

Finding Lane Lines on the Road

Reflection

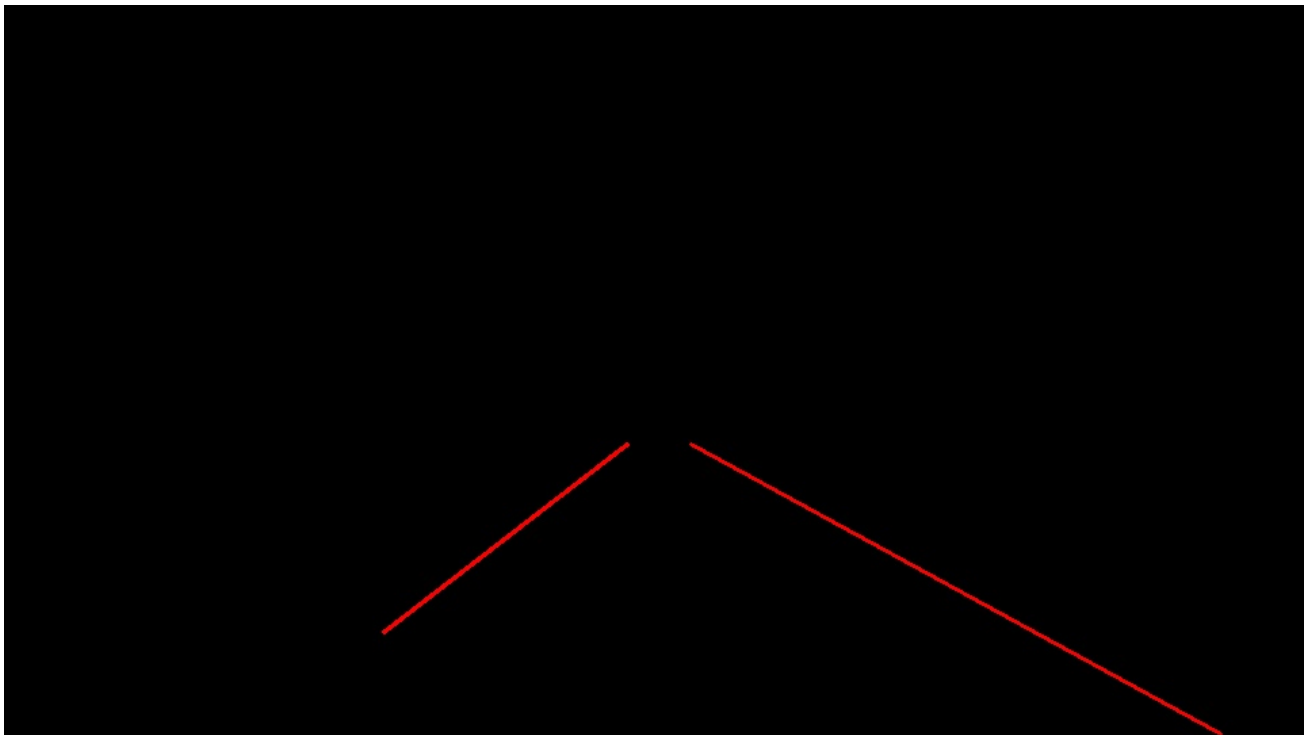
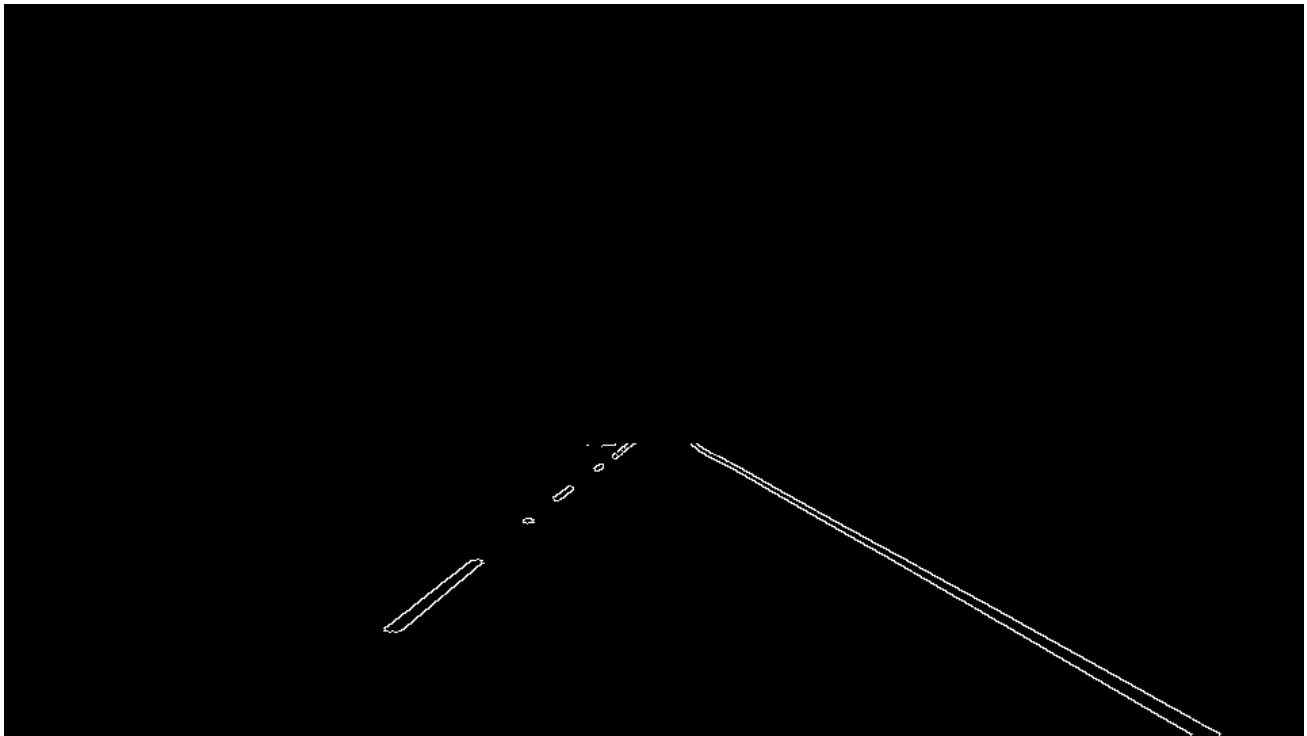
1. Describe your pipeline. As part of the description, explain how you modified the `draw_lines()` function.

My pipeline consisted of 6 steps. First, I converted the images to grayscale, then I used the Gaussian blur function to blur the noises. Thirdly, I used canny function to find edges. After that, I used `region_of_interest` function to draw a masked region and mask the edges to keep those in the front of the car. Then, I used `hough_lines` function to find lines in the masked picture. Lastly, I used the lines array I got from `hough_lines` function to draw lines on the picture by using `weighted_img` function.

The process are done as shown in following pictures:









In order to draw a single line on the left and right lanes, I modified the `draw_lines()` function by drawing two lines left and right separately. I compute the vertex point of all lines and get the 2 most remote points for each side. By remote I mean those points have minimal or maximal value of all vertex points of the points in the lines array.

2. Identify potential shortcomings with your current pipeline

One potential shortcoming would be current approach will not detect curved lines properly. As shown in the jupyter notebook file, the result for the challenge.mp4. The car should have information about a curved line like the direction of that line, the angel of that curve, ect.

Another shortcoming could be current approach rely on hand tuned parameters to work well under test videos. There is no guarantee this will also work for new environment. Not to mention even more complicated situations like lane lines under night views.

3. Suggest possible improvements to your pipeline

A possible improvement would be using other more powerful line detection methods other than `hough_lines`. Or we can train powerful Neural Network to mark lane lines if we have large enough dataset.

Another potential improvement could be to make program tune the functions itself. This could be done by changing parameters based on characteristics of input images. Other

possible computer vision pre-processing of the images could also be helpful.