IJARIIT INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 7, Issue 4 - V7I4-1850)
Available online at: https://www.ijariit.com

Lane detection and object detection system steering advisory for driver

Vinutha H.
vinuthamadhusudhana@gmail.com
RajaRajeswari College of Engineering, Bengaluru, Karnataka

Shivam Zagade
zshivam25@gmail.com
RajaRajeswari College of Engineering, Bengaluru, Karnataka

Saima Sharieff
sharieffsaima786@gmail.com
RajaRajeswari College of Engineering, Bengaluru, Karnataka

Saniya Taj
connectosaniya@gmail.com
RajaRajeswari College of Engineering, Bengaluru, Karnataka

***Abstract*— Lane detection and tracking modules are now regarded essential in the development of any Intelligent Transportation System (ITS). Different vision-based algorithms are being used in autonomous vehicles to determine road lanes are presented, reviewed, and compared in this proposed work. The main components are lane departure, lane tracking, collision avoidance, driver assistance system. Lane departure component assists in keeping the vehicle stable on the desired lane on the road. The front collision avoidance component will detect the road's frontal obstruction and shows up a pre-collision/proximity warning signal. The notice is based on the vehicle's speed and the object's distance from the cars. According to current research, the system can only identify one lane marking set in real time and is unable to provide extra lanes for assistance in scenarios such as lane closures, road repairs, and car accidents that impede the driving lane. The project proposes a method for determining the marking of two lanes that might be used in conjunction with a detecting system to detect obstructions on the road ahead of the vehicle. This method takes a strong approach to lane detection and performs admirably in a variety of lane types.**

Keywords*— *lane detection, object detection, image processing, image segmentation, openCV, haar cascade algorithm, etc.

I. Introduction

In today's world, traffic accidents have become one of the most significant issues. Roads are the most popular route of transportation because they provide the best links between other types of transportation. The most common traffic issue is driver irresponsibility, which is becoming increasingly significant as the number of vehicles on the road grows in number. One of the primary functions of the Intelligent Transportation System is to improve human safety and save lives (ITS). Intelligent transportation systems (ITS) are high-tech applications that attempt to provide cutting-edge services in the areas of transportation and traffic management. This technology allows different vehicle users to be better informed and use transportation networks in a safer, easier, more coordinated manner. Road lanes or white markers that assist the driver in distinguishing between the road and non-road areas can help to reduce these road accidents. A lane is a section of road that is designated for use by a single lane of cars in order to govern and guide drivers and decrease traffic congestion. This research proposed a lane detecting system for cars operating in dynamic contexts with complex road conditions. Generally speaking, lane detection algorithms require the following steps : (1) generation of lane markings, (2) grouping of lane markings, (3) fitting of lane models, and (4) time monitoring. The extraction of the correct lane is crucial to the successful production of the lane-mark. Many standard methods are used to detect a lane using details of Edge, colour, intensity, shape. In addition, the identification of lane Can be interpreted as an image segmentation problem. However, most approaches are susceptible to changes in lighting, temperature Condition and noise; and thus many conventional lane detection systems malfunction where there are major differences in the surrounding environment.

In certain high-end cars, protective systems are used to identify barriers and stop-offs, although none of them are entirely autonomous. Existing automotive automation features are insufficient to allow cars to drive themselves. Drivers are always in need; without them, the car would be unusable. However, with self-driving automobiles, we can ensure that cars are always present on the road. For typical cars, the driver must constantly monitor signals, safety signs, lanes, and barriers and make correct decisions accordingly. Self-driving cars are no longer a dream, but are now becoming reality. Companies declare their commitment to developing and releasing self-driving cars, and several of them discuss the level of autonomy that is currently being created. Self-driving cars can be dangerous for some people, but they also have benefits. This would result in less traffic congestion, lower emissions, lower overall travel expenses, and lower costs for new roads and services. It would also help in improve the mobility of elderly and physically disabled persons.

II. EXISTING SYSTEM

The findings were acquired utilising a sample of data feed from real-world road environment situations at the time of road testing. A monocular camera installed in the windshield of a non-autonomous car and data bus instruments were used to gather images and vehicle signals. The IPM algorithm can be used to determine the range of tracking for a variety of ROI sizes. The smaller ROI studied covers a range of 10.4 metres ahead of the vehicle, whereas the larger covers 34.5 metres.

III. PROPOSED SYSTEM

In certain high-end cars, protective systems are used to identify barriers and stop-offs, although none of them are entirely autonomous. Existing automotive automation features are insufficient to allow cars to drive themselves. Drivers are always in need; without them, the car would be unusable. However, with self-driving automobiles, we can ensure that cars are always present on the road. For typical cars, the driver must constantly monitor traffic signals, road safety signs present on road, barriers, and the lanes and make decisions accordingly. Self-driving cars are no longer a pipe dream; they are becoming a reality.

One of the most deadly types of error is driving, which results in fatalities and traffic congestion. Accidents are more frequent as a result of basic human errors such as talking on the phone while driving or using loud entertainment systems in cars. Aside from these mistakes, mental and physical limitations can also contribute to driving failure. These errors are becoming more common by the day, and it is becoming increasingly necessary to decrease them using today's technologies. Self-driving cars are the answer not only to reducing these errors, but also to expanding our driving capabilities and creating more efficient road management systems.

The built model prototype seeks to implement automation by handling duties such as self-driving by detecting lane lines. The system design for implementing the same will consist of three units first is an input unit containing a camera and image, the second is the processing unit which is our laptop that will act as a server, the neural network will be running within server and third is the object present in front will be detected.
Firstly, in the input unit, the camera to stream input data. Two client programs will be running on server one to stream images collected by the camera and another to send data for bot control. The processor then takes care of multiple tasks, including data collection, neural network training and steering prediction, stop sign and signal detection, distance measuring with a vision-based system, and giving commands to the bot.

A program is executing on the server to receive image frames and data. The image frames will be decoded into NumPy arrays after being transformed to grayscale. The convolutional neural network, which will be trained to produce steering predictions based on identified lane markers, will be used. For training and prediction, the lower half of the input image will be used. There will be input layer, hidden layers and output layers, there will be 3 nodes in the output layer each corresponding to steering control instructions that are left and right. It is possible to collect training data or to use datasets that are already available. Each frame will be clipped and turned into a NumPy array for training purpose of the system. The train label will then be linked with the train image. The npz file will then contain all of the paired image data and labels. The neural network will be trained using OpenCV. The weights will be saved in an XML file after training, and a new neural network will be built and loaded with the learned XML file to generate predictions. A shape-based technique will be employed for signal and stop sign detection as part of object detection, while a Haar feature-based cascade classifier will be used for object detection in the image captured. The Haar cascade is utilised because each object requires its own classifier. The server will be used to imitate button press actions. The output commands will be given and writes out low or high signals, simulating actions that are necessary to drive the car in the center of lane.

IV. IMPLEMENTATION

A. *Segmentation*

The use of some kind of color-space thresholding has been shown to be the most common method of segmenting in an input frame. In visual applications, HSV is the most extensively used colour space. This is due to the fact that humans have direct access to colour, which makes interpretation and modification easier. Binary output images from both colour spaces were compared in the proposed strategy, and it was observed that the RGB technique was more successful in segmenting red signs. The HSV thresholds were set to the following generous ranges: Hue: 134-180, Saturation : 0-155, Value 0-255 in OpenCV's "in Range" method. Even with the large red hue ranges, the segmented output image does not contain all indicators. In addition, the HSV segmentation has greater noise (non-signs). As a result, the RGB thresholding ranges were found to provide better segmentation for input frames captured with the web camera under Test Video 1 illumination circumstances, and were the segmentation method of choice in the proposed strategy.

B. *The Canny Edge Detection Technique*

The goal of edge detection technique is to identify the boundaries of objects within images present. A detection is used to look for areas in an image where the intensity changes dramatically. An image can be recognised as a matrix or an array of pixels. The light intensity at a certain location in the image is represented by a pixel. The intensity of each pixel is represented by a numeric number ranging from 0 to 255; a value of zero implies no intensity if something is entirely black, while 255 shows maximum intensity when something is completely white. A gradient is a series of pixels with varying brightness. A steep change is indicated by a strong gradient, whereas a shallow change is shown by a modest gradient. The brightness discontinuity at the spots where the gradient grows stronger is shown by the contour of white pixels. Because an edge is defined by the difference in intensity values between neighbouring pixels, this helps us discover edges in our image. Wherever there is a sudden change in intensity (rapid change in brightness), i.e., where there is a strong gradient, there is a bright pixel in the gradient image. We get the edges by tracing out all of these pixels. This notion will be used to detect the edges in our road image.

C. *Gaussian Blur*

In a grayscale image, each pixel is characterised by a single number that indicates the pixel's brightness. The common solution for smoothing an image is to change the value of a pixel with the average value of the pixel intensities around it. A kernel will average out the pixels in order to reduce noise. This kernel of normally distributed numbers (np.array ([[1,2,3], [4,5,6], [7,8,9]]) is applied to our entire image, smoothing it out by setting each pixel value equal to the weighted average of its neighbours. In this case is used is a 5x5 Gaussian kernel.

D. Edge Detection

An edge is a region in a picture where the intensity/color between consecutive pixels in the image changes dramatically. A strong gradient is defined as a steep change, whereas a shallow change is defined as a shallow change. In certain ways, an image can be thought of as a stack of matrices with rows and columns of intensities. This means that a picture can also be represented in 2D coordinate space, with the x axis traversing the width (columns) and the y axis traversing the image height (rows). The Canny function measures the change in brightness between neighbouring pixels by performing a derivative on the x and y axis. In other words, we're calculating the gradient (brightness change) in all directions. It then uses a series of white pixels to trace the strongest gradients.

E. Region of Interest

The image's dimensions are chosen to include the road lanes and to designate the triangle as our region of interest. The image's dimension is then used to generate a mask, which is effectively an array of all zeros. To make our region of interest dimensions white, we fill the triangle dimension in this mask with the intensity of 255. Now I'll use the canny picture and the mask to do a bitwise and operation, which will yield our final region of interest. definition of a region (image).

F. OpenCV

OpenCV is a programming library geared mostly at real-time computer vision. It was created in order to establish a standard infrastructure for computer vision applications and to speed up the incorporation of machine perception into commercial products.

G. Hough Transform

The Hough transform is a technique for extracting features. The technique's goal is to use a voting mechanism to locate imperfect instances of objects inside a given class of forms. This voting mechanism takes place in a parameter space (Hough Space), from which object candidates are obtained as local maxima in an accumulator space that is expressly generated using the Hough transform algorithm.

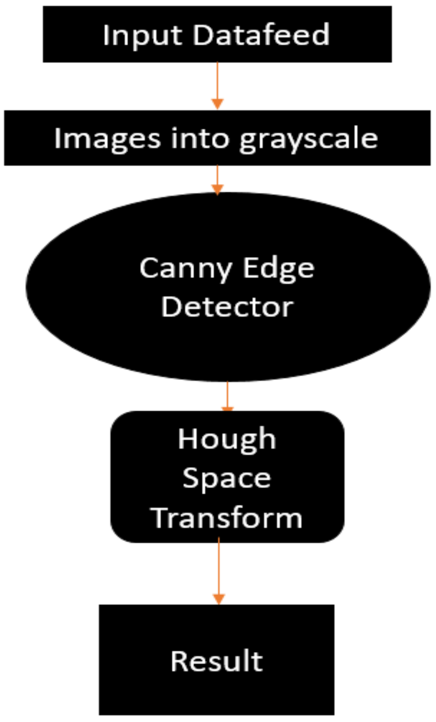

Fig. Work flow of proposed system.

V. TESTING

Software testing is an examination that is conducted to provide stakeholders with information about the quality of the product or service being tested. Product testing can also provide a company with an objective, unbiased view of the software, helping them to grasp and understand the dangers involved with its adoption. The process of executing a program or application with the goal of detecting software bugs is known as testing (errors or other faults detected). Software testing is process of evaluating one or more qualities of interest by executing a system component or a software component. These features, in general, indicate the degree to which the component or system under test is functional. The test types are:

- Unit Testing.
- Integration Testing.
- Validation Testing.
- User Acceptance Testing.
- Output Testing

A. Unit testing

Unit testing, a type of software testing, examines individual units or components of software. The goal of unit test is to ensure that each unit of software code works as intended. Unit testing is carried out by developers throughout the development (coding) phase of an application. Unit tests are used to isolate a part of code and ensure that it is correct. A unit can be defined as a single function, method, process, module, or object. Before going on to integration testing, unit testing is the initial phase in the testing process.

B. Integration Testing

Integration testing is a type of testing in which software components are conceptually connected and tested together as a single entity. A typical software project consists of a number of software modules created by a number of different programmers. The goal of this level of testing is to find flaws in how these software modules interact when they're put together.

C. Validation Testing

During the development phase or at the conclusion of the development process, determining if software fulfils specified business requirements. Validation testing ensures that the product meets the customer's requirements. It may also be described as showing that the product works as expected in the intended environment.

D. User Acceptance Testing

User Acceptance Testing (UAT) is a sort of testing in which the end user or customer verifies and accepts the software system before it is moved to the production environment. After functional, integration, and system testing, UAT is performed as the final step of testing.

An acceptance test's performance is essentially the user's show. User motivation and knowledge are essential for the system's proper operation. The aforesaid tests were carried out on the newly designed system, which met all of the requirements. The step by step presentation of tests performed is given in fig 1.1.

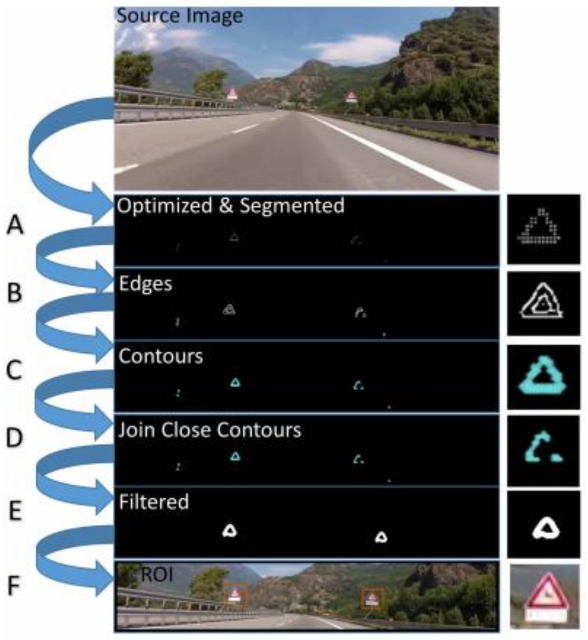

Fig. 1.1 Step-by-step Testing process implementation.

RESULTS

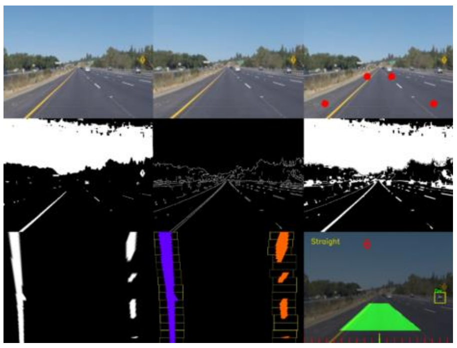

Fig. Lane and object detection

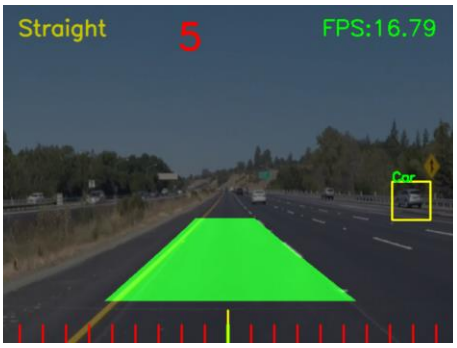

Fig. Image Processing done per step

REFERENCES

[1] Aditya Kumar Jain, "Working model of Self-driving car using Convolutional Neural Network, Raspberry Pi and Arduino", Proceeding of the 2nd International Conference on Electronics, Communication and Aerospace Technology (ICECA), IEEE, 2018, pp: 1630- 1635.
[2] Raivo Sell, Anton Rassõlkin, Mairo Leier, Juhan-Peep Ernits, "Self-driving car ISEAUTO for research and educational 19th international conference on Research and Education in Mechatronics (REM), IEEE, 2018, pp: 111-116.
[3] Jeremy Straub, Wafaa Amer and Christian Ames," An Internetworked Self-Driving Car System-of-Systems", 12th system of systems engineering conference, IEEE, 2017.
[4] Qudsia Memon, Muzamil Ahmed and ShahzebAli, "SelfDriving and Driver Relaxing Vehicle", in IEEE 2016, pp: 170-174.
[5] Mohammad Faisal Bin Ahmed, Md. Saef Ullah Miah, Md. Muneef Anjum Timu, Shakura Akter, Md. Sarker.b,"The Issues and the Possible Solutions for Implementing SelfDriving Cars in Bangladesh" in Regin 10 Humanitarian Technology Conference(R10-HTC), IEEE,2017, pp: 250-254.
[6] Margrit Betke, EsinHaritaoglu, Larry S.Davis,"Real-time multiple vehicle detection and tracking from a moving vehicle", in Machine Vision and Application, IEEE, 2000, pp: 69-83.
[7] Chun-Che Wang, Shih-Shinh Huang and Li-Chen Fu, PeiYung Hsiao "Driver Assistance System for Lane Detection and Vehicle Recognition with Night Vision", IEEE.
[8] R. Mohanapriya, L.K. Hema, Dipesh warkumar Yadav, Vivek Kumar Verma, "Driverless Intelligent Vehicle for Future Public Transport Based On GPS", in International Conference on Signal Processing, Embedded System and Communication Technologies and their applications for Sustainable and Renewable Energy (ICSECSRE `14), Vol. 3 Special Issue 3, April 2014, pp: 378-384.
[9] Ruturaj Kulkarni, Shruti Dhavalikar, Sonal Bangar, "Traffic Light Detection and Recognition for Self Driving Cars using Deep Learning" in 4th International Conference on Computing Communication Control and Automation(ICCUBEA), IEEE, 2018.
[10] Giuseppe Lugano, "Virtual Assistants and Self-Driving Cars", IEEE, 2017, pp: 1-5.
[11] Dong, D., Li, X., &Sun, "A Vision-based Method for Improving the Safety of Self-driving", 12th international conference on reliability, maintainability and safety (ICRMS),2018, pp: 167-171.
[12] T. Banerjee, S. Bose, A. Chakraborty, T. Samadder, Bhaskar Kumar, T.K. Rana, "Self Driving Cars: A Peep into the Future", IEEE, 2017, pp: 33-38.