

****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

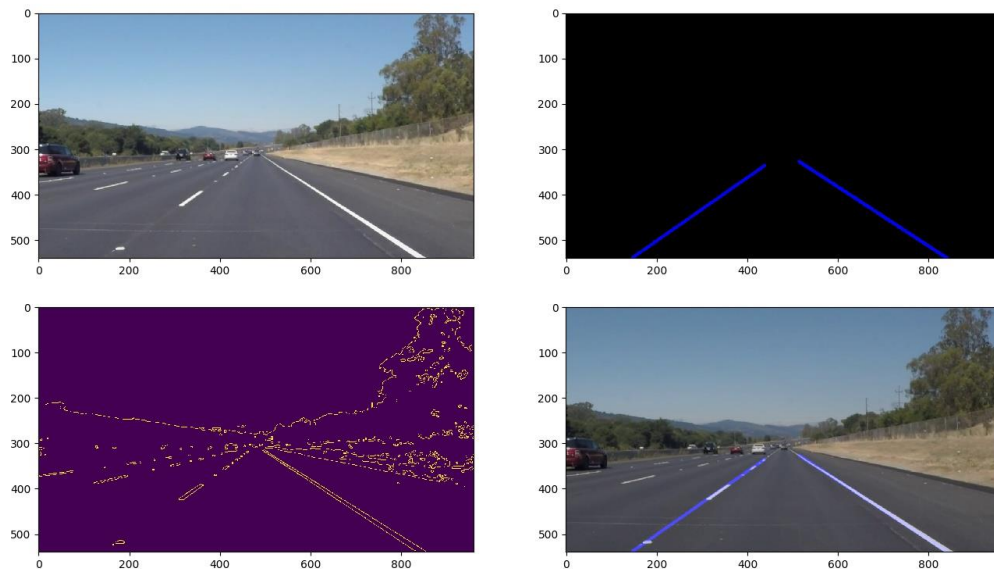
1. Describe your pipeline. As part of the description, explain how you modified the draw_lines() function.

My pipeline consisted of 5 steps.

- First, I converted the images to grayscale and HSV
- Then I split the image using two separate masks using cv.inRange function.
 - First Mask looked for Yellow Lines
 - Second Mask looked for White lines
- Then I Ran a Gaussian Blur on the Image with a kernel size of 5. (Started with a 3 and went upto a 9. Realized 9 was too blurry, so tuned it down to 5 (trial and error))
- Applied the Canny Transform and cropped the transform using Region of interest. I hardcoded the ROI for a 540x960 image/video. The Challenge video was modified to 540x960 before I ran the algorithm on it.
- I then ran the Hough Transform on the cropped Canny Image.
- I modified the draw_lines() function in the following ways.
 - First I classified the lines into right and left based on positive and negative slopes.
 - Then I calculated midpoints of each line and their distance from the centre of the image.
 - Then I calculated a weighted average of the points based on giving more weight to lines with a midpoint closer to the centre of the image. This reduced the probability that the lines found further away from the centre would not affect the main line that needed to be drawn.

- Then counting the number of frames since the start of the video/image, I started populating a cache with a stack (last-in-first-out) of 25 frames.
- I calculated an average on the entire stack and made that the new main line.
- Using the two points on the main line, I calculated the slope and knowing the end of the y-axis (i.e., 540), I calculated/extrapolated the third point x value.
- Then I plotted them.

To show the work on an individual picture



Other pictures and videos are present in the Git repository and the attached zip file.

2. Identify potential shortcomings with your current pipeline

One potential shortcoming would be what happens when the color of the road suddenly changes closer to the white range. This is seen in the Challenge video and I overcame that by increasing the cache of previous frames and averaging. But I feel there should be a better way to do it.

3. Suggest possible improvements to your pipeline

As mentioned in 2.