ÇANKAYA UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

Test Plan, Test Design Specifications and Test Cases and Test Results

Version 1.0.2

CENG 408

Innovative System Design and Development II

AutoCar

Büşra Nur BAHADIR
201511006
Zehra KASAP
201611678

Advisor: *Assist. Prof. Dr. Roya CHOUPANI*

Table of Contents

Abstract

Autonomous vehicles are the indispensable technology of the future, which arises from the combination of artificial intelligence and automotive. Along with many initiatives around the world, big automotive manufacturers are now carrying out serious work on autonomous vehicles. Autonomous vehicles produced by these companies have many features such as being able to go on a certain lane, self-parking, detecting objects and avoiding obstacles. However, these vehicles do not yet can change lanes in order to detect and give way to vehicles with higher priority such as ambulance and fire truck.

Our project aims to detect vehicles with higher priority in traffic by various sensors and cameras mounted on the autonomous vehicles and to recognize the emergency vehicle and clear their way, consequently, to minimize the losses caused by traffic density in emergency cases.

Key words:

Autonomous Vehicle, Artificial Intelligence, Transition Superiority

Özet

Otonom araçlar, yapay zekâ ve otomotivin bir araya gelmesiyle ortaya çıkan geleceğin vazgeçilmez teknolojisidir. Dünya çapında birçok girişimin yanı sıra otomotiv devleri de artık otonom araçlar konusunda oldukça ciddi çalışmalar yürütmekteler. Bu firmaların ürettiği otonom araçlar belirli bir şeritte gidebilme, kendi kendine park edebilme, objeleri tespit etme ve engellerden kaçınma gibi pek çok özelliklere sahipler. Ancak bu araçlar henüz ambulans, itfaiye gibi geçiş üstünlüğüne sahip araçların tespiti ve onlara yol vermek adına şerit değiştirme özelliğine sahip değiller.

Projemiz trafikte geçiş üstünlüğüne sahip araçların, otonom araçlar tarafından çeşitli sensörler ve kamera ile tespit edilip geçiş önceliğinin tanınmasını ve sonuç olarak acil vakalarda trafik yoğunluğu yüzünden meydana gelen kayıpların en aza indirgenmesini amaçlamaktadır.

Anahtar Kelimeler:

Otonom Araç, Yapay Zekâ, Geçiş Üstünlüğü

1. Introduction

1.1 Problem Statement

Emergency vehicle priority is an important issue in the traffic. If we clear the way for emergency vehicles as quickly as possible, we can increase the survival rate of the patients or people on the emergency situations. Our project aiming to add vehicle priority awareness feature to autonomous cars. Of course, autonomous cars not common for today but in the future, everyone expected to use them.

1.2 Background or Related Work

There are no official researches by big companies on the same problem, but some engineers wrote blogs and articles about it. They offered wireless communication system between cars to be able to solve the problem.

1.3 Solution Statement

To solve this problem, we will use cameras, emitter, and receiver sensors. With receiver sensor the vehicle will recognize emergency alert . And clear the road.

1.4 Motivation

In this project, we chose today's one of the most popular topic: Autonomous Vehicle. This project, which many engineers are curious about, was very suitable both for developing and keeping up with today's technology. We examined the studies on this subject. To keep up with this technology, we followed current developments and improved our knowledge on this field. Since we wanted to obtain more accurate results, we decided to use simulation environments rather than building the system with limited hardware.

2. Literature Search

2.1 Abstract

In recent years, autonomous vehicles which can travel without human intervention are the last point reached in the light of intelligent systems and can perform almost all tasks successfully in controlled environments. Road and traffic situation in autonomous vehicles are modelled with the help of communication systems and sensors, then the motion of the vehicle is designed by implementing various techniques and algorithms(1).

Autonomous vehicles are equipped with cameras, sensors and communication systems that enable them to produce enormous amounts of data. In this way, a lot of data will be produced to help the vehicle see, hear, and think while driving. According to Gartner, by 2020, 250 million cars will connect with each other and the surrounding infrastructure through various communication systems.

On this project we aimed to add emergency vehicle priority awareness feature to autonomous cars. When an emergency vehicle approaches, all other traffic must stop or move to the right side to allow the emergency vehicles pass through. But today's autonomous cars do not have features to recognize emergency vehicles. To solve this problem, we will use various sensors and cameras. With receiver sensor the vehicle will recognize emergency alert . And clear the road.

2.2 Introduction

The most remarkable vehicle technology in today is undoubtedly autonomous vehicles. This technology, which will direct the future, is actively developing and being used in the world. In this part of our report we will include brief information about autonomous vehicle technology.

Autonomous vehicles are automobiles that can move without any intervention by detecting the road, traffic flow and surrounding objects without the need of a driver with the help of the control system they have. These vehicles can detect objects around them by using technologies and techniques such as RADAR, LIDAR, GPS, Odometry, Computer Vision.

The autopilot drive of autonomous vehicles starts briefly with the ultrasonic sensors on its wheels, detecting the positions of vehicles that are braking or parked, and data from a wide range of sensors are analysed with a central computer system and events such as steering control, braking and acceleration are performed. This event can only be described as a beginning. As the computer technologies are more accessible and cheaper the future of driverless cars becomes more achievable. Although autonomous driving technology is still in its infancy, it is becoming more common and can fundamentally change our transportation system (and thus expand our economy and society).

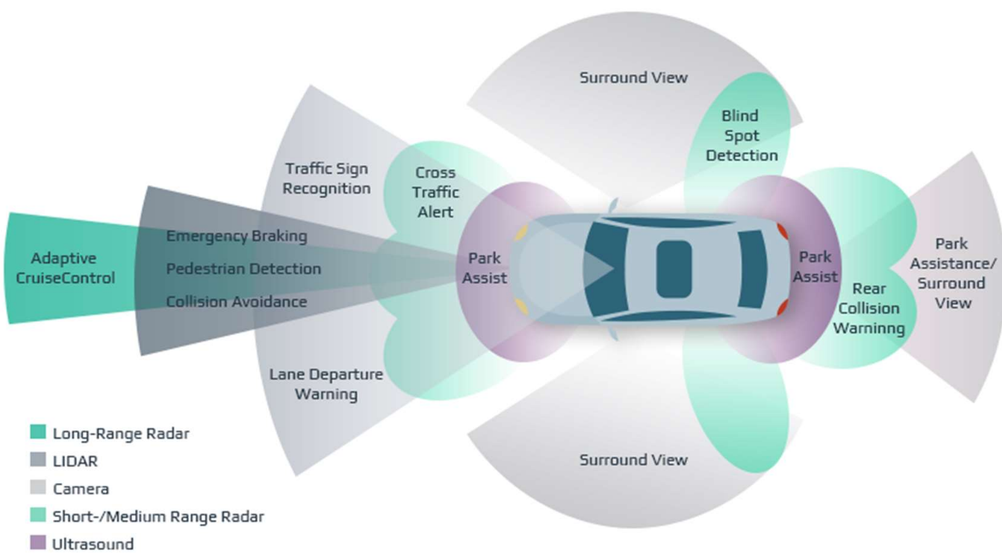


Figure 1 Shows Sensor Technologies on the Autonomous Car.

The Self Driving Cars have 5 different levels of driving automation.

Level 0- At the level zero, there is no automation. All tasks have been done by the driver.

Level 1- At the level One, the vehicle has Driver Assistance but there is still need to a driver to do the primary tasks like accelerating, brake and monitoring. The vehicle can assist with simple functions for example applying brake a little bit more when you get too close to another car on the road.

Level 2- The Level Two, also known as Partial Automation, steering, acceleration functions assisted by the vehicle. The drivers disengaged from some of their tasks but must be ready to take control of the vehicle in the emergency situations.

Level 3- At the Level Three (Conditional Automation), most important improvement is the vehicle started to use sensors (LIDAR, Radar etc.) to control all monitoring of the its environment. A Driver is still needed but on safety critical functions the vehicle can respond itself.

Level 4- At the Level Four (High Automation), besides to other levels features vehicle can determine when to change lanes, turn and use signals. But it cannot determine traffic congestion or merge onto the highway.

Level 5- The Level Five is Complete Automation. There is no need to a human driver to control any task of the vehicle. So, there is not a steering wheel, pedals, or brakes. All tasks are handled by the vehicle's itself.

With the increase in competition among firms in the automotive sector and the increase in the number of traffic accidents caused by driver error, investments in this field has increased and artificial intelligence has entered this field. Samples of autonomous vehicles are slowly released into traffic, and sales are scheduled to begin in 2020.

DRIVE AGX Pegasus - NVidia announced DRIVE AGX Pegasus which is an energy-efficient, high-performance AI computer for Level Five automation driverless car. Being an expert in software makes the company stand out.

Waymo - Waymo, formerly known as Google Car, which is a small, cute prototype autonomous vehicle. The exact date for release and its cost is still unknown. The clearest information about the vehicle is that it is completely autonomous. It is said to be integrated with the Google Maps application that works better than most navigation systems.

Tesla - Tesla is the first brand that comes to mind when it comes to electric cars, and it is now considered the best brand in autonomous driving technology.

2.3 History of Self Driving Cars

1925, Francis Houdina demonstrated the radio-controlled "American Wonder" for the first time. The American Wonder was a 1926 Chandler. It had a launcher on the tonneau and operated by a second car that follows it and emits a radio pulse captured by the transmitting antenna. The antenna was introducing the signal into the circuit breaker, which operated a small electric motor to guide each movement of the car.

1939, General Motors introduced the idea of autonomous vehicle design for the first time at the "FUTURAMA" exhibition at the New York World's Fair.

1953, RCA Labs successfully manufactured a miniature car that was guided and controlled by wires that were laid on the laboratory floor (2).

1956, The General Mobile Firebird II is equipped with receivers for detector circuits embedded in roadways.

1979, Stanford Cart was able to move indoors without human intervention with using the image processing algorithm called as The Cart's Vision Algorithm. This algorithm was inspired by the Blocks World planning method and consisted of reducing the image to the edges but proved unsuitable for outdoor use with many complex shapes and colours (3).

2.3.1 The First Attempts to Automate Vehicle Control

In the 1984, the first truly autonomous and self-sufficient vehicle created with the help of Carnegie Mellon University's Navlab and ALV (Autonomous Land Vehicle) projects. The vehicle could speed on road with 31 kilometres per hour. In the 1986 they added obstacle avoidance mechanism and in the 1987 the vehicle could work in the day and night conditions.

2.3.2 The First Self Driving Car

On March 13, 2004 in the Mojave Desert region of the United States, DARPA (Defence Advanced Research Projects Agency) organized a prize competition for American autonomous vehicles which named as DARPA Grand Challenge to accelerate the development of autonomous driving technologies that can be applied to military needs. Any vehicles could complete the difficult desert route. So, $1 million prize went unclaimed.

The second Grand Challenge with $2 million prize was held in the desert Southwest near the California, Nevada state line, on October 8,2015. At the total five vehicles successfully completed the challenge.

2.3.3 DARPA Grand Challenge

On March 13, 2004 in the Mojave Desert region of the United States, DARPA (Defence Advanced Research Projects Agency) organized a prize competition for American autonomous vehicles which named as DARPA Grand Challenge to accelerate the development of autonomous driving technologies that can be applied to military needs. Any vehicles could complete the difficult desert route. So, $1 million prize went unclaimed.

The second Grand Challenge with $2 million prize was held in the desert Southwest near the California, Nevada state line, on October 8,2015. At the total five vehicles successfully completed the challenge.

2.3.4 DARPA Urban Grand Challenge

On November 3, 2007 at the site of the Southern California Logistics Airport, in California the third autonomous car competition of the DARPA Grand Challenge has been organized with $2 million. The winner was Tartan Racing, a collaborative effort by Carnegie Mellon University and General Motors Corporation, with their vehicle "Boss", a heavily modified Chevrolet Tahoe (4). The Boss used perceptual, planning, and behavioural software to understand traffic conditions and decide on the way to the destination. The vehicle was equipped with various lasers, cameras and radars that allowed the planning of the route considering static and dynamic obstacles and moving objects. It is collected with algorithms used to find and recognize environmental information, lanes, parking limits, waypoints and more. In addition, algorithms have been applied to determine the dangerous behaviours of other drives. The most important features of the technology developed by the Tartan Racing team are:

- Driving in accordance with the road traffic rules (considering the priority rules of crossing intersections)
- Detection and tracking other vehicles over long distances
- Finding parking spaces and parking itself
- Maintaining safe distance to followed vehicles
- Reacting to non-standard events (e.g. traffic jams) (5).

2.3.5 The First Self Driving Car Approved for Road Traffic

The Tesla Model S is an all-electric five-door lift back sedan, produced by Tesla, Inc., and introduced on June 22, 2012. As of April 23, 2019, the Model S Long Range has an EPA range of 370 miles (600 km), higher than any other battery electric car (6). It also has Autonomous Driving option with software called Autopilot Firmware which offered the second level of automation. Also, it has features such as Enhanced Summon which is allows the car drive through a parking lot to find you without any help from a driver, and Sentry Mode to sense and record any suspicious activity around the car.

This is a summary of the history of autonomous vehicles from past to present. Different companies are joining the sector day by day and some software is aimed to be open source and contribute to the sector.

2.4 Solutions and Algorithms

2.4.1 Camera Image for Environment Recognition

Cameras are eyes of the autonomous vehicles, used for recognition of traffic lanes, identification of objects and image-based environment mapping. In addition to cameras, radar sensors (insensitive to weather, dust and pollution) and more expensive laser sensors - lidars are used (7) .

2.4.1.1 Traffic Lane Detection

The lines on the road indicate to the driver where the lane is and as a criterion for determining which direction the vehicle should travel and how the vehicle agent interacts harmoniously on the road. Identifying lanes on the road includes finding lines that are connected at one point.

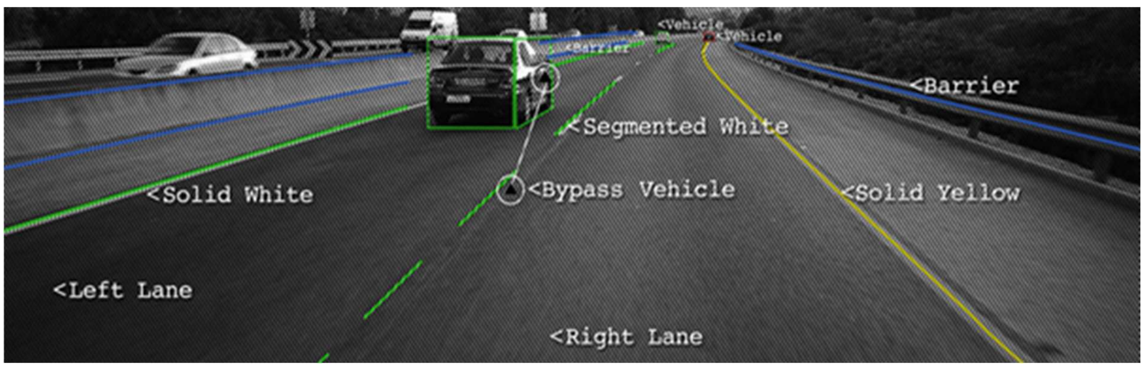


Figure 2 Shows line recognition of Self Driving Car.

The VP (Vanishing Point) method for finding road in a desert, proposed by C. Rasmussen (8), consists in calculating the dominant orientations in image segments (image resolution 640 x 480 pixels divided into 72 segments), estimating the position of the point of convergence and tracking that point in subsequent image frames. The Rasmussen method, developed to recognize the road in a desert area, today plays an important role in detecting road lanes in an image (9) and forms the basis for more advanced algorithms (10).

Among of many approaches for lane tracking Hough Transform and Spatial CNN are most popular ones.

Hough transform
Is a technique which can be used to isolate features of a shape within an image. Because it requires that the desired features be specified in some parametric form, the classical Hough transform is most commonly used for the detection of regular curves such as lines, circles, ellipses, etc. (11)

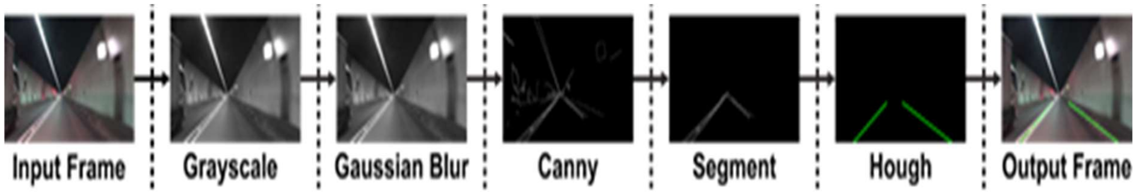


Figure 3 Shows Hough Transform on a given frame.

CNN and Spatial CNN

Although convolutional neural networks (CNNs) have proven to be effective methods for identifying lower image layers (e.g., edges, colour gradients) and deeper complex features and entities (e.g., object recognition), they are often having difficulties representing the "poses" of these features and entities (CNN is well suited for extracting semantics from raw pixels, but does not perform well when capturing spatial relationships such as rotation and translation relationships of pixels in a frame.). In a traditional layer-by-layer CNN, each convolutional layer receives input from its previous layer, performs convolution and nonlinear activation, and then sends the output to the next layer. Spatial convolutional neural networks (SCNN) takes this one step further by treating the rows and columns of each feature map as "layers", applying the same process in turn, allowing pixel information to pass messages between neurons in the same layer.

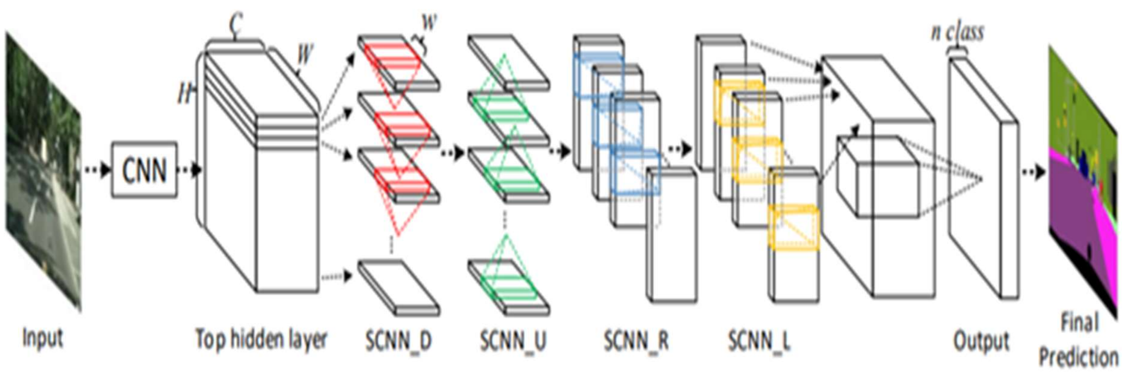


Figure 4 Shows how Spatial CNN works for Traffic Scene Understanding.

SCNN is relatively new, published on 2018, but have already outperformed the likes of ReNet (RNN), MRFNet (MRF+CNN), much deeper ResNet architectures, and placed first on the TuSimple Benchmark Lane Detection Challenge with 96.53% accuracy (12).

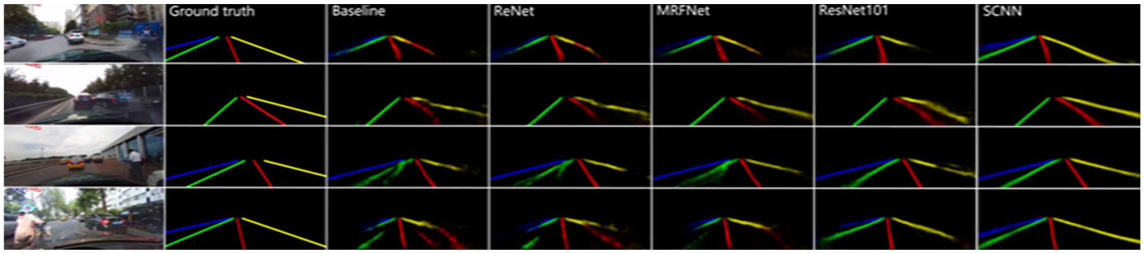


Figure 5 Shows accuracy rates among architectures.

2.4.1.2 Semantic Image Segmentation and Object Recognition

Segmentation is critical for image analysis tasks. Semantic segmentation describes the process of associating each pixel of an image with a category label, detects objects of a specific category, including people, road signs, and cars in complex images.

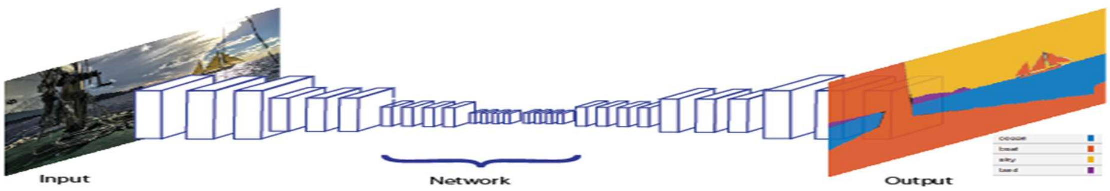


Figure 6 Semantic Segmentation

R-CNN
R-CNN is one of the Deep learning object detection algorithms based on CNN. Based on this, there is Faster R-CNN for faster object detection, and Mask R-CNN for faster object sample segmentation (image segmentation).

SSD (Single Shot MultiBox Detector)
SSD is a popular algorithm for object detection. Faster than R-CNN.

YOLO (You Only Look Once)
YOLO is an object detection algorithm that has recently become very popular. It is popular because it is much faster than other detection algorithms. You can use this algorithm to train any object you want to detect.

TensorFlow Object Detection API
TensorFlow simplifies object detection using a pre-trained object detection model. Objects can be easily detected with pre-trained models.

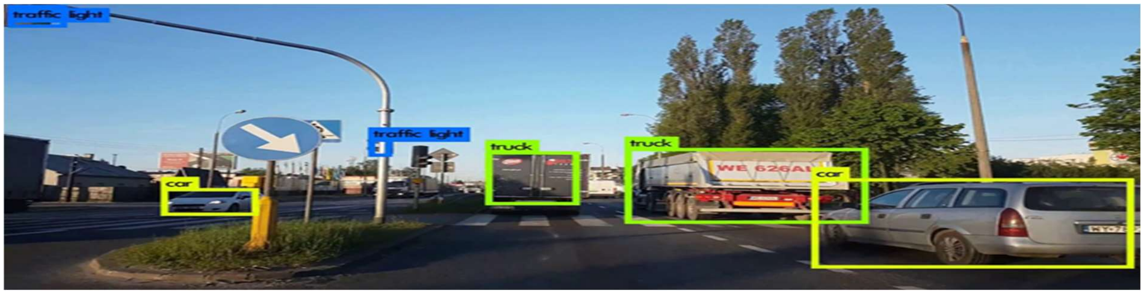


Figure 7 Object Detection on Self Driving Car

2.4.1.3 Visual SLAM (Image based environment mapping)

Simultaneous Localization and Mapping (SLAM) can accurately determine the location of the user relative to the environment. Visual SLAM is a specific type of SLAM system that the trajectory of the platform and the location of all landmarks can be estimated online without any prior knowledge of the location. In the case of an algorithm that uses an image that must identify and track landmarks in each frame of the camera image, the requirements for the device (computing power) used are very strict.
The framework is mainly composed of three modules as follows:

1. Initialization
2. Tracking
3. Mapping

According to the purposes of applications the following two additional modules are also included in Visual SLAM algorithms:

1. Re-localization
2. Global map optimization

Some vSLAM Algorithms:

- EKF-SLAM (EKF - Extended Kalman Filter)
- FastSLAM algorithm which uses RBPFs (Rao-Blackwellised Particle Filters)
- RGB-D SLAM which utilizes image and image depth (Kinect) (13).
- ORB-SLAM and ORB-SLAM2 for single images, stereo images (stereoscopic vision) and RGB-D cameras (14) . The algorithm utilizes the ORB (Oriented FAST and Rotated BRIEF) descriptor (15).
- LSD-SLAM an algorithm that generates depth maps from individual image frames without using landmarks (16).
- L-SLAM – an algorithm that reduces the dimensionality of FastSLAM algorithms (17).

2.4.2 Using Sensors for Environment Recognition

The autonomous vehicles use three main type of sensors which are cameras, radar and lidar. Using all these sensors together provides the car a visual effect of the surrounding environment and help it detect the speed and distance of nearby objects and their three-dimensional shape. Furthermore, sensors are called as inertial measurement unit helps to track the acceleration and position of the vehicle.

Radar (Radio Detection and Ranging) sensors has long been used in the automotive industry to determine the speed, range, and direction of movement of objects. Radar sensors complement camera vision during low visibility such as night driving and improve the detection capabilities of autonomous vehicles.

By emitting an invisible laser at an alarming rate, the LIDAR (Light Detection and Ranging) sensors can draw detailed 3D images from the instantaneously bounced signal. These signals form a "point cloud" that represents the environment around the vehicle to enhance the safety and diversity of sensor data. The LIDAR provides driverless cars a 360-degree view 3D images of their surroundings, shape and depth for the surrounding cars and pedestrians and finally the geographic area of the road. Moreover, like radar, it works well in low light conditions.

Figure 8 Point Cloud with LIDAR

Therefore, a less advanced and relatively inexpensive set of devices such as ultrasonic sensor mounted on, for example, a car bumper (parking sensor) that measures the distance to an obstacle.

2.4.2.1 Sensor Data Based on Environment Mapping (SLAM)

SLAM always uses several different sensor types, and the functionality and limitations of the various sensor types have been the main drivers of the new algorithm.

The SLAM algorithm can be classified by sensor type:

Radar:

- GraphSLAM offline algorithm (18), based on constructing graphs and mapping environmental grid (19).
- Graph-Based SLAM based on constructing graphs and finding node configurations of minimal error.
- Real-Time Radar SLAM, which utilizes FastSLAM and GraphSLAM, and which is performed within 45 milliseconds (IntelCore i7-3830K) (20).

LIDAR:

- ML-SLAM (ML – Maximum Likelihood) based on maximum likelihood estimation (21).
- Credibilist SLAM (or C-SLAM) (22) based on TBM (Transferable Belief Model) (23).
- ICP-SLAM (24) which utilizes ICP (Iteractive Closest Point) method of registering three-dimensional shapes (25).
- Google's Cartographer SLAM, which utilizes, in addition to a lidar, IMU (Inertial Measurement Unit) and images from cameras (26), (27).

The LOAM algorithm (Lidar Odometry And Mapping), which consists in using distance measurements made by a biaxial lidar moving in six degrees of freedom, is also used for environment mapping (28).

2.4.3 Sensor Data Fusion

Sensor data fusion helps combining data which delivered from multiple sensor sources and gives better information accuracy than using these sources individually.

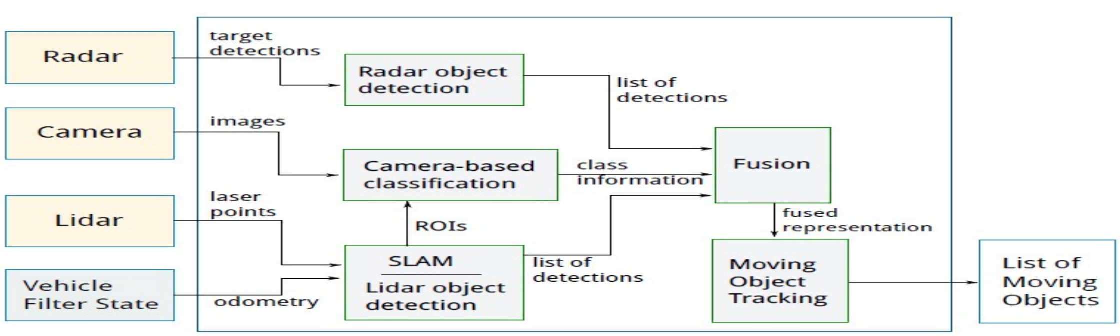


Figure 9 Multiple Sensor Perception System

For merging multiple sensor data Kalman Filter algorithm (One of the most popular fusion algorithm) has been used since the date when it invented in 1960 by Rudolph Kalman. This algorithm also known as linear quadratic estimation (LQE), uses measurement series of a dynamic system which are observed over time, handles noise and other inaccuracies with producing unknown variables to give more accurate results by estimating a joint probability distribution over the variables for each timeframe. For example, multiple sensors can give different distance value of an object. Kalman Filter allows us to estimate real distance value with better accuracy.

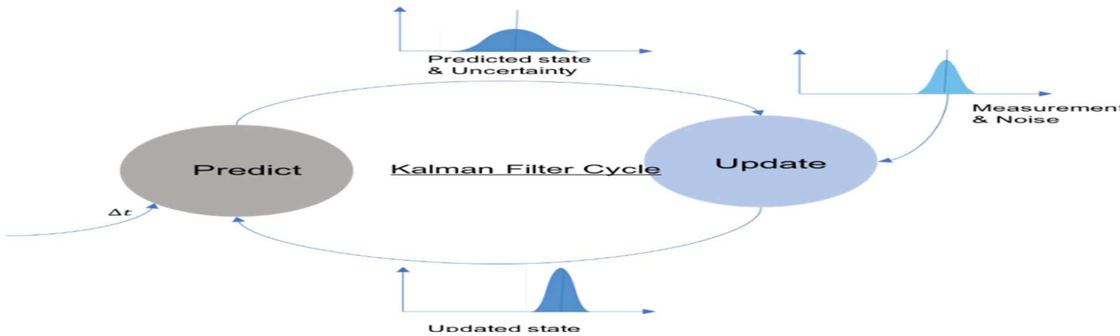


Figure 10 Kalman Filter Circle

2.4.3.1 Detection, Classification and Tracking of Moving Objects in a 3D Environment (DATMO)

Getting reliable perception of the surrounding environment is most important step of building an intelligent vehicle. This step is usually divided into two sub-tasks: simultaneous localization and mapping (SLAM) and transport detection and monitoring objects (DATMO). The purpose of SLAM is to provide the vehicle with a map consisting of static entities of the environment while DATMO uses that map to detect and track dynamic entities (29).

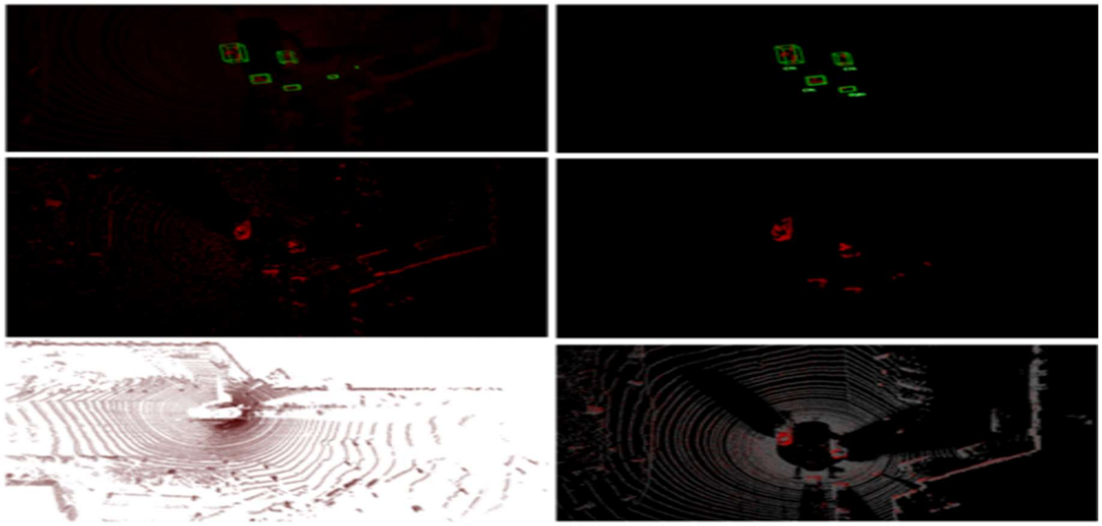


Figure 11 Experimental results showing the detection, classification and tracking of moving objects.

2.4.4 Decision Making and Planning

At the beginning self-driving vehicles were mostly semi-autonomous because their designed functions were typically limited to lane tracking, adaptive cruise control and a few other basic functions. A wider range of features were demonstrated in the 2007 DARPA Urban Challenge (DUC) (30). Although the performance of self-driving vehicles was still far below the level of human drivers, this event shows the applicability of autonomous driving in urban environments and presents significant research challenges for autonomous driving.

2.4.4.1 DDM (Driving Decision Making)

The Driving Decision Mechanism (DDM) is identified as the key technology to ensure safe driving of autonomous vehicles, which is primarily affected by vehicle status and road conditions. With the sensors, the autonomous vehicle can detect and collect traffic information, including vehicle conditions and road conditions in real time, to enter the data processing program designed for some data processing to obtain input variables of the DDM. Based on these input variables, DDM searches for relevant information and matches the exact driving decisions with the learning experience, and then transmits the decision order to the control system. These learning experiences refer to driving decision-making rules in DDM which are obtained by learning a large amount of actual driving experience. The control system then continues to operate the control actuators (including the steering system, pedals and automatic shifting).

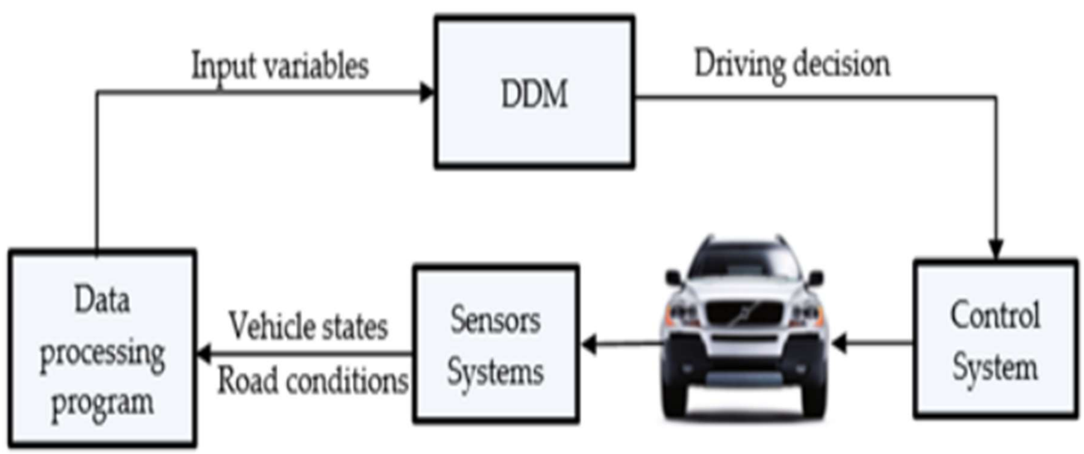


Figure 12 Schematic architecture of the driving decision-making process of autonomous vehicle. DDM: Driving Decision-making Mechanism.

methods for decision making:

- Decision trees (31), (32) and diagrams (33).
- Partially Observable Markov Decision Processes (POMPD) (34).
- Machine learning:

- Support Vector Machine Regression (SVR) with Particle Swarm Optimization (PSO) (35).
- Markov Decision Processes (MPD) with Reinforcement Learning (RL) (36).
- Deep Reinforcement Learning (DRL) (37).

2.4.4.2 Mission Planning (Route planning)

The route that the autonomous vehicle takes is called as the route plans. To find your own vehicle and determine the route on the map, you first need to apply GPS sensors and geographic location data. Additionally, there are solutions that use maps based on image semantics and 3D lidar data. Search algorithms are made with Lattice Planning and Reinforcement Learning.

A * (Star Search Algorithm) -one of the algorithms used to find the shortest path.

Lattice Planning done by repeating the original path of the possible connection. Suitable for restricted environments, it can cause difficulties in evasive maneuvers.

Reinforcement learning -is a behavioural-inspired machine learning approach that deals with the actions an object must take to obtain the highest rewards in the environment.

The behaviour planner is responsible for making decisions to ensure that the vehicle follows traffic rules and interacts with other agents in a routine, safe manner, while gradually evolving along the path of the mission planner.

2.4.4.3 Control of the Vehicle Movement

Control section sets the steering, speed, and braking state of the vehicle.

Classical Control Methods:

Nonlinear control, including tracking the trajectory of vehicle motion and maintaining the trajectory by manipulating the vehicle along a specified route.

Generally used controller structure in many applications is **Feedback-feedforward control**. This controller structure can reduce the negative effects of parameter changes, modelling errors and unnecessary interferences. Additionally, it can change the transient behaviour of the system and the effects of measurement noise.

The most widely used feedback control method is PID (Proportional Integral Derivative) control. The steering control is based on the information of the lane tracking system. Driving results can be optimized by transferring results from the lane tracking system to other neural networks.

We can also use the model in the control application. The control method using system modelling to optimize the forward time interval is called as **Model Predictive Control (MPC)** in the literature.

Parallel autonomous systems are providing additional security by taking on driver duties in dangerous situations. The parallel autonomous system can also take over the operation of the vehicle at the request of the driver.

2.4.4.4 Artificial Intelligence for Prediction

In the Prediction part, the vehicles predict the behaviour of the vehicle or human being in its environment. It predicts the direction and speed of the vehicle. In this way, an autonomous vehicle can react to different events in advance. For this purpose, Recurrent Neural Network used.

Recurrent Neural Networks - Convolutional Neural Networks (CNN) process information in specific image frames regardless of the information they have learned from previous frames. In addition, the RNN architecture supports memory so that it can benefit from past determinations when calculating future forecasts. Therefore, RNN provides a natural way to predict the next step.

2.5 Conclusion

Self-Driving Cars are one of the most important and interesting technology of the future. These cars using various kinds of sensors, analytical algorithms, and AI to perform specific tasks (object recognition, mission-planning, decision-making, controlling) and perceives their environment.
With today's technology fully autonomous cars could not invented yet and big companies spending so many investments for be able to constantly develop it. And Autonomous cars still have problems like ethical issues (Trolley dilemma, is killing one person to save five people is wrong or not.) and what will be the rules of the driver's license test for such a driver for future.
This literature review was aiming look deeper into the Self Driving Car algorithms and systems, so it does not provide a list of best algorithms for Self-Driving Cars. Because performance of the algorithms depends on:

1. The computing power of the device that affects the speed at which the algorithm executes.

2. The type and quality of sensor depends on the size and purpose of the vehicle.

We aimed to add emergency vehicle priority awareness feature to autonomous cars. In our project, we plan to use Image Processing, PID algorithms and test the results in simulation environment. The Autonomous Vehicle Drive Simulator that we will use need to provide us to simulate sensors such as Lidar, GPS, radar and gives potential sensor outputs, with these outputs and by trying out possible traffic scenarios we will improve the software that we will make.

According to these outputs from sensors and cameras, our primary targets are; Lane tracking system with Hough Transform, identifying objects and traffic signs with Object Identification methods, and the approaching emergency vehicle will be detected with emergency alert from emitter and camera data, it will determine the direction and the geographical position of the vehicle.

2.6 Simulations for Autonomous Car

Using self-driving car simulators are good for saving money and time. And providing more accuracy with being able to test many traffic scenarios in it. Nvidia Drive Sim is most popular one among the driving simulator but it’s not open source and requires hardware like Drive AGX. Carla and Apollo Simulators are open source and provides many sensors, weather conditions etc. yet they do not have environment voice which we need to use for simulating emergency car sirens. We contacted with companies of other simulators as Cognata and Metamoto to requesting demo versions, but demo versions were not enough for demonstrating our project. At the end we decided to use Webots which is a simulator for scientific use and robots. They recently provide libraries for autonomous cars and has environment voice.

3. Summary

3.1 Technology Used

This software will be developed with Webots Simulator and its libraries for autonomous cars in Python. Webots is a professional mobile robot simulation software package. It offers a rapid prototyping environment, that allows the user to create 3D virtual worlds with physics properties such as mass, joints, friction coefficients, etc (38).

We chose Webots because it is cross platform, open source, supports many languages including Python and has environment sound. It is also providing tools as OpenStreeMap to create our own world and SUMO (Simulation of Urban MObility) interface to simulate traffic and sensors as LIDAR, radar, GPS etc. It does not provide Emergency car models so we will try to create them in Blender and export to it. But modelling a car is requires modelling knowledge and skills so If we cannot make it, we will add sirens with speaker to normal car models which are the simulator provides. For voice assistant we will use Google Speech to Text API.

4. Software Requirements Specification

4.1 Introduction

Autonomous vehicles are automobiles that can move without any intervention by detecting the road, traffic flow and surrounding objects with the help of the control system they have. These vehicles can detect objects around them by using technologies and techniques such as RADAR, LIDAR, GPS, Odometry, Computer Vision. The autopilot drive of autonomous vehicles starts briefly with the ultrasonic sensors on its wheels, detecting the positions of vehicles that are braking or parked, and data from a wide range of sensors are analysed with a central computer system and events such as steering control, braking and acceleration are performed. This event can only be described as a beginning. As the computer technologies are more accessible and cheaper the future of driverless cars becomes more achievable. Though still in its infancy, self-driving technology is becoming increasingly common and could radically transform our transportation system (and by extension, our economy and society).

4.1.1 Purpose

We aimed to add emergency vehicle priority awareness feature to autonomous cars. In our project, we plan to use Image Processing , PID algorithms and test the results in simulation environment. The Autonomous Vehicle Drive Simulator that we will use need to provide us to simulate sensors such as LIDAR, GPS, radar and gives potential sensor outputs, with these outputs and by trying out possible traffic scenarios we will improve the software that we will make.

When an emergency vehicle approaches in emergency status (siren), with receiver sensors the vehicle will recognise emergency alert from emergency vehicle's emitter and then will switch to an available line to clear emergency vehicle's way. This feature not only emptying based on one lane rule because emergency vehicle can approach from left lane, try to make an emergency corridor or can use shoulder of the road.

4.1.2 Intended Audience and Reading Suggestions

This Software Requirement Specification Report intended for software developers, software architects, testers, project managers and documentation writers. Before reading this report reading our Literature Review may help to understand working principles of sensors and algorithms. This report includes overall description of the product and requirement specification of the project.

4.1.3 Project Scope

On this project we aimed to add emergency vehicle priority awareness feature to autonomous cars. The autonomous car that we simulate will be able to detect the emergency vehicles, and their location and direction. The autonomous car will use visual information; therefore, it is required to have multiple cameras mounted on the vehicle.

Our system will include:

- Lane detecting and following
- Object recognition and auto brake
- Virtual drive assistant
- Route Planning
- Emergency vehicle priority awareness

Current autonomous cars already have the first four of these features. We will add fifth one as a new feature to autonomous cars.

To be able to success on this project our car at least needs to follow the lane, not hit objects on the road and let the emergency vehicle pass by changing lanes. Other features may change according to the possibilities of the simulator.

4.1.4 Definitions, acronyms, and abbreviations (Glossary / Terminology)

Terminology	**Definition**
User	A person who interacts with the system.
SRS	The report that provides an overview of all system components.

Terminology	Definition
Autonomous Driving	A control mode which a vehicle doesn't need a driver attention.
Emergency Vehicle	An emergency vehicle is any vehicle that is designated and authorized to respond to an emergency in a life-threatening situation (39).
Vehicle Priority	When a vehicle with higher priority approaches, all other traffic must stop or move to the right side to allow the vehicles pass through.
System	System software is a type of computer program that is designed to run a computer's hardware and application programs (40).
Control Unit	The control unit (CU) is a component of a computer's central processing unit (CPU) that directs the operation of the processor (41).

4.1.5 References

- IEEE Std 830™-1998(R2009) Recommended Practice for Software Requirements Specifications.

4.2 Overall Description

This section will explain the aspects of the Self Driving Car system and requirements.

4.2.1 Product Perspective

Self-Driving: Classification systems that monitor traffic signs, use cameras, monitor other systems, use radar and laser sensors.

LIDAR: Optical remote sensing technology to measure distance to target by illuminating with light.

GPS: Space-based satellite navigation system that provides time and location information anywhere.

Digital Maps: The process in which data collection is compiled and formatted in a virtual image.

Adaptive Cruise Control: Tracks distances to adjacent vehicles on the same lane. Detects objects in front of a vehicle at risk of emergency collision.

Lane Assist: Tracks the position of the vehicle in the lane.

Sound and Light Sensors: It is a system that will be used specially to identify cars with priority. It will also be used to detect surrounding objects, lanes, and other tools.

4.2.2 Product Functions

Lane Lines Detection: The system detects highway lane lines. Distinguishes dashed lines and straight lines. Provides warning in case of loss of lanes. Image analysis techniques are used to define lines.

Tracking Environment: Tracks objects around the vehicle using the scanner and algorithms. Sensors monitor the position of objects as they move within the scanning range. So, the system behaves according to objects.

Detection of Traffic Signs: Using image processing techniques and various algorithms, studies are done to classify traffic signs. Recognize traffic signs. The vehicle behaves according to the colours of the traffic lights.

Vehicle Detection and Tracking: The system performs vehicle detection and tracking events. It adjusts the speed and position according to the behaviour of the vehicles around it during highway driving. Sudden braking performs events such as lane change. Keeps track distance always constant with a vehicle in front.

Apply Braking: When a pedestrian step in front of the vehicle and a collision is possible, sudden braking has been performed. When the obstacles, pedestrians, and vehicles on the road are lifted, the vehicle accelerates to regain its speed.

Road Planning: This project aims to is to create a road planner that can create safe trajectories for the vehicle to follow. On the highway track, there are other vehicles, all at different speeds. The car transmits its position along with sensor fusion data that estimates the position of all vehicles on the same side of the road.

Right of priority vehicles: Recognizes vehicles such as fire trucks or ambulances using sound sensors and image processing techniques. The system changes lanes to give way to these vehicles.

4.2.3 User Classes and Characteristics

This section covers all user characteristics and expectations relating to the user of the system. The user in the context of this project would be considered works under certain assumptions. These assumptions include that the user can interact with the system, through driving, and by either the visual light, warning sound. In the event of a conflict with other systems in the vehicle, the user has the control to disable the system.

The vehicles that will be equipped in the system used in our project will be autonomous. Therefore, the user interaction with the system is expected to be minimal. We assume that the user has little or no knowledge of the system.

4.2.4 Operating Environment

The hardware, software and technology used should have following specifications:

- Processor with speed of 2GHz
- Continuous power supply
- Ability to use camera, microphone, and other services of the system
- 1GB memory or more

The software being developed will be tested on Webots which an open source and cross platform simulator is.

4.2.5 Constraints

This section covers the constraints for the system, which also includes descriptions of safety-critical properties for the system.

The system is designed to help the user operate the vehicle, and the system should always be operating correctly. There are some situations where the system requires restraint. In the absence of lane markings (an unmarked lane, dirt road, bad weather, etc.), the system becomes ineffective and must give the user an audible or visual warning. This will continue until road conditions with clearly defined lane markings are met.

If there is a problem with the cameras used to detect these strips, the system will not work again. This could also be a malfunction of cameras and sound sensors that see priority vehicles. In this case, the system switches itself off again and should display a warning to the user again. This problem is valid until the camera or sound sensors are repaired. Sensors will still be used to obtain distance information. The sensor data will be used with the camera for objects around the vehicle. It requires stable and fast internet connection when locating objects around. Otherwise it cannot communicate quickly with other tools.

The system must be configured to respond to commands in 500 msec. If there is an obstacle on the road during driving, the response time and brake release time must always be the same to achieve the desired deceleration. In all scenarios, the vehicle speed must be constant.

4.2.6 Assumptions and Dependencies

Various assumptions and dependencies went into the creation of project in order to ensure that the system works safely and efficiently.

4.2.6.1 Assumptions

- The main assumption in this project is that the system works well when there are no environmental factors. (Bad weather, holes, slope, etc.)
- Lane markings are assumed to be distinct.
- In our system, it is assumed that all traffic signs and the presence of all objects around the vehicle can be clearly and seen.
- Priority vehicles such as ambulances and fire trucks are clearly recognized by their voice and image recognition.
- If the vehicle is running, it is assumed that the system is always on and scanned.
- It is assumed that all system elements are operating properly and there are no abnormal conditions.

4.3 Requirement Specification

This section includes requirements to build our software system and description of its behaviour.

4.3.1 External Interface Requirements

The information provided in this section ensures that the system communicates correctly with external components.

4.3.1.1 User Interfaces

User can start and turn-off the system. And may can see informations about car from dashboard like the speedometer, tachometer, odometer, engine coolant temperature gauge, and fuel gauge, turn indicators, gearshift position indicator, seat belt warning light, parking-brake warning light, and engine-malfunction lights. And can give order like “speed up to 70 km/h”, “change the left lane” with virtual assistance.

4.3.1.2 Hardware Interfaces

Hardware interface may include sensors and necessarily control unit (e.g. DRIVE AGX Kit) but for now we are trying to make and test everything on simulation environment so, there will not be any external hardware used except our computers.

4.3.1.3 Software Interfaces

Since its on simulation environment there are no external software interface requirements. But if we can build a prototype, Software Interfaces will change with respect to the hardware which we will be using.

4.3.1.4 Communication Interfaces

There are no external communications interface requirements.

4.3.2 Functional Requirements

This part of the report includes functional requirements of the project.

4.3.2.1 Use Cases

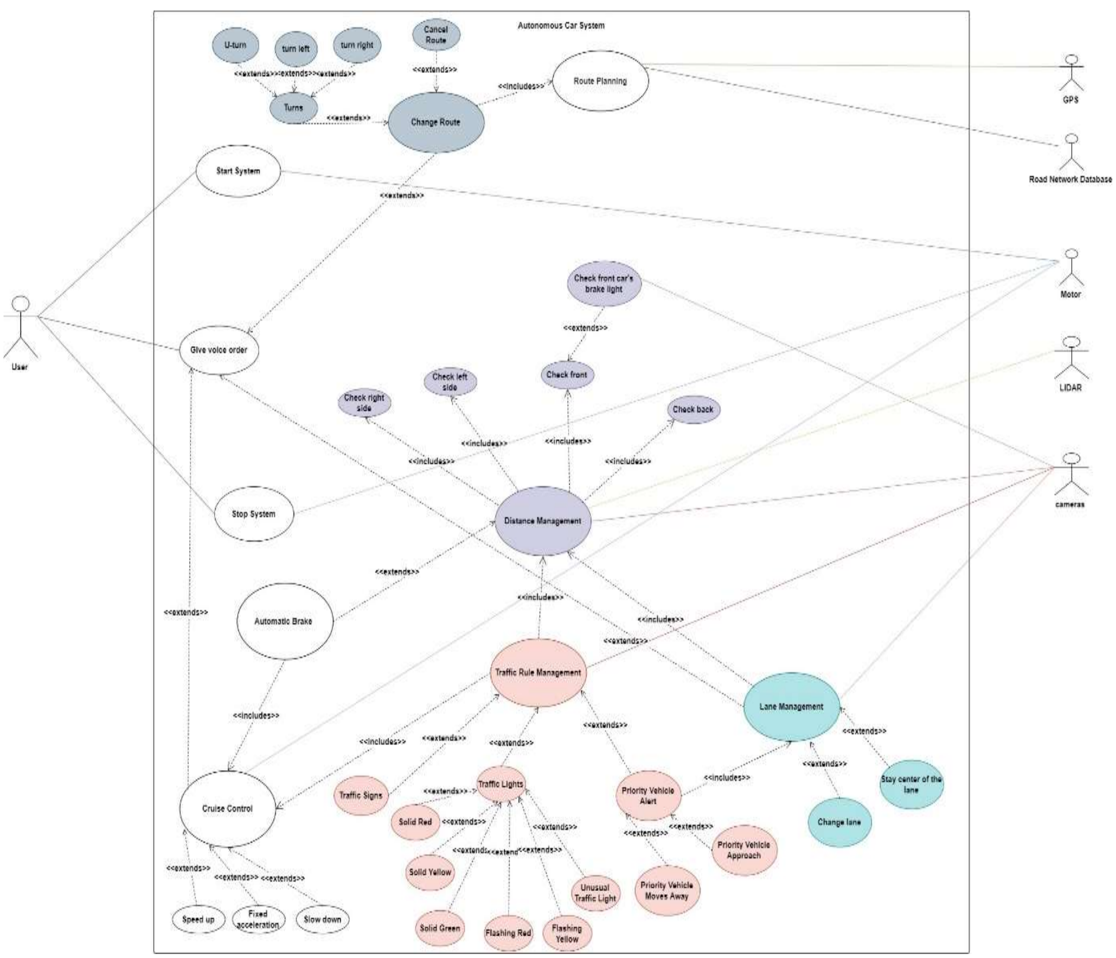


Figure 13 Use Case Diagram of our project.

Use Case Title	Description
Start System	User turns on the system.
Give Voice Order	User gives basic orders to system with voice.
Route Planning	System takes input location from user, verify its existence, finds shortest path to it.
Change Route	Changes the input location or cancels it.
Distance Management	Checks vehicles environment and finds estimate distance to objects.
Automatic Brake	Stops the car when an object appears in front of the car.
Lane Management	Manages the vehicle to stay centre of the lane or change the lane base of the Priority Vehicle Alert or user's request.
Cruise Control	Controls the acceleration. Slows down, speeds up or fixes acceleration of the vehicle.
Priority Vehicle Alert	Displays alert on the dashboard and takes necessarily actions when an Emergency Vehicle approaches or moves away.
Solid Red	The traffic signal in front of the car has a solid red light.
Solid Yellow	The traffic signal in front of the car has a solid yellow light.
Solid Green	The traffic signal in front of the car has a solid green light.
Flashing Red	The traffic signal in front of the car has a flashing red light.

Use Case Title	Description
Flashing Yellow	The traffic signal in front of the car has a flashing yellow light.
Unusual Traffic Light	The traffic signal in front of the car not working correctly (no lights on, all lights on etc.).
Traffic Signs	When there is a Traffic Sign in front of the car.
Stop System	User turns off the system.

a. ***Use Case: Start System***

Description: User turns on the system to use the vehicle.

Functional Response: The systems will turn on the sensors and checks other components. Waits for destination input.

b. ***Use Case: Give Voice Order***

Description: User can give orders by voice to change the destination or acceleration.

Functional Response: With the system's virtual assistant system will get the order and based on it Cruise Control Module or Lane Management Module or Change Route Modules will be activated.

c. ***Use Case: Route Planning***

Description: System takes destination input from user.

Functional Response: System gets destination input. If can't find destination on GPS or road network database displays error message. If else, it calculates shortest path to destination and shows route.

d. ***Use Case: Change Route***

Description: User can change route with voice order or can withdraw it.

Functional Response: Voice order will be recognized by system's virtual assistant. Based on the user's order new shortest path will be calculated or the vehicle will be parked on highway shoulder or parking areas until new order comes.

e. Use Case: Distance Management

Description: Checks vehicle's environment and finds estimate distance to objects with the data comes from cameras and LIDAR.

Functional Response: The computer vision data is sent to the machine learning model for classification. The model classifies the selected area of the environment is clear or not. And tries to balance distance between objects and the vehicle.

f. Use Case: Automatic Brake

Description: Stops the car when an object appears in front of the car.

Functional Response: When an object suddenly appears in front of the car, system automatically stops the car. The car remains stationary so long as the obstacle is present. Once cleared, it continues its destination.

g. Use Case: Lane Management

Description: Manages the lane operations.

Functional Response: In the lane it's balances the vehicle to stay in the center of the lane with the computer vision data comes from cameras. If there is a request from user to change the line uses Distance Management to check environment and changes the line based of the request.

h. Use Case: Cruise Control

Description: It controls the speed of the motor.

Functional Response: It changes the speed of the motor based on user's order or responds comes from Distance Management module.

i. Use Case: Priority Vehicle Alert

Description: Displays alert on the dashboard and takes necessarily actions when an Emergency Vehicle approaches or moves away.

Functional Response: When an emergency vehicle approaches, with receiver sensor the vehicle will recognize emergency alert and with cameras and sensors it will check if emergency vehicle is behind of the car and not on the opposite side of the road, then will switch to an available line to clear emergency vehicle's way. This module not only emptying based on one lane rule because emergency vehicle can approach from left lane, try to make an emergency corridor or can use shoulder of the road.

j. Use Case: Solid Red

Description: When there is a traffic light in front of the car with a solid red light. It means stop the car.

Functional Response: The car will slow down and stop until next signal input from computer vision.

k. Use Case: Solid Yellow

Description: When there is a traffic light in front of the car with a solid yellow light. It means get ready.

Functional Response: The car will slow down until see the next input. If it's red Solid Red Use case will be on, if it's green Solid Green Use case will be on.

l. Use Case: Solid Green

Description: When there is a traffic light in front of the car with a solid yellow light. It means go.

Functional Response: The car accelerates straight if there is no obstacle on the road.

m. Use Case: Flashing Red

Description: When there is a traffic light in front of the car with a flashing red light. It means the same as a stop sign. After stopping, proceed when safe and observe the right-of-way rules.

Functional Response: The car will slow down and stop until next signal input from computer vision

n. Use Case: Flashing Yellow

Description: When there is a traffic light in front of the car with a flashing yellow light. It warns to be careful. Slow down and be especially alert.

Functional Response: The vehicle will slow down and after checking environment with Distance Management. The car will continue its destination.

o. Use Case: Unusual Traffic Light

Description: When there is a traffic light in front of the car with no lights on or more than one lights on.

Functional Response: The vehicle will slow down and after checking environment with Distance Management. The car will continue its destination.

p. ***Use Case: Traffic Signs***

Description: When there is a Traffic Sign in front of the car.

Functional Response: The computer vision data is sent to the machine learning model for classification. The model classifies the sign and acts based on it.

q. ***Use Case: Stop System***

Description: Turns off the system.

Functional Response: When user wants to turn off the system, vehicle parks automatically in a safe area and stops the system.

4.3.3 Non-functional Requirements

This part includes non-functional requirements of the project.

4.3.3.1 Performance Requirements

Performance Requirement	**Description**
Response Time	This system work in real-time. So, response time (T_response=T_actuation-T_event) must be 500 msecs.
Error Handling	When an unpredictable failure occurs, system need to recover briefly.
Workload	System should be able to handle many inputs from its environment in different challenging weather and traffic conditions.
Scalability	Sensors and other used hardware tools effective on it but we’re

Performance Requirement	Description
	working on simulation there is no scalability requirement.

4.3.3.2 Safety Requirements

Safety Requirement	Description
Harm Protection	The autonomous car system needs to avoid accidents and shall not injure passengers.
Hazard Protection	The autonomous car system shall not start moving when its doors are still open. And notify user when safety belt has not worn.
Safety Error Identification	The vehicle needs to identify the system errors.
Safety Error Reporting	The system needs to notify user in the dangerous situation and let the user take vehicle's control when system not working well.
Ethical Considerations	The system needs to take ethical decisions in case of emergency.
Detection and Response	The vehicle needs to follow the traffic rules, detects and responds to objects in its environment.

4.3.3.3 Security Requirements

- The system needs to secure the sensors from external and internal threats.
- The algorithms behind these sensors must be secured.

4.3.3.4 Software Quality Attributes

Quality Attribute	**Description**
Correctness	The system needs to recognize objects, follow the traffic rules and avoid accidents.
Reliability	Every functionality on the code must be able to work smoothly without failure under given normal conditions.
Learnability	The system needs to be simple enough to learn by users.
Robustness	The autonomous car must work properly under given abnormal traffic or weather conditions.
Maintainability	When an unpredictable failure occurs, system needs to correct defects their cause, repair it and report the failure.
Extensibility	Ability to extend the system is limitless. New functionalities can be added to system anytime.
Testability	Every functionality on the code must be able to work smoothly in simulation environment.
Efficiency	The system should work with maximum performance with minimum used energy and fuel.
Portability	The system should work on Linux and Windows.

4.3.3.5 Business Rules

Our project uses Webots Self-Driving Vehicle Simulator. All the codes belong to us will be open source.

5. Software Design Description

5.1 Introduction

This Software Design Description Report provides detailed information about the requirements of the AutoCar system software. This document includes the working principles of the proposed methods, and designs of the simulation.

5.1.1 Purpose

Our main purpose is "Increasing the survival rate of the patients or people on the emergency situations by clearing the emergency vehicle's way with adding emergency vehicle priority awareness feature to autonomous cars". We maybe cannot apply this project to our lives since generally people do not use autonomous cars today, but it may help to humans in the future when all cars on the roads become autonomous.

5.1.2 Project Scope

On this project we aimed to add emergency vehicle priority awareness feature to autonomous cars. The autonomous car that we simulate will be able to detect the emergency vehicles, and their location and direction. The autonomous car will use visual information; therefore, it is required to have multiple cameras mounted on the vehicle.

Our system will include:

- Lane detecting and following
- Object recognition and auto brake
- Virtual drive assistant
- Route Planning
- Emergency vehicle priority awareness

Current autonomous cars already have the first four of these features. We will add fifth one as a new feature to autonomous cars.

To be able to success on this project our car at least needs to follow the lane, not hit objects on the road and let the emergency vehicle pass by changing lanes. Other features may change according to the possibilities of the simulator.

5.1.3 Definitions, acronyms, and abbreviations (Glossary / Terminology)

Terminology	Definition
User	A person who interacts with the system.
SDD	The report that provides an overview of system's design.
Autonomous Driving	A control mode which a vehicle doesn't need a driver attention.
Emergency Vehicle	An emergency vehicle is any vehicle that is designated and authorized to respond to an emergency in a life-threatening situation (42).
Vehicle Priority	When a vehicle with higher priority approaches, all other traffic must stop or move to the right side to allow the vehicles pass through.
System	System software is a type of computer program that is designed to run a computer's hardware and application programs (43).
Control Unit	The control unit (CU) is a component of a computer's central processing unit (CPU) that directs the operation of the processor (44).
ADAS	Advanced driver-assistance systems.
LIDAR	Light Detection and Ranging, is a remote sensing method that uses light in the form of a pulsed laser to measure ranges (variable distances) to the Earth (45).

Terminology	Definition
GPS	The Global Positioning System (GPS), originally NAVSTAR GPS, is a satellite-based radio navigation system owned by the United States government (46).

5.1.4 Overview of the Document

The remaining sections and their contents are listed below. Section Architectural Design that defines the stage of project development. The subheading of this topic includes the identification and explanation of the problem and the technologies used. The architect details the general diagrams of the simulator program as a subtitle of the Design section. Sequence Diagram, Activity Diagram and Class Diagram. The continuation of section includes the elaboration and detailed explanation of the project plan. In Environment section, we have detailed and demonstrated the tools in the simulation environment.

5.1.5 Motivation

In this project, we chose today's one of the most popular topic: Autonomous Vehicle. This project, which many engineers are curious about, was very suitable both for developing and keeping up with today's technology. We examined the studies on this subject. In order to keep up with this technology, we followed current developments and improved our knowledge on this field. Since we wanted to obtain more accurate results, we decided to use simulation environments rather than building the system with limited hardware. For this purpose, we have searched simulators and chose the most appropriate one which is 'Webots'. 'Webots' provides environment sound, radar sensor, Lidar sensor, map support etc.

5.2 Architecture Design

5.2.1 Description of Problem

Emergency vehicle priority is an important issue in the traffic. If we clear the way for emergency vehicles as quickly as possible, we can increase the survival rate of the patients or people on the emergency situations. Our project aiming to add vehicle priority awareness feature to autonomous cars. Of course, autonomous cars not common for today but in the future, everyone expected to use them.

5.2.2 Technologies Used

This software will be developed with Webots Simulator and its libraries for autonomous cars in Python.

5.2.3 Architecture Design of AutoCar

5.2.3.1 Sequence Diagram

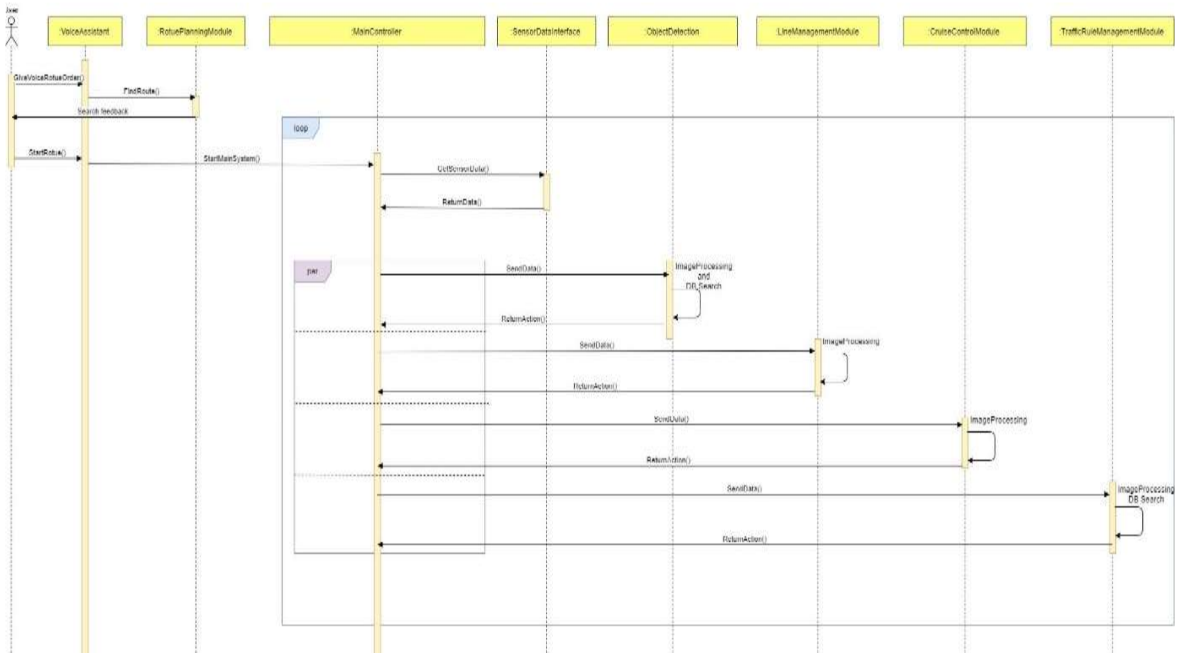


Figure 14 Sequence Diagram of our project.

Figure 14 shows sequence diagram of our project. The user gives order to voice assistant to choose the route and start it. After it starts Main controller handles everything by obtaining data from sensors and sends them to sub modules, takes necessarily actions.

5.2.3.2 Activity Diagram

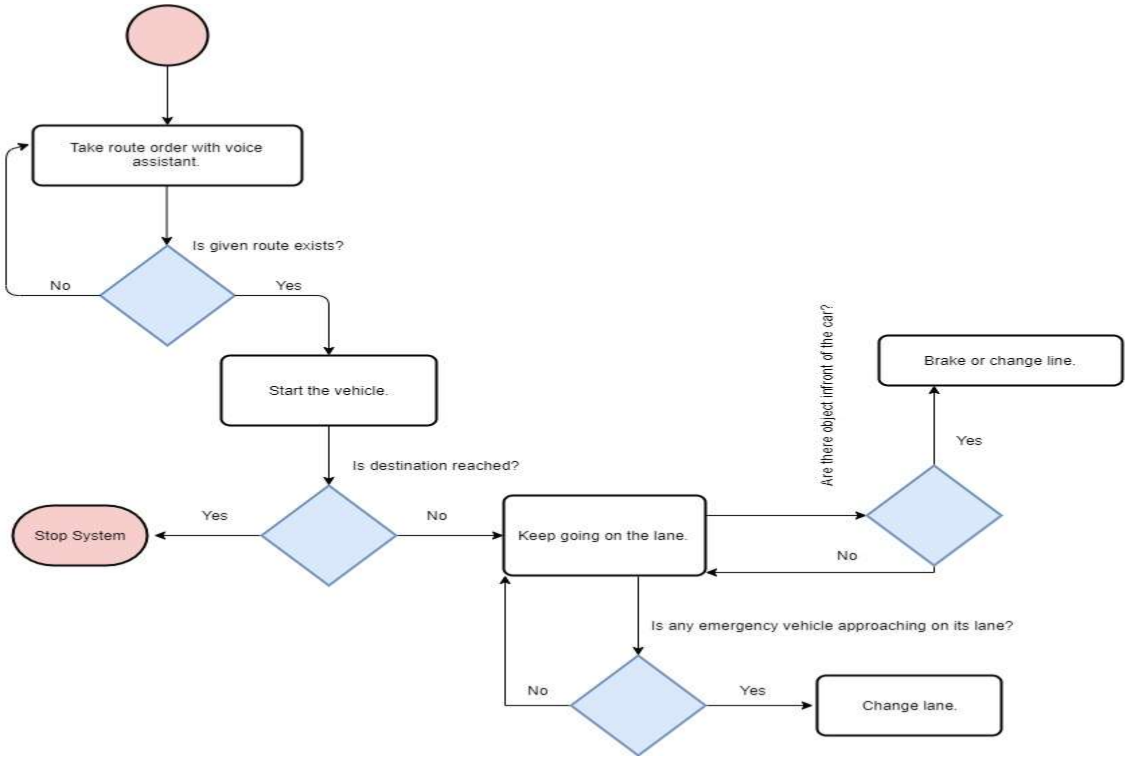


Figure 15 Activity Diagram of our project.

Figure 15 shows activity flows in our program. User gives order to system by voice assistant after system starts to run it tries to reach destination by itself.

5.2.3.3 Class Diagram

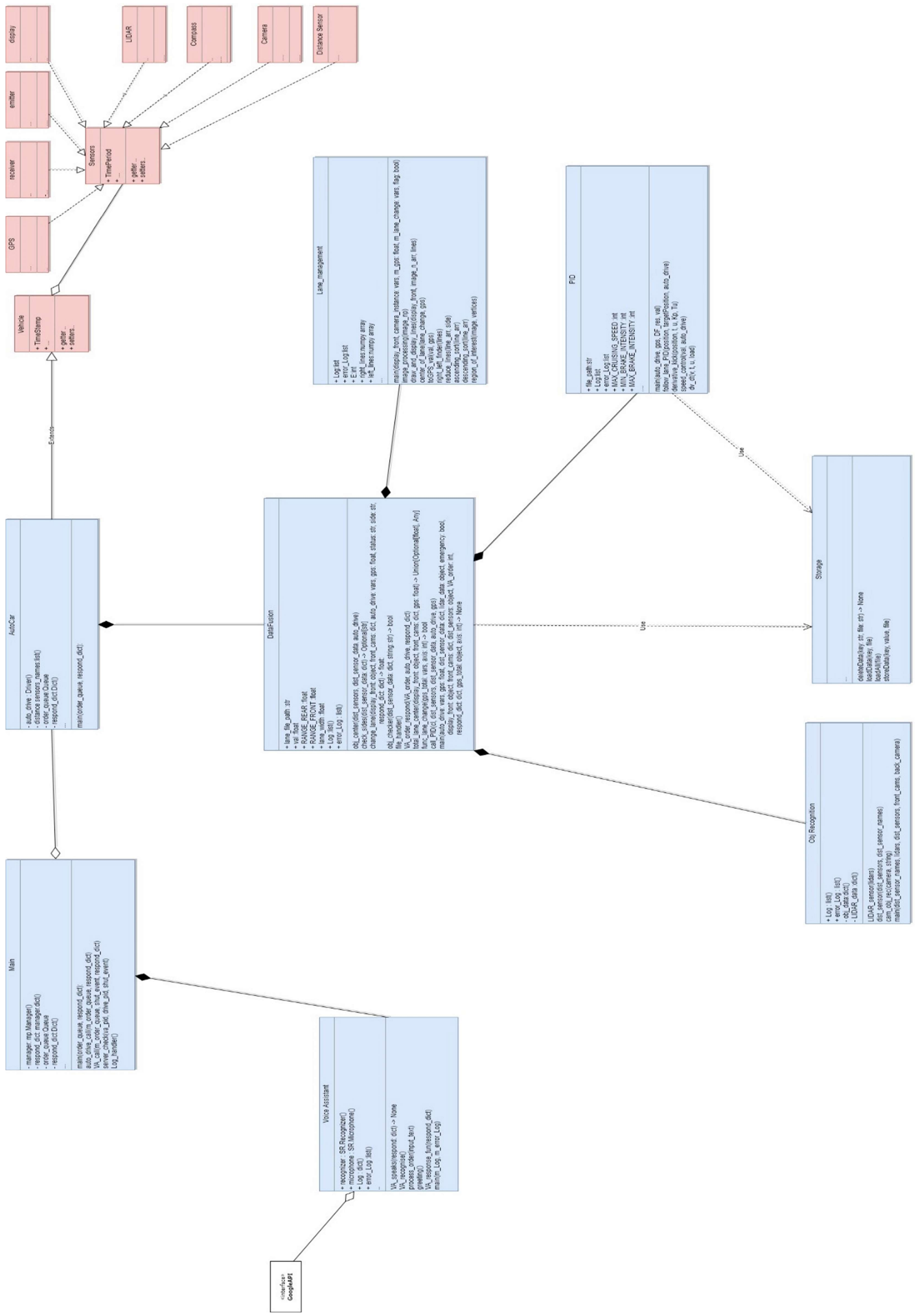


Figure 16 Class Diagram of our project.

5.2.3.4 Database Diagram

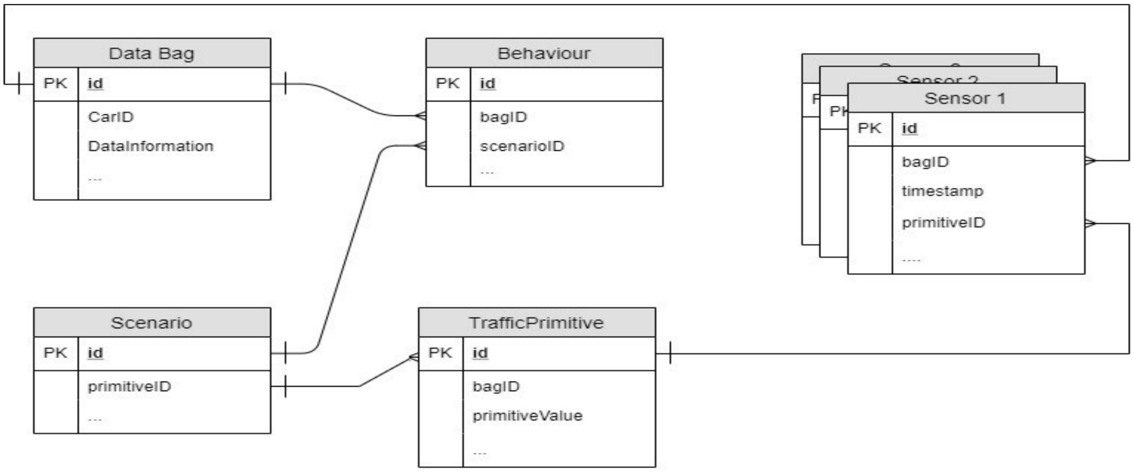


Figure 17 Database Diagram of our project.

In our Database we have ‘DataBag’, ‘Scenario’, ‘TrafficPrimitive’ and ‘Behavior’ tables to relate all the driving data. We do not need prior knowledge of driving scenarios and primitives, but we need manually defined several hypo-thetical scenario conditions. For persistent data and logs pickle and csv files has been used but for future to test more this DB can be used.

5.2.4 Architecture Design of AutoCar

5.2.4.1 Create Route

Summary: The user can select his/her location, determine the target position. With giving order by voice assistant.

Actor: User.

Precondition: The car must be connected to the Internet and keep the location settings open.

Basic Sequence:

1. The user calls voice assistant to start it.
2. The user tells desired destination location.
3. The user can select when to start travel.
4. The user can cancel the route.

Exception:

- Location error due to cellular data or another external factor.
- Incorrect orientation if map information is out of date.

Post Conditions: User must tell destination location information.

Priority: Medium.

5.2.4.2 Driving State

Summary: The car will try to reach the destination.

Actor: None.

Precondition: The car must be connected to the Internet and keep the location settings open.

Basic Sequence:

1. Start engine.
2. Follow the path to the destination.
3. Avoid collisions.
4. Consider traffic rules.
5. Clear the way if emergency vehicle approaches.

Exception: None.

Post Conditions: Sensors need to work well.

Priority: High.

5.2.4.3 Change Direction

Summary: The user can change the path with giving order by voice assistant.

Actor: User.

Precondition: The car must be connected to the Internet and keep the location settings open.

Basic Sequence:

1. The user calls voice assistant to start it.
2. The user tells desired direction or lane.
3. The route will recalculate by system.
4. The car follows the new path.

Exception: If there is not such a lane (ex: one right lane when there is not exists.) or direction (ex: when user orders to turn left but there is no way.).

Post Conditions: Sensors need to work well.

Priority: Medium.

5.2.5 Project Plan

In our work plan everything is clearly described from start and finish dates to owner of the work and given workday for the job. Due to current situation of Covid-19 we could not talk face to face but used communication applications.

CENG 408 Work Plan		10.02.2020 -14.03.2020	17.02.2020-21.02.2020	24.02.2020-28.02.2020	02.03.2020-06.03.2020	09.03.2020-13.03.2020	16.03.2020 -20.03.2020	23.03.2020 -27.03.2020	30.03.2020-03.04.2020	06.04.2020-10.04.2020	13.04.2020-17.04.2020	20.04.2020-24.04.2020	13.04.2020 -17.04.2020	27.04.2020-01.05.2020	04.05.2020-08.05.2020	11.05.2020-22.05.2020
Documentations	**Work Owner**	WEEK 1	WEEK 2	WEEK3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12	WEEK 13	WEEK 14	WEEK 15-16
All Documentation Updates	Büşra Nur															
Test Plan Documents	Team															
Installation Guide	Büşra Nur												Due Date			
Project Raport	Büşra Nur													Due Date		
Poster	Zehra														Due Date	
Presentation	Zehra															Due Date
Coding	**Work Owner**								DEMO			First Release				
Lane detection (Image processing)	Büşra Nur															
Object recognition and tracking	Büşra Nur															
Data Fusion and PID	Büşra Nur															
Voice assistant	Büşra Nur															
Multiprocessing between VA and Driver	Büşra Nur															
~~Voice assistant~~	~~Zehra~~															
Testing	**Work Owner**															
Autonomous Car Features	Büşra Nur															
Voice assistant	Büşra Nur															

Figure 18 Work Plan of our project

5.3 Environment

5.3.1 Modelling Environment

In our project we used Webots simulation to be able to test our codes. Among of many self-driving simulators. Webots is a professional mobile robot simulation software package. It offers a rapid prototyping environment, that allows the user to create 3D virtual worlds with physics properties such as mass, joints, friction coefficients, etc. (47). We chose Webots because it is cross platform, open source, supports many languages including Python and has environment sound. It is also providing tools as OpenStreeMap to create our own world and SUMO (Simulation of Urban MObility) interface to simulate traffic and sensors as LIDAR, radar, GPS etc.

Figure 19 A car using LIDAR from sample Webots world.

6. Test Plan

6.1 Introduction

6.1.1 Version Control

Version No	Description of Changes	Date
v1.0.2	Second Version with Voice Assistant and Multiprocessing	May 15, 2020

6.1.2 Overview

AutoCar System is a Simulated Autonomous Car System. The use case of the system had been determined in SRS document will be tested on the simulation environment.

6.1.3 Scope

This document encapsulates the test plan of the use cases, test design specifications and the test cases correspond to test plan.

6.1.4 Terminology

Acronym	Definition
PID	The PID proportional-integral-derivative controller is a control loop method is a commonly used feedback controller method in industrial control systems.

Acronym	Definition
Data Fusion	Data Fusion is the process of integrating multiple data sources to produce more consistent, accurate and useful information than any data source provides.
Emergency Vehicle	The emergency vehicle is any vehicle designed and authorized to respond to the emergency in a life-threatening situation.

6.2 FEATURES TO BE TESTED

This section lists and gives a brief description of all the major features to be tested. For each major feature there will be a Test Design Specification added at the end of this document.

6.2.1 Driving State

In the simulation the vehicle must follow the its lane, avoid collisions with the objects on the road and consider traffic rules.

6.2.2 Emergency Awareness

In this part vehicle need to detect emergency vehicle and change its lane to clear emergency vehicle's path.

6.2.3 Voice Assistant

In this part system need to take input from user for basic orders as changing lane, speed. It includes display to show system values to user.

6.3 FEATURES NOT TO BE TESTED

6.3.1 System accuracy on different cars

We will not test on different cars to check difference.

6.3.2 System accuracy according to weather

Simulation environment does not provide different types of weather (rainy, snowy...).

6.3.3 Traffic Lights and Path Planning

These parts have lower priority than other cases for us and we have very limited time for this project.

6.4 ITEM PASS/FAIL CRITERIA

To be able to success on this project our car needs to follow the lane, not hit objects on the road and let the emergency vehicle pass by changing lanes.

6.4.1 Exit Criteria

- 100% of the test cases are executed
- All High and Medium Priority test cases passed

6.5 REFERENCES

(48)https://github.com/CankayaUniversity/ceng-407-408-2019-2020-Autonomous-Car/wiki/SDD

(49) https://github.com/CankayaUniversity/ceng-407-408-2019-2020-Autonomous-Car/wiki/SRS

6.6 TEST DESIGN SPECIFICATIONS

6.6.1 Driving State (DS)

6.6.1.1 Sub features to be tested

a. Lane Detection and Management (DS.LDM)

System need to detect lane and manage to stay on the center of the lane. And find if there is a one left or one right lane exits. With processing the image data from front camera.

b. Object Recognition and Tracking (DS.ORT)

System need to recognize objects in its environment and trigger PID system if there is a need.

c. Data Fusion and PID (DS.DFPID)

System need to make data fusion according to data from sensors. And take actions according to result or in voice order case with PID.

6.6.1.2 Sub features to be tested Test Cases

TC ID	Require ments	Priority	Scenario Description
DS.LDM.01	3.2.1.7.	H	Take image data from front camera. Process it to find lines.
DS.LDM.02	3.2.1.7.	H	Find center of the lane.
DS.LDM.03	3.2.1.7.	H	Decide line directions and find if next lanes on the sides are exists.

TC ID	Requirem ents	Priority	Scenario Description
DS.DFPID.01	3.2.1.8.	H	Take current velocity data and update it according to result from DS.ORT.01 with PID control.
DS.DFPID.02	3.2.1.7.	H	Take GPS data and lane data from DS.LDM.02 to follow current lane.
DS.DFPID.03	3.2.1.6.	H	Take sensors data from (radar, lidar.. and DS.ORT.01) and side lane data from DS.LDM.03 to take necessarily actions (lane changing, increase brake intensity...)

TC ID	Requirem ents	Priority	Scenario Description
DS.ORT.01	3.2.1.5.	H	Take image data from both front and back cameras. Process it to find objects and return relative positions of near objects to DS.DFPID.03.

6.6.2 Emergency Awareness (EA)

6.6.2.1 Sub features to be tested

a. Recognize Emergency Vehicle's Siren (EA.REVS)

System need to recognize emergency vehicle's siren and check position of the vehicle DS.ORT.01.

If vehicle's position is in the same lane with the car it needs to change lane.

6.6.2.2 Test Cases

TC ID	Requirements	Priority	Scenario Description
EA.REVS.01	3.2.1.9.	H	Recognize emergency siren.
EA.REVS.02	3.2.1.9.	H	Take data from DS.ORT.01 to check position of the emergency vehicle and if vehicle's position is in the same lane with the car trigger DS.DFPID.03 to change lane.

6.6.3 Voice Assistant (VA)

6.6.3.1 Sub features to be tested

a. Activate Voice Assistant (AVA)

After the system starts, the voice assistant is starts with the system. With the trigger sentence, user activates the voice assistant.

b. Speech Recognition and Respond (SRR)

The desired command will be received by voice and will be convert into text with API. After this command is confirmed, the process starts. System values vary according to this process on the screen.

6.6.3.2 Test Cases

TC ID	Requirements	Priority	Scenario Description
VA.AVA. 01	3.1	H	Actuate the voice assistant starting with the system with the trigger phrase(“Hi car”ex.).
VA.AVA. 02	3.1	H	VA will continue to listen even if the user does not command.
TC ID	**Requirements**	**Priority**	**Scenario Description**
VA.SRR. 01	3.2.1	H	It translates the "Change lane " command, which the user says verbally, into text and makes the system understand.
VA.SRR. 02	3.2.2	M	With “Speed up" command, which the user says verbally, into text and makes the system understand.
VA.SRR.03	3.2.3	M	With “Slow down" command, which the user says verbally, into text and makes the system understand.

6.7 Detailed Test Cases

6.7.1 DS.LDM.01

TC_ID	DS.LDM.01
Purpose	Process image data from front camera to find lines.
Requirements	3.2.1.7
Priority	High.
Estimated Time Needed	1 sec
Dependency	Simulation should be started. Camera device must be exist on the car.
Setup	Camera must be working.
Procedure	[A01] Get image data from camera device.
	[A02] Process data to find lines
	[A03] Reduce lines.
	[V01] Observe that lines founded successfully
Cleanup	New image data input

6.7.2 DS.LDM.02

TC_ID	DS.LDM.02
Purpose	Find center of the lane.
Requirements	3.2.1.7
Priority	High.
Estimated Time Needed	1 sec
Dependency	DS.LDM.01 need to work correctly.
Setup	Simulation should be started. Camera device must be exist on the car.
Procedure	[A01] Get line data from DS.LDM.01.
	[A03] Find center of the lane.
	[A04] Draw lines on display screen.
	[V01] Observe that lines displayed on the screen correctly.
Cleanup	New lane data input

6.7.3 DS.LDM.03

TC_ID	DS.LDM.03
Purpose	Decide line directions and find if next lanes on the sides are exists.
Requirements	3.2.1.7
Priority	High.
Estimated Time Needed	1 sec
Dependency	DS.LDM.01 need to work correctly.
Setup	Simulation should be started. Camera device must be exist on the car.
Procedure	[A01] Get line data from DS.LDM.01.
	[A02] Decide line directions
	[A03] Find if side lane exists.
	[V01] Observe that existed side lanes has been found successfully.
Cleanup	New lane data input

6.7.4 DS.ORT.01

TC_ID	DS.ORT.01
Purpose	Process image data to find objects and return relative positions of near objects to DS.DFPID.03.
Requirements	3.2.1.5.
Priority	High.
Estimated Time Needed	1 sec
Dependency	Camera device must be exist on the car.
Setup	Camera must be working.
Procedure	[A01] Get image data from the cameras.
	[A02] Find objects.
	[A03] Set flag if emergency vehicle exists on the road
	[V01] Observe that relative positions of near objects to returned to DS.DFPID.03 and flag has been set on the existence of emergency car.
Cleanup	New image data input

6.7.5 DS.DFPID.01

TC_ID	DS.DFPID.01
Purpose	PID control for velocity.
Requirements	3.2.1.8.
Priority	High.
Estimated Time Needed	1 sec
Dependency	Necessarily devices must be exists on the car.
Setup	Simulation should be started.
Procedure	[A01] Get current velocity of the car.
	[A02] Apply PID on the velocity data.
	[V01] Observe that the updated velocity has been successfully set.
Cleanup	New velocity data input

6.7.6 DS.DFPID.02

TC_ID	DS.DFPID.02
Purpose	PID control for position.
Requirements	3.2.1.7.
Priority	High.
Estimated Time Needed	1 sec
Dependency	Necessarily devices must be exists on the car.
Setup	Simulation should be started.
Procedure	[A01] Get current position of the car from GPS.
	[A02] Apply PID on the position data.
	[V01] Observe that the updated steering angle has been successfully set.
Cleanup	New position data input

6.7.7 DS.DFPID.03

TC_ID	DS.DFPID.03
Purpose	Data Fusion.
Requirements	3.2.1.6.
Priority	High.
Estimated Time Needed	1 sec
Dependency	Necessarily devices must be exists on the car.
Setup	Simulation should be started.
Procedure	[A01] Get sensor data.
	[A02] Process data.
	[V01] Observe that data processed successfully and PID controllers triggered in the necessarily situations.
Cleanup	New data input

6.7.8 EA.REVS.01

TC_ID	EA.REVS.01
Purpose	Recognize emergency siren.
Requirements	3.2.1.9.
Priority	High.
Estimated Time Needed	1 sec
Dependency	Sound sensor device must be exist on the car.
Setup	Simulation should be started.
Procedure	[A01] Get sound data from environment with sensor.
	[A02] Process it.
	[V01] Observe that emergency siren has been successful recognized.
Cleanup	New sound data input

6.7.9 EA.REVS.02

TC_ID	EA.REVS.02
Purpose	Trigger DS.DFPID.03 to change lane.
Requirements	3.2.1.9.
Priority	High.
Estimated Time Needed	1 sec
Dependency	EA.REVS.01 recognizes siren.
Setup	Simulation should be started.
Procedure	[A01] Check positions of the cars.
	[V01] Observe that system trigger DS.DFPID.03 to change lane.
Cleanup	New sound data input

6.7.10 VA.AVA.01

TC_ID	VA.AVA.01
Purpose	To activate the voice assistant with the trigger sentence.
Requirements	3.1
Priority	High.
Estimated Time Needed	5 sec
Dependency	Necessarily devices must be exists on the car.
Setup	Simulation should be started.
Procedure	[A01] Say the trigger sentence.
	[V02] Voice Assistant will be displayed on the screen.
Cleanup	-

6.7.11 VA.AVA.02

TC_ID	VA.AVA.02
Purpose	Voice assistant waits for a command.
Requirements	3.1
Priority	High.
Estimated Time Needed	-
Dependency	Necessarily devices must be exists on the car.
Setup	Simulation should be started.
Procedure	[A01] Say the trigger sentence.
	[V02] Observe the voice assistant.
Cleanup	New command input

6.7.12 VA.SRR.01

TC_ID	VA.SRR.01
Purpose	Change the lane of the vehicle.
Requirements	3.2.1
Priority	High
Estimated Time Needed	5 sec
Dependency	VA.AVA.01 and VA.AVA.02 must work successfully
Setup	The system should find the algorithm of the desired command.
Procedure	[A01] Say the trigger sentence. ("hi car")
	[V01] Observe the voice assistant.
	[A02] Say the "Go to the right lane" or "Go to the left lane"
	[V02] The assistant takes the verbal command and translates it into writing with API.
	[V03] Trigger necessary functions in the system.
Cleanup	-

6.7.13 VA.SRR.02

TC_ID	VA.SRR.02
Purpose	Change the speed of the vehicle.
Requirements	3.2.2
Priority	Medium
Estimated Time Needed	5 sec
Dependency	VA.AVA.01 and VA.AVA.02 must work successfully
Setup	The system should find the algorithm of the desired command.
Procedure	[A01] Say the trigger sentence. (“hi car”)
	[V01] Observe the voice assistant.
	[A02] Say the “Increase vehicle’s speed”
	[V02] The assistant takes the verbal command and translates it into writing with API.
	[V03] Trigger necessary functions in the system.
Cleanup	-

6.7.14 VA.SRR.03

TC_ID	VA.SRR.03
Purpose	Change the speed of the vehicle.
Requirements	3.2.3
Priority	Medium
Estimated Time Needed	5 sec
Dependency	User must pass Speech to Text test scenarios.
Setup	The system should find the algorithm of the desired command.
Procedure	[A01] Say the trigger sentence. (“hi car”)
	[V01] Observe the voice assistant.
	[A02] Say the “Decrease vehicle’s speed”
	[V02] The assistant takes the verbal command and translates it into writing with API.
	[V03] Trigger necessary functions in the system.
Cleanup	-

6.8 TEST RESULTS

6.8.1 Individual Test Results

Table of Individual Test Results

TC ID	Priority	Date Run	Run By	Result	Explanation
DS.LDM.01	H	05.15.2020	Büşra Nur Bahadır	Pass	Processed image data from front camera to find lines.
DS.LDM.02	H	05.15.2020	Büşra Nur Bahadır	Pass	Found center of the lane
DS.LDM.03	H	05.15.2020	Büşra Nur Bahadır	Pass	Decided directions of the lines.
DS.ORT.01	H	05.15.2020	Büşra Nur Bahadır	Pass	Found objects and returned relative positions.
DS.DFPID.01	H	05.15.2020	Büşra Nur Bahadır	Pass	PID control for velocity works.
DS.DFPID.02	H	05.15.2020	Büşra Nur Bahadır	Pass	PID control for position in the lane works.
DS.DFPID.03	H	05.15.2020	Büşra Nur Bahadır	Pass	Data Fusion works and triggers PID controls according to results of Lane management and object recognition.
EA.REVS.01	H	05.15.2020	Büşra Nur Bahadır	Pass	Recognize emergency siren by emitter and receivers.
EA.REVS.02	H	05.15.2020	Büşra Nur Bahadır	Pass	Triggered DS.DFPID.03 to change lane.

VA.AVA.01	H	05.15.2020	Büşra Nur Bahadır	Pass	Voice assistant gave respond to user.
VA.AVA. 02	H	05.15.2020	Büşra Nur Bahadır	Pass	Voice assistant waits user to give order in 5 sec. worked.
VA.SRR.01	H	05.15.2020	Büşra Nur Bahadır	Pass	Lane changed with user order.
VA.SRR. 02	M	05.15.2020	Büşra Nur Bahadır	Pass	Car speed had been increased by user order.
VA.SRR.03	M	05.15.2020	Büşra Nur Bahadır	Pass	Car speed had been decreased by user order.

6.8.2 Summary of Test Results

Priority	Number of TCs	Executed	Passed
High	12	12	12
Medium	2	2	2
Low	0	0	0
Total	14	14	14

We have executed 14 test cases and 14 test cases are passed. Also, all the high and all the medium priority test cases are passed. Exit criteria is met.

6.8.3 Exit Criteria

We have executed all test cases and all of them are passed. Software development activities are completed within the anticipated timeline. Exit criteria is met.

Criteria	Met or Not
100% of the test cases are executed	M
All High and Medium Priority test cases passed	M

7. Conclusions

In last semester we did researches about Self Driving Car algorithms and systems. We found that Autonomous cars do not have vehicle priority awareness feature and decided to add this feature as an improvement on this field in our project. According to reports we prepared last semester about our system needs this semester we coded AutoCar system.

The simulated autonomous car that has been coded, can be able to detect emergency vehicles, their location and direction with the sensors and cameras which are mounted on it. When the autonomous car recognises the emergency vehicles, it will change the lane to clear the emergency vehicle's way. AutoCar have features such as lane detection and tracking, object recognition and automatic braking, voice assistant, and emergency vehicle priority awareness. The current autonomous cars already have the fourth of these features. We added fifth feature as an innovation to autonomous cars.

In Lane Detection OpenCV and Hough Probabilistic Line, in PID control of speed momentum balance equation and for lane position PID Ziegler Nichols method, for Voice Assistant Google Speech to Text API has been used.

In last semester we told that to be able to success on this project our car at least follows its lane, clear emergency car's way by changing line, take commands via voice assistant and avoid objects on the road. AutoCar does all of them.

8. Acknowledgement

We are grateful for guidance we have received from our advisor Dr Instructor Roya CHOUPANI.

9. Installation Guide

For Simulation environment Webots has been used on this project:

Download Webots

After download the program download Python 2.7 or Python 3.7

For Windows:

```
run pip3.py on your PYTHON_PATH/Scripts
```

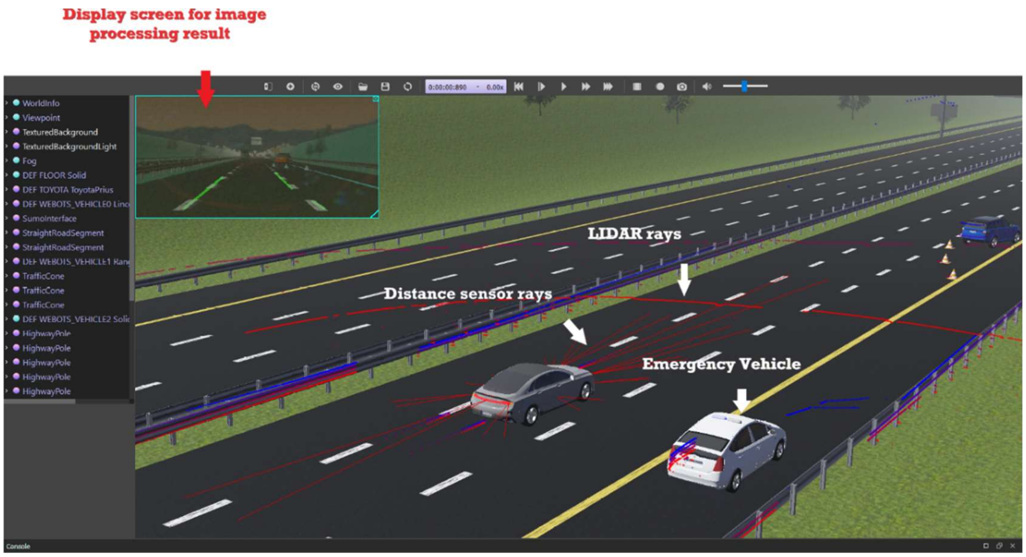
For Linux:

```
sudo apt-get install python-pip
```

Use the pip command to install:

```
pip install keyboard
pip install scipy
pip install numpy
pip install opencv-python
pip install pickle
pip install SpeechRecognition
pip install gTTS
pip install playsound
pip install pipwin
pipwin install pyaudio
```

After these you can simply open the .wbt file on the project. And just start by clicking play button.



You can watch the video by ctrl clicking on it ↓

To be able to see distance sensor rays use ctrl+F10.
For enabling to start and stop ambulance siren use e and q keys.
To call voice assistant just say Jarvis.
VA responds to =>

speed of the car
speed up
slow down
change lane
exit

On the sample projects you can find a same named .wbt file we edited that world environment to use for testing our codes.

If you want to look at the codes or develop more features on the project : File> Open Text File and choose any .py or for using PyCharm on it you can follow this manual.

10. References

1. **[Online] https://dergipark.org.tr/tr/download/article-file/614340.**

2. [Online] https://en.wikipedia.org/wiki/Self-driving_car.

3. [Online] http://cyberneticzoo.com/cyberneticanimals/1960-stanford-cart-american/.

4. [Online] https://en.wikipedia.org/wiki/DARPA_Grand_Challenge.

5. Cargegie Mellon University: Tartan Racing. [Online] http://www.cs.cmu.edu/~tartanrace/tech.html.

6. [Online] https://en.wikipedia.org/wiki/Tesla_Model_S.

7. *Sensor and object recognition technologies for self-driving cars. Computer-Aided Design and Application.* . Hirz M., Walzel B. 2018.

8. *Grouping dominant orientations or Ill-structured road following. International Conference on Computer Vision and Pattern Recognition.* C.:, Rasmussen. s.l. : IEEE, 2004.

9. *Rapid Vanishing Point Estimation for General Road Detection.* O., Misik. 2012.

10. *Vanishing point detection for self-driving car using harmony search algorithms. Swarm and Evolutionary Computation.* Moon Y. Y., Geem Z. W., Han G. T. 2018.

11. [Online] http://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.html.

12. [Online] https://towardsdatascience.com/tutorial-build-a-lane-detector-679fd8953132.

13. *International Conference on Robotics and Automation.* s.l. : IEEE, 2012.

14. Mur-Artal E., Tardos J.D. ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. Transactions on Robotics PP (99). IEEE, 2016.

15. Rublee E., Rabaud V., Konolige K., Bradski G. :ORB: an efficient alternative to SIFT or SURF. http://www.willowgarage.com/sites/default/files/orb_final.pdf .

16. Engel J., Shöps T.,. Cremers D.: LSD-SLAM: Large-Scale Direct Monocular SLAM. Computer Vision. ECCV, 2014. .

17. Petridis V., Zikos N.:. L-SLAM: „Reduced dimensionality FastSLAM algorithms. The 2010 International Joint Conference on Neutral Networks (IJCNN). 2010. .

18. Thrun S., Montemerlo M.:. The GraphSLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures. The International Journal of Robotics Research. 2006. .

19. Colleens T., Colleens J. J., Ryan C.:. Occupancy grid mapping: An empirical evaluation. Control & Automation. IEEE, 2007. .

20. Shuster F., Keller C., Rapp M. Haueis M.: Curio C.:. Landmark based radar slam using graph optimization. Intelligent Transportation Systems (ITSC). IEEE, 2016 .

21. Thrun S., Burgard W., Fox D.:. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. International Conference on Robotics and Automation. IEEE, 2000 .

22. Trehard C., Pollard E., Bradai B., Nashashibi F.:. Credibilist SLAM Performances with Different Laser Set-ups. 13th Interational Conference on Control, Automation, Robotics and Vision (ICARCV). 2014. .

23. Smets P., Kennes R.:. The transferable belief model. Artificial Inteligence (66), No. 2. Elsevier, 1994. .

24. Mendes E., Koch P., Lacroix S. ICP-based pose-graph SLAM. International Symposium on Safety, Security, and Rescue Robotics (SSRR). IEEE, 2016. .

25. Besl P. J., McKay H. D. A method for registration of 3-D shapes, IEEE PAMI (14), nr 2. IEEE, 1992 .

26. A., Datta. Google releases SLAM tool Cartographer to open source community. Geospatial World, 2016.https://www.geospatialworld.net/blogs/google-open-sources-slam-tool-cartographer .

27. Nuchter A., Bleier M., Schauer J., Janotta P. Impoving Google's Cartographer 3D Mapping by Continuous-Time SLAM. https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2-W3/543/2017/isprs-archives-XLII-2-W3-543-2017.pdf .

28. Zhang J., Singh S. LOAM. *Lidar Odometry and Mapping in Real-time. Conference: Robotics: Science and Systems Conference..* 2014.

29. [Online] https://lig-membres.imag.fr/aycard/html/Publications/2012/Azeem12.pdf .

30. Buehler, M. and Iagnemma, K. TheDARPAUrbanChallenge:. AutonomousVehiclesinCityTraffic Volume 56. . Berlin, Germany : s.n., 2009.

31. M., Olsson. Behavior Trees for decision-making in Autonomous Driving. 2016. http://www.diva-portal.org/smash/get/diva2:907048/FULLTEXT01.pdf .

32. Hu M., Liao Y., Wang W. Decision Tree-Based Maneuver Prediction for Driver Rear- End Risk-Avoidance Behaviors in Cut-In Scenarios. Journal of Advanced Transportation. 2017. .

33. Claussmann L., Carvalho A., Schildbach G. A path planner for autonomous driving on highways using a human mimicry approach with binary decision diagrams. European Control Conference. 2015. .

34. Ulbrich S., Mauerer M. Probabilistic Online POMDP Decision Making for Lane Changes in Fully Automated Driving. 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). IEEE, 2013. .

35. Zhang J., Liao Y., Wang S., Han J. Study on Driving Decision-Making Mechanism of Autonomous Vehicle Based on an Optimized Support Vector Machine Regression. Applied Sciences. 2017. .

36. Learning, Mueller M. A.: Reinforcement. MDP Applied to Autonomous Navigation. 2017. http://aircconline.com/mlaij/V4N4/4417mlaij01.pdf .

37. N., Legrand. Deep Reinforcement Learning for Autonomous Vehicle Control among Human Drivers. 2017. https://ai.vub.ac.be/sites/default/files/thesis_legrand.pdf .

38. [Online] https://cyberbotics.com/doc/guide/introduction-to-webots.

39. [Online] https://en.wikipedia.org/wiki/Emergency_vehicle .

40. [Online] https://whatis.techtarget.com/definition/system-software .

41. [Online] https://en.wikipedia.org/wiki/Control_unit .

42. [Online] https://en.wikipedia.org/wiki/Emergency_vehicle .

43. [Online] https://whatis.techtarget.com/definition/system-software .

44. [Online] https://en.wikipedia.org/wiki/Control_unit .

45. [Online] https://oceanservice.noaa.gov/facts/lidar.html.

46. [Online] https://en.wikipedia.org/wiki/Global_Positioning_System.

47. [Online] https://cyberbotics.com/doc/guide/introduction-to-webots.

49. AutoCarSRS. [Online] https://github.com/CankayaUniversity/ceng-407-408-2019-2020-Autonomous-Car/wiki/SRS.

50. SDD, AutoCar. https://github.com/CankayaUniversity/ceng-407-408-2019-2020-Autonomous-Car/wiki/SDD. [Online]

Figure 1: https://medium.com/deep-learning-turkiye/otonom-araclardaki-derin-ogrenme-mantigi-nasil-calisir-9f0fb59ba0a5
Figure 2: https://towardsdatascience.com/tutorial-build-a-lane-detector-679fd8953132
Figure 3: http://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.html
Figure 4, 5: https://towardsdatascience.com/tutorial-build-a-lane-detector-679fd8953132
Figure 6: https://www.mathworks.com/help/vision/ug/getting-started-with-semantic-segmentation-using-deep-learning.html
Figure 7: https://medium.com/deep-learning-turkiye/otonom-araclardaki-derin-ogrenme-mantigi-nasil-calisir-9f0fb59ba0a5
Figure 8: https://blogs.nvidia.com/blog/2019/04/15/how-does-a-self-driving-car-see/
Figure9: https://www.researchgate.net/profile/Jelena_Kocic3/publication/329153240/figure/fig2/AS:696154882842627@1542987666684/Multiple-sensor-perception-system-28.ppm
Figure 10: https://medium.com/@wilburdes/sensor-fusion-algorithms-for-autonomous-driving-part-1-the-kalman-filter-and-extended-kalman-a4eab8a833dd
Figure 11: https://lig-membres.imag.fr/aycard/html/Publications/2012/Azeem12.pdf

Figure 12: Article, Study on Driving Decision-Making Mechanism of Autonomous Vehicle Based on an Optimized Support Vector Machine Regression Junyou Zhang, Yaping Liao, Shufeng Wang and Jian Han.