Lane Segmentation for Self-Driving Cars using Image Processing

Aman Tanwar[1], Jayakrishna[2], Mohit Kumar Yadav[3], Niraj Singh[4], Yogita Hambir[5]

[1,2,3,4,5]Department of Computer Engineering, Army Institute of Technology, Pune, Maharashtra, India

---***---

Abstract - *Technology is advancing day by day, more advanced cars are being built every year but still we are not able to reduce the no. of road accidents. Approximately 1.35 million people die each year as a result of road traffic crashes. Road traffic crashes cost most countries 3% of their gross domestic product. More than half of all road traffic deaths are among vulnerable road users: pedestrians, cyclists, and motorcyclists.*

When the vehicle is four-wheeler and an accident occur than the chances of serious injuries or even deaths increases. We need more efficient systems which can prevent the accidents and help us to reduce them. One of the most common mistakes committed by human driver is talking on phone while driving or not paying attention on the road. Sudden change of the lanes leads to accident.

A lane detection system can be built and which can identify the lanes and indicate the driver on sudden alteration in the lanes. Most of the car companies have ongoing projects on these technologies. This can be done with the help of image processing.

I. INTRODUCTION

A lane segmentation system is built using image processing. Image processing can be done with help of the python library like OpenCV. OpenCV provides various functions and tools to work on frames captured by the camera. With the help of OpenCV many complex calculations can be done easily.

For making a prototype we need a camera, a bot and a raspberry pi. The camera will be mounted on the top of the bot and the raspberry pi will be fit on it. The raspberry pi will be operated with the help of a battery. The camera will capture the live events and provide them to raspberry pi. The camera captures frames which are then passed to the raspberry pi which does further processing on the frames. We can even control the frame rate too. We need to import NumPy library in our code which provides a high-performance multidimensional array object, and tools for working with these arrays.

For better efficiency and good outcome, we need powerful cameras.

The model architecture is shown in figure 1.

A. Morphological Transformations:

Morphological transformations are operations which are performed on the images on the basis of their shapes. The image is first converted in binary form and then the transfor-mations are applied. Two inputs are given to the function. First is the original image and second is the kernel or the structuring element which decides the nature of the operation. There are many types of morphological transformations. The two basic types are Erosion and Dilation.

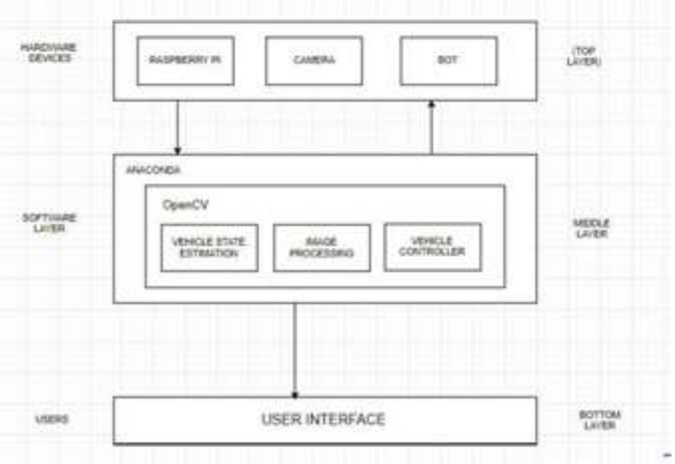

Fig. 1. Three-layer architecture.

1) Erosion:: Erosion is similar to soil erosion the unwanted part of the image gets eroded away after erosion. Mostly the boundaries of the foreground object are eroded away. The two inputs original image and structuring element are compared. After applying this operation on the frame, we get three conditions: -

1.Fully match

2.Partial match

3.No match

If there is full match then we replace the pixel in the real image by 1. If there is partial match then we replace the pixel in the real image by 0. And if there is no match the pixel is replaced by 0.

In erosion black pixels are increased and white pixels are decreased. After erosion the image gets shrank.

2) Dilation:: Dilation is opposite if erosion. It increases the white region int the image and reduces the black portion in the image. The size of the foreground object also increases. Similar

to erosion the two inputs the original and the structuring element are compared. After applying this operation on the frame, we get three conditions: -

1.Fully match

2.Partial Match

3.No match

If there is full match then we replace the pixel in the real image by 1. If there is partial match then we replace the pixel in the real image by 1. And if there is no match the pixel is replaced by 0.

Normally in cases of noise removal cases, the erosion is followed by dilation. Erosion removes the white noise but also shrinks the size of the object. After this we perform dilation since noise is gone it won't come back but our object area which shrank after erosion will increase. It is very helpful in joining parts of broken object.

Performing dilation just after erosion is called opening.

B. Hough Line Transform:

Hough transform is an effective technique for detecting lines and other shapes in an image. It uses basic mathematics and some methods to join the different edges in an image. These edges can be detected by using different tools like sobel edge detector, canny edge detector or any other tool. The main draw backs of Hough Transform are computational complexity and storage requirement. Therefore, modified versions such as modified Hough transform (MHT) and windowed random Hough transform (RHT). Although, as per our project require-ment we will stick to the standard Hough transform (SHT) in this paper.

Once the edges are detected by using the above mentioned tools, we can simply loop through them and can somehow find out the shape but this is very time-consuming and ineffective method. Hence, we use Hough transform which gives more weight age to the points which are already in a line and allows the points on the image to "vote" and due to the mathematical properties of transform "voting" helps in finding out the most prominent line in the image. Joining the non-black points in the x-y plane.

II. PROPOSED METHODOLOGY:

Hough transform works on the principle that managing a collection of points i.e a line is much more difficult than managing a point. Firstly, we will see how to transform a line into a point. A line is basically represented on a x-y plane and it has two parameters i.e slope and intercept. Therefore, we create a new coordinate for representing the parameters. This is called parameter space or m-c space since its made from the m-c coordinates. This way we can represent a line into a point and the same is shown in the figure (a) and figure (b).

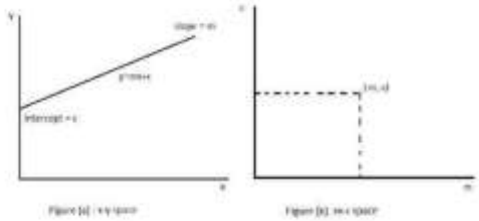

Fig. 2. Graph Plot 1.

Now the question arises how does this space transformation help in detecting the lines and different shapes in the given image.

Well, the Hough transformation basically represents the non-black points of x-y plane in the m-c space. These points become line in the m-c space (as shown in the figure) and intersect at a common point, this point is nothing but the line

Thus, the lines and shapes in the given image gets effectively identified.

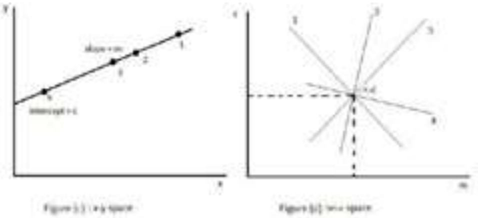

Fig. 3. Graph Plot 2.

III. FEASIBILITY TEST:

We implemented Hough line transform taking a video from source https://www.youtube.com/watch?v=KEYzUP7-kk U. The images of successful implementation are shown below:

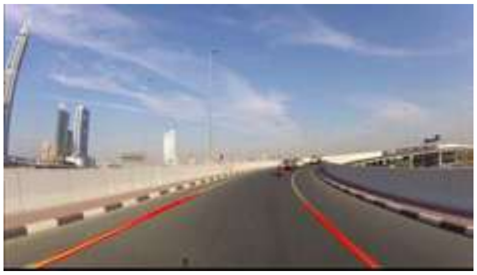

Fig. 4.

IV. CHALLENGES:

There are many challenges like the moving vehicles in front of our vehicle will make difficult to identify the lanes. The climate will be problem. During night It becomes difficult to identify the lanes and even the obstacles or vehicles. If there is fog the camera won't be able to detect the lanes. Moreover, we require good equipment's like good quality camera and faster processing systems.

The traffic conditions in current scenarios are not ready to accept this technology. Many times, people do not follow traffic rules. Sometime the traffic light is not functioning. There

are many places which don't have markings. Speed limit set is not followed. There are many flaws in the current traffic management system. All the challenges must be taken care of while designing the system. How the system should react in cases of camera failure of any other technical error, should be described precisely.

The model should be tested under every possible situation.

V. ADVANTAGES:

Successful implementation of the project will help in mak-ing the future fully automated cars: driver less cars. Will surely help in reducing the number of accidents per year. Can be used in many applications like in military for war operations.

Unmanned vehicles are required in the places where humans can't go. In organizations to carry loads. Will reduce the time and increase the efficiency of work. A single man can't drive for 24 hours continuously but a robot can. Will be beneficial for the taxi services in the future.

VI. CONCLUSION:

We have successfully understood the method of lane segmentation for self-driving cars using image processing. Self-driving cars are the future of automated transport system, proper inclusion of imperative features like adaptive cruise control, collision avoidance system, night vision, blind spot detection and traffic sign detection will make the project substantially useful for safe-traffic and military purposes.

VII. REFERENCES:

1) Lane Detection Based on Connection of Various Feature Extraction Methods by Mingfa Li, Yuanyuan Li, and Min Jiang,2018.

2) Towards End-to-End Lane Detection: an Instance Segmen-tation Approach Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, Luc Van Gool,2018.

3) Design and Implementation of Traffic Sign and Obstacle Detection in a Self-Driving Car using SURF Detector and Brute Force Matcher-Renjith.R,Reshma R,Arun K V,2017.

4) Lane following and lane departure using a linear-parabolic model-Cla´udio Rosito Jung*, Christian Roberto Kelber, 2017.

5) Self Driving Cars: A Peep into the Future-T.Banerjee,Bose, A.Chakraborty, T.Samadder,Bhaskar Kumar, T.K.Rana,2017.

6) End-to-End Ego Lane Estimation based on Sequential Transfer Learning for Self-Driving Cars-Jiman Kim, Chanjong Park, 2017.

7) Implementation of Vehicle Detection Algorithm for Self-Driving Car on Toll Road Cipularang using Python Language-Mochamad Vicky Ghani Aziz,Hilwadi Hindersah,2017