Advanced Lane Detection

*Udacity Term - 1*

*Period October to January*

## System & Software Specification

OS - Windows 7

Hardware: Intel  i7 core CPU

Programming language: Python 3.x

 Python Libraries used:

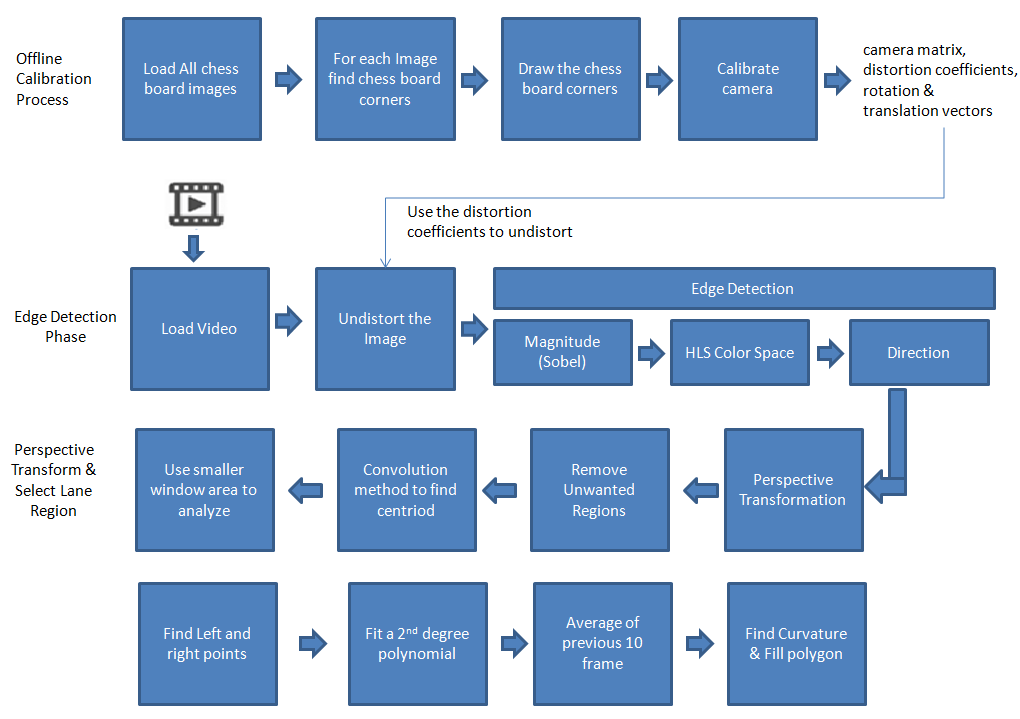
* OpenCV:  library name "cv2" . Used for image processing
* Numpy: Array related functionality
* Matplotlib: used for plotting images
* Math: Used for finding square roots
* Glob: To load images from folder
* moviepy.editor: For loading and creating a video

## Description

Objective: Goal of the project is to calibrate the camera with the images proved by Udacity. Undistort the frames in the video. Apply HLS color conversion and find the edge gradient and magnitude and direction. Apply Perspective transform and find the lane path. Find the lane curvature and print the same on the video.

## Approach

The overall flow of the program is as shown below



### Calibration

The Image provided by udacity is used for calibration of the camera. For each image the chess board corners are identified and calibrate the camera.

We get the camera intrinsic parameters, the distortion coefficient, the rotation and translation matrix.

The **camera matrix** is

[[ 1.15694047e+03 0.00000000e+00 6.65948820e+02]

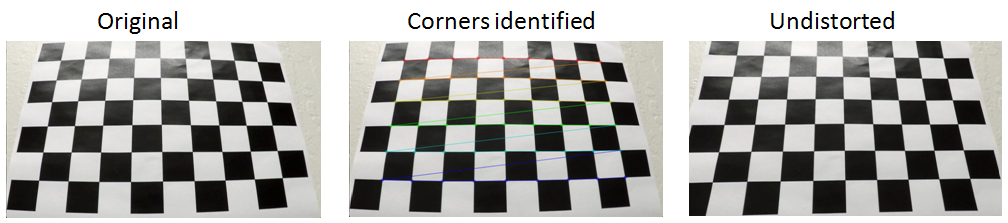
[ 0.00000000e+00 1.15213880e+03 3.88784788e+02]

[ 0.00000000e+00 0.00000000e+00 1.00000000e+00]]

The **Distortion coefficients** are

[[ -2.37638054e-01 -8.54042100e-02 -7.90999575e-04 -1.15882283e-04 1.05726077e-01]]

Sample of the calibration is as shown below.



### Loading Phase

The Video files are loaded using the opencv python library module “moviepy” . All the frames are passed to a function “processFrame” for processing.

### Edge Detection

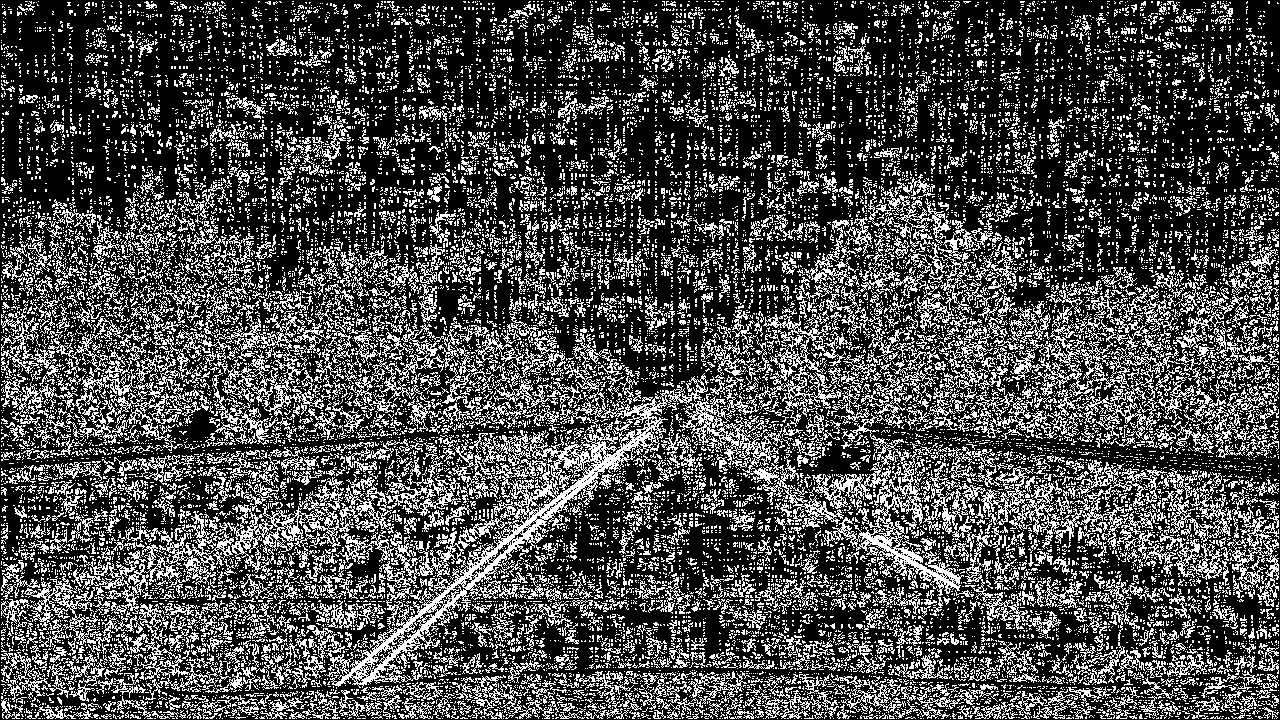
The frames loaded from the video are first undistorted using the camera matrix. The undistorted image is used for edge detection.

We fine the magnitude of the edge gradient using soblel operator.

The output of the Magnitude is shown as below



We find the direction of the gradient and a sample of the output is as shown below



We also convert the frame to HLS space and binaries the same using suitable threshold.



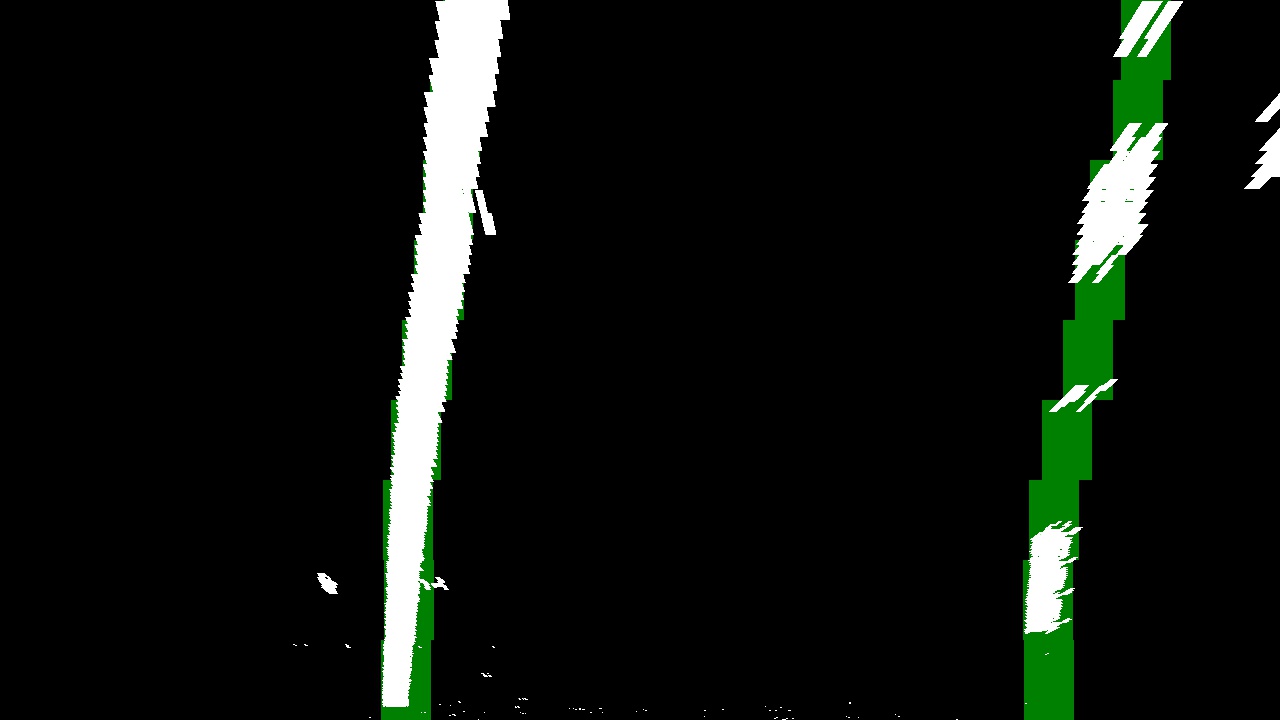
We combine all the above information to extract the edges as much as possible



Once the edges are detected the perspective transformation is done.

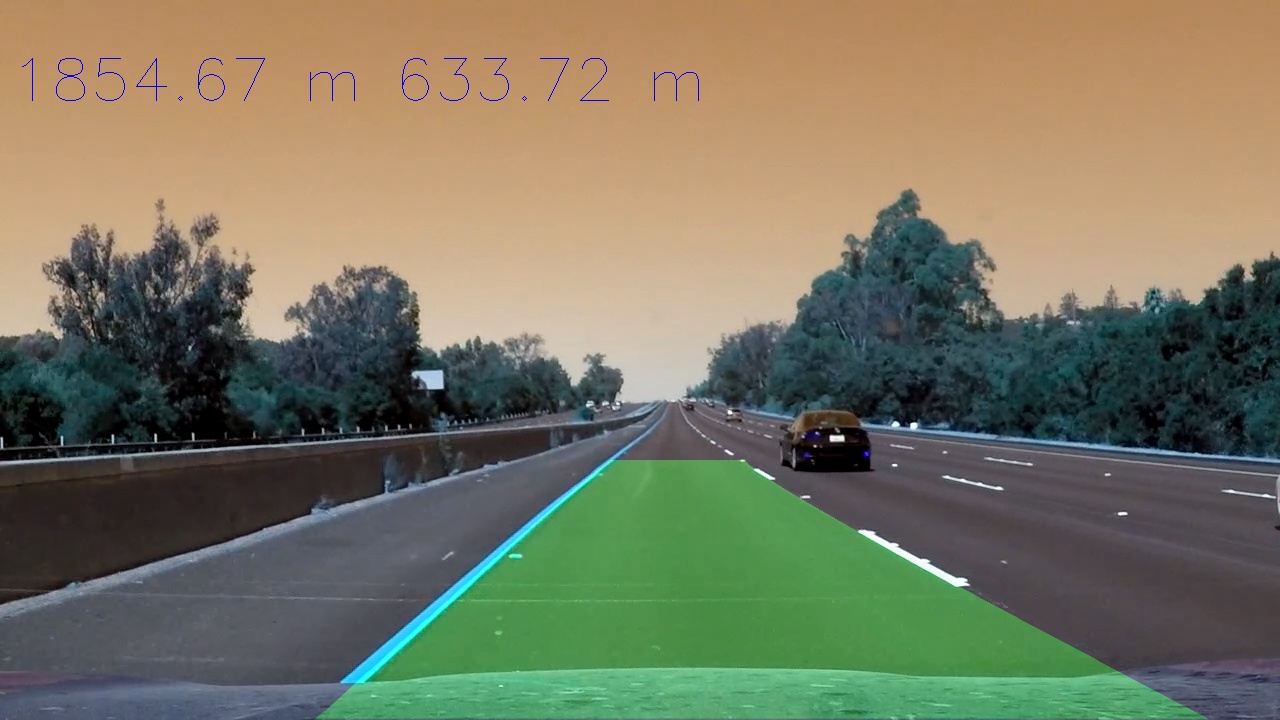


Using the warped image, we use the convolution method to identify the histograms of smaller windows in the frame and segregate the left and right lane



Using the above, the left and right points are identified and a 2nd order polynomial curve is fitted. And the curvature is estimated.

We create blank image and draw the lines on to it. And fill the polygon. We unwarp the image and draw it back to the original image.



## Improvements

The lane is not very smoothly detected. Some more smoothening has to be done.