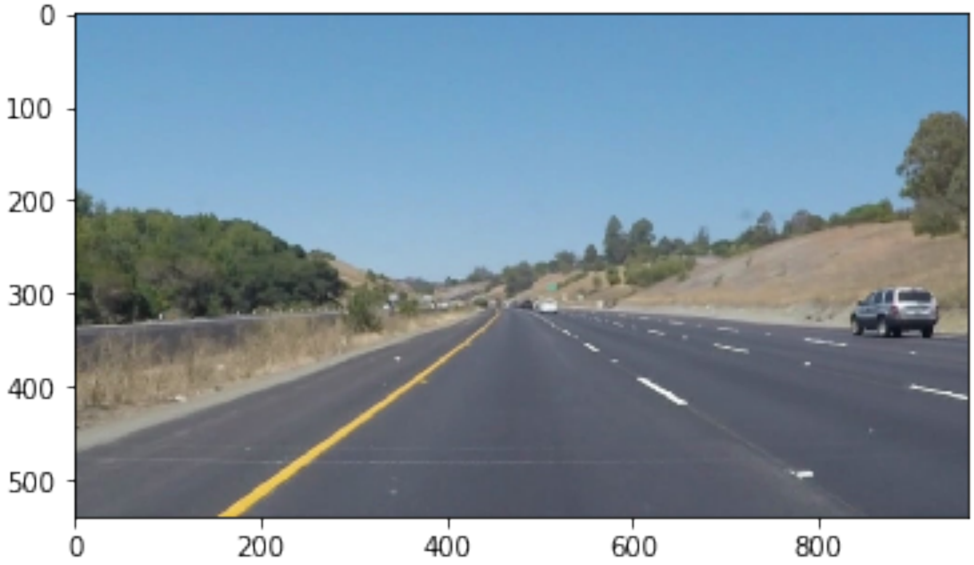
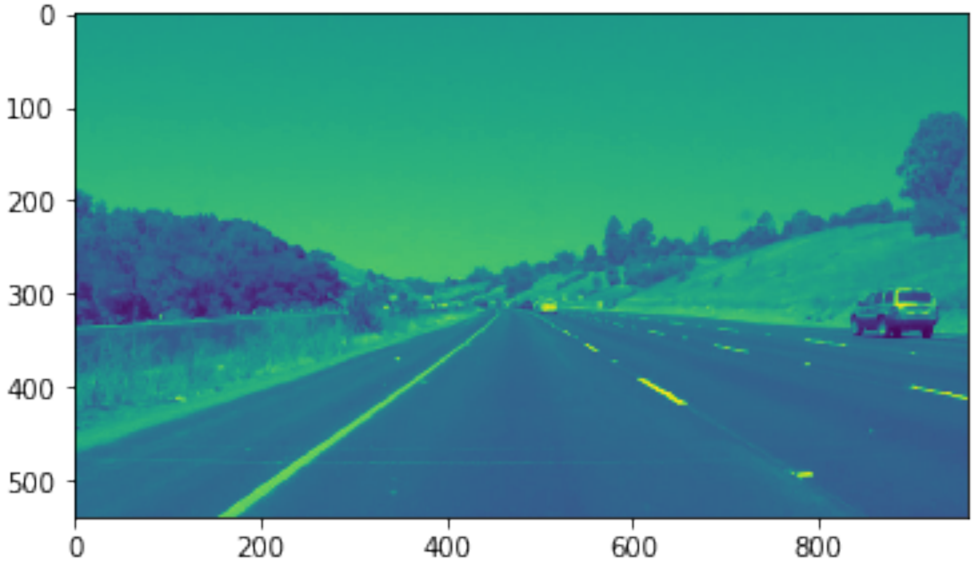
Finding Lane Lines on the Road

My pipeline consisted of 5 steps:

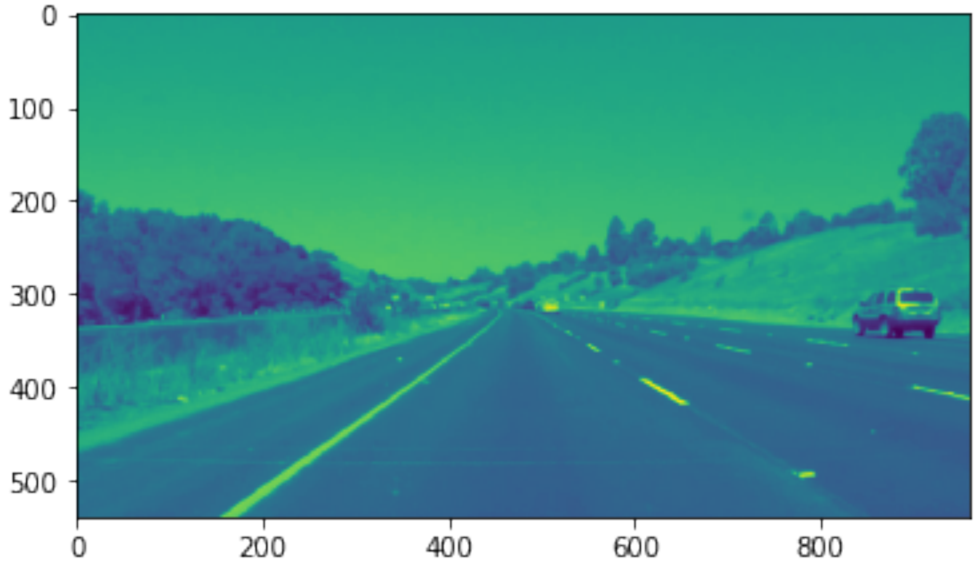
0. Loading the Image



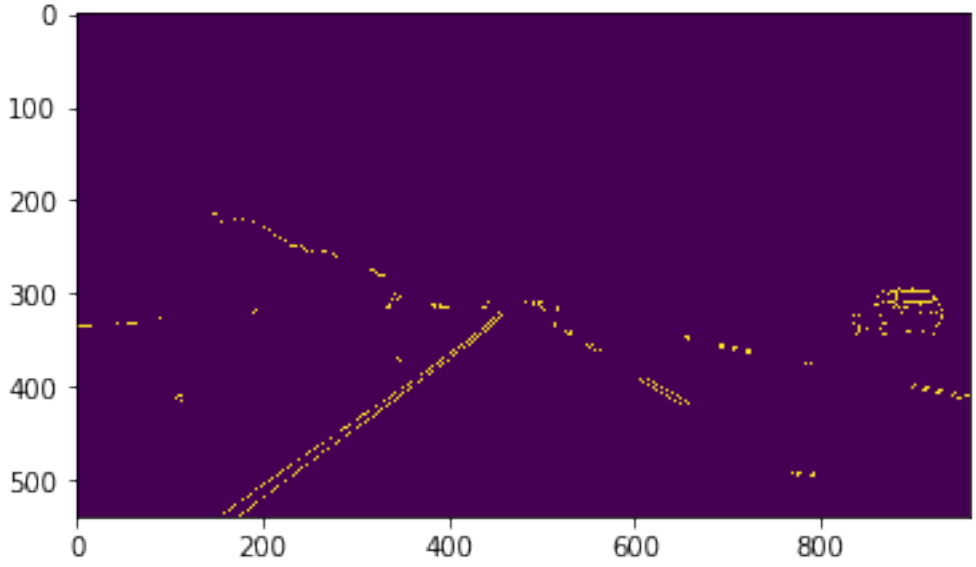
1. Transfering images into grayscale



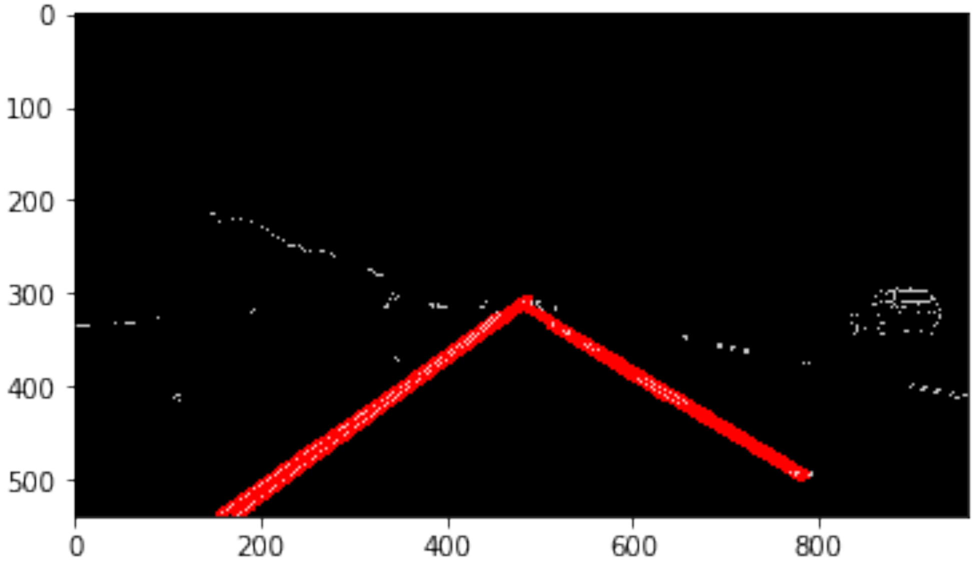
2. Performing Gaussian smoothing



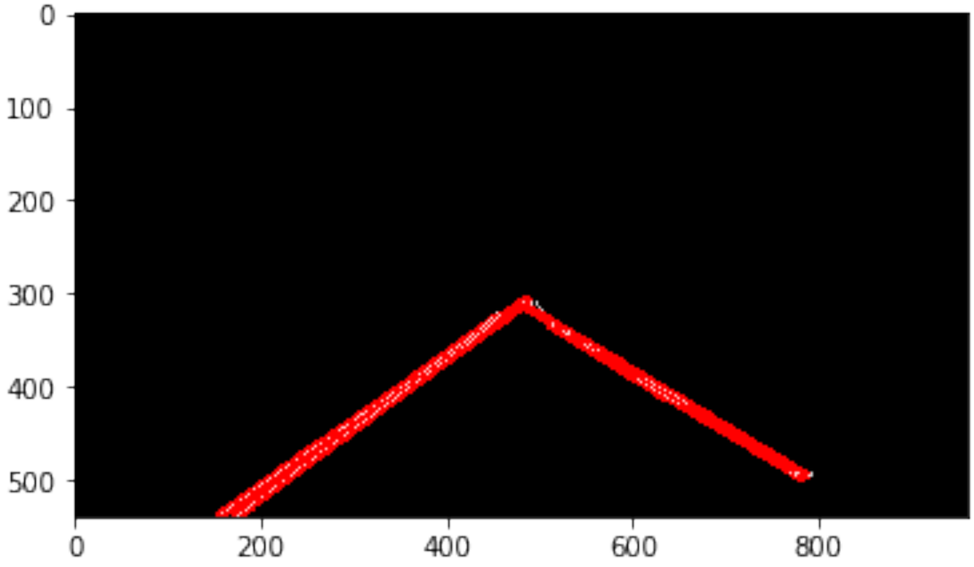
3. Running Canny edge detection



4. Hough Transformation



5. Selecting only relevant part of image



In order to draw a single line on the left and right lanes, I modified the draw_lines() function by:

1. Dividing all lines into two groups with:
    a. positive slope
    b. negative slope

2. Per group calculating the average x1, y1, x2, y2 values

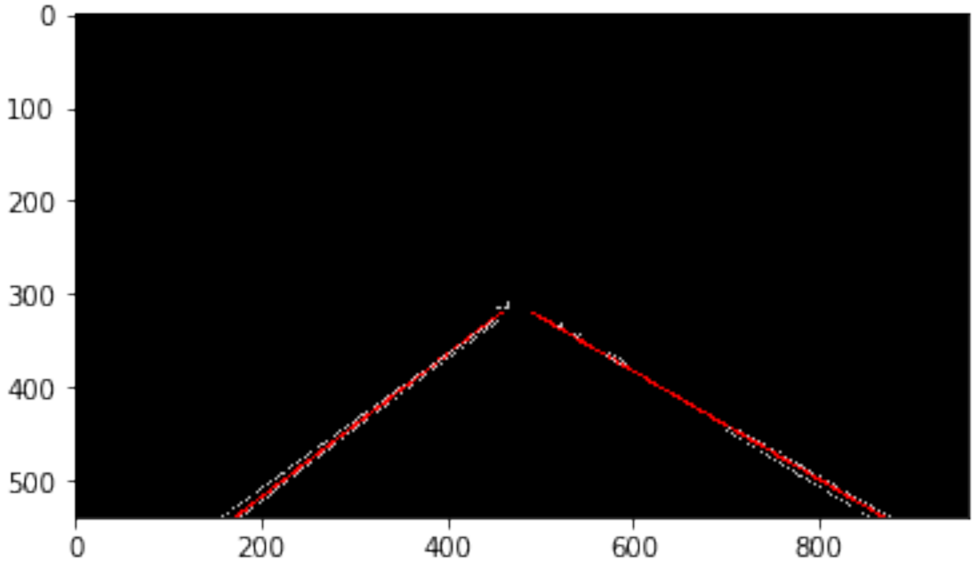
3. Setting the lower border of the image as y_low
   (I used an inverse axis scaling here. This was easier for me to handle.)

4. Finding a proper value for y_high by hand.

5. Calculating the x values from the slopes and the y values.

6. Drawing a combined positive slope line and a combined negative slope line with cv2.line

The overall result looks like this:



2. Identify potential shortcomings with your current pipeline

One potential shortcoming would be what would happen when:

1. The Canny edge detection delivers more than one pos. slope line or more than one neg. slope line, which do not belong to the same real line.
2. If there are generally any other lines at any angles from the Canny edge detection.
3. If the car is not driving in between the lines, but e.g. right on top of one, then the other one will just be cut out.
4. It only works for lines which are straight in the closer proximity of the car,
   not while the car is driving in a curve.
5. If lines would be partly covered by other vehicles, the result would deteriorate massively.
6. Bad weather and illumination conditions are not compensated for.
7. Special situations like entry and exit lanes are not being considered so far.
8. …

3. Suggest possible improvements to your pipeline

Possible improvement would be to:

1. Only chose line segments within the “draw_lines” function,
   which fits within a tight range of either the slope of a normal left or a normal right line.
2. Do the region select early in the pipeline and then throw away the edge
   which comes from the corner of the selected region
   before going into the Hough transformation.
3. For these videos: Filter quite exactly for the colors of the white and yellow lines,
   to get rid of the cement wall an its shadow in the challenge video.