International Research Journal of Education and Technology
Peer Reviewed Journal
ISSN 2581-7795

# REAL TIME ROAD LANE DETECTION FOR AUTONOMOUS DRIVING USING COMPUTER VISION

Mrs.M.Kalaiselvi.ME., R.Kayalvizhi, N.Devadharshini, D.Krishna
Assistant Professor, Department of Computer Science and Engineering
Vivekanandha College of Technology for Women, Tiruchengode.
kalaiselvi.mayilsamy@gmail.com, rkayalvizhi17cser@gmail.com,
ddnateshddnatesh@gmail.com,prikrishna1302@gmail.com

***Abstract*—Road Lane detection plays a vital role in the Advanced Driver assistive systems and it improves the vehicle's safe driving. However, Road lane detection is a complex problem because of the varying road conditions that one can encounter while driving. In this project, a vision based lane detection approach capable of reaching realtime operation with robustness to lighting change and shadows is presented. The lane boundaries, lane direction and its radius of curvatures were detected from a stream of videos. The video is recorded from a camera mounted on the top of a vehicle. We have corrected the camera distortion in the input frame. HLS thresholding and Canny edge thresholding techniques are carried out to the undistorted image for getting focus on the lane lines in the binary image. Then the resulted frame is warped into the bird's eye by applying the perspective transformation technique. The respective lane line pixels are identified using sliding window approach and then left and right lane lines are identified by fitting second-degree polynomials. The lane curvature and deviation from the lane centre are also computed after the identification of the lane. The identified lane boundaries are warped back onto the input image and the radius of lane curvature and vehicle position is calculated and displayed with appropriate comments. Hence this technique is enforced using python programming language and for processing the images Open CV is used.**

***Keywords*—**Advanced Driver assistive systems, HLS thresholding, Canny edge thresholding, binary image, perspective transformation, sliding window, second-degree polynomial, python programming, Open CV.

## I. INTRODUCTION

With the rapid development of society, automobiles have become one of the transportation tools for people to travel. As more and more vehicles are driving on the road, the number of car accidents is increasing every year. Advanced driver assistance systems which include lane departure warning (LDW), Lane Keeping Assist, and Adaptive Cruise Control (ACC) can help people analyse the current driving environment and provide appropriate feedback for safe driving or alert the driver in dangerous circumstances.

Lane detection has been applied in an intelligent vehicle system to reduce the chances of road accidents. Identifying lanes on the road is a common task performed by all human drivers to ensure their vehicles are within lane lines when driving, so as to make sure traffic is smooth and minimise chances of collisions with other cars in nearby lanes. Similarly, it is a critical task for an autonomous vehicle to perform. It turns out that recognizing lane lines on roads is possible using well known computer vision techniques.

## II. LITERATUREREVIEW

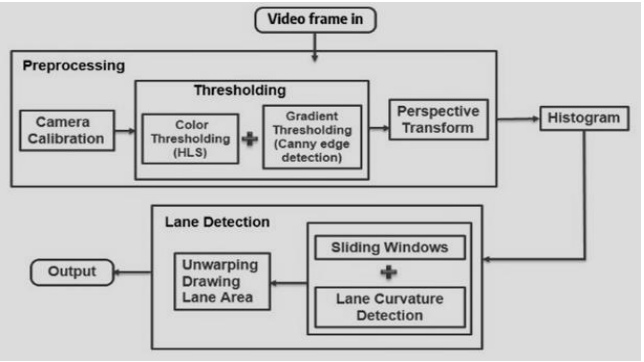
*A. HuiKong,Jean-YvesAudibert,JeanPonce,2009 [1]*

This paper addresses decomposing the road detection process into two steps: the estimation of the vanishing point associated with the main (straight) part of the road, followed by the segmentation of the corresponding road area based on the detected vanishing point.

*B. S.SaiTej,M.Sravani,Ch.AjaySumanth,M.RamNitin2009[2]*

This model is based on image processing and road detection in self-driving vehicles. In this process of finding the road in the image captured by the vehicle, we can use some algorithms for vanishing point detection using Hough transform space, finding the region of interest, edge detection using canny edge detection algorithm and then road detection.

## III. SYSTEMARCHITECTURE




## IV. MODULES

Working of computer vision algorithm can be architecture as a single model or pipeline of models. Since this process takes a series of related steps, pipeline of models can be considered by passing specified input each model as corresponding previous model output.

Pipeline of Models: The road detection pipeline follows these models:

i) Computationofcameracalibrationmatrixanddistortioncoeffici entsfromaset of chessboard images.

ii) Applying distortion correction on raw images.

International Research Journal of Education and Technology
Peer Reviewed Journal
ISSN 2581-7795

iii) Production of a bird's eye view image via perspective transform.

iv) Histogram of bird's eye view image.

v) Using sliding windows to find lane line pixels.

vi) Fitting of second degree polynomials to identify left and right lines composing the lane.

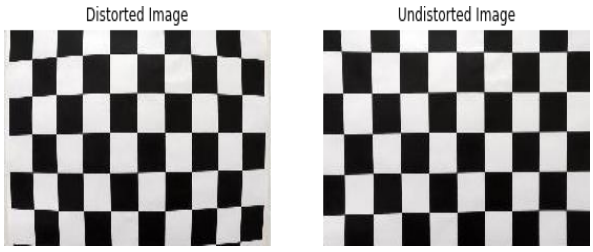
vii) Computation of lane curvature and deviation from lane centre.

viii) Warping and drawing of lane boundaries on image as well as lane curvature information.

Model1: Computation of camera calibration matrix and distortion coefficients from asset of chessboard images

To compute the camera calibration matrix and distortion coefficients, we use multiple pictures of a chessboard on a flats face taken by the same camera. The distortion matrix was used to un-distort a calibration image and provides a demonstration that the calibration is correct. Open CV has a convenient method called find Chessboard Corners that will identify the points where black and white squares intersect and reverse engineer the distortion matrix this way.

The cv2 find Chessboard Corners function to store the object points (3D points in real world space) and image points (2D points in image plane) of the grid corners. These object points and image points are used in cv2. Calibrate Camera () to return the calibration matrix, distortion coefficients, rotation and translation vectors. Next we run our chessboard finding algorithm over multiple chessboard images taken from different angles to identify image and object points to calibrate the camera.




Model2: Applying distortion correction on raw images

The calibration data for the camera that was collected in model1 can be applied for raw images to apply distortion correction. It may be harder to see the effects of applying distortion correction on raw images compared to a chessboard image.

Before Un-distorting raw image

After Distortion Correction

Model3: Application of color and gradient thresholds to focus on lane lines

We apply color and edge thresholding in this section to better detect the lines, and make it easier to find the polynomial that best describes our left and right lanes later.

A. Color Thresholding

There are actually many ways to achieve this result, but we choose to use HLS where S channel may provide with great results depending on the lighting situation.

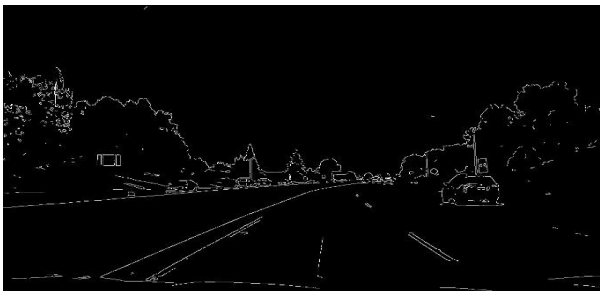
B. Edge thresholding

This section mainly performs the overall edge detection on the frame image, using the improved canny edge detection algorithm. The concrete steps of canny operator edge detection area follows:

i) First, we use a Gaussian filter to smooth the image (pre processed image). Gaussian smoothing is used to reduce the noise from image. We use this pre-processing step to remove many detected edges and only keep the most prominent edges from the image.

ii) Then we use the Sobel operator to identify gradients, that is change in color intensity in the image. Higher values would denote strong gradients, and therefore sharp changes in color.

International Research Journal of Education and Technology
Peer Reviewed Journal
ISSN 2581-7795



We naturally combine both color and Sobel thresholded binary images.

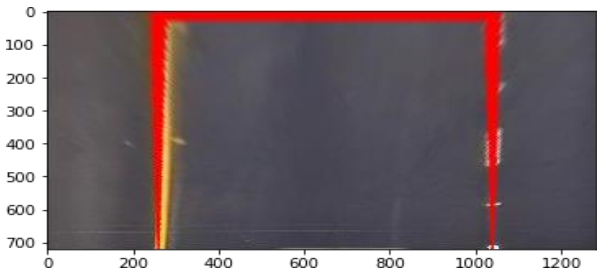
Model4: Production of a bird's eye view image via perspective transform

Images have perspective which causes lanes lines in an image to appear like they are converging at a distance even though they are parallel to each other. To make them to view as a parallel line by transforming the image to a 2D Bird's eye view where the lane lines a realways parallel to each other.

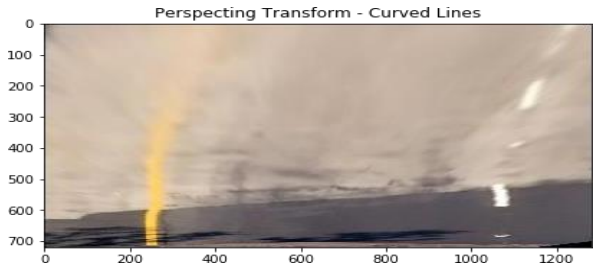
To perform the perspective transform, we identified 4 source points that form a trapezoid on the image and 4 destination points such that lane lines are parallel to each other after the transformation. The destination points were chosen by trial and error but once chosen works well for all images and the video since the camera is mounted in a fixed position.

The perspective transform, then application of color and gradient thresholding enable us to clearly identify the position of the lanes on the bird's eye view image.

The perspective transform produces the following type of images:



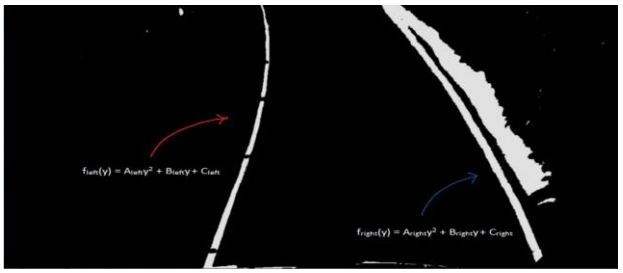

Bird's eye view on curved lanes

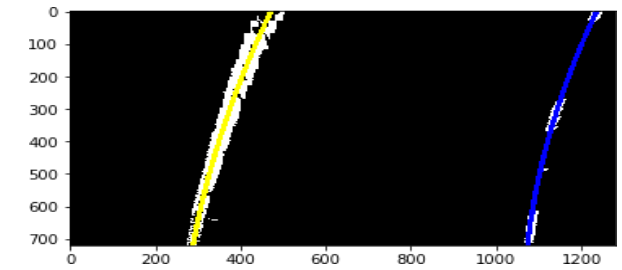


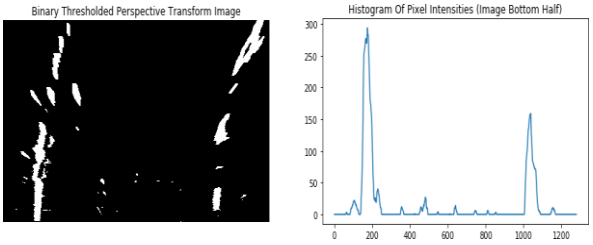

Model5:Histogram of bird's eye view image

The maximum probability region can be determined by observing the histogram of the bird's-eye view, which produces two distinct peaks, one for the left lane and the other for the right.

Once the input image is preprocessed, the next step is to locate and map the lane lanes in the image space. The approach would be to plot a histogram of pixels that are non-zero in the lower half of the binary image to observe the pattern. We then compute a histogram of our binary thresholded images in the y direction, on the bottom half of the image, to identify the x positions where the pixel intensities are highest.




Model6: Using sliding windows to find lane line pixels

As the pixel values are now binary, the peaks can represent where most of the non-zero pixels are located and thus area good indicator of the lane lines. Thus, the x-coordinates from the histogram serve as a starting point to search for the respective lanes. The concept of sliding windows approach will be applied here, which is essentially a window with a margin being placed around the line's centre.

Model7: Fitting of second degree polynomials to identify left and right lines composing the lane

From then, we simply compute a second degree polynomial, via num py's poly fit, to find the coefficients of the curves that best fit the left and right lane lines which mean the window template is slid across the image from left to right and any overlapping values are summed together, creating the convolved signal. The peak of the convolved signal is where the highest overlap of pixels are and it is the most likely

position for the lane marker. Methods have been used to identify lane line pixels in the rectified binary image. The left and right lines have been identified and fit with a curved polynomial function.

Model 8: Computation of lane curvature and deviation from lane centre

The next step is to compute the radius of curvature which can be calculated with a circle that closely fits nearby points on a local section of a curve. The radius of curvature of the curve at a particular point can be defined as the radius of the approximating circle.

International Research Journal of Education and Technology
Peer Reviewed Journal
ISSN 2581-7795

We took the measurements of where the lane lines are and estimated how much the road is curving, along with the vehicle position with respect to the centre of the lane. We assumed that the camera is mounted at the centre of the car.

We also compute the car's distance from the centre of the lane by offsetting the average of the starting (i.e.bottom) coordinates for the left and right lines of the lane, subtract the middle point as an offset and multiply by the lane's pixel to real world width ratio.

Model 9: Warping and drawing of lane boundaries on image as well as lane curvature information

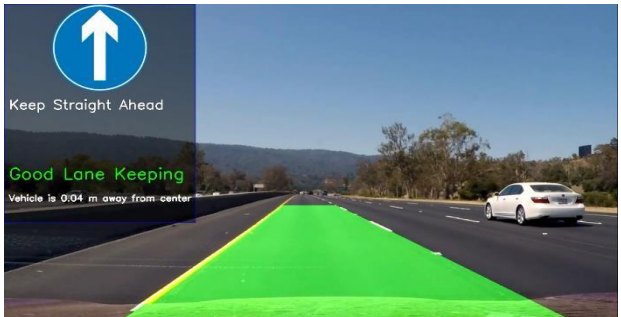
Finally, we draw the inside of the lane in green and unwarp the image, thus moving from bird's eye view to the original undistorted image. The fit from the rectified image has been warped back onto the original image and plotted to identify the lane boundaries.

## V. OUTPUTSCREENSHOTS

The output will be as follows if there is a left curve ahead and the lane area is colored with green. As the vehicle is within the lane line so the output screen is displayed with Good Lane Keeping. The vehicle is deviated 0.07m away from center and the curvature range is 3948m.


Theoutputwillbeasfollowsifthereisa Rightcurveahead.Thevehicle is deviated 0.15m away from center and the curvaturerangeis2425m.


Theoutputwillbeasfollowsifthereexistsnocurves.Thevehicleis deviated 0.07m away from center. As the road is straight, nocurvaturerangeisdisplayedintheoutputscreen.




## VI. CONCLUSION

When we drive, we use our vision to decide where to go. The lines on the road detected by the model that show us where the lanes are act as our constant reference for where to steer the vehicle. This steering is also done automatically. Naturally, one of the first things we would like to do in developing a self-driving vehicle is to automatically detect lane lines using analgorithm.Thusinourprojectweproposedaroadlanedetectionby performingcameracalibration,colorandgradientthresholding,perspectivetransformationandslidingwindowstodetectlanelines.Weusedrealtimevideosasaninputinsteadofimages also we used second degree polynomial and histogram based approach to detect curved roads. And it works well in all lighting conditions.

## REFERENCES

[1] ArchitRastogi,ComputerVision:LaneFindingthroughImage Processing, 6 Sep 2020,https://medium.com/swlh/computer-vision-lane-finding-through-image-processing-516797e59714.

[2] Broggi A, Caraffi C, Fedriga R.I, and Grisleri P, "Obstacledetection with stereo vision for off-road vehicle navigation,"IEEE International Workshop on Machine Vision for IntelligentVehicles,2005.

[3] Chiu K.Y and Lin S.F, "Lane detection using color-basedsegmentation,"IEEE IntelligentVehicles Symposium,2005.

[4] General Road Detection From A Single Image, TIP-05166-2009,ACCEPTED,HuiKong,Member,IEEE,Jean-YvesAudibert, and Jean Ponce, Fellow, IEEE Willow Team, EcoleNormaleSuperieure / INRIA / CNRS, Paris, France Imagineteam,EcoledesPontsParisTech,Paris,France.

[5] Gowri Pushpa G, Sai Tej S, Sravani M, Ajay Sumanth C.H,Ram Nitin M, "Road Detection from a picture using ComputerVision,"IEEEProceedingsinIntelligenttransportationSystems,pp.456–464,2001.

[6] Kaske A, Husson R, and Wolf D, "Chi-square fitting ofdeformabletemplatesforlaneboundarydetection,"IARAnnual Meeting,1995.

[7] KongH,AudibertJY,andPonceJ,"Vanishingpointdetection for roaddetection,"CVPR, 2009.

[8] McCall J.J and Trivedi M.M, "Video based lane estimationand tracking for driver assistance: Survey,system,andevaluation,"IEEETrans.onIntelligentTransportationSystems, pp. 20–37, 2006.

[9] SeyedMostafaLatifi,SamaneSharifiMonfared,BilalSedefandLavdieRada,"Roadlanedetectionthroughimageandvideoprocessingusingedgedetectionand Houghtransformforautonomousdrivingpurposes".

[10] Sparbert J, Dietmayer K, and Streller D, "Lane detectionand street type classification using laser rangeimages,"IEEEProceedingsinIntelligenttransportationSystems,pp. 456–464, 2001.

[11] Wang Y, Teoh E.K, and Shen D, "Lane detection andtrackingusingb-snake,"ImageandVisionComputing,pp.269–

International Research Journal of Education and Technology
Peer Reviewed Journal
ISSN 2581-7795