

Writeup Project - 1

Lane Finding

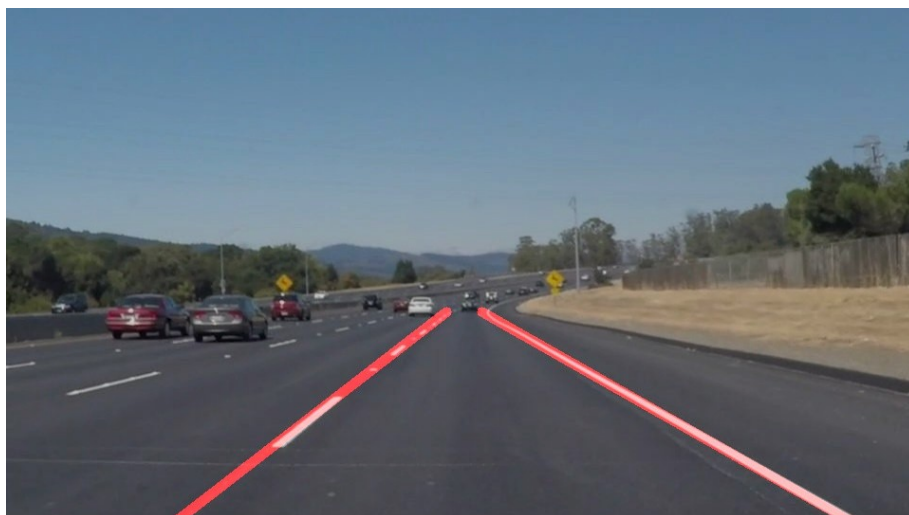
Kushagra Pundeer

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

My pipeline consisted of 5 steps. First, I converted the images to grayscale, then I applied Gaussian Blur on the grayscale image. After that Canny Edge detection method is implemented. Once I got the Canny edges, I applied the Hough Transforms and then used the weighted image function to display the plots. In the plots broken lines are seen. Now I went back into and modified the draw_lines() function to extrapolate and average out the left and the right lines to cover the whole lane with the lane lines.

In the draw_line() function, I defined an array to store the left and the right lines. Magnitude of slope was chosen to be between 0.5 and 1.0 to get rid of any large or small slopes. The sign of the slope would classify the lines as left or right. Once I got the left and the right lines and their slopes I removed any outliers. After this average left and average right lines were evaluated. These are then extrapolated and plotted. Top and bottom limits 320 and 540 were chosen approximated and fine tuned for accuracy.

The pipeline is then iterated among all test images and the results were verified.





2. Identify potential shortcomings with your current pipeline

It works well with the two videos provided. It does not always work well with the optional video. The issue is that the speed in the optional video or the frames per second are higher, which means that the pipeline should be able to adjust to the quick changes to the slopes. Also, the roads are much more curvier at times which modifies out top y position used to extrapolate lines (320) .

3. Suggest possible improvements to your pipeline

Pipeline for the video can be refined by adding more frames per second in the video for the optional video. Using polyfit function (curve fitting) or adjusting the top y position used for extrapolation.