Write up Template

You can use this file as a template for your write up if you want to submit it as a markdown file. But feel free to use some other method and submit a pdf if you prefer.

Finding Lane Lines on the Road

The goals / steps of this project are the following:

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

Reflection:

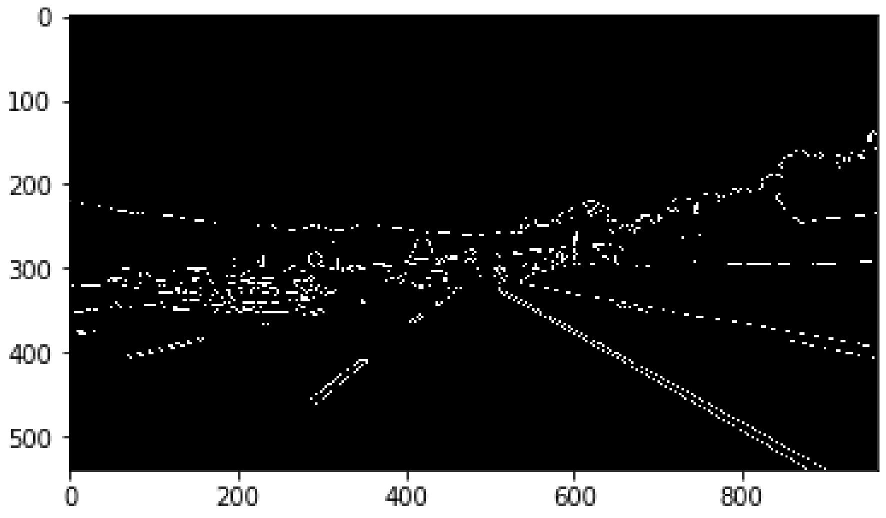
1. Describe your pipeline. As part of the description, explain how you modified the draw_lines() function.

My pipeline consisted of the following steps:

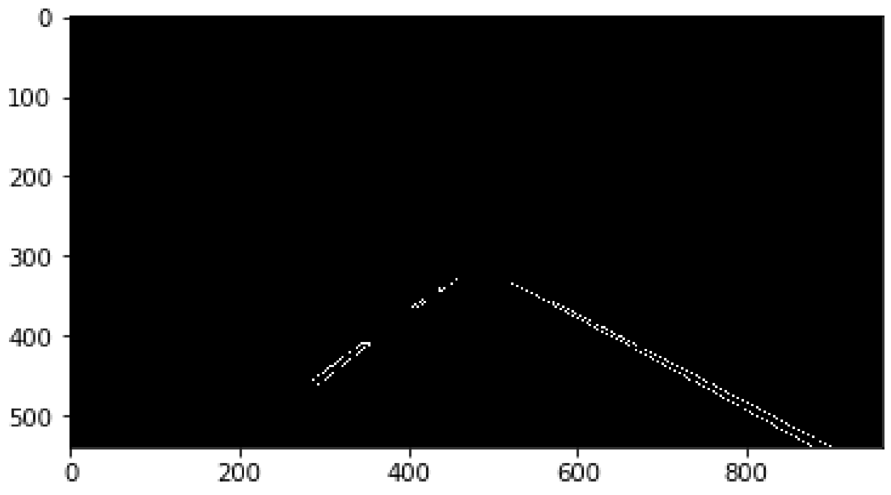
1. Conversion of the RGB image to Grayscale:

2. Applying Gaussian blur to the image with kernel size of 5:

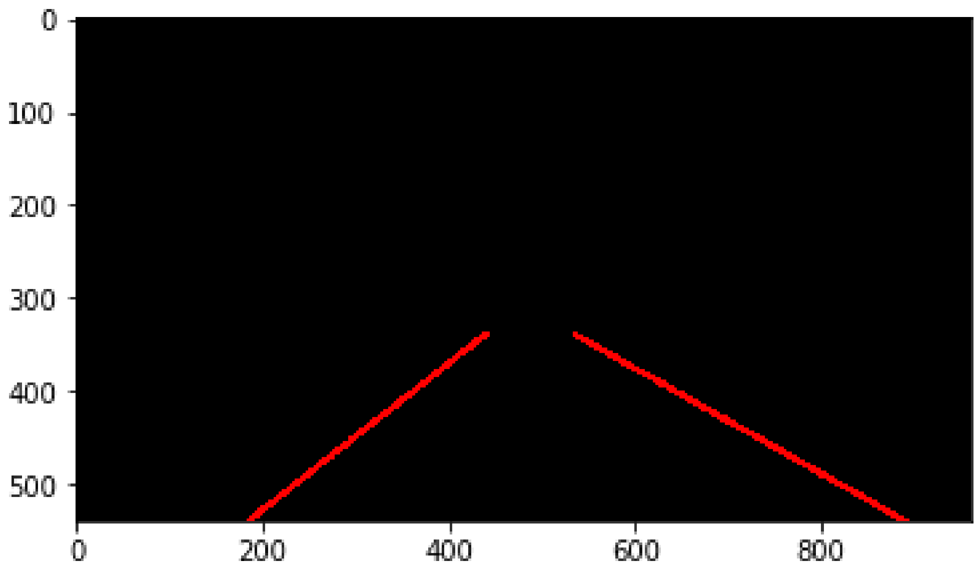
3. Applying Canny edge detection (lower threshold = 50, higher threshold =150)



4. Masking the image to only consider a certain area of interest where we expect lane lines



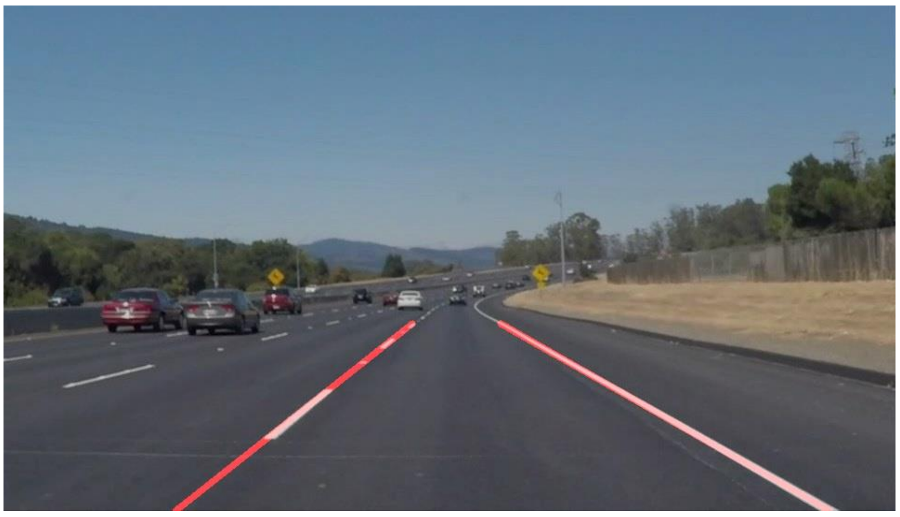
5. Running the Hough transform (rho = 2, theta = np.pi/180, threshold = 15, min line len = 30, max line gap = 10) to identify lines and converting those lines into straight lines.



In order to draw a straight single line on both left and right lanes, I modified the draw_lines() function which works as follows:

a. Separate the right and left line based on their slope
b. Average the line ending x, y coordinates for each side
c. Calculate the slope and intercept of each average line
d. Using the slope and intercept, extend the lines to the bottom and apex of the lane
e. Plot both the left and right lines.

I focussed on optimizing the hough parameters with the idea to get long continuous lines. That worked well for the test images as depicted below in the figure.



However, the above parameters did not work well for the video inputs, which required change in the parameters again and finally could get a good video output with lane lines detected.

2. Identify potential shortcomings with your current pipeline

Some of the potential shortcoming are discussed below..

1. While this pipeline works best for test images, there are some issues while using it with videos. In both of the videos, it works well, but you can see that some of the lines are jumpy and not in line with the lane.

2. For the optional challenge video, the pipeline does not work properly, would need to work out on the same. Especially the shade on the road caused troubles but as well the lane detection seemed unstable and the lanes were jumping around. Once I figure out, will resubmit.

3. I think in low lighting scenarios, the pipeline may not work as expected.

3. Suggest possible improvements to your pipeline

One possible improvement to the pipeline for video input would be have a moving average of the lines created from each image, so that the distortion in the lane lines detection is minimized.

Another potential improvement could be to use advanced detection techniques of white and yellow lanes using HSL, before converting the image to grayscale. Not sure on this.

The optimal parameters for Gaussian blur, the hough transform, etc , might be different based on the input video. I think we might be able to use some Machine Language models to based on the quality of the image,etc.