Finding Lane Lines
Udacity Self-Driving Car Nanodegree Project#1

Overview:

My most memorable experience in car driving class is to turn on the car key for the first time, to push the gas paddle and keep the car on the lane. To control the car, I constantly look at the lanes and tried to steer it straight and for sure I did not look at the pedestrians.

Therefore, the most basic thing for self driving car is to automatically detect lane lines using an algorithm.

Python and OpenCV will be used to detect lane lines in video images.

Reflections:

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

My find lane detection pipeline consisted of the following steps in processing an image:

1. Covert Driving scenery image from RGB to gray scale
2. Apply a slight Gaussian blur to gray scale image
3. Perform Canny edge detection to detect lane
4. Define a region of interest and remove the undesired portions of image
5. Retrieve Hough lines
6. Find the Average slope of lane lines and coordinates to draw the lines.
7. Apply these Lane lines to the original image

In order to draw a single line on the left and right lanes, modification the draw_lines() is:

After getting all the lines from Hough transforms functions, we draw each line.

The modification occur while drawing on each line, we classify each line by its slope value. The line with negative slope is on one batch, the one with positive slope is on another batch. We then draw positive slope line and negative slope line by determine 2 points that make up each line. For each line, one of the points is on the horizon line and another one is on the hood line.

But before that we do that we need to rely on np.polyfit() to find the x coordinate of the 2 points that make up negative/positive slope line given that we know their y-coordinate.

2. Identify potential shortcomings with your current pipeline

In latest version I figured out how to remove extra edges of the polygon defining the region of interest. The draw_lines() was not able to perform well when the lane was curvy. One way to fix it is to draw smaller line. And find better way to approximate the slope and drawing lines.

3. Suggest possible improvements to your pipeline

1. Need a better improvement in generating region of interest in order to remove unwanted lane lines.
2. Need a better way to implement draw_lines() to do it faster, to approximate slope better and to draw smaller line better.