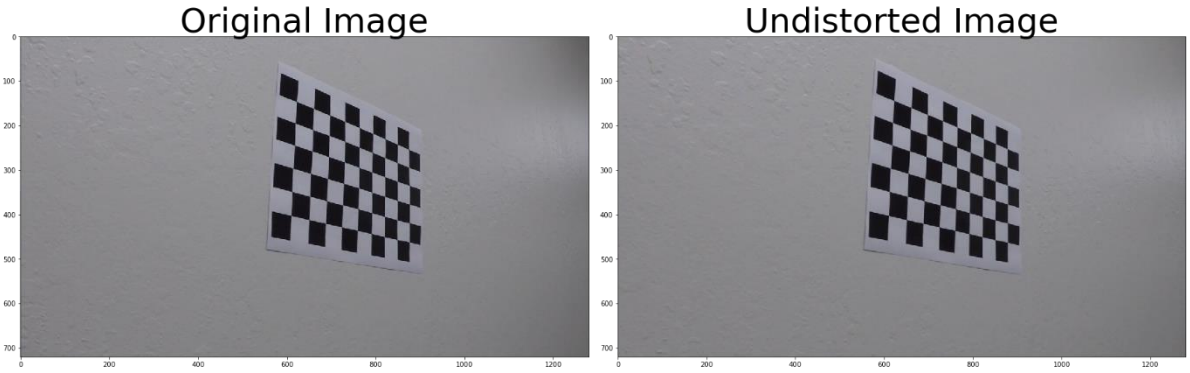
Project 4 Write Up

1. Briefly state how you computed the camera matrix and distortion coefficients. Provide an example of a distortion corrected calibration image.

The code in project 4-1.ipynb contains the code for this section. I read in the calibration images using the glob function due to multiple images. The corners were 9x6, unlike the lesson which had 8x6. I created 2 arrays (Objpoints and imgpoints) to store the data. Imgpoints is for the 2D points in distorted image and objpoints is for the 3d array in real world. The arrays are empty to be appended in the pipeline. I initialized the coordinates to zero using np.zeros. I converted to grayscale and using the findchessboardcorners function, I found the corners and calibrated using the cv2.calibratecamera function. I undistorted using the cv2.undistort and it undistorted the image such as the below calibration image.



###Pipeline (single images)

####1. Provide an example of a distortion-corrected image.

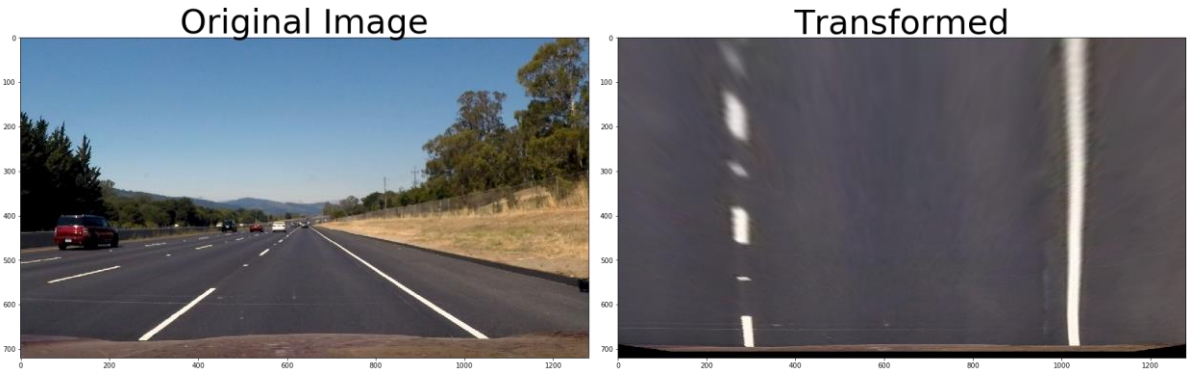
I ran this through the project4-1.ipynb notebook mentioned above to undistort the original image below.

####2. Describe how (and identify where in your code) you used color transforms, gradients or other methods to create a thresholded binary image.

I created project4-2.ipynb notebook to show the combination of a Sobel threshold, gradient magnitude, and direction of gradient to show the combined image below. I used a threshold of 20-100 for the x and y orientation of the Sobel Threshold, 30-100 in the gradient magnitude and .7-1.3 for the directional gradient.

####3. Describe how (and identify where in your code) you performed a perspective transform and provide an example of a transformed image.

I created project 4-3.ipynb which has the perspective transform portion of my code. It resulted in the below transform:



The source and destination points are as follows:

src=np.float32([[250,700],[600,450],[685,450],[1050,700]])

dst=np.float32([[300,700],[300,50],[1000,50],[1000,700]])

####4. Describe how (and identify where in your code) you identified lane-line pixels and fit their positions with a polynomial?

I created project 4-4.ipynb which has the lane finding and fitting portion of my code.

####5. Describe how (and identify where in your code) you calculated the radius of curvature of the lane and the position of the vehicle with respect to center.

I created Project 4-5.ipynb code which shows the code to calculate the radius of curvature. The variables are left_curverad, right_curverad.

####6. Provide an example image of your result plotted back down onto the road such that the lane area is identified clearly.

Sure, Below!

###Discussion

####1. Briefly discuss any problems / issues you faced in your implementation of this project. Where will your pipeline likely fail? What could you do to make it more robust?

Unfortunately, the roads across our country is not as well defined as the roads in this course. Many roads do not have lines well painted. So how do you develop a path when all you have is one side of lane lines? Obviously, this code is only possible While (lanes=2). While Lane line is one side or less, the car would have to look at other times such as barriers to determine drivable path. I look forward to training our classifier on all the different styles of barriers 😊