SDCND-Term1

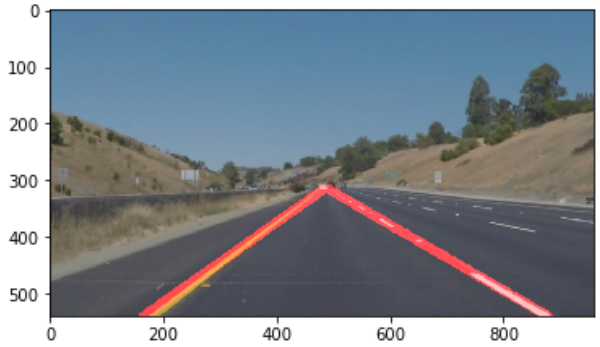
Finding Lane Lines on the Road

Writeup

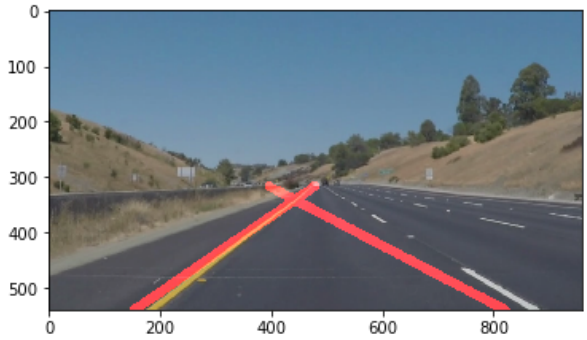
Reflection

1. Description

My pipeline consists of 6 steps that I learned in the supporting video lectures. Firstly I convert the input image to grayscale, then I apply gaussian filter despite the canny function does it internally. I noticed, without this extra filter the results are quite different. Below is an example with and without the gaussian filter.



Final outcome with Gaussian filter

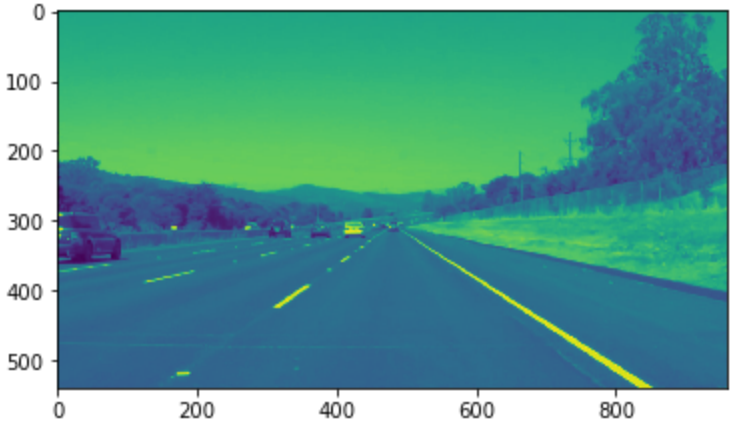


Final outcome without Gaussian filter

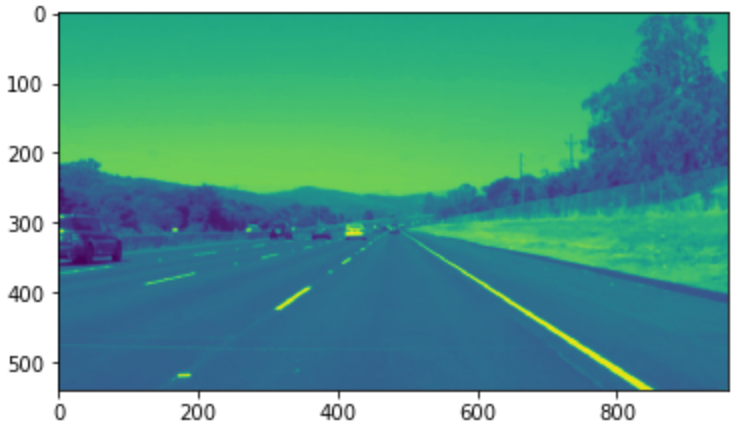
Then to find edges, the Canny function is called on the gaussian image. Next the unwanted portions of the image are masked by selecting a polygon of the interested region only. The fifth step is a tricky one mainly because it calls the hough_lines() function which in turn calls the **draw_line()** function. The hough_lines() function gives all the line segments and draw_line() function does all the interesting work for this assignment. Finally the lines are overlayed onto the original image using the weighted_img() function.

Below are 6 images recorded at each of the 6 steps.

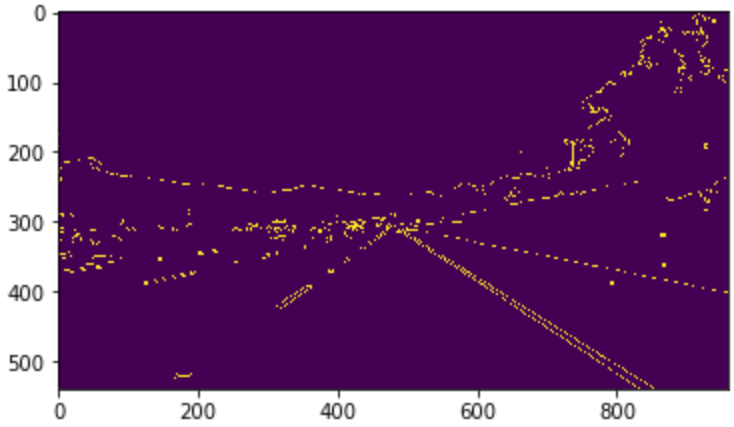
Convert to grayscale



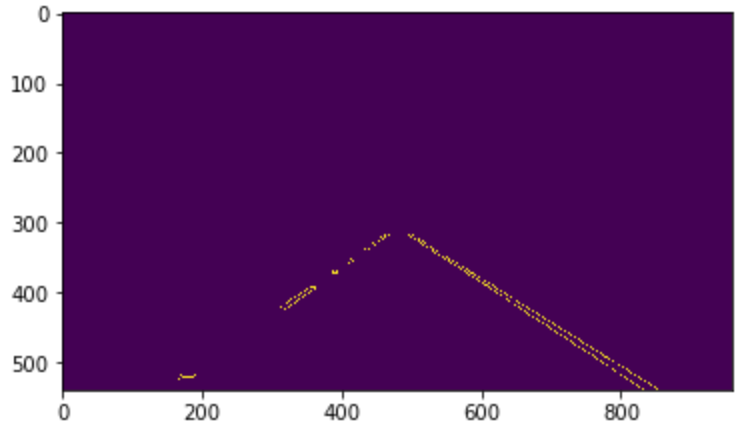
Apply Gaussian filter



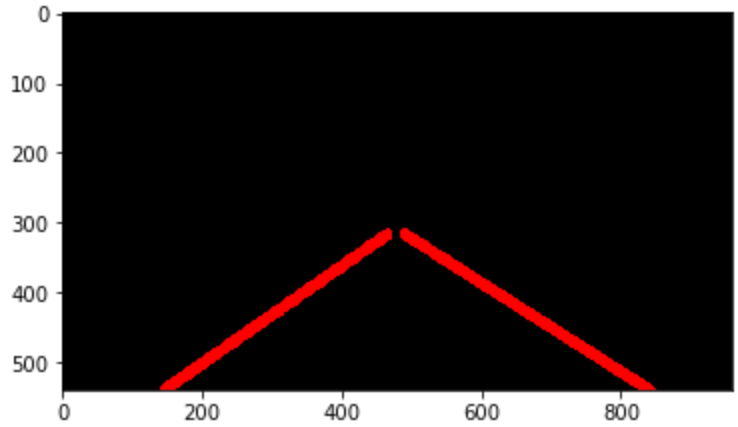
Apply Canny algorithm



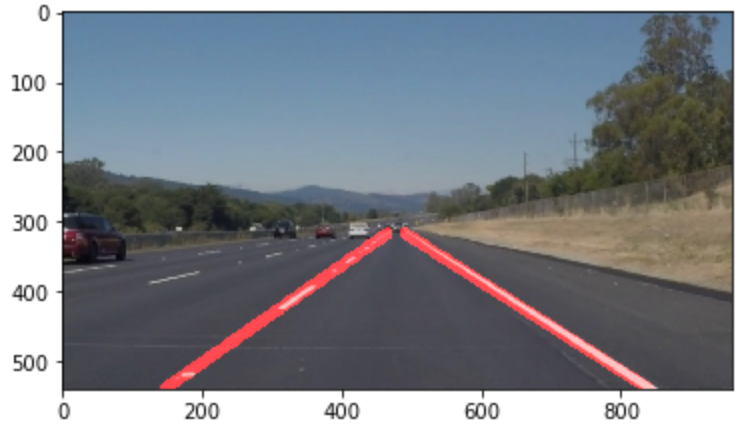
Mask unwanted region



Find lines using Hough's method and run through the draw_lines() function.



Final output, lines overlaid on the original image.



The draw_line() function: The first step is to identify if a line segment belongs to left or right lane. This is simply by calculating the slope of each line. Assuming the line segments are in an almost straight line, (i.e not an image of a turning road), a positive slope means it belongs to left line and a negative slope means it belongs to right line.

Then to draw a left line, the line segments in left list are averaged and the top most x and y and the bottom x coordinates are calculated. A left line is then drawn using the cv2.line() function. Similarly to draw a right line, similar values are calculated from the right list and the right line is drawn.

2. Potential shortcomings

Does not work for an image or a video of a turning road, with turning lane lines.

3. Possible improvements

To fix this problem, a further improvement to the draw_line() function is required. This function categorizes the line segments correctly only if the input image is of a straight road. Basically for a right turning road image, the slope of both right and left lines are negative and for a left turning road image the slope is positive. This needs to be accounted and a new technique is required to classify the segments.

A second improvement required to solve this problem is to fit the segments in an arc instead of a straight line.