Project : Finding Lane Lines

24.11.2018

Submitted By:

Lalu Prasad Lenka

Overview

The project aimed at creating a pipeline for line identification take road images from a video as input and return an annotated video stream as output.

Goals

1. Make a pipeline that finds lane lines on the road
2. Reflect on your work in a written report

Reflection

I. Pipeline Description

My pipeline consists of five steps.

1. First I resized the given image to 540 x 960 to make sure pipeline gets same input image size and then converted the color image to gray.

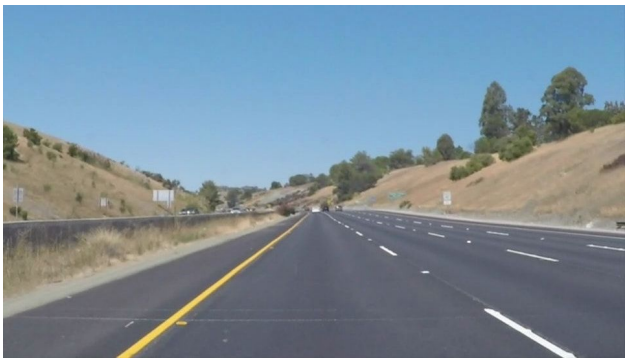

Fig.1- Original Color Image

Fig.2- Grayscale Image

2. The second step is to smooth the image using gaussian_blur helper function.

Fig.3 - Blurred Image

3. The third step is to detect edges in the image using canny edge detection where i used low_thresold as 80 and high_thresold as 160 since it was advised that the ratio should be 1:2.

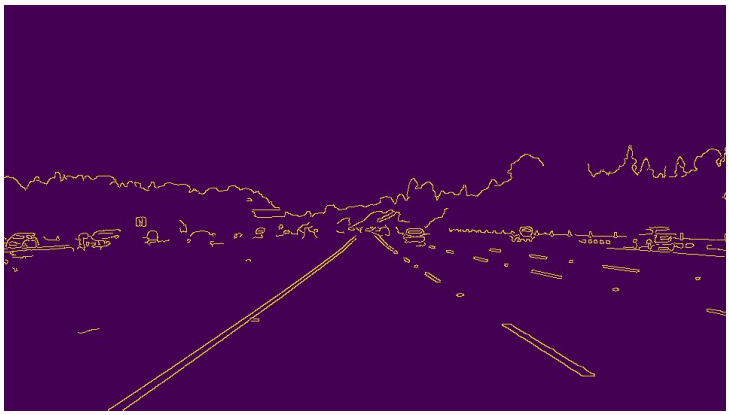

Fig.4 - Canny edge detected image

4. The fourth step is to create a masked image to find only the region of interest and do hough transform on that region.

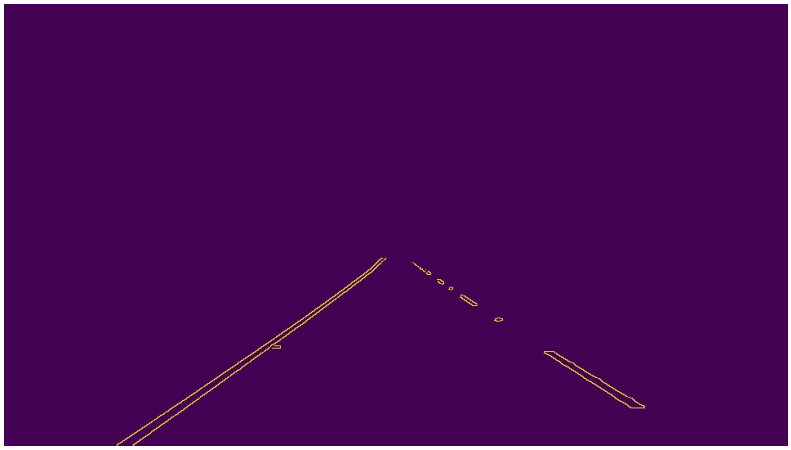

Fig.5 - Masked image, contains region of interest

5. The final step is to find the lanes using hough transform and draw lines on original image.

Fig. 6- Lane detected in original image.

The parameters used for hough transform are as follows:

- rho = 1 # distance resolution in pixels of the Hough grid
- theta = (np.pi/180) # angular resolution in radians of the Hough grid
- threshold = 15 # minimum number of votes (intersections in Hough grid cell)
- min_line_len = 60 #minimum number of pixels making up a line
- max_line_gap = 30 # maximum gap in pixels between connectable line segments

In order to draw a single line on the left and right lanes, I modified the draw_lines() function by averaging the slope and coordinates which helps to average/extrapolate the line segments you detect to map out the full extent of the lane.

The changes in draw_lines() helped detecting lanes in videos too.

II. Potential shortcomings with current pipeline

- It's definitely failing in optional challenge where road are curved and have dividers.
- It might fail in night or foggy conditions
- Region of interest depends on the position camera over car and in traffic situation the region of interest can be compromised, which isn't robust.
- Incase of shadows of trees and build the program might fail.

III. Suggest possible improvements to your pipeline

- One possible improvement could be to detect curvy lanes, multiple lanes and lanes with a lot of discontinuity.
- Other improvement could to tune the algorithm so that it can not only detect curves with discontinuity, noise, desired slope but also it can perform shadow and illumination correction and detect/ differentiate between different curves.
- Noise filter, shadow and illumination correction can help to detect lanes even when divider or tree shadow is there.
- Clustering of hough lines using weighted centroids and angle filtering technique can help to find slope of tangent of desired curve in curvy roads.
- Continuous frame feedback can help in detecting discontinuous curve.