Road Lane Segmentation

A project report
submitted by

Atharav Yadav(IIT2022129)
Nirbhay Paliwal(IIT2022156)
Aditya Bhardwaj(IIT2022157)
Shivam Kumar(IIT2022158)
Pratik Sarvaiya(IIT2022205)

under the supervision of
Dr. Navjot Singh

In partial fulfillment of
the requirements for the degree of
Bachelor of Technology (Information Technology)

To the
Department of Information Technology
Indian Institute of Information Technology, Allahabad

Contents

Chapter 1

Introduction

Image processing is a field of computer science which involves performing the operations on images for enhancement, noise reduction, restoration of the image and extract useful information from the image. It includes various techniques like sharpening, filtering, edge detection, point detection and morphological operation to manipulate the value of the pixels.

Segmentation is technique used in image processing which partitions the image into group various groups of pixels which are called image segments. Using this technique, image is divided into meaningful regions for easier analysis. Segmentation isolates objects or boundaries within an image which helps to focus on specific areas of interest. Here are some of the examples where segmentation is used : detecting and isolating lane marking on the roads, separating snow and non-snow region, medical image segmentation - separating healthy tissues from cancerous ones, isolating buildings from surrounding greenery, separating water bodies in satellite or drone image.

Lane segmentation of the road is a subset where image enhancement techniques are used to identify and extract the lane marking of interest. It employs image processing techniques such as edge detection, image filtering and thresholding to segment the road lanes from its background. This task is of utmost importance for self-driving vehicles and Advanced Driver Assistance Systems (ADAS) In order that the vehicles able to stay in the lane thus improving safety and navigation.

The reason road lane segmentation is vital is that it can detect road lanes in real time and assist in the safe operation of driverless vehicles. It is also applicable in self-driving vehicle systems, ADAS lane-keeping and lane departure alert systems, and road monitoring systems for the improvement of safety and efficient movement of vehicles along the road.

For Road Lane Segmentation, we are using two approaches :

- **Canny Edge Detection** and **Hough Transformation**
- **Thresholding** and **Morphological Operations**

Chapter 2

Approach 1 : Canny Edge detection + Hough Tranformation

Libraries Used

- python
- Open CV
- MatPlotlib
- numpy

Steps involved in the first approach are :-

2.1 Grayscale conversion

First of all images are converted from RGB to grayscale. This step helps in detecting edges in upcoming steps and reducing the underlying colour complexity of image.

2.2 Gaussian blur

Now to reduce the noise and smoothening Gaussian blurring is done, helps get rid of unnecessary details and objects. Kernel size is also needed to adjust the amount of blurring, if kernel size increases blurring increases.

2.3 Canny Edge Detection

This steps helps in detecting the edges of the images after this areas with higher change in intensity gets highlighted.Requires two parameter

- Lower Threshold , High Threshold :- All the edge gradient below the Lower Threshold are not considered as edge and edge gradient exceeding the High Threshold are considered as a strong edge for canny edge detection.

2.4 Region of Interest

After the canny edge detection image only has edges but it still has some unnecessary objects like tree and other objects. These can be removed by setting a region of interest, it enables program to focus on selected area which is typically the bottom part of image contained in trapezoid shape. We can set the vertices of according to our need and purpose of image processing. Enlarging the area of trapezoid enables us to segment far located roads but at the same time this leads to detection of other objects like trees, railings etc.

2.5 Hough Lines

The image with edges goes through hough transformation this detects the lines in image it marks the points which can be potentially be part of straight line, the points may be broken or may have certain noise in them.

2.5.1 Equation

The equation for ρ in terms of x and y is given by: [3], ...

$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta)$$

- ρ (rho) is The perpendicular distance from the origin to the line.
- θ (theta) is The angle between the perpendicular line and the horizontal axis (measured in radians).

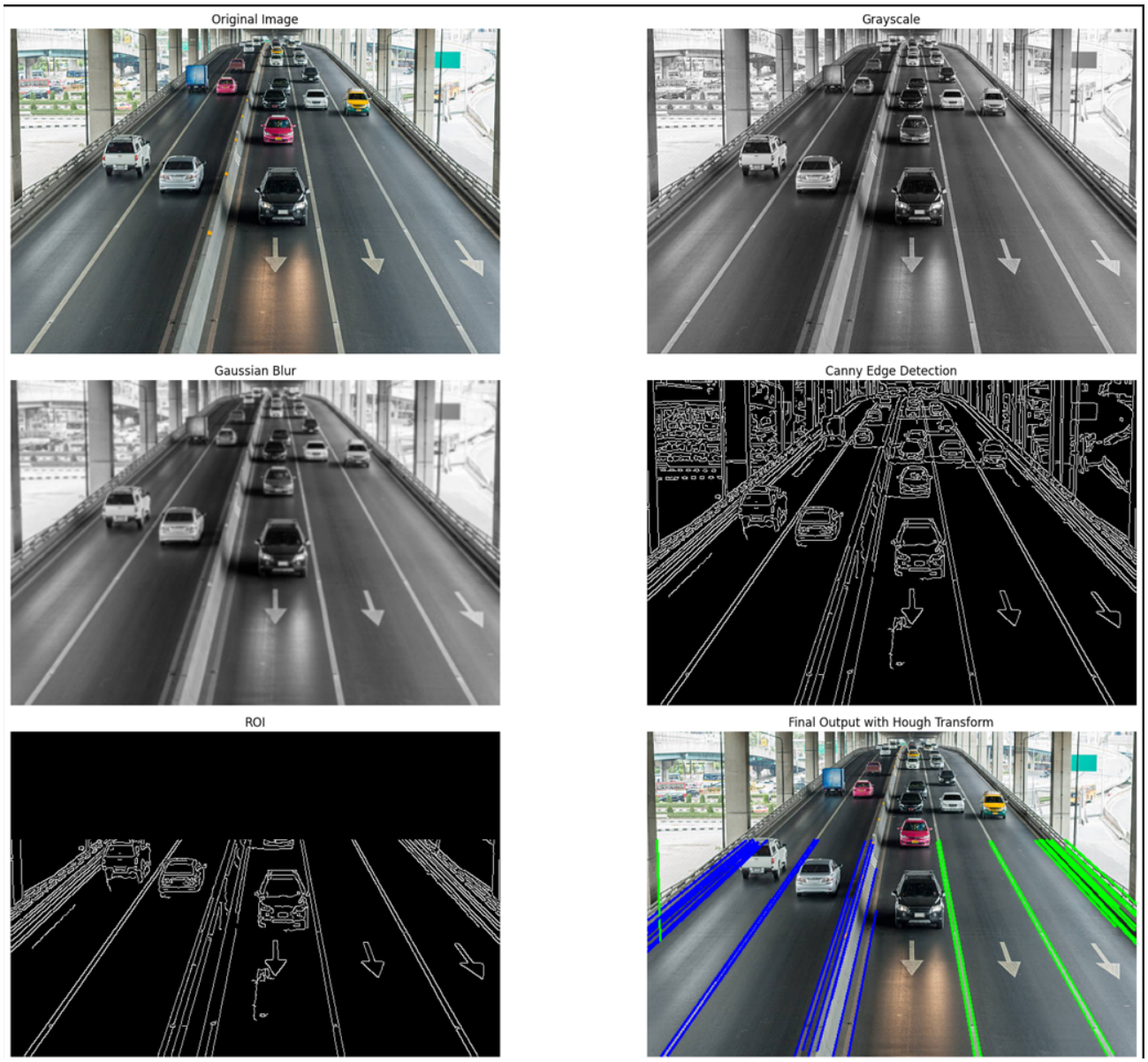
The following parameter are used in hough transformation. [2]

- rho - Distance resolution in pixels.
- theta - Angle resolution in radians.
- threshold - Minimum number of votes for valid line.
- minLenLine - Minimum allowed length of line.
- maxLineGap - Maximum allowed gap between line for joining them.

2.6 Drawlanes

In final step of this method we are segregating the left lanes and right lanes by giving each a colour, lines with positive slopes is given one colour and the lines with the negative slopes is given other colour, To prevent noisy horizontal lines from getting into the image a threshold for the slope is also kept.

APPROACH 1



Figure[4]

Chapter 3

Approach 2 : Thresholding + Morphological operation

Libraries used

- python
- Open CV
- MatPlotlib
- numpy

3.1 Grayscale conversion

Firstly the image is converted to grayscale which helps to focus on only the intensity changes in the image.

3.2 Power law Transformation

To make road look more darker, power law transformation is used, the parameter gamma used is greater than 1 making the already darker intensity (grayscale/255) to look even darker.

3.3 Gaussian bluring

As in approach:1 in this approach also, it is used to get rid of noise and unwanted Gaussian blur is applied.

3.4 Binary Thresholding

Binary thresholding converts white areas to white and others to black, with different thresholds required based on lighting conditions.

3.5 Region of interest

In this approach, we also create a region of interest, just like in approach 1.

3.6 Dilation

Dilation is a morphological operation that increases the size of objects in obtained binary image. It works by adding pixels to the edges of an object, hence expanding its area. This eventually helps in connecting on those broken lines.

Working : In each a matrix of some size say 5x5 is slides to all the pixels presents in image if any pixel in the image is white then the central pixel is set to white.

It takes iteration and kernel size as the parameter, increasing the number of iteration makes white segments even thicker making broken lines to join.

3.7 Contour Colouring

Contour is the curve which connects the boundary points of any object present in the image. In this step after joining the points we colour the region inside the curve.

APPROACH 2

Figure[4]

Chapter 4

Comparison and Conclusion

4.1 F1 Score[1]

The F1 score is a measure of a model's accuracy that considers both the precision and the recall. It is defined as the harmonic mean of precision and recall.

$$F1 = 2 \cdot \frac{P \cdot R}{P + R}$$

Where:

$$Precision(R) = \frac{TP}{TP + FP}$$

$$Recall(R) = \frac{TP}{TP + FN}$$

For our experiments, we calculate the F1 score for both approaches to determine which method performs better in segmenting the road lanes.

4.2 Comparison

In this section, we will compare the two approaches based on performance metrics, including accuracy, precision, recall, and F1 score.

Table 4.1: Comparison of Performance Metrics

Approach	**Precision**	**Recall**	**F1 Score**
Canny + Hough	60%	62%	61%
Thresholding + Morphological	68%	71%	69%

4.3 Conclusion

In conclusion, both the image has its own advantages and disadvantages in some conditions using Canny Edge Detection will be effective than Thresholding method and vice versa depending on dataset.

Canny Edge method may be better in real-time application as it is prone to noise and more generalised method, depending less on lighting condition, but on the other

hand gives distorted lines in case of road turns.While the thresholding method is good for detecting the turn, it requires tuning the lighting conditions.

Bibliography

[1] Jason Brownlee. *Machine Learning Mastery with Python*. Machine Learning Mastery, 2016.

[2] GeeksforGeeks. *Hough Transform in Computer Vision*. GeeksforGeeks. Available online: `https://www.geeksforgeeks.org/hough-transform-in-computer-vision/` (Accessed: 2024-09-27). 2024.

[3] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Pearson, 2018.

[4] iStock. *Traffic on a Multilane Highway*. Accessed: September 27, 2024. 2024. URL: `https://www.istockphoto.com/photo/traffic-on-a-multilane-highway-gm174793271-22192898`.