Finding Lane Lines on the Road

The goal of this project is to make a pipeline that finds lane lines on the road.

Description of the pipeline:

The pipeline consisted of the following steps.

- Read the image and convert it to grayscale so that pixel gradients can be calculated for edge detection.
- Apply Gaussian blur to blur the image to remove noisy gradients
- Apply canny edge detection to find edges in the image
- Mask the image to find the region of interest(lane lines)
- Apply Hough transform on the region of interest to find lines
- The lines detected by Hough transform are passed to the draw_lines() function to find solid lane lines

draw_lines() function:

The draw lines function was modified in the following way.

- Calculate the slope and average the x and y coordinates of each line derived from the Hough transform
- The small positive slopes designate right lane lines and the negative slopes designate the left lane lines
- Create an array of positive and negative slopes respectively.
- Create an array of averaged x coordinates (derived from step 1) and an array of averaged y coordinates for the positive slopes for each line of the hough transform. Do the same for negative slope.
- Find the averages for positive slope and negative slope respectively.
- Find the average of the x and y coordinate arrays for left and right slopes respectively to find a single point on the left and right slopes respectively.
- The average slopes, x and y coordinates are then used to create lane lines.
- The y_top and y_bottom coordinates are fixed to determine the extent of the line .x_top and x_bottom is then calculated using the average slope, x and y coordinates from the previous step.
- The x_top_right, y_top_right, x_bottom_right, y_bottom_right for the right slope is then passed to cv2.line function to draw the right lane. Do the same for the left lane.

Following are the outputs of the six test images after applying the pipeline.

solidWhiteCurve

solidWhiteRight

solidYellowCurve

solidYellowCurve2

solidYellowleft

whiteCarLaneSwitch

Shortcomings of the current pipeline

- This pipeline would not work for different weather and light situations. This is evident when the pipeline is run against the challenge video, it is unable to detect the lane lines when the road is light colored or when shadows cover the lane lines.
- It will also not be able to detect sharp curves as we are fitting a straight line and then extrapolating it.
- It is unable to detect the road for all scenarios.

Improvements to the pipeline

- To detect lane lines under different light conditions we can make use of HSV representation.
- Detecting curves better could be done by finding small lines across the curve and then fusing them together non-linearly.
- A 3D view of the surroundings will help determine the flat road.