# Project 1 Finding Lane Lines on the Road

## 1. Pipeline description

For video processing, we use opencv to load the video. Then we extract frames from video using OpenCV functions and process these frames one at a time as per the description below:

Step 1: Convert RGB frame to gray scale and filter it using a Gaussian filter to blur the image.
Step 2: Detect edges using Canny edge detection algorithm.
Step 3: Define a polygon region of interest to discard any detected edges that fall outside of such region.
Step 4: Compute Hough transform to find slope of the lane lines.
Step 5: To draw the lines, negative slopes, positive slopes from the hough transform and its y-intercept are stored in separate array. From this, we can find the average of positive slope, negative slope, and average of the respective y-intercept. This allows us to draw line using equation of line *($y = mx+b$), where,*
*$y$ = y coordinate*
*$m$= slope*
*$x$ = x coordinate*
*$b$ = y-intercept*

step 6: IIR filter is used to average out the slope and y- intercept for smooth transition in between the frames. We then make use of these new average slopes and average intercept to draw lines in the frames.
Step 7: Output a video from these frames using OpenCV. The video is outputted inside folder '/Output_video/test_output_swr.avi'.

## 2. Potential shortcomings with the current pipeline

The pipeline would present a bad behavior when almost horizontal edges are detect. This would add noise to the average slope and intersect values. The pipeline also cannot filter the shades in the road from the lines in the road. If the lines are curved, then the equation $y=mx+b$ is not valid and the pipeline fails to detect lines accurately.

## 3. Possible improvements to the pipeline

1. Filter out slope that are closer to 0.
2. Do not update the line if the slope changes by a lot between subsequent frame. Keep using the current slope.