Lane Departure Warning System triggers warning for prevention when a vehicle is going out of the lane. The goal of this project is to detect lane lines from single images using Canny detector-Hough transform and written in Python and OpenCV library. The system can be further modified to detect lanes from a video stream. The edge detection and straight line detection is done by

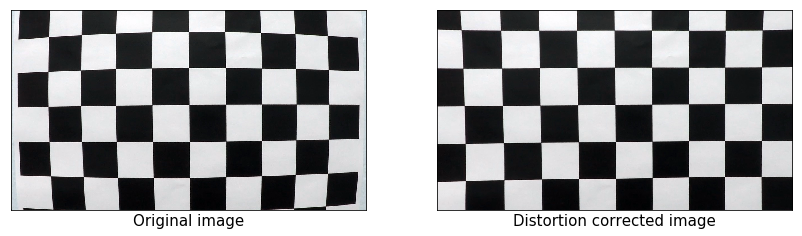
Initial Image

Step 1: Compute Camera Calibration

The camera calibration process is being used to determine a relationship between 3D in reality and projection on camera as 2D (pixels)

A chessboard image from different angles has been used to determine the input image position against the chessboard and to locate the internal corners of the board using the “cv2.findchessboardcorners” function. The function determines what image array corresponds to the internal corner location of the chessboard.

Additionally, “cv2.calibrateCamera()” function is used to look out for the camera intrinsic and extrinsic parameters for different views of calibration pattern. These parameters are used in cv2.undistort function to rectify the image produced on the camera. Undistort function transforms the input image with parameters which are tangential and radial in view.



Output : 

Step 2: Region of Interest

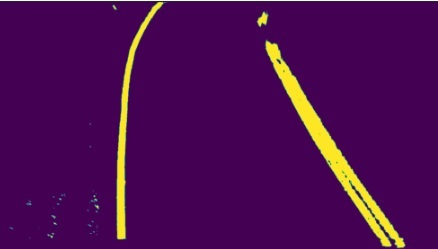
In order to make this perspective warp image,"cv2.getPerspectiveTransform" and cv2.warpPerspective functions are used. Four points are placed on the image so that the points only cover the areas in which our system is interested. Using this method, It will allow us to work in a region where the road lanes are defined. Each camera will have a different focal length and frame so for each camera we might have different coordinates which will be used to define the region of interest.



Original Image Region of Interest

Step 3 : HSL Color Space

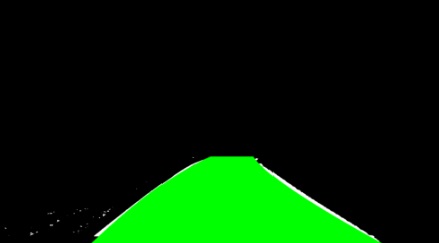
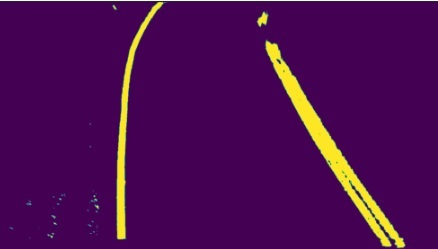
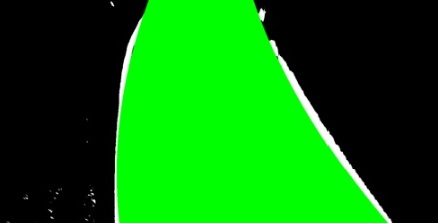
The images in which shadows occur are hard to detect the lanes so on the basis of human vision perception. The image is converted into HSL (Hue, saturation, Lightness) color space. We can accomplish this by using the function “cv2.cvtColor” which makes the image to be easily distinguishable by separating the yellow and white color from the region of interest.



Region of Interest HSL Color Space

Step 4 : Detection and Imposing of lane lines

We are now having a binary image which contains the pixels in color which are in the lane boundaries. We have implied a sliding window search technique which scans the image initially from to top with the pixels present in the image. If sufficient pixels are present in the image, the next image will be centered in their mean position. In this way we are following the path of the pixel. Once the lane is detected then it will use a polynomial function through the points which allows us to produce a smooth line which is to be considered as the best approximation. Now the system will superimpose the pixels identified to the original image structure.



HSL Color Space Lane Detection Append

Step 5 : Final Output

Now the system will read the coordinates and will calculate the lane direction through the coordinates which are received from step4 and will superimpose the information that is calculated in the original image.