GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

PREDICTION OF ROAD LANE WIDTH

A Project Report of Capstone Project 2

Submitted by

Vishal Singh Rathore

(1613101848)

in partial fulfillment for the award of the degree of

Bachelor of Technology

IN

Computer Science and Engineering

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the Supervision of
ABHAY KUMAR, M.Tech.,
Assistant Professor

APRIL / MAY- 2020

SCHOOL OF COMPUTING AND SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report **"PREDICTION OF ROAD LANE WIDTH "** is the bonafide work of "**VISHAL SINGH RATHORE (1613101848)"** who carried out the project work under my supervision.

SIGNATURE OF HEAD

Dr. RAJU SHANMUGAM
MSc,BSc(Mathematics)
Professor & Dean,
School of Computing Science & Engineering

SIGNATURE OF SUPERVISOR

ABHAY KUMAR , M.Tech,
Assistant Professor
School of Computing Science & Engineering

ABSTRACT

This paper describes a novel approach of detecting lanes. The projected width of lanes can be computed precisely by a method of lane detection algorithm to predict the relative position and features of lane markings. Besides, a camera's accurate tilt angle and the lane width can be acquired by dynamic calibration. The proposed approach uses prediction of a lane's projective width and the way of tracking to accelerate lane detections and to describe positions of lane markings on both sides effectively when one side is occluded.

Driving assistance systems and the researches of autonomous vehicles require information of lanes to decide driving routes of vehicles. Obstacle detection systems are also applied to determine the positions of obstacles, which is important to driving safety. Obstacles inside driving lanes require more attention, so information of obstacles and driving lanes is needed to judge the impact of front obstacles on the driving safety.

In this paper, information of lanes is calculated with geometric projection and dynamic calibration. A Finite State Machine (FSM) is applied to the extraction of lane features, and then is combined with lane tracking to capture complete information of lanes with the known information of one side of the lane when the other side is occluded.

Lane detection can be started from capturing lane features. On most occasions, there are lane markings on both sides of the driving lane, while sometimes only the edges of the road exist without any lane marking. Most parts of the lane markings are like two parallel ribbons with some changes, for example, being straight or curve, being solid lines or dash lines, and in the color of white, yellow or red.

In addition, edge detection requires features of edges on a lane's both sides. If obstacles such as the preceding vehicles occlude one side, errors may arise.

TABLE OF CONTENTS

List of Acronyms

ABS . . . . . Antilock Braking System

ACC . . . . . Adaptive Cruise Control

ADAS . . . . . Advanced Driver Assistance System

AHO . . . . . Autonomous Highway Overtaking

AHS . . . . . Automated Highway System

AV . . . . . Autonomous Vehicle

BSM . . . . . Blind Spot Monitoring

CACC . . . . . Cooperative Adaptive Cruise Control

CC . . . . . Cruise Control

List of Figure

List of Symbols

a . . . . . Acceleration of the host vehicle (m/s2)

dHLO . . . . . Initial distance between the host and lead vehicle (m)

dHFO . . . . . Initial distance between the host and follower vehicle (m)

ddetection . . . . . Distance at which the host vehicle radar detects lead vehicle (m)

LH . . . . . Wheelbase of the host vehicle (m)

lw . . . . . Lane width (m)

tH . . . . . Headway time (s)

tLLC . . . . . Time taken by host vehicle for the left lane.

Chapter 1
Introduction

1.1 Overview

This is the final report of the master thesis project "Autonomous highway overtaking" carried out at HAN
Automotive Research, in partial fulfilment of the requirements for the degree "Master of Science in Automotive Systems" at HAN University of Applied Sciences, Netherlands.
Overtaking on the highway or two-lane roads is one of the major tra_c safety problems. Making a mistake while
doing this maneuver can lead to terrible accidents. That is why all the e_orts to develop driving aids for this operation are one of the main issues of Intelligent Transportation Systems (ITS). This has given rise to the need for the development of overtaking assistance systems . The basic idea is to describe functions capable of mitigating accidents caused by mainly longitudinal and lateral movement of the vehicle by providing comfort and safety to the driver during an overtaking maneuver. Overtaking ADAS can be grouped according to three phases of driving
Using the machine learning algorithms and python it will predict the width of the road .It will also tell the distance from the roadside and also guide for overtaking the vehicles.The trained machine become the powerful feature in the driverless car as the vehicle get proper assistance in the overtaking and parking also .It can also used in the cruise control of the car make the more impact of the car and easy in controlling the car.

Active safety is currently a key topic in the automotive industry, which fosters the development of Autonomous Vehicle functions. Various advanced driver assistance systems (ADAS) and active safety systems have the potential to improve road safety, driver comfort, fuel economy, and traffic flow by assisting

the driver during different driving conditions. It is estimated that human error is a contributing factor in more than 90% of all accidents. In order to save the human lives caused by road accidents, it is hence of interest to develop such systems using modeling and simulation tools which is quick and more efficient as compared to the real driving testing. Overtaking is one of the most complex maneuvers with the high risk of collision (75% human error) so the automation of this maneuver still remains one of the toughest challenges in tdevelopment of autonomous vehicles.

Since overtaking is one of the complex maneuvers and so many factors affect it, the automation of this maneuver has been considered to be one of the toughest challenges in the development of autonomous vehicles. Overtaking involves a great interaction between both longitudinal (throttle and brake) and lateral (steering) actuators. Nowadays, in the field of driver assistance systems and automated driving, development approaches for lateral maneuver control are the very big challenge.

1.2 Motivation

"Self-driving cars are the natural extension of active safety and obviously something we should do." _ Elon Musk

The automobile has become a part of many lives since it is estimated that there are more than 1.28 billion automobiles as of 2015 around the world. Yet, many lives are lost in vehicle accidents. Several groups of collected data from a crash survey reveal that 94% of the crashes are accounted by drivers, 2 percent of the crashes are accounted by vehicle components, 2% of the crashes are accounted by the environment, and the other 2% of the crashes are accounted by unknown reasons . Driver's error contributes to over 75% of road crashes especially in overtaking maneuvers . The World Health Organization (WHO) reports that more than 1.2 million people each year or 3000 persons per day die globally as a result of tra_c accidents. By the year 2030, road traffic accidents are estimated to be the fifth most common cause of death in the world, after heart disease, strokes, emphysema and other respiratory tract infections [15]. Looking beyond

the fatality number, it is estimated that 1,35,000 peopleare seriously injured on European roads each year (European Commission, 2017).

Figure 1.1 shows the proportion of lane change accidents with passenger cars as the main causal factor for the road types of urban, rural, and motorway for the years 1985–1999 . It is clear that on average more than 5% of all accidents take place during a lane change. Lane change crashes constitute around 2.9% of the total accidents on motorways . It is also clear that a majority of these accidents occur on motorways or trunk roads (Accident database of Volkswagen Accident Research and GIDAS).

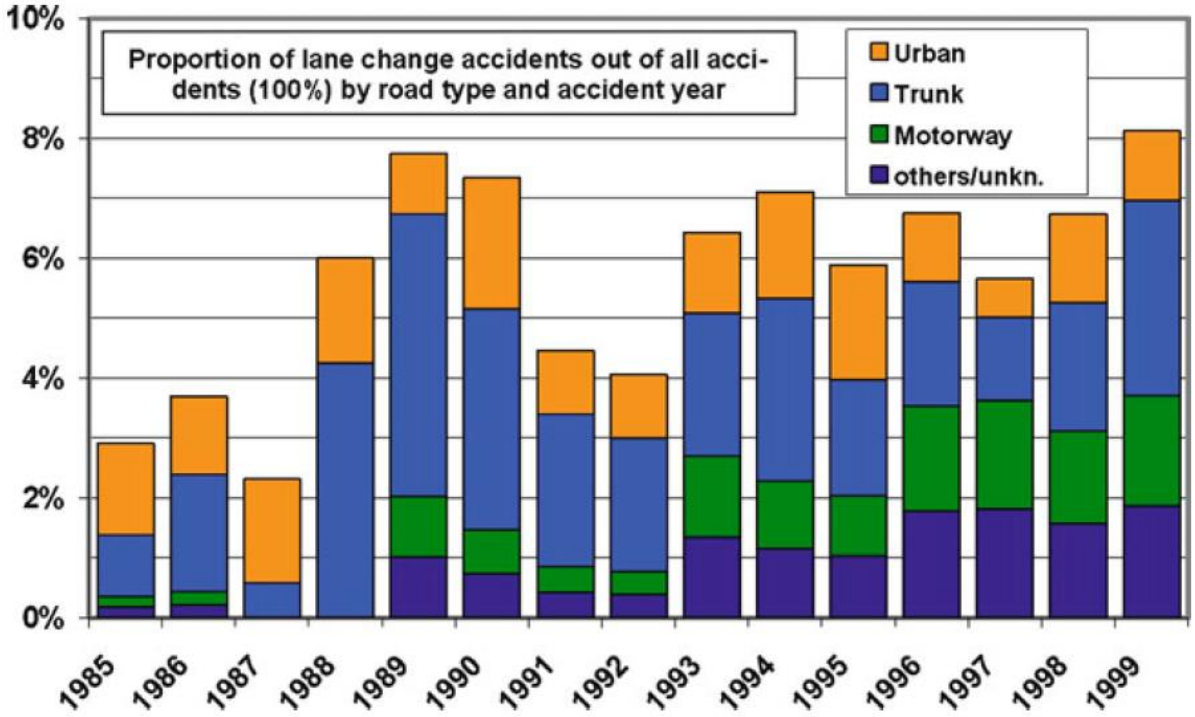


Figure 1.2 Proportion of lane change accidents with passanger cars .

1.3 Problem Definition

To better understand the problem and the scope of this thesis, this section formulates the problem, defines the scenario of interest, and states the objectives and main research questions involved. Finally, a list of assumptions considered for this thesis is given.

One of the problems is to avoid collisions in autonomous highway overtaking and to keep the maximum comfortand safety while executing the

maneuver. The scope of this thesis lies in the development of test protocol and automated driving system that autonomously controls the lateral and longitudinal movement of the host vehicle.

Chapter 2

Literature review

2.1. Machine Learning

Machine learning is a branch of artificial intelligence that aims at solving real life engineering problems. It provides the opportunity to learn without being explicitly programmed and it is based on the concept of learning from data. It is so much ubiquitously used dozen a times a day that we may not even know it. The advantage of machine learning (ML) methods is that it uses mathematical models, heuristic learning, knowledge acquisitions and decision trees for decision making. Thus, it provides controllability, observability and stability. It updates easily by adding a new patient‘s record. The application of machine learning models on human disease diagnosis aids medical experts based on the symptoms at an early stage, even though some diseases exhibit similar symptoms. One of the important problems in multivariate techniques is to select relevant features from the available set of attributes. The common feature selection techniques include wrapper subset evaluation, filtering and embedded models. Embedded models use classifiers to construct ensembles, the wrapper subset evaluation method provides ranks to features based on their importance and filter methods rank the features based on statistical measurements.

2.2 Computer aided diagnosis systems

A computer aided medical diagnosis system generally consists of a knowledge base and a method for solving an intended problem. On the basis of the query posted to the system, it provides assistance to the physicians in diagnosing the patients accurately. The knowledge base of such medical systems relies on the inputs that spring up from the clinical experience of field experts. Knowledge acquisition is the process of transforming human expert knowledge and skills acquired through clinical practice to software. It is quite time consuming and labor intensive task. Common methods like Case Based Reasoning (CBR) solves

the knowledge acquisition problem to some extent because the past records are maintained in a database, including possible remedies, past clinical decisions, preventive measures and expected diagnostic outcome measures. During patient diagnosis, the clinical database is matched for analogous past patient‘s record for taking suitable decisions. Some of the major problems faced during the development of an expert diagnosis system are: medical experts are less interested to share their knowledge with others, experience knowledge (called common sense) is practically impossible to be separated and designing a unique expert system for diagnosing all diseases is difficult.

2.3. Software reliability

Software reliability is defined as the probability that a system will not have a failure over a specified period of time under specific conditions. The knowledge of software reliability is very vital in critical systems because it indicates the design perfection . In 11 this work, the primary aim is to enhance the software reliability of the computer aided diagnosis systems using machine learning algorithms. To provide quality treatment and prevent misdiagnosis are the prime motivations for developing a medical diagnosis system. Diagnosing a disease of a patient accurately is a great challenge in medical field. A huge amount is spent on advanced primary health care devices based on software reliability research as they are considered as critical systems. There are several software reliability models available in the literature; however, none of the models are perfect.

An important research issue is choosing a suitable estimation model based on a specific application. One advantage of software reliability over hardware reliability is that a mechanical part surely undergoes ageing; suffer from wear and tear problem over time and usage; however software do not rust or wear out. Software reliability is a vital parameter for software quality, functionality and performance. Some common software reliability models are prediction and estimation models like bathtub curve, exponential, Putnam etc.

2.4. Supervised learning

Supervised learning is the most common form of machine learning scheme used in solving the engineering problems . It can be thought as the most appropriate way of mapping a set of input variables with a set of output variables. The system learns to infer a function from a collection of labeled training data. The training dataset contains a set of 12 input features and several instance values for respective features. The predictive performance accuracy of a machine learning algorithm depends on the supervised learning scheme. The aim of the inferred function may be to solve a regression or classification problem. There are several metrics used in the measurement of the learning task like accuracy, sensitivity, specificity, kappa value, area under the curve etc. In this work, the aim is to classify the patients as healthy or ill based on the past medical records. Before solving any engineering problem, it is vital that it is necessary to choose a suitable algorithm for the training purpose based on the type of the data.

The selection of a method depends primarily on the type of the data as the field of machine learning is data driven. The next important aspect is the optimization of the chosen machine learning algorithms.

2.5. Classification task

Classification task is a classical problem in the field of data mining which deals with assigning a pre-specified class to an unknown data.

A learning model is built based on the relationship between the predictor attribute values and the value of the target. The challenge is to correctly predict the class based on learning of past data. In machine learning, this kind of classification problems are referred to as supervised learning. Hence, we need to provide a data set containing instances with known classes and a test data set for which the class has to be determined. The success of the classification ability largely depends on the quality of data provided for learning and also the type of machine learning algorithm used . For example, the classification techniques can be used to predict

the fraud customers in a bank who apply for a loan or classify mangoes whether 13 they are good or bad and lots of other real time applications. The most common type of classification problem is binary classification, where the target has two possible values like good or bad, yes or no etc. There are several methods for measuring the classification performance like confusion matrix, lift curve, receiver operator characteristics etc.

2.6. Optimization

Every machine learning algorithm has a specific technique of learning and is based on the values of their parameters. When an algorithm is applied to solve a classification problem with a different set of parameters, the classification accuracy also differs abruptly in each case . The challenge in machine learning to find the most suitable parameter values of the algorithms that solves an engineering problem to the best possible way in terms of performance metrics. Therefore, one has to fine tune the algorithm parameters that best suits the problem. There are several optimization techniques like genetic
algorithm, particle swarm optimization , Tabu search methods
The focus of the study is to calibrate the algorithm parameters using design of experiment method.

2.7. Development Tools

Anaconda (Python distribution)

Anaconda is a free and open-source distribution of Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and

maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as **Anaconda Distribution** or **Anaconda Individual Edition**, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, which are both not free.

Atom

Atom is a free and open-source text and source code editor for macOS, Linux, and Microsoft Windows with support for plug-ins written in Node.js, and embedded Git Control, developed by GitHub. Atom is a desktop application built using web technologies.

Chapter 3

Existing System

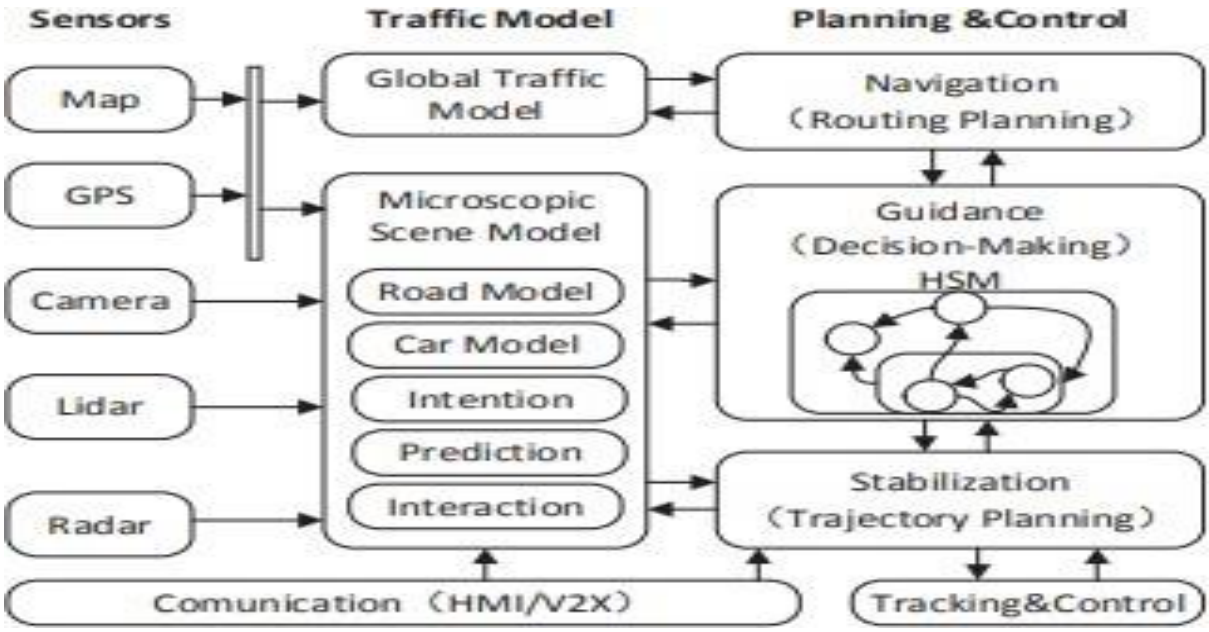


Fig 3.1 Existing System

A tracking problem is usually not an initial value problem, and one needs to harvest the boundary conditions at the final time. Even if one assumes that, a solution to the boundary value problem exists, the nature of the trajectory cannot be guessed from the problem description if the system fails to have an analytic solution, as also pointed out . A numerical procedure is the norm in such cases and requires an assumption that the system remains reachable under the assumed set of initial values and sampling interval.

The possible solution for comfort oriented vehicle following with leading vehicle movement prediction treated as disturbance controller is presented. It is dealing with the execution of optimal , speed trajectory planning is done in a way, that modifying an optimal speed trajectory leads to the smallest deviation from the desired speed while the vehicle is moving. These approaches are treating the problem locally and partially and don't give an energy consumption based decision if a vehicle should overtake.

Chapter 4

Proposed System

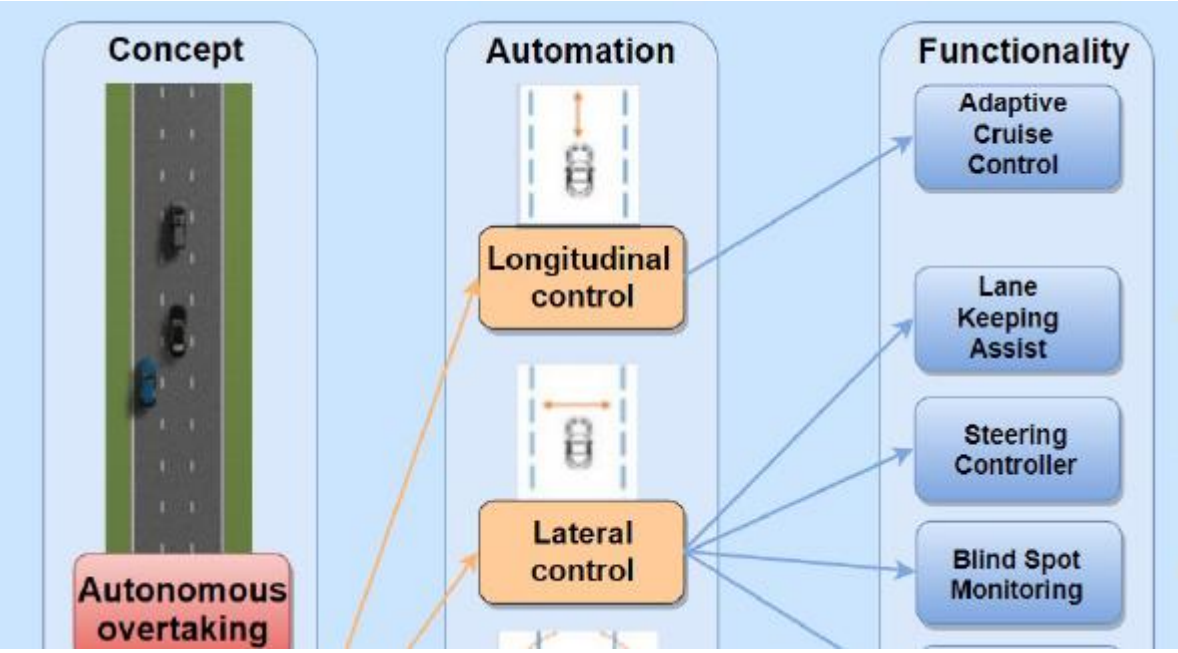


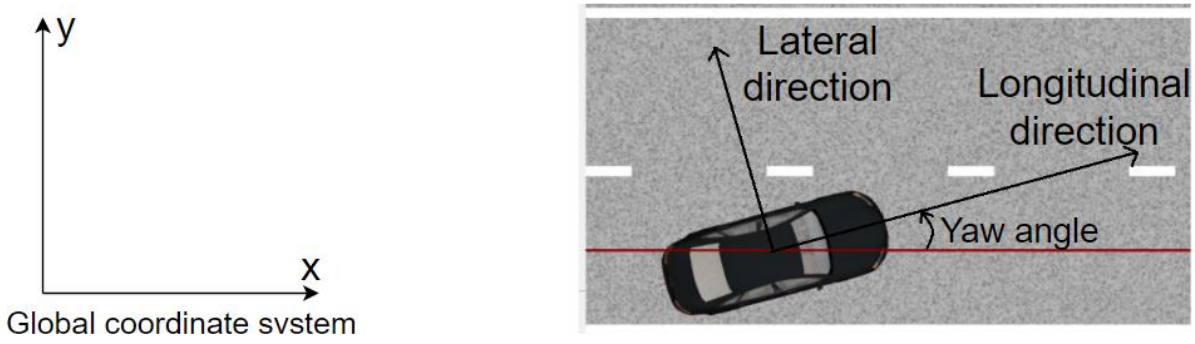
Fig 4.1 Proposed System

This project presents an advanced lane detection technology to improve the efficiency and accuracy of real-time lane detection . The lane detection module is usually divided into two steps: image preprocessing and the establishment and matching of line lane detection model.

Figure 1 shows the overall diagram of our proposed system where lane detection blocks are the main contributions of this paper. The first step is to read the frames in the video stream. The second step is to enter the image preprocessing module. What is different from others is that in the preprocessing stage we not only process the image itself but also do colour feature extraction and edge feature

extraction . In order to reduce the influence of noise in the process of motion and tracking, after extracting the colour features of the image, we need to use Gaussian filter to smooth the image. Then, the image is obtained by binary threshold processing and morphological closure. These are the preprocessing methods mentioned in this project.

Figure 4.2: Direction and angle conventions related to the host vehicle



In this project I will attempt to find lane lines from a dash cam video feed. Once we detect lane lines, we will mark them on the original video frame and play it back. This all will be done online and without any lag using OpenCV functions. Our approach here would be to develop sequence of functions to detect lane lines. We will use a ‘sample’ image while writing this functions, and once we are able to detect lane lines on few ‘sample’ images successfully, we will club complete program into a function which can accept live feed image and return the same image frame with lane lines highlighted.

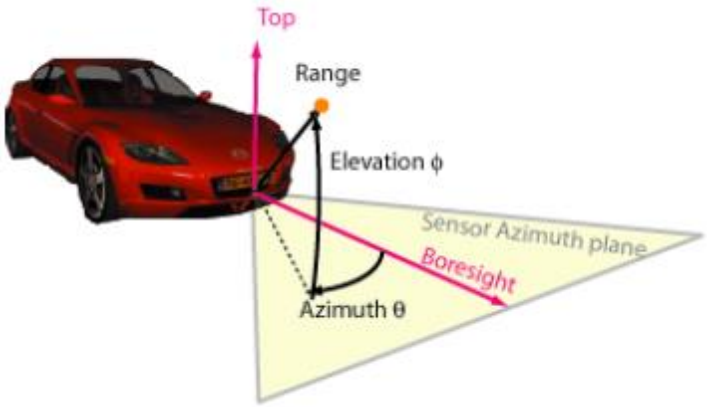


Figure 4.3: Sensor readings

Using the machine learning algorithms and python it will predict the width of the road .It will also tell the distance from the roadside and also guide for overtaking the. It can also used in the cruise control of the car make the more impact of the car and easy in controlling the car. The trained machine will able to calculate and it will predict the width of the lane that will help to drive the car in its lane with the proper guidance .

In this project , based on the previous preprocessing, we firstly extract the colour features based on the white colour and then extract the edge features based on the straight lane. Because the high-speed section is the traffic accident-prone section, the high-speed road section mostly is the straight line lane . Therefore, in order to obtain a very high recognition rate, we successively carry on colour detection and edge detection to the lane. This paper combines colour features extraction and edge features extraction, and the experiment proves that the recognition rate and accuracy of lane detection are greatly improved.

Our main contribution in this paper is to do a lot of work in the preprocessing stage. We proposed to perform colour transform of HSV in the preprocessing stage, then extract white, and then perform conventional preprocessing operations in sequence. Moreover, we selected an improved method proposed in the area of

interest (ROI). In this paper, based on the proposed preprocessing method (after HSV colour transform, white feature extraction, and basic preprocessing), one-half part of the processed image is selected as the area of interest (ROI). In addition, we performed twice edge detection. The first is in the preprocessing stage, and the second is in the lane detection stage after the ROI is selected. The purpose of performing twice edge detection is to enhance the lane recognition rate.

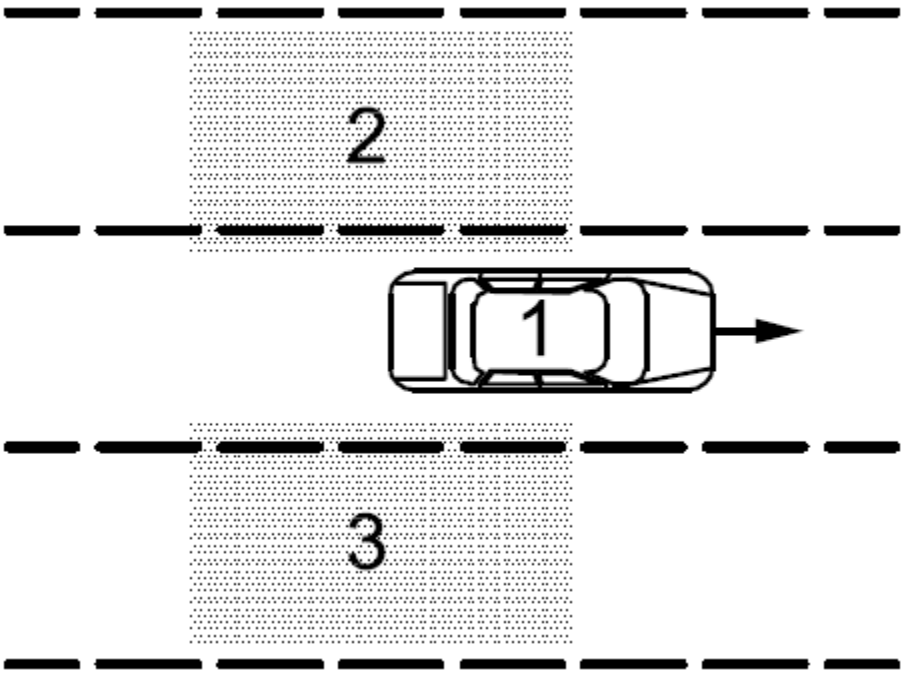


Figure 4.4: Adjacent zones

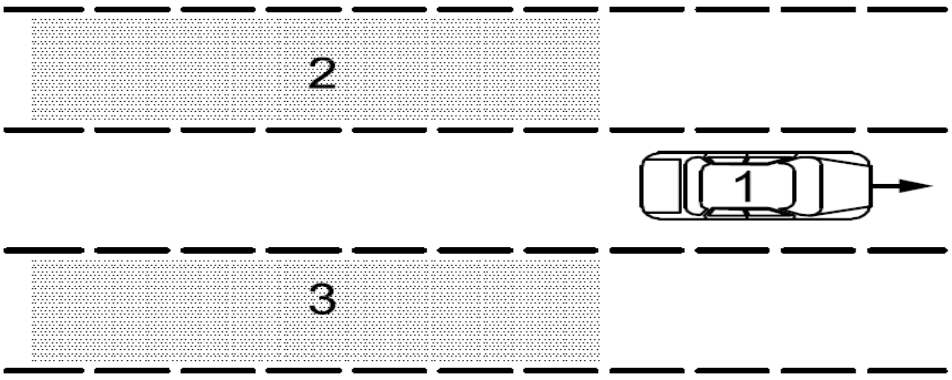


Figure 4.5: Rear zones

Canny Edge Detecter Algorithm -The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986. Canny also produced a *computational theory of edge detection* explaining why the technique works.

Canny edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed.

- Detection of edge with low error rate, which means that the detection should accurately catch as many edges shown in the image as possible
- The edge point detected from the operator should accurately localize on the center of the edge.
- A given edge in the image should only be marked once, and where possible, image noise should not create false edges.

Process Of Canny Edge Detection Algorithm

1. Apply Gaussian filter to smooth the image in order to remove the noise
2. Find the intensity gradients of the image
3. Apply non-maximum suppression to get rid of spurious response to edge detection
4. Apply double threshold to determine potential edges
5. Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges

Block Diagram

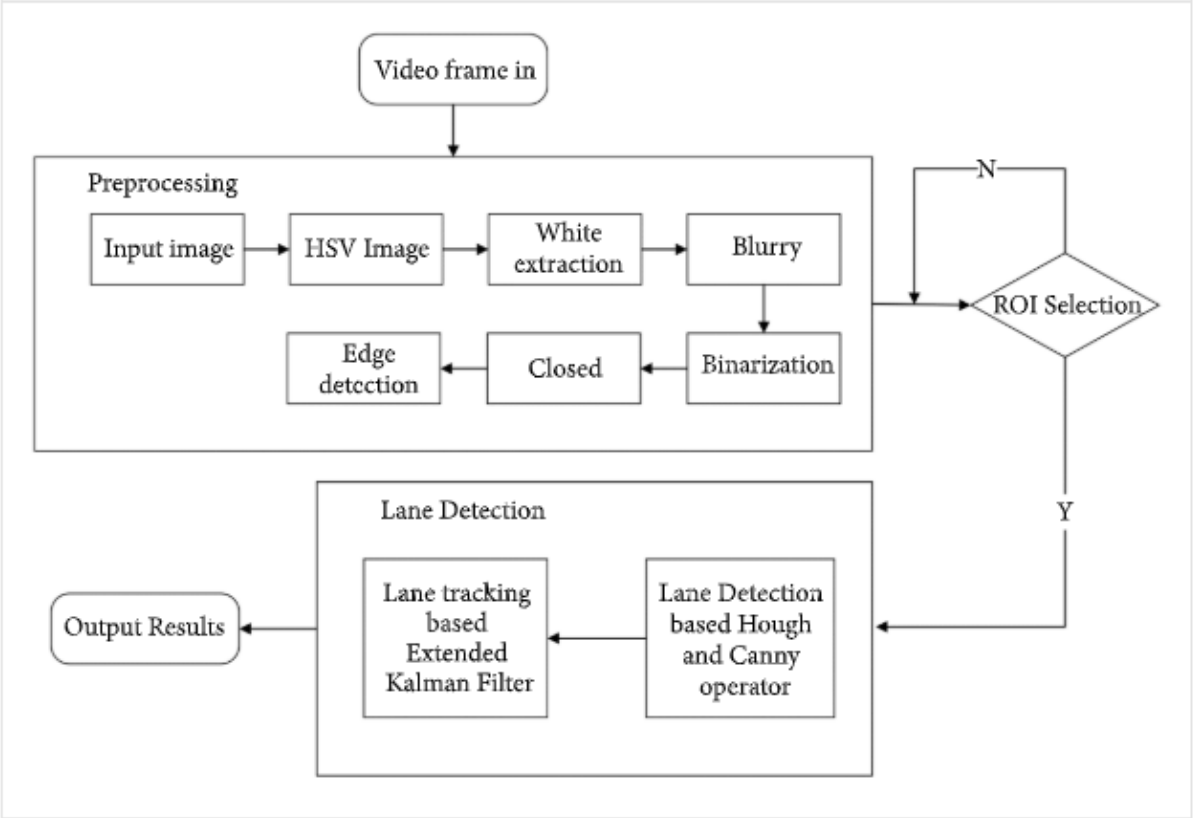


Figure 4.6: Block Diagram

Chapter 5

Methodolgy

5.1 Processing a video

We will feed in our sample video for lane detection as a series of continuous frames (images) by intervals of 10 milliseconds. We can also quit the program anytime by pressing the ‘q’ key.

5.2 Applying Canny Detector

The Canny Detector is a multi-stage algorithm optimized for fast real-time edge detection. The fundamental goal of the algorithm is to detect sharp changes in luminosity (large gradients), such as a shift from white to black, and defines them as edges, given a set of thresholds. The Canny algorithm has four main stages:

Figure 5.1: Canny Edge Detection

5.2.1 Noise reduction

As with all edge detection algorithms, noise is a crucial issue that often leads to false detection. A 5x5 Gaussian filter is applied to convolve (smooth) the image to lower the detector's sensitivity to noise. This is done by using a kernel (in this case, a 5x5 kernel) of normally distributed numbers to run across the entire image, setting each pixel value equal to the weighted average of its neighboring pixels.

$$\mathbf{B} = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * \mathbf{A}$$

5.2.2 Intensity gradient

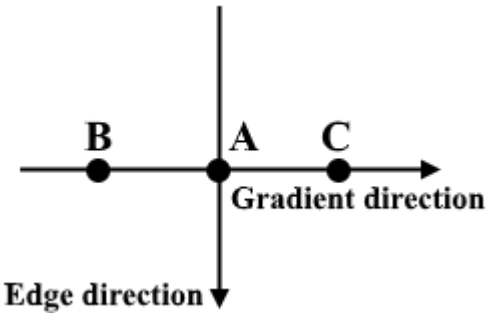
The smoothened image is then applied with a Sobel, Roberts, or Prewitt kernel (Sobel is used in OpenCV) along the x-axis and y-axis to detect whether the edges are horizontal, vertical, or diagonal.

$$Edge\_Gradient\ (G) = \sqrt{G_x^2 + G_y^2}$$

$$Angle\ (\theta) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

5.2.3 Non-maximum suppression

Non-maximum suppression is applied to “thin” and effectively sharpen the edges. For each pixel, the value is checked if it is a local maximum in the direction of the gradient calculated previously.



A is on the edge with a vertical direction. As gradient is normal to the edge direction, pixel values of B and C are compared with pixel values of A to determine if A is a local maximum. If A is local maximum, non-maximum suppression is tested for the next point. Otherwise, the pixel value of A is set to zero and A is suppressed.

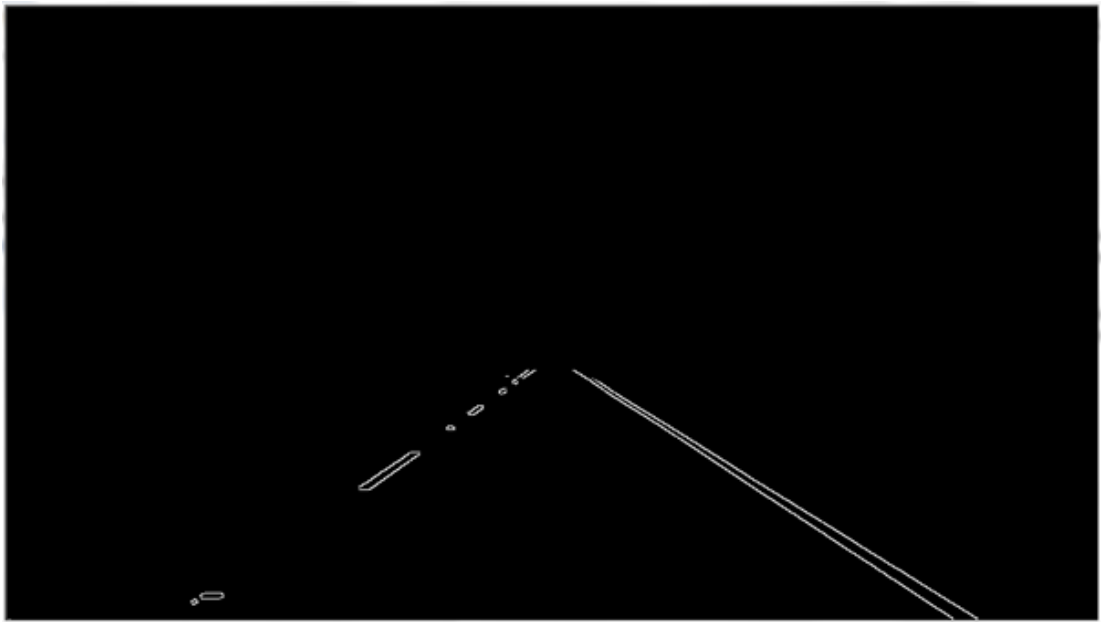


Figure 5.2: Point Of Interest

5.2.4. Hysteresis thresholding

After non-maximum suppression, strong pixels are confirmed to be in the final map of edges. However, weak pixels should be further analyzed to determine whether it constitutes as edge or noise. Applying two pre-defined minVal and maxVal threshold values, we set that any pixel with intensity gradient higher than maxVal are edges and any pixel with intensity gradient lower than minVal are not edges and discarded. Pixels with intensity gradient in between minVal and maxVal are only considered edges if they are connected to a pixel with intensity gradient above maxVal.

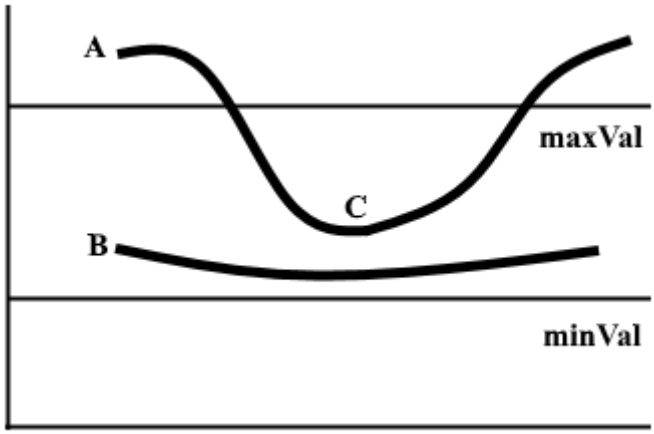


Figure 5.3: Hysteresis Example of two lines

Edge A is above maxVal so is considered an edge. Edge B is in between maxVal and minVal but is not connected to any edge above maxVal so is discarded. Edge C is in between maxVal and minVal and is connected to edge A, an edge above maxVal, so is considered an edge.

For our pipeline, our frame is first grayscaled because we only need the luminance channel for detecting edges and a 5 by 5 gaussian blur is applied to decrease noise to reduce false edges.

5.2.5 Segmenting lane area

We will handcraft a triangular mask to segment the lane area and discard the irrelevant areas in the frame to increase the effectiveness of our later stages.

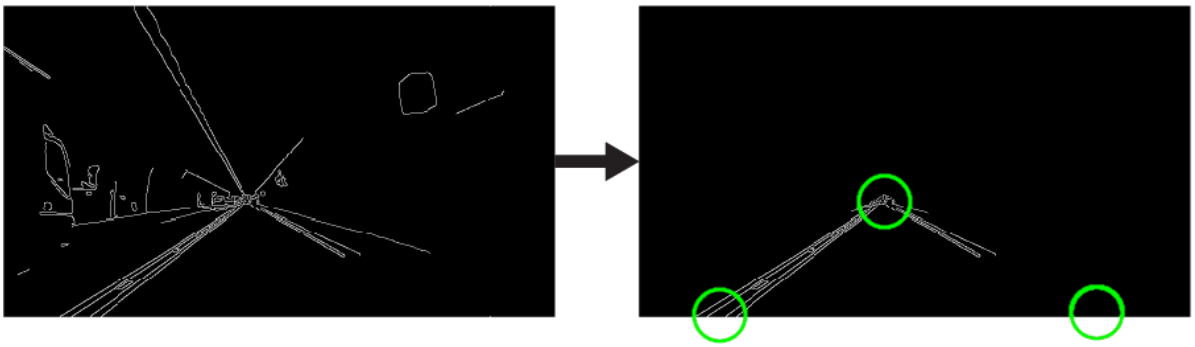


Fig:5.4 The triangular mask will be defined by three coordinates, indicated by the green circles.

Chapter 6

Implementation

Source code :

```
import cv2

import numpy as np

def make_coordinates(image,line_parameters):

    slope,intercept=line_parameters

    print(image.shape)

    y1=image.shape[0]

    y2=int(y1*(3/5))

    x1=int((y1-intercept)/slope)

    x2=int((y2-intercept)/slope)

    return np.array([x1,y1,x2,y2])

return



def average_slope_intercept(image,lines):
```

```
    left_fit=[]
    right_fit=[]
    for line in lines:
x1,y1,x2,y2=line.reshape(4)
        parameters=np.polyfit((x1,x2),(y1,y2),1)
        slope=parameters[0]
        intercept=parameters[1]
        if slope <0:
            left_fit.append((slope,intercept))
        else:
            right_fit.append((slope,intercept))
    left_fit_average=np.average(left_fit,axis=0)
    right_fit_average=np.average(right_fit,axis=0)
    left_line=make_coordinates(image,left_fit_average)
    right_line=make_coordinates(image,right_fit_average)
    return np.array([left_line,right_line])
```

```
def canny(image):
    gray=cv2.cvtColor(image,cv2.COLOR_RGB2GRAY)
    blur=cv2.GaussianBlur(gray,(5,5),0)
    canny=cv2.Canny(blur,50,150)
    return canny
def display_lines(image,lines):
    line_image=np.zeros_like(image)
    if lines is not None:
        for x1,y1,x2,y2 in lines:
            cv2.line(line_image,(x1,y1),(x2,y2),(255,0,0),10)
    return line_image
def region_of_interest(image):
    height=image.shape[0]
    polygons=np.array([
    [(200,height),(1100,height),(550,250)]
    ])
```

```
    mask=np.zeros_like(image)
    cv2.fillPoly(mask,polygons,255)
    masked_image=cv2.bitwise_and(image,mask)
    return masked_image
cap=cv2.VideoCapture("test2.mp4")
while(cap.isOpened()):
    _, frame=cap.read()
    canny_image=canny(frame)
    cropped_image=region_of_interest(canny_image)

lines=cv2.HoughLinesP(cropped_image,2,np.pi/180,100,np.array([]),minLineL
ength=40,maxLineGap=5)
    averaged_lines=average_slope_intercept(frame,lines)
    line_image=display_lines(frame,averaged_lines)
    combo_image=cv2.addWeighted(frame,0.8,line_image,1,1)
    cv2.imshow("result",combo_image)
```

```
    if cv2.waitKey(1)==ord('q'):
        break
cap.release()
cv2.destroyAllWindows
```

Chapter 7

Input / Output

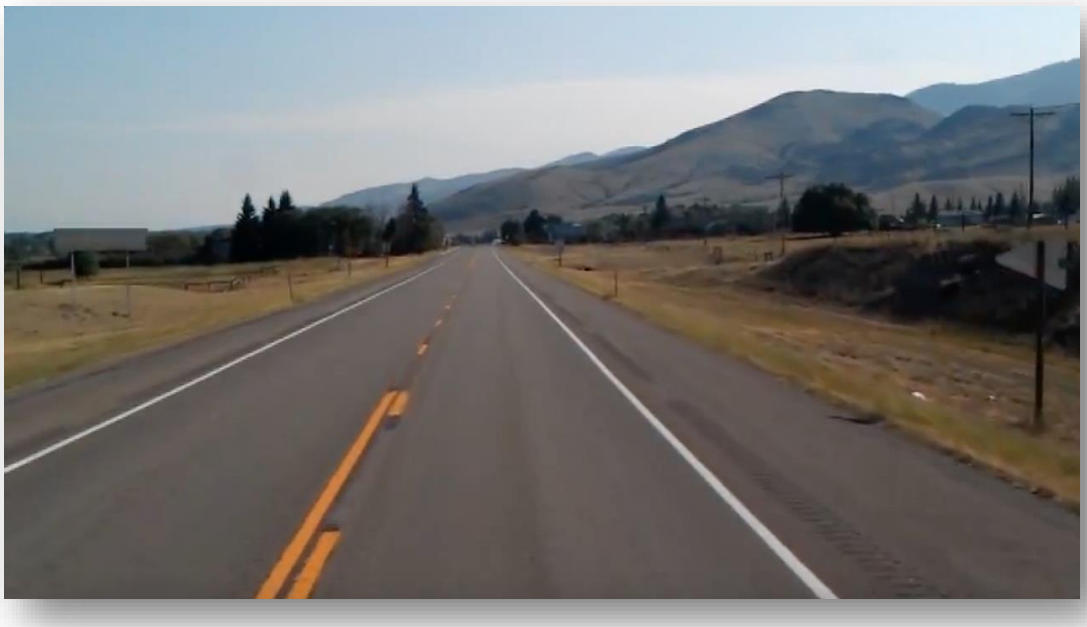


Fig 7.1:Input (Sample Image)

Output Screenshot

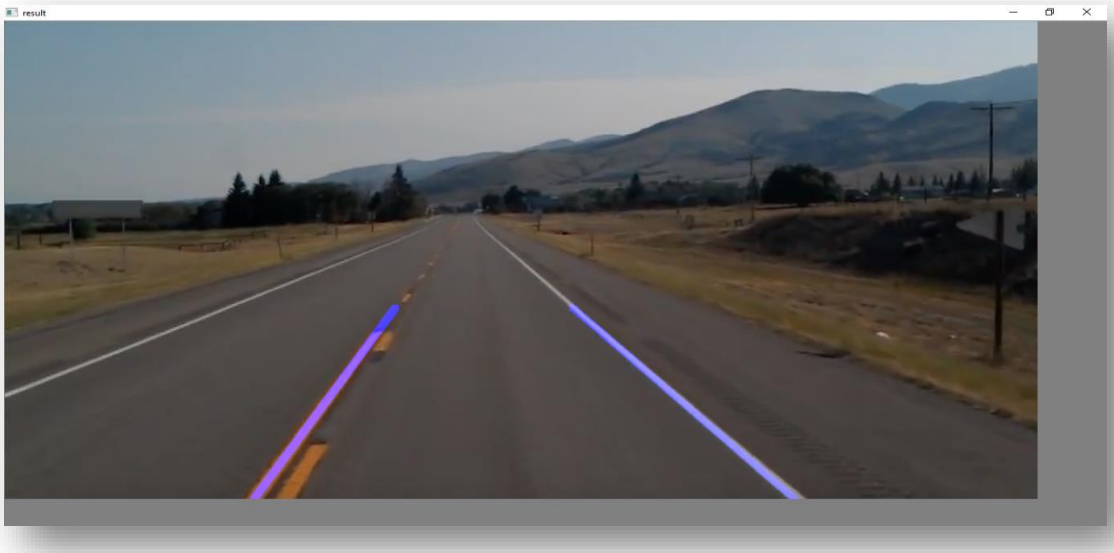


Fig 7.2**: Output Screen**

Chapter 8

Conclusion and Result

In this project , the problem of autonomous highway overtaking was solved. The test protocol for highway overtaking assist was developed which was further used for the development of an automated driving system for the autonomous overtaking. The developed test protocol was validated analytically using mathematical equations and the automated driving system was tested virtually. The simulation results were found to be in accordance with the desired host vehicle behavior. The system drives the host vehicle through the selected use cases in a safe and efficient manner, while interacting with target vehicles operating in the traffic environment. The proposed prediction of road with length overtaking system has the following characteristics:

_ Safe, comfortable, and robust:

while driving.

The characteristics of the fuzzy controller that were observed during its performance validation stage were quite satisfactory. In fact, based on the results of the performance validation, it was concluded that the

developed fuzzy controller was suitable for application to the steering control even at a higher speed. Thus,

the system is robust enough.

divides tactical planning from decision-making and control. Also, the development of subsystems is

performed independently which might be beneficial to the future optimization of the system under study and

implementation into a real vehicle. The system can be said to be optimized for real-time implementation.

Chapter 9

References

1. G. Hegeman, K. Brookhuis, S. Hoogendoorn, et al., “Opportunities of advanced driver assistance systems towards overtaking,” EJTIR, vol. 5, no. 4, pp. 281–296, 2005.
2. F. Tillema, “PRAUTOCOL, Certification of drivers and cars in the field of autonomous driving (RAAKSME), Smart mobility, HAN Automotive Research,” 2017.
3. A. Khodayari, F. Alimardani, and H. Sadati, “Modeling and intelligent control system design for overtaking maneuver in autonomous vehicles,” Iranian Journal of Mechanical Engineering Transactions of the ISME, 2012.
4. “Automated vs. Autonomous Vehicles: Is There a Di_erence?.” https://www.autotrader.com/ car-news/automated-vs-autonomous-vehicles-there-difference-273139, 2018. [Online accessed 25-04-2018].
5. J. E. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, F. Niewels, and J. M. Zöllner, “Testing of advanced driver assistance towards automated driving: A survey and taxonomy on existing approaches and open questions,” in Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on,
pp. 1455–1462, IEEE, 2015.
6. G. Hegeman, Assisted overtaking: An assessment of overtaking on two-lane rural roads. No. T2008/4, TRAIL Research School Delft, the Netherlands, 2008.
7. N. C. Basjaruddin, K. Kuspriyanto, D. Saefudin, E. Rakhman, and A. M. Ramadlan, “Overtaking assistant system based on fuzzy logic,” Telkomnika (Telecommunication Computing Electronics and Control), vol. 13,no. 1, pp. 76–84, 2014.

8. B. W. Smith and J. Svensson, “Automated and autonomous driving: Regulation under uncertainty,” 2015.
9. M. Schaefer, “Collision avoidance of highway tra_c,” Master’s thesis, Czech Technical University in Prague, Faculty of Electrical Engineering, 2014.
10. “Beyond The Headlights:ADAS and Autonomous Sensing, Texas Instruments, WCP,” 2016.