

Here I will consider the rubric points individually and describe how I addressed each point in my implementation. The output images are saved in the folder output_images in the workspace. Project video is saved as Project_Video_Output.mp4 in the workspace.

1. Camera Calibration(see Fig1)
 - Find chessboard corners
 - Use the identified corners to compute calibration coefficient
2. Undistortion of test images(see Fig2)
 - Use the calibration coefficient in step 1 to undistort the test images
3. Thresholding(see Fig3)
 - Used a combination of color and gradient thresholds to give a binary image as an output that distincts lane lines. Used trial and error to find the threshold parameters.
4. Perspective Transform(see Fig4)
 - Transformed the binary image in step3 to a birds eye view image using a function from opencv library. I identified a set of source and destination points through trial and error that gave close to parallel lines.
5. Identifying lane lines(see Fig5)
 - Drew an histogram to find the peaks, and using the top two peaks found the starting point of the two lanes.
 - Setting the hyperparameters of window length and margin, found the indices of left and right lane pixels.
 - Fit the left and right lane pixels using a polynomial
6. Highlighting the area around the identified lane lines(see Fig6)
 - For the next frames of a video, to avoid doing the sliding window search again, a region around the identified lanes(with some margin) were identified.
7. Calculating the Radius of Curvature
 - Wrote a function to find the values of radius of curvature and offset from centre and converted those values to meters.
8. Draw an image with the boundaries drawn(see Fig7)
 - Drew the image back with the lane lines drawn using cv2.getweighted() function
9. Created a single pipeline that takes an image and outputs an image with lane lines drawn.
10. Applied the created pipeline to a video.

Possible issues and how can I make the pipeline more robust:

1. The pipeline searches for the lane lines from scratch in each frame. This can be inefficient and searching within a margin of the previous frames can help.
2. For the challenging video, exploring different color spaces and finding better threshold values can help.

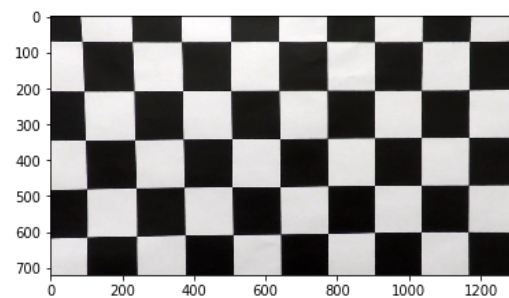


Fig1

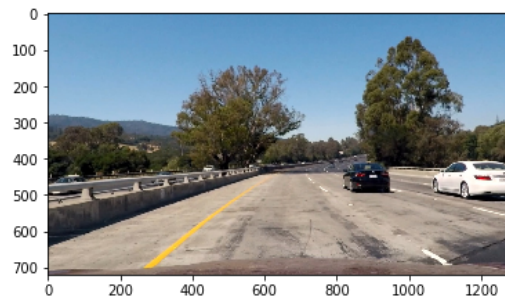


Fig2



Fig3

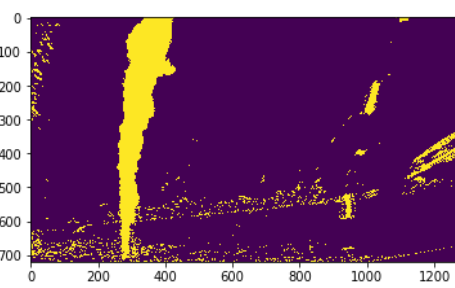


Fig4

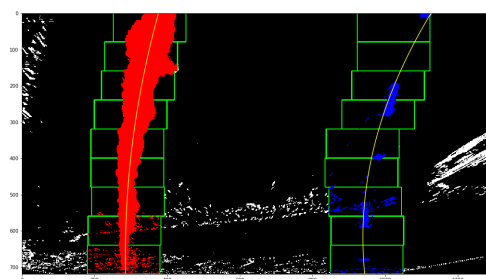


Fig5

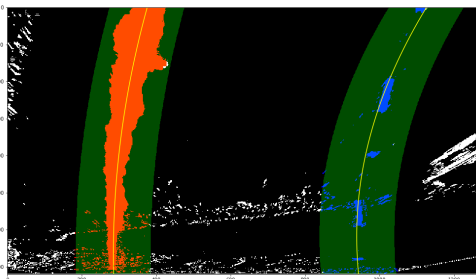


Fig6

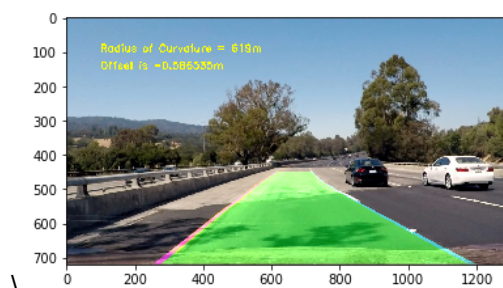


Fig 7