Finding Lane Lines on the Road

Hyunjin Cho

March 2021

1 Pipeline and draw_lines() function modification

My pipeline consisted of 5 steps. First, I converted the images to gray-scale, then I applied Gaussian blur function with a kernel size 7. After that, I applied canny edge detection. I applied an trapezoid image mask to select region of interest. At last, I applied Hough-Line transform .

In order to draw a single line on the left and right lanes, I modified the draw_lines() function with choosing boundary of acceptable slope for both left and right. Next step, I divide two section. One for left lane and other is right lane. For left lane line, I selected the region where the slope is negative. I also collect all the points of the lines that accept all the current step. In here, I only select the points where line begin in the middle because if I select two point with using maximum and minimum, the line will not continue all the way to bottom edge. Due to this issue, to solve this issue, I use line function y = mx + b to extend line. For right lane line, vice-versa.

In Figure1 to Figure6 shows bottom, output of the lines of image in the test_images are created:

2 Identify potential shortcomings with your current pipeline

I tried minimize potential shortcoming. However, if I select parameter Hough-Line, the lane can be show more better.

3 Suggest possible improvements to your pipeline

Selecting parameter for Hough-Line and canny edge detection can help to select better.

Figure 1: solidWhiteCurve

Figure 2: solidWhiteRight

Figure 3: solidYellowCurve

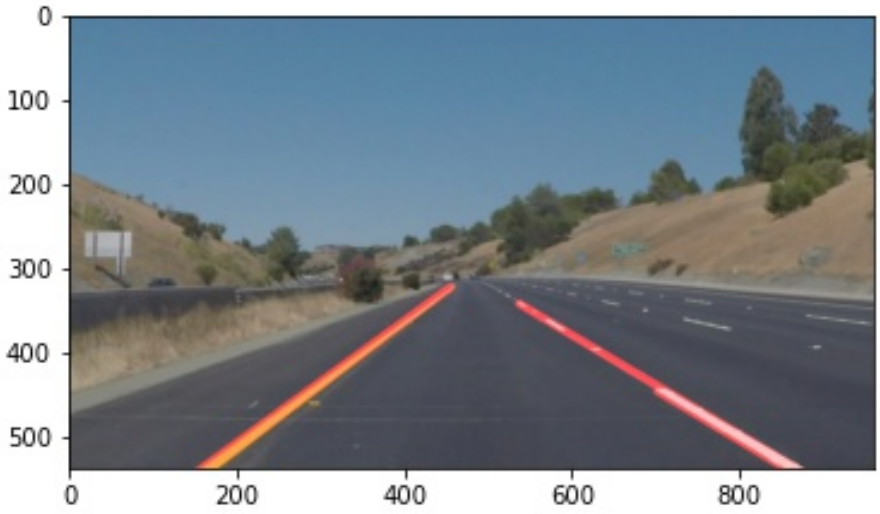

Figure 4: solidYellowCurve2

Figure 5: solidYellowLeft

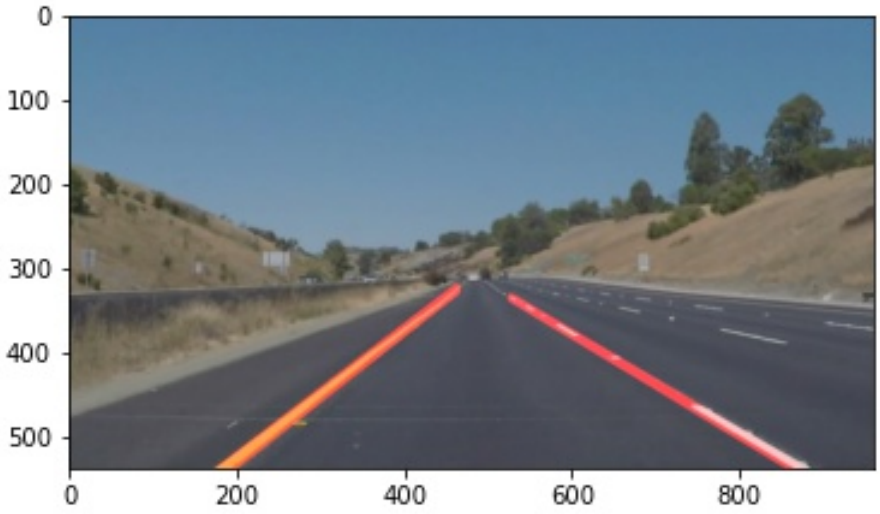

Figure 6: whiteCarLaneSwitch