Project 1 Lane Lines Writeup **Hardik Patel**
November 16, 2017

Detecting Lane Lines

Pipeline description

My pipeline was as follows -

1. Apply Gaussian Blurring
2. Apply Canny Edge detection
3. Apply Region of interest - trapezoid.
4. Apply hough space transformation
5. Finally, overlay the lines on the original image

For the `draw_lines()` function, I calculate the slope of each line detected and contained within the "line" variable. I then separate the slopes based on the whether it is positive or negative. I average slope for each lane and chose a random point in the line variable to make the line go through it. I then calculate the intercept of the line on the x-axis and use the two points to draw the line.

Shortcomings with the Pipeline

One significant problem is that I couldn't get the best values required for the canny and hough transform functions. Hence, in some frames, especially in the second video, I am unable to detect the lines. This in turn puts my function in a problem, as to average I divide by the number of lines. If the number is 0, then division by 0, is impossible. Hence in those frames my line is is a completely different place than the lane. I believe this stems from the wrong inputs to the two function. Unfortunately, even after experimenting with different values, I couldn't get a robust output.

Another, problem that is less significant is even when I chose color as [255, 0, 0], the color in the mage outputs was still set to blue, in the images. Something is overwriting the color and I cannot figure out what.

Improvements to pipeline

I believe my main problem is the above shortcoming I mentioned, hence to improve my pipeline I need to use better values for the Hough and Canny functions.
Another improvement in the `draw_lines()` function would be to not chose a random point and average all the points.