Finding Lane Lines on the Road

Reflection

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 used Gaussian blur filter on the grayscale images. Canny edge detector was used on that image. This was followed by choosing the region of interest, followed by Hough transformation. Weighted image was done to print the lane lines in the original image.

In order to draw a single line on the left and right lanes, I modified the draw_lines() function by first splitting the values into negative gradient and positive gradient. Negative gradient refers to the right lane, while the positive gradient refers to the left lane. Upon segregating based on the gradient, the formula of the line was obtained via np.polyfit. The parameters from the np.polyfit are used to find the coordinates of the starting and ending line. Another function of drawline was created to ease the process. After that, follow the steps on the first graph and modify the parameters of the Hough transformation to achieve a non-flickering and straight line representing lanes.

2. Identify potential shortcomings with your current pipeline

One potential shortcoming would be what would happen when we are exposed to curved lanes.

Another shortcoming could be based on the environment conditions that may introduce noise and affect the reading. Environment condition could refer to different lightings, wet road after rain,

3. Suggest possible improvements to your pipeline

A possible improvement would be to tweak the parameters and identify the optimal set of parameter configuration to further enhance the final result.

Another potential improvement could be to ensure under all environment conditions the pipe line works.