Advanced Lane Detection Project4

#NOTE: Most of the effort was put in tuning the parameters to make the lane finding possible with descent radius of curvature detection.
Files: Output_images- tested images
Project.ipynb
output.mp4

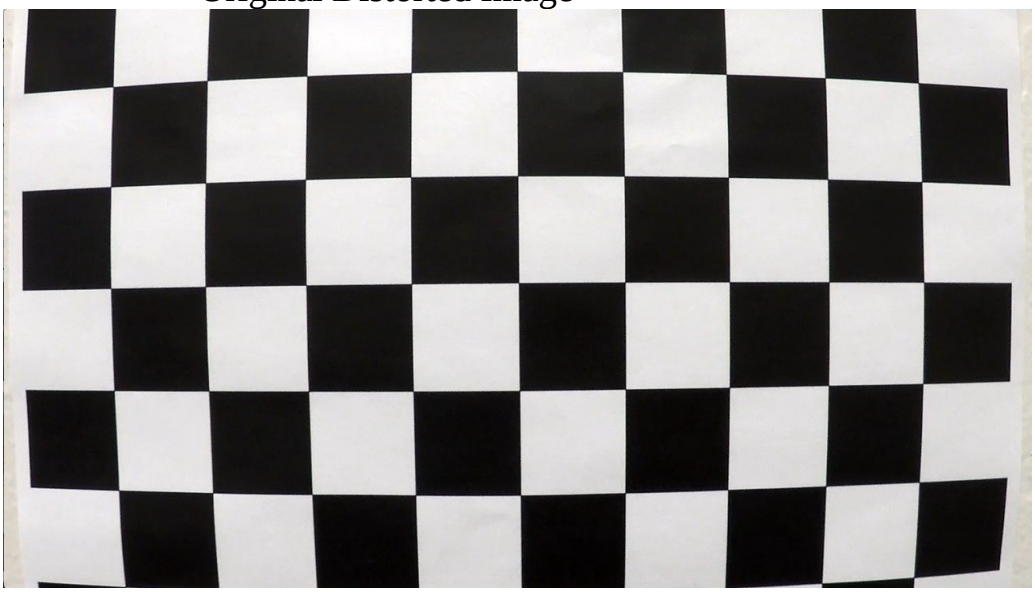
Advanced Lane detection Project involves following steps:

- Camera Calibration
- Color/Gradient Thresholding
- Perspective Transform
- Finding Lane Line
- Computing radius of Curvature

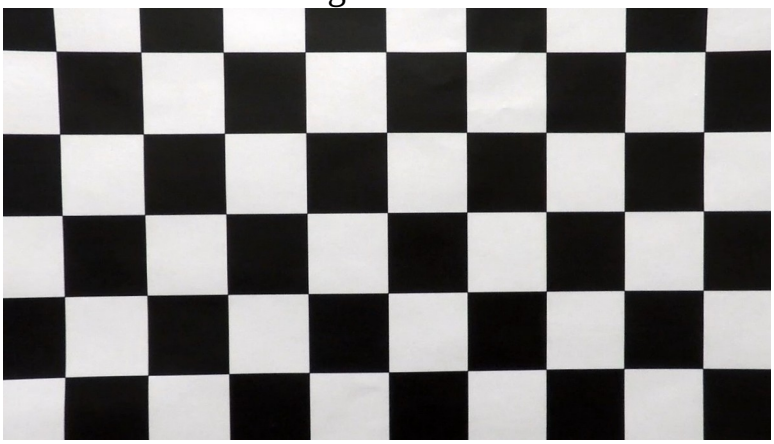
Camera calibration:

A chess board is used to calibrate distortions that occur during capture of images caused by lens. Transform matrix and the distance measure is computed using opencv library for a chess board and the same matrix and distance metric can be used for any images to undistort the image. OpenCV uses the 9x6 corners to find the measurements. Here is an example rectifying the distortion caused by camera. This is shown in project.ipynb notebook. Refer to jupyter notebook for implementation.

Original Distorted Image

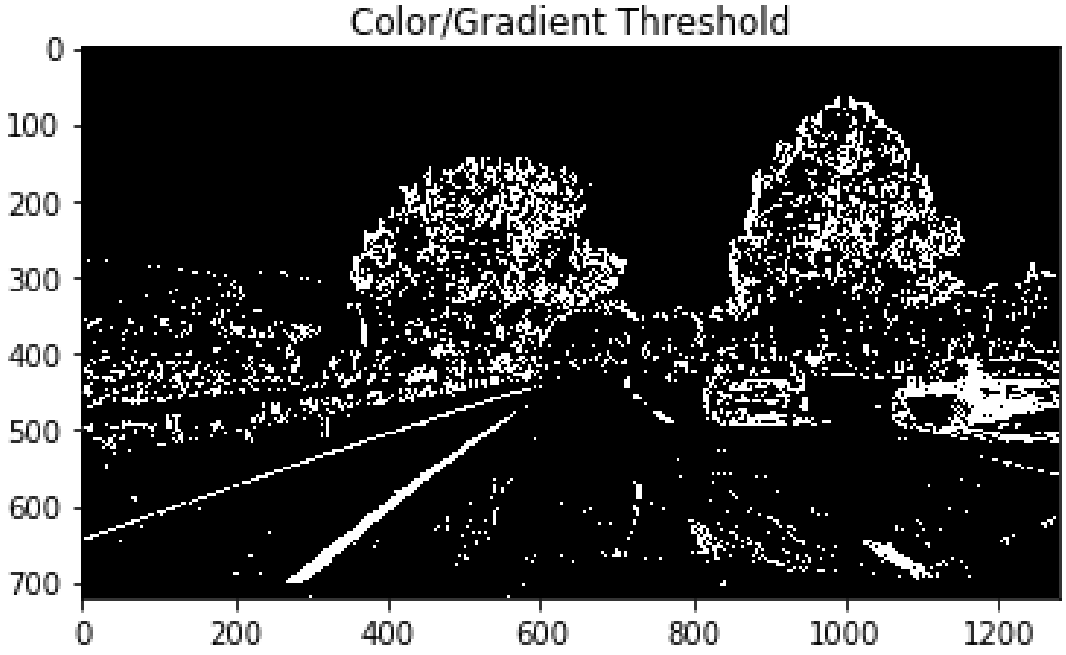


Undistorted using Camera Calibration



Color and Gradient Thresholding:

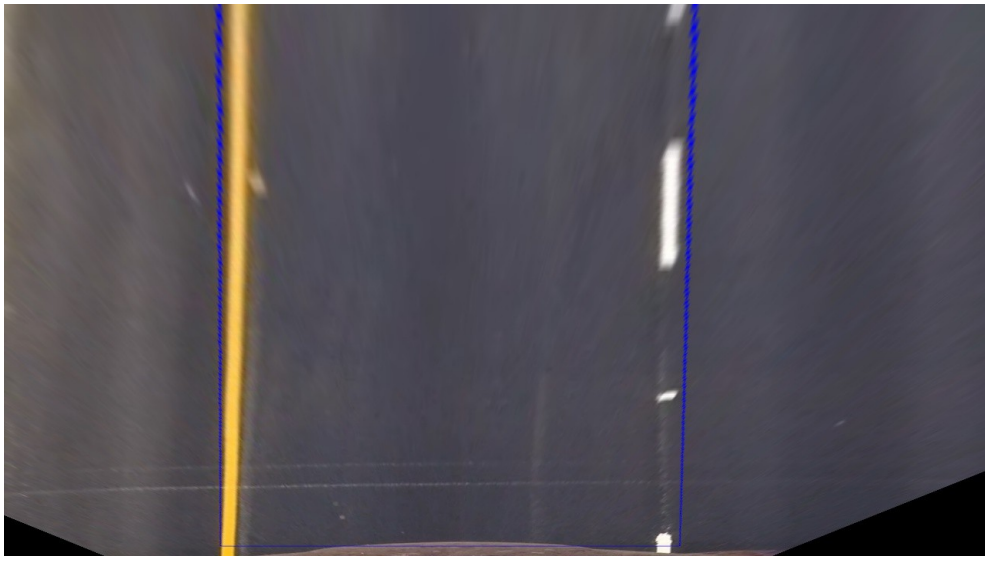
HLS based Color Thresholding has been employed. The saturation channel is thresholded. Absolute Gradient method is employed. The parameters must be tweaked to get better thresholded image.



Perspective Transform:

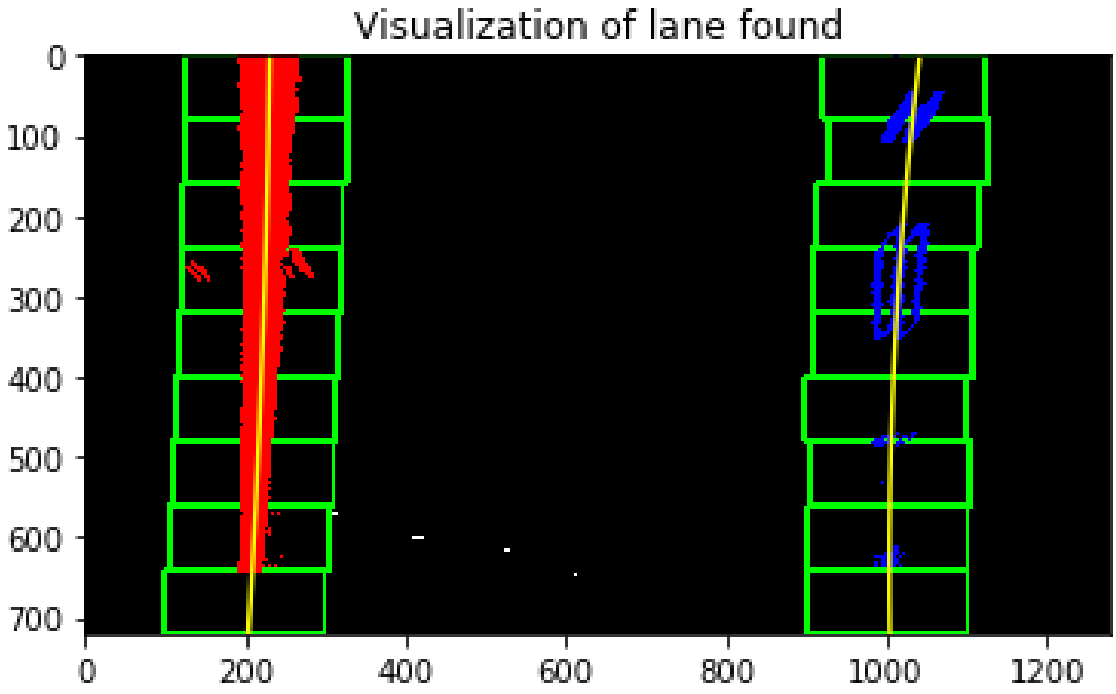
To obtain the top view of the lane lines four points are selected from the image and cast on to new four points mentioned in new image space.The method explained in class has been employed

Here is an example:



Finding Lane Lines:

The lane lines are found by window searching for how the image to obtain histogram of non zero pixels. The max values serve as initial centroids and then a windowed search is done as explained in class and as is shown below:



Computing the lane lines:

The radius of curvature formula as described in the class videos and material has been used and once the calculations are done the lane lines detected are inverse warped to fit on to the undistorted image so that the lane is marked and visible on the image as shown below

Applying the Pipeline to Video:

Once the parameters are fine tuned on to the test images the pipeline software is created with hard coded parameters to obtain the lane marked video with curvature indicated in the video as shown in **output.mp4**
The pipeline averages out the co-efficients of previous frames to obtain best fit and the window search is done only for one frame and subsequent 5 frames are not subjected to window searched to obtain the model rather the previous coefficients and binary warped image is used to calculate new lane models.

Discussion:
Challenges faced and tackled:

- Considerable amount of time was needed in fine tuning parameters
- Observable difference in deviation of car from the center lane when approximation method (using previous best fit curve and margin to find lane in new image) was used. Though this was reduced by consolidating to previous data, a considerable change can be made in the approach for finding best fit to avoid huge differences in deviation of car from center lane

Potential Challenges:

- Untested roads- The video on which the pipeline was tested worked fine but it will fall on rough terrain roads, roads without lane lines indicated as this computer vision based pipeline was hard coded with parameter which varies in different situations.

- Vehicle in the Front- In the video shown there is no vehicle in front of the car so this technique of thresholding and perspective transform worked but in heavy traffic situations thresholding and perspective transform wouldn’t work.
- Snow/Dirty Roads- Given that the road is flat and there are no vehicles in the front this pipeline can still fail under snow /dirt covered roads. So this may work in California but not in places like Chicago, Boston or Alsaka during winter!