

# Reflection

> see the notebook named *Lane\_Finding\_Project (1 of 11)* for the work I did for this project.

## 1. Describe your pipeline. Explain how the `draw_lines()` function works.

The first version of the pipeline uses grayscale, Canny Edge Detection, Gaussian Blur, a draw lines function, and a region of interest function/mask. I manually played around with different parameters until the individual images looked good, and then transferred the pipeline to the video.

The `draw_lines()` function looks at the slopes of the the left and right lanes and only draws lines for slopes of a certain range. These are the additional parameters I played around with to make the images/videos better quality.

The second version of the pipeline looked at the challenge video and I was able to identify three problem spots which I then worked on separately to try to bring out the lanes on the highway. The main problem areas were where trees and their shadows coincided with the lanes, where the lane lines were in areas of low contrast (still working on this at the time of submission), and the missing line for the lane segments on the right side.

I played around with having ROI before Gaussian Blur, and also having multiple Canny Edge Detections and Hough Transform throughout the pipeline. With each iteration, my images would improve. However, I was not able to apply this to the video — there was a processing bug that I was not able to figure out.

## 2. Identify potential shortcomings with your current pipeline.

Focusing on the important adjustments, I would make the `draw_lines` function better by limiting the left side to a certain slope and the right side to a different certain slope, rather than lumping the slopes together. That way we ignore the right slopes on the left side and vice versa. As you can see from my jupyter notebook, I wasn't able to figure out how to do a low contrast lane finding function. I also wasn't sure what to do with images that needed different processing processes.

## 3. Suggest possible improvements to your pipeline.

In the future, it would be a good idea to identify the best way to adjust the parameters through computer vision. It should account for things like having different colored lanes, or areas of high edges that are not related to the lane but still exist in the data. Maybe there could be an algorithm or machine learning technique to choose the best parameters for the Canny Edge Detection, Hough Transform, and Gaussian Blur. This would address the problem I came across for needing to use different processes to optimize different types of images.