

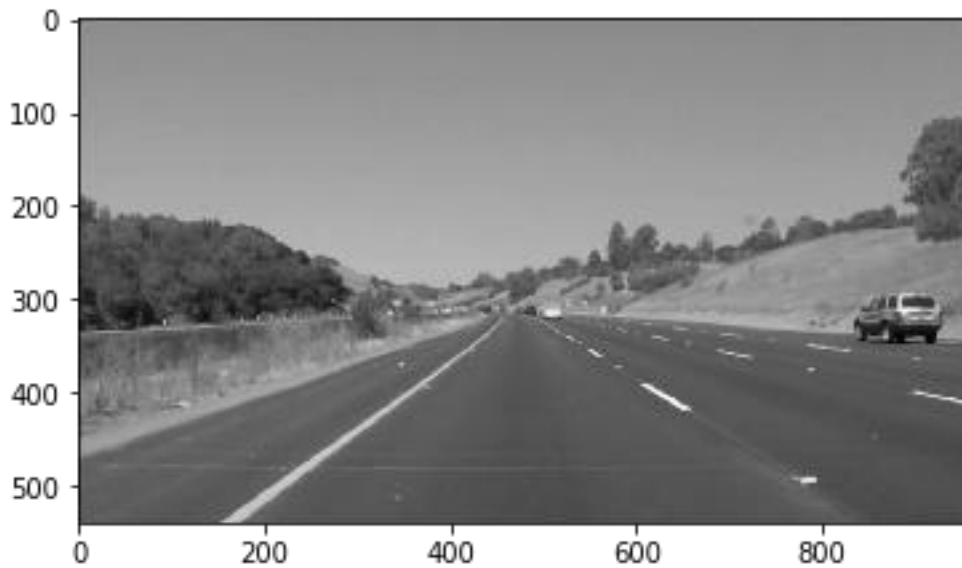
Finding Lane Lines on the Road

The goals / steps of this project are the following:

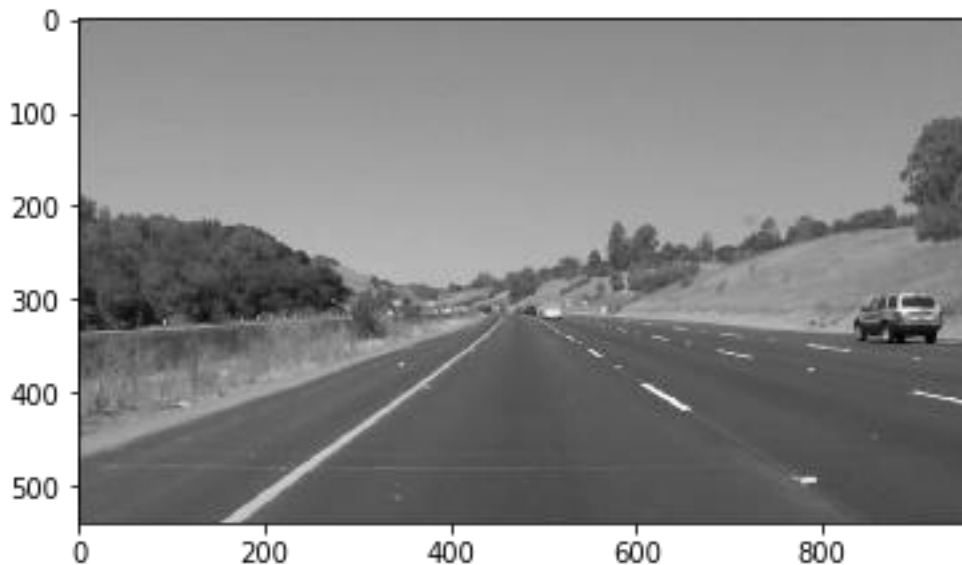
- Make a pipeline that finds lane lines on the road
- Reflect my work in a written report

Reflection

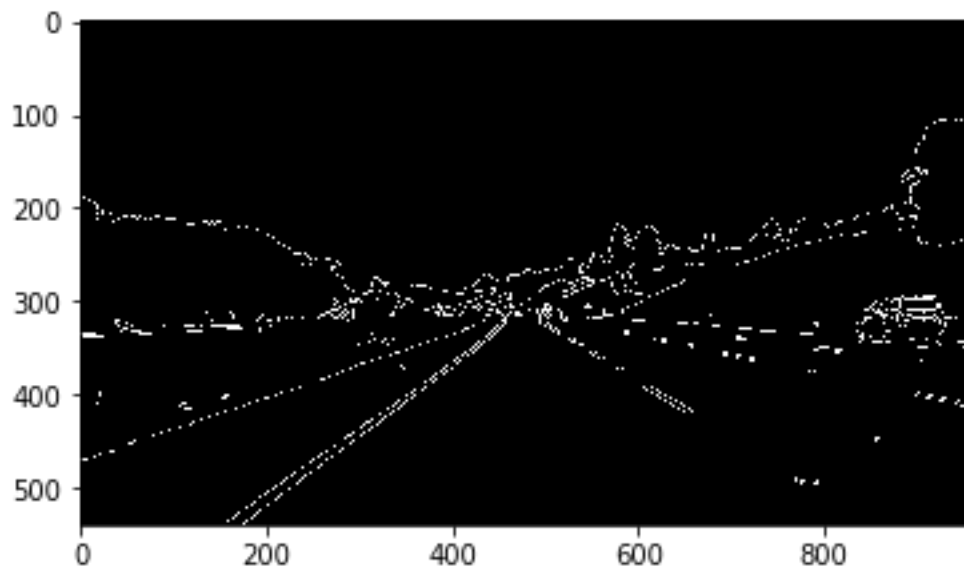
- My pipeline consisted of 5 steps.
 - First, I converted the images to grayscale



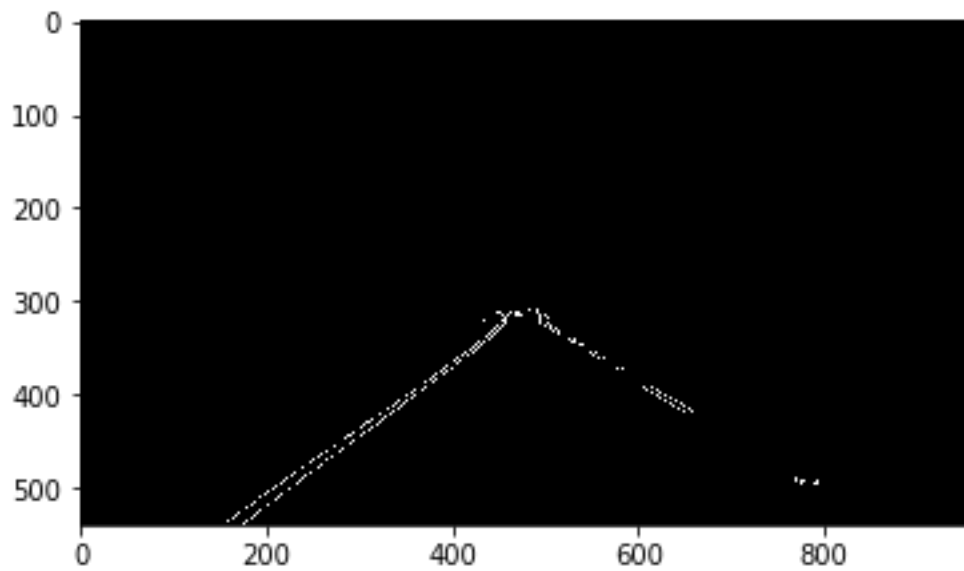
- Then I applied Gaussian Blur with kernel size=5



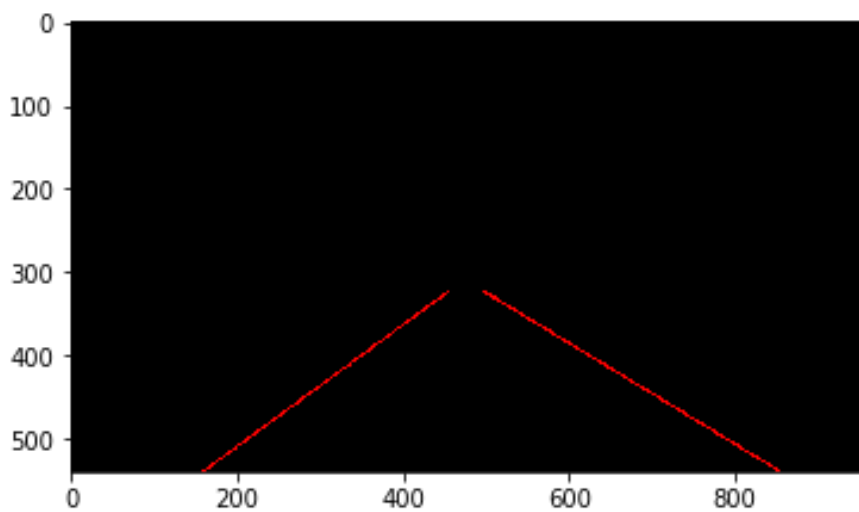
- After I used Canny Edge Detection to get the edges with min_threshold=50 and max_threshold=150



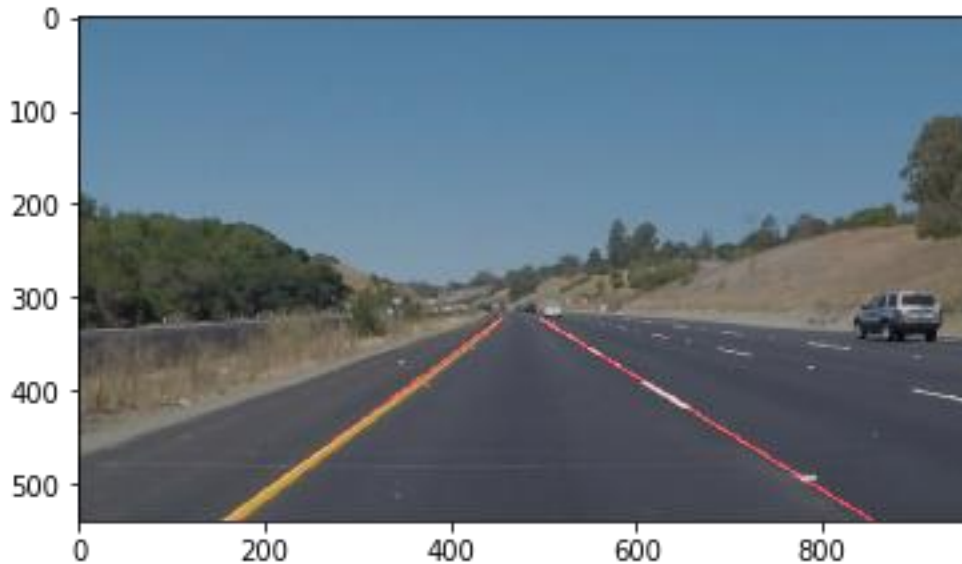
- I applied region masking using a polygon



- Then I applied Hough Transformation to get the lines.
- After that I interpolated the hough lines



- and finally combined them with the original image



- In order to draw a single line on the left and right lanes, I modified the draw_lines() function by
 - Find slopes of all lines But only retain lines where $\text{abs}(\text{slope}) > \text{threshold}$.
 - Separate lines into left and right lines based on whether their respective slope are positive or negative
 - Interpolate over all line segments to find the best fit of right and left lane lines
 - use polyfit to run linear regression to construct the final line

Potential shortcomings

This works goon on the actual data, so it will fail on different set of input like bad weather condition or night time drive or in shadows.

It fails on challenge video.

Possible improvements

For improvement I'll apply a more advance color filtration, Hough Transform ,image masking and interpolation methods.