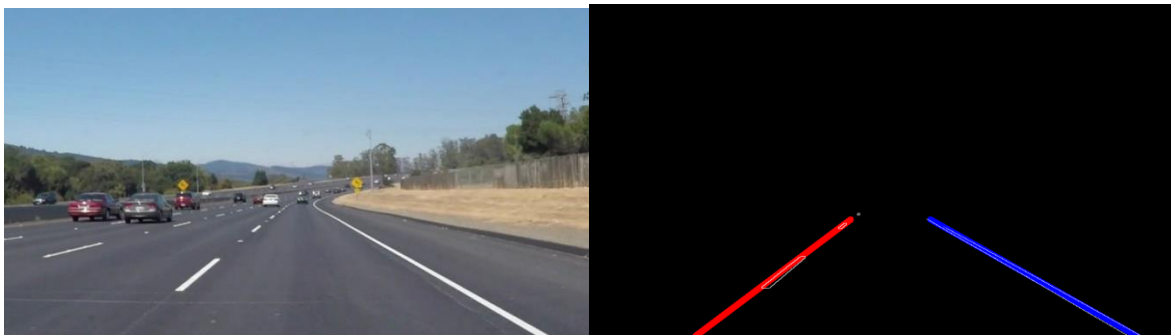


Finding Lane Lines on the Road

- 1) The pipeline consisted of first converting the image to grayscale, then I filtered the image with a Gaussian Blur with a kernel of size 5. Afterwards, I masked the image of unwanted color values and used the Canny algorithm to determine the images edges. With the images edges, I used a second mask to gain information on only the region of interest. With only the edges in the edges in the region of interest, I used the Hough Lines algorithm to determine the lines in the image that match the parameters, which are the lane lines. With the start and end points of each line, we can draw lines on top of the lane lines in the original image.

In order to draw a single line on the left and right lines, I modified the `draw_lines()` function by first determining which lines belong to the left lane or right lane by the sign of the slope. Afterwards, the slope and the y intercept can be averaged for the left and right lane separately. With the average slope and y intercept of each lane, we have the equation of the lane line necessary to gain the position of each point in the lane. The lane was then drawn from the bottom of the image to the end of the region of interest.



- 2) Unfortunately the current pipeline is rigid and generalizes the conditions of the specific road. If road conditions change, or the magnitude of the gradient of the image changes, the ranges tested are not valid and will give incorrect results. The model for the correct conditions needs to be more flexible.
- 3) A possible improvement would be to tighten the boundary of the region of interest to have less room for error. Another potential improvement could be to make the boundaries on the color masking more flexible to filter yellow and white lines separately and combine the end result.