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

My pipeline consisted of 6 steps. First, I converted the images to grayscale, then I smoothen the grayscale image with the gaussian_blur helper function (with the size of the kernel being 3). In the third step, I detect the edges in the smoothened grayscale image using the canny helper function with the low and high thresholds being 50 and 150, respectively. In the next step, I marked out the interested region with a trapezoid in the lower portion of the image. In the fifth step, I detect Hough line segments from the masked Canny-edge image and draw them all in a separate picture using the hough_lines helper function. Lastly, I overlay the Hough lines on the original color image using weighted_img helper function.

In order to draw a single line on the left and right lanes, I modified the draw_lines() function by finding the end points of the extrapolated left and right lines. To find these end points, the Hough line segments are binned according to the sign of their slopes, and then extrapolated. Extra cares are exerted so that the x-valued of the upper end of the extrapolated left (right) lane does not exceed (go below) 5/9*width of the picture (4/9*width of the picture), respectively.

2. Identify potential shortcomings with your current pipeline

One potential shortcoming would be what would happen when there are closer by cars in front, for example in city traffic when cars are travelling with lower speeds. Right now, we rely on the masked region to rule out Hough lines detected on the cars in front. In all the tested images and videos the cars in front are reasonably ahead so that they are out of the masked region; hence, the Hough lines in those cars are masked out. This might not be so when cars are travelling with lower speed in more heavy traffics.

3. Suggest possible improvements to your pipeline

A possible improvement would be to further classify detected lines to figure out whether they are from the road or from the other cars so that we can rule out the latter in our final lane detection algorithm.