AI-Based Navigation and anti-collision system for Autonomous Vehicles

To Phuoc Hung

Table of Contents

[1. Overview 2](#_Toc180235093)

[2. Motivation 2](#_Toc180235094)

[3. Background and Problem Statement 2](#_Toc180235095)

[4. Objectives 3](#_Toc180235096)

[5. Research Design 3](#_Toc180235097)

[5.1. Automated Navigation system 3](#_Toc180235098)

[5.1.1. Supervised Learning 3](#_Toc180235099)

[5.1.2. Reinforcement Learning 5](#_Toc180235100)

[5.2. Real-time collision avoidance system 6](#_Toc180235101)

[5.2.1. Spacial Consciousness 6](#_Toc180235102)

[5.2.2. Movement Prediction 6](#_Toc180235103)

[6. Research Instrumentation 7](#_Toc180235104)

[7. Methodology 8](#_Toc180235105)

[8. References 8](#_Toc180235106)

# Overview

The goal of the proposed project is to create an AI-powered self-driving car system that can drive itself from point A to point B in a town while avoiding lane invasions and keeping in its lane. To avoid collisions, the system has a real-time anti-collision system that recognizes and avoids approaching cars. It continuously scans the surroundings and guarantees safe navigation with the use of LiDAR sensors.

To check the solution's quality in realistic traffic situations, it will be tested in CARLA - an open-source autonomous driving simulator.

# Motivation

The concept of creating an AI-powered navigation system for self-driving cars appeals to me since it solves one of the most pressing issues in autonomous driving: safe and efficient navigation in complicated urban areas. Self-driving cars have the potential to transform transportation by improving road safety, alleviating traffic congestion, and increasing mobility. However, guaranteeing that self-driving vehicles can safely negotiate unexpected traffic and prevent collisions remains a big technical challenge.

This project aims to improve the safety and functionality of autonomous vehicles by employing AI, including machine learning and real-time decision-making algorithms. I am very interested in AI's capacity to tackle real-world problems, and I feel that applying it to autonomous car navigation will help to advance the technology in significant ways. Furthermore, the addition of an anti-collision system that avoids obstacles in real time emphasizes the importance of vehicle safety, which is a personal interest of mine in the field of autonomous driving research. This project nicely matches my ambition to merge cutting-edge AI technology with practical, real-world applications to better the future of transportation.

# Background and Problem Statement

One of the major issues with existing self-driving systems is their ability to stay inside specified lanes while traveling, particularly in cities where traffic can be chaotic and unpredictable. Lane invasions and drifting out of lanes can result in accidents or injurious driving reactions. Furthermore, current collision avoidance technologies are not always efficient in avoiding rapidly moving cars, causing serious dangers.

This research aims to solve these basic challenges by creating an AI-based navigation system for self-driving vehicles that can maintain lane integrity and avoid crashes in real time. The proposed technology will allow a self-driving automobile to safely go from point A to point B by automatically adapting to its surroundings and avoiding approaching vehicles or obstacles.

# Objectives

1. **Create an automated navigation system** that can drive across the city safely, following lane boundaries and preventing lane invasions.
2. **Implement a real-time collision avoidance system** that detects and dodges approaching vehicles or other obstacles, ensuring safe maneuvering through unpredictable traffic scenarios.
3. **Integrate advanced perception systems** using sensors such as LiDAR, cameras to continuously monitor the vehicle’s surroundings and provide accurate data for decision-making.

# Research Design

## Automated Navigation system

### Supervised Learning

We will use the **RoutePlanner** in the CARLA simulator to plan a route for the car to follow from spawn point A to destination point B.

We will divide the path into multiple points to define where the car has to apply **steering angle** to make a turn at a lane intersection.

**Buiding and Training**

A **Convolutional Neural Network (CNN)** model will be built for this task.

The input to the CNN will be:

* The **segmentation image** captured from the front-view camera of the vehicle.

**A road with palm trees and buildings

Description automatically generated**A road with buildings in the background

Description automatically generated

Figure 1: Image Segmentation taken from front-view camera sensors in CARLA simulator.

* The **next** **point**, which the car needs to navigate toward, is provided by CARLA’s ground truth path.

A car on the road

Description automatically generated

Figure 2: Generated waypoints from RoutePlanner

The output of the model will be:

* The predicted **steering angle**, required for the car to adjust its direction and move toward the next waypoint.

**Testing and Validation**

The trained model will be tested in the CARLA environment to ensure that it effectively predicts steering angles and guides the car to the desired waypoints.

The performance will be tested by analyzing the accuracy of the steering predictions as well as the vehicle's ability to arrive at its destination without collision or entering a different lane.

### Reinforcement Learning

A diagram of a robot

Description automatically generated

Figure 3: [Classic “agent-environment loop"](https://www.gymlibrary.dev/content/basic_usage/)

**Agent**: The learner or decision-maker.

During training, the agent's input consists of observations and the rewards or penalties that the environment returns. After then, it will change its behavior to maximize the cumulative reward while minimizing the penalty.

In this research, I will use the PPO algorithm in Stable Baselines 3 (SB3), a set of reliable implementations of reinforcement learning algorithms in PyTorch.

**Environment**: Everything the agent interacts with.

In this research, I will use Gym, a standard API for reinforcement learning, and a diverse collection of reference environments.

The **customized reward function** in the environment might be diverse for various conditions, such as:

* If the car has a collision while driving or not → reset the environment.
* Whether the car has any lane invasion or not → reset the environment.
* If the distance traveled during an espisole is less than a certain number until it reaches the finished condition (did it crash too soon?).
* Otherwise, if it travels a long distance without triggering any penalties while driving, it will earn a certain number of reward point.

The **action space** in the environment should be continuous, covering the maximum steering angle from left to right of the car, in CARLA, it in a range of 0 to 8 continuously.

## Real-time collision avoidance system

### Spacial Consciousness

To solve this task, I will use the LiDAR sensor to collect spatial surrounding data, and from that cloud point data, I will estimate and reconstruct the 3D environment of the surrounding space.

Then I'll use DBSCAN - Density-Based Spatial Clustering of Applications with Noise, which is offered by sklearn, to discover core samples with high density, expand clusters from each group of cloud points, and reduce noise cloud points.

I make a bounding box and a centroid for each cluster:

* Defined a bounding box for vehicle classification task - detect many types of objects, including trucks, cars, motorcycles, bicycles, and pedestrians.
* For vehicle tracking, I use KDTree, a SciPy data structure for efficient nearest-neighbor searches (centroid).

### Movement Prediction

Once we are aware of the surrounding space, we can determine the direction in which other vehicles are moving by tracking their centroid over a certain number of frames.

A group of colorful objects

Description automatically generatedBy calculating the distance between the previous and current centroid positions of each object, and dividing this distance by the time interval (1 / frame rate of the LiDAR sensor), we can estimate the speed of each vehicle.

*LiDAR sensor is attached on the top of the car.*

*yellow: motocycles*

*blue: cars*

*brown: trucks*

Figure 4: Vehicle Classification and Movement Prediction

# Research Instrumentation

* **Simulation, Collecting Data and Testing**
  + **CARLA Simulator**: Simulates real-world driving conditions (urban environments, traffic) for testing the AI-based navigation and anti-collision system.
  + **LiDAR Sensor:** Collects 3D point cloud data of surrounding spatial environment (vehicles, obstacles) around the car.
* **Artificial Intelligence and Machine Learning**
  + **Tensoflow - Convolutional Neural Network (CNN)**: Building custom model to predict steering angles, guiding the vehicle towards waypoints.
  + **Stable Baselines3 (SB3)** - **Proximal Policy Optimization:** AI agent to make decisions for automated navigation using reinforcement learning.
  + **Sklearn - DBSCAN (Density-Based Spatial Clustering)**: Identifies clusters in point cloud data to detect vehicles and distinguish them from noise.
  + **Spicy - KDTree (K-Dimensional Tree)**: Tracks vehicles over time by efficiently matching current and previous centroids for movement estimation.
* **Visualization and Data Processing**
  + **Numpy:** Efficient numerical computations, handling arrays, matrices, and performing mathematical operations on data such as calculating distances between centroids.
  + **Open3d:** Visualizes point cloud data, detected clusters, and object trajectories in 3D space.
  + **Opencv:** Provides tools for image processing, including detecting and tracking objects in video frames and integrating camera data for vehicle navigation.

# References