ENPM673 - Perception for Autonomous Robots

Project 2

Lane Detection and Turn Prediction for self-driving cars

Yash Savle, Pranav Jain, Rodrigo Perez-Vicente

March 11,2020

Introduction

One of the essential algorithms used for self-driving cars is lane detection. In this project we aim to design a Lane detection Algorithm for car, which also provides turn prediction.

Problem 1

The objective of this problem was to enhance the quality of the night vision videos recorded during night time. The improvement of the visibility in video was implemented by processing each frame of the video individually. We have adjusted the brightness of he image and also increased the contrast value to improve the visibility. Then we performed gamma correction on the image and results after the processing are as follows:

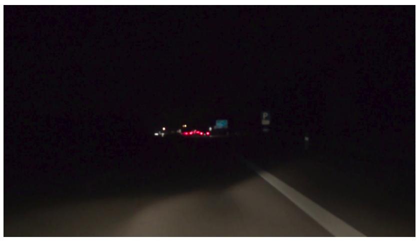

Figure 1: Input Video

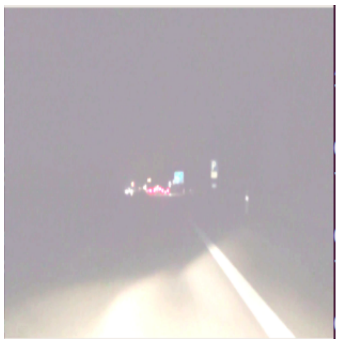

Figure 2: Result after Processing the video

Problem 2

Problem 2 Focuses on Implementation of Lane detection and Turn Prediction. We have implemented the image processing pipeline to obtain the desired output as following.

Pre-processing

Frames are individually extracted from the video and processing is carried out on them sequentially.Then the frame is cropped down to extract the region of interest i.e the bottom half of the image.

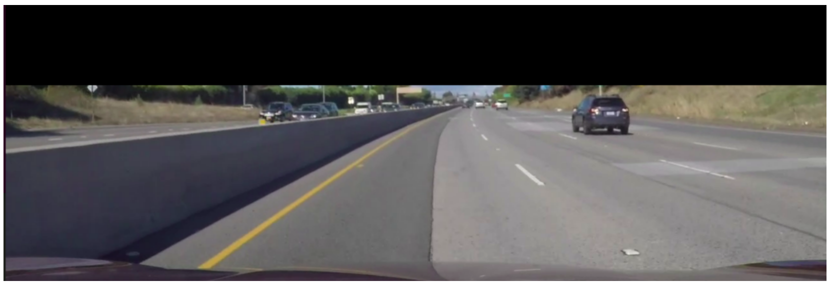

Figure 3: Cropping out sky part of the frame

As we have to detect the yellow and white lines from the image we must convert it to HSV and create 2 filters for white an yellow. To reduce the high freuency noise we perform Gaussian blurring on the gray scale image.

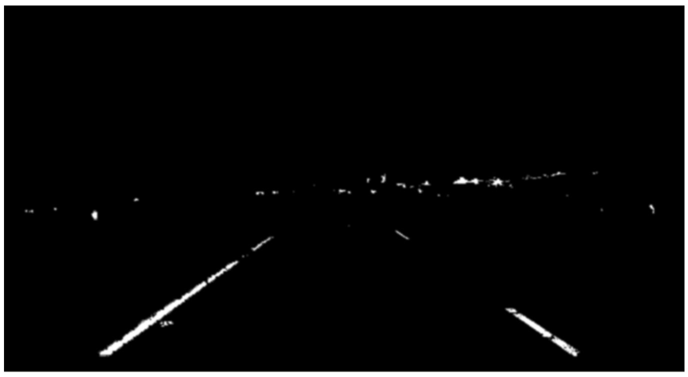

Figure 4: Gaussian Blurring

Next we apply our mask on the above image and to detect the lanes in white and yellow. Then we perform erosion on he image to de-noise it.

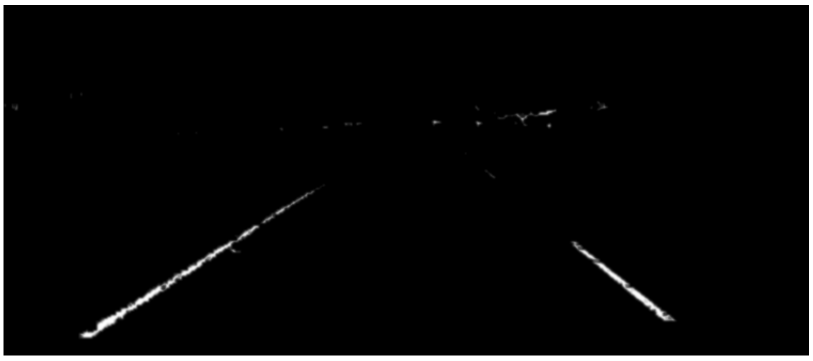

Figure 5: line detection

Canny Edge detection

In order detect the lanes completely we perform canny edge detection on the image obtained from above prepossessing.

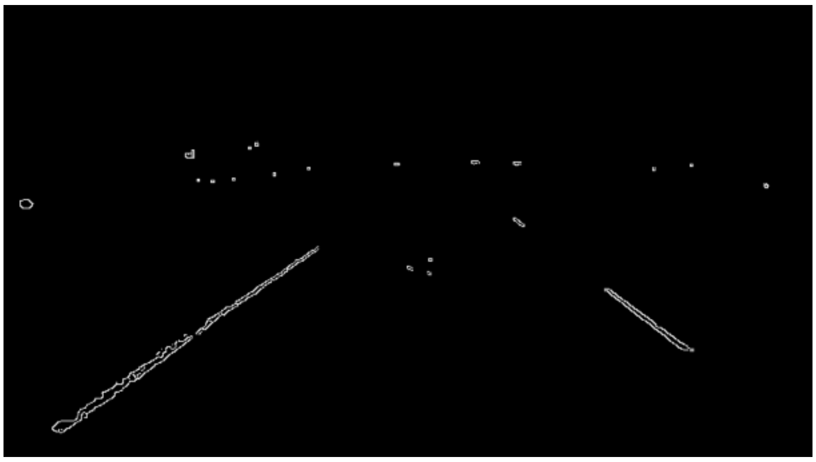

Figure 6: Canny Edge Detection

Hough Transform

The basic concept of Hough Transform is to check whether a given set of points are co-linear To find the location of lanes from the edge detected image we perform hough transform which provides us multiple lines.Then we determine the highest line and extrapolate the lines to form one single lane. The Hough lines that best match the lane are shown below:

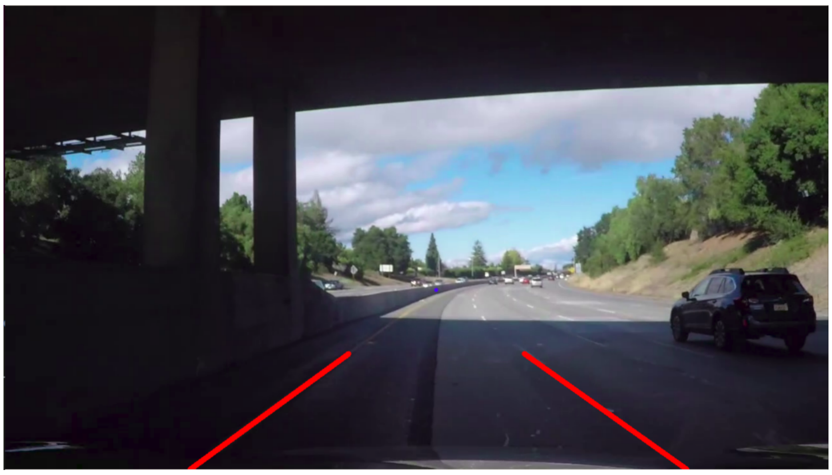

Figure 7: Single Hough lanes

To compute the Hough algorithm we called the in-build OpenCV function cv2.HoughLines(). The basic principle of Hough transform is that line equations, $y = mx + n$, can be transformed into polar coordinates $r = x\cos(\theta) + y\sin(\theta)$. As shown in the image bellow:

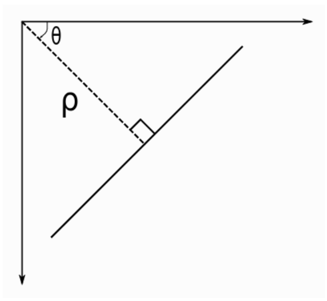

Figure 8: Polar Coordinates

It is also known that a point (x, y) can be represented by infinite (r, θ) points, which represent all the lines in which that point can be on. Then the Hough transform will obtain the (r, θ) curves that represent each point (x, y) and then if a considerably amount of curves intersect with each other the (x, y) points in those (r, θ) curves are considered to be in the same line.

The cv2.HoughLines() function goes through every point and stores them in different data structures depending on if their (r, θ) curves intersect or not. The data structures with most values are the ones to be considered as lines. In this function there are also four inputs to take into account to identify if a group of points are line candidates or not:

- ρ is the resolution in pixels.
- θ is the resolution in radians.
- **threshold** is the resolution in radians.
- **min**, minimum number of points in a line.
- **max**, maximum distance between points inside a line.

Turn Prediction

Implementation of Turn Prediction was carried out by comparing the centre of the lane along with the centre of car. Based on the displacement of the centre from the centre of car the turn direction is determined i.e if the centre of lane moves towards left side the car must steer left, and similarly for the right side.

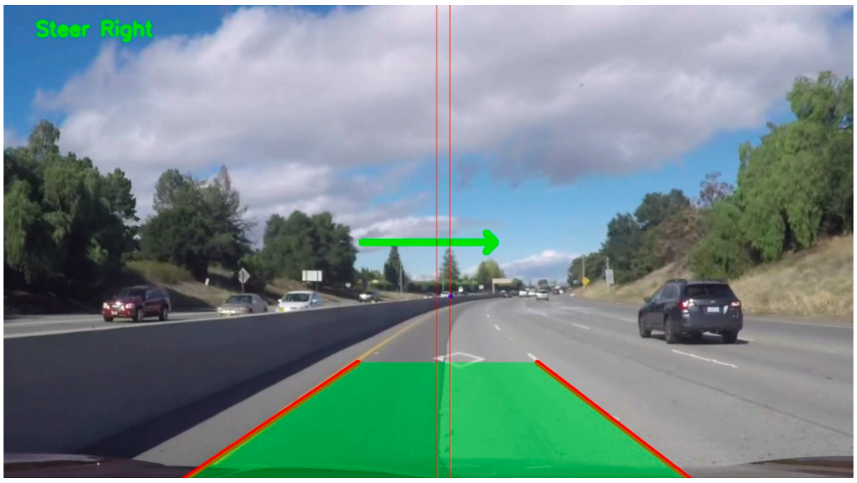

Figure 9: Turn direction and center lane

Conclusion

The pipeline designed for lane detection is successfully detecting the lanes as well as predicting turns. Same pipeline will also work for different videos and it will also take into account the lane changes.

Videos

The output videos can be seen in the link bellow. For Problem 2 the Output2 was computed by storing all the images from data1 into a list and then converting the frames to a video with an specific frame rate.Then the processing pipeline for the video will be the same as for Output1 Click for videos