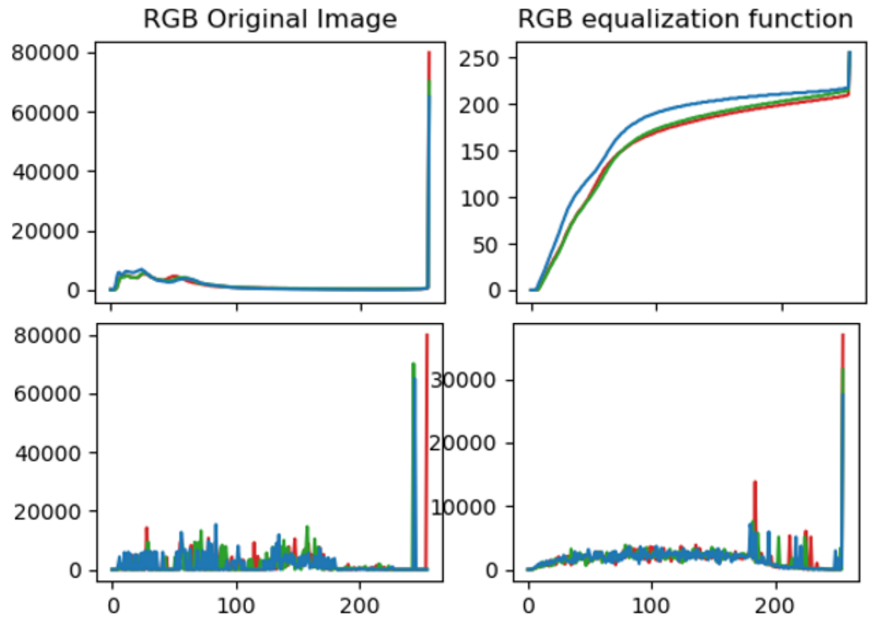
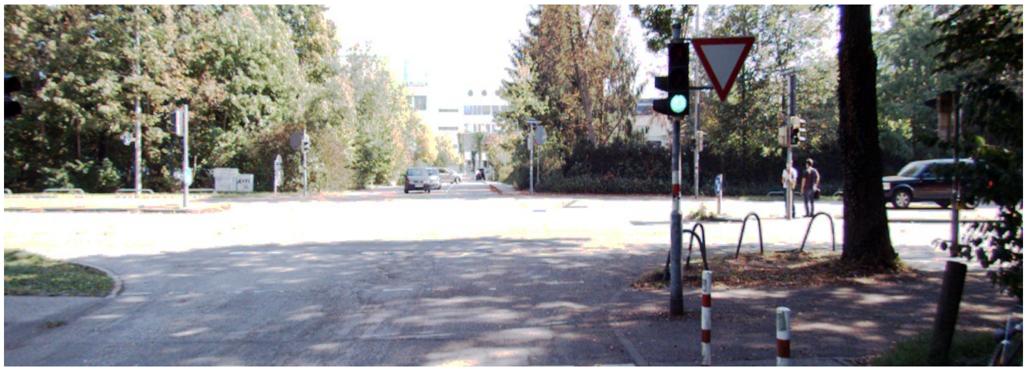
Problem - 1

- The given images/images are Highly Saturated and have a smaller number of Dark Values and many light Values with a very few values between these values. To do Image Processing, we would ideally like all the regions in the image to be high contrast.
- Histogram Equalization is a method through which we can simultaneously brighten some dark values and darken some light values, while still using the full extent of the available dynamic range. That is, we would boost the values in the middle range to a higher value which in turn utilizes the full dynamic range. As our image is an over saturated image, there is a loss of information so we want to spread these intensity values to the whole dynamic range to extract information hence Histogram Equalization can be performed.

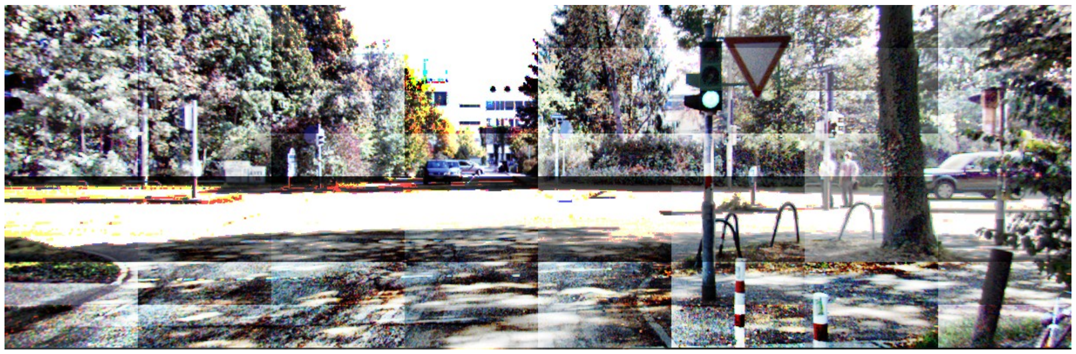
- As seen in the above graph, after performing the Histogram equalization, still most of the values are high intensities because there are very few pixels in between high and low in the original image, but we amplified these values by a certain amount.
- In the original image the dark values have an intensity like a steep curve and fall off, but we have distributed these values evenly over a certain range.
- To Enhance the contrast, we can partially compensate for the histogram unevenness, by a linear blend between the cumulative distribution function and original function. this can be seen in the below image which is 70 percent equalized with 30 percent retained from the original image.
- A potential drawback of Utilizing the full Dynamic range is that we could see colors which are little bit off as we enhance the intensities in each channel this might twist the colors as we can see in the below image where color intensities are present in Histogram image but not in original image. Another drawback is that noise in dark regions can be amplified and become more visible.

- While global histogram equalization can be useful, for some images it might be preferable to apply different kinds of equalization in different regions. Adaptive Histogram Equalization performs Histogram Equalization over a window thus considers the intensities only in the window. This gives us High contrast but the difference between the boundaries of objects in the image is accentuated as we can see in the image below.





Histogram Equalized Image

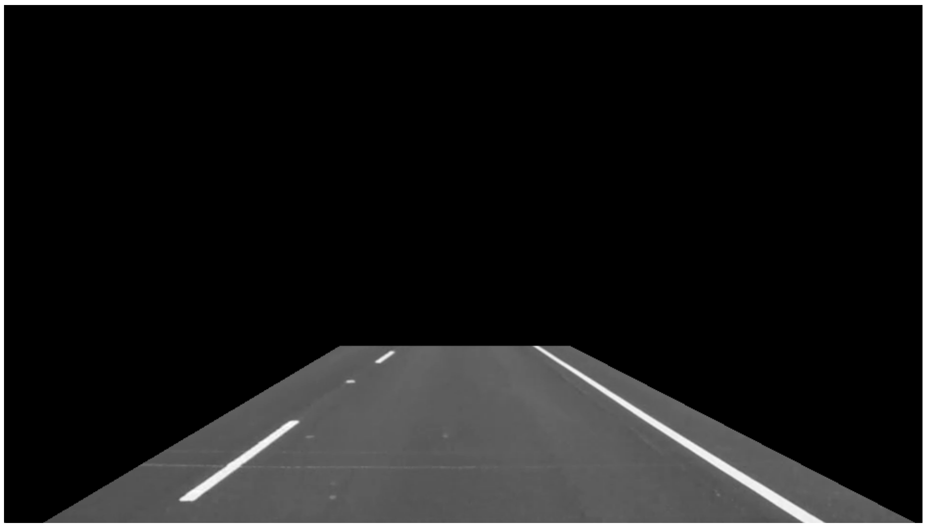


Adaptive histogram

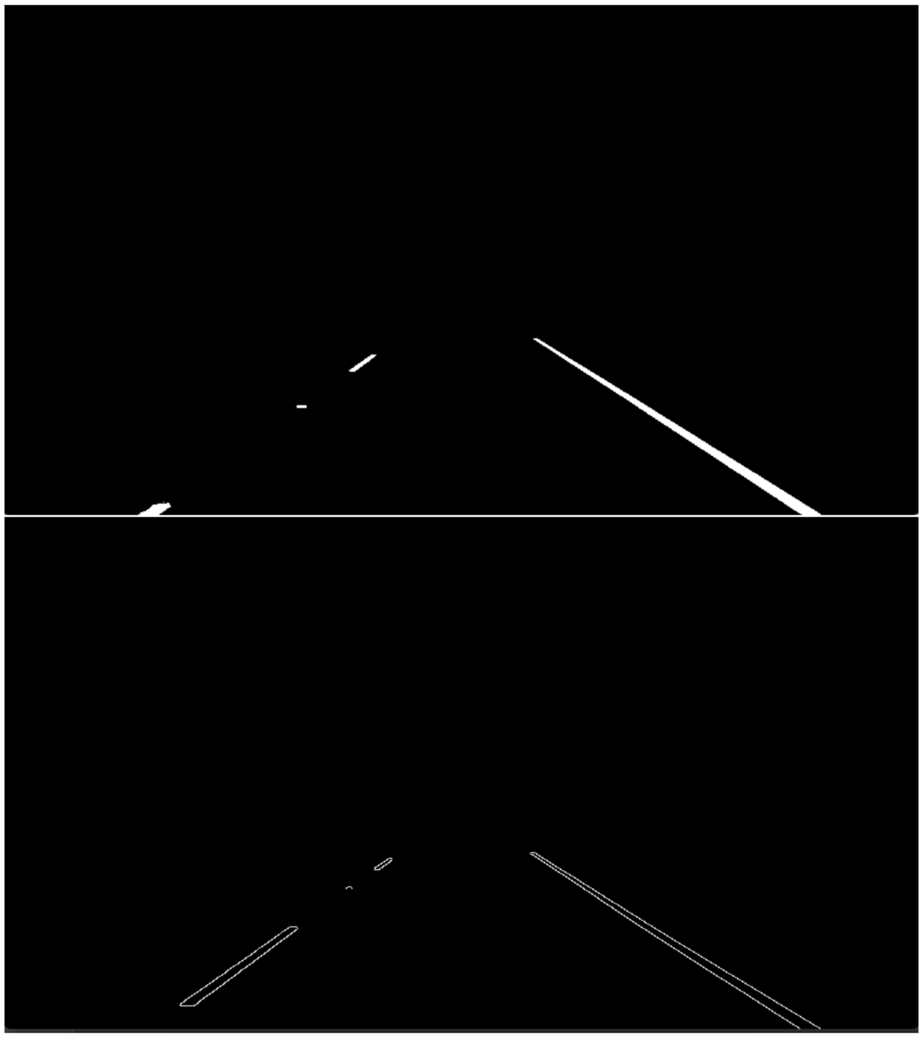
Problem 2

Consider the pipeline that is used:

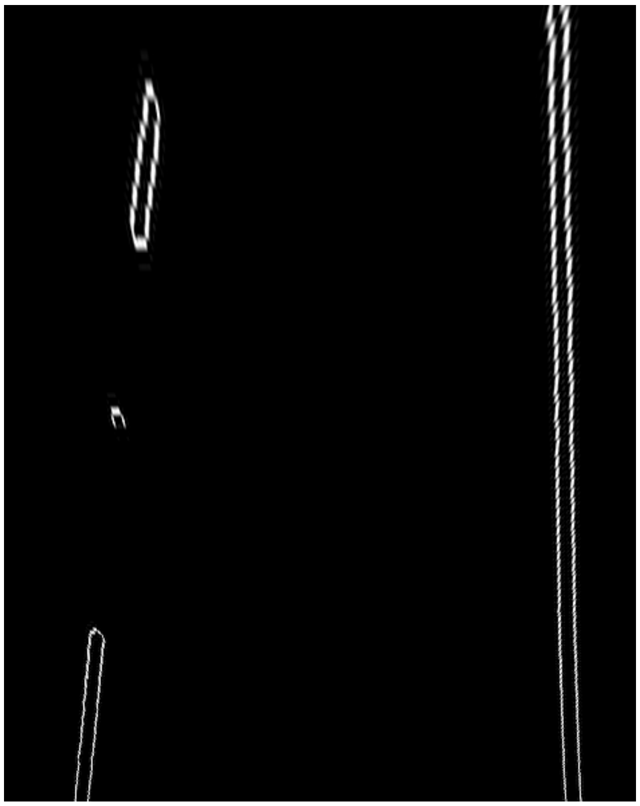
- The frame is converted to grayscale first and a polygon is defined with coordinates on the frame such that it isolates the region of interest which is in our case the lanes on the road. This is achieved by masking the image with the polygon and filling the rest of the area with zeros so that only the road and the lanes stands out.



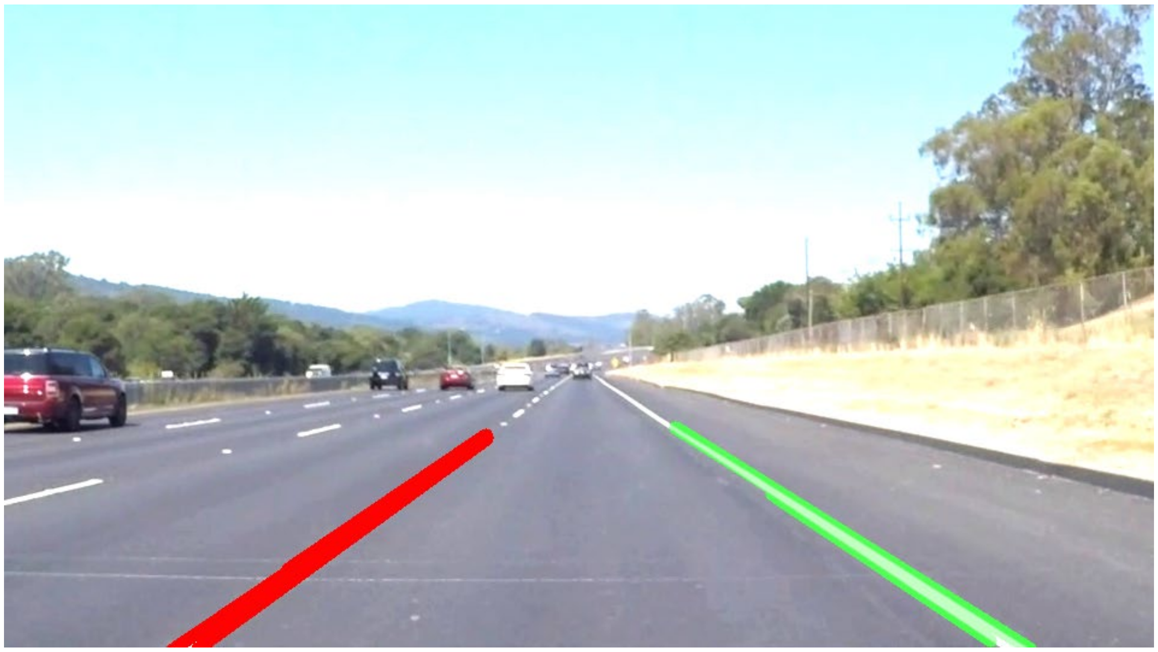
- Gaussian Blur is applied to this image and then binary thresholding is done with the image after which the canny edge detection is used to get the following output.



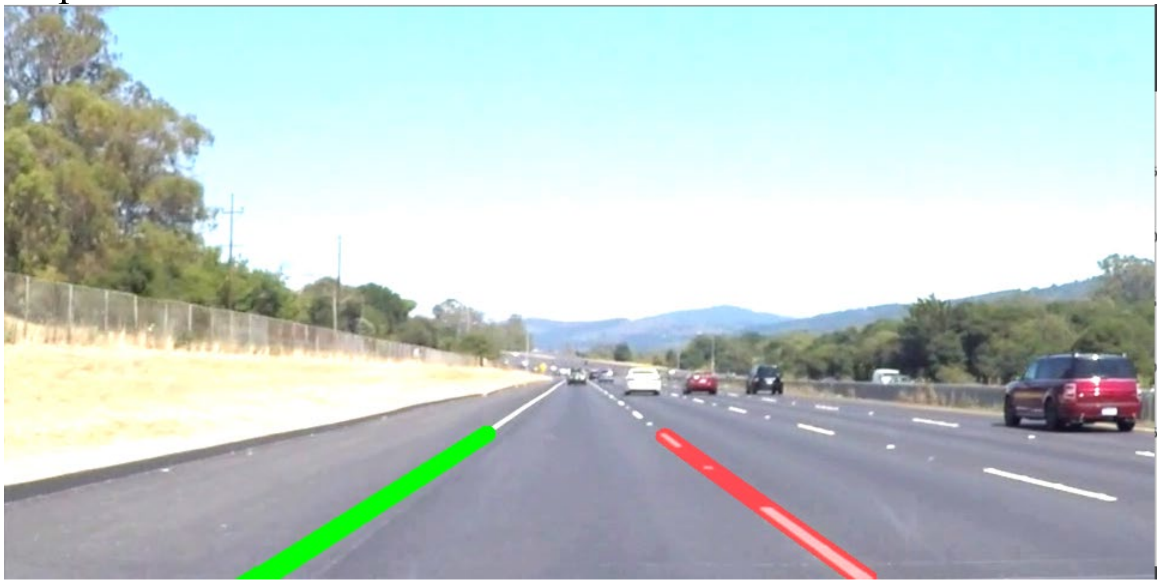
- To get the birds eye view of the frame, we get the transformation matrix from the inbuilt functions that we have in OpenCV and then warp it onto the window that we designate to get the following output.



- We take the histogram and sum the intensities on both the lanes and intuitively we know that the sum of the intensities of the dashed lane will always be less than that of the continuous lane.
- We draw the two lines using Hough Lines by giving proper parameters and assigning a color value and add the lines back onto the frame.



- The video has been flipped horizontally and tested to give the output as follows:



To get a best fit line joining points, we use Hough space where we project the coordinates in the cartesian frame onto the Hough space. Hough space is basically having an ordinate of the intercept values and the abscissa as the slope values. All family of lines passing through a point will be represented in the form of a line in the Hough space having the m and c values which are the slope and the intercept values.
Since the slope of the vertical line cannot be determined, we modify the cartesian frame in the parametric equation and then the m-c space becomes as sinusoidal wave. The point of intersection of the sinusoidal waves gives us the line of best fit.
We used Hough lines to get the best fit line on the lane here. I played around with the values of the parameters to get the best fit line.

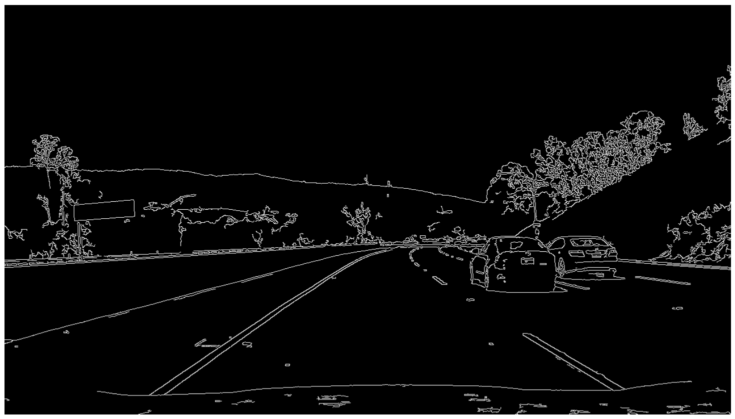
Problems faced:
The main problem was trying different methods to isolate the lane in the region of interest. The sequence in the image processing was the challenge as I had umpteen ways of trying different things but finally went with the best method which was highlighting the lanes.
I also wanted to try to represent the lanes with one single line where I wanted to calculate the average slope and represent the lanes with one single best fit line.

This method will generalize well if the lanes are straight since Hough lines are used and we can only get straight lines out of them. These might not work when there is a curvature to the road and we will end up seeing the straight lines even on curved lanes.

Problem-3

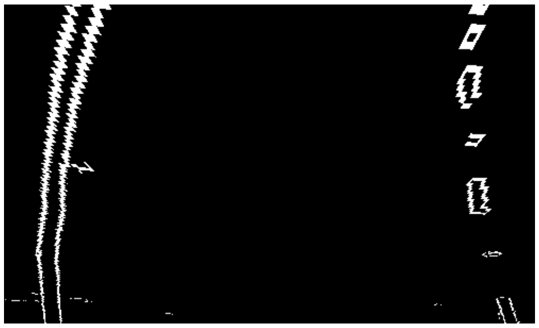
Consider the following pipeline used for this problem:

- Knowing that the yellow lane will be detected better in a different color map. I tried separating the channels and viewing it just to get the best standout image of the lanes.
- Sobel operator was my first choice in the list of my experiments, but the edge detection was not as good as canny which seemed to identify the edges better as it had a lot of noise. Canny, with proper thresholding gave me the best detection of lines.
- I have used the gray channel and have done thresholding to perform the lane detection. The output is as follows:

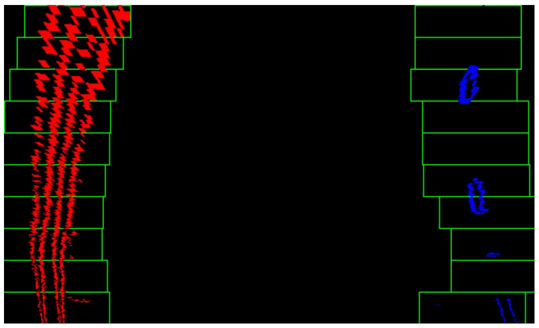


- The region of interest has been isolated by careful observation and perspective transformation has been done just to convert it to the bird’s eye view. The curved lanes are better visible from the bird’s eye view.

- The following is the warped gray scale image



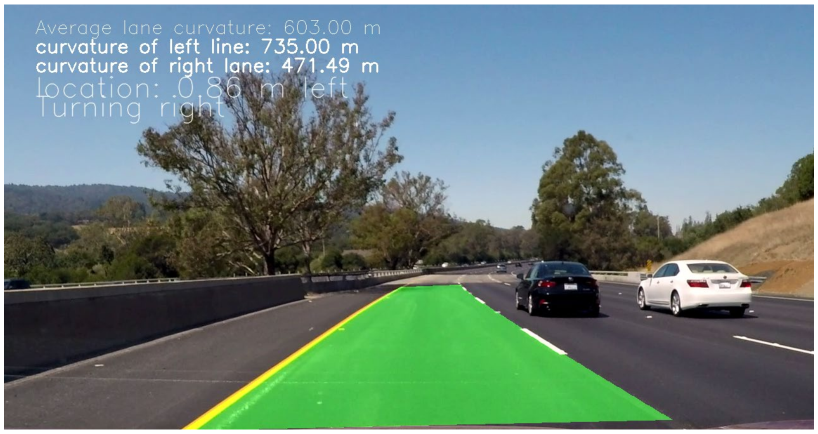
- I have taken the sum of intensities in the warped image column wise. We take the histogram and sum the intensities on both the lanes and intuitively we know that the sum of the intensities of the dashed lane will always be less than that of the continuous lane. But to detect the curved lanes, the coordinates of the maximum values along the columns are taken and the sliding window method is used wherein a sliding window is considered from bottom to top and this window is shifted to the left or right based on the mean of the previous window.
- The following is the sliding window output in the S-channel.



- Now we get the pixels in the s-channel. Through the Sliding window we have the pixels coordinates of lanes. To fit a curve, we need a Polynomial.
- We use cv2.polyfit function to fit our polynomial. We get the coefficients of the polynomial from our function.
- The values of the coefficients a,b,c are obtained by polyfit and we substitute all y values that is 0 to 720 and get the corresponding x location.
- We use cv2.fillpoly to fill the area between these lanes.

- The image coordinates are converted to real world distances and are used to calculate the curvature.
- We can now predict the turn to see if left curvature is greater than the right then we go right and vice versa.

The following is the final output.


Average lane curvature: 603.00 m
curvature of left line: 735.00 m
curvature of right lane: 471.49 m
Location: 0.86 m left
Turning right

This method will generalize well on curved lanes as it takes the sliding window method in there is a prediction of the turn. However, I had big trouble in detecting the lane when the car is entering the bridge in the sunlight. The lanes were not detecting well and then I tried going ahead with the S-channel which seemed robust even under the sunlight.