IMPLEMENTATION: PIPELINE

The aim of the project is to identify the lane markings on a series of video images by making use of the various helper functions provided. The answer was formulated using the appropriate parameters for the helper functions and also figuring out the order of the steps involved. After this, it was required to implement a function that will fit/ extrapolate a straight line through the line segments that are part of the lane markings on either side to represent one continuous lane for either side of the lane.

Steps involved:

1. Extract image frame from the video.
2. Convert the image to grayscale
3. Apply Gaussian Blur
4. Implement Canny edge detection
5. Extract the region of interest
6. Apply Hough transformation
7. Filter the line segments
8. Fit a straight line representing the average of all line segments in a lane.
9. Overlay the detected line segment over the original image/ video frame.

Parameters chosen:

1. Grayscale : None
2. Gaussian Blur kernel size: 5
3. Canny Edge: low threshold : 10, high threshold 250
4. ROI : Vertices were chosen
5. Hough lines: rho: 2,theta : np.pi/180, threshold: 15,min line length : 7, max line gap: 20
6. Weighted image : α=0.7, β=1., γ=0.

POTENTIAL SHORTCOMINGS

1. One shortcoming of the current implementation is that the horizon is not chosen dynamically. This would present challenges when the roads are not flat; for e.g: hilly terrain
2. Lane markings are treated only as straight lines, which means the detection will fail at curves and turns
3. Stray markings and zebra crossings maybe wrongly identified as lanes.

POTENTIAL IMPROVEMENTS

1. Color of the lane could be identified as yellow or white based on the pixels corresponding to the lanes identified
2. Lane departure warning could be implemented based on the deviation between center of the lane and center of the image.