Report P1 - Finding Lane Lines on the Road

1. Describe your pipeline. As part of the description, explain how you modified the draw_lines() function.

My pipeline consisted of 6 steps. First the image is copied and its original content is keep safe for final use. Then the image is converted to grayscale to be applied the Gaussian smoothing. After that is used the Canny Edge Detection in the region of interest that is the polygon made using the chosen vertices. Finally the pipeline does the Hough Transform to find Lines from Canny Edges. After all that the lines are inserted in the original image using the weighted function.

The draw_lines() function was modified only the parameter thickness of the line from 2 to 5.

Below are the six images generated along the project:

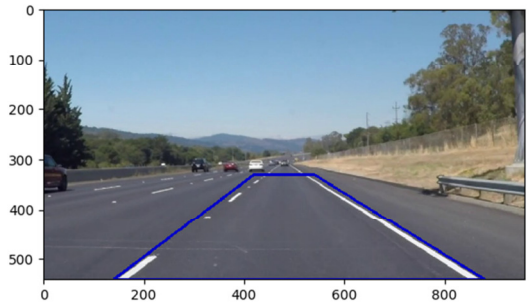

Figure 1 – Area of interest.

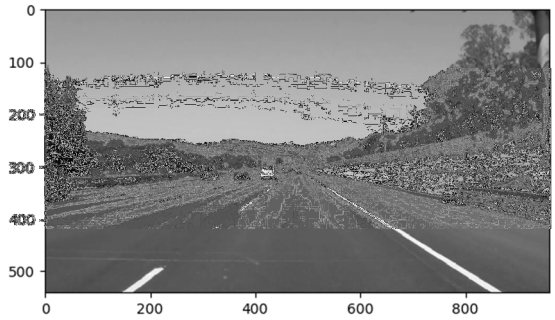

Figure 2 – Original image in gray scale

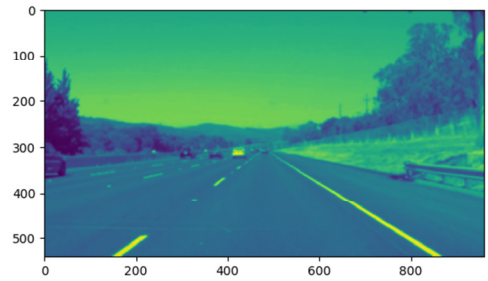

Figure 3 – Image after Gaussian smoothing / blurring.

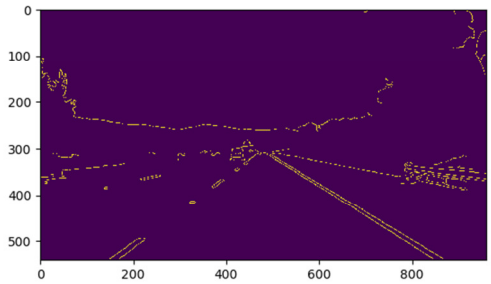

Figure 4 – Image with Canny Edge Detection

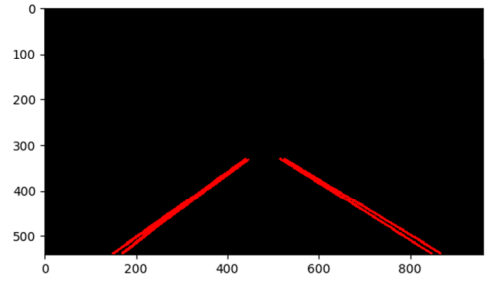

Figure 5 - Hough Transform to find Lines from Canny Edges only in the region of interest.

Figure 6 – Final image processed with the lines in red under the original image.

2. Identify potential shortcomings with your current pipeline

This current pipeline seems not faster enough to use in real time systems applications.

The processing of each frame has lot of computer cost and seems not the best approach. Besides that, some frames are more difficult than others and the result is not always good, for instance, the challenge video needs a different approach.

3. Suggest possible improvements to your pipeline

One improvement could be the choice of the interest's area considering any type of road. This pipeline must works with straight and curved paths for instance.

Other improvement could be the line length. How far the lines should go in front of the car? It must depend of the car speed at the moment.