“VISION”

A Project Work

Submitted in the partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING
IN
CSE (AIML)

Submitted by:
JAHANVI CHAWLA
UID – 18BCS6108

SEJAL GULATI
UID – 18BCS6116

SHIVAM
UID – 18BCS6138

SACHIN KAUSHIK
UID – 18BCS6119

YUVRAJ KOUNDAL
UID-18BCS6114

Under the Supervision of:

MONIKA SINGH

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
APEX INSTITUE OF TECHNOLOGY
CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413, PUNJAB

SEPTEMBER - 2021

DECLARATION

I, **‘Shivam’**, student of **‘Bachelor of Engineering in CSE- AI & ML’**, **session: 2018-22**, Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Punjab, hereby declare that the work presented in this Project Work entitled ‘**Vision’** is the outcome of our own bona fide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Shivam

Candidate UID: 18BCS6138

Date: 21st November, 2021

Place: Chandigarh University

.

ABSTRACT

Lane detection is a difficult problem to solve. For decades, it has piqued the interest of the computer vision community. Lane detection is essentially a multi-feature detection problem that has proven to be difficult for computer vision and machine learning techniques to solve. Despite the fact that various machine learning algorithms are employed for lane detection, they are mostly used for classification rather than feature construction. Modern machine learning algorithms, on the other hand, can be used to find features with high recognition value and have shown to be successful in feature detection tests. However, these strategies have not been fully implemented in terms of lane detection efficiency and accuracy.

We present a new preprocessing and ROI selection method. The main purpose is to extract white features using the HSV color transformation, add preliminary edge feature detection in the preprocessing step, and then pick ROI based on the proposed preprocessing. The lane is detected using this new preprocessing method. Along with us, we used another approach in which we use Sliding Window algorithm, warped perspective transforms, and polynomial fits to detect lane lines.

ACKNOWLEDGEMENT

It was a great pleasure to undertake this project. This project is made under the shadow of Ms. Monika Singh. It is a precise project regarding the Detecting Lane while driving a car. I would like to express my special gratitude to our mentor who gave this golden opportunity to work on this wonderful project on a topic which has a great social relevance and a brilliant future scope- which also helped me in doing a lot of Research and I came to know about many new and emerging technologies. I am really thankful to them.

Secondly I would also like to thank my team mates for their dedicated efforts and for working with me on this educational project. This project was made for educational purpose as well as for personal research work which was carried individually at personal level and in team as well

Table of Contents

1 INTRODUCTION

1.1 Problem Definition

Lane detection is a prominent topic in machine learning and computer vision, and it's been used in smart vehicle systems. The lane detection system gathers data from lane markers in a complex environment and uses it to accurately predict the vehicle's position and trajectory in relation to the lane. Automobiles have been one of the primary modes of mobility for people as society has progressed. There are an increasing number of automobiles of various types on the small route. Every year, as more automobiles enter the road, the number of people killed or injured in car accidents rises. The question of how to drive safely when there are a lot of cars on the road and the roads are narrow has become a hot topic. Lane departure warning (LDW), Lane Keeping Assist (LKA), and Adaptive Cruise Control (ACC) are examples of advanced driver assistance systems that can help people analyze their current driving environment and provide appropriate feedback for safe driving or notify the driver in dangerous situations. This type of auxiliary driving system is predicted to improve with time.

1.2 Problem Overview

The difficulty in predicting the road traffic environment, however, is a bottleneck in the development of this system. According to the findings, the likelihood of accidents is significantly higher than typical in the complex traffic environment when vehicles are many and speeds are excessive. The key perceptual signals of human driving in such a complex traffic condition are road color extraction and texture recognition, as well as road boundary and lane marking. Lane detection, on the other hand, is critical in the lane departure warning system. The task of lane detection is split into two parts: edge detection and line detection. In lane detection, line detection is just as crucial as edge detection. When it comes to line detection, we normally use one of two ways: feather-based methods or model-based methods. In this paper, we suggest a lane recognition method that is suitable for a wide range of complicated traffic conditions, particularly where road speeds are excessive. We first preprocessed each frame image before selecting the processed images' region of interest (ROI). Finally, for the ROI area, we only needed an edge detection vehicle and line detection.

1.3 Software Specification

Anaconda(Jupyter notebook), Visual studio
Languages used-
Python 3.0, HTML, CSS
Libraries used-
Matplotlib, Numpy, Pandas, CV2, math, pickle

2 LITERATURE REVIEW

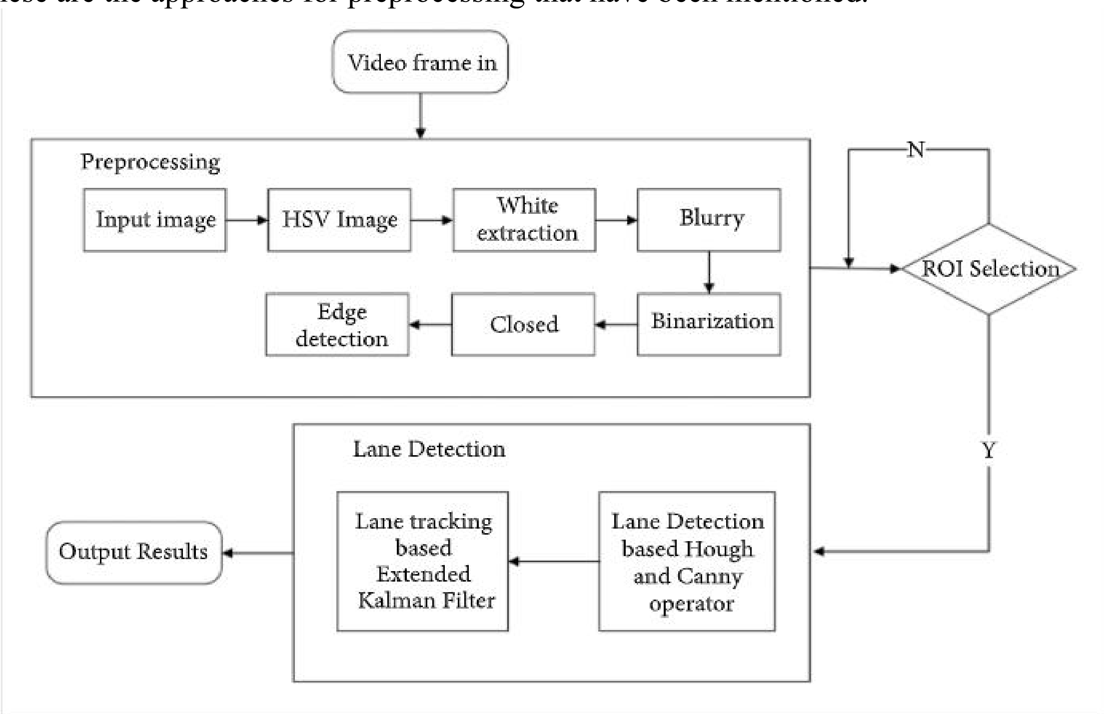
We provided a new preprocessing method and ROI selection mechanism in this approach. First, we transformed the RGB color model to the HSV color space model and retrieved white features from the HSV model during the preprocessing stage. Simultaneously, preliminary edge feature detection is added to the preprocessing step, and the ROI area is chosen from the area below the picture based on the proposed preprocessing. Existing preprocessing methods simply conduct operations like greying, blurring, X-gradient, Y-gradient, global gradient, thresh, and morphological closure when compared to existing approaches. Furthermore, there are numerous methods for determining the ROI area. Some of them select the ROI region based on the lane's edge feature, while others select the ROI area based on the lane's color feature. These existing technologies do not give reliable and timely lane information, making lane identification more difficult. Experiments in this research show that the suggested strategy outperforms the existing preprocessing and ROI selection methods in lane detection.

Improve the efficiency and accuracy of real-time lane detection using enhanced lane detection technology. Typically, the lane detection module is split into two parts:

- picture preprocessing
- line lane detection model creation and matching

Figure 1 depicts the overall diagram of our proposed system, with the key contributions of this research being the lane detecting blocks. Reading the frames in the video stream is the first step. The image preparation module is entered in the second stage.

What sets us apart from the competition is that during the preprocessing step, we not only process the image but also extract color and edge features. After extracting the color information of the image, we need to smooth the image with a Gaussian filter to limit the influence of noise in the motion and tracking process.
After that, binary threshold processing and morphological closure are used to create the image. These are the approaches for preprocessing that have been mentioned.



The adaptive region of interest (ROI) in the preprocessed image is then selected. Lane detection is the final phase.
The Canny operator is used to detect the edge of the lane line, followed by the Hough transform to detect the line lane. Finally, to detect and track lane lines in real time.

Also, in the second approach, we will make use of Sliding Window algorithm.

1. The series of steps to accomplish the goal for this project are as follows:-

2. Compute the camera calibration matrix and distortion coefficients.

3. Apply a distortion correction to raw images.

4. Create a threshold binary image.

5. Warp the detected lane boundaries back onto the original image and display numerical estimation of lane curvature and vehicle position. And then in turn detect the lane.

3 PROBLEM FORMULATION

The already existing problem in automated vehicles is perceiving lane such that they never tend to cross over to the opposite side or to turn to an illegal lane. Hence, this very project helps the automated i.e. the self-driving vehicles to keep track of the lane and keep a easy flow of the traffic. Since, it could not be possible with the help of tools such as Machine Learning and Computer Vision, an extensive use of such technology had been used in it to bring out the maximum accuracy and hence, the safety and reliability of our project.

The task that we wish to perform is that of real-time lane detection in a video. There are multiple ways we can perform lane detection. We can use the learning-based approaches, such as training a deep learning model on an annotated video dataset.

The approach we will be using will include dividing the work into following objectives:

1. Preprocessing (Colour selection, region of interest selection, grayscaling, gausian smoothing) , C***anny edge detection and Hough transform line detection***.

The goal is to piece together a pipeline to detect the line segments in the image. Then get an average and draw them onto the image for display. After that we get to the video streaming. The future scope of its potential is that it would be used in almost every self-driving vehicle such that it increases its accuracy and use domain over the time.

2. The second approach uses advanced techniques that builds on the earlier one by using thresholds for different color spaces and gradients, **S*liding window technique and curve fitting,*** warped perspective transforms, and polynomial fits to detect lane lines.

4 RESEARCH OBJECTIVES

The proposed research is aimed to carry out work leading to the development of an approach for Detecting the lane while driving. The proposed aim will be achieved by dividing the work into following objectives:

1.Preprocessing (Colour selection, region of interest selection, grayscaling, gausian smoothing), C***anny edge detection and Hough transform line detection***.

The goal is to piece together a pipeline to detect the line segments in the image. Then get an average and draw them onto the image for display. After that we get to the video streaming. The future scope of its potential is that it would be used in almost every self-driving vehicle such that it increases its accuracy and use domain over the time.

2.The second approach uses advanced techniques that builds on the earlier one by using thresholds for different color spaces and gradients, S***liding window technique***, warped perspective transforms, and polynomial fits to detect lane lines.

5 METHODOLOGY

5.1 Preprocessing

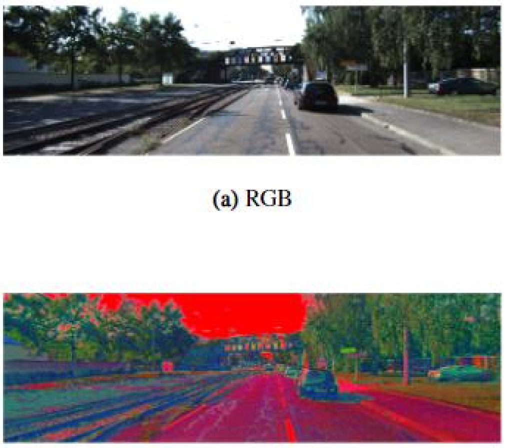
Preprocessing is a key aspect of image processing as well as lane detection. Preprocessing can help minimize the algorithm's complexity, minimizing the time it takes to run the programme later. The video input is a RGB-based color image sequence obtained from the camera. Many studies use various picture preprocessing approaches to increase the accuracy of lane detection.
A typical picture preparation approach is to smooth and filter graphics. The basic goal of filtering is to reduce image noise and improve the image's effect. For 2D images, low-pass or high-pass filtering can be utilized; low-pass filtering (LPF) is useful for denoising, and image blurring and high-pass filtering (HPF) are used to determine image boundaries. An average, median, or Gaussian filter could be employed to execute the smoothing procedure.

5.2 Color Transform

Color model transform is an important aspect of machine vision, and it's also crucial for lane detection in this paper. The actual road traffic environment, as well as the intensity of light, all cause noise that obstructs color detection. We can't tell the difference between white lines, yellow lines, and vehicles in the backdrop. The RGB color space used in the video stream is particularly sensitive to light intensity, therefore processing light at different periods has an undesirable effect. The RGB sequence frames in the video stream are color-converted to HSV color space images in this article. HSV (hue, saturation, and value) is an abbreviation for hue, saturation, and value. When compared to other colours, the values of white and yellow colours in the V-component are particularly brilliant and easy to extract, providing a good foundation for the following colour extraction.

Figure 5

Two images of different colour spaces. (a) RGB and (b) HSV colour transform.



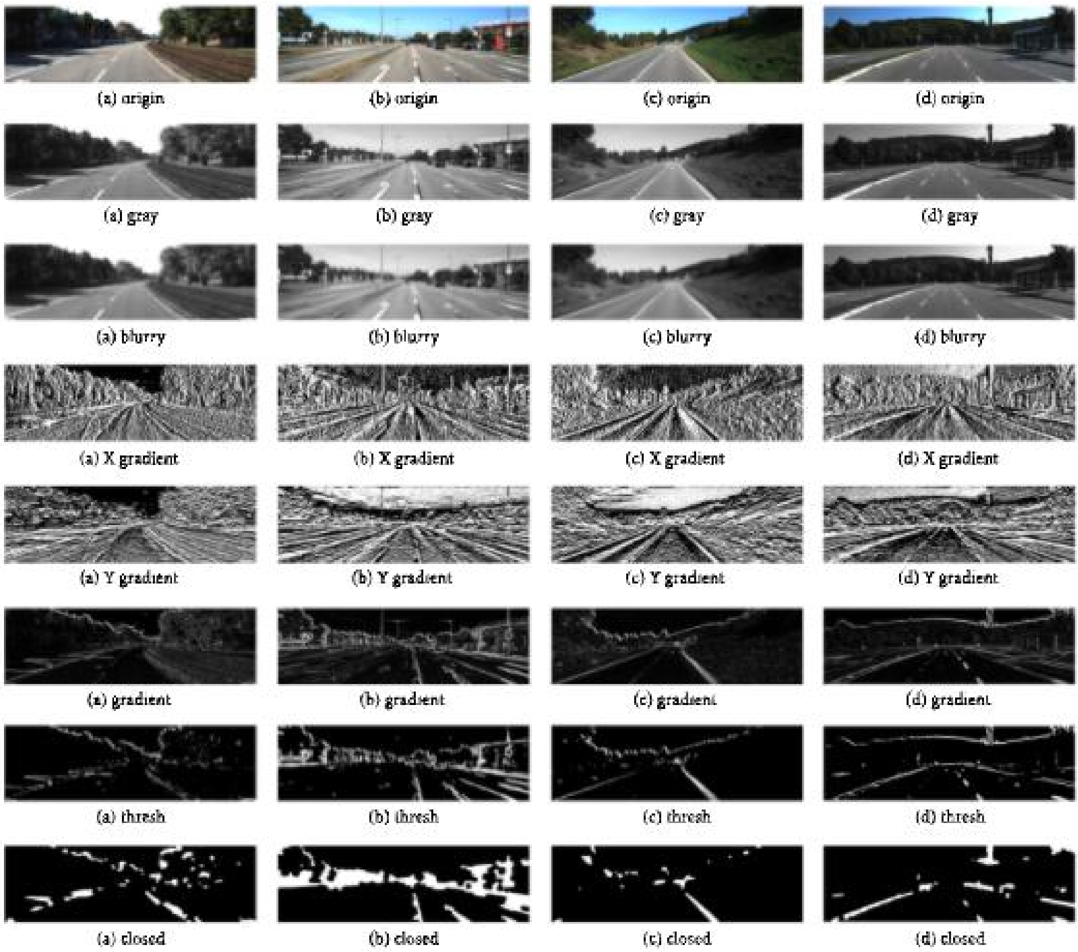
(a) RGB

(b) HSV

5.3 Basic Preprocessing

The video will be preprocessed for a huge number of frames. Individual photos are gray-scaled, blurred, and have their X-gradients, Y-gradients, global gradients, thresh of frame, and morphological closure determined. During the preprocessing phase, an adaptive threshold is established to account for various illumination situations. After that, we delete the spots in the binary conversion image and do the morphological closing procedure. Basic preprocessed frames aren't likely to be very effective in removing noise. Although preliminary lane information can be collected following the morphological closure, the results show that there is still a significant amount of noise.

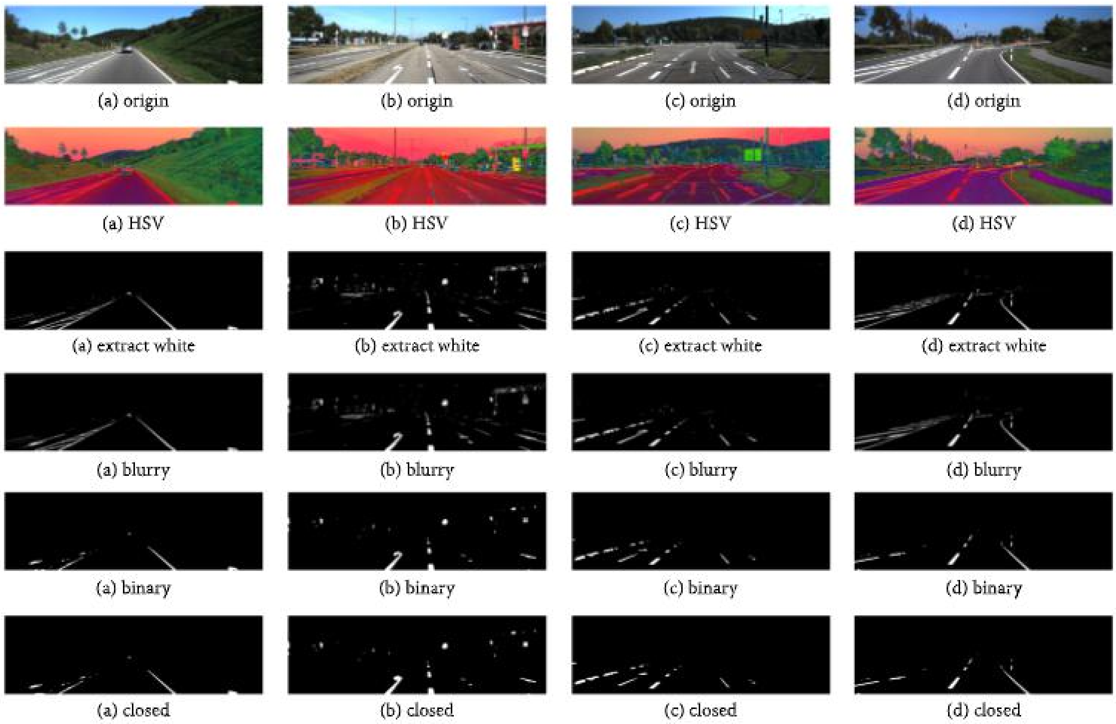
Basic preprocessing of sample frames (a), (b), (c), and (d).



5.4 Color Extraction

We add a feature extraction module to the preprocessing stage to improve the accuracy of lane detection. The goal of feature extraction is to maintain any lane-related features while removing nonlane-related ones. This work focuses on extending feature extraction to color. We add the white feature extraction after the image greying and color model conversion, and then do the traditional preprocessing operations one by one. The color extraction process presented in this paper is depicted in Figure below.

Adding white extraction in preprocessing of sample frames (a), (b), (c), and (d).



5.5 Edge Detection

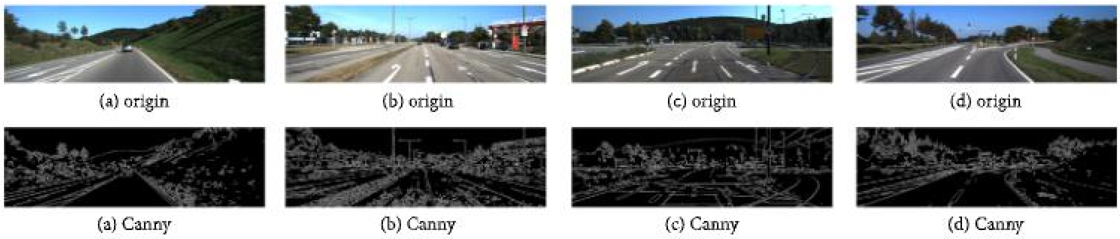
Edge detection two times in a row; the first time is to extract a wide range of edges throughout the full frame image. The edge detection is done again after the lane identification and ROI selection in the second. This detection enhances lane detecting accuracy even more. Using the updated Canny edge detection technique, this portion primarily performs overall edge detection on the frame image. The following are the concrete steps of Canny operator edge detection:

- We smooth the image (preprocessed image) with a Gaussian filter, then calculate the gradient magnitude and direction with the Sobel operator.

- The non-maximal value of the gradient amplitude must then be suppressed.

- Finally, we must detect and link edges using a double-threshold technique. Figure displays the image after Canny edge detection was used to extract it.

Canny edge detection of sample frames (a), (b), (c), and (d).

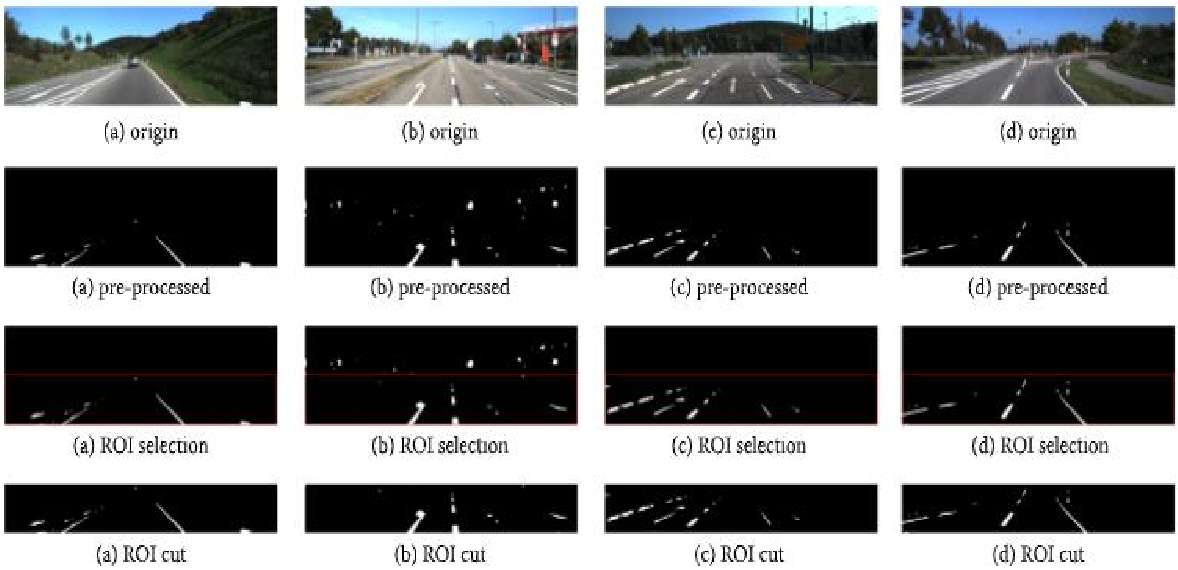


5.6 ROI Selection

We can observe that the obtained edge comprises not only the required lane line edges, but also other unnecessary lanes and the edges of the surrounding fences after edge detection by Canny edge detection. To get rid of these unnecessary edges, identify a polygon's visual area and only save the visible region's edge information. The camera is fixed in relation to the car, and the car's relative location to the lane is similarly fixed, so the lane is essentially maintained in a fixed area in the camera.

We can use an adaptive area of interest (ROI) on the image to reduce image redundancy and algorithm complexity. The input image is only set on the ROI area, and this strategy can improve the system's speed and accuracy. The standard KITTI road database is used in this paper. Each frame of the vehicle's running video is divided into two pieces, with one-half of the lower section of the image frame serving as the ROI area. The ROI selection of sample frames (a), (b), (c), and (d) treated by the suggested preprocessing is shown in Figure. After being processed by the proposed preprocessing method, the images of the four separate sample frames were able to substantially display the lane information, although the upper half of the image contains a lot of nonlane noise in addition to the lane information. As a result, the ROI area was chopped out of the lower half of the image (one-half).

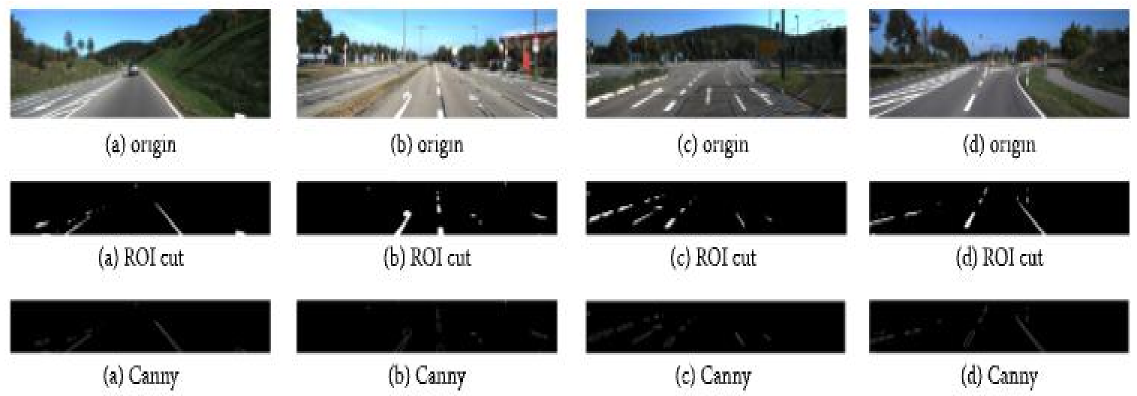
ROI selection of sample frames (a), (b), (c), and (d) based on the proposed preprocessing.



5.7 Feature Extraction

For lane detection, feature extraction is critical. Canny transform, Sobel transform, and Laplacian transform are some of the most frequent edge detection methods. We chose Canny transform because it is superior. Following the proposed ROI selection, we performed Canny edge detection, as seen in the figure.

Canny edge detection of sample frames (a), (b), (c), and (d) after the proposed preprocessing.



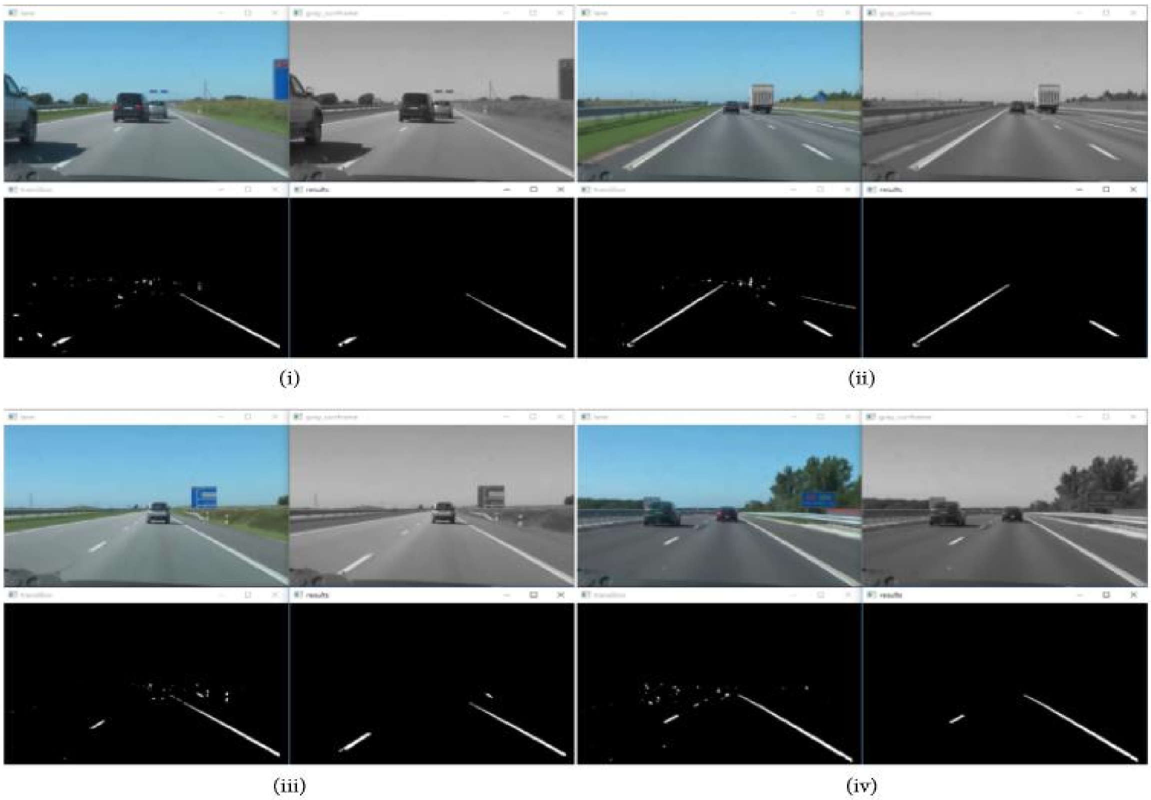
5.8 Lane Detection

Lane detection is divided into two types: lane edge detection and linear lane detection. This part provides the fundamental lane detection functions and conducts lane detection based on the proposed ROI selection and better preprocessing.

Lane detection methods include feature-based and model-based approaches. In this study, the method-based feature is utilised to detect the colour and edge features of lanes in order to increase lane detection accuracy and efficiency.

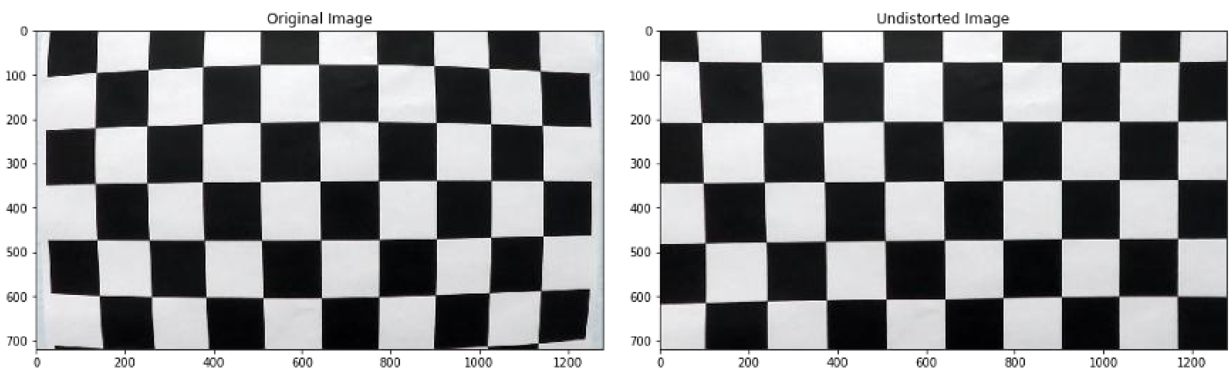
Straight lane detection can be accomplished in two ways. One option is to draw lane lines in the corresponding area of the original image using the Hough line detection function provided by the OpenCV package, which is widely used for image processing. Self-programming is the alternative option. The ROI area is traversed in the header file to perform line detection for a given range of angles.

Different moments (i), (ii), (iii), and (iv) in one video.



5.9 Compute the camera calibration matrix and distortion coefficients

All cameras use lenses and one of the problems with lenses is that they have some radial distortion. To remove this distortion, we used OpenCV functions on chessboard images to calculate the correct camera matrix and distortion coefficients. This can be achieved by finding the inside corners within an image and using that information to un-distort the image.
The distortion matrix was used to un-distort a calibration image and provides a demonstration that the calibration is correct. An example shown here in figure below, shows the before/after results after applying calibration to un-distort the chessboard image.

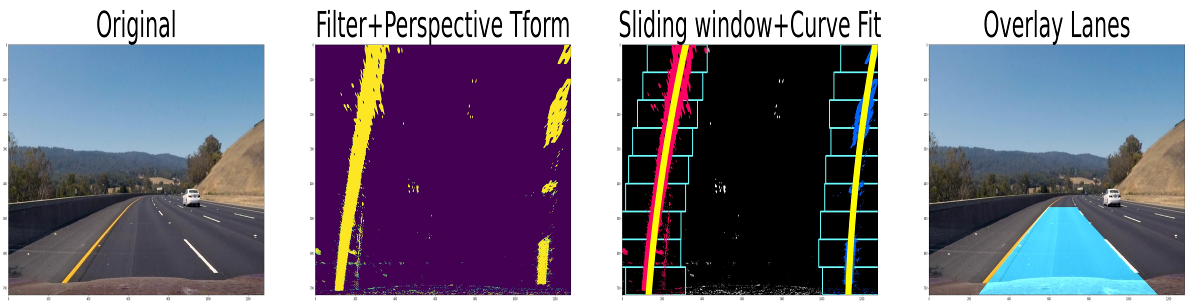


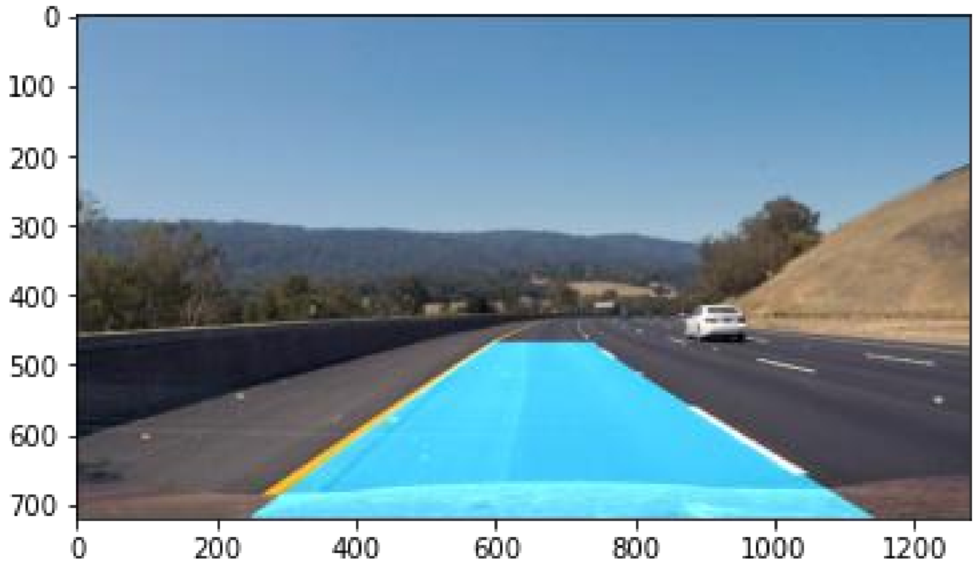
5.10 Apply a distortion correction to raw images

The calibration data for the camera that was collected 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, but if you look closer at right of the image for comparison, this effect becomes more obvious when you look at the white car that has been slightly cropped along with the trees when the distortion correction was applied.

5.11 Pipeline, Perspective view and Sliding Window approach with Curve plotting

A sliding window technique in an image processing applications is a common approach to path mapping from extracted features. Mapping a path inside an image requires finding a series of points representing the path. Previous approaches find these points by sliding a window along the path in fixed increments across one image dimension. After each slide, the center of the window in the other dimension is adjusted so that the window maximally covers the path in that area. This approach, however, fails to map paths that experience sharp curvature since the windows slide along only one dimension.




0
100
200
300
400
500
600
700
0
200
400
600
800
1000
1200

6 RESULTS AND DISCUSSION

This section evaluates the overall performance of the system, after solving most of the problems discovered in earlier developed stages of the scheme.
In a nutshell, this project will successfully be able to perceive lanes and would be providing a smooth flow of automated vehicles.

We made use of two techniques in our project to build a hybrid model:-

1.Colour selection, region of interest selection, grayscaling, gausian smoothing, ***canny edge detection and hough transform line detection***.

The goal was to piece together a pipeline to detect the line segments in the image. Then get an average and draw them onto the image for display. After that we get to the video streaming. The future scope of its potential is that it would be used in almost every self-driving vehicle such that it increases its accuracy and use domain over the time.

2.The second approach uses advanced techniques that builds on the earlier one by using thresholds for different color spaces and gradients, ***sliding window techniques***, warped perspective transforms, and polynomial fits to detect lane lines.

In the near future, eventually every vehicle would be integrated with an automated driving feature and for that to be possible our project i.e., '**VISION**' would be playing an important role.
Not limited to this, with an extensive reach of it, it could be applied to public transport vehicles so that they can be automated to drive themselves.

7 REFERENCES

[1] Abhishek Goyal, Mridula Singh, Anand Srivastava, “Lane Detection on Roads using Computer Vision,” International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-9 Issue-1, October 2019

[2] Y. W. Seo, “Detection and Tracking the Vanishing Point on the Horizon”, CMU-RI-TR-14-07, May 2014

[3] S. Neuendorffer, T. Li, D. Wang, "Accelerating OpenCV Applications with Zynq-7000 All Programmable SoC using Vivado HLS Video Libraries", XAPP1167 (v3.0), Xilinx Inc., USA, June 24, 2015

[4] M. Bellino, Y. L. de Meneses, P. Ryser, Jacot, “Lane Detection Algorithm for an Onboard Camera”, SPIE proceedings of the workshop on Photonics in the Automobile, 2004

[5] G. Velez, O. Otaegui, “Embedded Platform for Computer Vision-Based Advanced Driver Assistance Systemss: A Survey”, Donostia-San Sebstian, Spain

[6] S. Matuska, R. Hudec, M. Benco, “The Comparison of CPU Time Consumption for Image Processing Algorithm in Matlab and OpenCV”, IEEE 978-1-4673- 1179-3/12, 2012.

[7] Y. Wang, D. Shen, E. K. Teoh, “Lane Detection using Spline Model”, Pattern Recognition Letters, 21, 677-689, 2000

[8] G. Hamarneh, K. Althoff, R. Abu-Gharbieh, "Automatic Line Detection", Project Report for the Computer Vision Course, Sept. 1999.

[9] G. P. Stein, E. Rushinek, G. Hayun, Amnon Shashua, “A Computer Vision System on a Chip: A Case Study From The Automotive Domain”, Mobileye Vision Technologies Ltd. www.mobileye.com, Israel

[10] W. Anne, "Embedded Vision: Which standards are necessary to prepare the sector for the future?", Artikel Id:10857677, www.ibv.vdma.org, Nov. 2015