

AN EFFICIENT LANE DETECTION USING DEEP LEARNING MODEL

Priyanga S¹, Moohambikai M²

¹Computer Science &Engineering , RVS College of Engineering,priyngaselvaraj93@gmail.com

²Computer Science &Engineering, RVS College of Engineering,mooahambika@gmail.com

Abstract—Introduction of Selfdriving cars has led to the requirement of developing new technologies and concepts related to it. Lane Detection is one among them, whose variants also has their own applications in different computer vision models. The objective of the work is detecting lanes that describe the path for these self-driving cars. Lanes are detected with the help of the white lines that are in both sides of the lanes. The core idea is to use frame masking and hough transform. masking can be used for detecting the white lines and hough transform which in general is used for detecting geometrical shapes will be employed for lane detection. The developed model will be used in detecting the lane from a video that contains road.

Keywords—Lane Detection, Frame Masking, Hough Transform.

I. INTRODUCTION

1.1 Need For Lane Detection

Autonomous vehicles are the one that are capable of operating themselves and they don't need any human intervention. This makes the vehicle to sense all things that surround it which includes detection of lanes in the road. Detecting the lanes becomes the inherent part of other works such as the controlling the steering and breaking. The problem of detecting the lane can be formulated as, given an image as shown below, the objective of the model is to detect the lane.

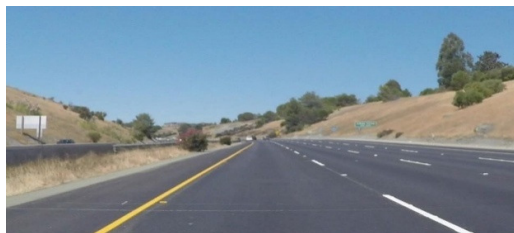


Figure 1.1 Image in which the Lanes should be detected

1.2 Objectives

To design the following models

- Lane Detection model based on Hough Transformation
- A Deep learning model to detect the Lanes

While this phase of the work concentrates on the designing of a model based on hough transform. The second phase of the work would concentrate on designing the deep learning model.

1.3 Deep Learning

Deep learning is the concept that varies from the earlier machine learning models in such a way that the performance of the system varies with the amount of data that has been used for training the system. The following figure 1.2 depicts it. It is inferred from the figure that the performance of the older machine learning model learns till a rate and then it becomes constant, there will be no improvement in the performance with respect to the increase in the size of the training data. But in case of deep learning the performance of the system constantly increases with the increase in the size of the data.

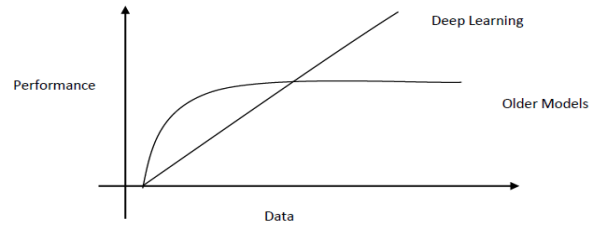


Figure 1.2 Advantage of deep learning model than the other models

II. PROPOSED SYSTEM

The proposed system is twofold, it includes a model based on hough transform for detecting the lanes and designing of a deep neural network model for the same purpose and identify the better model. This phase of the work concentrates on designing a model for detecting lane in an image.

2.1 System Architecture

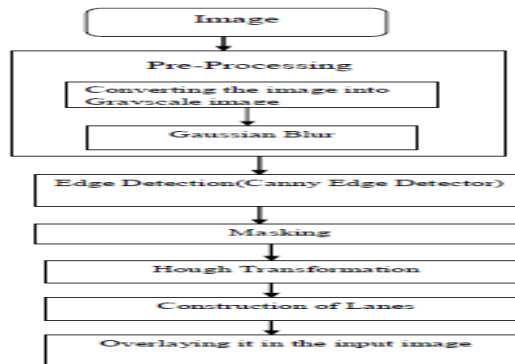


Figure 2.1 System architecture

III. SYSTEM MODULES

The modules include

- Pre-Processing
- Edge Detection
- Masking
- Detection and Construction of Lanes

3.1 Pre-processing

Pre-processing includes two steps, converting the image in to gray scale image and applying Gaussian blur. The original image given as input is shown below figure 3.1.1.

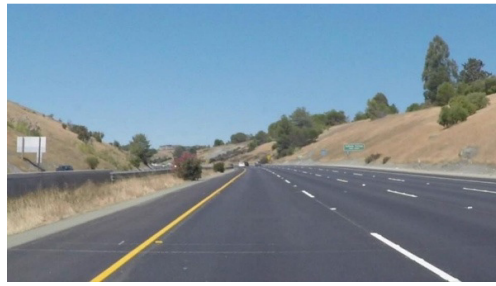


Figure 3.1.1 Input Image

The first process is to convert the image into a single channel image called as the grayscale image which is shown in below figure 3.1.2.

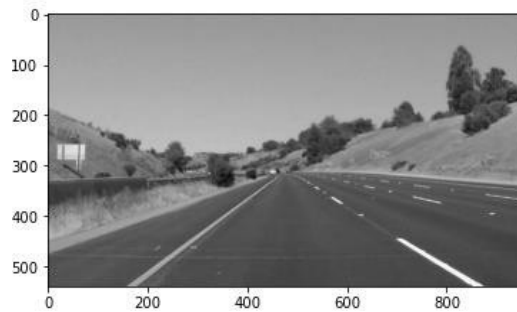


Figure 3.1.2 Gray Scale Image

3.2 Edge Detection

The next step is to detect the edges. This is done with the canny edge detection algorithm. The primary objective of the canny edge detection algorithm is to extract the structural information that is more useful than the other information in the given image. Canny edge detection is applied in various computer vision applications. The objective of the canny edge detection algorithm is as follows.

- The algorithm should be capable of finding almost all the edges in the given image
- Every identified image should be marked only once and the false edges should be avoided.
- The edge point detected from the operator should accurately localize on the center of the edge.

The steps involved in the canny edge detection algorithm are as follows.

- Smoothing of an image with Gaussian filter
- Finding the intensity gradients of the image
- Getting rid of spurious responses to edge detection by applying non-maximum suppression
- Determining the edges by applying double threshold
- Tracking the edges by hysteresis
- Removing the weak edges.

we can find edge gradient and direction for each pixel as follows:

$$\text{Edge_Gradient}(G) = \sqrt{G_x^2 + G_y^2}$$

$$\text{Angle}(\theta) = \tan^{-1}(G_y/G_x)$$

Gradient direction is always perpendicular to edges. It is rounded to one of four angles representing vertical, horizontal and two diagonal directions.

Non-maximum suppression

The image magnitude produced results in thick edges. Ideally, the final image should have thin edges. Thus, we must perform non maximum suppression to thin out the edges. Non maximum suppression works by finding the pixel with the maximum value in an edge. Non-maximum suppression is applied to find the locations with the sharpest change of intensity value. The algorithm for each pixel in the gradient image is:

1. Compare the edge strength of the current pixel with the edge strength of the pixel in the positive and negative gradient directions.
2. If the edge strength of the current pixel is the largest compared to the other pixels in the mask with the same direction (e.g., a pixel that is pointing in the y-direction will be compared to the pixel above and below it in the vertical axis), the value will be preserved. Otherwise, the value will be suppressed.

In some implementations, the algorithm categorizes the continuous gradient directions into a small set of discrete directions, and then moves a 3x3 filter over the output of the previous step (that is, the edge strength and gradient directions). At every pixel, it suppresses the edge strength of the center pixel (by setting its value to 0) if its magnitude is not greater than the magnitude of the two neighbors in the gradient direction.

The result obtained with the process of canny edge detection is given in the following figure 3.2.

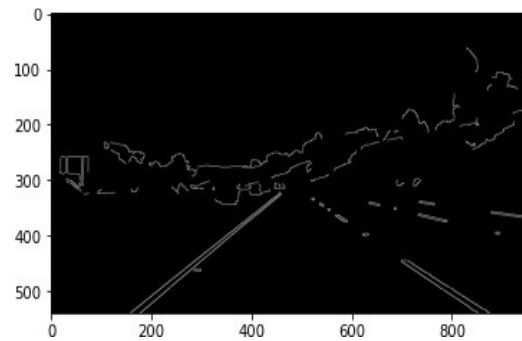


Figure 3.2 Canny Edge Detection

3.3 Masking

It can be observed from the above image that, in addition to the edges that represent the road lanes, there are also various other edges. Those edges are not region of concern for the road lane detection. The objective of masking is to remove those edges and extract only the edges that represent the lane of the roads. In order to extract the region of interest a polygon is formed that covers the region of interest. The rest of the pixels surrounding the polygon are removed. The resulting image is shown in the following figure 3.3.

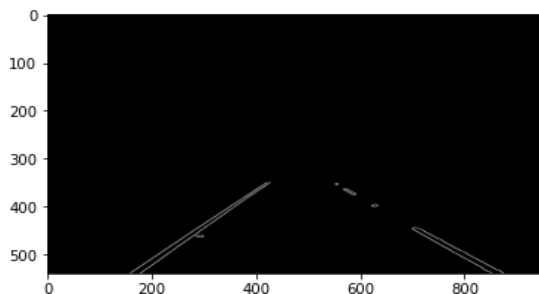


Figure 3.3 Masking

3.4 Construction of Lines

Hough transform is a method that is commonly used for detecting various shapes such as the lines, circles that occur in an image. It is used here in order to find the lines in the image which is masked in the previous step. The concept that is followed in detecting the lines is as follows.

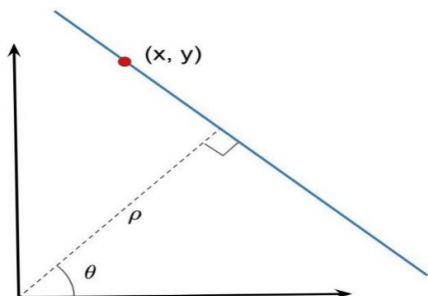


Figure 3.4.1 Polar coordinate representation of the system

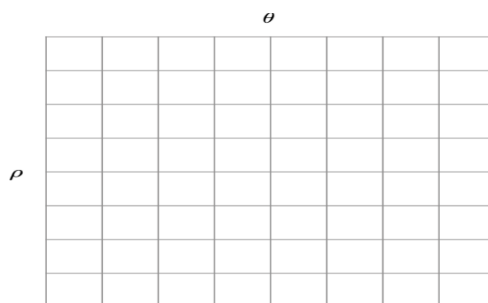


Figure 3.4.2 Accumulator used in Hough line transform

Once the accumulator is created, every cell of it represents the line. The idea is that if there is a visible line in the image, an edge detector should fire at the boundaries of the line. These edge pixels provide evidence for the presence of a line. The output of edge detection is an array of edge pixels $[(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)]$. Using this technique, we can find lines from the pixel outputs of the canny edge detection output. The result is shown in the following figure 3.4.4.

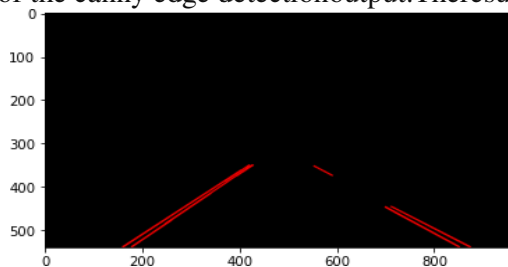


Figure 3.4.3 Hough transformation

The extrapolated line is overlaid on the original image and the resulting image is given in 3.4.5.

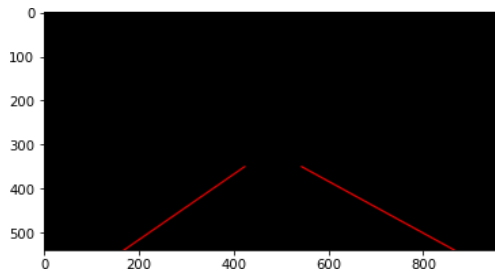


Figure 3.4.4 Construction of lanes



Figure 3.4.5 Overlaid image

IV. CONCLUSION

A model for detecting the Road lanes in the images is developed in this work. It uses the concept of Hough transform for lane detection. The steps that precede that include pre-processing such as converting the image to gray scale and edge detection with Canny edge detection mechanism.

REFERENCES

- [1] Hu, J., Xiong, S., Zha, J. et al. Lane Detection and Trajectory Tracking Control of Autonomous Vehicle Based on Model Predictive Control. *Int. J. Automot. Technol.* 21, 285–295 (2020).
- [2] Raja Muthalagu, Anudeep Sekhar Bolimera, V. Kalaichelvi, “Lane detection technique based on perspective transformation and histogram analysis for self-driving cars”, *Computers & Electrical Engineering*, Volume 85, 2020.
- [3] Liu, Yin-Bo & Zeng, Ming & Hao, Meng. (2020). Heatmap-based Vanishing Point boosts Lane Detection.
- [4] Munir, F., Azam, S., & Jeon, M. (2020). LDNet: End-to-End Lane Detection Approach using a Dynamic Vision Sensor.
- [5] Rossi, Alfa & Ahmed, Nadim & Salehin, Sultanus & Choudhury, Tashfique & Sarwar, Golam. (2020). Realtime Lane detection and Motion Planning in Raspberry Pi and Arduino for an Autonomous Vehicle Prototype.
- [6] Yinggang Li and Xiaobo Lu 2020. ,” Efficient Dense Spatial Pyramid Network for Lane Detection”, *Journal of Physics: Conference Series*, Volume 1575, 5th Annual International Conference on Information System and Artificial Intelligence [ISAI2020] 22-23 May 2020, Zhejiang, China.
- [7] Leonardo Cabrera Lo Bianco, Jorge Beltrán, Gerardo Fernández López, Fernando García, Abdulla Al-Kaff, “Joint semantic segmentation of road objects and lanes using 36 Convolutional Neural Networks”, *Robotics and Autonomous Systems*, Volume 133, 2020.
- [8] T. N. A. Nguyen, S. L. Phung and A. Bouzerdoum, "Hybrid Deep Learning-Gaussian Process Network for Pedestrian Lane Detection in Unstructured Scenes," in *IEEE Transactions on Neural Networks and Learning Systems*.
- [9] Jau Woei Perng, Ya Wen Hsu, Ya Zhu Yang, Chia Yen Chen, Tang Kai Yin, Development of an embedded road boundary detection system based on deep learning, *Image and Vision Computing*, Volume 100, 2020.
- [10] Satish Kumar Satti, K. Suganya Devi, Prasenjit Dhar, P. Srinivasan, “A machine learning approach for detecting and tracking road boundary lanes”, *ICT Express*, 2020.