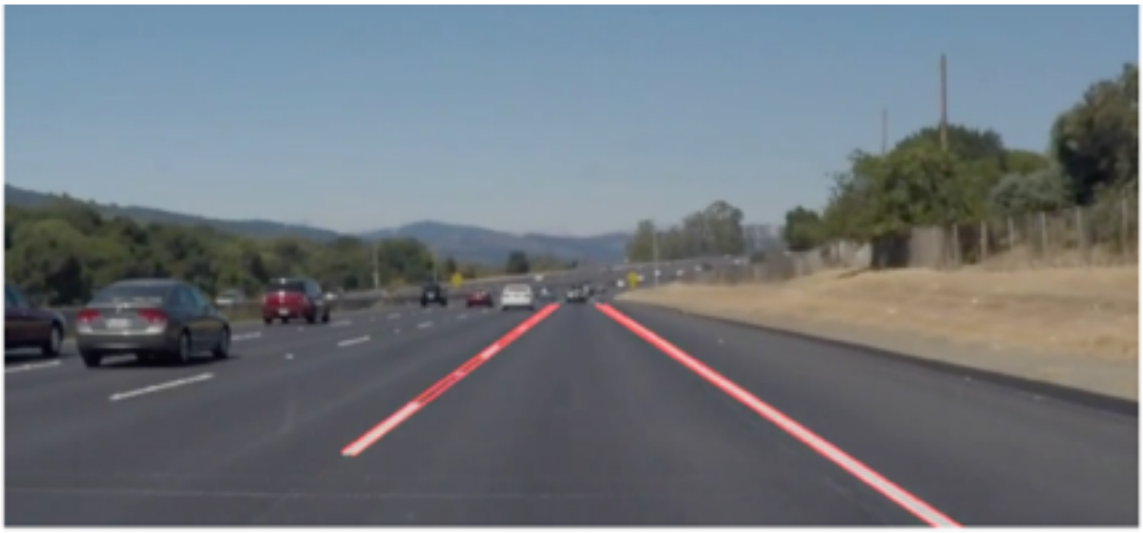
Finding Lane Lines on the Road

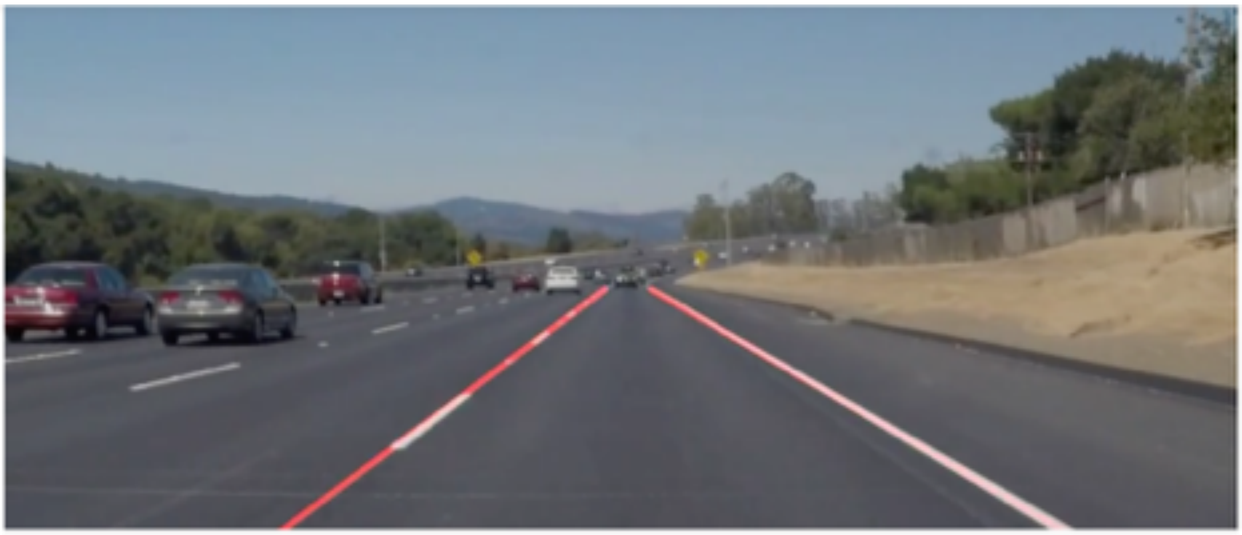
1. Description

First, I converted the images to grayscale.Then I uses the Canny function to find the edges of the lane lines in an image of the road .Canny runs a gradient on the image to find sharp changes in the pixel intensities. I limited my area of interest by using quadrilateral region mask using cv2.fillPoly() function. In the end I ran a Hough Transform to accomplish my task of finding lane lines.I tried running my pipeline on all sample images provided in the git repo and kept on trying different function parameters for function HoughLinesP until results were satisfactory.

I then ran my pipeline on the sample videos but the results were not at par with the video P1_example.mp4 . The dotted lane lines were not consistent and I learnt that somehow i need to draw a single line on the left and right lanes to get better results.Following is the snapshot of the video before modifying draw_lines() function. (**Notice the left lane line is not consistent**)



In order to draw a single line on the left and right lanes, I modified the draw_lines() function by separating lines on the basis of slope/angle and then extrapolated them using np.polyfit(). Following is the snapshot of my final code -



2. Potential shortcomings with the current pipeline

One potential shortcoming would be what would happen when there are additional neighbouring objects like cars etc and these objects might mess up my lane lines.
Also, i have currently hardcoded the quadrilateral vertices to limit my area of interest and I am not sure how my code will behave in more curvy roads (like roads on mountains) .

3. Possible improvements

A possible improvement would be to able to identify neighbouring objects and isolate the lane lines from these objects.