(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

NISHANT KETU

(1613101455/16SCSE101373)

in partial fulfilment 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
Dr. R. RAJKUMAR, M.Tech., Ph.D.,
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 “**NISHANT KETU (1613101455)”** who carried out the project work under my supervision.

SIGNATURE OF HEAD

Dr. MUNISH SHABARWAL,
PhD (Management), PhD (CS)
Professor & Dean,
School of Computing Science & Engineering

SIGNATURE OF SUPERVISOR

Dr. R. RAJKUMAR, M.Tech., Ph.D.,
Assistant Professor
School of Computing Science & Engineering

Abstract

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. The scope of this thesis is to develop the test protocol and automated driving system for the normal or accelerative and flying overtaking maneuver in a highway context.

One of the main challenges for advanced driver assistance systems (ADAS) is the environment perception problem. One factor that makes ADAS hard to implement is the large amount of different conditions that have to be taken care of. The main sources for condition diversity are lane and road appearance, image clarity issues and poor visibility conditions. A review of current lane detection algorithms has been carried out and based on that a lane detection algorithm has been developed and implemented on a mixed criticality platform. The final lane detection algorithms consists of preprocessing steps where the image is converted to gray scale and everything except the region of interest (ROI) is cut away. OpenCV, a library for image processing has been utilized for edge detection and hough transform. An algorithm for error calculations is developed which compares the center and direction of the lane with the actual vehicle position and direction during real experiments. The lane detection system is implemented on a Raspberry Pi which communicates with a mixed criticality platform. The demonstrator vehicle can achieve a measured speed of 3.5 m/s with reliable lane keeping using the developed algorithm.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS/ABBREVIATIONS

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Using the machine learning algorithm and python it will predict the width of 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 the development of autonomous vehicles.

1.2 OVERALL DESCRIPTION

There is a global trend to make vehicles more autonomous to reduce human error and workload. Most modern vehicles include safety-critical systems where a failure can cause great damage to both humans and the environment. When the implementation is a safety-critical system it is important to be aware of the risks that are present and how to cope with them. One other increasingly important trend in the design of real-time and embedded systems is the integration of components with different criticality onto the same hardware platform.

The individual goal and expected outcome from this thesis is a study of existing road and lane detecting systems. Then comparing different systems to determine which is suitable for implementation in the safety critical system that the group is developing. The last part of the project is to implement the lane keeping algorithm on the RTOS of the mixed criticality system to demonstrate the functionality.

This degree project will comply with the applied research methodology where information is gathered from accepted and well-known sources and applied to solve specific problems. To gain knowledge in the field of lane detection systems a literature study will be performed which will guide the development direction of the project.

The work performed in this project is carried out in an as ethical and sustainable way as possible. As always when dealing with automation, it can be important to consider how the system will be used and how the people involved will be affected. One big concern when dealing with automated vehicles is how the decisions are made in situations where accidents occur. In fact there will not even be accidents, but rather decisions made by the computer in the car that led to the situations. The era of automated vehicles will also introduce completely new security threats as the computers in the vehicles can be hacked and overtaken which can lead to injuries and death. Only when the system has been confirmed as safe and secure it can be deployed to real production vehicles.

1.3 PURPOSE

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. This review will explore the relationship between lane width and the safety of Vehicles on urban roads. Research into roadway design and drivers safety has been limited. The review will then examine the effect of lane width on driver (motor vehicle) behavior and then on drivers behavior and safety.

Reduced motor vehicle speeds can significantly improve vulnerable road user, particularly pedestrians and cyclists, safety. It has been shown that if road lane widths on urban roads were reduced, through various mechanisms, it could result in a safety environment for all road users.

1.4 MOTIVATION AND SCOPE

The scope of this thesis is to develop the test protocol and automated driving system for the accelerative/normal and flying overtaking maneuver in a highway context. Firstly, the test protocol for the overtaking maneuver is developed which is in accordance with the ISO standards (ACC), Euro NCAP (safety assist protocols), and various rules and regulations set by few governments (The Netherlands, UK, and Province of Alberta) for overtaking maneuver. Secondly, an automated driving system for the overtaking maneuver is developed which can be realized as a combination of decision-making algorithm.

An algorithm for error calculations is developed which compares the center and direction of the lane with the actual vehicle position and direction during real experiments.

The individual goal and expected outcome from this thesis is a study of existing road and lane detecting systems. Then comparing different systems to determine which is suitable for implementation in the safety critical system that the group is developing.

The scope of this work extends to investigating lane detection and platoon driving for small vehicles operating in a constructed environment. Machine learning approaches for lane detection are not within the scope of the project. The results will to some extent depend on the platform that the use case is built upon. In the case of objects on the track some collision avoidance system will be developed and will initially only consist of an emergency break of the vehicle.

CHAPTER 2

LITERATURE REVIEW

2.1 MOTIVATION

The individual goal and expected outcome from this thesis is a study of existing road and lane detecting system.Then comparing different system to determine which is suitable for implementation in the safety critical system that the group is developing.The scope of this thesis is to develop the test protocol and automated driving system for the normal or accelerative and flying overtaking maneuver in a highway context.

2.2 ARCHITECTURE

2.2.1 ARCHITECTURAL DIAGRAM

Safety issues along urban expressways have gained increasing attention for their direct impact on traffic operation, especially after a substantial amount of reconstruction work that altered the cross section designs to create additional travel lanes.

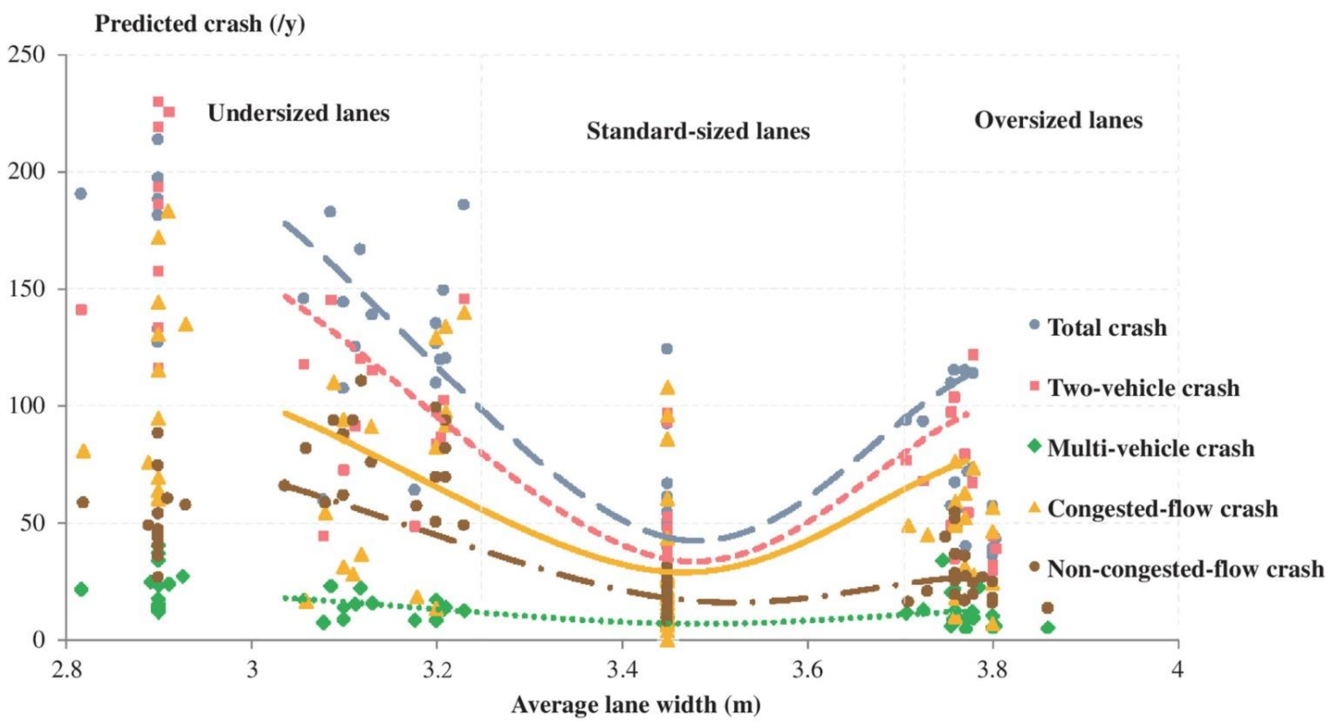


Figure 1 : Architectural Diagram

Here, it examine the impact of lane widths on crash frequency for various types of crashes based on field data obtained from the expressway system Lane width data are grouped into three categories in this diagram : undersized lanes (average lane width ≤3.25 m), standard-sized lanes (average lane width around 3.45 m), and oversized lanes (average lane width≥3.75 m).

Figure 2 : Road Lane Line Detection

2.3 PROPOSED MODEL

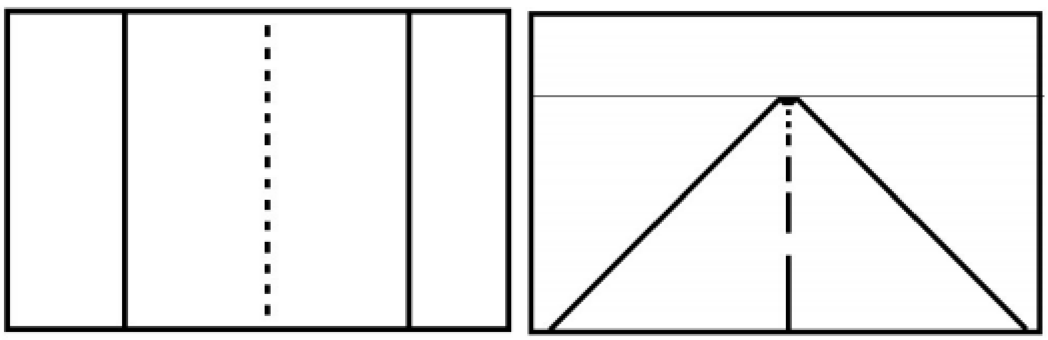
Figure 3 **:** Calibration of the tilt

Figure 4 **:** Compensation for lack of information about lanes.

As here seen in (a), four huge black marks upon drive marking are utilized to analyze angle, and two black arc denotes the last and the first borders. (b) indicates that the last side does not have any knowledge about path labeling while the other side still would be visible. The angular location of the right path labeling can be calculated with the knowledge about the breadth of the path. At last the group are calculated and the two groups with the most number of lines in them are the one that are considered as lanes.

2.3.1 ROAD MODEL

The majority of lane detection systems initially propose a model of a road. This model can be both something simple as straight lines or more complex splines. Some researchers make the assumption that the road is two parallel lines in the image, this can be done after an operation called inverse perspective mapping, which produces an bird's-eye view perspective . One other common method is to assume that the lanes have a common vanishing point where the both lanes meet and use that as a reference for the lines in the image. Both these two perspectives can be seen below in figure



(a) Inverse perspective mapping (b) Vanishing point

Figure 5 **:** Road model

2.3.2 CAMERA MODEL

The location of any place in this 3-D globe coordinates (*X*, *Y*, *Z*) forecasted to a 2-D representation level(*u*, *v*) can be computed using viewpoint and position variation. Plotting a 3-D location on to the 2-D representation stage is a many-to-one conversion . The *Y*, *Z* coordinates are linked to the *v*-coordinate on which the end is forecasted, while the *X* and *Z* coordinates requires the forecasting on to the *u*-coordinate. In contradiction, plotting a place on the frontline view of the camera on an image plane is a one-to-one conversion. Because the place is on the view, it's *Y*-coordinate can be concluded, and thus the *Z*-coordinate controls where on the *v*-coordinate the point is forecasted.

The three gradients are computed clockwise when glancing along the turning axis concerning the origination. Figure 1 shows a camera model for figure development method, where *Ow* represents the origination of the globe coordinates (*X*, *Y*, *Z*), and *Oi* denotes the origination of the image coordinates (*u*, *v*, *w*). Hence, this process can be used to approximate the interspace connecting the camera and a point *P*1.

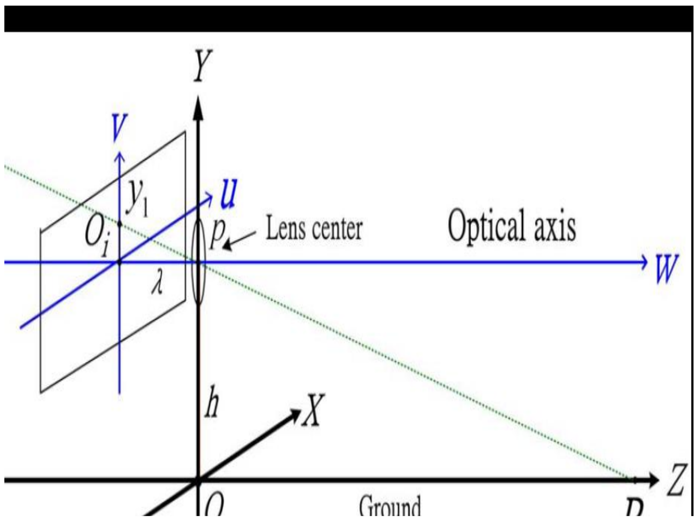


Figure 6 : Camera model for image formation.

2.4 TECHNOLGY USED

2.4.1 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.

2.4.2 PYTHON

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991. Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc) and runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick. It can be used for rapid prototyping, or for production-ready software development and can connect to database systems. It can also read and modify files.

2.4.3 LIDAR

Light detection and ranging (LIDAR) is a modality that has been used to a large extent in the development of autonomous vehicles for research purposes. The LIDAR measures the environment around the vehicle in 3D. The LIDAR sends out light pulse and measures the time for it to come back. As it is an active light source it is not dependent of having good natural lighting as with a regular camera. LIDAR sensors can perform well in certain situations for example in rural areas to detect road boundaries, but are not well suited for multilane roads without vision data. As the LIDAR only measures 3D structures it is not able to detect road markings, although some research have been done on intensity measurement with LIDAR which would make it possible to detect line markings to some extent. One huge drawback with this modality is that the sensors are still very expensive and thus not yet an alternative for implementation in regular passenger vehicles.

2.5 ALGORITHM

2.5.1 LANE DETECTION ALGORITHM

This section will describe the algorithms used for lane detection that have been implemented on the demonstrator of this project.

1. The lane detection process starts with grabbing a frame from the Raspberry Pi camera and applying a few preprocessing steps to the image.
2. The following step is to convert the image to gray scale to prepare it for next coming operations. Figure shows how the acquired image looks in the first stages of the lane detection process.

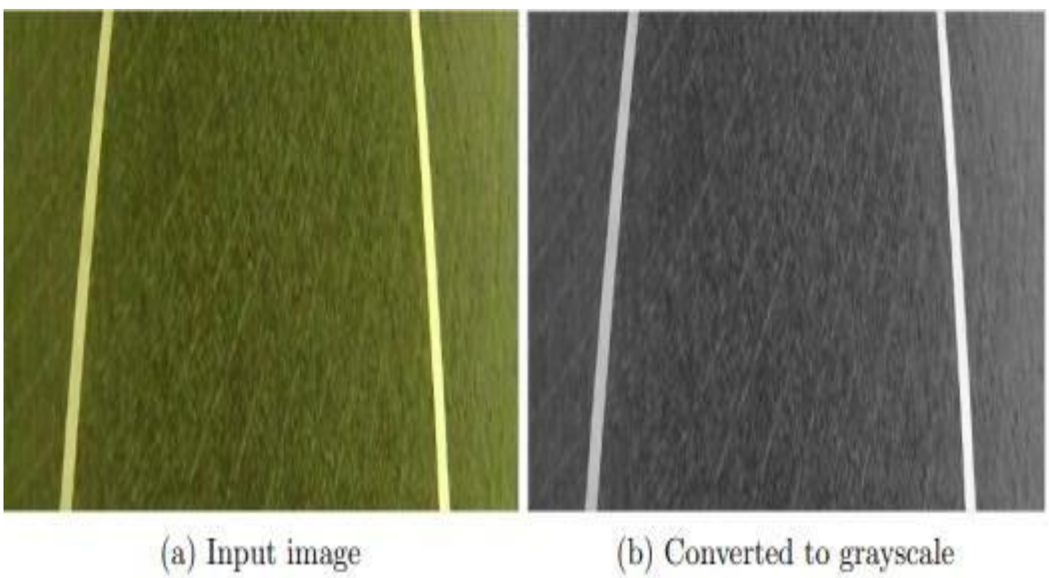


Figure 7 : Lane detection image stage process

The gray scale image is the input to the canny edge detection function. As described in the state of the art section the output of the canny function is a thresholded image where all the pixels that are part of edges are set to white and all pixels that are not part of edges are set black. Using the OpenCV library function Canny.

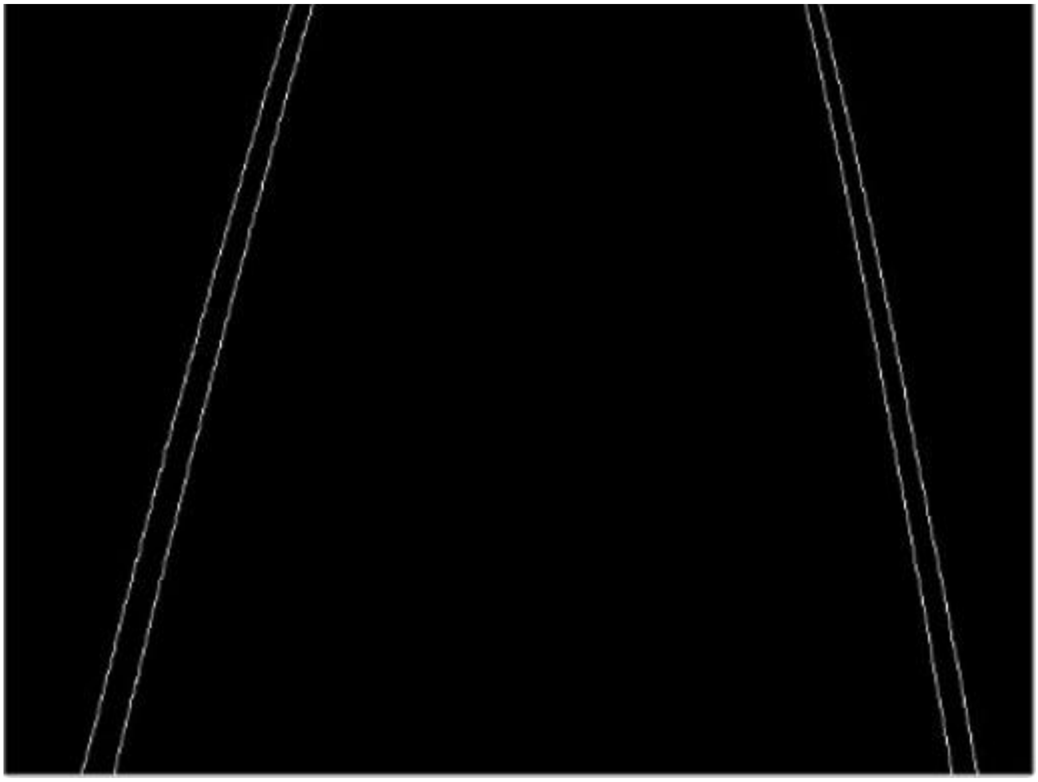


Figure 8 **:** Threshold image(after canny function)

This thresholded image is used as an input to the Hough transform function that is used for line detection. The two figures below show the input image with lines drawn in different colors. The different colors of the lines indicate what kind of line it is. The red lines in the image are all the lines that the lane detection algorithm finds. From the red lines that are close to each other, blue lines indicate the center of the road marking. The green line show the center of the road lane. The concept behind the lane detection algorithm is described in below figure

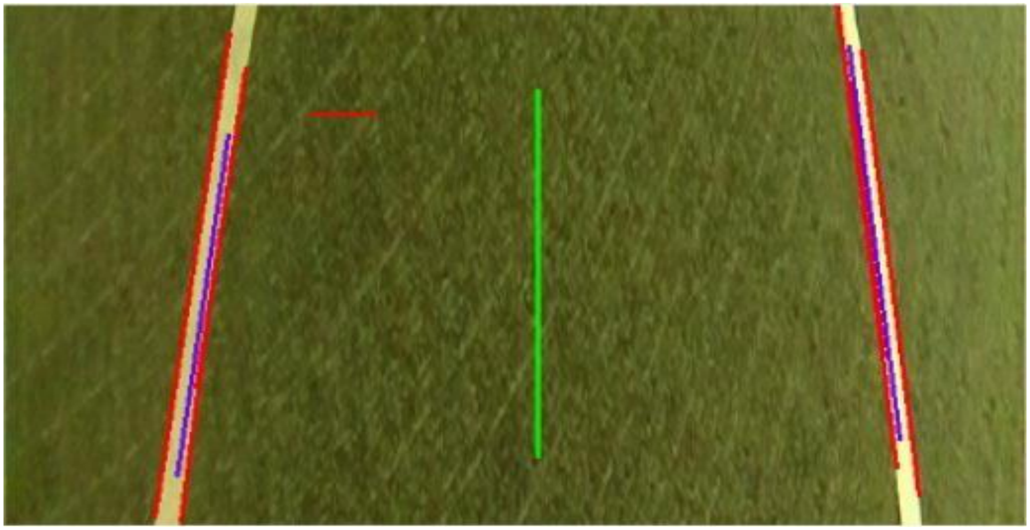


Figure 9 **:** Center of road lane(green lines)

There were a lot of problems when developing the algorithm in terms of false positives when evaluating the lines in the image. A solution was developed in order to eliminate the false lines in the image and only keep the lines that are part of a road lane. The concept behind this lane detection algorithm solution is to group lines in the image that are very close to each other. So for instance if we find several lines on both the left and right lane of the road, these form two groups of lines because the lines are close to each other. If there are other lines in the image that are not very close to these two lanes, they will be put in separate groups. There can be multiple groups depending on how many false lines that are detected. In the end the groups are evaluated and the two groups with the most number of lines in them are the one that are regarded as lanes. The short red horizontal line visible in 3.8 is the threshold distance for lines to be grouped in the same group.

2.6 TECHNIQUES USED

2.6.1 LANE MARKING EXTRACTION

Lane Marking Extraction(LME) Finite State Machine (FSM) was recommended to take out the characteristics of byroad labeling. Two points, *PA* and *PB*, are placed in each tier for observation. The gap connecting the two points is *dm* . When *PA* and *PB* proceed concurrently from the left to the right, the variation between their gray levels *Gd*, would transform respect to them. Every time when proceeding single element to the right, a new *Gd* named *Gin* rises. *Gin* is an input signal of LME FSM. If properties of path labeling occurs in the region m where *PA* and *PB* are proceeding, the input of *Gin* would form the situation of LME FSM changes from state 0 one by one to state 5. Hence, the location and dimension of each path labeling can be observed based on the changes of the state.

$$dm = ratio \times wm \ (N)$$

where *dm* means the interspace between *PA* and *PB*. In Fig. 4, when properties of path labeling occurs in their image, their respective situations to *PA* and *PB* may be five feasible types because of the right move of *PA* and *PB*.

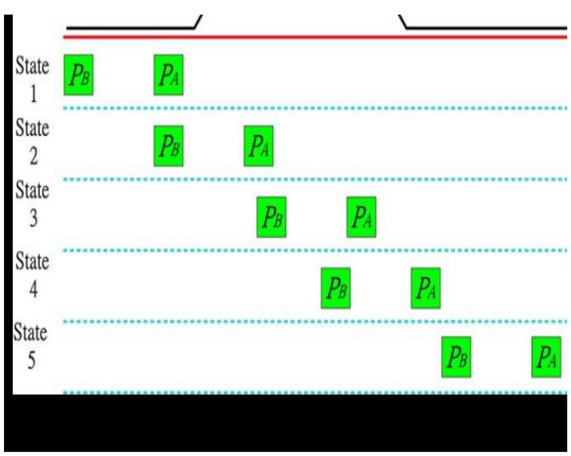


Figure 10 **:** In different states, lane marking's relative positions to *PA* and *PB*.

2.6.2 SINGLE AND TRACKING MODE

The capture of path labeling is systematized as single mode and tracking mode. In single mode, the observant area is the entire image. Properties equivalent to way labeling is viewed for by fuzzy considering. After the characteristics found, the gradient and the breadth of the road and the road labeling would be computed with active standardization. Then the network will initiate to look again to form the outcome more accurate. Because the slant of camera and the thickness of the road need knowledge of two ways, the frontline side of the way is deployed for calibration. When the detection of single mode is completed, the tracking mode is requested. Few path labeling are dotted borders and few may be blocked. Hence, the properties created from path labeling cannot denotes the road in the entire region. Lane labeling normally could not be changed a lot in the successive figures, so the explored zone of path labeling is limited in the region near to the found marks in the latest face in the tracking mode. Thus the two markings of the anterior path near to the camera are both required in the single mode.

Figure 11 **:** capture on regular roads

Figure 12 **:**.Roads with two middle lines

2.7 IMPLEMENTATION

2.7.1 IMPLEMENTATION

Implementation 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_averag e)

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
```

```
#image=cv2.imread('test_image.jpg') #lane_image=np.copy(image)

#canny_image=canny(lane_image)
#cropped_image=region_of_interest(canny_image)
#lines=cv2.HoughLinesP(cropped_image,2,np.pi/180,1
00,np.array([]),minLineLength=40,maxLineGap=5)

#averaged_lines=average_slope_intercept(lane_imag e,lines)

#line_image=display_lines(lane_image,averaged_lines
)

#combo_image=cv2.addWeighted(lane_image,0.8,line
_image,1,1) #cv2.imshow("result",combo_image)
#cv2.waitKey(0) 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/180,10
0,np.array([]),minLineLength=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
```

2.7.2 SAMPLE INPUT AND OUTPUT

INPUT

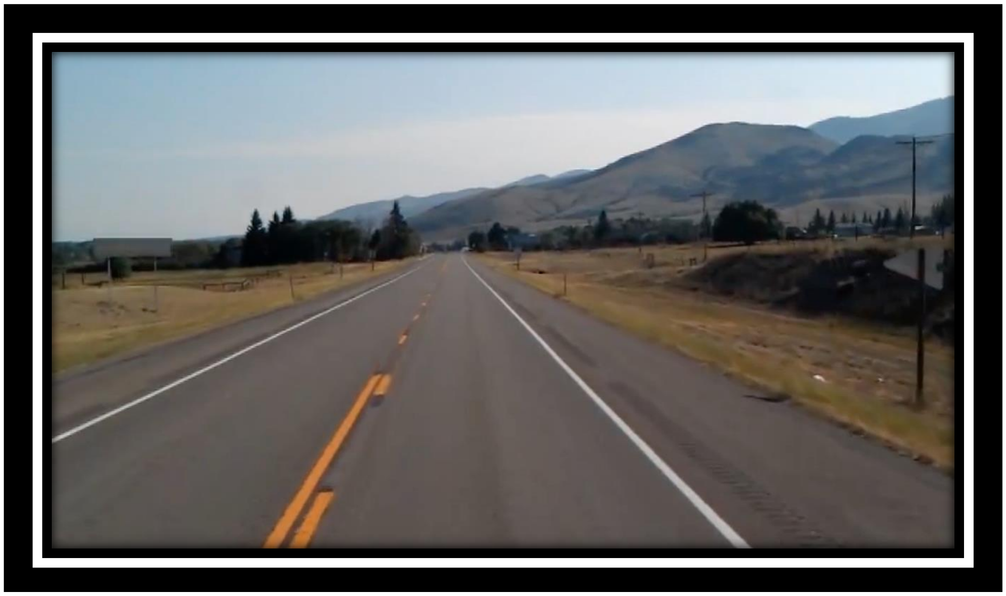


Figure 13 **:** Input

OUTPUT

Figure 14 **:** Output

2.8 HARDWARE AND SOFTWARE REQUIREMENTS

2.8.1 HARDWARE REQUIREMENTS

Processor – DualCore
Speed – 1.1 GHz
RAM – 512 MB (min)
Hard – 20 GB
KeyBoard – Standard Windows Keyboard
Mouse – Two or Three Button Mouse

2.8.2 SOFTWARE REQUIREMENT

Operating System – Windows XP,7,8,10
Technology – Mysql, python, ML
Tools – Pycharm, ML tools

CHAPTER 3

CONCLUSIONS

3.1 RESULT AND DISCUSSIONS

The safety of developed system was guaranteed by ensuring that the host vehicle remain outside the safe time gap during overtaking. The host vehicle, cars was able to execute the overtaking maneuver without exceeding safe limit longitudinal acceleration, heading angle, yaw rate, and velocity, and thus, maintaining the comfort level while driving. The characteristics of fuzzy controllers that were observed during its performance validation stage were quite satisfactory. The fact are based on the results of performance validation System was concluded that the developed fuzzy controllers were suitable for application to steering control even at a higher speed. the system is robust enough.

The developed system is feasible since it uses the sensors and ADAS systems (ACC) which are available in the automotive market. It has shown to a very capable structure that divides tactical planning from decision-making and control. Also, the development of subsystem is performed independently which might beneficial to future optimization of the system under study and in implementation into a real vehicle. The system can be said to be optimized for real-time implementation.

The test protocol for highway overtaking assist was developed which was further used for development of an automated driving systems for autonomous highway overtaking. The developed test protocol was validated analytically using mathematical equations and the automated driving system was tested virtually.

3.2 FUTURE WORKS

Future work of this thesis should focus on the following aspects:

(1) Since overtaking is one of the most complex maneuvers which is classified into four categories (Accelerative, Flying, Piggy backing, and 2+), the scope of this thesis could not cover all the categories because of the time constraints. This work was limited to the test protocols and automated driving systems development for the accelerative or normal overtaking category. The simulations for the flying overtaking were also carried out .

(2) The future extension of this thesis could be test procedure development, verification and validation of control strategy for the remaining two categories, Piggy backing.

(2) Using another control strategy **:** there are two control strategies based on which the overtaking headway threshold and distance vary. The control strategy used in this thesis is ’Constant time gap strategy’ with a headway time of 0.8 s which is more pragmatic for overtaking maneuver. It will be interesting to use the first control strategy viz. ’Constant control spacing strategy’ since the achievable traffic capacity is more in this strategy.

3.3 CONCLUSIONS

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 highway 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 autonomous highway overtaking system has the characteristics like Safe, comfortable, and robust.

The safety of the developed system was guaranteed by ensuring that the host vehicle remains outside the safe time gap during overtaking. Also, the host vehicle was able to execute the overtaking maneuver without exceeding the safe limits of longitudinal acceleration, heading angle, yaw rate, and lateral velocity, it was concluded that the developed fuzzy controller was suitable for application to the steering control even at a higher speed.

Feasible and modular framework of autonomous highway overtaking system . The developed system is feasible since it uses the sensors and ADAS systems. The development of subsystems performed independently which might be beneficial to the future optimization of system under study and implementation into a real vehicle. The simulation for the flying overtaking were also carried out. The future extension of this thesis could be test procedure development, and verification and validation of the control strategy for remaining two categories, Piggy backing.

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 (RAAK SME), 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 Difference?.” 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) B. F. Wu and C. T. Lin, “Real-Time Fuzzy Vehicle Detection Based on Contour Size Similarity,” *Int. J. Fuzzy Systems*, vol. 7, No. 2, pp. 54-62, June 2019.

(8) A. Broggi, M. Bertozzi, Lo Guarino, C. Bianco, and A. Piazzi, “Visual perception of obstacle and vehicles for platooning,” *IEEE Trans. Intelligent Transport. Syst.*, vol. 1, no. 3, pp. 164-176, Sept. 2019.

(9) J. McCall and M. M. Trivedi, ``Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 20_37, Mar. 2015.

(10) B. F. Wu, C. T. Lin, and C. J. Chen, “A Fast Lane and Vehicle Detection Approach for Autonomous Vehicles,” in *Proc. the 7th IASTED International Conference Signal and Image Processing*, Aug., 2018, pp. 305-310.