transformhad been used to connect edge pixels from the Canny edge detector together and make a line. As an output of Hough transforms, all the dashed and solid lane lines in the given image will appear. The result of the Hough transform can be seen in Fig. 8 [11].

III. EXPERIMENT SETUP

In this section, we give more detail on the experiment setup and outputs. The architecture of the proposed method for lane detection has been already shown in Fig.1. For detecting laneson the road, from the point of view of a 48-megapixel camera of a mobile phone (Note 9 pro) on the forehead of the car, we used open-source as well as real data collection testing purposes by the authors themselves. Thus, 30 pictures and

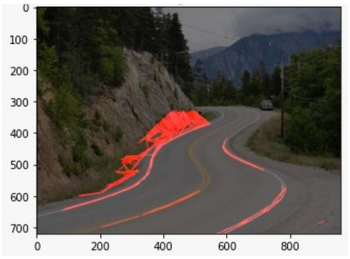


Fig. 8: Result image



Fig. 9: Images from the test set taken by the authors of this work

10 videos had been taken for research purposes. All those videos will be provided for research purposes by request. All of the aforementioned test pictures in this study are taken fromdeserts in Iran as seen in Fig.9.

As we described above camera calibration is the first pre- processing step taken. To achieve such calibration we usePython inbuilt function cv2.undistort() from Open CV-Library. This step is followed by cv2color() function of Open CVlibrary to achieve the grayscale filter. The noise removal or overall smoothing of the image said in other words isimplemented by using Gaussian blur filter. Through experi- ments, it was noticed that for different cameras the size ofthe Gaussian filter varies. To cope with such a problem we propose an optimization method using a loop for five differentsizes of kernels as described above. The algorithm takes into account the scenario of provided and not provided accurate line detection. For the first scenario, the algorithm will compute the bestfit to one of the five already detected line detection resultsas discussed above and then use as kernel size this filter for the following image and video processing.

In case of unprovided correct segmentation, the algorithm will require the user to choose among the five shown results as the best fit results of lane detection. Based on the user reply the kernel size is chosen for that camera and its image processing. For the second case the authors are aware that the user may be unaware of the difference between the shown result, but asthe process is required only once a professional advice can be easily provided by specialized people in this field.

Canny edge detection is the following step to be applied.



Fig. 10: The result of our lane detection on our test data

For such a process we use cv2.Canny() function of Open CV library with a low threshold of 50 and a high threshold of 150 on the already blurred images.

Masking the region of interest automatically is the next step considered. For masking the region of interest, firstly, a black image with the exact size of the test image had been made and then the triangle of region of interest using cv2.fillPolly() function based on Eq.1 is constructed. After- ward, cv2.bitwise˙and() function had been used to overlay the result of Canny edge detection and masked picture.

The Hough transform had been performed to convert the edges to lines, using cv2.HoughLinesP() function of OpenCV. After the Hough space is difined rho and theta resolutionthreshold are chosen, and the number of intersections sur-passing those values is considered a "line" is determined. In this study, the rho is defined as three and the low thresholdas 50, and the high threshold as 150. Furthermore, each line extracted from Hough transform has been drawn using cv2.line() function of the Open CV library. The result of this method is shown on four different test images in Fig.10.

A. Finding the Slopes of Lines

Since cv2.HoughLinesP() finds all Lines, there could be multiple lines returned. Thus, it is needed to understand whether the road lines coincide with the right or left side ofthe vehicle. Thus, the slope of each line will be determined using Eq. 4.

$$slope = (y_2 - y_1)/(x_2 - x_1), \quad (4)$$

where (x_1, y_1) and (x_2, y_2) are two points in each line representing the lower and upper left and right points of aline. The slope threshold is 0.5 to count a line as curved.After counting a line as curved and calculating its slope, appointing the left or right lane is the next step. For this aim, the image will be divided into half, and if line's x values are higher than the center value and the calculated slope valueis positive, it is considered a right lane and if the x valuesof a line are lower than the center value and the slope valueis negative it is considered as a left lane. After appointing

Test Videos	Our method	Hough Transform
Video 1	0.924	0.85
Video 2	0.913	0.85
Video 3	0.911	0.85

TABLE I: TP rate comparison of the proposed architecture in comparison with old models.

lines as right or left lanes, while using their averaged *x* and*y* values, the polynomial fitting had applied to create a first- degree polynomial shown in Eq. 5 and Eq. 6. This will avoid lines out of the area of interest such as lines above the roador right and left not part of the road.

$right_m, right_b = np.polyfit(right_x, right_y, 1)$ (5)

$left_m, left_b = np.polyfit(left_x, left_y, 1)$ (6)

The Eq. 6 can be solved by choosing a *y* value to find the new *x* value which will be used to create a linear line across the image frame. After these steps, a linear line will be made which represents the right or left lanes on a video in real-time [15].

IV. EXPERIMENTAL RESULTS

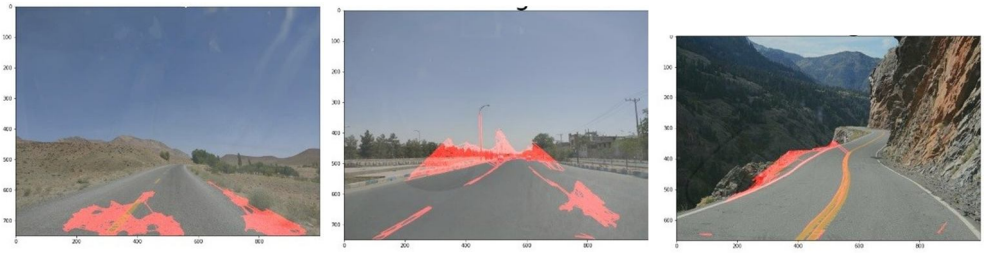
To evaluate the success of the method proposed in this study,two measuring factors had been used, Run time and TruePositive. The author had captured videos and images, as it was mentioned before. The summation of the videos shown in this work, where previous work would struggle for correct linedetection is about 105 sec. This does not limit the results of this work and can be easily proven by any user. For measuringthe success of the lane detection system proposed in this study, firstly, TP rate had been used based on Eq. 7 that can be seen in Table I.

TP = TrueLaneDetectedPixels/TotalPixelsDetected. (7)

The second factor of evaluation of the method proposed in thisstudy was the time of execution which is shown in Table II. Interms of the accuracy of lanes detected by the algorithm, it canbe claimed as our accuracy was more than 90 percent in each experimented image and video. However, in terms of executiontime, there is a place for improvement, also it is almost fast enough to use in real-time [24]. For a better understating of the improvement made with the proposed architecture we show some compared images from the experimented date- set. Fig. 11 shows the results of the study's test data with existing models for the same data set experiment of Fig. 10. We can easily notice that the proposed architecture improves drastically the lane detection accuracy in comparison withpreviously existing Hough transform based methods. Eventhough there is a loss of speed, which does not limit the real time application of the model, we gain a high quality of the line detection.

Test Case	Duration	Time of Execution
Image 1	0	almost 0
Image 2	0	almost 0
Image 3	0	almost 0
Image 4	0	almost 0
Video 1	25sec	52.3sec
Video 2	30sec	54sec
Video 3	50sec	1min 1s

TABLE II: Time duration of the proposed model



(a) Test image 1 (b) Test image 2 (c) Test image 3

Fig. 11: The result of traditional Hough transform on low quality pictures before optimizations of this study

V. CONCLUSION

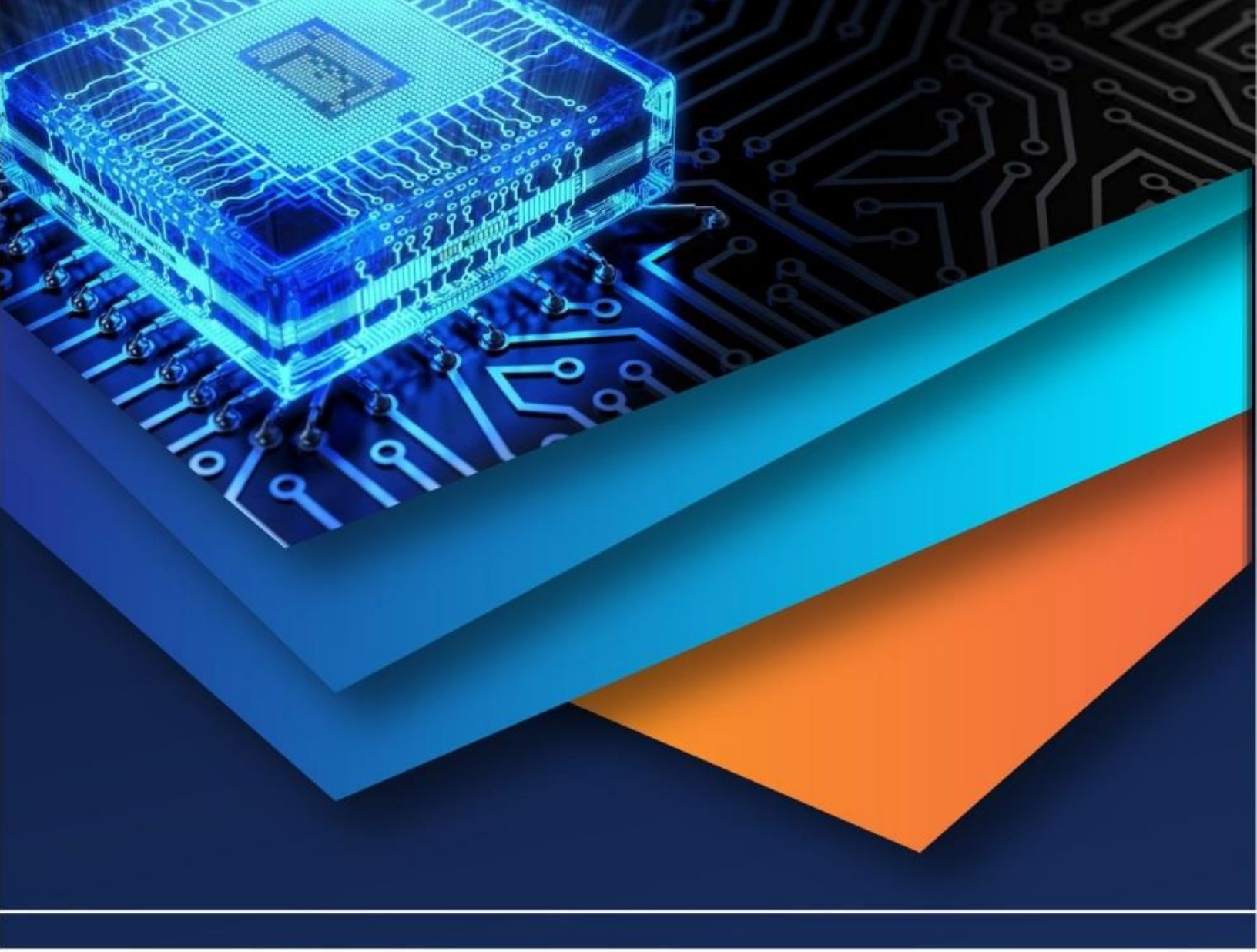
In this study, our aim was to propose and develop a pure image and video processing approach to detect the lanes onthe road automatically in the most possible optimized wayfor better usage in real-time processing. The main method of this study was based on previous works on Hough transform- based road lane detection and we added an automatic camera calibration and furthermore an optimization for making and choosing the best kernel size in Gaussian blur to make the method scale able for all

sizes of camera's frames for easeof use. The method proposed by this work started with some prepossessing steps like camera calibration, grey scale, Gaussian blue and further, we proposed a novel idea to make the masking the region of interest in the process completely automatic. As shown in the aforementioned figures and tables,the results of lane detection in this study on test images and test videos were incurably amazing and almost fast, compatible for aiming the goal mentioned before as real-time processing. However, more improvements on better accuracy of the curve detection can be provided.

REFERENCES

[1] C.Y. Kuo, Y.R. Lu and S.M. Yang, On the Image Sensor Processing for Lane Detection and Control in Vehicle Lane Keeping Systems, Interna- tional Program on Energy Engineering, National Cheng Kung Univer- sity, Tainan 70101, Taiwan, 8 April 2019, https://www.mdpi.com/1424- 8220/19/7/1665/pdf.
[2] Archit Rastogi, Computer Vision: Lane Finding Through Image Process- ing, 6 Sep 2020, https://medium.com/swlh/computer-vision-lane-finding- through-image-processing-516797e59714
[3] Qi, Wang Li, Fu Zhenzhong, Liu. (2010). Review on Camera Calibration. 3354 - 3358. 10.1109/CCDC.2010.5498574.
[4] Kanan, Christopher Cottrell, Garrison. (2012). Color-to-Grayscale: Does the Method Matter in Image Recognition. PloS one. 7. e29740. 10.1371/journal.pone.0029740.
[5] "Learn the Latest Tech Skills; Advance Your Career." Udacity, https://www.udacity.com/.
[6] Chin-Pan Huang, Chaur-Heh Hsieh, Jin-Long Li. A Robust Lane De- tection Method Using Adaptive Road Mask. Department of Computer and Communication Engineering, Ming Chuan University, Taiwan. (2015) https://www.academia.edu/10351901/A˙Robust˙Lane˙Detection
[7] Samuel, Moveh Mohamad, Maziah mad saad, Shaharil Hussein, Mohamed. (2018). Development of Edge-Based Lane Detection Algo- rithm using Image Processing. JOIV: International Journal on Informatics Visualization. 2. 19. 10.30630/joiv.2.1.101.
[8] Canny J. F., 1986. A Computational Approach to Edge Detection. IEEE Trans. Pattern Anal. Machine Intel, vol. PAMI-8, no. 6, pp. 679-697.
[9] Matas, Jiri Galambos, Charles. (1998). Progressive Probabilistic Hough Transform.
[10] Fokkinga, Maarten. (2011). The Hough transform. J. Funct. Program.. 21. 129-133. 10.1017/S0956796810000341.
[11] N. Kiryati, Y. Eldar, A.M. Bruckstein, A probabilistic Hough trans- form, Pattern Recognition, Volume 24, Issue 4, 1991, Pages 303- 316, ISSN 0031-3203, https://doi.org/10.1016/0031-3203(91)90073-E. (https://www.sciencedirect.com/science/article/pii/003132039190073E)
[12] Statistical Bulletin: School Leavers 1992/93, https://books.google.co.uk/books?id=98qnxgEACAAJ, 1995 ,Department of Education
[13] R. O. Duda, and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures", the Comm. ACM, Vol.15, No.1, 1972, pp. 11-15.
[14] Kim, Sunghoon Park, Jeong-Ho Cho, Seong Park, Soonyoung Lee, Kisung Choi, Kyoungho. (2007). Robust Lane Detection for Video-Based Navigation Systems. 2. 535 - 538. 10.1109/ICTAI.2007.20.
[15] Zuroski, Kathyrn. (2021). Finding the slope of a line using Interactivate.
[16] Lim, Kai Li Braunl, Thomas. (2019). A Methodological Review of Vi- sual Road Recognition Procedures for Autonomous Driving Applications.
[17] Sharma, Anmol Kumar, Maneesh Gupta, Rajkamal Kumar, Rajesh. (2021). Lane Detection using Python.
[18] Pansambal, Suvarna Kumar, Udaya. (2019). Lane Datasets for Lane Detection. 0792-0796. 10.1109/ICCSP.2019.8698065.
[19] Lu, Pingping Cui, Chen Xu, Shaobing Peng, Huei Wang, Fan. (2021). SUPER: A Novel Lane Detection System. IEEE Transactions on Intelligent Vehicles. PP. 1-1. 10.1109/TIV.2021.3071593.
[20] Mongkonyong, Peerawat Nuthong, Chaiwat Siddhichai, Supakorn Yamakita, Masaki. (2018). Lane detection using Randomized Hough Transform. IOP Conference Series: Materials Science and Engineering. 297. 012050. 10.1088/1757-899X/297/1/012050.
[21] Dai, Xingjian Xie, Jin Qian, Jianjun Yang, Jian. (2020). Lane Detection Combining Details and Integrity: an Advanced Method for Lane Detection. 1-6. 10.1145/3446132.3446145.
[22] Fan, C. Xu, J. Di, S. Shi, X.. (2013). Lane detection based on parallel Hough transform. Journal of Computational Information Systems. 9. 4893-4900. 10.12733/jcis6333.
[23] Xiao, F., Jin, L. Haopeng, W. (2010). A study of image retrieval based on hough transform, 3rd IEEE Computer Science and Information Technology (ICCSIT), Chengdu, China.
[24] Cao J, Song C, Song S, Xiao F, Peng S. Lane Detection Algorithm for Intelligent Vehicles in Complex Road Conditions and Dynamic Environ- ments. Sensors (Basel). 2019 Jul 18. SongSong Cao, J.; Song, C.; Song, S.; Xiao, F.; Peng, S. Lane Detection Algorithm for Intelligent Vehicles in Complex Road Conditions and Dynamic Environments. Sensors 2019, 19, 3166. https://doi.org/10.3390/s19143166.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)