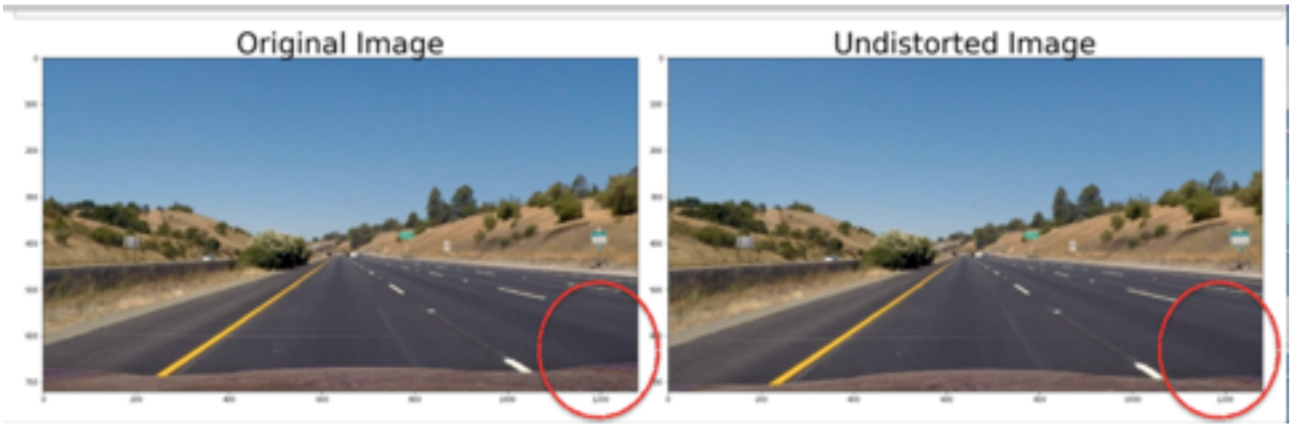
ADVANCED LANE FINDING

Pipeline

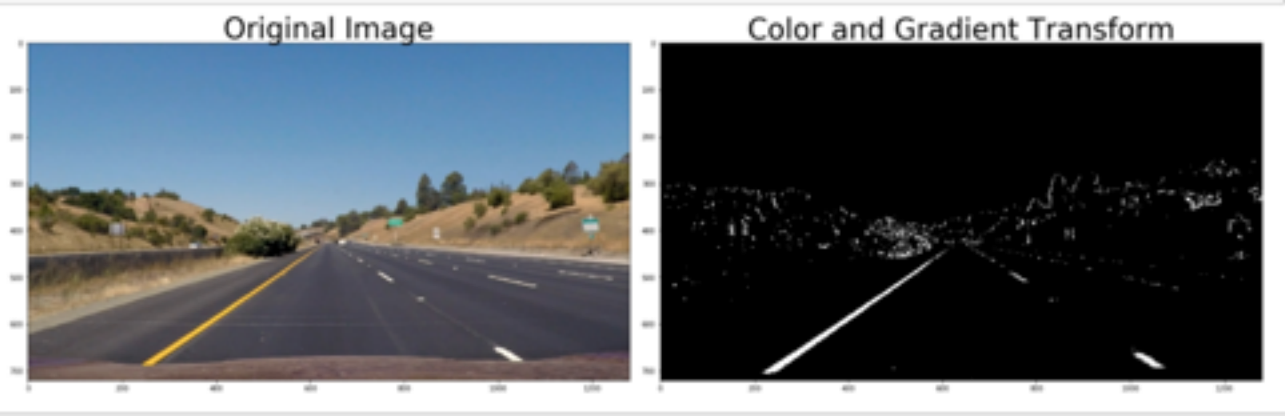
1. Camera Calibration using Chessboard images (9x6 corners)
   Using a set of 20 Images.

2. Apply Distortion correction to raw images using cv2.calibrateCamera functions.



3. Color and Gradient Tranform

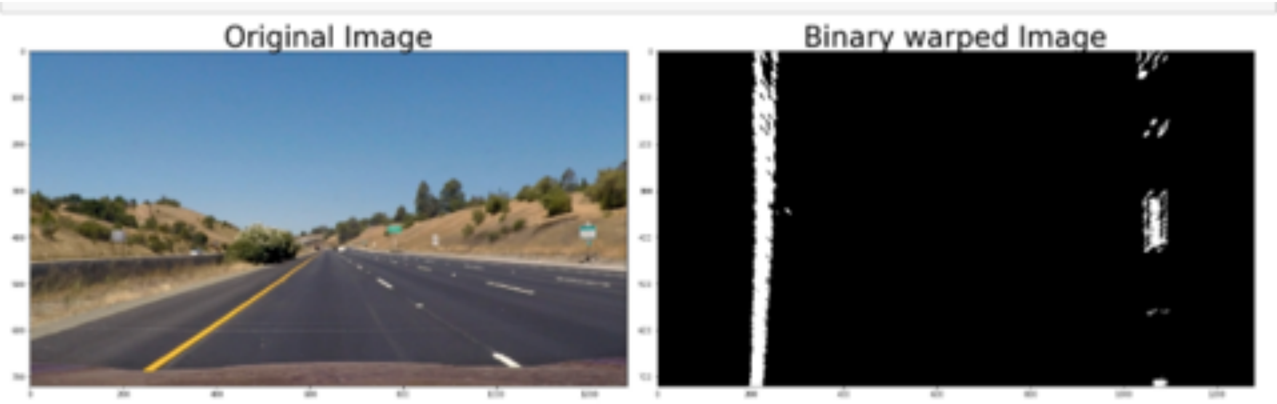
Used a combination of thresholding along
Sobelx (Gradient along X direction) and
S(Saturation) threshold in HSV color space to bring out the lanes.



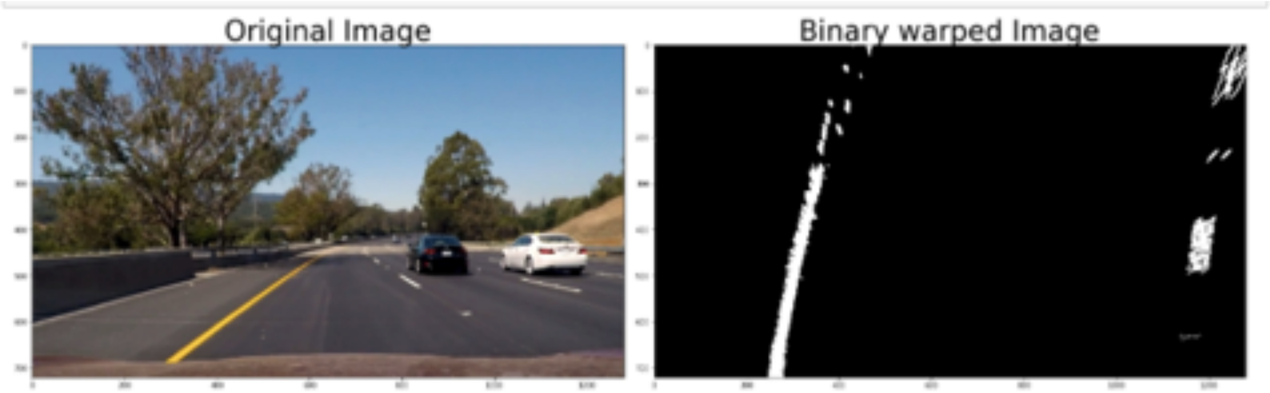
3. Perspective Transform:

Used the straight line images to identify the SRC and Destinaltion pixels and create a Perspective Transformation using CV2 functions.

Straight Lines:



Curved Lines:



4. Detecting Lane Lines in first frames: Using Histogram.

Detecting the strong edges on both half in the x direction. Identify the X coordinate for the lanes.

Use a sliding window to detect the lane with a margin width from bottom to the top of frame.

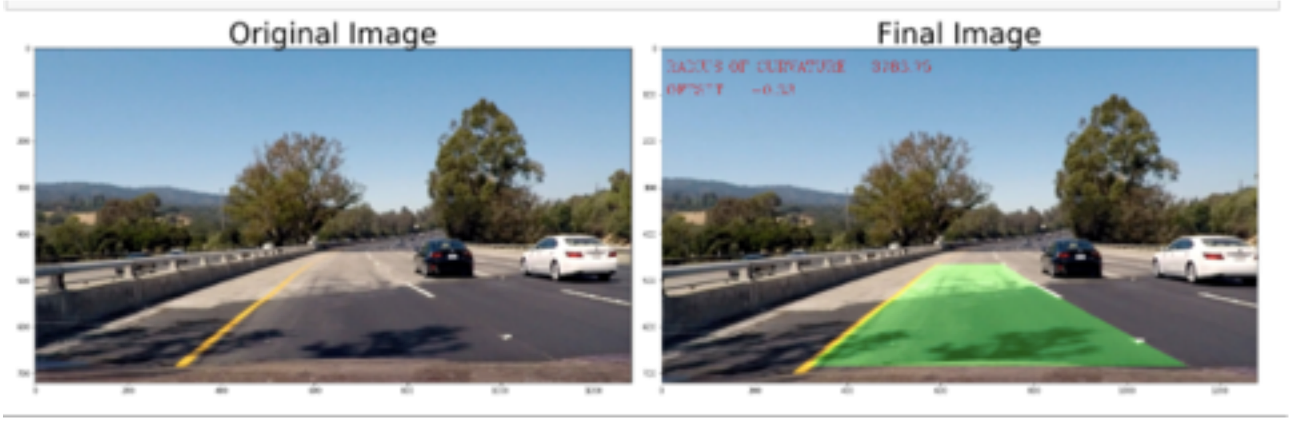
Finding the polynomial coefficients on lane detected.

5. Detecting further lanes in successive frames using the search with prior method. Where we need not use the window to scan the entire frame, but look for pixels next to the lane already detected in the previous frame. As the curvature will not change drastically.

6. Calculation of radius of curvature and offset and superimposing on the image. As per the formulas shared in the previous lessons.

7. Warp the detected lane boundaries back onto the original image.

8. Output visual display of the lane boundaries and numerical estimation of lane curvature and vehicle position.



RESETING: If the image is not able to identify lanes in left/right for consecutive 5 frames. It will use the histogram lane detection to identify the lost lane again.

You should add some final discussion to your writeup. We expect here that you will answer the following questions:

what problems / issues did you face in your implementation of this project?
where will your pipeline likely fail?
what could you do to make it more robust?
Could you please expand your final section answering these questions in more details?

Problems and Solutions.

1. In a particular segment of the video, where no lanes was detected due to temporary shadow. The lane tracking stopped completely after this point.

   Solved by calculating the lanes using histogram function again when consecutive frames are lost.

2. Radius of curvature was varying drastically in each frame. This was smoothened out by using exponential smoothing.

Review Finding corrected: Proper conversion made for pixel to metre conversion

Points to improve, Exceptions in current logic:

1. Current algorithm detects lane (White and yellow) in low lighting also.
But the algorithm might fail, when surface have different colours (Like concrete section/asbestos)

2. The algorithm is likely to not to detect lanes with large bends (Small radius or curvature) or drastic changes in lane curvature in consecutive frames (Hair Pin Curves). Because we are using search with prior approach.

Conclusion:
Further output video and Images can be found in the project submission. The algorithm is robust in low light and shadows in the image. And identifies curvature of the road and offset accurately.