

Finding Lane Lines on the Road

Goals

1. *Make a pipeline that finds lane lines on the road*
2. *Reflect on work in a written report*

Reflection

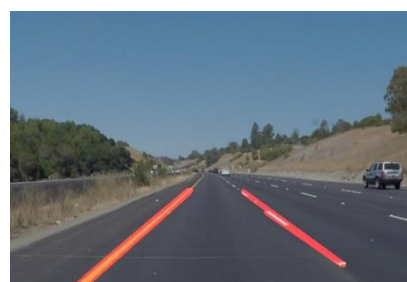
Describe your pipe line. As part of the description, explain how you modified the `draw_lines()` function

My pipeline consisted of 6 steps:

1. convert the images to grayscale using given helper function
2. apply gaussian noise kernel with the size of 5
3. run canny edges detection
4. create the region of interest (a trapezoid shaped area) and extract the edges within the area
5. doing Hough Transformation, to select and connect the lines
[NOTE] It's a bit tricky there, since normally the case would be we directly get the continuous lane lines like below



but for those lane lines not starting from border of the image, the result would be like below, the lines are “chopped”



In order connect those lines, a new helper function has been introduced--- 'select_lines()'. This function would first filter out some lines(most of them are noise lines) by giving range of accepted slope rate. Also this helper would identify whether it's a left margin or right margin. And choose the down most lines and extend the lane lines to meet the region border, with output like below



6. combine the extended lines and initial image together

Identify potential shortcomings with your current pipeline

1. One of the shortcoming would be the potential extended left/right margin would not hold the same width of the lines compare to others
2. The robustness of the algorithm would still be a problem since what if there are other cars running just in front of your car, this would disturb the edge detection a lot, as well as the Hough Transformation.
3. Once trying to make a sharp turn, it would be hard to track the gradient, make it harder to filter out the valid edges.

Suggest possible improvements to your pipeline

1. One possible solution would be filter the lines, and only select those that have close gradient to the mean gradient of all lines. Keep a few lines instead a lot of lines.
2. For the sharp turn use case, it would be better to select more short lines to achieve the smooth of the identified lane lines. But the region of interest have to be given in a more narrow area.