Advance Lane Finding Project

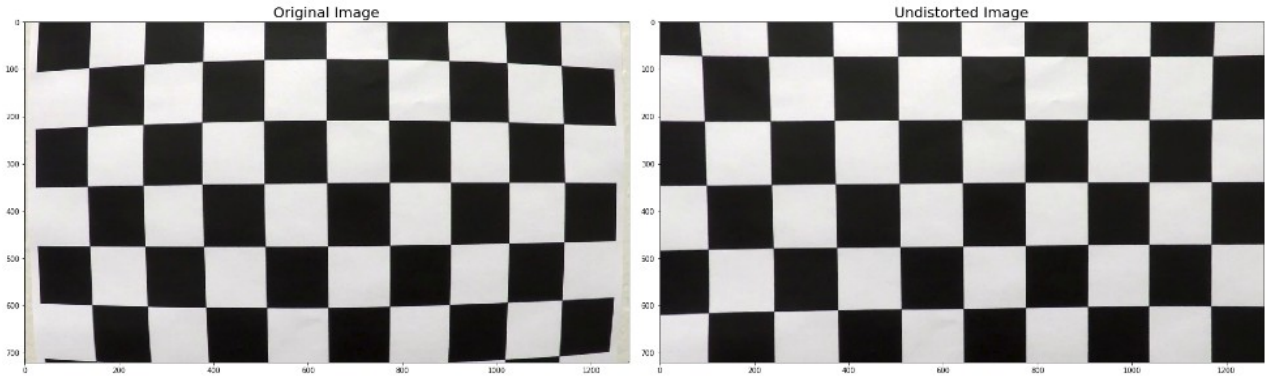
The goal of the project were as follows:

- Compute the camera calibration matrix and distortion coefficients given a set of chessboard images.
- Apply a distortion correction to raw images.
- Use color transforms, gradients, etc., to create a thresholded binary image.
- Apply a perspective transform to rectify binary image ("birds-eye view").
- Detect lane pixels and fit to find the lane boundary.
- Determine the curvature of the lane and vehicle position with respect to center.
- Warp the detected lane boundaries back onto the original image.
- Output visual display of the lane boundaries and numerical estimation of lane curvature and vehicle position

Camera Calibration:

1. Have the camera matrix and distortion coefficients been computed correctly and checked on one of the calibration images as a test?

I start by preparing "object points", which will be the (x, y, z) coordinates of the chessboard corners in the world. Here I am assuming the chessboard is fixed on the (x, y) plane at z=0, such that the object points are the same for each calibration image.
Thus, objp is just a replicated array of coordinates, and objpoints will be appended with a copy of it every time I successfully detect all chessboard corners in a test image. imgpoints will be appended with the (x, y) pixel position of each of the corners in the image plane with each successful chessboard detection.
I then used the output objpoints and imgpoints to compute the camera calibration and distortion coefficients using the cv2.calibrateCamera() function.
I applied this distortion correction to the test
image using the cv2.undistort() function and obtained this result:

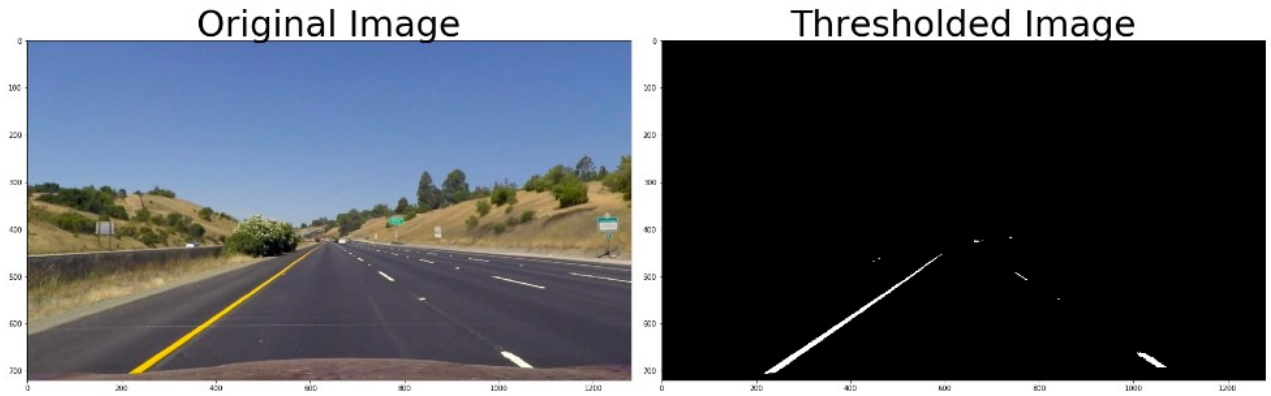


Pipeline (single images)
1. Has the distortion correction been correctly applied to each image?

undistort is particularly noticeable by the change in shape of the car hood

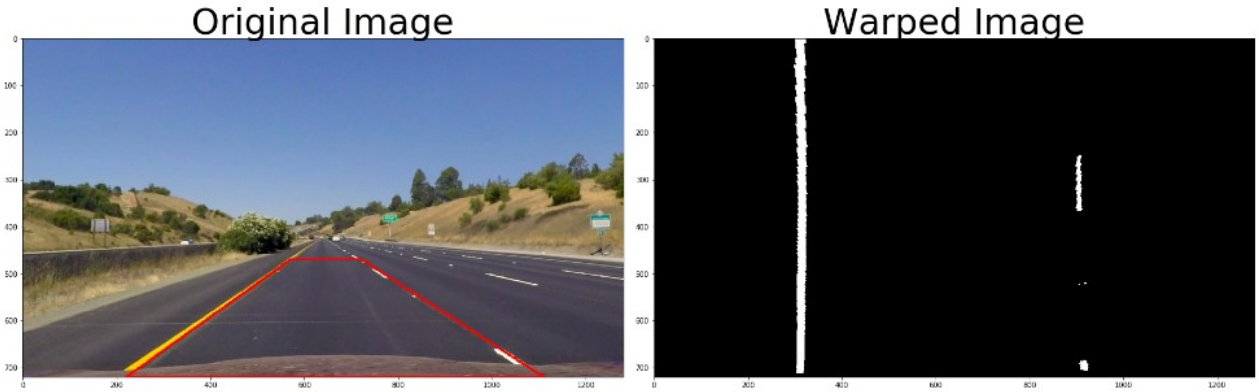
2. Has a binary image been created using color transforms, gradients or other methods?

Yes it is created with masking.



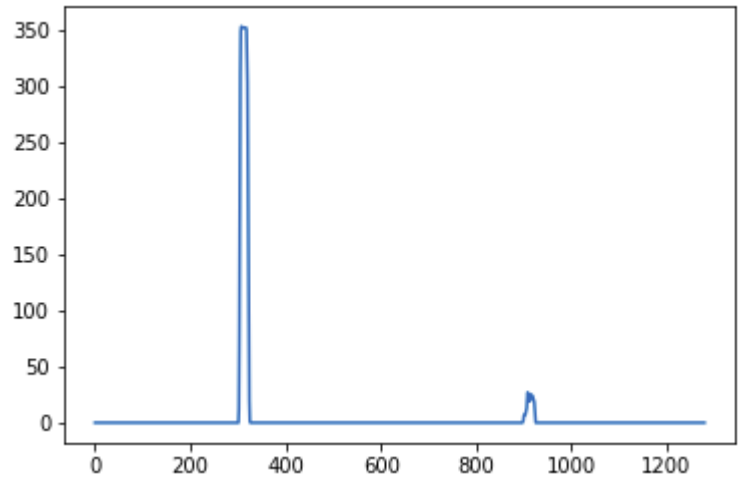
3. Has a perspective transform been applied to rectify the image?

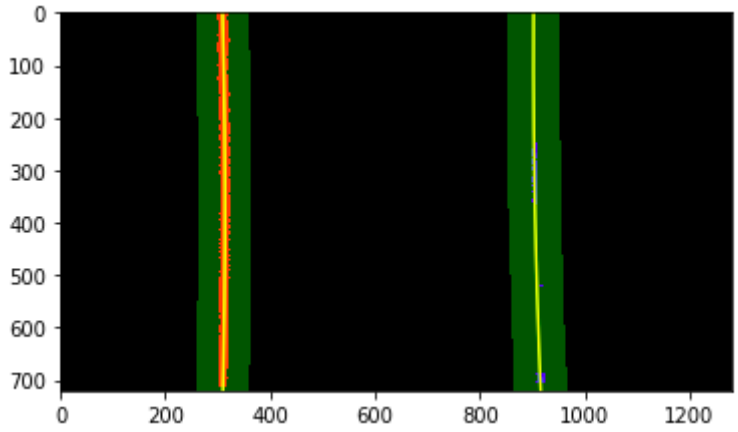
Yes perspective transformation been applied to rectify the image with bird’s eye view



4. Have lane line pixels been identified in the rectified image and fit with a polynomial?

First historgram was created to identify the lane pixels and then sliding window approach was applied with polynomial fit.





5. Having identified the lane lines, has the radius of curvature of the road been estimated? And the position of the vehicle with respect to center in the lane?

Yep, sure did!

Pipeline (video)
1. Does the pipeline established with the test images work to process the video?

There is a video file with name `project_video_output.mp4`

Discussion

Gradient & Color Thresholding

I experimented with gradient and color channnel thresholding. R & G channel were good in identifying lanes, silimarly S and L channel were also performing well hence I took mix of them.

The lanes lines in the challenge and harder challenge videos were extremely difficult to detect. They were either too bright or too dull.

Points of failure & Areas of Improvements
The pipeline seems to fail for challenge video and harder challenge video.

This video has tunnel and harder challenge has sharper turns.

For improving it, I think I need to work on perspective transform and sliding window approach.