

# Finding Lane Lines on the Road

---

## Writeup Template

---

**You can use this file as a template for your writeup if you want to submit it as a markdown file. But feel free to use some other method and submit a pdf if you prefer.**

---

### Finding Lane Lines on the Road

The goals / steps of this project are the following:

- Make a pipeline that finds lane lines on the road
  - Reflect on your work in a written report
- 

### Reflection

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

My pipeline consisted of the following steps:

1. Make a copy of the image
2. Convert the image to grayscale
3. Apply the gaussian blur with a kernel size of 5
4. Apply Canny Edge Detection with a low threshold of 100 and high threshold of 200
5. Mask the area
6. Apply the hough transform but instead of highlighting the found line segments, draw a solid line based on the longest line segment found

How I modified the `draw_lines()` function

- I first made a copy of the original `draw_lines()` function into `draw_lines_extend()`
- After that I calculated the slope and the length of the line
- Based on the slope I separated the data into left and right line tuples

- I would then try to find the longest line to use the slope and intercept (The code is set up to use the n longest lines, but after trying the 5 longest, 3 longest and 2 longest, my best results were with the longest)
- After determining the slope, I defined the y-coordinates of my new lines
- Then using the slope and intercepts that I calculated above, I would calculate the x-coordinates of the new lines
- I also tried to place boundaries on the x-coordinates to limit the range of values
- Each x-coordinate was cast into an `int()` variable
- Then I drew two lines based on the coordinates that I calculated above

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

### Potential shortcomings

- The region of interest was pretty tight in order to tune the line identification more quickly. It would be nice to improve the robustness of finding the lines to accommodate video that was not centered or started to bend more aggressively. My current pipeline could not handle the challenge video at all
- The averaging function was susceptible incorrect lines identified. The slope calculation in the current setting is based on a single line segment so if the lane markers were faint or only consisting of reflectors, this pipeline would probably not work
- The pipeline also could not detect which side of the solid line to use. This would cause the line extensions to jump side to side in the videos

## **3. Suggest possible improvements to your pipeline**

### Improvements

- Be able to handle the challenge video
- I would like to make the lane markers jitter or jump less
- Handle more curved roads