Project One: Lane Detection

Problem Statement

The objective of the project was to develop a software pipe line to detect lanes from road images and video.

Pipeline

Following image processing functions were used in sequence to extract lane features within the image:

1. Grayscale: Grayscale the image as canny edge detection needs image in gray scale format.
2. Gaussian Blur: Blur the image to make it easier for canny function to detect the edges.
3. Canny Edge Detection: identifies edges by finding collinear high intensity pixels.
4. Mask by region of interest in the image where lanes are to be detected: also ensure other features such as trees or adjacent lanes are not picked up by the edge detector
5. Hough Transform: transforms image lines to Hough space to obtain candidate lines that represent lanes
6. **Draw Lines: Hough transform returned multiple left and right lane line coordinate tuples (candidate lines). In order to detect a single left lane and a single right lane a smoothing function was applied within draw lines function. The steps involved were as follows:**

    a. **Use line multiple tuples (x1, y1) and (x2, y2) obtained by the Hough transform to calculate slope.**
    b. **Partition them into positive and negative slopes for left and right lane.**
    c. **Average slopes obtained for left lane lines and right lane lines.**
    d. **Find centroid for left and right lane.**
    e. **Project line until the y vertex for the masked region is reached.**

Tuning Lane Detection:

The initial lanes detected were not perfectly aligned with the actual lanes and seemed to jitter a lot with the videos.

Grid search on all pipeline parameters:

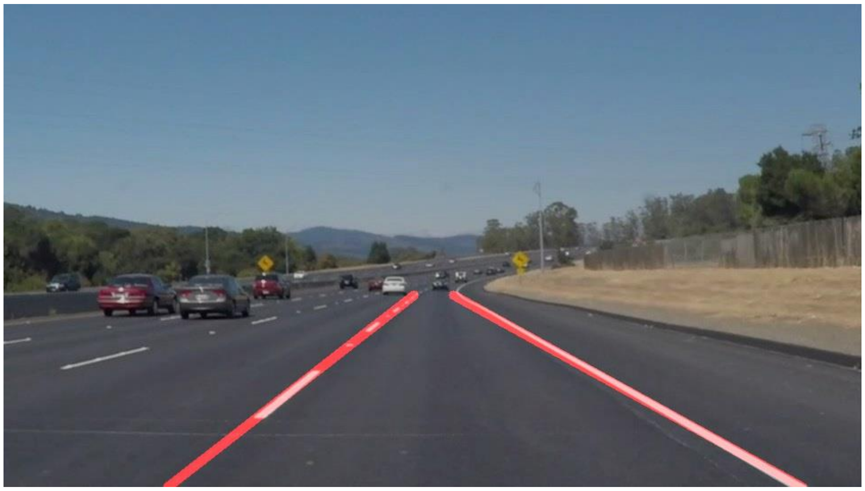
I calculated ideal slope manually on an image for left and right lane.

Then, I created an error calculation function that took multiple instances of all parameters used in the pipeline such as: kernel_size, low_threshold, high_threshold, rho, theta, minLineLength, maxLineGap and calculated difference between ideal left/right slope and (average_left_slope /average_right_slope) which were calculated using line tuples obtained from Hough transform.

Then parameter combinations that resulted in min error were chosen.

Final Result

The lane detection on both images and videos was significantly improved.



The alignment was further improved by throwing out slopes that were much farther away from ideal. Manual tuning of these bounds further improved the accuracy.

Further improvements:

1. Jitter on videos can possibly be improved by applying some sort of smoothing of parameters between successive frames.
2. The challenge video will require perspective transform and quadratic curve fitting for good lane detection.