Int. Jrl. of **Human Computations & Intelligence**
Vol. 02, Issue 03, 2023
ISSN: 2583-5696

RESEARCH ARTICLE

Road Lane Detection System Using Image Processing

Bhavana N . Darshan T . Sushanth S . Shiny Fernandes . Ankitha C

School of Computig and Information Technology
REVA University, Bangalore Karnataka India

Received: 19 April 2023 / Revised: 25 April 2023 / Accepted: 8 May 2023

DOI: 10.5281/zenodo.7937958

Abstract – Real-time automated road lane detection is an indispensable part of intelligent vehicle safety system. The most significant development for intelligent vehicles is driver assistance system. This driver assistance system holds great promise in increasing safety, convenience and efficiency of driving. The driver assistance system involves camera-assisted system which takes the real-time images from the surroundings of the vehicle and displays relevant information to the driver. Thus, intelligent vehicles automatically collect the road lane information and vehicle position relative to the lane. Consequently, the system used by the intelligent vehicles provides the means to alert the drivers which are swerving off the lane without prior use of the blinker. So, intelligent vehicles will clearly enhance traffic safety if they are extensively taken into use. Hence the current research work is taken up to develop a middle man system that takes the image and processes it to provide the ROI in the image.
Index Terms – Road Lane Detection, Image Processing, Edge Detection, Region of Interest

I. INTRODUCTION

Autonomous vehicles have gradually received stronger attention due to its great commercial value, but still they have some safety issues that need to be overcome, such as detecting proper lanes on which the vehicles can be driven Moreover it can also endanger the lives of passengers Nearly 60,000 accidents have occurred due to poor road conditions. Road lane detection is an essential part of autonomous driving systems. It involves detecting and tracking the boundaries of lanes on a road to assist the vehicle in staying within its lane and avoiding collisions. Lane detection is accomplished using computer vision techniques, including image processing, edge detection, and machine learning algorithms.

The first step in lane detection is to acquire an image of the road using cameras mounted on the vehicle. Once the image is captured, the next step is to preprocess it to enhance its quality and reduce noise. This preprocessing step may involve techniques such as contrast adjustment, color space conversion, and filtering.The next step is to detect the edges of the road lanes. Edge detection

MILESTONE RESEARCH FOUNDATION
Int. Jrl. of Human Computations & Intelligence
Vol. 02, Issue 03, 2023
ISSN: 2583-5696

algorithms are used to identify the pixels in the image that represent the edges of the road. These algorithms may include Canny edge detection or Sobel edge detection, among others.

Once the edges are detected, the next step is to group them into lanes. Lane detection algorithms use heuristics and machine learning algorithms to group the edges into individual lanes. These algorithms may include Hough transform, Kalman filter, or deep learning-based methods. Finally, the detected lanes are tracked over time to ensure that the vehicle remains within its lane. This process involves using motion estimation algorithms to predict the movement of the lanes over time. In conclusion, road lane detection is a critical component of autonomous driving systems. By detecting and tracking the boundaries of lanes on the road, these systems can help ensure the safety and reliability of the vehicles.

II. RELATED STUDIES

[1]The work at hand proposes a fast and robust ego- lane detection algorithm that aims to be as simple as possible to enable real-time computation while being able to adapt to a variety of driving scenarios. The algorithm combines and extends three independent pieces of research as follows. Road segmentation is performed on a "shadow- free" representation of an RGB image using a histogram correlation method, and serves as a contextual prior for the detection of lane markings, which are extracted by means of ridge or "creaseness" detection. The work at hand presents an integrative vision-based approach to aid an autonomous vehicle in navigating in urban environments, by combining and extending a set of independent pieces of research. A previously published road segmentation algorithm was extended to an approach based on histogram which significantly improved segmentation performance.This alteration proves viable especially in dense urban.

[2]An algorithm for road lane detection was proposed, based on dynamic videos that were recorded by the on-board camera installed inside a moving car. The method applied in the algorithm was a combination of lane detection, vanishing point and image subtraction. The method succeeded in the implementation of road triangle extraction, road-width calculation for scale factor determination and Region of Interest (ROI) for road boundary object detection (vehicle). Hough transform was still able to track the loss of lane marks by assuming the lane was still there by counting the number of lost frames.[3]roadlane detection using feature extraction and filtering. This algorithm is capable to detect lane on straight and curved roads under different day light conditions, shadows and other noises.[4] this is all about how we can achieve this thing entirely using image processing without even knowing knowledge Machine Learning, Artificial Intelligence and any other extra concepts.

The Major Algorithms or techniques used are Canny edge detection and Hough line transform which is an root of this project this paper is all about how we can achieve this thing entirely using image processing without even knowing knowledge Machine Learning, Artificial Intelligence and any other extra concepts. The Major Algorithms or techniques used are Canny edge detection and Hough line transform which is an root of this project Each component depends on each other. and if one did not work, the whole system would fail to give a removed background. However, the tracked lanes cut

MILESTONE RESEARCH FOUNDATION
Int. Jrl. of **Human Computations & Intelligence**
Vol. 02, Issue 03, 2023
ISSN: 2583-5696

across the vehicle during the tracking, thus, when masking was applied and the image multiplication was executed, part of the vehicle was lost. [5] The objective of this paper is to evaluate the performance of different edge detectors at different conditions in order to find which edge detectors work better. There is therefore the need to know which edge detector performs better in terms of simplicity, image quality and speed. Four edge detectors which are Sobel, Canny, Prewitt and Roberts will be evaluated for edge detection and lane detection based on simplicity; quality of image and time (speed) taken to perform the operation.

While Roberts technique did not only run at the shortest time, but also was the most accurate in detecting lane markings for the same algorithm. Roberts edge detector proved to be more suitable in terms of simplicity, image quality and speed, whereas Canny, Sobel and Prewitt on the other hand prove to be the more sensitive and time consuming.[6] A fast and improved algorithm with the ability to detect unexpected lane changes is aimed in this paper. Based on the characteristics of physical road lane, this paper presents a lane detection technique based on H-MAXIMA transformation and improved Hough Transform algorithm which first defines the region of interest from input image for reducing searching space; divided the image into near field of view and far field of view. The proposed method has been developed using image processing programming language platform and was tested on collected video data. Promising result was obtained with high efficiency of detection.

III. THE PROPOSED APPROACH

In our proposed system, the road lane detection algorithm has 8 stages. The first stage is Converting our RGB image into Grayscale image. In the second stage, we perform image enhancements using Gaussian smoothening or median filtering. In the third stage, we apply Canny Filter to extract the edge detection of the image. In the Fourth stage, the Hough's Transform is applied to detect two sets of solid lanes(Left and right lanes). In the Fifth stage, we detect the Intersection Point when the two solid lanes (left and right lanes) intersects and overlaps each other and we use an horizontal reference line so that we have 3 points of intersection. In the Sixth stage, we apply flood fill technique to all the pixels to detect the different connected components which provides us with the region of relatively similar intensities. In the Seventh stage, we perform Feature Extraction to calculate the width of each connected component and remove many unwanted regions using Unwanted Region Subtraction with Filtering. And In the Eighth stage we find the Region Of Interest(ROI) by highlighting the prominent regions of the image found in the previous stage. And In the end we get the processed image.

1. RGB IMAGE

An RGB image, sometimes referred to as a truecolor image, is stored as an m-by-n-by-3 data array that defines red, green, and blue color components for each individual pixel. RGB images do not use a palette. The color of each pixel is determined by the combination of the red, green, and blue intensities stored in each color plane at the pixel's location. Graphics file formats store RGB images as 24-bit images, where the red, green, and blue components are 8 bits each. This yields a potential of 16 million colors. The precision with which a real-life image can be replicated has led to the nickname "truecolor image."

An RGB MATLAB® array can be of class double, uint8, or uint16. In an RGB array of class double, each color component is a value between 0 and 1. A pixel whose color components are (0,0,0) is displayed as black, and a pixel whose color components are (1,1,1) is displayed as white. The three color components for each pixel are stored along the third dimension of the data array. Component can take any value between 0 (black) and 1 (white). For example, if each component takes a value between 0 and 1, the total range of possible colors is 0 to 255. When these three numbers are added together, the resulting number represents the light's intensity. Each R, G and B level can range from 0% to 100% full intensity. Each level is represented by the range of decimal numbers from 0 to 255 -- 256 levels for each color -- equivalent to the range of binary numbers from 00000000 to 11111111 or hexadecimal 00 to FF. The total number of available colors is 256 x 256 x 256 or 16,777,216. RGB mode is best used when digital designs are in your project pipeline. While many printers nowadays are capable of converting RGB files to CMYK, it's best to avoid conversion at this stage if possible, because your colors will not be accurate in the final output.

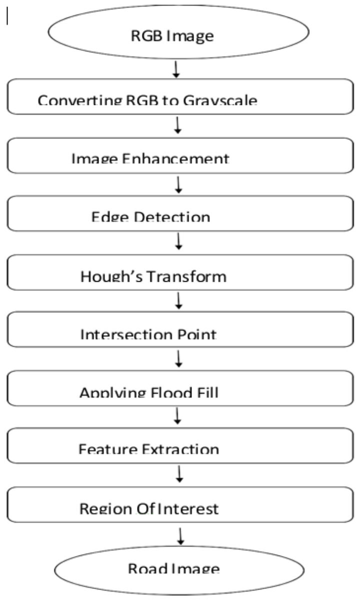
2. Converting RGB to Grayscale

Image formation using sensor and other image acquisition equipment denote the brightness or intensity I of the light of an image as two dimensional continuous function F(x, y) where (x,y) denotes the spatial coordinates when only the brightness of light is considered. Sometimes three-dimensional spatial coordinate are used. Image involving only intensity are called gray scale images. Similar to one-dimensional time signal, sampling for images is done in the spatial domain, and quantization is done for the brightness values. In the Sampling process, the domain of images is divided into N rows and M columns. The region of interaction of a row and a Coolum is known as pixel. The value assigned to each pixel is the average brightness of the regions.

The position of each pixel was described by a pair of coordinates (xi, xj).The resolution of a digital signal is the number of pixel is the number of pixel presented in the number of columns × number of rows. For example, an image with a resolution of 640×480means that it display 640 pixels on each of the 480 rows. Some other common resolution used is 800×600 and 1024×728,among other. Resolution is one of most commonly used ways to describe the image quantity of digital camera or other optical equipment. The resolution of a display system or printing equipment is often expressed in number of dots per inch. For example, the resolution of a display system is 72 dots per inch (dpi) or dots per cm. Gray levels represent the interval number of quantization in grayscale image processing. At present, the most commonly used storage method is 8-bit storage. There are 256 gray levels in an 8bit gray scale image, and the intensity of each pixel can have from 0 to 255, with 0 being black and 255 being white we. Another commonly used storage method is 1-bit storage. There are two gray levels, with 0 being black and 1 being white a binary image, which, is frequently used in medical

Int. Jrl. of **Human Computations & Intelligence**
Vol. 02, Issue 03, 2023
ISSN: 2583-5696

images, is being referred to as binary image. As binary images are easy to operate, other storage format images are often converted into binary images when they are used for enhancement or edge detection.



3. Image Enhancement

Image enhancement refers to the process of highlighting certain information of an image, as well as weakening or removing any unnecessary information according to specific needs. For example, eliminating noise, revealing blurred details, and adjusting levels to highlight features of an image.

4. Edge Detection

Edge detection is the process of identifying boundaries between different regions in an image. An edge is defined as a sharp transition in intensity between adjacent pixels in an image. Edge detection is a critical step in many computer vision and image processing applications, as it allows us to extract useful information from images, such as object boundaries, shape, and texture. There are many edge detection algorithms available, and each has its own strengths and weaknesses.

Fig. 1: Image Before Edge Detection

MILESTONE RESEARCH FOUNDATION
Int. Jrl. of Human Computations & Intelligence
Vol. 02, Issue 03, 2023
ISSN: 2583-5696

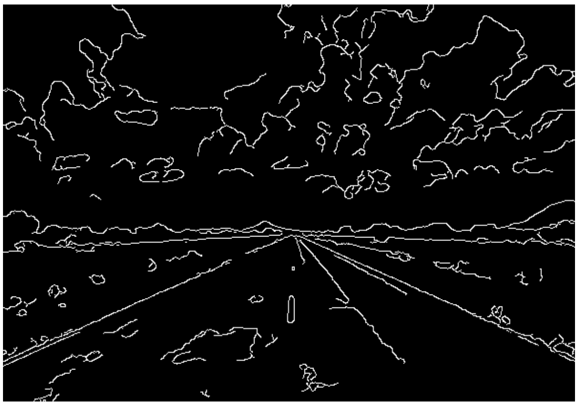


Fig. 2: Image After Edge Detection

5. Hough's Transform

An edge point produces a cosine curve in the Hough Space. From this, if we were to map all the edge points from an edge image onto the Hough Space, it will generate a lot of cosine curves. If two edge points lay on the same line, their corresponding cosine curves will intersect each other on a specific (ρ, θ) pair. Thus, the Hough Transform algorithm detects lines by finding the (ρ, θ) pairs that have a number of intersections larger than a certain threshold. It is worth noting that this method of thresholding might not always yield the best result without doing some preprocessing like neighborhood suppression on the Hough Space to remove similar lines in the edge image.

6. Intersection Point

The intersection point is generally hidden in the intersection region and exhibits no obvious gray features, therefore, it cannot be directly detected. The ridge line fitting method is proposed to accurately detect the coordinates of the intersection points by using the ridge lines of the grid lines as references. The ridge lines of the grid lines are firstly detected, secondly, the segments of the ridge lines with intense disturbances are eliminated and the rest of the segments are fitted. Finally, the intersection points of the horizontal and vertical fitted lines are detected as the lower intersection points of the grid.

7. Applying Flood Fill

Flood fill (also known as seed fill) is an algorithm that determines the area connected to a given node in a multi-dimensional array. It is used in the "bucket" fill tool of a paint program to fill connected, similarly colored areas with a different color and in games such as Go and Minesweeper for determining which pieces are cleared. When applied on an image to fill a particular bounded area with color, it is also known as boundary fill. The flood fill algorithm takes three parameters: a start node, a target color, and a replacement color. A queue-based implementation using Breadth–first search (BFS) is done. Depth–first search (DFS) starts from the source node in the matrix, replace its colour with the replacement colour and recursively explore all its valid eight adjacent pixels, and replace their colour. We are replacing the colour of every processed node, next time as it will have a different colour.

8. Feature Extraction

Feature extraction is a part of the dimensionality reduction process, in which, an initial set of the raw data is divided and reduced to more manageable groups. So, when you want to process it will be easier. The most important characteristic of these large data sets is that they have a large number of variables. These variables require a lot of computing resources to process. So, Feature extraction helps by selecting and combining variables into features, thus, effectively reducing the amount of data. These features are easy to process, but still able to describe the actual data set with accuracy and originality.

9. Region of Interest

A region of interest (ROI) is a portion of an image that you want to filter or perform some other operation on. You define an ROI by creating a binary mask, which is a binary image that is the same size as the image you want to process with pixels that define the ROI set to 1 and all other pixels set to 0. You can define more than one ROI in an image. The regions can be geographic in nature, such as polygons that encompass contiguous pixels, or they can be defined by a range of intensities. In the latter case, the pixels are not necessarily contiguous. Image processing functions operate not only on entire images but also on image areas. Image region of interest (ROI) is a rectangular area that may be either some part of the image or the whole image. ROI of an image must be defined by the size and offset from the image. The origin of an image shall be in the left and our final output image with ROI will be on the right

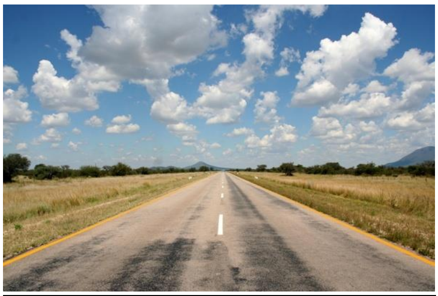
10. Implementation Details

Python is the programming language used in this execution because of its clear, brief, and intuitive syntax and it has very vast library with support for image processing and easy to complete the task. The IDE used for this execution is Pycharm IDE.

IV. RESULTS and CONCLUSION

This proposed model is able to detect the road lane using computer vision and various python libraries, and this Model overcomes some of the challenges that are commonly faced with road lane detection models. This paper presents a detailed overview of our proposed system. We were able to identify detect the road lane using computer vision and various python packages and algorithms. For Future scope for this project we can add an AI feature that can shift between different edge detection filters to increase the efficiency and accuracy of this algorithm, We can even highlight different lanes with different colors in an multi-lane scenario with proper passing lane and incoming lane.

MILESTONE RESEARCH FOUNDATION
Int. Jrl. of **Human Computations & Intelligence**
Vol. 02, Issue 03, 2023
ISSN: 2583-5696

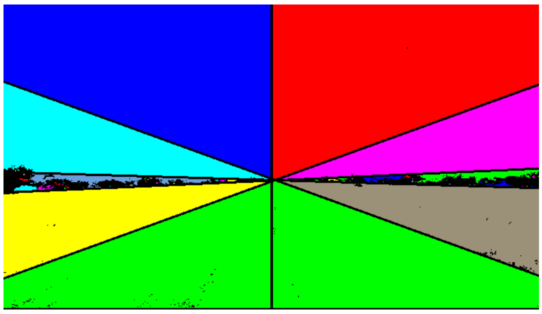

Step – 1


Step – 2


Step – 3


Step – 4


Step – 5


Step – 6


Step – 7

REFERENCES

1. McCall, J. C., & Trivedi, M. M. (2006). Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *IEEE transactions on intelligent transportation systems*, *7*(1), 20-37.

2. Amditis, A., Bimpas, M., Thomaidis, G., Tsogas, M., Netto, M., Mammar, S., ... & Cicilloni, R. (2010). A situation-adaptive lane-keeping support system: Overview of the safelane approach. *IEEE Transactions on Intelligent Transportation Systems*, *11*(3), 617-629.
3. Nur, S. A., Ibrahim, M. M., Ali, N. M., & Nur, F. I. Y. (2016, November). Vehicle detection based on underneath vehicle shadow using edge features. In *2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)* (pp. 407-412). IEEE.
4. Brutzer, S., Höferlin, B., & Heidemann, G. (2011, June). Evaluation of background subtraction techniques for video surveillance. In *CVPR 2011* (pp. 1937-1944). IEEE.
5. Chiu, K. Y., & Lin, S. F. (2005, June). Lane detection using color-based segmentation. In *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.* (pp. 706-711). IEEE.
6. Sun, T. Y., Tsai, S. J., & Chan, V. (2006, September). HSI color model based lane-marking detection. In *2006 ieee intelligent transportation systems conference* (pp. 1168-1172). IEEE.
7. Wang, Y., Teoh, E. K., & Shen, D. (2004). Lane detection and tracking using B-Snake. *Image and Vision computing*, *22*(4), 269-280.
8. Thouheed Ahmed, S., & Sandhya, M. (2019). Real-time biomedical recursive images detection algorithm for Indian telemedicine environment. In *Cognitive Informatics and Soft Computing: Proceeding of CISC 2017* (pp. 723-731). Springer Singapore.
9. Wei, X., Zhang, Z., Chai, Z., & Feng, W. (2018, August). Research on lane detection and tracking algorithm based on improved hough transform. In *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)* (pp. 275-279). IEEE.
10. Al-Shammari, N. K., Alzamil, A. A., Albadarn, M., Ahmed, S. A., Syed, M. B., Alshammari, A. S., & Gabr, A. M. (2021). Cardiac stroke prediction framework using hybrid optimization algorithm under DNN. *Engineering, Technology & Applied Science Research*, *11*(4), 7436-7441.
11. Bottazzi, V. S., Borges, P. V., Stantic, B., & Jo, J. (2014). Adaptive regions of interest based on HSV histograms for lane marks detection. In *Robot Intelligence Technology and Applications 2: Results from the 2nd International Conference on Robot Intelligence Technology and Applications* (pp. 677-687). Springer International Publishing.
12. Srivastava, S., Singal, R., & Lumba, M. (2014). Efficient lane detection algorithm using different filtering techniques. *International Journal of Computer Applications*, *88*(3).
13. Tolba, M., & Moustafa, M. (2016). GAdaBoost: accelerating adaboost feature selection with genetic algorithms. *arXiv preprint arXiv:1609.06260*.
14. Ahmed, S. T., Basha, S. M., Ramachandran, M., Daneshmand, M., & Gandomi, A. H. (2023). An Edge-AI enabled Autonomous Connected Ambulance Route Resource Recommendation Protocol (ACA-R3) for eHealth in Smart Cities. *IEEE Internet of Things Journal*.
15. Jin, C., Wang, X., Miao, Z., & Ma, S. (2017, October). Road curvature estimation using a new lane detection method. In *2017 Chinese Automation Congress (CAC)* (pp. 3597-3601). IEEE.
16. Xu, F., Chen, L., Lou, J., & Ren, M. (2019). A real-time road detection method based on reorganized lidar data. *PloS one*, *14*(4), e0215159.
17. Kumar, B. M., Guduru, R. K. R., Srinivas, A., Ana, F., Ramudu, K., & Dhiman, G. (2022). Wavelength assignment in optical fiber with intelligent optimization and assignment scheme for static and dynamic traffic intensity based Photonic networks. *Optical and Quantum Electronics*, *54*(8), 526.
18. Karthik, P., Kumar, B. M., Suresh, K., Sindhu, I. M., & Murthy, C. G. (2016, May). Design and implementation of helmet to track the accident zone and recovery using GPS and GSM. In *2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)* (pp. 730-734). IEEE.
19. Nagarathna, C., Kumar, B. M., Bhavana, N., Manjushree, T. L., & Pattan, D. (2022). Improve the Efficiency of Large RFID Network Using Enhanced Security Data Delivery Model for Machine Learning Based Network Intrusion Detection System–A Survey. *International Journal of Human Computations & Intelligence*, *1*(4), 10-17.
20. Chattoraj, M., & Vinayakamurthy, U. R. (2021). A self adaptive new crossover operator to improve the efficiency of the genetic algorithm to find the shortest path. *Indonesian Journal of Electrical Engineering and Computer Science*, *23*(2), 1011-1011.