Creating responsible AI for detecting aggressive driving in traffic scenarios

Thesis to be submitted in partial fulfillment of the requirements for the degree

of

Master of Technology in Computer Science and Engineering

by

Sonu Kumar
22CS60R73

Under the guidance of

Prof. Soumyajit Dey

COMPUTER SCIENCE ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

Department of COMPUTER SCIENCE ENGINEERING
Indian Institute of Technology,
Kharagpur
India - 721302

CERTIFICATE

This is to certify that we have examined the thesis entitled **Creating responsible AI for detecting aggressive driving in traffic scenarios**, submitted by **Sonu Kumar**(Roll Number: *22CS60R73*) a postgraduate student of **Department of Computer science and Engineering,** in partial fulfillment for the award of degree of Master of Technology in Computer Science and Engineering. We hereby accord our approval of it as a study carried out and presented in a manner required for its acceptance in partial fulfillment for the Post Graduate Degree for which it has been submitted. The thesis has fulfilled all the requirements as per the regulations of the Institute and has reached the standard needed for submission.

Prof. Soumyajit Dey

Department of Computer science and Engineering,
Indian Institute of Technology,
Kharagpur

Place: Kharagpur
17 April 2024

ACKNOWLEDGEMENTS

The completion of this research paper has been made possible through the support, guidance, and contributions of various individuals and organizations, to whom I extend our sincere gratitude. First and foremost, I would like to express my deepest appreciation to my supervisor Soumyajit Dey, whose invaluable insights, encouragement, and mentorship have been instrumental in shaping the direction and outcomes of this study. Their expertise and unwavering support have served as a constant source of inspiration throughout the research process.

I extend my sincere gratitude to my PhD Guide, Ashiqur Rahaman Molla, for his consistent guidance and support throughout this thesis. His expertise and willingness to assist have been invaluable, providing clarity on various aspects of the research and offering valuable resources whenever needed. His dedication and encouragement have been instrumental in overcoming challenges and achieving milestones in this study.

Sonu Kumar
IIT Kharagpur
Date: 17 April 2024

ABSTRACT

The advancement of autonomous vehicles relies heavily on accurate simulation environments and robust algorithms for real-time decision-making. This thesis presents a comprehensive study on the detection and assessment of overtaking maneuvers in autonomous driving scenarios. The research combines scene generation using Scenic and the CARLA simulator with sensor data collection, trajectory prediction using LSTM models, and overtaking detection algorithms. The study begins by reviewing the literature on scene generation, sensor data collection, trajectory prediction, and overtaking detection. It highlights the importance of each component in enabling safe and efficient autonomous driving systems. Following the literature review, the methodology section describes the setup for scene generation, sensor configuration, data collection, preprocessing, LSTM model training, and overtaking detection algorithm development.

Experiments are conducted in a simulated environment to evaluate the performance of the proposed approach. Results demonstrate the effectiveness of the trajectory prediction model in forecasting the future movements of surrounding vehicles. Furthermore, the thesis introduces a novel methodology for quantifying aggressive overtaking maneuvers. By analyzing relative acceleration, velocity, and distance, the Time to Overtake (TTO) metric is computed to assess the severity of overtaking maneuvers. A decision tree model is explored to enhance overtaking scenario prediction accuracy.

In conclusion, this thesis contributes a robust AI-based solution for enhancing road safety by identifying and mitigating dangerous overtaking incidents using LSTM recurrent neural networks. The findings underscore the potential of AI-driven technologies to significantly reduce the occurrence of accidents caused by reckless driving behaviors. As road safety re- mains a shared societal concern, this research paves the way for further advancements in AI applications within the domain of traffic management and accident prevention.

Contents

1 Introduction **1**

2 Literature Review: **5**

2.1 Optical Flow Analysis 7

2.1.1 Pre-Processing and Segmentation 7

2.1.2 Convolutional Neural Network(CNN) 8

2.1.3 Repetitive Pattern Removal 8

2.1.4 Tracking and Behavior Detection 8

2.1.5 Limitations: 9

2.2 Deep Convolutional Network-based detection 9

2.2.1 Limitations: 11

3 Methodology: **12**

3.1 Scene Generation and Simulation: 13

3.2 Installation and Configuration of Sensors: 15

3.3 Data Collection Process: 18

3.4 Preprocessing of Sensor Data: 19

3.5 Trajectory Prediction Using LSTM: 20

3.5.1 Training of LSTM Model: 22

3.5.2 Validation: 23

3.5.3 Model Integration: 25

3.6 Overtake detection: 25

3.6.1 Determining Overtaking Scenarios 26

3.6.2 Quantifying Aggressive Overtaking 26

4 Results **29**

5 Conclusion **32**

6 Future Work **33**

Bibliography **34**

List of Figures

2.1 The system flow chart of optical flow analysis 7
2.2 View with rear camera 10

3.1 Flow chart of steps involved in the methodology 13
3.2 overtaking scenes at different timestamps 15
3.3 Segmented vehicle for tracking 18
3.4 Architecture of Lstm-Rnn based motion predictor 21
3.5 Comparision graph for actual and predicted x and y coordinate of the target vehicle. 24
3.6 Flow chart for aggressive overtake quantification 28

4.1 Predicted vs Actual value of different parameters of succeeding vehicle over 100 timesteps 30

Chapter 1

Introduction

In recent years, the advancement of autonomous driving technologies has promised a revolution in transportation, offering the potential to significantly improve road safety, efficiency, and accessibility. However, the realization of this vision hinges on the development of robust and responsible AI systems capable of navigating complex real-world environments with human-like proficiency and safety standards. One critical aspect of autonomous driving is the ability to detect and respond to aggressive driving behaviors, such as hazardous overtaking maneuvers, in order to mitigate the risk of accidents and ensure the safety of all road users.

The objective of this report is to present a comprehensive framework for detecting aggressive overtaking in traffic scenarios using responsible AI techniques. The proposed framework integrates state-of-the-art technologies and methodologies from various domains, including scene generation, sensor fusion, trajectory prediction, and machine learning, to create a holistic solution for enhancing road safety in autonomous driving environments.

Motivation

The motivation behind this research stems from the pressing need to address the challenges posed by aggressive driving behaviors in the context of autonomous vehicles. While significant progress has been made in the development of self-driving systems, they still face inherent limitations in accurately perceiving and responding to dynamic and unpredictable traffic scenarios, particularly those involving aggressive maneuvers such as overtaking. According to the National Highway Traffic Safety Administration (NHTSA), aggressive driving contributes to a significant portion of traffic accidents, highlighting the urgency of developing proactive measures to detect and mitigate such behaviors in autonomous driving systems.

Moreover, as autonomous vehicles become increasingly prevalent on public roads, it is imperative to ensure their ability to coexist harmoniously with human-driven vehicles, which may exhibit a wide range of driving styles and behaviors. By equipping autonomous systems with the capability to identify and appropriately respond to aggressive overtaking maneuvers, we can not only enhance the safety of autonomous vehicles but also foster greater acceptance and trust among human drivers and pedestrians.

Problem Statement: Despite the promising advancements in autonomous vehicle technology, the detection and assessment of overtaking maneuvers remain significant challenges. Overtaking maneuvers, when executed unsafely, can pose serious risks to road users, leading to accidents and fatalities. Current approaches to detecting and assessing overtaking maneuvers often lack the accuracy and real-time capabilities required for effective autonomous driving systems. Therefore, there is a pressing need to develop innovative methodologies that can accurately identify and assess overtaking maneuvers to enhance road safety in autonomous driving scenarios.

Research Objectives: The scope of this research encompasses the development and evaluation of a responsible AI framework specifically tailored for detecting aggressive overtaking in traffic scenarios encountered by autonomous vehicles. The key objectives of the study include:

1. Investigate the integration of scene generation tools, such as Scenic, and simulation platforms like the CARLA simulator, to create realistic driving environments for autonomous vehicle testing.

2. Explore the use of advanced sensor technologies, including radar and segmentation cameras, to collect real-time data on surrounding traffic conditions.

3. Develop LSTM-based trajectory prediction models to forecast the future movements of surrounding vehicles accurately.

4. Design overtaking detection algorithms capable of identifying hazardous overtaking maneuvers based on predefined criteria and real-time sensor data.

5. Introduce a novel methodology for quantifying the aggressiveness of overtaking maneuvers using metrics such as Time to Overtake (TTO) and collision probability.

6. Evaluate the performance of the proposed framework through extensive experiments conducted in simulated environments, demonstrating its effectiveness in enhancing road safety in autonomous driving scenarios.

This research focuses on the detection and assessment of overtaking maneuvers in autonomous driving scenarios using a combination of scene generation, sensor data collection, LSTM-based trajectory prediction, and overtaking detection algorithms. The study primarily investigates the effectiveness of the proposed framework in simulated environments, with potential applications in real-world autonomous driving systems. The successful development of a robust framework for detecting and assessing overtaking maneuvers has significant implications for the advancement of autonomous vehicle technology. By enhancing road safety and reducing the risk of accidents caused by reckless driving behaviors, the proposed framework can accelerate the widespread adoption of autonomous vehicles, leading to safer and more efficient transportation systems globally.

Preference of AI over a predefined set of rules

Well-defined rules can be effective in certain scenarios, but the complexity and dynamic nature of traffic scenarios, especially when it comes to overtaking maneuvers, may require a more adaptive and data-driven approach that AI can provide. There can be various reasons that lead to the requirement of adaptive learning such as:

- Complexity of Traffic Scenarios: Traffic scenarios can be highly dynamic and unpredictable, making it challenging to capture all possible situations with predefined rules. AI, particularly machine learning algorithms like LSTM, can analyze vast amounts of data to detect patterns and make predictions in real time, adapting to changing conditions more effectively than rule-based systems.

- Nuanced Driving Behaviors: Overtaking maneuvers can involve subtle variations in behavior depending on factors such as vehicle speed, distance, and surrounding traffic conditions. AI algorithms can learn from data to recognize these nuanced behaviors and make informed decisions, whereas rule-based systems may struggle to account for all possible variations.

- Scalability and Generalization: AI models trained on diverse datasets can generalize well to unseen scenarios, whereas rule-based systems may require extensive manual tuning and maintenance as the complexity of scenarios increases. AI offers scalability by learning from data, allowing for more robust and adaptable solutions as traffic scenarios evolve over time.

- Handling Uncertainty: AI models can handle uncertainty inherent in real-world data, such as sensor noise or variability in driver behavior, by learning statistical relationships and making probabilistic predictions. This ability to quantify uncertainty can enhance the reliability and safety of autonomous driving systems in challenging environments.

- Continuous Learning: AI systems can be designed to continuously learn and improve over time as more data becomes available, allowing for iterative refinement of algorithms and adaptation to new challenges. This capability of continuous learning is particularly valuable in dynamic domains like autonomous driving.

Organization of the Report

This thesis is organized into several sections, each focusing on different aspects of the research. Chapter 2 provides a comprehensive review of the literature related to scene generation, sensor technologies, trajectory prediction, and overtaking detection. Chapter 3 outlines the methodology employed in the research, including scene generation setup, sensor configuration, data collection procedures, LSTM model training, and overtaking detection algorithm development. Chapter 4 presents the experimental results and evaluates the performance of the proposed framework. Chapter 5 concludes the thesis by summarizing the key findings and contributions of the research. Finally, Chapter 6 discusses the findings, implications, limitations, and future research directions.

Chapter 2

Literature Review:

This section offers a review of existing research on responsible AI systems designed to detect aggressive driving behaviors in traffic scenarios. It investigates various methodologies, including the collection of real-world video data and the generation of synthetic traffic scenarios. The machine learning algorithms and computer vision techniques employed are discussed, along with ethical considerations during AI algorithm development. The goal is to gain insights into diverse approaches and methodologies for building responsible AI systems to detect aggressive driving. This analysis serves as a foundation for our new methodology, which aims to combine the strengths of previous works while addressing their limitations to create a more accurate and reliable system for detecting and preventing aggressive driving. Importantly, the new methodology incorporates ethical considerations to ensure responsible AI use in transportation, ultimately improving road safety.

Scene Generation

Scene generation plays a crucial role in creating realistic environments for testing autonomous vehicles. Tools such as Scenic(1) (a probabilistic programming language for scene generation) offer the capability to generate complex driving scenarios with customizable parameters, enabling researchers to simulate various traffic conditions and scenarios. Previous studies have utilized scene-generation tools to assess the performance of autonomous driving systems in challenging situations, including intersections, highway merges, and urban environments (2; 3). These simulations provide valuable insights into the behavior of autonomous vehicles and their interactions with other road users.

Sensor Technologies

Advanced sensor technologies are essential for providing autonomous vehicles with real-time data on their surroundings. Radar and segmentation cameras are commonly employed to detect and track nearby vehicles, pedestrians, and obstacles. Research in this area focuses on improving sensor accuracy, range, and resolution to enhance perception capabilities in diverse environmental conditions. Additionally, fusion techniques, such as sensor fusion and multi-modal perception, have been proposed to integrate data from multiple sensors for more robust and reliable object detection and tracking.

Trajectory Prediction

Accurate trajectory prediction is critical for anticipating the future movements of surrounding vehicles and planning safe and efficient driving maneuvers. Long Short-Term Memory (LSTM) networks have emerged as a popular approach for trajectory prediction due to their ability to capture temporal dependencies in sequential data(4). These models are trained on historical vehicle trajectories to forecast their future trajectories, enabling autonomous vehicles to proactively respond to potential collision scenarios and navigate complex traffic situations.

Overtaking Detection

Detecting aggressive overtaking maneuvers is essential for ensuring road safety in autonomous driving scenarios. Overtaking maneuvers are typically classified based on their execution relative to other vehicles, such as lane changes, lane splitting, etc. Recent studies have proposed various machine learning-based approaches for overtaking detection, leveraging features such as vehicle speed, acceleration, relative position, and surrounding traffic conditions(5). Additionally, metrics such as Time to Overtake (TTO) and collision probability are used to quantify the aggressiveness of overtaking maneuvers and assess their potential risk to road safety.

2.1 Optical Flow Analysis

The optical flow-based detection method reviewed here presents an innovative approach to real-time identification of aggressive driving behavior in traffic scenarios (6). It leverages motion cues and optical flow information to categorize driving behavior, offering a potential solution to road safety challenges.

This method comprises four key steps: pre-processing and segmentation, convolutional neural network (CNN), repetitive pattern removal, and tracking and behavior detection(5). Figure 2.1 shows the flow chart of different modules involved in behavior detection using optical flow.

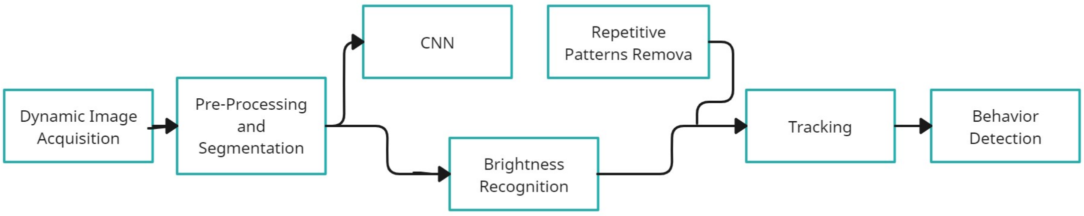

Figure 2.1: The system flow chart of optical flow analysis

2.1.1 Pre-Processing and Segmentation

To improve operational efficiency, we reduce the size of the original image by excluding unnecessary elements like the sky, buildings, and traffic signs in the upper part of the image. Subsequently, we identify the region of interest (ROI) before applying the image segmentation algorithm. Motion cues are acquired by utilizing tracking points, where the settings for these tracking points are of significant importance.

We employ a predetermined quantity of tracking points in fixed positions. More precisely, we establish one tracking point for every 10x10 pixel region within the ROI. Subsequently, we apply the pyramid model of the Lucas-Kanade optical flow to compute optical flow data for these tracking points. When a vehicle moves forward, there is a notable presence of optical flow surrounding it. Analyzing the optical flow of a feature point enables us to categorize it into one of five groups, considering both its motion direction and magnitude. These categories include (i) road and sky, (ii) landmarking, (iii) overtaking vehicle, (iv) object moving further away, and (v) uncertain region. For instance, when the optical flow of a feature point exhibits minimal motion, it may be categorized as belonging to the road or sky areas. Conversely, if the optical flow primarily shows significant motion in the y-direction and converges toward the vanishing point (where parallel lines appear to meet), we assign the point to the landmarking region. In cases where the optical flow prominently moves in the x-direction away from the vanishing point, it signifies the presence of an overtaking vehicle. Conversely, if the optical flow displays substantial x-direction motion toward the vanishing point, it indicates an object moving into the distance.

2.1.2 Convolutional Neural Network(CNN)

Initially, a convolutional neural network is trained using an image dataset that includes motorcycle images, the front and rear views of cars, repetitive patterns, background scenes, and lane markings.

2.1.3 Repetitive Pattern Removal

In specific scenarios, the segmentation algorithm may generate inaccurate outcomes, particularly when there is a vehicle in the opposing lane, causing confusion for the algorithm. This issue arises because the optical flow tracking feature points coincide with repetitive patterns, while the convolutional neural network detects the vehicle in the opposing lane. Consequently, this stage aims to detect repetitive patterns within the segmented area by employing multiple feature points.

2.1.4 Tracking and Behavior Detection

Following the use of CNN for segmentation and the elimination of repetitive patterns, we commence object tracking until the object either disappears or overtaking ceases. Leveraging the Lucas-Kanade optical flow, a technique that offers a means to estimate the movement of distinctive features across consecutive images, we maintain continuous object tracking. This aids in the detection of object movement over an extended duration, accommodating variations in shape, size, and scale and offering a more comprehensive trajectory. Nevertheless, due to constraints such as low camera resolution and segmentation errors, the potential for false detections exists. To mitigate these false detections, we scrutinize whether the tracking direction remains apart from the vanishing point for an extended period. This validation step effectively eliminates erroneous detections.

2.1.5 Limitations:

While the optical flow-based detection method shows promise in identifying aggressive driving behavior, it is essential to consider its limitations:

Sensitivity to Environmental Factors: Optical flow analysis can be sensitive to environmental conditions such as lighting, weather, and road surface. Adverse conditions may affect the accuracy of motion cues.

Dependency on ROI Selection: The accuracy of this method relies on selecting an appropriate region of interest (ROI) within the image. The choice of ROI can impact the detection and categorization of driving behavior.

Challenges with Real-time Processing: Implementing optical flow analysis in real-time applications may pose computational challenges, especially in resource-constrained environments.

Limited to Motion-based Cues: This method primarily relies on motion-based cues, which may not capture all aspects of aggressive driving behavior.

In conclusion, the optical flow analysis in the system calculates the displacement of pixels between consecutive frames of the synthetic traffic videos, which can reveal the motion and direction of objects in the scene. The CNNs in the system are trained on a large dataset of annotated aggressive driving behaviors to classify the driving input as either aggressive or non-aggressive.

2.2 Deep Convolutional Network-based detection

In recent years, deep learning-based object detection(7) techniques have shown remarkable performance in various computer vision applications. In this study, we propose a deep convolutional network-based approach for vehicle safety that utilizes the rear camera installed in the vehicle to capture data. The captured data is then processed using object detection techniques based on deep learning like *YOLO* (8) to detect the vehicles. The detected vehicles are monitored and tracked to determine their speed and trajectory. This approach helps to prevent potential accidents by analyzing the approaching speed to determine potential hazards or anomalies, and statistical regression techniques are applied to assess the degree of risk for overtaking situations.

The incoming video frames undergo an object detection process using the You Only Look Once (YOLO) deep neural network, which generates a list of bounding boxes for detected vehicles. Object tracking is then applied to trace the trajectories of these vehicles. The trajectory represents a sequence of bounding boxes that correspond to the same vehicle across successive frames, encompassing frames in which the vehicle was not initially detected.

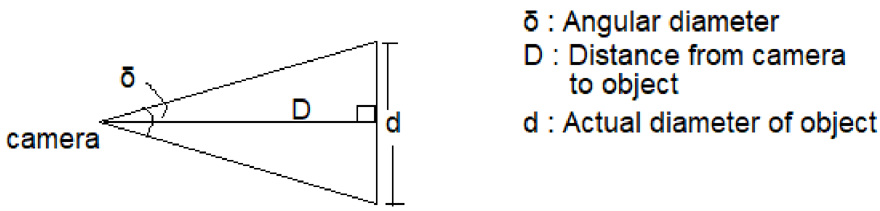

Figure 2.2: View with rear camera

Figure 2.2 represents the angular diameter of an object (also referred to as the apparent diameter), which can be expressed as a function of the actual diameter of the object, d (in meters), and the distance from the camera to the object, D (also in meters), as follows:

$$\tan(\delta/2) = \frac{d/2}{D} \tag{2.1}$$

where δ is the angular diameter in radians and arctan is the inverse tangent function.

When $D >> d$, i.e., when θ is very small, we can use the small angle approximation and simplify the equation to:

$$\delta = \frac{d}{D} \tag{2.2}$$

Assuming the object maintains a consistent velocity relative to the camera, we can write:

$$\delta = \frac{d}{e + vt} \tag{2.3}$$

$$\frac{1}{\delta} = \frac{e + vt}{d} \tag{2.4}$$

where e is the distance of the object at time $t = 0$ and v is the relative velocity of the vehicle in relation to the camera. At every moment a vehicle is detected, its angular diameter δ is estimated by taking the square root of the bounding box's area (in pixels) associated with the vehicle. The slope of the linear function of equation (2.4) can be computed, which represents the relative velocity v of the vehicle in relation to the camera, obtained through linear regression. Finally, a threshold is set on the relative velocity v to detect approaching vehicles that have an excessively high velocity as dangerous.

2.2.1 Limitations:

While the deep convolutional network-based approach offers robust vehicle detection and tracking capabilities, it is essential to be aware of its limitations:

Hardware and Processing Requirements: Implementing deep learning-based object detection can be computationally intensive, necessitating high-performance hardware for real-time processing. This can be a limitation in resource-limited environments.

Dependency on Data Quality: The effectiveness of deep learning models depends on the quality and diversity of the training data. Inadequate or biased training data may lead to detection inaccuracies.

Limited to Visual Data: This approach primarily relies on visual data from the rear camera, which may not capture other sensor data (e.g., audio, vehicle speed) that could enhance the accuracy of aggressive driving detection.

Chapter 3

Methodology:

After reviewing various research papers and exploring multiple approaches to determine the decisive factors for identifying dangerous overtaking maneuvers, we realized that relying solely on image tracking might not provide sufficient information due to potential conflicts with other vehicles present at the same timestamp. To address this limitation, we delved into the literature to explore alternatives based on kinematics data. One promising concept involves the development of an LSTM (Long Short-Term Memory) model(4; 9). This model will predict a vehicle's trajectory, considering parameters such as lateral and longitudinal location, velocity, and acceleration. The goal is to leverage this model to determine whether a vehicle is likely to overtake and, if so, whether the maneuver would be aggressive or smooth. So, the main focus of this paper is to develop an LSTM model that will help in trajectory prediction and based on the predicted trajectory a decision can be made regarding the overtaking behavior of the target vehicle. The approach combines the utilization of kinematics data and Long Short-Term Memory (LSTM) models(10).

In this section, we provide a comprehensive overview of the methodology adopted for the detection and assessment of overtaking maneuvers in autonomous driving scenarios. The methodology encompasses scene generation, sensor configuration, data collection, preprocessing, trajectory prediction using Long Short-Term Memory (LSTM) models, overtaking detection algorithms, and the quantification of aggressive overtaking maneuvers. Figure 3.1 describes the flow chart of the different modules involved in our methodology.

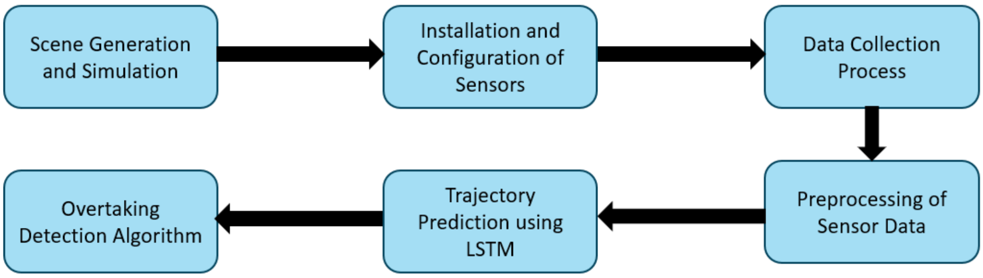

Figure 3.1: Flow chart of steps involved in the methodology

3.1 Scene Generation and Simulation:

In our research methodology, Scene Generation and Simulation play a pivotal role in creating dynamic and realistic driving environments for studying overtaking maneuvers in autonomous driving scenarios. We leverage the power of Scenic(1), a probabilistic programming language, in conjunction with the CARLA(11) simulator to generate diverse and challenging traffic scenarios.

We use Scenic for the scene generation as it ensures diversity in traffic generation and allows us to manipulate all parameters related to the scene, definitions of all actors, and their behavior such as:

- Parameter Variation: Scenic allows for the specification of a wide range of parameters, such as road layout, weather conditions, time of day, and traffic density. By varying these parameters across simulations, diverse scenarios can be generated.
- Randomization: Introducing randomization in scene generation can create variability in factors such as vehicle trajectories, pedestrian behavior, and road conditions. Randomizing elements within specified ranges adds unpredictability and diversity to the generated scenes.
- Scenario Generation Rules: Scenic allows users to define rules and constraints for scenario generation. By specifying different rules for road layouts, traffic patterns, and environmental conditions, diverse scenarios can be generated based on specific criteria and objectives.

- Iterative Refinement: Generating scenes iteratively and refining the parameters based on feedback and evaluation results can lead to the creation of more diverse and realistic scenarios over time. Continuous refinement based on observed patterns and emerging scenarios helps enhance diversity.

- Incorporating Edge Cases: Including edge cases and rare events in scenario generation adds complexity and diversity to the generated scenes. By considering outlier scenarios and extreme conditions, the generated scenes become more comprehensive and diverse, covering a wider range of potential real-world situations.

Scenic provides a flexible and intuitive platform for defining complex traffic scenarios with customizable parameters. Through Scenic, we specify the characteristics of the road network, including lane configurations, intersections, traffic density, and vehicle behaviors. By incorporating probabilistic elements into the scenario generation process, we can simulate a wide range of driving conditions, including scenarios where overtaking maneuvers occur. CARLA serves as our simulation platform, offering a high-fidelity 3D environment for visualizing and interacting with the generated scenarios. It provides realistic physics simulation, accurate vehicle dynamics, and sensor feedback, allowing us to observe and analyze overtaking maneuvers in a controlled setting. By interfacing Scenic with CARLA, we can seamlessly translate the scenario specifications into interactive simulations, enabling comprehensive studies of overtaking behaviors.

The integration of Scenic with CARLA is a pivotal aspect of our methodology. By seamlessly combining the capabilities of both tools, we can generate dynamic and realistic driving scenarios that are essential for evaluating the performance of our overtaking detection and assessment algorithms. This integration enables us to conduct thorough experimentation and analysis in a simulated environment before transitioning to real-world testing, thereby ensuring the robustness and reliability of our research findings. Figure 3.2 illustrates a simulated overtaking scenario generated using the Scenic code integrated with the CARLA simulator. In this scenario, two vehicles are depicted: the ego car and the overtaking vehicle. Initially, the overtaking vehicle follows its designated lane and approaches the ego car. As the overtaking vehicle reaches a predefined distance from the ego car, it begins to transition from the slower lane to the faster lane, indicative of an overtaking maneuver. Once the overtaking vehicle surpasses a threshold distance from the ego car, it returns to the slower lane. This dynamic scenario captures the complexity of real-world overtaking maneuvers and enables detailed analysis of vehicle interactions and behaviors.

Figure 3.2: overtaking scenes at different timestamps

3.2 Installation and Configuration of Sensors:

The successful deployment and configuration of sensors are essential steps in creating a robust simulation environment for capturing the data of overtaking the vehicle accurately. In this section, we discuss the installation and configuration process for two key sensors: the Radar sensor and the Semantic Segmentation Camera.

Radar Sensor: The Radar sensor, identified by the blueprint sensor.other.radar, is instrumental in capturing dynamic elements within the driving environment and evaluating their movement characteristics. To install and configure the Radar sensor:

- Blueprint Selection: Choose the appropriate blueprint for the Radar sensor, denoted as sensor.other.radar, to instantiate it within the simulation environment.
- Blueprint Attributes Configuration: Configure the following attributes of the Radar sensor blueprint to tailor its functionality:
- Horizontal FOV: Set the horizontal field of view in degrees to determine the breadth of the sensor's coverage.
- Points per Second: Define the number of points generated by all lasers per second to influence the sensor's data capture rate.
- Range: Specify the maximum distance for raycasting measurements in meters to ensure adequate coverage of the surrounding environment.

- Sensor Tick: Determine the simulation time between sensor captures to facilitate synchronized data acquisition.
- Vertical FOV: Specifies the vertical field of view in degrees, contributing to the sensor's comprehensive observation capabilities.

Output Attributes: The sensor output is provided as a carla.RadarMeasurement object contains an array of carla.RadarDetection elements, each representing a detected point. These detections include attributes such as altitude, azimuth, depth, and velocity, offering detailed information about the observed objects' positions and movements.

To process the raw data from the Radar sensor, it can be easily converted into a manageable format using libraries like numpy. By extracting the necessary attributes from the carla.RadarDetection objects, valuable insights can be gained regarding the surrounding objects' dynamics and behavior.

However, due to its broad scope, the radar may capture a multitude of objects beyond the intended target. To refine the data and focus solely on the succeeding vehicle, an additional layer of segmentation is required. Table 3.1 shows the overview of data collected by only radar where we can see that at the same timestamp, it has captured the data of all static and dynamic actors whatever was present in its scope such as vehicle and traffic lights.

Time	Velocity	Altitude	Azimuth	Depth
1712471925.0503125	-0.55362797	-0.7472633	-0.006592939	88.77638
1712471925.0503125	-3.7135756	-0.1753399	-0.0010306409	41.875984
1712471925.0503125	-2.8725607	-0.38246295	0.029639982	42.65673
1712471925.1456518	-0.29892394	0.61102396	-0.0035073885	13.526114
1712471925.1456518	-1.6298248	-0.0190182	-0.008828517	17.756086
1712471925.1456518	-0.9372402	0.33377877	-0.0042908294	60.34778
1712471925.1456518	-1.5722092	0.03256105	0.00045648686	36.38689
1712471925.1456518	-2.3178613	-0.6685263	-0.0132649485	56.59538

Table 3.1: Radar Sensor Data

Semantic Segmentation Camera: To address the need for focused data collection, a Semantic Segmentation Camera, identified by the blueprint sensor.camera.semantic_segmentation, is employed. This specialized camera classifies every object in sight based on predefined tags, providing a detailed understanding of the scene composition. By segregating elements such as vehicles, pedestrians, and infrastructure, the Semantic Segmentation Camera facilitates the isolation of the succeeding vehicle within the driving environment.

- Blueprint Selection: Select the appropriate blueprint for the Semantic Segmentation Camera, identified as sensor.camera.semantic_segmentation, to instantiate it within the simulation environment.
- Blueprint Attributes Configuration: Configure the following attributes of the Semantic Segmentation Camera blueprint to customize its functionality:
- Resolution: Specify the resolution of the captured images for semantic segmentation to influence the image quality and detail.
- Field of View (FOV): Set the horizontal and vertical field of view angles to shape the camera's observation perspective.
- Sensor Tick: Set the simulation time between sensor captures to ensure synchronized data acquisition with other simulation components.

Output Attributes: The output of the Semantic Segmentation Camera, provided as a Carla. The image object contains pixel data with tag information encoded in the red channel. Each pixel's red value corresponds to a specific object tag, enabling the classification of scene elements into distinct categories. By utilizing tools like the CityScapesPalette in carla.ColorConverter, the raw image data can be processed to visualize the semantic segmentation results effectively.

The Semantic Segmentation Camera's output includes a range of tags representing various scene elements, such as roads, sidewalks, buildings, vehicles, pedestrians, and more. These tags provide valuable contextual information for analyzing the driving environment and detecting relevant objects and structures. Figure 3.3 shows the image captured by the Semantic Segmentation Camera where only the succeeding vehicle has been focused with the help of segregation using the Convolutional Neural Network(CNN) technique.

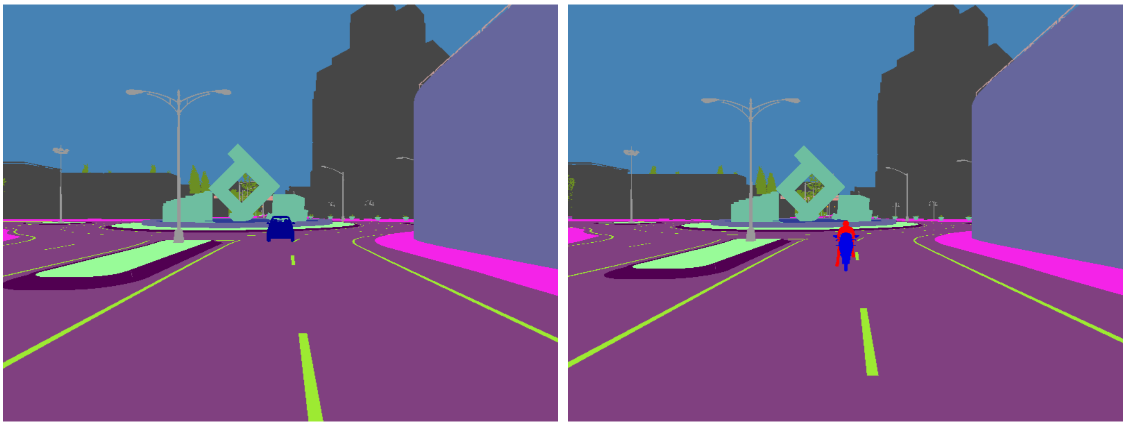

Figure 3.3: Segmented vehicle for tracking

3.3 Data Collection Process:

The data collection process involves the systematic gathering of relevant information from various sensors and sources to build a comprehensive understanding of the driving environment. In the context of overtaking maneuver detection, the data collection process is tailored to capture essential parameters related to the movement and behavior of vehicles on the road.

To ensure accurate and focused data collection, we combine the output of both the Radar sensor and Semantic Segmentation Camera according to timestamp. This integration enables the extraction of targeted data pertaining specifically to the succeeding vehicle. By leveraging the comprehensive segmentation provided by the camera alongside the dynamic object detection capabilities of the radar, a refined dataset is obtained, focusing exclusively on the relevant vehicle of interest. This integration ensures precise and actionable insights into the behavior and movements of the succeeding vehicle, essential for autonomous driving systems' decision-making processes. Table 3.2 shows the overview of combined data collected by both sensors where we focused on the succeeding vehicle only, here the data captured by the sensors are merged based on timestamp.

time	velocity	depth	center_x	center_y
1712471925.0503125	-0.55362797	-0.7472633	-0.006592939	88.77638
1712471925.0503125	-3.7135756	-0.1753399	-0.0010306409	41.875984
1712471925.0503125	-2.8725607	-0.38246295	0.029639982	42.65673
1712471925.1456518	-0.29892394	0.61102396	-0.0035073885	13.526114
1712471925.1456518	-1.6298248	-0.0190182	-0.008828517	17.756086
1712471925.1456518	-0.9372402	0.33377877	-0.0042908294	60.34778
1712471925.1456518	-1.5722092	0.03256105	0.00045648686	36.38689
1712471925.1456518	-2.3178613	-0.6685263	-0.0132649485	56.59538

Table 3.2: Combined data collected by the Sensors

3.4 Preprocessing of Sensor Data:

For training our LSTM model, we meticulously curated a dataset within the Carla driving simulator by deploying a vehicle and capturing a comprehensive set of essential features at distinct time intervals across a spectrum of dynamic traffic scenarios. The process entailed the use of different maps of Carla for different traffic scenarios within Carla's realistic simulation environment, enabling us to precisely replicate real-world traffic situations. We fitted the subject vehicle to record vital data, including local and global vehicle coordinates, vehicle velocity, and vehicle acceleration. As the simulation unfolded, we systematically collected data over multiple timesteps, ensuring the dataset accurately represented the nuances of real traffic behavior. This dynamic approach allowed us to generate a dataset with diverse driving scenarios, each reflecting the intricacies of genuine road conditions, and serve as a robust foundation for training our LSTM model.

In this work, we primarily utilized the dataset generated in Carla for training and vehicle trajectory dataset sourced from the NGSIM (Next Generation Simulation) project (12) for validation. This dataset stands out due to its richness in parameters and its relevance to real-world traffic scenarios. The collected data covers a wide range of driving situations, from regular traffic flow to more complex scenarios with lane changes and overtaking maneuvers. To ensure data quality and reliability, a series of preprocessing steps were undertaken. These steps aimed to remove errors and reduce noise in the dataset. In particular, we focused on the following aspects:

Data Cleaning: Any obvious errors or outliers in the collected data were identified and addressed. This involved the detection and correction of data points that deviated significantly from expected values, ensuring that the dataset was free from erroneous entries.

Noise Reduction: Noise in the data, arising from sensor inaccuracies or environmental factors, can impact the model's performance. Techniques such as smoothing and filtering were applied to mitigate the effects of noise, resulting in a cleaner and more reliable dataset.

Dataset Split: The dataset was divided into two main parts: one for training and the other for validation. This partitioning allowed us to train the model on a substantial portion of the data and evaluate its performance on an independent set to assess generalization capabilities.

The trajectory data for each vehicle consists of a sequence of observations over consecutive time steps, typically with a time interval between each data point. A sample excerpt of the dataset, showing the local coordinates (x and y), vehicle velocity (vel), and vehicle acceleration (acc) for a target vehicle over the last 5 contiguous timesteps, is presented in Table 3.3:

x_cord	y_cord	vel	acc
16.4671958	35.38042657	40	0
16.44659441	39.38160839	40.01234854	0.123485436
16.42599068	43.38154079	39.99985466	-0.124938868
16.40539161	47.38077972	39.99291978	-0.069348801
16.3848042	51.37988112	39.9915439	-0.013758733

Table 3.3: Trajectory data of target vehicle for last 5 timesteps

3.5 Trajectory Prediction Using LSTM:

The core of our research revolves around the application of Long Short-Term Memory (LSTM) neural networks. LSTM networks are well-suited for sequential data processing and have gained recognition for their effectiveness in capturing temporal dependencies, making them the ideal choice for our trajectory prediction mode.

Why LSTM-based RNN: Trajectory prediction in traffic scenarios is a complex task that requires models capable of capturing temporal dependencies and patterns in sequential data. LSTM-based RNNs have emerged as a powerful tool in this domain due to their unique architecture, which overcomes several limitations associated with traditional RNNs. LSTMs are designed to handle sequences of data where long-range dependencies exist. In the context of our research, vehicles in traffic exhibit intricate and often non-linear behaviors over time. LSTM networks excel at capturing these nuanced patterns and relationships by selectively remembering or forgetting information from previous time steps, effectively avoiding the vanishing gradient problem that plagues traditional RNNs.

Moreover, LSTM networks have been successfully applied in various fields, including speech recognition, natural language processing, and image recognition, where sequential data processing is crucial. Their adaptability to a wide range of sequential data types underscores their versatility and effectiveness. In the specific task of trajectory prediction, where we aim to estimate future states of surrounding vehicles based on historical data, LSTM-based RNNs provide an ideal framework. They can analyze the past states of vehicles, including their positions, velocities, and accelerations, and make accurate predictions about their future trajectories. By leveraging the power of LSTM-based RNNs, our methodology not only addresses the complexity of trajectory prediction in traffic scenarios but also enhances our ability to identify dangerous overtaking maneuvers proactively. This choice of model architecture aligns with our goal of creating a robust AI system for detecting aggressive driving behaviors and contributes to the overall success of our research. Figure 3.4 describes the architecture of the Lstm-Rnn-based motion predictor that we have used for trajectory prediction which includes data transformation for better learning of dependency.

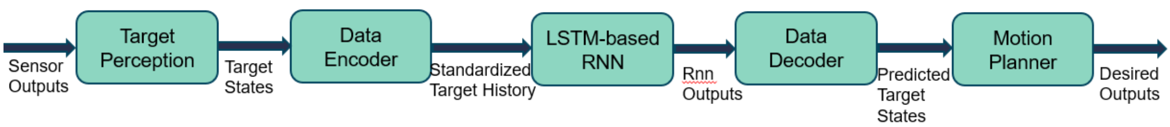

Figure 3.4: Architecture of Lstm-Rnn based motion predictor

3.5.1 Training of LSTM Model:

Training an LSTM model involves several key steps to ensure that the model learns effectively from the data and generalizes well to unseen instances. We outline each of these steps in detail to provide insight into our methodology and approach to model development.

- **Data Preprocessing:** Before diving into model development, the initial step is data preprocessing. The raw dataset, as described in the previous section, undergoes essential preprocessing steps to ensure that it is well-suited for training an LSTM-based model. The primary aspects of data preprocessing include:

  Standardization: The dataset is standardized using a StandardScaler. This process scales the data to have a mean of 0 and a standard deviation of 1. Standardization is a crucial step that ensures uniform scaling of all features, facilitating consistent training and prediction. It helps the model learn effectively from the input data.

- **Data Preparation:** The dataset is divided into input-output pairs, a fundamental requirement for training a supervised learning model like an LSTM. Specifically, we employ a sequence-to-sequence approach, where a specified number of past timesteps (n past) are used to predict a future timestep (n future). In our model configuration, n past is set to 5, signifying that the model takes into account the past 5 timesteps to make predictions about the subsequent timestep. This configuration forms the basis for generating training examples to teach the model the relationship between past observations and future outcomes.

- **LSTM Model Architecture:** Our model's architecture is centered on a sequence-to-sequence design, consisting of two main components: an encoder and a decoder. The encoder processes historical data, while the decoder is responsible for generating future predictions. These components work collaboratively to capture the temporal dependencies in the data, allowing the model to make accurate trajectory predictions. A critical feature of the model is its 64-dimensional LSTM latent space. This latent space is designed to capture the underlying patterns and relationships in the data, helping the model make predictions with a high degree of accuracy. By employing a latent space of this dimensionality, we enable the model to effectively represent complex temporal dependencies in the data.

- **Model Training:** Training the model is a pivotal stage in the development process. We utilize the Mean Squared Error (MSE) loss function as the optimization criterion, which quantifies the difference between the model's predictions and the actual ground truth. Additionally, we apply the Adam optimizer, a popular choice for training deep learning models, to adjust the model's weights and biases during training. To ensure that the model generalizes well and does not overfit the training data, early stopping is implemented during training. Early stopping helps prevent the model from learning noise or minor fluctuations in the data by monitoring performance on a validation set. When performance on the validation set no longer improves, training is halted, resulting in a model that provides reliable predictions on new, unseen data.

The detailed model development process outlined above forms the foundation of our research. By preprocessing the data, structuring it into input-output pairs, and designing a sequence-to-sequence LSTM model, we have established a framework for trajectory prediction. This framework is further enhanced by efficient training and early stopping mechanisms to ensure the model's accuracy and generalization capabilities.

3.5.2 Validation:

Evaluation of the model's performance is a critical step to assess the effectiveness of the LSTM-based trajectory prediction model in predicting vehicle trajectories accurately. The evaluation process is a multifaceted approach, comprising the following key steps:

- **Prediction Generation:** The first phase of the evaluation process involves using the trained LSTM model to generate predictions for future timesteps. This is accomplished through a systematic process implemented in the code. The code incorporates a loop that iteratively predicts the next timestep, updating the input sequence accordingly. This iterative prediction generation is fundamental in assessing the model's ability to capture temporal dependencies and produce accurate trajectory forecasts.

- **Inverse Scaling:** Once predictions are generated, it is imperative to transform these predictions from the standardized scale back to the original scale of the dataset. Standardization, as applied during data preprocessing, is crucial for training the model effectively. However, to ensure that the predictions are interpretable and relevant in the context of the original dataset, an inverse transformation is performed using the StandardScaler. This step is essential for making meaningful interpretations and decisions based on the predictions.

- **Visualization:** A key aspect of the validation process involves the use of visualizations. Visual representations enable a direct and intuitive assessment of the model's accuracy in predicting vehicle trajectories. The code includes visualizations that compare the predicted values with the actual values for key parameters such as 'x cord,' 'y cord,' and 'velocity.' These visualizations serve as a valuable tool for assessing the model's predictive accuracy and identifying any discrepancies between the predictions and the actual observations.

  Figure 3.5 below shows the deviation of the predicted value from the actual value of positions over the next 100 timesteps. We can observe these graphs depict predictions based on the last 5 timesteps. By visually comparing these predictions with the actual trajectory data, we gain insights into the model's performance and its ability to capture the complex and dynamic nature of vehicle movements.

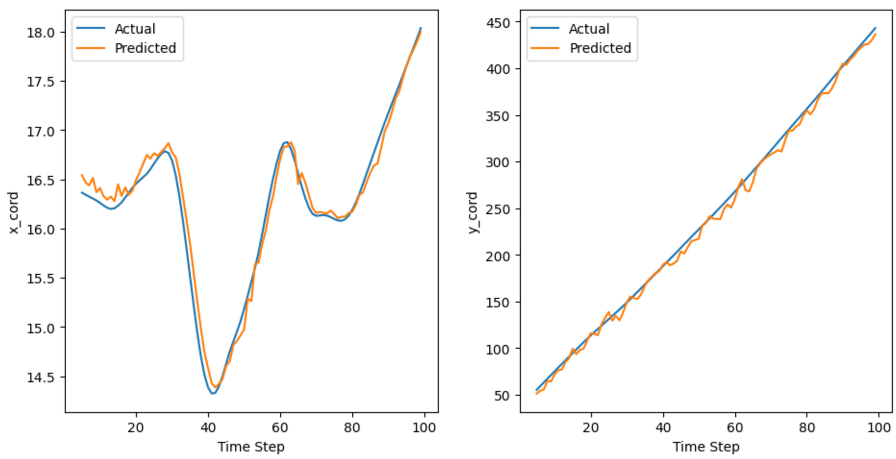

Figure 3.5: Comparision graph for actual and predicted x and y coordinate of the target vehicle.

3.5.3 Model Integration:

The integration of the LSTM-based trajectory prediction model into the subject vehicle plays a pivotal role in realizing the goal of identifying dangerous overtaking maneuvers. This integration process facilitates the generation of future trajectory data based on the trajectory data of the target vehicle, captured using onboard sensors. The subsequent future trajectory data becomes a vital component in the overtake detection process.

The model integration phase involves the following key steps:

- **Data Input:** Trajectory data of the target vehicle is captured using an array of sensors installed on the subject vehicle. These sensors provide critical information about the target vehicle's movements, including local and global coordinates, vehicle velocity, and vehicle acceleration.

- **Model Execution:** The pre-trained LSTM-based model is executed by passing the captured trajectory data as input. The model processes this input data to generate predictions for the future trajectory of the target vehicle. The model's ability to capture temporal dependencies and complex patterns in the data is leveraged to make accurate predictions regarding the target vehicle's future path.

- **Future Trajectory Data:** The output of the model execution is the future trajectory data of the target vehicle. This data includes predicted local and global coordinates, vehicle velocity, and acceleration, among other parameters. The predicted trajectory provides insights into how the target vehicle is likely to maneuver in the near future.

3.6 Overtake detection:

The future trajectory data is a vital component in the overtake detection process. The model-generated trajectory predictions are analyzed to determine whether the target vehicle is likely to initiate an overtaking maneuver and, if so, whether the maneuver is anticipated to be aggressive or smooth. This decision is based on the model's analysis of the future trajectory data, considering factors such as lateral and longitudinal movements, relative speeds, and accelerations.

3.6.1 Determining Overtaking Scenarios

To identify overtaking scenarios, specific conditions are considered. For instance, if the distance between the ego vehicle and the target vehicle falls below a defined threshold, the difference in velocity exceeds a certain limit, and the lateral distance of the target vehicle approaches adjacent lanes, it is likely to change lanes.

Incorporating Decision Tree Models:

To enhance our ability to predict overtaking scenarios, we have explored the possibility of implementing a separate model, potentially employing a decision tree approach. This model would utilize parameters such as relative velocity and position to predict the intentions of the target vehicle. By considering these additional factors and their interplay, we aim to refine our overtaking scenario detection and increase its accuracy.

By combining these conditions and, if necessary, leveraging decision tree models, we aim to develop a robust framework for identifying overtaking scenarios accurately. This methodology enables us to proactively recognize situations where overtaking is likely, providing valuable insights for subsequent analysis and evaluation.

3.6.2 Quantifying Aggressive Overtaking

In the context of identifying aggressive overtaking, we have devised a comprehensive method to assess the severity of overtaking maneuvers. Central to this methodology is the computation of the Time to Overtake (TTO), a critical metric that aids in determining the nature of overtaking maneuvers. Given that we have access to relative acceleration (a_r), relative velocity (v_r), and relative distance (d_r) (predicted using the LSTM model), we can calculate the required time using the following equations:

$$d_r = v_r + \frac{1}{2}a_r t^2 \tag{3.1}$$

$$v_r^2 = u_r^2 + 2a_r d_r \tag{3.2}$$

Here's what each variable represents:

- d_r denotes the relative distance between the ego vehicle and the target vehicle.
- v_r represents the relative velocity between the ego vehicle and the target vehicle.

- a_r signifies the relative acceleration between the ego vehicle and the target vehicle.
- t is the time interval considered for prediction.
- u_r is the initial relative velocity between the ego vehicle and the target vehicle before the overtaking maneuver begins.

Once we have determined the TTO using these equations, we introduce a predefined threshold (T) as a reference point. If the calculated TTO is less than this threshold ($TTO < T$), we categorize the overtaking maneuver as aggressive. To further quantify the potential risk associated with such aggressive overtaking, we calculate a collision probability(p) using the following formula:

$$p = \frac{T - TTO}{T} \tag{3.3}$$

Conversely, if the TTO exceeds the predefined threshold ($TTO > T$), we classify the overtaking maneuver as safe.
This methodology, which combines quantitative metrics and probabilistic assessment, enables us to effectively distinguish between aggressive and safe overtaking maneuvers, providing valuable insights into the degree of risk associated with each overtaking event. The whole quantification process can be seen in the flow chart shown in Figure 3.6, where after getting the future parameters of the succeeding vehicle from the trajectory prediction model, we try to find time to reach the ego vehicle and based on that we classify the intention of succeeding vehicle.

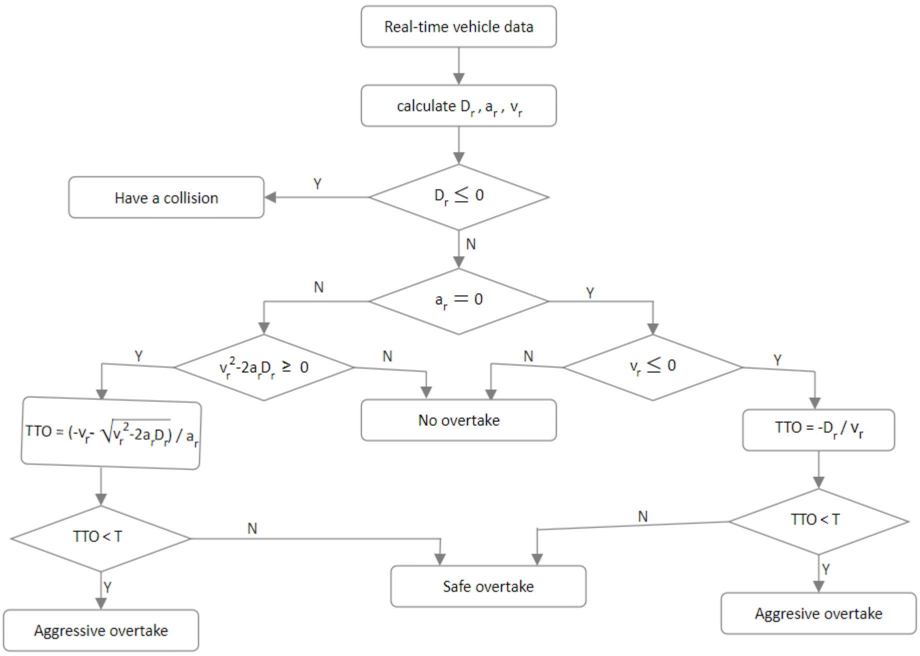

Figure 3.6: Flow chart for aggressive overtake quantification

Chapter 4

Results

In pursuit of enhancing road safety through proactive detection of dangerous overtaking maneuvers, this study employed a multifaceted approach integrating scene generation, data collection using advanced sensors, and LSTM-based trajectory prediction.

Scene Generation using Scenic Language with Carla Simulator: Utilizing the Scenic language within the Carla simulator, diverse traffic scenarios were simulated to capture real-world dynamics crucial for data generation. Scenic's flexibility allowed for the creation of Overtake Scenarios. These simulated environments provided the foundation for capturing nuanced driving behaviors essential for training the LSTM model.

Data Generation using Sensors: Advanced sensors such as Radar and Semantic Segmentation Camera were deployed within the Carla simulator to collect comprehensive data on vehicle interactions and behaviors. The integration of radar data with semantic segmentation imagery enabled the extraction of targeted information pertaining specifically to the behavior of the succeeding vehicle during overtaking maneuvers.

Building LSTM for Trajectory Prediction: The core of our methodology revolved around the application of Long Short-Term Memory (LSTM) recurrent neural networks for trajectory prediction. LSTM networks, renowned for their effectiveness in capturing temporal dependencies, were trained on the dataset generated from Carla simulations. By analyzing historical vehicle trajectories, including positions, velocities, and accelerations, the LSTM model learned to make accurate predictions about future vehicle movements. This predictive capability formed the basis for detecting potential risks associated with overtaking maneuvers.

The trajectory prediction model was successfully trained and evaluated, achieving high levels of accuracy on both the training and test datasets. The performance metrics for the trained model are as follows:
Train Accuracy: 95.98%
Test Accuracy: 85.65%

These results demonstrate the effectiveness of the LSTM-based trajectory prediction model in accurately forecasting future vehicle trajectories based on historical data.

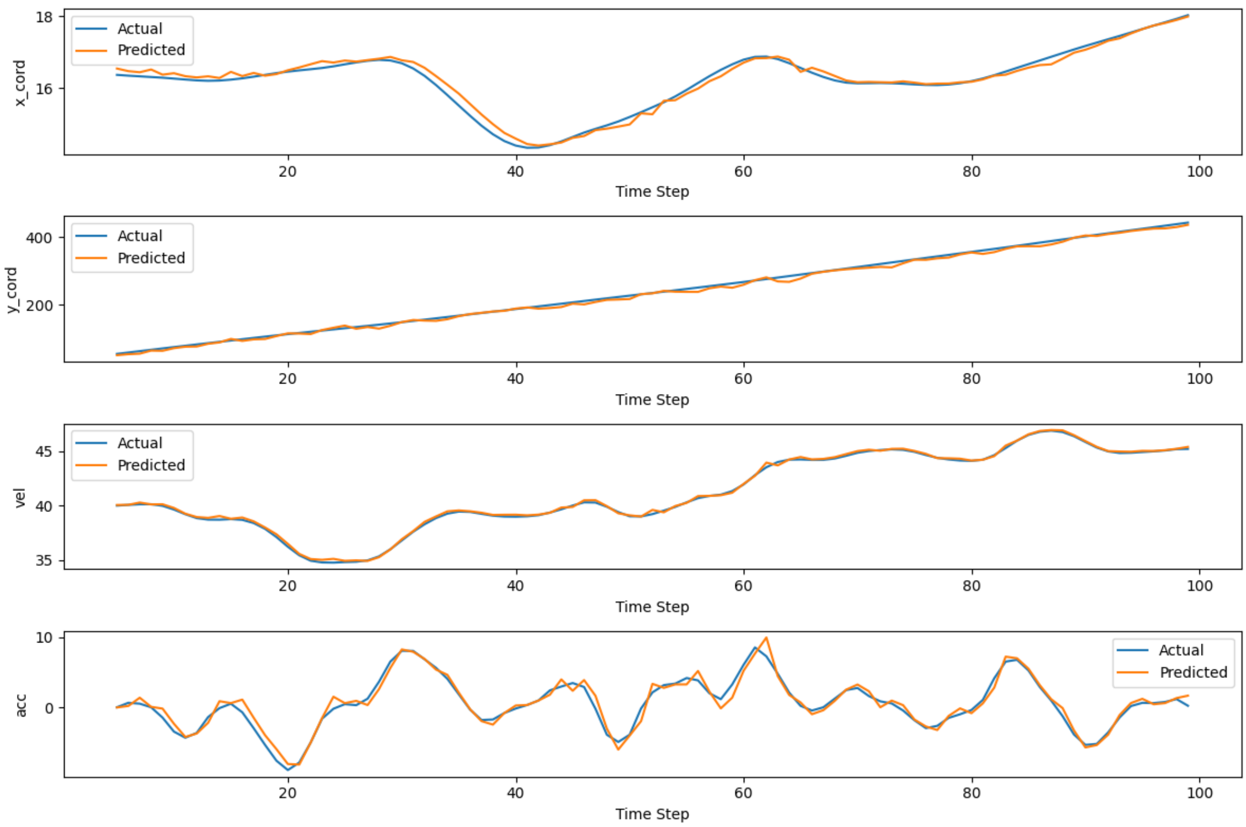

Figure 4.1: Predicted vs Actual value of different parameters of succeeding vehicle over 100 timesteps

Figure 4.1 illustrates the comparison between different actual parameters such as longitudinal position, lateral position, velocity, and acceleration of the succeeding vehicle and their corresponding predicted values generated by the trained LSTM model for the next 100 timesteps. The plot provides visual insights into the model's performance by showcasing the alignment between actual observations and predicted trajectories for key parameters such as position, velocity, and acceleration. This comparison underscores the model's ability to capture the complex dynamics of vehicle movements and make accurate predictions about future trajectories.

Chapter 5

Conclusion

In the pursuit of enhancing road safety and addressing the pressing issue of dangerous overtaking maneuvers, this study has leveraged cutting-edge technologies and methodologies to develop an effective solution. By employing the scenic within Carla simulator, we simulated diverse traffic scenarios, capturing real-world dynamics crucial for data generation. Through meticulous data collection using advanced sensors such as Radar and Semantic Segmentation Camera, we obtained comprehensive insights into vehicle interactions and behaviors, laying the groundwork for subsequent analysis. The core of our approach revolves around the utilization of Long Short-Term Memory (LSTM) recurrent neural networks, integrated with computer vision techniques, to detect hazardous overtaking incidents proactively. Our AI system, trained on the data generated from Carla simulations, exhibits remarkable accuracy in identifying potential risks associated with overtaking maneuvers across varied traffic scenarios. By providing a robust AI-driven solution capable of preventing hazardous overtaking, we have taken a significant stride toward mitigating road accidents and their dire consequences. The outcomes of this research not only contribute to enhancing road safety but also underscore the transformative potential of AI-driven technologies in addressing complex challenges in traffic management and accident prevention.

As road safety remains a shared concern, the findings of this study pave the way for further advancements in AI applications within the realm of traffic management and accident prevention. By addressing the menace of dangerous overtaking, this research emphasizes the importance of innovation, collaboration, and continued efforts in creating safer road environments for present and future generations.

Chapter 6

Future Work

While our current study has successfully utilized the Scenic with Carla simulator to generate overtaking scenarios and implemented Python code for data collection using sensors within the Carla simulator environment, there remains a critical avenue for future exploration and development. One significant challenge encountered during our research lies in the limitations of the Scenic language, particularly in its inability to directly incorporate sensors that can be mounted onto the ego vehicle for data capture purposes in run time. Consequently, an essential area for future work involves the development of an Application Programming Interface (API) that seamlessly integrates our scene generation code written in Scenic with the data collection code written in Python, executed through the Carla Python API.

The proposed API would serve as a bridge between the Scenic environment and the Carla simulator, facilitating the synchronization and coordination of scene generation and sensor-based data collection processes. By enabling communication between these two components, the API would empower us to leverage the capabilities of Scenic for scenario creation while seamlessly integrating data capture functionality through Python scripts utilizing Carla's sensor capabilities that help in getting data at run time and that data can be fed to the prediction model.

Bibliography

[1] D. J. Fremont, T. Dreossi, S. Ghosh, X. Yue, A. L. Sangiovanni-Vincentelli, and S. A. Seshia, "Scenic: a language for scenario specification and scene generation," Jun. 2019. [Online]. Available: http://dx.doi.org/10.1145/3314221.3314633

[2] E. Vin, S. Kashiwa, M. Rhea, D. J. Fremont, E. Kim, T. Dreossi, S. Ghosh, X. Yue, A. L. Sangiovanni-Vincentelli, and S. A. Seshia, "3d environment modeling for falsification and beyond with scenic 3.0," 2023.

[3] G. Bagschik, T. Menzel, and M. Maurer, "Ontology based scene creation for the development of automated vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1813–1820.

[4] Y. Jeong, S. Kim, and K. Yi, "Surround vehicle motion prediction using lstm-rnn for motion planning of autonomous vehicles at multi-lane turn intersections," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 1, pp. 2–14, 2020.

[5] L.-T. Wu and H.-Y. Lin, "Overtaking vehicle detection techniques based on optical flow and convolutional neural network." in *VEHITS*, 2018, pp. 133–140.

[6] L.-T. Wu, V. L. Tran, and H.-Y. Lin, "Real-time overtaking vehicle detection based on optical flow and convolutional neural network," in *Smart Cities, Green Technologies and Intelligent Transport Systems: 7th International Conference, SMARTGREENS, and 4th International Conference, VEHITS 2018, Funchal-Madeira, Portugal, March 16-18, 2018, Revised Selected Papers 7*. Springer, 2019, pp. 227–243.

[7] J. García-González, I. García-Aguilar, D. Medina, R. M. Luque-Baena, E. López-Rubio, and E. Domínguez, "Vehicle overtaking hazard detection over onboard cameras using deep convolutional networks," in *17th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2022) Salamanca, Spain, September 5–7, 2022, Proceedings*. Springer, 2022, pp. 330–339.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[9] Z.-N. Li, X.-H. Huang, T. Mu, and J. Wang, "Attention-based lane change and crash risk prediction model in highways," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 22 909–22 922, 2022.

[10] A. Zyner, S. Worrall, J. Ward, and E. Nebot, "Long short term memory for driver intent prediction," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1484–1489.

[11] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," 2017.

[12] "Ngsim: Next generation simulation, fhwa, u.s. department of transportation, washington, dc, usa, may 2007. [online]. available: http://www.fhwa.dot.gov/."