Republic of Tunisia
Ministry of Higher Education
and Scientific Research

University of Sfax

National School of Electronics
and Telecommunications of Sfax

**Engineer in :**
Data Engineering
and Decision Systems

**Report :**
End of The Year Project

**Ordre Nb:** PFA-IDSD-23-03

# End of The Year Project

***Presented at***

**National School of Electronics and Telecommunications of Sfax**

***Field of study:***

**Data Engineering and Decision Systems**

## AI in Self Driving Cars

***by***

**Hanine Mahmoudi**
**Mohamed Aziz Ben Lazreg**

Presented on: 24/04/2024 before the jury of commission:

| | |
|---|---|
| **Mr. Mahdi Louati** | **Examinator** |
| **Mr. Ramzi Zouari** | **Supervisor** |

# DEDICATIONS

I dedicate this effort to my loving parents Abderazak and Jamila Benlazreg , whose unwavering support and encouragement have been instrumental in my journey. Your belief in me has been a constant source of inspiration, and I am grateful for your guidance and love. This end of the year project report is a testament to your unwavering faith in my abilities. Thank you for always being there for me.

I want to take this moment to express my deep gratitude to The National School of Electronics and Telecommunications, ENET'Com Sfax. The experience has proven to be invaluable and enlightening.

I also wish to convey my sincere thanks to my supervisor, Mr. Ramzi Zouari, whose guidance and support were indispensable throughout this journey . His expertise and insights were instrumental in helping me overcome challenges and achieve my goals.

I am equally grateful to my fellow my colleague Hanine Mahmoudi . Working with her during this journey was a pleasure, and her collaboration, support, and camaraderie made the experience even more enjoyable and rewarding.

Lastly, I would like to give a special mention to my friends Hamdi Kacem, Aya Gharbi,Oussama Boubaker, Zouhair Ayeb, and Ala Jouirou, whose unwavering support throughout the journey was greatly appreciated. Their encouragement and friendship helped me stay focused and motivated during the period.

**Mohamed Aziz Benlazreg**.

In loving memory of my late father, Gharbi Mahmoudi, whose guidance and wisdom continue to inspire me every day. I hope to make him proud with my accomplishments.

To my dear mother, Fathia Ben Braiek, whose endless love and sacrifices have been the cornerstone of my journey. Your unwavering support has fueled my determination and kept me going through every challenge.

To my sister, Ichrak, your belief in me has been a constant source of strength. Thank you for standing by my side and cheering me on, even in the toughest times.

I wish to convey my sincere thanks to my supervisor, Mr. Ramzi Zouari, whose guidance and support were indispensable throughout this journey. His expertise and insights were instrumental in helping me overcome challenges and achieve my goals.

To my dearest friends, Takwa, Omar, Ranim, and Adam, you're the family I chose. Your companionship, laughter, and unwavering support are my guiding light. Thank you for shaping who I am.

And to Aymen, who goes above and beyond. Your presence during this unforgettable year, has brought endless joy and laughter even in the toughest times. I'm incredibly grateful for you.

Last but not least, to my colleague Mohamed Aziz Ben Lazreg, thank you for your collaboration and dedication to our shared goals. Together, we have achieved great things, and I am thankful for the opportunity to work alongside such talented individuals.

**Hanine Mahmoudi**.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

**AI** Artificial Intelligence

**CARLA** Car Learning to Act

**ML** Machine Learning

**DL** Deep Learning

**CRISP-DM** Cross-Industry Standard Process for Data Mining

**RL** Reinforcement Learning

**LIDAR** Light Detection and Ranging

**CNN** Convolutional Neural Network

**GPS** Global Positioning System

# GENERAL INTRODUCTION

In recent years, the pursuit of autonomous driving technology has accelerated, driven by advancements in artificial intelligence and machine learning. Reinforcement learning (RL) has emerged as a prominent approach in this domain, offering a means to train intelligent agents through interaction with dynamic environments. This report delves into the application of RL techniques in the context of autonomous driving, specifically focusing on the integration of semantic segmentation models for lane-following detection within the CARLA simulator.

The utilization of semantic segmentation models plays a pivotal role in enabling autonomous vehicles to perceive and interpret their surroundings effectively. By segmenting images captured by vehicle-mounted cameras into meaningful semantic regions, such as road, vehicles, pedestrians, and lane markings, these models facilitate crucial decision-making processes for navigation and control. In this study, we explore the integration of two semantic segmentation models tailored for lane-following detection, leveraging their outputs to inform steering commands in a simulated driving environment.

The CARLA simulator provides a realistic and dynamic platform for testing and evaluating autonomous driving algorithms in various scenarios. Through its high-fidelity rendering, accurate physics simulation, and extensive suite of sensors, CARLA enables researchers and developers to assess the performance and robustness of their autonomous systems in a controlled yet realistic setting.

This project consists of three main parts. The first part is an introduction that provides a general overview of the field of autonomous vehicles. It examines the role of artificial intelligence in these vehicles and the importance of lane following semantic segmentation models in autonomous driving. Additionally, this part also introduces the CARLA simulator, detailing its features and capabilities.

The second part focuses on integrating artificial intelligence into autonomous vehicles. It begins with a general introduction, followed by an in-depth analysis of reinforcement learning in simulated environments. This part also examines the UNet semantic segmentation model, describing its architecture, training, and evaluation. It concludes with a discussion on the lane following model, its performance, and evaluation.

Finally, the third part presents a synthesis of the obtained results, conclusions drawn from this project, and suggestions for future work. It offers an overall perspective on how artificial intelligence can be applied in the field of autonomous vehicles, highlighting challenges and opportunities for future research.

Chapter

1

# INTRODUCTION

## Contents

## 1.1 INTRODUCTION

In recent years, the pursuit of autonomous driving technology has accelerated, driven by advancements in artificial intelligence and machine learning. Reinforcement learning (RL) has emerged as a prominent approach in this domain, offering a means to train intelligent agents through interaction with dynamic environments. This report delves into the application of RL techniques in the context of autonomous driving, specifically focusing on the integration of semantic segmentation models for lane-following detection within the CARLA simulator.

The utilization of semantic segmentation models plays a pivotal role in enabling autonomous vehicles to perceive and interpret their surroundings effectively. By segmenting images captured by vehicle-mounted cameras into meaningful semantic regions, such as road, vehicles, pedestrians, and lane markings, these models facilitate crucial decision-making processes for navigation and control. In this study, we explore the integration of two semantic segmentation models tailored for lane-following detection, leveraging their outputs to inform steering commands in a simulated driving environment.

The CARLA simulator provides a realistic and dynamic platform for testing and evaluating autonomous driving algorithms in various scenarios. Through its high-fidelity rendering, accurate physics simulation, and extensive suite of sensors, CARLA enables researchers and developers to assess the performance and robustness of their autonomous systems in a controlled yet realistic setting.

## 1.2 Overview Of Self-Driving Cars

- Definition and Concept :

  Self-driving cars, also known as autonomous vehicles (AVs), mark a significant leap forward in the realm of transportation technology. These innovative vehicles are engineered to function independently, without the need for constant human oversight. Instead, they rely on a sophisticated blend of cutting-edge technology, including an array of sensors to detect their surroundings, actuators to control movement, and advanced software algorithms to process information and make decisions. By leveraging these components, self-driving cars are able to interpret the world around them, assess potential hazards, and chart a safe course to their destination with precision and efficiency. This transformative concept of self-driving cars paints a picture of a future where individuals can liberate themselves from the responsibilities of manual driving, allowing them to relax and enjoy the journey while the vehicle handles the task of navigation autonomously.

- History of Self-Driving Cars :

  The history of self-driving cars unfolds across distinct time periods, each marked by significant milestones and advancements in autonomous vehicle technology. From the conceptual stages before 1980 to the thriving industry of the 2010s, this journey offers valuable insights into the evolution of self-driving technology and its potential implications for the future of transportation. Let's explore each era and delve into the pivotal events that have shaped the landscape of self-driving cars.

### 1.2.1 AI's role in autonomous vehicles

Artificial Intelligence (AI) plays a pivotal role in the development of autonomous vehicles, revolutionizing transportation by enabling vehicles to perceive, interpret, and act upon their surroundings without human intervention. At the core of AI-driven autonomy lies the fusion of various technologies such as machine learning, computer vision, and robotics, allowing vehicles to navigate complex environments, make decisions in real-time, and adapt to changing conditions. AI algorithms enable autonomous vehicles to detect and classify objects, understand traffic signs and signals, plan optimal routes, and execute precise maneuvers, thereby ensuring safe and efficient operation on roads. By harnessing the power of AI, autonomous vehicles hold the promise of reducing accidents, alleviating traffic congestion, and transforming mobility for individuals worldwide.

### 1.2.2 Significance of lane following semantic segmentation models in autonomous driving

Lane following and semantic segmentation models are integral components of autonomous driving systems, playing critical roles in enhancing vehicle perception, navigation, and decision-making capabilities. Lane following models enable vehicles to maintain their position within lane boundaries, facilitating smooth and safe navigation along roadways. By accurately tracking lane markings and detecting deviations, these models ensure precise steering control and lane-keeping maneuvers, essential for highway driving and urban navigation. Semantic segmentation models, on the other hand, provide vehicles with the ability to understand and interpret the surrounding environment by assigning semantic labels to individual pixels in sensor data. This enables vehicles to differentiate between various objects and obstacles such as vehicles, pedestrians, road signs, and traffic signals, enabling informed decision-making and proactive avoidance of potential hazards. Together, lane following and semantic segmentation models contribute to the overall safety, efficiency, and reliability of autonomous driving systems, paving the way for the widespread adoption of autonomous vehicles on public roads.

## 1.3 CARLA Simulator

### 1.3.1 Introduction

CARLA (Car Learning to Act), developed to support the comprehensive development, training, and validation of autonomous driving systems, stands as a cornerstone in the realm of autonomous vehicle research. It offers not only open-source code and protocols but also a rich repository of open digital assets including urban layouts, buildings, and vehicles, purposefully crafted for this very endeavor. These assets are freely accessible, providing a robust foundation for developers and researchers to build upon. CARLA's simulation platform embodies flexibility at its core, allowing for the specification of sensor suites, environmental conditions, and full control over static and dynamic actors within the simulated environment. With the ability to generate maps adhering to the ASAM OpenDRIVE standard using tools like RoadRunner, CARLA empowers users to craft bespoke environments tailored to their specific research needs.

**Figure 1.1: Screenshots from the CARLA simulator**

### 1.3.2 Overview of CARLA's features and capabilities

CARLA's features are as extensive as they are impressive, ranging from its scalable server multi-client architecture to its autonomous driving sensor suite. The platform's flexible API grants users unparalleled control over various simulation aspects, including traffic generation, pedestrian behaviors, weather conditions, and sensor configurations. Noteworthy capabilities such as fast simulation mode, maps generation tools, and traffic scenarios simulation engine, ScenarioRunner, further solidify CARLA's status as a versatile and powerful tool for autonomous driving research. Integration with ROS via the ROS bridge expands CARLA's utility, enabling seamless connectivity with the Robot Operating System. Additionally, the provision of Autonomous Driving baselines as runnable agents within CARLA, including AutoWare and Conditional Imitation Learning agents, serves as a valuable resource for researchers and developers seeking to benchmark their algorithms.

## 1.4 Work Methodology

In this project, the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology served as the foundational framework for my work methodology. Employing CRISP-DM ensured a systematic approach to navigating the intricacies of data mining and analytics projects. From initial business understanding to final deployment, each phase of the project followed a structured and iterative process. By adhering to the steps of business understanding, data understanding, data preparation, modeling, evaluation, and deployment, I effectively planned, executed, and assessed the data mining solution. CRISP-DM's adaptability allowed for flexibility in responding to evolving project requirements and incorporating new insights. Ultimately, leveraging CRISP-DM as the work methodology facilitated the delivery of a robust and tailored data-driven solution aligned with the project's objectives.

## 1.5 CONCLUSION

This introductory chapter has provided a comprehensive overview of key aspects related to self-driving cars and the CARLA Simulator, laying the groundwork for the subsequent sections of the report. The discussion began by highlighting the increasing significance of autonomous vehicles in the modern transportation landscape, emphasizing the pivotal role of artificial intelligence and the importance of lane following semantic segmentation models in ensuring safe and efficient autonomous driving. Furthermore, the introduction introduced the CARLA Simulator as a fundamental tool for testing and evaluating autonomous driving algorithms, outlining its features and capabilities. Moving forward, the outlined work methodology, guided by the CRISP-DM framework, will provide a structured approach to addressing the project objectives effectively. Overall, this introductory chapter sets the stage for the in-depth exploration and analysis that will follow, with the aim of contributing valuable insights to the field of autonomous driving technology.

Chapter

# 2 AI for Autonomous Vehicles

## Contents

## 2.1 INTRODUCTION

In this chapter, we delve into the practical application of reinforcement learning within simulated environments provided by Carla. Specifically, we explore the integration of reinforcement learning techniques alongside our semantic segmentation and lane-following models. We detail the development, training, and testing processes of these models, along with their evaluation using various metrics. Finally, we assess the performance of these models within the Carla simulator to gauge their effectiveness in real-world-like scenarios.

## 2.2 Reinforcement Learning in Simulated Environments

### 2.2.1 Introduction

Autonomous driving technology holds immense potential to revolutionize transportation systems, offering benefits in safety, efficiency, and accessibility. Reinforcement learning (RL) presents a promising approach for training autonomous agents to navigate complex environments through trial and error learning. In this project, we harness the power of RL to develop an autonomous car system capable of learning to drive in simulated environments. Our goal is to create an adaptive and robust driving agent that can safely follow lanes, avoid obstacles, and reach its destination autonomously.

### 2.2.2 Reinforcement Learning Overview

Reinforcement learning (RL) is a machine learning paradigm where an agent interacts with an environment to learn optimal actions through trial and error. The agent receives feedback in the form of rewards or penalties based on its actions, guiding it towards desirable behaviors. RL involves defining an environment, agent, actions, observations, rewards, and a policy for action selection. Through iterative training episodes, the agent learns to maximize its cumulative rewards by adjusting its policy based on observed states and received rewards.

### 2.2.3 Components of the Project

Our project comprises several key components essential for training an autonomous driving agent using RL. These components include the agent (the car in the simulated environment), the environment (the simulated town where the car operates), actions (such as steering and throttle inputs), observations (camera images from the car's perspective), rewards (feedback from the environment based on agent actions), and policy (the strategy for action selection learned by the agent).

## 2.3 Semantic Segmentation Model

Semantic segmentation is essential for autonomous vehicles as it categorizes each pixel in an image, enabling precise understanding of the environment. By developing a segmentation model, we empower autonomous cars to interpret surroundings accurately, ensuring safe navigation and informed decision-making. This model supports tasks like lane detection and object recognition, enhancing vehicle performance. Testing this model in the CARLA simulator environment simulates real-world scenarios, validating its effectiveness in various driving conditions and ensuring practical reliability.

### 2.3.1 UNet Model Overview

The U-Net model architecture is ideal for training on self-driving car datasets due to its effectiveness in semantic segmentation tasks. With its unique design and skip connections, U-Net accurately delineates object boundaries within images, crucial for identifying pedestrians, vehicles, and road boundaries. Its ability to preserve spatial information makes it well-suited for analyzing complex scenes encountered in autonomous driving scenarios. By leveraging U-Net, we aim to develop a robust segmentation model to enhance the perception capabilities of autonomous vehicles. The U-Net architecture consists of a contracting path followed by an expanding path, which resembles a "U" shape, hence the name.

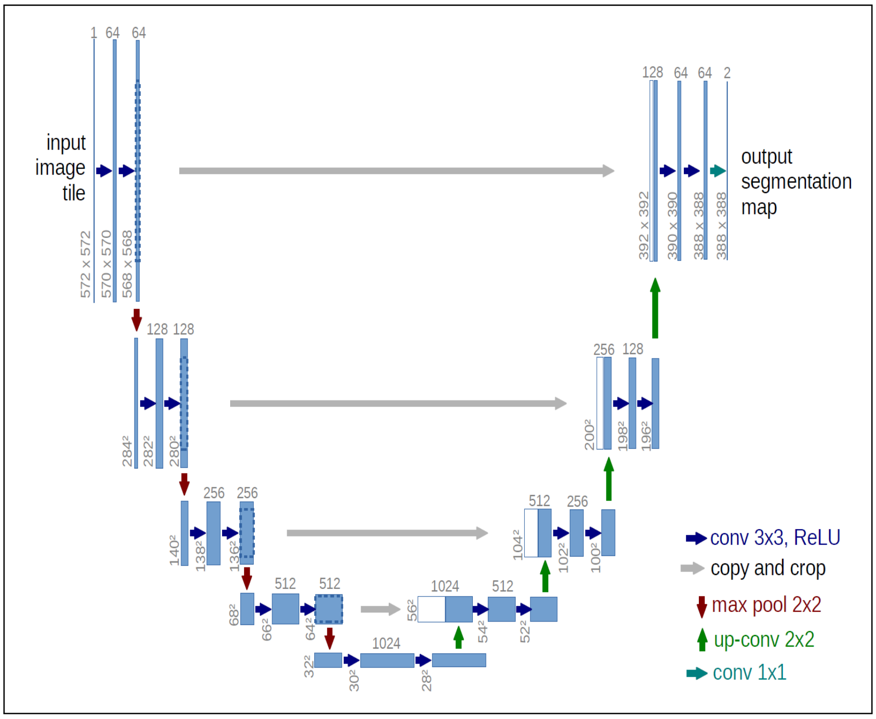



**Figure 2.1: U-net Model Architecture**

#### 2.3.1.1 Encoder (Contracting Path)

The Encoder, also known as the Contracting Path, is composed of convolutional layers followed by max-pooling layers. This part of the network is responsible for extracting features from the input image. The Encoder begins with an input image ,often referred to as a tensor, which is passed through a stack of convolutional layers .

Each convolutional layer applies a set of learnable filters to the input image.These operations help identify meaningful information in the image, which is essential for semantic segmentation. After each convolutional operation, a non-linear activation function such as ReLU (Rectified Linear Unit) is applied element-wise to introduce non-linearity into the network. ReLU specifically sets all negative values in the feature map to zero, while retaining positive values unchanged.

By setting negative values to zero, ReLU introduces a thresholding effect that helps the network learn and represent complex patterns in the data. This promotes the sparsity of activations and allows the network to focus on the most important features while suppressing less relevant information.

After each convolutional block, max-pooling is applied to reduce the spatial dimensions of the feature maps (downsampling) while retaining the most important features.

Max pooling divides the input feature maps into non-overlapping regions (typically 2x2 or 3x3), known as pooling windows or kernels, for each pooling window, max pooling retains only the maximum value within that window and discards all other values. The maximum value effectively represents the most important feature within the region, hence the term "max pooling". By retaining only the maximum values, max pooling effectively reduces the spatial dimensions of the feature maps,for example, a 2x2 max pooling operation with a stride of 2 reduces the size of the feature maps by half along both the height and width dimensions. Max pooling is applied periodically in the contracting path of the U-Net architecture to progressively downsample the feature maps.

This process allows the network to capture hierarchical representations of the input image, where The encoder compresses the input into a compact representation with semantic information, utilized by the decoder to generate accurate segmentation masks.

#### 2.3.1.2 Decoder (Expanding Path)

In the U-Net architecture, the decoder, also known as the expanding path, is responsible for reconstructing high-resolution segmentation masks from the low-resolution feature maps produced by the contracting path. The decoder mirrors the structure of the contracting path but performs upsampling instead of downsampling.

The decoder starts by upsampling the low-resolution feature maps obtained from the contracting path to restore their original spatial dimensions. This process aims to recover the spatial details lost during downsampling in the contracting path. Into the decoder, skip connections from the contracting path are integrated , which contain high-resolution spatial information and provide additional detail-rich features that help in reconstructing high-resolution segmentation masks with precise object boundaries.

By combining the upsampled feature maps with skip connections, the decoder ensures that both low-level and high-level features are utilized in the segmentation process. This integration enables the network to capture fine details and intricate patterns while maintaining a global understanding of the image context. Convolutional layers within the decoder further refine the combined feature maps by learning more abstract representations. These layers enhance the network's ability to distinguish between different classes and accurately segment objects in the input image. The decoder concludes with a final convolutional layer that produces the segmentation masks. This layer maps the learned features to the desired number of classes, generating probability distributions over the classes for each pixel in the output masks.

In summary, the decoder in the U-Net architecture plays a crucial role in reconstructing high-resolution segmentation masks by combining upsampling operations, skip connections, convolutional layers, and a final convolutional layer to produce accurate and precise segmentation results. .[1]

### 2.3.2 Data overview , Preprocessing and Augmentation

Our dataset, sourced from the CARLA self-driving car simulator, contains images and labeled semantic segmentations aimed at pixel-level image classification through semantic segmentation. This dataset aids self-driving cars in segmenting camera images into meaningful sections, enhancing their environmental interpretation. Captured under various weather conditions during simulated car journeys, the dataset's consecutive images provide a comprehensive view of real-world scenarios. With 5000 RGB images and corresponding semantic segments, it encompasses diverse scenes and objects, each meticulously labeled with categories like buildings, fences, people, vehicles, and traffic signs. .[2]

**Figure 2.2:** ***Example of RGB image and corresponding segmentation mask***

During preprocessing , we prepare our dataset by reading image and mask paths and converting them into arrays. This involves decoding image files, converting them to floating-point type, resizing them, and ensuring binary mask arrays. Subsequently, we construct a data generator to handle the dataset, which includes caching, shuffling, and batching the data for training.

While analyzing the dataset, we observed a class imbalance, where the number of pixels belonging to each category is not uniformly distributed. This imbalance can lead overreliance on accuracy as the sole evaluation metric, as the model might prioritize the majority class and perform poorly on less frequent ones. To address this challenge and improve model performance across all classes, data augmentation techniques will be employed. .[3]

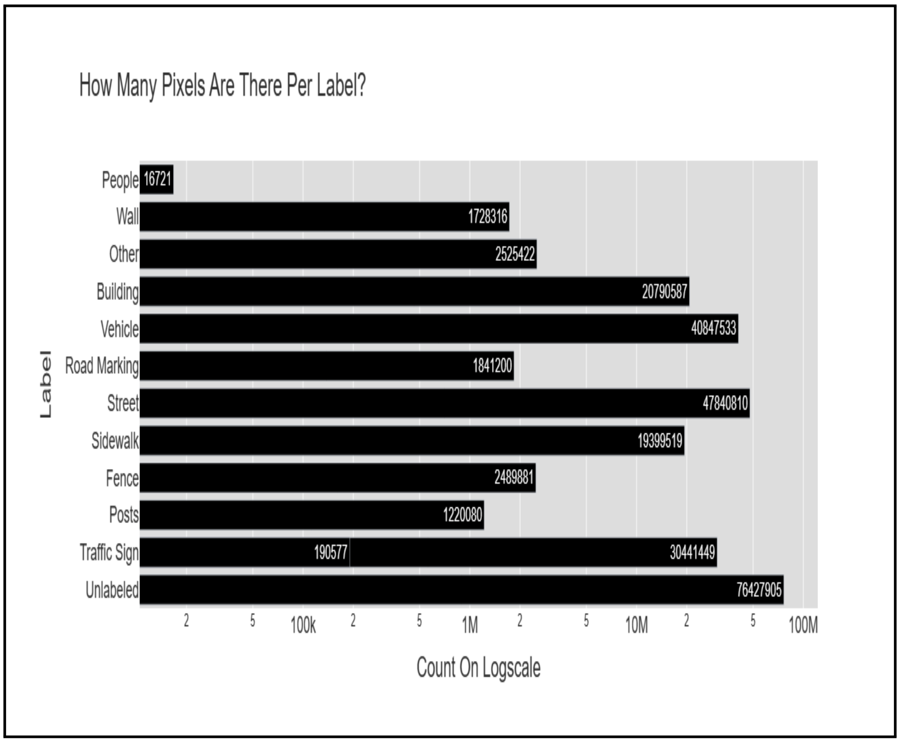



**Figure 2.3:** ***Imbalanced Distribution of Pixels in CARLA Semantic Segmentation Dataset***

Data augmentation is a technique commonly employed in deep learning to artificially expand the size and diversity of a training dataset. This is achieved by applying random transformations to existing images, such as cropping, flipping, rotating, and color jittering.

In the initial phase of our project, we evaluated a U-Net model for image segmentation on the preprocessed dataset without incorporating data augmentation. We observe persistent underfitting in the accuracy graph, with both training and test accuracy remaining low and stagnant. Additionally, the presence of the EarlyStopping callback suggests potential termination of training due to lack of improvement, further confirming the model's struggle with capturing dataset complexities.

In the next stage of our project, we will explore the integration of data augmentation techniques to address the limitations observed in the initial evaluation. We employed various techniques to enhance the robustness of our semantic segmentation model, ensuring its applicability in real-world scenarios.

For instance, incorporating zoom augmentation mirrors the challenges faced in autonomous driving systems, where vehicles must accurately detect objects even at considerable distances.

By simulating zooming effects in our training data, our model learns to effectively handle object recognition tasks at varying scales, thereby improving its adaptability to different viewing distances. Additionally, the inclusion of noise augmentation, inspired by adverse weather conditions, introduces variability into the dataset, enabling the model to better generalize to real-world scenarios where environmental conditions may not always be optimal. Moreover, random rotations mimic the unpredictable nature of camera orientations in practical settings, enabling our model to effectively handle images captured from different angles.

By integrating these data augmentation techniques, our model becomes more resilient to the diverse challenges encountered in real-life scenarios, ultimately enhancing its performance and reliability in autonomous driving applications. .[4]

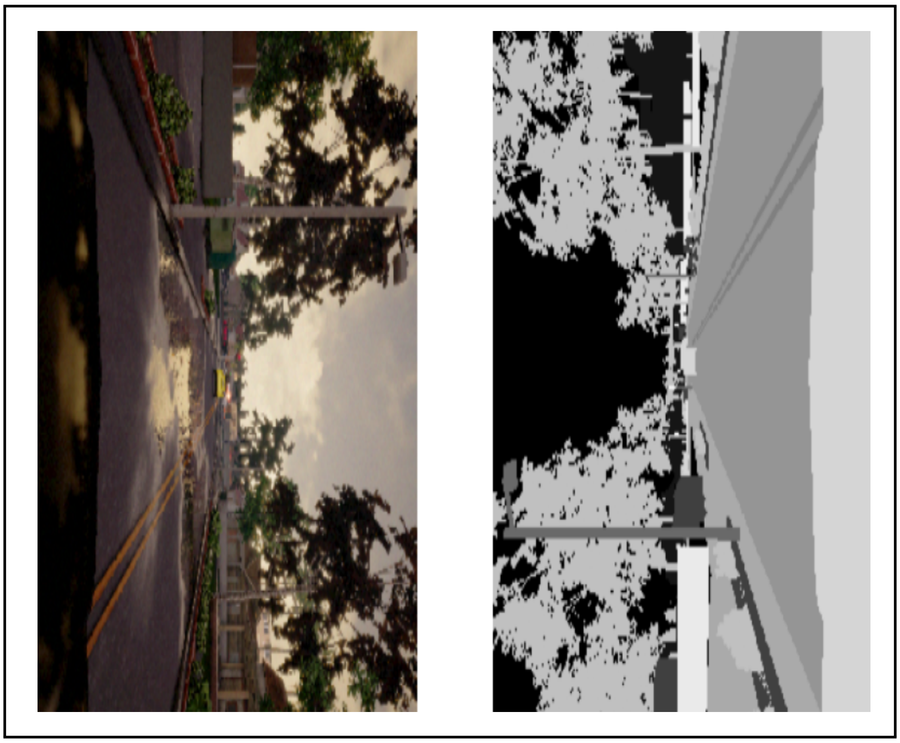


**Figure 2.4:** ***Augmented image and mask with random horizontal flip***

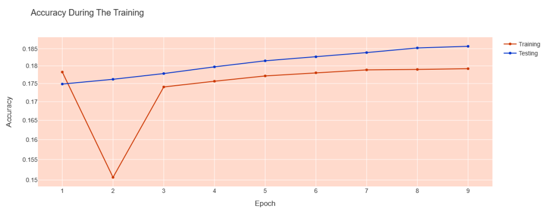
By introducing these variations, we aim to enhance the model's ability to learn generalizable features and improve its performance on unseen data. We observe significant improvement in both training and test accuracy over epochs, with the training accuracy climbing from around 0.8 to nearly 0.98. Despite a slight deviation, the test accuracy shows a positive upward trend, indicating robust generalization to unseen data. Overall, the fact that both the training accuracy and test accuracy are high and increasing suggests that the model is being trained and tested very well after applying data augmentation.

### 2.3.3 Model architecture and training

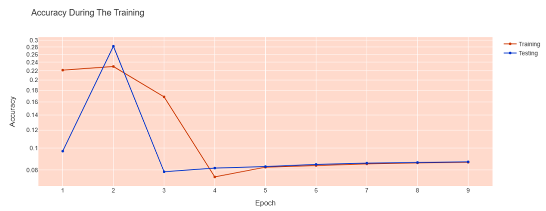
To construct our model architecture, we first establish encoding and decoding blocks. The encoding block function is designed to generate both the next layer's output and the skip connection output, vital for information preservation. Conversely, the decoding block function merges the skip-connection input with the previous layer, undergoes processing, and produces an output. By incorporating these blocks, we develop a comprehensive model that efficiently captures features through encoding and reconstructs the image through decoding, facilitating robust semantic segmentation.

In this section, three unique U-Net models for semantic segmentation are explored. While all adhere to the basic U-Net architecture with contracting and expanding paths, they differ in internal design and activation functions, particularly in convolutional layer hyperparameters. The aim is to identify optimal hyperparameters that suit the specific requirements of semantic segmentation.

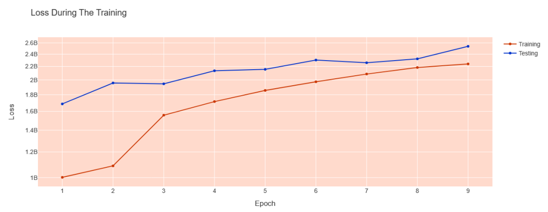
Our initial model architecture for semantic segmentation, referred to as Model 1, was based on the classic U-Net structure. It employed ReLU activation functions after each convolutional layer to introduce non-linearity. Dropout with a probability of 0.3 was applied only in the deepest block of the contracting path to prevent overfitting. Additionally, max pooling with a (2, 2) pool size was utilized after each convolutional block in the contracting path (except the deepest block with dropout) to reduce spatial dimensionality and focus on higher-level features. The number of filters in the convolutional layers doubled progressively down the contracting path (typically starting at 32) and halved up the expanding path, capturing increasingly complex image features. This initial model provided a foundation for further investigation into hyperparameters to potentially enhance performance. .[5]



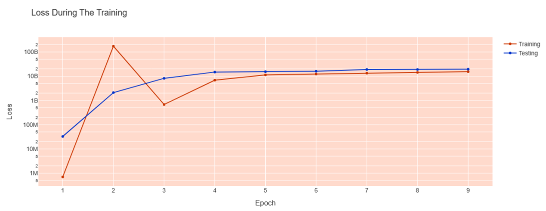

**(a) Training and Testing Accuracy of Initial Model before data augmentation**




**(b) Training and Testing Accuracy of Initial Model after data augmentation**




**(c) Training and Testing Loss of Initial Model before data augmentation**

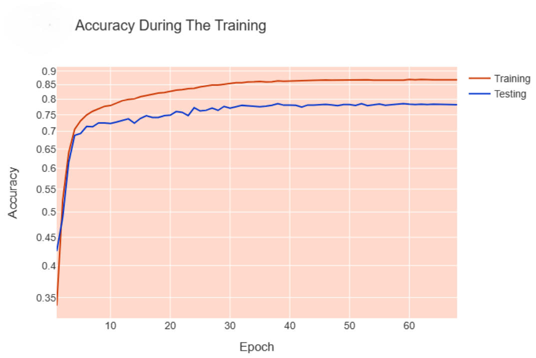



**(d) Training and Testing Loss of Initial Model after data augmentation**

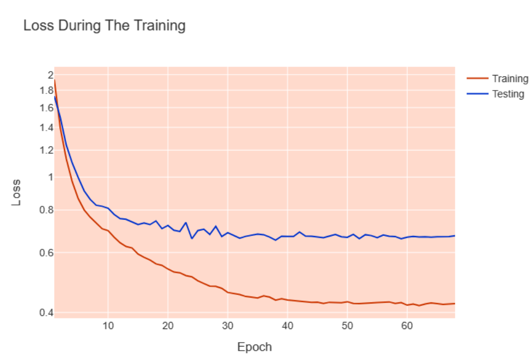
**Figure 2.5: Comparison of Initial Model Performance Before and After Data Augmentation**

While analyzing these graphs of Comparison of Initial Model Performance Before and After Data Augmentation, it becomes evident that the performance of the initial model did not exhibit significant improvement even after the implementation of data augmentation techniques. This observation underscores the need for a closer examination of the model's hyperparameters, as the anticipated enhancement in performance was not achieved through data augmentation.

Building upon the foundation of Model 1, we developed Model 2 with the objective of enhancing training stability and potentially accelerating convergence. To achieve this, we incorporated Batch Normalization as a hyperparameter after each convolutional layer in both the contracting and expansive paths of the U-Net architecture. Batch Normalization introduces a normalization step that helps alleviate the vanishing/exploding gradient problem, a common challenge in deep neural networks. This technique essentially standardizes the activations of neurons within a layer, preventing individual neurons from dominating the learning process and allowing the network to learn more efficiently. [6]




**(a) Training and Testing Accuracy of Model 2 after data augmentation**

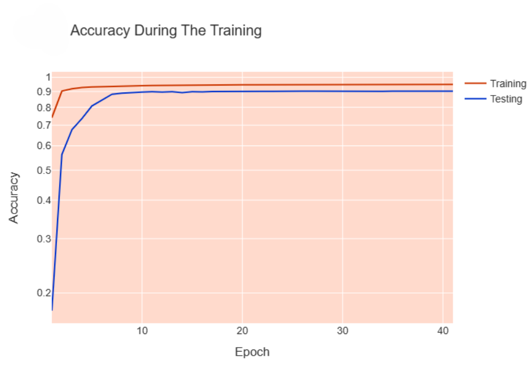



**(b) Training and Testing Accuracy of Model 2 after data augmentation**

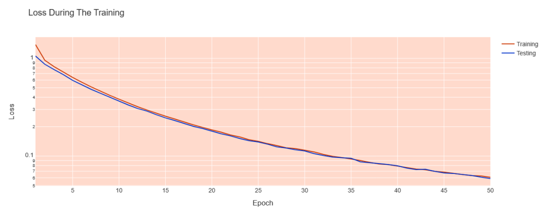
**Figure 2.6: Model 2 Performance After Data Augmentation**

Following Model 2's success with Batch Normalization, we designed Model 3 to explore the potential benefits of combining Batch Normalization with Leaky ReLU activation. This model aims to achieve faster convergence and potentially improve training stability compared to both Model 1 and Model 2. We replaced the standard ReLU activation function used in Model 1 and 2 with Leaky ReLU. This modification allows for a small non-zero gradient even for negative inputs, theoretically leading to faster convergence during training, especially for deeper networks like U-Nets. Similar to Model 2, Batch Normalization remains a crucial hyperparameter applied after each convolutional layer in both the contracting and expansive paths. This technique continues to contribute to stabilizing the training process and addressing the vanishing/exploding gradient problem.

By incorporating both hyperparameters, Model 3 offers a more comprehensive exploration of U-Net architecture modifications for semantic segmentation,to potentially achieve faster convergence and improved training stability compared to the baseline Model 1. .[7]




**(a) Training and Testing Accuracy of Model 3 after data augmentation**




**(b) Training and Testing Accuracy of Model 3 after data augmentation**

**Figure 2.7: Model 3 Performance After Data Augmentation and The combination of Leaky ReLU and Batch Normalization**

Model 3 demonstrated notable performance enhancements post-training, achieving a training accuracy of 0.95 and testing accuracy of 0.92. Favorable values were also observed for both training and testing losses, highlighting effective learning and generalization abilities. Consequently, Model 3 was chosen for subsequent evaluation and integration into the CARLA simulator environment. Its strong performance metrics and improved training stability position it as a viable option for real-world applications, especially in autonomous driving systems. Selecting Model 3 aims to enhance navigation reliability and efficiency across various driving scenarios, thereby advancing autonomous vehicle technology.

### 2.3.4 Lane Following Model

The convergence of RL, represented by Stable Baselines 3, and CARLA, our driving simulator, constituted the foundation of our experimentation. Initial attempts at training proved futile, primarily due to a lack of effective model convergence, resulting in recurrent collisions within the simulator. Consequently, several pivotal modifications were undertaken to refine the system's functionality and enhance training efficacy.

Behind the operational facade lies a seamless interaction between the CARLA simulator and the RL framework. The simulator transmits front-facing camera images to the RL environment, meticulously segmented to categorize objects by color. These images serve as the bedrock for RL training, wherein the agent endeavors to optimize actions to maximize rewards while navigating the virtual environment.

The images relayed from the simulator undergo a meticulous preprocessing regimen to extract pertinent road information. This preprocessing entails cropping the non-essential top portion and converting the images into a simplified format amenable to efficient processing. Furthermore, a dedicated CNN model is trained to discern steering outputs from these images, augmenting the agent's comprehension of the dynamic environment.

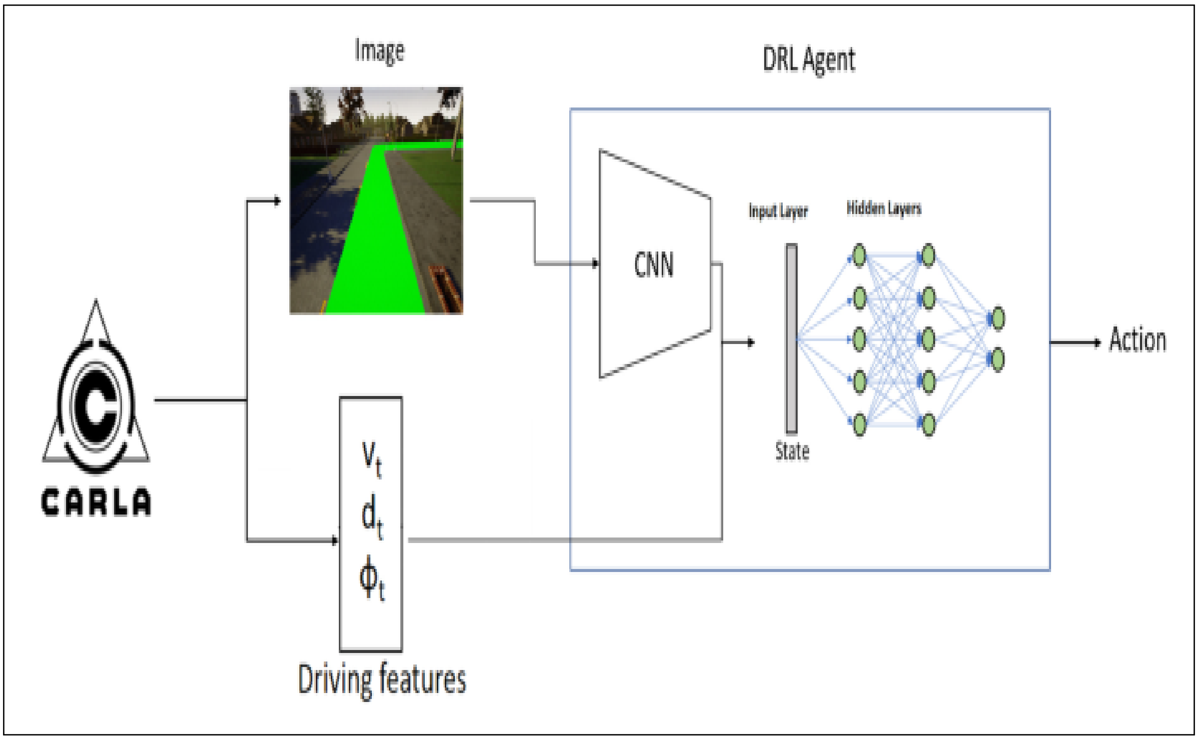



**Figure 2.8:** ***General Overview***

### 2.3.5 Model evaluation

Model evaluation is essential for refining and selecting the most suitable model for our dataset, ensuring optimal performance both currently and in the future.

To execute evaluations, we generate segmentations and assess their performance against ground truth labels. During this process, we meticulously compared the predicted segmentations produced by our model with the ground truth segment images. This thorough analysis revealed a remarkable similarity between the predicted and actual segmentations, indicating the model's accurate identification and classification of semantic features in the images. These findings strongly suggest that our model performed exceptionally well in delineating various objects and structures within the images. The close resemblance between the predicted and ground truth segmentations provides compelling evidence of our model's effectiveness and reliability in semantic segmentation tasks.

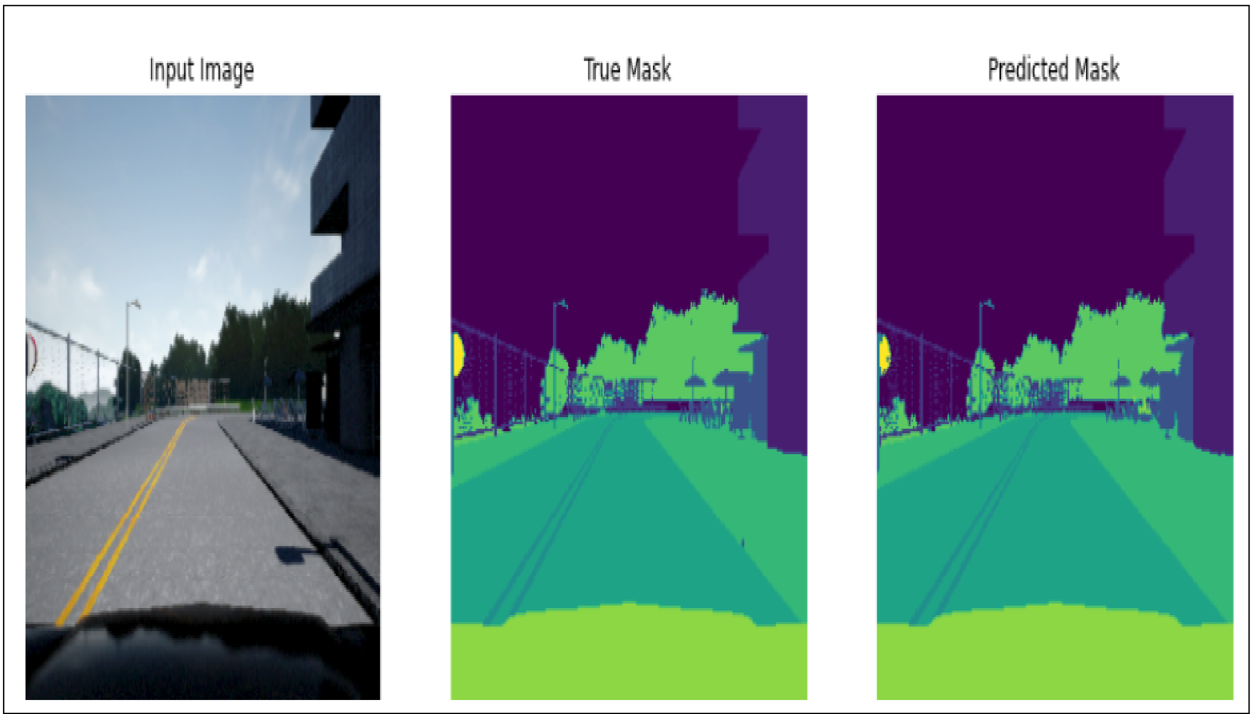



**Figure 2.9:** ***Comparison of Input Image, True Mask, and Predicted Mask***

While model accuracy is commonly used, it may not provide a comprehensive assessment, especially in tasks like image segmentation with class imbalances.

To address this, we employ additional metrics such as precision, recall, specificity, TDR, IoU, and F1-score, offering deeper insights into performance. By calculating these metrics from the confusion matrix comparing predicted segmentations with ground truth labels, we gain a nuanced understanding of our model's ability to classify image pixels accurately. .[9]

| | Class | Recall | Precision | Specificity | IoU | TDR | F1-Score |
|---|---|---|---|---|---|---|---|
| 0 | All Classes | 0.89 | 0.94 | 1.0 | 0.85 | 0.89 | 0.91 |
| 1 | Class 1 | 1.0 | 0.99 | 1.0 | 0.99 | 1.0 | 0.99 |
| 2 | Class 2 | 0.98 | 0.98 | 1.0 | 0.96 | 0.98 | 0.98 |
| 3 | Class 3 | 0.89 | 0.81 | 1.0 | 0.74 | 0.89 | 0.85 |
| 4 | Class 4 | 0.86 | 0.91 | 1.0 | 0.8 | 0.86 | 0.88 |
| 5 | Class 5 | 0.4 | 0.82 | 1.0 | 0.37 | 0.4 | 0.54 |
| 6 | Class 6 | 0.8 | 0.92 | 1.0 | 0.75 | 0.8 | 0.86 |
| 7 | Class 7 | 0.99 | 0.99 | 1.0 | 0.98 | 0.99 | 0.99 |
| 8 | Class 8 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 9 | Class 9 | 0.99 | 0.98 | 1.0 | 0.97 | 0.99 | 0.98 |
| 10 | Class 10 | 0.95 | 0.98 | 1.0 | 0.93 | 0.95 | 0.96 |
| 11 | Class 11 | 1.0 | 1.0 | 1.0 | 0.99 | 1.0 | 1.0 |
| 12 | Class 12 | 0.95 | 0.92 | 1.0 | 0.88 | 0.95 | 0.93 |
| 13 | Class 13 | 0.77 | 0.93 | 1.0 | 0.73 | 0.77 | 0.84 |

**(a) Evaluate predicted segmentations of the training images**

| | Class | Recall | Precision | Specificity | IoU | TDR | F1-Score |
|---|---|---|---|---|---|---|---|
| 0 | All Classes | 0.89 | 0.94 | 1.0 | 0.85 | 0.89 | 0.91 |
| 1 | Class 1 | 1.0 | 0.99 | 1.0 | 0.99 | 1.0 | 0.99 |
| 2 | Class 2 | 0.98 | 0.98 | 1.0 | 0.96 | 0.98 | 0.98 |
| 3 | Class 3 | 0.89 | 0.81 | 1.0 | 0.74 | 0.89 | 0.85 |
| 4 | Class 4 | 0.86 | 0.91 | 1.0 | 0.8 | 0.86 | 0.88 |
| 5 | Class 5 | 0.4 | 0.82 | 1.0 | 0.37 | 0.4 | 0.54 |
| 6 | Class 6 | 0.8 | 0.92 | 1.0 | 0.75 | 0.8 | 0.86 |
| 7 | Class 7 | 0.99 | 0.99 | 1.0 | 0.98 | 0.99 | 0.99 |
| 8 | Class 8 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 9 | Class 9 | 0.99 | 0.98 | 1.0 | 0.97 | 0.99 | 0.98 |
| 10 | Class 10 | 0.95 | 0.98 | 1.0 | 0.93 | 0.95 | 0.96 |
| 11 | Class 11 | 1.0 | 1.0 | 1.0 | 0.99 | 1.0 | 1.0 |
| 12 | Class 12 | 0.95 | 0.92 | 1.0 | 0.88 | 0.95 | 0.93 |
| 13 | Class 13 | 0.77 | 0.93 | 1.0 | 0.73 | 0.77 | 0.84 |

**(b) Evaluate predicted segmentations ofthe training images the test images**

**Figure 2.10: Comparison of Predicted Segmentation Evaluation Metrics on the training images and test images**

To organize and analyze these metrics effectively, we construct a dataframe containing rows for each class and columns representing the evaluation metrics. This dataframe facilitates a detailed examination of model performance during both training and testing phases, allowing us to identify areas of strength and areas for improvement across different semantic classes.

While metrics like precision, recall, and accuracy provide valuable insights into performance, relying solely on these metrics may not provide a comprehensive understanding of real-world effectiveness, particularly for complex systems like self-driving cars. To truly evaluate the efficacy of our semantic segmentation model in real-world scenarios, we utilized a simulator. Using the CARLA simulator, we simulated driving experiences to observe the model's performance under diverse conditions, closely resembling those encountered on actual roads. This approach enabled us to accurately assess the model's capabilities in scenarios mirroring real-world driving environments.

## 2.4 Experimentation and Results

The integration of the lane following model within the reinforcement learning (RL) framework is critical for enabling the autonomous car to navigate roads effectively. The lane following model serves as a decision-making component, analyzing observations from the environment (e.g., camera images) to determine appropriate steering actions. These steering actions are then incorporated into the RL agent's action space, allowing it to learn optimal driving behaviors through trial and error. The reward structure evaluates the effectiveness of the lane following model, providing feedback based on the agent's ability to stay within lanes, avoid collisions, and achieve driving objectives.

### 2.4.1 Testing and Validation with CARLA

CARLA's role in testing and validating autonomous driving algorithms cannot be overstated. By offering a realistic yet controlled simulation environment, developers can rigorously assess the performance of their algorithms under diverse conditions. Through CARLA, users can configure sensor suites, define traffic scenarios, and execute simulations with precision, enabling comprehensive evaluation of perception, planning, and control algorithms. The platform's fast simulation mode allows for rapid iteration and testing, while its extensive sensor models provide access to privileged information essential for algorithm refinement. CARLA's integration with ROS facilitates seamless interaction with external systems, streamlining the development and testing process further. Using this PythonAPI, points belonging to the map can be obtained easily, as well as actual vehicle position and speed. These points are called spawn points and they are determined by how the map was created. But what is really important is that CARLA also provides waypoints, which do not depend on how the map was built, and can be used for navigation. The simulator contains a path planning algorithm, both global and local. The global route planner is based on A* algorithm, and is able to built a route between two map points and returning the set of waypoints that joins them, forming a path.

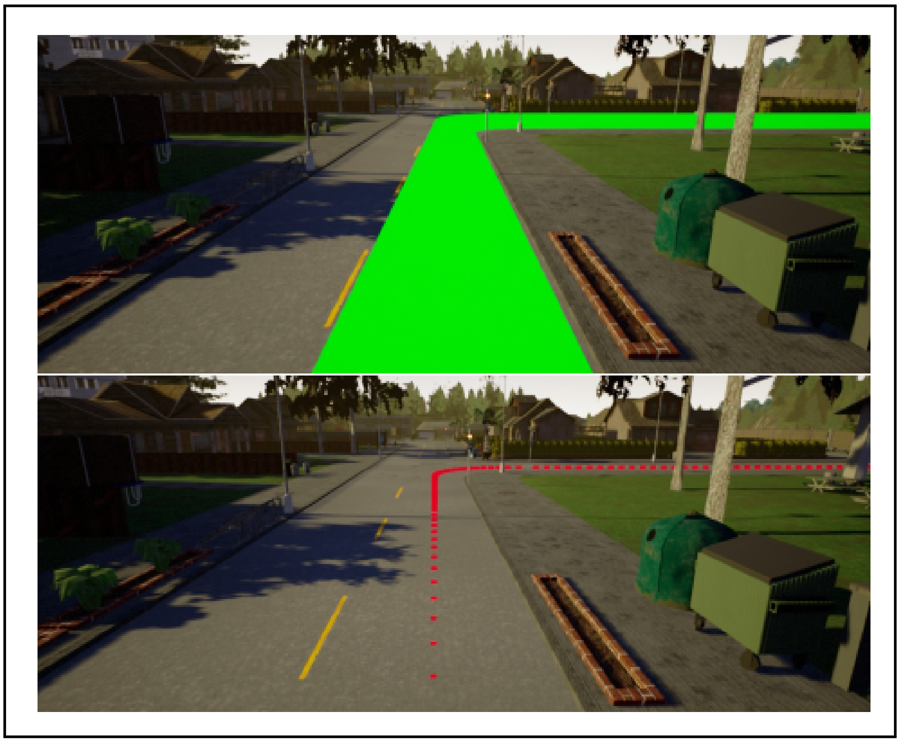


**Figure 2.11:** ***Route with waypoints***

In our project, we harnessed CARLA's versatile environment to craft custom scenarios tailored to our research objectives. Leveraging the platform's flexibility, we simulated diverse urban landscapes, each presenting unique challenges for our autonomous driving algorithms. From dense traffic scenarios to adverse weather conditions, CARLA enabled us to meticulously test and refine our algorithms in a controlled yet realistic setting. Utilizing CARLA's comprehensive documentation and recommended reading materials, we navigated the platform with ease, leveraging core features such as actors, sensors, and the Traffic Manager to create dynamic and immersive simulations. We seamlessly incorporated our algorithms into the simulation environment, enabling comprehensive testing and validation of our autonomous driving system.

To test the performance of the Agents, we compare results with other navigation modes such as manual driving (hand-crafted) and autonomous driving using a Classical Waypoint Tracking Controller The experiment carried out consist on learn how to drive in a trajectory where a origin-destination pair is selected that the trajectory includes straight sections and curves to the left and right.

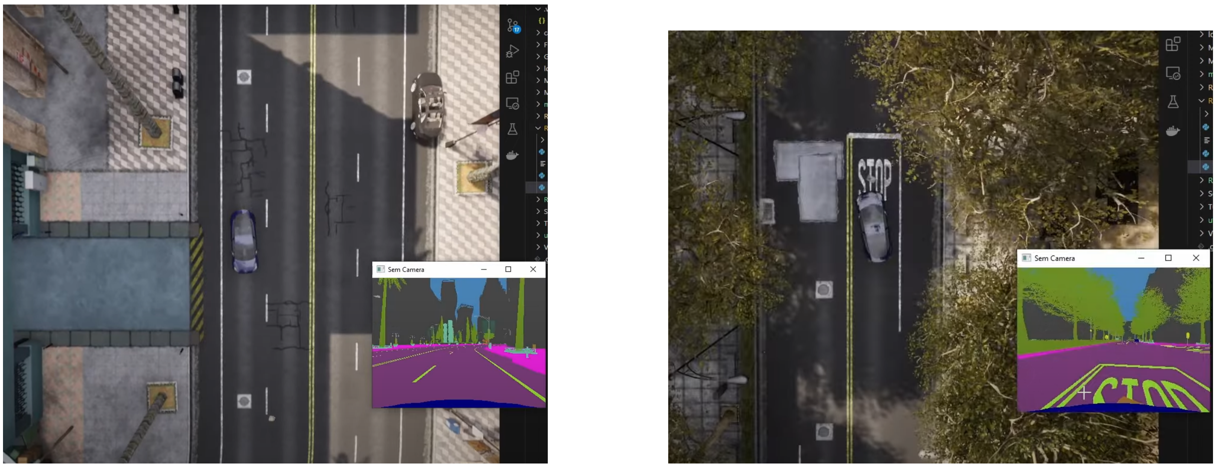



**Figure 2.12: Point of view from our perspective and from the car's perspective**

1) At the beginning of each episode, the CARLA Python-API generates a path using a global planner (origin, destination and waypoints).

2) Each timestamps (step), the corresponding visual and driving features vector are obtained, which are introduced in the agent.

3) In each timestamp of the agent training process, the agent weights are updated, the reward is calculated.

4) The episode ends when the destination is reached or a collision or lane departure occurs.

5) Throughout the process, the accumulated reward is calculated. This value will be used to determine which episode has made the best trajectory (bestreward-episode).

6) Training ends when the maximum number of episodes is reached.

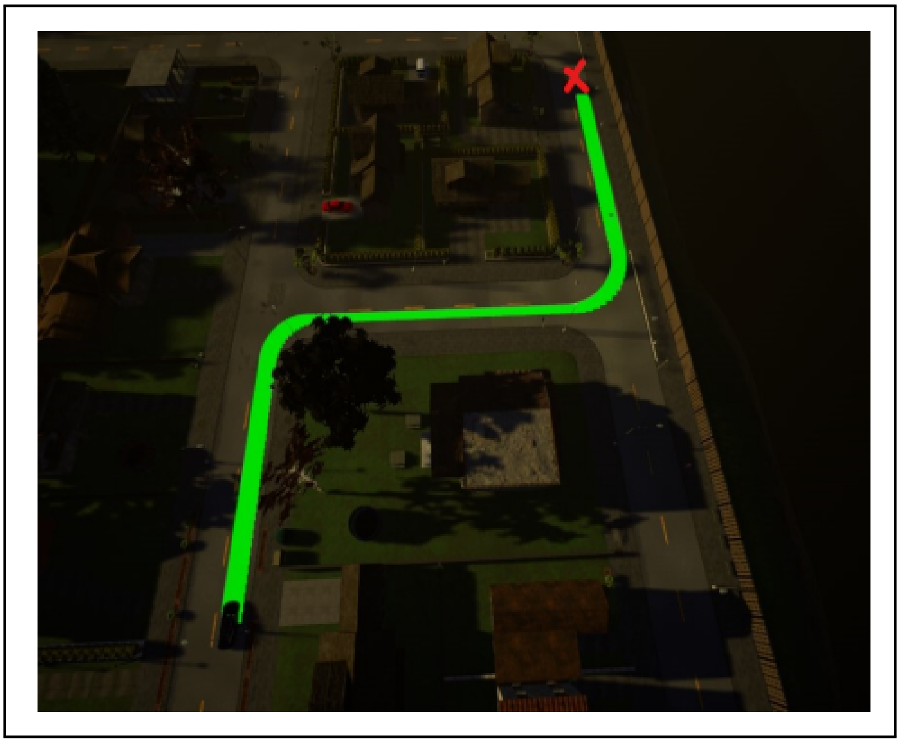


**Figure 2.13:** ***Navigation-trajectory on CARLA's obtained by A* algorithm provided by Carla PythonAPI.***

## 2.5 Conclusion

In conclusion, this chapter has offered a comprehensive examination of the integration of reinforcement learning with semantic segmentation and lane following models in the domain of autonomous driving. By employing the Carla simulator, we have effectively showcased the efficacy of our approach in simulated driving environments.Our journey commenced with the development and training of reinforcement learning algorithms, enabling autonomous vehicles to acquire optimal decision-making strategies through environmental interactions. Leveraging reinforcement learning techniques has endowed our models with adaptability and resilience, essential attributes for navigating intricate driving scenarios.

Moreover, our semantic segmentation and lane following models have emerged as pivotal elements in enhancing the perceptual and navigational capabilities of autonomous vehicles. Utilizing advanced methodologies in computer vision and deep learning, these models accurately interpret the surrounding environment, facilitating safe and efficient navigation on roadways.

The evaluation of our models through various metrics has offered valuable insights into their performance and efficacy. Through rigorous testing within the Carla simulator, we have gained valuable knowledge regarding the strengths and limitations of our models.

Looking forward, the insights gleaned from this chapter serve as a solid foundation for further advancements in autonomous driving research and development. With ongoing refinement and innovation, we are poised to overcome existing challenges and propel the field towards a future of safe and efficient autonomous transportation.

# GENERAL CONCLUSION

In this project, we embarked on a comprehensive exploration of artificial intelligence (AI) applications in the domain of self-driving cars, aiming to enhance their perception and decision-making capabilities. Through the development of semantic segmentation and lane detection models, we addressed critical aspects of scene understanding and navigation essential for autonomous vehicles. These models leverage state-of-the-art techniques in computer vision and deep learning to accurately identify objects, delineate road boundaries, and interpret complex traffic environments.

Furthermore, our integration of reinforcement learning within the Carla simulator enabled us to evaluate and refine these models in a simulated yet realistic environment. By subjecting our AI systems to diverse scenarios and training them to navigate complex urban landscapes, we simulated real-world driving experiences and honed their adaptability and robustness.

The semantic segmentation models played a pivotal role in scene understanding, enabling the vehicle to perceive and categorize objects with precision. This capability is crucial for safe navigation, as it allows the autonomous vehicle to recognize pedestrians, vehicles, traffic signs, and other obstacles in its vicinity. Similarly, the lane detection models provided essential spatial awareness, allowing the vehicle to maintain its position within designated lanes and execute smooth and safe maneuvers.

Through the synergy of these AI-driven technologies, our project advances the frontier of autonomous driving by addressing fundamental challenges in perception, navigation, and decision-making. By harnessing the power of AI, we pave the way for safer, more efficient, and ultimately, more accessible transportation solutions. As we continue to innovate and refine our models, we contribute to the realization of a future where self-driving cars revolutionize mobility and redefine the way we experience transportation.

# BIBLIOGRAPHY

[1] **"Self Driving Cars: All You Need to Know."** Elizabeth Spoehel and Shankar Banik. "Self Driving Cars: All You Need to Know." *IEEE*, 2022.

[2] **ObjectAug: Object-level Data Augmentation for Semantic Image Segmentation**. Jiawei Zhang, Yanchun Zhang, Xiaowei Xu. Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University, Shanghai, China. Available online: https://arxiv.org/pdf/2102.00221.pdf.

[3] **Importance-Aware Semantic Segmentation for Autonomous Vehicles**. IEEE Transactions on Intelligent Transportation Systems, Vol. 20, No. 1, January 2019, pp. 137. Bike Chen, Chen Gong, Member, IEEE, Jian Yang, and Member, IEEE.

[4] **CARLA: An Open Urban Driving Simulator**. Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio López, and Vladlen Koltun. Intel Labs, Toyota Research Institute, Computer Vision Center, Barcelona.

[5] **"Reinforcement Learning, Fast and Slow."** Matthew Botvinick, Sam Ritter, Jane X. Wang, Zeb Kurth-Nelson, Charles Blundell, and Demis Hassabis. *Trends in Cognitive Sciences*, vol. 23, no. 5, pp. 408-422, 2019.

[6] **U-Net: Convolutional Networks for Biomedical Image Segmentation**. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. arXiv:1505.04597v1 [cs.CV], 18 May 2015. Computer Science Department and BIOSS Centre for Biological Signalling Studies, University of Freiburg, Germany.

[7] **Synthetic Dataset Generation for Adversarial Machine Learning Research**. Xiruo Liu[*], Shibani Singh[*], Cory Cornelius[1], Colin Busho[2], Mike Tan[2], Anindya Paul[1], Jason Martin[1].

arXiv:2207.10719v1 [cs.CV], 21 Jul 2022. https://github.com/carla-simulator/carla/pull/4992.

**AI In Self Driving Cars**

**Hanine Mahmoudi**
**Mohamed Aziz Ben Lazreg**

**Résumé:**

Notre projet vise à explorer les applications de l'intelligence artificielle (IA) dans le domaine des voitures autonomes, en mettant particulièrement l'accent sur l'apprentissage par renforcement et la prise de décision. Nous avons développé des modèles de segmentation sémantique et de détection de voies pour améliorer la compréhension de l'environnement par les véhicules autonomes. Ces modèles utilisent des techniques de pointe en vision par ordinateur et en apprentissage profond pour identifier avec précision les objets et délimiter les limites de la route.

**Mots clés:**Intelligence artificielle (IA), Véhicules autonomes, Apprentissage par renforcement, Segmentation sémantique, Vision par ordinateur, Apprentissage profond.

**Abstract :**

Our project aims to explore the applications of artificial intelligence (AI) in the field of autonomous vehicles, with a particular focus on reinforcement learning and decision-making. We have developed semantic segmentation and lane detection models to enhance the understanding of the environment by autonomous vehicles. These models use cutting-edge techniques in computer vision and deep learning to accurately identify objects and delineate road boundaries.

**Key-words:** Artificial intelligence (AI), Autonomous vehicles,Reinforcement learning, Semantic segmentation, Computer vision, Deep learning.

**تلخيص:**

يهدف مشروعنا إلى استكشاف تطبيقات الذكاء الاصطناعي في مجال السيارات ذاتية القيادة، مع التركيز بشكل خاص على التعلم التعزيزي واتخاذ القرارات. لقد قمنا بتطوير نماذج لتقسيم الدلالي واكتشاف المسارات لتعزيز فهم البيئة من قبل السيارات الذاتية القيادة. تستخدم هذه النماذج تقنيات متطورة في رؤية الحاسوب والتعلم العميق لتحديد الكائنات بدقة وتحديد حدود الطريق.

**الكلمات المفاتيح:** الذكاء الاصطناعي ، السيارات ذاتية القيادة، التعلم التعزيزي، نماذج لتقسيم الدلالي، مجال رؤية الحاسوب، التعلم العميق.