

# Udacity Self Driving Car Engineer

## Project 1 – Finding Lane Lines on the Road

### Writeup

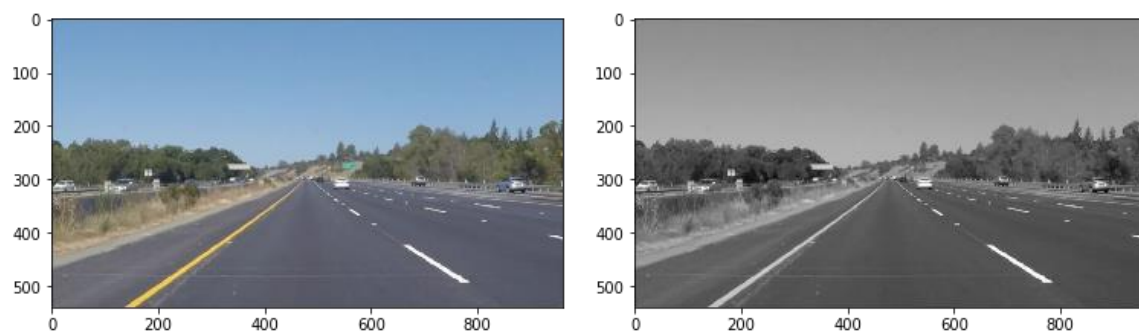
The goals of this project are the following –

1. Make a pipeline that finds lane lines on the road
2. Reflect on your work in a written report

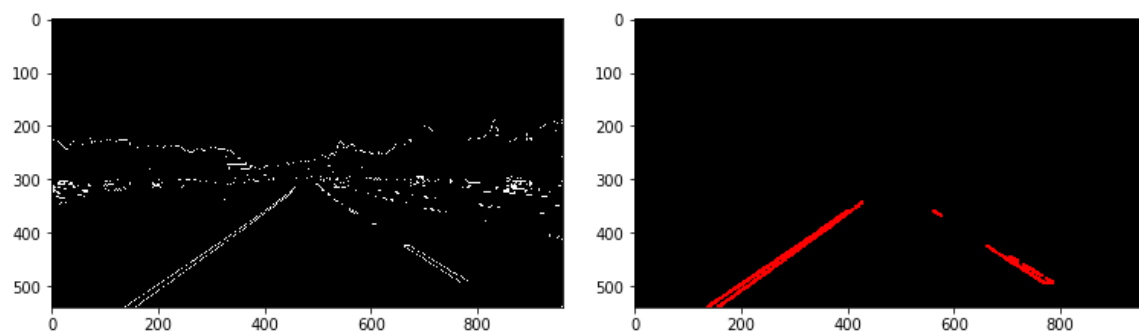
### Reflection

#### 1. Description of Pipeline

The original image is first converted into grayscale.

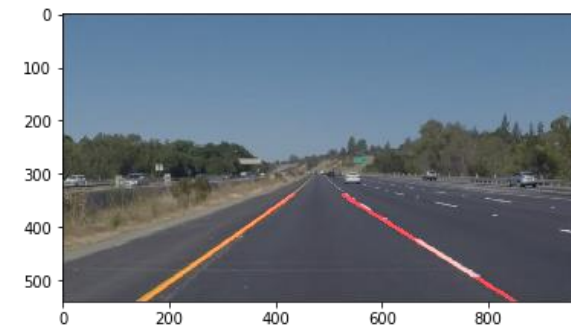
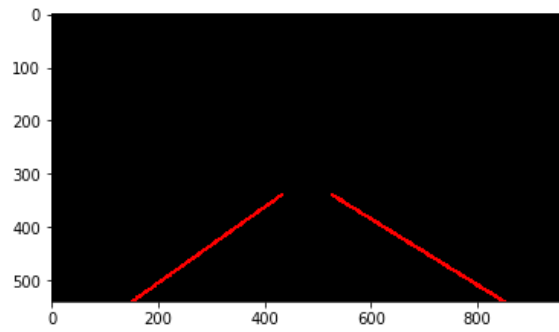


The grayscale image is then passed through Canny to output the edges. The edges are then masked over a region of interest. Hough transform as taught in the lesson then produces lines.



The draw\_lines function is modified to calculate the slope and intercept of the hough lines and average these. Once averaged, extrapolated line is drawn from the bottom edge of the image with average slope and average intercept.

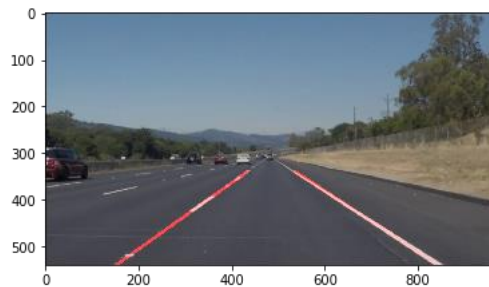
Finally, these extrapolated lane lines are overlaid on the original image to produce the result image.



Once the pipeline is tested on a variety of images, it is then copy pasted into the Process Image function. Video files provided in the project are then tried out with the pipeline.

### Output of all the Test Images:

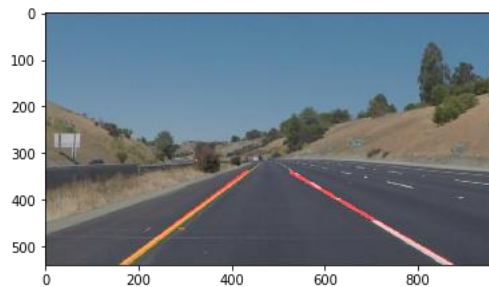
solidWhiteRight.jpg



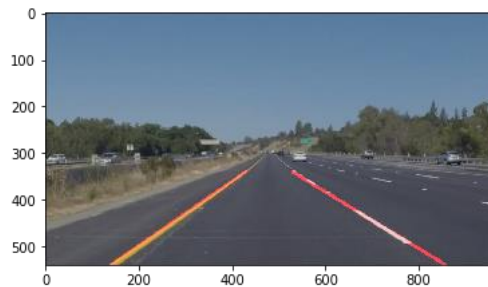
solidWhiteCurve.jpg



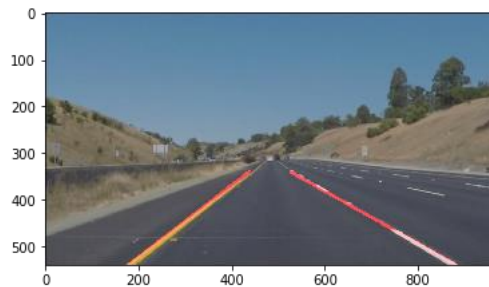
solidYellowCurve2.jpg



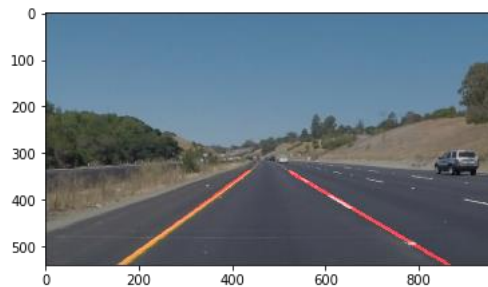
solidYellowLeft.jpg



whiteCarLaneSwitch.jpg



solidYellowCurve.jpg



The first video with solid white line works very well with the initially tuned pipeline.

The second video with yellow line does not work very well with the initially tuned pipeline.

Additional tuning efforts – Changes are made to the tuning parameters such as canny thresholds, minimum line length, max line gap, masked area etc. These tuning efforts are done with a goal to get a stable line detection on both sides of the road lane.

Video is ran through the pipeline and tested for output of the hough lines. Repeated tweaks are made to the tuning parameters so as to get rid of the “noisy” hough lines. The noise in the detected hough lines contribute to the average slope and average intercept and makes the lane detection unstable.

After a certain amount of tuning, a reasonably decent output is produced that roughly detects left and right lane lines.

## **2. Shortcomings in the current pipeline**

The output through the pipeline is successful in roughly detecting the lane lines, but it could still use some more refinement.

The pipeline is currently “hard” coded with tuning parameters. It is not very adaptable to different situations and lighting conditions. The challenge video failure through the pipeline shows some of these shortcomings.

## **3. Possible improvements to the pipeline**

Another method to average and extrapolate the hough lines might be appropriate.

Multiple sets of tuning parameters might be needed and a method to switch between these depending on conditions might be appropriate.