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

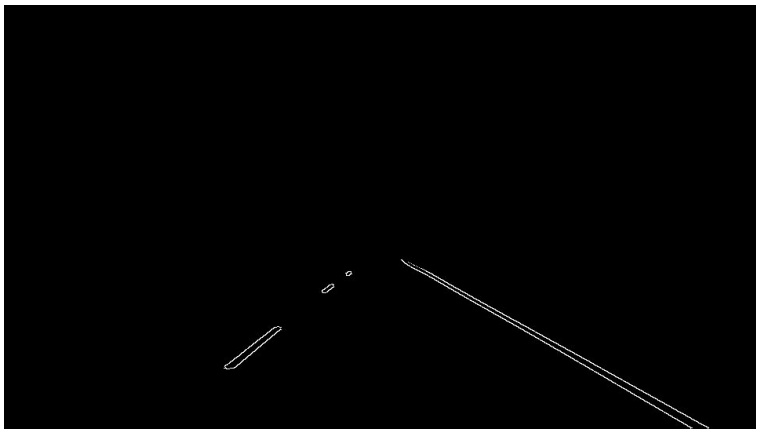
My pipeline consisted of the following steps.

1. Convert the image to grayscale

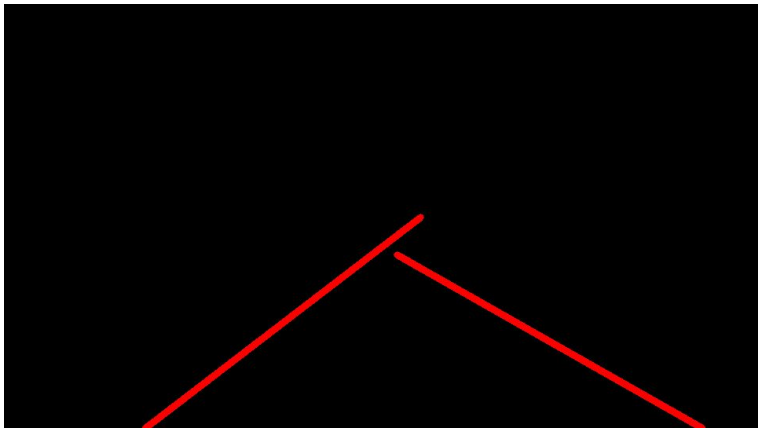
2. Clean up the image by applying a Gaussian Blur which removes noise and smooths out tiny details in the image

3. Run Canny Edge Detection to detect edges in the above blurred image using intensity variation

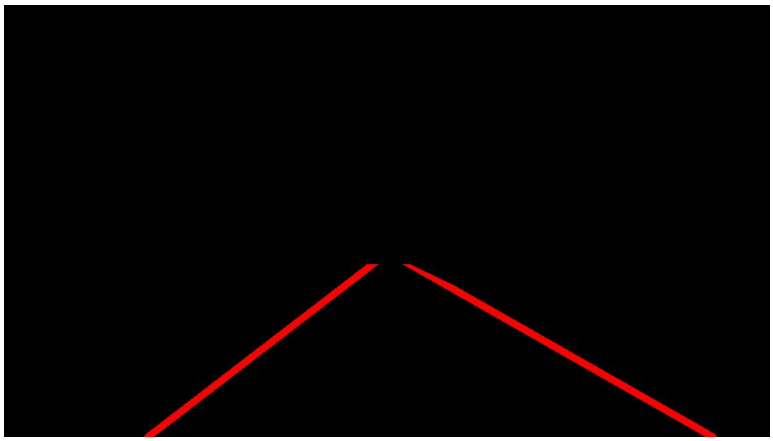
4. Setup vertices for a region of interest in the lower half of the image and eliminate edges from the previous step that are outside this region of interest



5. Use HoughLines algorithm to find all lines passing through the edge points that we got in the previous step
6. Modify the draw_lines function to consider only those lines that are not too vertical or too horizontal as these would not be valid in case of lane lines. Also group the lines (and their centers) into left and right lanes based on their slope. Calculate the average slopes and centers for the left and right groups of lines. Based on the center and slope calculate left line and right line end bottom points assuming one of the end-points for each will cross the bottom of the image. Similarly calculate the top endpoints by extrapolating the bottom and center points.

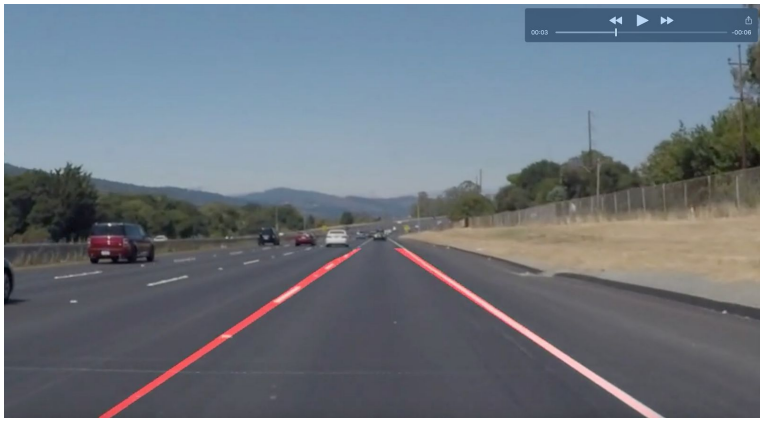


7. Run region of interest again to limit the extrapolated lines calculated in the previous step



8. Overlay the lines on the original image as final output.

Some examples of good lane detection:



2. Identify potential shortcomings with your current pipeline

One potential shortcoming as seen in the challenge.mp4 video is that extreme variations in the light can cause the line detection to not work correctly. It can end up detecting dark shadows on the road from trees, lane separators etc. as lines. Secondly, the road has patchy segments the line detection incorrectly detects lanes in those patches. Thirdly, if there lane lines on the road have faded or the color of lane lines is not too different from the road itself even then the pipeline doesn't work well. A couple failure example:

3. Suggest possible improvements to your pipeline

A possible improvement would be to somehow use color information to as an intermediate step to help supplement the edge and line detection algorithms. Another improvement could be use some grouping algorithm that can detect outliers from the left and right lines and ignore the outliers.