

# Finding Lane Lines on the Road

Zheng Chen

## 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 5 steps. First, I converted the images to grayscale, then I define a kernel size and apply Gaussian smoothing. After this, I define parameters for Canny edges and apply. Next, I create a masked edges image and run Hough on edge detected image. At last, I draw the lines on the edge image and save it.



Figure 1 Annotated Video

In order to draw a single line on the left and right lanes, I modified the `draw_lines()` function by calculating the center points and mean slopes of two lines. I save all coordinates and slopes in arrays and determine the mean value of coordinates of center point and slopes of two lines.

$$x = \frac{y - (y_{mean} - slope_{mean} * x_{mean})}{slope_{mean}}$$

From this equation, we can easily get the x value through chosen y value. And two lines are determined.



Figure 2 Annotated Video with improved draw\_lines() function

## 2. Identify potential shortcomings with your current pipeline

One potential shortcoming would be what would happen when noise exists. In the challenge video, there are some cracks on the road which are also detected and affect the results.

Another shortcoming could be universality. When we face a curve like in the challenge video, since two lines are straight, we cannot give an accurate result. And the range of slope needs to be determined more carefully.

## 3. Suggest possible improvements to your pipeline

A possible improvement would be to filter noises like road cracks.

Another potential improvement could be to take length of each line into consideration. I can use weighted average to calculate coordinates of center points and slopes.