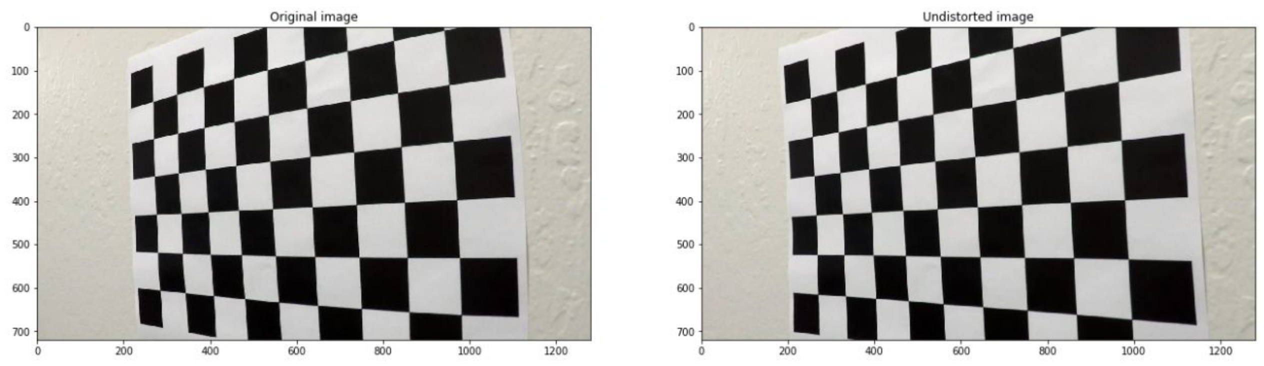
Project №2

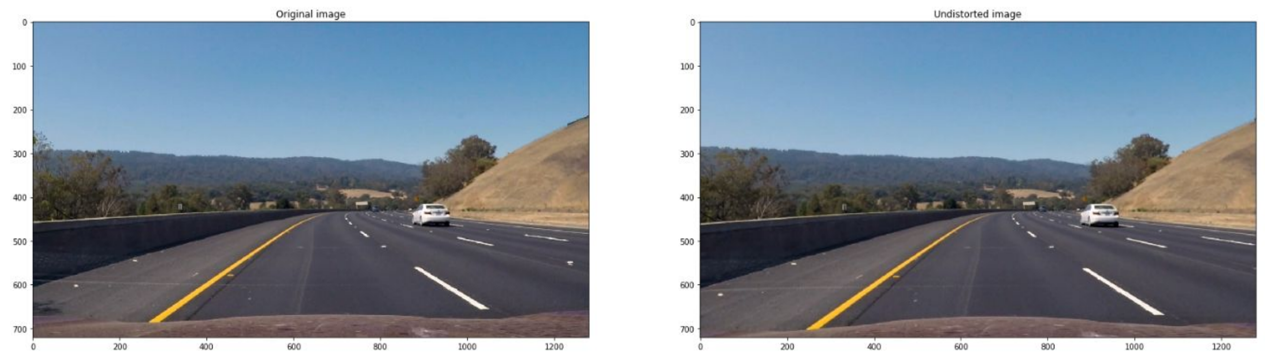
Advanced Lana lines finding

Following steps were applied in order to receive an algorithm, that finds lane lines marking:

1. Camera Calibration

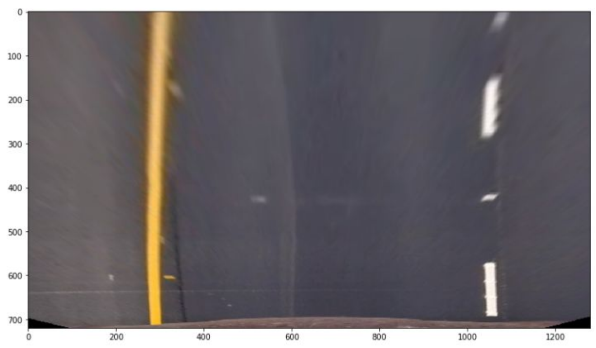
The pipeline for calibration is in “Camera Calibration.ipynb” file. There i have defined a 9x6 points on example images of chess boards. Then were created few empty lists, for an object points and for an image points. All the defined points at chess board images were added to the image points list. At the same time an sequence of artificially created points (were the chess board points supposed to be) were added to the object points list. Due to those both lists and the funktion “cv2.calibrateCamera” was created an matrix (calibration_pickle.p). With that matrix i could undistort all the test images.





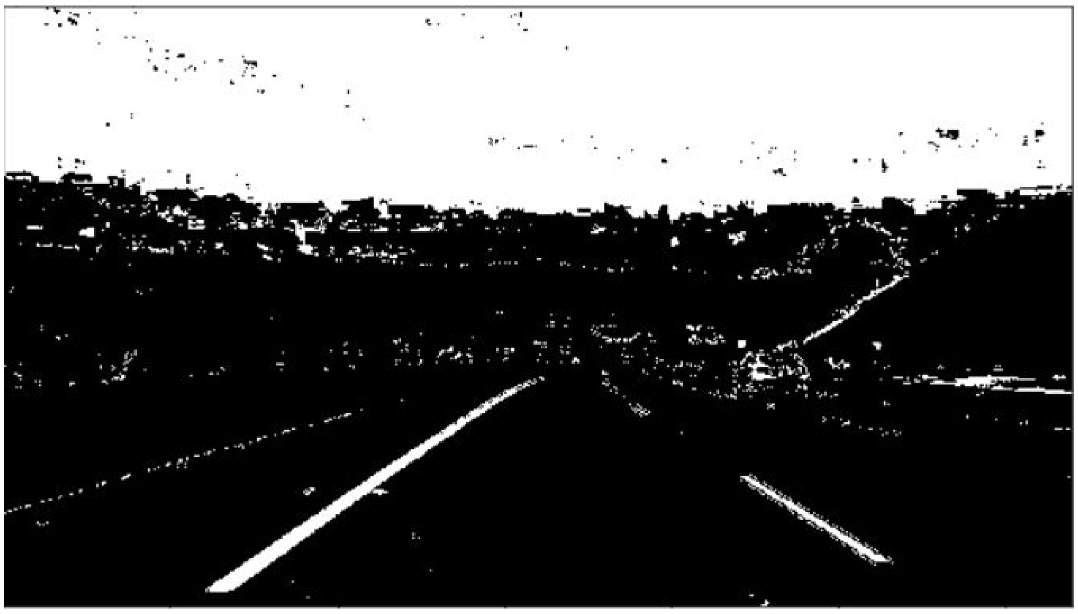
2. Bird eye view

Than i need to bring my image to the “Bird eye view” in order to be able to see the lines parallel. First i defined ROI points and a list of points, where the ROI points supposed to be on the “Bird eye view” image. With the funktion “cv2.getPerspectiveTransform” i got a necessary view.



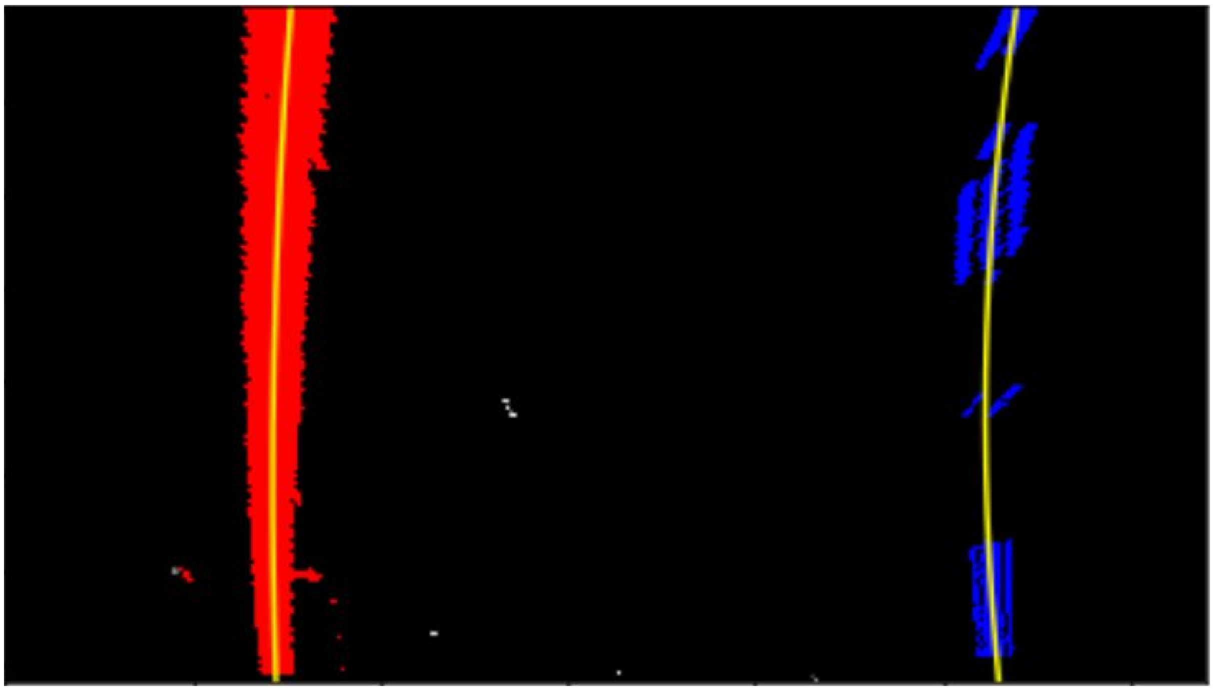
3. Binary image

Applying several filters like sobel filter and filter, which use S part of HLS and V of HSV stream with a certain thresholds and then combining them i could get a binary image.

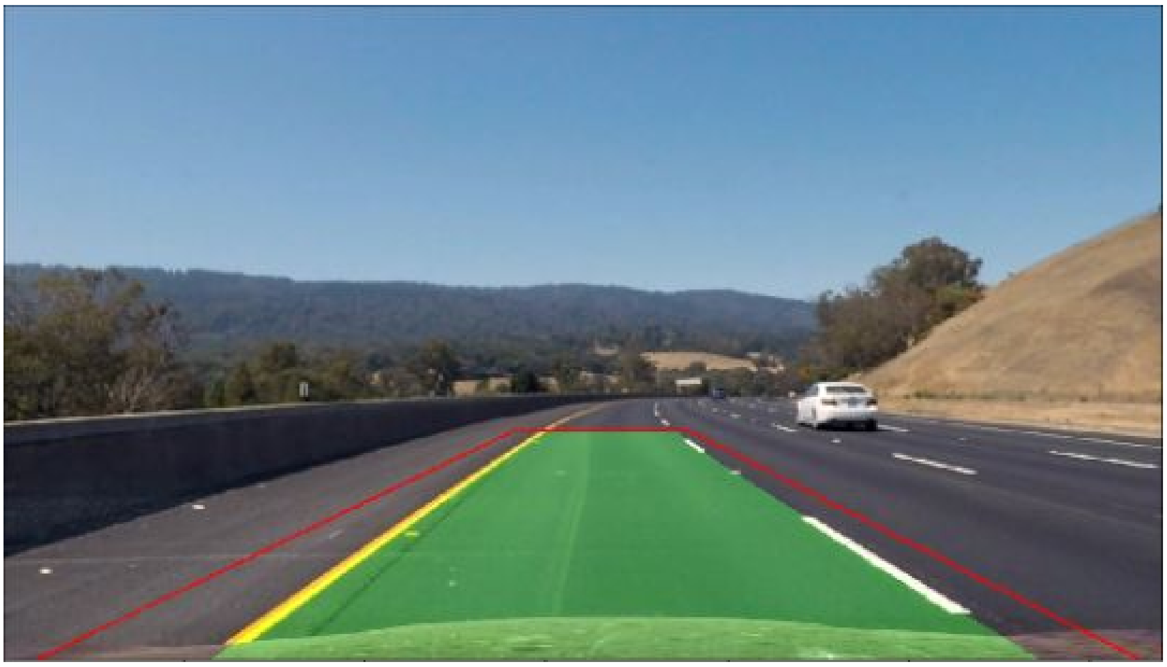


4. Finding lane lines

In order to be able to find the lane lines pixels firstly i have to bring an image to a “Bird eye view” format, and then apply a sliding windows approach. That gives me coordinates of all pixels of lanes in a ROI. Then i fit a polynomial with a “np.polyfit” funktion.



After the first polynomials were found, i could look for a next one nearby the first one. With a funktion “Search around the poly” i could successfully do it, if no pixels were found, i used again a “sliding windows approach”. Then using a “cv2.fillPoly” i could colored the space between the two founded lanes with a certain color. Using a invertional transform matrix i could bring that image from a “Bird eye view” format back to the normal one. Using the function “cv2.addWeighted” i could overlap the image with founded lines with a original one.



5. Finding value of the curvature

Using this Formel:

$$R_{curve} = \frac{(1+(2Ay+B)^2)^{3/2}}{|2A|}$$

i could get an value of a curvature. Defining the bottom coordinates of a polynomials, finding an average of them, subtracting the value of the middle of an image i could get a position of a vehicle with respect to the middle of the road.

6. The video, where the entire approach was applied, you can find on the YouTube: https://www.youtube.com/watch?v=0jroYybB21Y