Dr. AMBEDKAR INSTITUTE OF TECHNOLOGY

Near Jnana Bharathi Campus, Bengaluru - 560056

(An Autonomous Institution, Aided by Government of Karnataka)

Project Synopsis

On

“AUTONOMOUS CAR”

Submitted By

Kumar Vedant

[1DA19CS078]

Under the Guidance Of

Mr. Srinivasa A H
Associate Professor, Dept. of CSE
Dr. AIT

Department of Computer Science & Engineering

2021-22

Dr. Ambedkar Institute of Technology

Department of Computer Science & Engineering

Mini Project Batch-Id:	78
Title of the Mini Project:	Autonomous Car

Details of Project Members

Sl. No	USN	Name	Contact Number	E-mail Id
01	1DA19CS078	Kumar Vedant	9019600754	1da19cs078.cs@drait.edu.in

Details of Guide

Name of the Guide	Designation	Signature of the Guide
Mr. Srinivasa A H	Associate Professor	

CONTENTS

Chapter 1

INTRODUCTION

Self-deriving cars are developed by various companies like Tesla, Audi, and Google etc. The idea of autonomous cars are originated by the arising of accident due to careless people driving and in this way harmful accident extremely occur.

The purpose of our project is to add-on a safety feature on vehicles that is lane detection so that vehicle may not distract from the road when driver is not in his senses. This can make car driving robust and easy for anyone and a lot safer than before. For this purpose we have gone through many articles and read papers on it. Many companies and Developers are interestingly participating their part in this field. What we did is that we have used modern Image processing techniques in combination with Machine Learning models. We are using Open-CV python to overcome our goals.

We started with a simple lane detection module based on Open-CV python to get our edges and lines in the image detected but the most important challenge was to interpret these lines with the image to make sense of lanes. For this we have used slop-intercept methods. Then the race begin for integrating our code to the hardware. And eventually it became unique and innovative by introducing mathematical calculations. For calculations we have used Numpy and Math libraries. Python-Math provides many useful built-in methods for advance calculations like area under the curve and arc-tan curves. And Arc tan proved to be more efficient that slope intercept calculations.

Chapter 2

LITERATURE REVIEW

Self-driving cars are software based cars that runs own their own. Self-driving cars are an advance technological development in the field of automotive industry that provides both comfort and safety features for drivers. Basic module of this vehicle are lane detection and object detection. Many popular self-driving vehicles have features like they take control when they observe miss mindedness sleepiness of drivers. According to a research by ASIRT (Association for Safe international road travel) on average 3,700 people lose their lives every day on the roads. An additional 20-50 million suffer non-fatal injuries, often resulting in long-term disabilities (WHO, 2021). This mostly happens due to human mistakes. To avoid such mistakes self-driving cars come to the ground.

According to Gringer Bonnie's debate on "History of the Autonomous Car" In titlemax.com the first self-driving model that was proposed by General motors in 1939 that was guided by radio-controlled electromagnetic fields generated with magnetized metal spikes embedded in the road. In 1958 GM's design was produced on commercial level and car's front was embedded by coils that were used as sensors. In 1969 John McCarthy put forward his thoughts on self-driving vehicle where he proposed an Idea of lane detection using camera. (Gringer, 2021.)

Prof. B. Wang, V. Frémont and S. A. Rodríguez tells in his paper on 'Color-based road detection and its evaluation' tells that Self-driving cars works on image processing techniques on the basis of feature extraction. Every processing is done on each frame of the video. Machines does not take video as a stream of events it takes it as a stream of frames captured at each minimum instant of time. And all processing is done using those frames. For a camera every image is a 2d mesh of colored pixels. And it's our sense that how we can interpret those colored pixels and turn the frames according to our need. (Wang, 2014)

There are many approaches to implement lane detection. We will discuss all we have seen or learned and then propose ours. First approach that C. Ma and M. Xie, told in their research paper is detecting the lanes by thresholding the given image and calculating left or right on the basis of white pixels. In this method each frame is warped and it is calculated in what direction is the more white pixels. As value of white in colors is 255 and of black is zero. So values off all pixels is calculated column vise and checked for the highest number of value. The side having the highest value means there are more white pixels and vehicle turns that way. This technique is only implementable in a controlled environment where we have a path created with white papers or stuff like that, it cannot work on roads because we have to adjust values of threshold for every environment and it's not possible on commercial production level. There can be improvements in this method by implementing various steps after it or by using this method (Ma, 2010)

Before canny edge detection to improve edges, another approach proposed by Ammar N. Abbas, Muhammad Asad Irshad is by applying machine learning algorithms that detects the feature sets of raw frames to create dataset of image and steering angle and train data using that dataset. On the basis of trained model, the values of steering are predicted by comparing each image with the path. This method can only do prediction for a path that is already followed and trained and cannot run on new areas. We need to put in every single road to train model that we need to use in our daily life because it can only predict the steering angle according to image data. Also it makes it slower as it has to compare matrices of arrays to train a model. And also it takes in a lot of data to train so its loss of storage also. So this method is totally not applicable according to our perspective. (Abbas, 2021)

Some developers are using the method of detecting lanes by using Hough transformation where raw image is converted into edges using cv2's built-in method of canny and lines are created using collinear points collected by the method of HoughlinesP and after that they are drawn over the picture. If we want to conclude the knowledge gained through this literature study we can make a model plan we will use the first method to convert the image to threshold so we can get a clear edge planning so edge detection can be easier by using canny edge detection. One unique thing we have added is cropping the color mapped parts of the image then thresholding. Detailed method will be discussed below. After thresholding canny edge detection will be applied and to get rid of non-necessary tiny edges we have applied Gaussian blur. At the end Hough transformation is applied Hough transform will generate lines information of collinear and adjacent pixels that will be used to draw lines. (Deng, 2018)

Now let us discuss how the processing works in a self-driving car. It takes raw frame captured at each instant and converts it into threshold of black and white color then it detects the edges. Edges are observed under Hough-lines method where collinear points are collected in to array of arrays and each array is then used as drawing points to draw straight lines. Orientations and length of line to be detected can be set in Hough-Lining. Then the angle of drawn lines is calculated and left or right decision is taken accordingly.

Not every method calculates the angle, some follows the color approach and some follows similar pixels approach. But we will be using angle approach.

Chapter 3

OBJECTIVES

- Use OpenCV to identify lane lines for a self-driving car.
- Train a Perceptron-based Neural Network to classify between binary classes.
- Train Convulational Neural Networks to identify between various traffic signs.
- Train Deep Neural Networks to fit complex datasets.
- Build and train a fully functional self driving car to drive on its own.

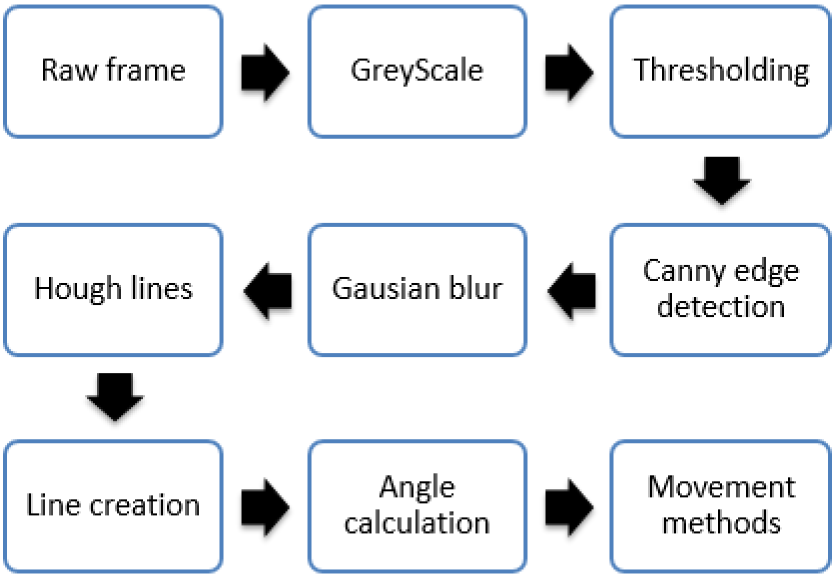
Chapter 4

METHODOLOGY

System Architectue

Self-driving cars works on image processing techniques on the basis of feature extraction. Every processing is done on each frame of the video. Machines does not take video as a stream of events it takes it as a stream of frames captured at each minimum instant of time. And all processing is done using those frames. For a camera every image is a 2d mesh of colored pixels. It takes raw frame captured at each instant and converts it into threshold of black and white color then it detects the edges. Edges are observed under Hough-lines method where collinear points are collected in to array of arrays and each array is then used as drawing points to draw straight lines. Orientations and length of line to be detected can be set in Hough-Lining. Then the angle of drawn lines is calculated and left or right decision is taken accordingly.

Figure 1: General Block Diagram for System Architecture



Chapter 5

REQUIREMENTS SPECIFICATION

- Hardware Specification:–
    - Laptop Model – DELL Inspiron 11 3148
    - Processor – Intel Core i3-4030U CPU
    - RAM – 4GB DDR3L 1600 MHz

- Software Requirements:–
    - Anaconda Distribution
    - Jupyter Notebooks
    - Text Editor (IDE) – Visual Studio Code
    - Programming Language – Python 3.9.0 and above
    - Pyrhon Libraries
        - NumPy
        - OpenCV
        - Keras

Chapter 6

APPLICATIONS

- The main aim behind drivers for achieving autonomous driving is the reduction of traffic accidents by eliminating human error, increasing road capacity and traffic flow by reducing distance between cars and making use of traffic management information, relieving the car occupants from driving and navigation activities and allowing them to engage in other activities or rest.

- A driverless car requires the combination of several techniques among which GNSS. These techniques will enable to guide autonomously a land vehicle from one point to another using public roads. In autonomous driving, GNSS can be used for navigation by determining the vehicle location and speed. With this information the vehicle route can be decided using digital maps. Lane and attitude determination could also benefit from GNSS if the accuracy is good enough. If the location information is shared among cars, GNSS could be a part of a short-range situation awareness system (awareness of other vehicles in the road and collision avoidance) although it is not expected that GNSS is the sole means of information for short-range situation awareness.

- Examples of autonomous vehicle projects are:

    - **Google driverless car** - Project by Google that involves developing technology for driverless car.

    - **EUREKA Prometheus Project** - The programme for a European Traffic of Highest Efficiency and Unprecedented Safety was the largest R&D project ever in the field of driverless cars and was funded by the European Commission.

    - **BRAIVE** - Prototype autonomous vehicle developed by VisLab of the University of Parma.