Lane Detection Using Image Processing In Python

Ujjwal Sadani[1] Pallavi Sambhare[2]
[1]BE Student [2]Professor
[1,2]Department of Information Technology
[1,2]G H Raisoni College of Engineering Nagpur, India

Abstract — Some current technical advancements in road safety are being employed, and driver inattention makes it one of the primary causes of these accidents. The number of accidents is rising at an alarming pace these days. Technological advancements are required to lower the accident rate and assure their safety. Using lane detection devices, which identify lane boundaries in the road and warn cars when they pass poor lane markers, is one method. Several highly developed traffic systems that use technology must include lane keeping systems. Despite the fact that this is a challenging objective to accomplish given the various road conditions experienced, particularly while driving at night or during the day.

Keywords: Lane Detection, Image Processing, OpenCV, Hough Transform, Canny Edge Detection.

I. INTRODUCTION

The fast societal development has led to the rise of the automobile as a mode of mobility. On the congested road, there are a rising number of different kinds of automobiles. Intelligent car systems have made use of lane detection, a hot issue in the fields of machine learning and computer vision [1]. It is a new field that has applications in the business sector. The vehicle's location and trajectory with respect to the lane are correctly approximated by the lane detection system, which uses lane markers in a complex environment [2]. The lane departure warning system heavily relies on lane detection at the same time. The two fundamental components of lane detection are edge detection and line detection. In the process of lane detection, line detection is just as crucial as edge detection.

The Hough transform and convolution-based methods are the most often used lane line detectors. Lane detection is the technique of identifying lane markers on the road and delivering the positions to an intelligent system. Intelligent vehicles in intelligent transportation systems collaborate with the infrastructure to provide a safer environment and better traffic flow. The usage of a lane detecting system can range from simple techniques like displaying the location of the lanes on an external display to more sophisticated ones.to avoid collisions with other cars, complex tasks like instantaneous lane change prediction are required.

II. OVERALL DESCRIPTION

Using Python, one of the most popular computer languages for this purpose, and OpenCV, we have identified lane lines in photos for this project. The term "Open-Source Computer Vision" refers to a module that contains a variety of helpful tools for image analysis [3]. Several other programming languages, including Python and Java, are supported by OpenCV.

It can analyse pictures and movies to find items, people, or even human handwriting. Moreover, NumPy is used in this project for a few basic image processing methods. The lane detection module consists of two steps:

1) Gaussian Smoothing and Canny Edge Detection and
2) ROI selection and lane line identification using the Hough Transform.

III. GAUSSIAN SMOOTHING

In image processing, Gaussian blur is sometimes referred to as Gaussian smoothing. It is the result of applying a Gaussian function to blur a picture [4]. With the use of a smoothing filter, the picture is blurred to reduce noise. In graphics software, the Gaussian Smoothing effect is frequently used to minimise picture noise. The function that is utilised to calculate the average of the nearby points is expressed by the smoothing algorithm's "kernel," or basic building block. A kernel with the shape of a Gaussian, or a normal distribution curve, is referred to as a Gaussian kernel.

We will first convert the image to grayscale before applying the Gaussian filter so that it will be simpler for us to discern between the results of the clever edge detector

IV. CANNY EDGE DETECTION

Edge detection is a method of image processing that locates areas in a digital image where there are discontinuities—basically, sharp changes in the image brightness. The areas along the image's borders (or limits) are those where the brightness varies noticeably. The Canny edge [5] detector is perhaps the most popular and efficient technique, and because it initially applies a gaussian filter, it is more noise-resistant than other techniques. The OpenCV manual alone contains a vast amount of knowledge.

In conclusion, the algorithm will initially exclude pixels below the low threshold and find strong edge (strong gradient) pixels above the high threshold. The following step involves including pixels connected to strong edges whose values fall between the low threshold and high threshold.. The identified edges are traced out in white pixels in the output edges, while all other pixels are dark. Before to running Canny, we will additionally apply Gaussian smoothing, which essentially involves averaging out noise and erroneous gradients. While the OpenCV cv2.Canny() method already performs Gaussian smoothing internally, we explicitly add it here since doing more smoothing might provide a different result and cv2()_.Canny does not allow for changing this option

V. REGION MASKING

In an image, a region is a group of connected pixels that have similar properties. Because they may correlate to a items in a scene, regions are significant for image interpretation. Region Masking [6] is the process that is underneath manytypes of

image processing, including edge detection, motiondetection, and noise reduction. Masking a region is a view- specific graphic that can be used to hide certain and show certain elements in a view. Image editing is a non-destructive method. Considering that the front-facing camera that takes the picture is fixedly positioned such that the lane markings always display in the same general area of the picture [7]. To demonstrate the simplest scenario, we use a triangular mask here, but we could also use a quadrilateral or, in theory, any polygon.We would now use edge detection algorithm for detecting lane lines rather than relying on color. This wouldbe a more reliable and efficient model.

VI. HOUGH TRANSFORM

Finally, we utilise the Hough Transform to choose the lane lines in the edge detected picture. Straight lines can be found using the Hough Transform [8]. According to the instructions, an edge detection pre-processing is advisable before applying the transform. Essentially, a line may be located by counting the crossings of bends [9]. The adding curves As two lines meet, extra points are added to the line that is represented by that junction. We may define a threshold as the bare minimum intersections required to identify a line. The Hough Transform does this. It maintains track of the points on each curve where the image's points cross. It defines it to be a line if there are more intersections than a predetermined threshold.

VII. LANE DETECTION PIPELINE

With the use of a camera mounted in the car, the standard lane line identification method first captures a picture of the road. The image is then converted to grayscale to expedite processing. Second, noise in the image will make it challenging to spot the correct edge. As a result, filters like the Gaussian filter should be used to reduce noise. The edge detector then creates an edge picture by obtaining the edges using a clever filter with automated thresholding. After identifying the edges, the edged picture would next be submitted to the line detector. With an edge-detected picture, the Hough Transform is also utilised to find the necessary polygon. When the image has completed all of these intermediate phases, a pipeline is created employing these steps as functions with the source picture as the only parameter. After testing and validating the photos, we may proceed to using the same principle to lane videos. In this case, the lane detection pipeline processes each frame of the video as an image. This procedure is carried out for each frame in the video, and then all of the video's frames are combined to produce a single video file with the result for lane detection.

We have developed a lane detection pipeline now that the methods and features we must employ have been identified. Our input image will go through each of the necessary processes listed above using this function.

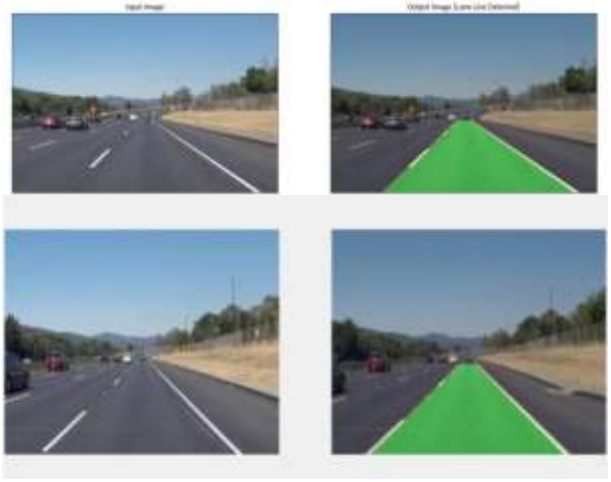
Our input image will go through each of the necessary processes listed above using this function

VIII. IMPLEMENTATION

Jupyter Notebook will be used for code development as well as to enter the video and picture files. The generated file will then be shown as a little Python GUI (Graphical User Interface) application. Tkinter and the Python moviepy package are used for this.

IX. RESULTS AND OUTPUT

Figure 1 displays the result as a lane detected picture on our test photographs, and Figure 2 displays the result after running our method on the test video.



X. APPLICATIONS

With a driverless automobile that isn't totally automated, it can be utilised for lane detection to help the driver.

A. Advantages

- This does not need complicated and advanced machine learning, deep learning, or neural networks; rather, it helps both human drivers and autonomous vehicles to drive safely. instead uses straightforward and understandable image processing and computer vision algorithms.
- Because it makes use of free and open-source software like Python, Jupyter, and OpenCV, its development costs are low.

B. Disadvantages

- Uniformity of the markings present on the lanes isan important factor in order to minimize confusion

XI. CONCLUSION

Lane detecting systems are likely to be increasingly in demand as a result of all the advancements being made in the autonomous car sector. We attempted to properly propose the lane detecting code with this project without needing to develop lengthy machine learning or deep learning-based codes. The robustness and flexibility of the detection findings are improved by the Hough transform. The application scope of our study will be expanded by the availability of quality

datasets for photographs of rural roads. One of the downsides, noise, may be reduced with the use of effective noise minimization algorithms. Although we haven't utilised it, object detection and tracking might be used to add more characteristics to the algorithm.

XII. FUTURE SCOPE

The implementation of lane detection utilising a guided image filtering technique, such the Gabor filter, might be studied in the future to increase the algorithm's resilience even more. Another issue is taking care of fog removal. In the near future, it will be possible to adapt the Hough Transformation to evaluate both curved and straight roads. The application scope of our study will be expanded by the availability of quality datasets for photographs of rural roads. One of the negative aspects is the noise level. To enhance the outcomes in various environmental situations, such as sunny, foggy, rainy, etc., several procedures should be done. Moreover, the system may be enhanced to include lane markings, road sign recognition, and lane identification on rural or less used roads in addition to lane lines.

REFERENCES

[1] Shopa, P., N. Sumetha and P.S.K Pathra. "Traffic sign detection and recognition using OpenCV", International Conference on Information Communication and Embedded Systems (ICICES2014), 2014
[2] "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixelwise labelling," arXiv preprint arXiv:1505.07293, 2015.
[3] J. Long, E. Shelhamer, and T. Darrell, "LANE DETECTION TECHNIQUES" – A Review." in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.
[4] M.Cordts,M. Omran,S.Ramos, T.Rehfeld,M. Enzweiler, R.Benenson,U. Franke,S.Roth, and B.Schiele, "The cityscapes dataset for semantic urban scene understanding," in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[5] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P.H Torr, "A Layered Approach to Robust Lane Detection at Night." 2015, pp. 1529–1537.
[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," arXiv preprint arXiv:1512.03385, 2015.

