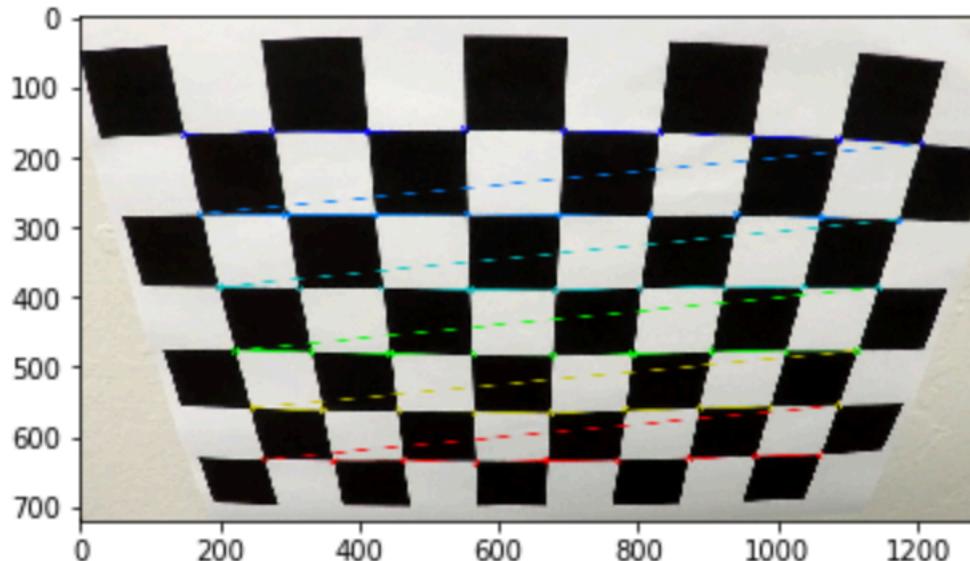
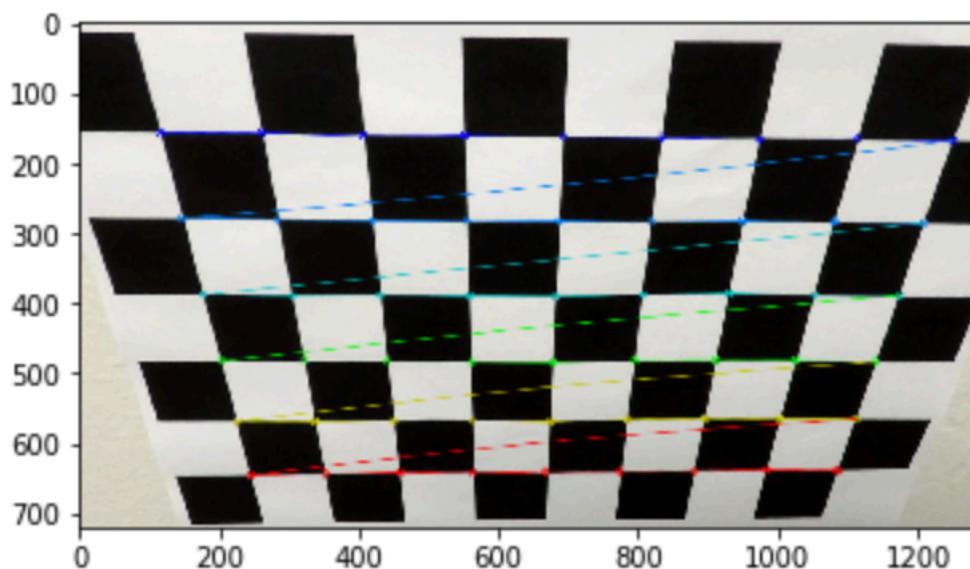


Camera Calibration

1. Use the chessboard "calibration2" because we can have the whole look of the 9x6 chessboard.
2. Turn it into gray, and use cv2.drawChessboardCorners to find corners

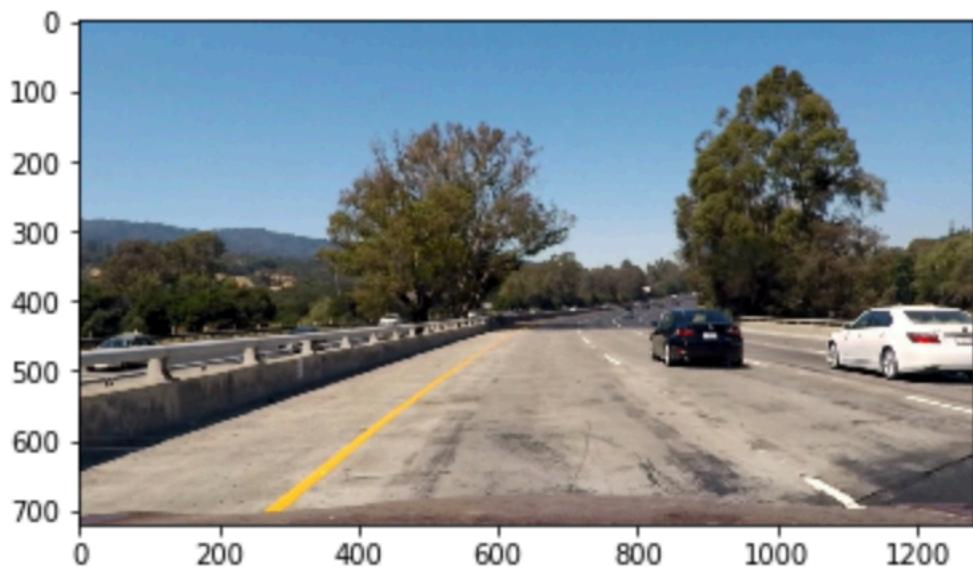


3. Use the objpoints and imgpoints from ret == True method, we can use cv2.calibrateCamera to find matrix that we can use to distort images.
4. Use cv2.undistort with mtx from last step.

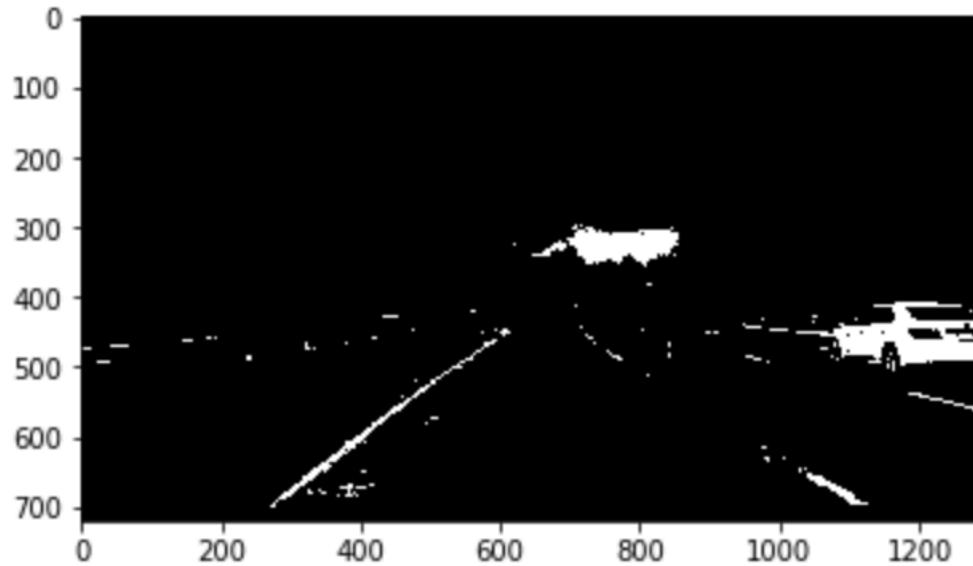


Pipeline (test images)

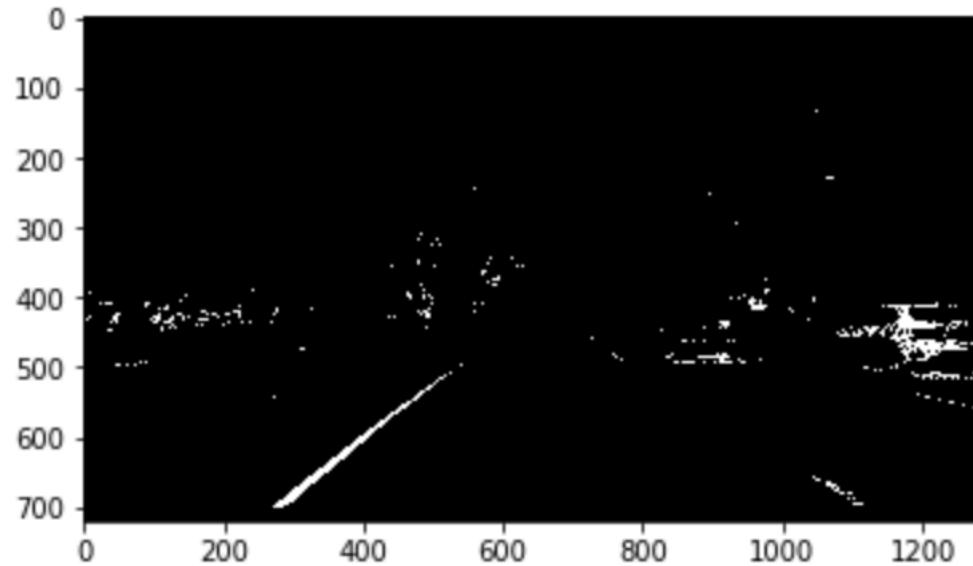
1. Undistort the image using matrix from chessboard calibration step.



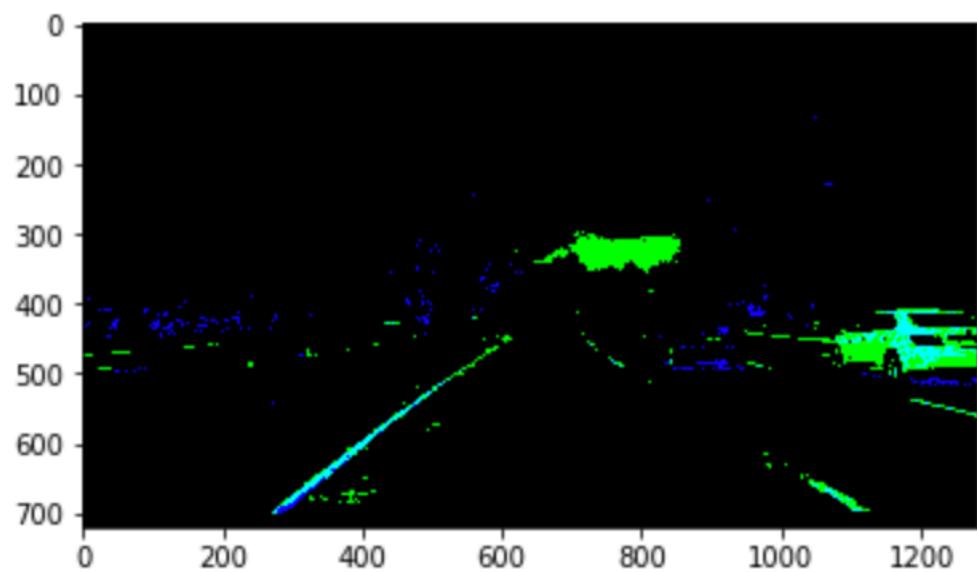
2. Apply gray mask with a threshold of 200~250. A bigger low limit will lead to more noise from upper lane that is not clear.



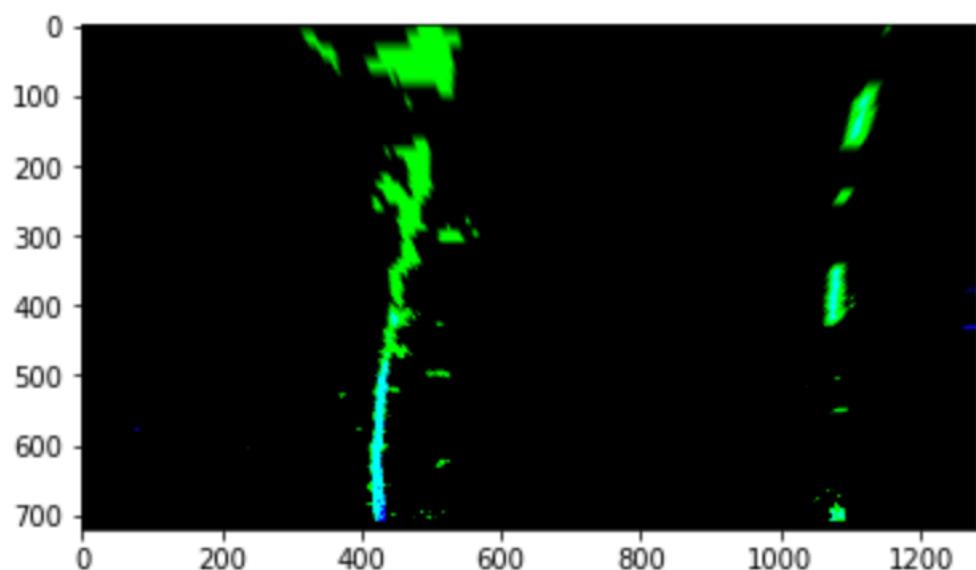
3. Apply saturation mask using s_channel from HLS transform method. The threshold is 230~250. The mask filter out most of the noises from trees and leave pixels with strong color.



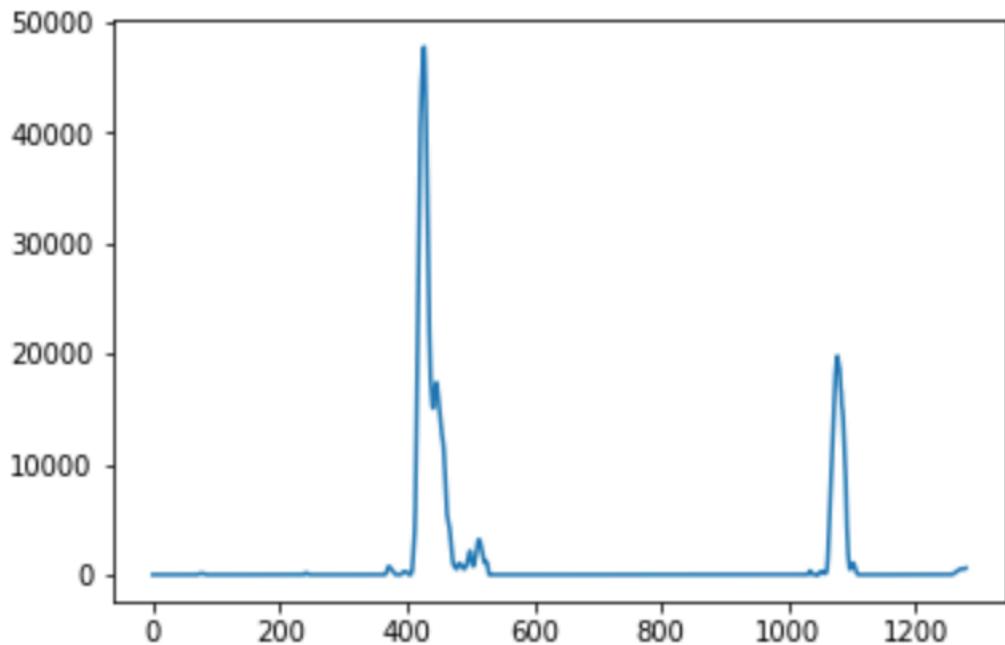
4. Apply the gray mask and saturation mask together.



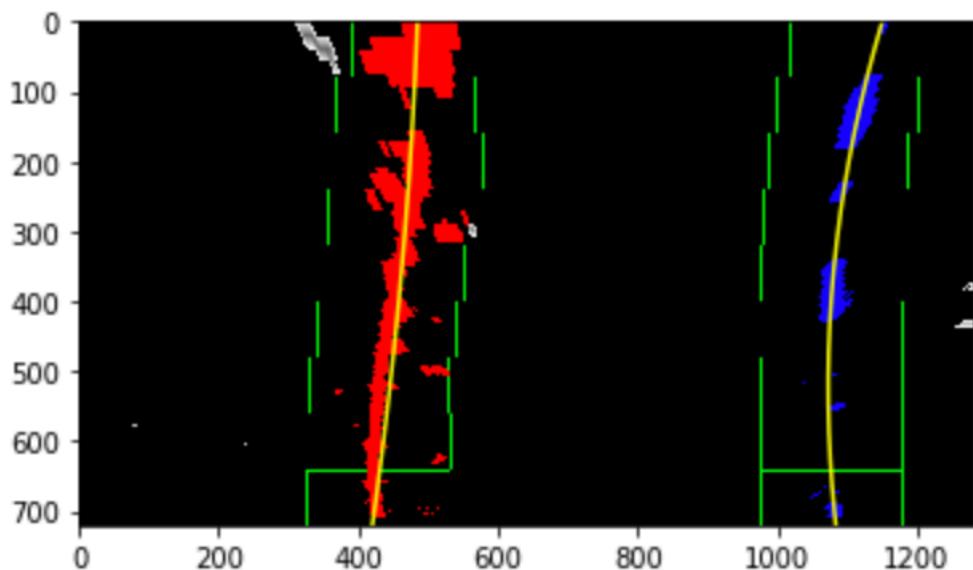
5. Apply the perspective transformation using cv2.warpPerspective. Before doing this, I need to set up source points and destination points, which is a important factor of transformation quality.



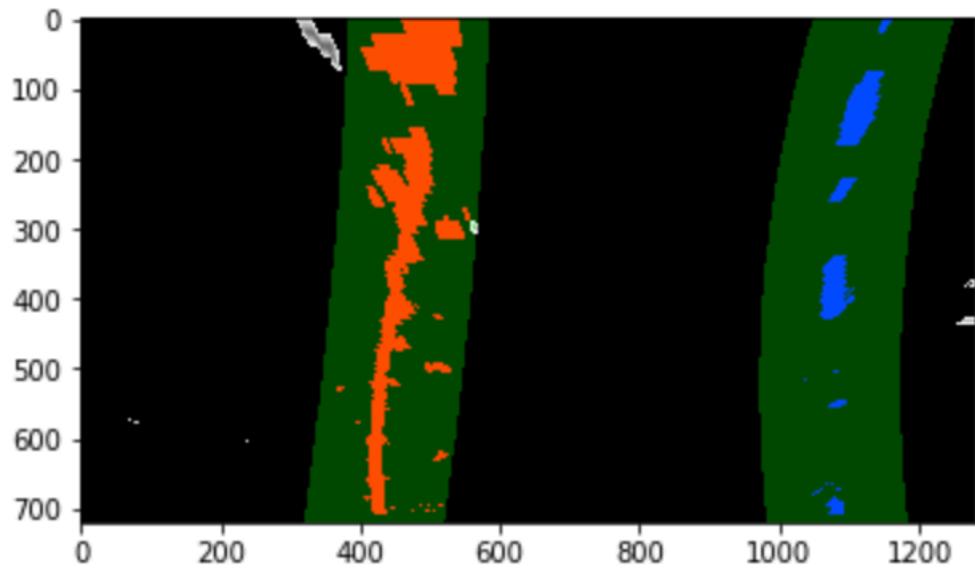
6. Use sum to plot a histogram of the diversity of pixels. Peaks in the histogram should be the lanes we are trying to identify. As shown in the Udacity materials, we divide it into two parts and assume the peaks(originated from argmax method) are lanes.



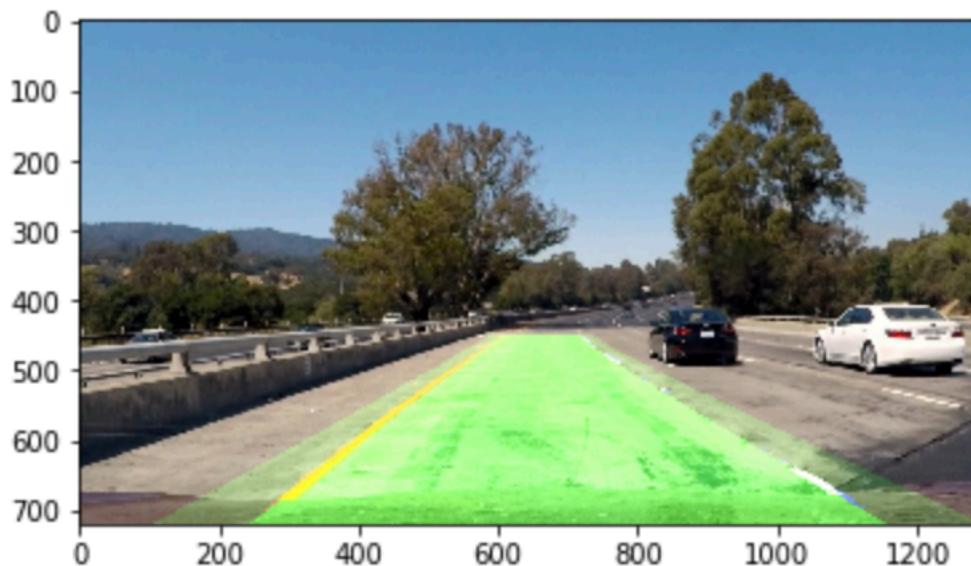
7. Use windows method to identify the lanes pixels. As shown in the Udacity materials, we started from the button(y) and peaks(x), go up window by window and rectangle in pixels inside the windows. We pre set a minpix to ensure that our windows will focus on the assumed lanes. Finally, we use all the collected points to fit poly lines and acquire the weights of the poly lines.



8. Draw the search area by drawing 4 lines from poly weights acquired earlier with x combinations with margins.



9. Use cv2.warpPerspective to transform the image from 8 to real image. We need to use the Minv created from cv2.getPerspectiveTransofrm. Use cv2.fillpoly to draw the rectangle between lans.



10. Curve radiation are calculated using method from Udacity materials and adjusted according to the meters per pixel index.

11. Make the video

