Machine Learning for Autonomous Vehicle Navigation in Unstructured Environments

By Dr. Raquel Basu

Associate Professor of Information Systems, National University of Singapore (NUS)

1. Introduction to Autonomous Vehicle Navigation

As we advance towards creating vehicles and robots that can operate over a wide array of environments both indoors and outdoors, the emphasis in autonomous vehicle navigation is moving away from conventional control based architectures towards using machine learning models. These models are designed with the capability to learn from first principles and hence can be applied across a broad array of scenarios. In a review by Dong et al., the authors have noted that recent advancements in machine learning have improved the capabilities of autonomous navigational systems in detecting objects, understanding the semantic map of the environment, and planning and executing optimized path trajectories. However, the mechanisms for safety and human-like interpretability of these models requires further attention. Furthermore to improve and encourage further advancements in machine learning based autonomous navigation, the authors have proposed a standardization of datasets and benchmark environments. Deep learning models can also be used for robot navigation in agricultural settings wherein we can design conservation tillage equipment on top of autonomous field robots to detect row crops and end-rows using image-based classification and detection algorithms. For these deep learning based systems to work in outdoor agricultural settings, they must overcome environmental challenges such as varying terrain, weather, and illumination conditions.Researchers have also proposed a navigation model that is fully based on learning-based strategies compared to the traditional offline training method to improve robot navigation by reducing redundancy.

[1] [2]With the advancements in deep learning models and sensors, there has been significant progress in developing systems to navigate robots and autonomous vehicles [3] in complex unstructured environments without being reliant on any form of infrastructure. Vision-based navigation in particular has evolved from pre-designed control algorithms to learning-based

systems. Deep learning techniques such as convolutional neural networks (CNN), reinforcement learning, and imitation learning are being actively developed to accomplish the task of autonomous navigation.

1.1. Challenges in Unstructured Environments

embodiment and bootstrapping of 'common sense' in the term of dynamics models can be achieved seamlessly with learning paradigms. Deep learning neural networks can be trained with supervised, reinforcement, as well as self-supervised learning strategies over a variety of raw sensory inputs to inherently capture and encode high-order complex system dynamics. Classification learning over raw sensory input sequences can capture binary constraint regions on joint angular velocities (input changes), classifying them as safe or unsafe. This can be directly exploited to identify desired system basins-of-attraction where stable navigation is intended. Modeling continues over entire system state trajectories, which can directly capture the convex or non-convex-nature of planned/visible paths. System identification of unknown/ill-defined system-parameters, such as wheel–terrain compliance, along with camera intrinsic and wheel locations, can be directly learned from labeled image-action pairs. These learned self-supervised models can be directly learned from labeled image-action pairs. These learned self-supervised models can be far more accurate in modeling the system than traditional explicitly defined dynamical models. Hence these learning-based paradigms can achieve effective navigation control in unknown environments, without requiring any explicit human cognitive understanding of robot-environment dynamics.

Among various learning-based navigation strategies, deep learning has especially shown immense significance as it can autonomously learn representations in complex real-world environments, leading to better generalizability, without conducting an explicit understanding of the surrounding environment or identifying explicit constraints on robot dynamics. Especially in the past decade, the substantial increase in learning capabilities of neural networks has led them to solve complex tasks in human-level performance. This has pushed researchers to build learning-based control frameworks where learning policies are utilized to directly map raw sensory observations to an action, without building explicitly specified robot models [4].

Autonomous navigation in unstructured outdoor environments, such as non-paved, off-road, natural terrains, poses significant challenges due to complex terrain and environmentally induced disturbances [5]. While robots can consistently navigate in structured environments using path planning algorithms, these traditional approaches are limited in challenging unstructured terrains due to uncertain terrain dynamics, such as unpredictable disturbances experienced during motion (e.g., lateral slipping, rover rollover, terrain sinkage) and nonlinearity introduced by wheel–terrain interactions. Inspired by the brain, computation can be both planned and learned. In the context of off-road navigation, learning-based control strategies can be more adaptive and able to account for complex environmental factors impacting robot dynamics.

2. Fundamentals of Machine Learning

In recent years, instead of solving the perception of the environment, then planning and then finally controlling the motion, a method called end-to-end learning has been favored. In the autonomous navigation context, the drone or robot is trained to navigate the environment entirely based on sensory feedback using a suitable controller [6]. By removing this hierarchical structure in the collected training data by training the network to directly map input stimulus with an appropriate labeled output signal, researchers have shown remarkable success. Noticeably, works have include the usage of convolutional neural networks (CNNs), recurrent neural networks (RNNs) and long short-term memory networks (LSTMs) for this purpose, in most cases.

[7] [3]Classically, the autonomous navigation framework for robots entails perception, motion planning, and control. Perception consists of determining object presence, movement, or both within the environment through sensory feedback. A robot then processes this information to codify a trajectory for its movement throughout the environment referred to as motion planning. Finally, the robot has to execute the designed trajectory with the use of control systems. This hierarchy of action has created a structured process for researchers in the field to target their efforts, aiming at developing a solution for each subsystem.

2.1. Supervised Learning

One extensively studied application of supervised learning in the machine learning and control literature is Dyna-Net. Dyna-Net is an extrapolation method, which is shown to effectively predict unexplored paths even in unfamiliar environments by learning a dynamics model. Although being able to predict multiple steps into the future, an important limitation of Dyna-Net is that the quality of planned paths is high for only a short while until longer range multiple steps ahead predictions start diverging away from the ground truth. In this work, we improve the long-term utility of machine learning in uncertain environments by ensuring controlled exploration and reducing model error by improving the state representation. We consider the simple problem of point-to-point navigation, i.e., training the policy to perform two DOF planar navigation by supervised learning using synthetic data generated in an on-the-fly simulated environment.

Recent machine learning literature on autonomous navigation in unstructured environments focuses mostly on supervised, reinforcement, or imitation learning paradigms. Supervised learning is, perhaps, the most common approach, in which the primary learning objective is navigation safety. Approaches based on supervised learning often employ Uncertainty-Aware Learning to predict confidence or error estimates in environmental and control prediction models. Such methods can provide a principled mechanism to handle activestate uncertainty and can be used to generate more conservative steering commands. Training data in [ref: 459b9914-5ab9-4eb4-9393-e4b659b4aa0f, ref: 6b3ae222-a55a-4e72-af67-38b529946045] embodying supervised learning signals is necessary, but is it sufficient to address the complicated and uncertain nature of control and perception in outdoor terrain. These methods often become conservative and utilize rules that are agnostic to the dynamic and uncertain nature of outdoor terrain. Thus, models may require knowledge of real-time terrains so that possibilities of traversable and non-traversable terrains are updated in real-time [8].

2.2. Unsupervised Learning

Reinforcement learning (RL) is an aspect of machine learning that correlates to a specific subarea of artificial intelligence, i.e., the study of algorithms and their implementations allowing algorithms to reach autonomous decision-making. Its main characteristic is the explicit emphasis on achieving long-run goals that are not known at the time of learning, but whose optimal setting can be learned through multiple states over time. Autonomous vehicles, for instance, benefit from reinforcement learning because they can learn optimal plans without requiring large training data sets, human demonstrations, or manually designed controllers by experts. The general framework of RL has three main components for training an autonomous system: the sensor modality (such as cameras or lidars), the action space that the vehicle should choose from, and a reward signal indicating the achievement of the task. Unlike imitation learning, agents receive a global scalar feedback at sub-optimal states through simulation of the environment. So, RL can potentially surpass the skills of human drivers if provided with proper simulators [6]. Another method in unsupervised learning for robotic navigation is the use of latent variable models (LVM). The idea of unsupervised learning is to learn a so-called generative model of the training data (i.e., the states of the environment available in the logs). This generative model tries to explain the data as well as possible, such that it can be used for various purposes, such as imputing sensor measurements, future state prediction, or estimating entities of interest, such as world representations. This assists in learning high-level abstractions about the environment, a crucial capability for solving tasks that far outpace hand-engineered sensors and representations. The remaining challenge is to show that learned representations generalize to novel scenes. This is important for transfer from the training to the test setup, and may generalize across different environments. Using the learned latent space, model-based LVMs can outperform model-free approaches in terms of sample-efficiency by allowing the generation over multiple future steps. Given that learned models do not overfit, they can provide long-term predictions with a reduced probability of divergence that usually harms the effectiveness of model-based methods. Furthermore, they can be used for interpreting the reasoning behind the overall behavior of the system at each step of the execution [4].

2.3. Reinforcement Learning

RL has been increasingly used in high-dimensional continuous state and action space and proves to be capable of learning robust navigation policies autonomously [6]. Underlying RL methods have been significantly evolving, especially in recent years with the successful application of deep neural networks. This super trend also results in the success of numerous applications, such as image classification, object detection and tracking, video game playing, and autonomous system control. In particular, deep reinforcement learning (DRL) has shown significantly superior performance in several challenging environments, where traditional RL algorithms could not operate well due to the curse of dimension and state-action distribution, as well as the complexity of real world sensory outputs. A DRL algorithm can autonomously learn a high-quality navigation policy by interacting with environment, and offers the massive potential of adaptive learning and superior generalization over expertwrittens scripts. As sated by related works, deep reinforcement learning based planning and control methods are now attracting rising attention by harness ing simulation environments for data-driven learning. Chaplot et al. presented a way to train a DRL algorithm in a car-following environment with the look and lead strategy, which can be conducted in an open societal-district simulation platform (OpenDS) [9].

Self-driving cars rely on highly detailed maps for localization and path planning, which are not suitable for unstructured terrains. Mapless navigation methods have better adaption to environmental changes [10]. Deep neural networks (DNN) can generate task-specific control signals by learning to mimic human expert controls, allowing the policy to function on its perceptions. However, this imitative/path imitation learning has various shortcomings such as sample complexity and lack of exploration. Both sample efficiency and exploration problems could be handled by reinforcement learning (RL) algorithms. In addition, the neural network policy involved in the RL algorithm learns to predict the value and optimizes parameters by temporal-difference learning, which is different from the gradient fitting performed in supervised learning. The neural network function can be trained according to real-world driving attributes, and RL method could improve the generalization of the system, and as well as the model-based and model-free combination, allowing trainings in complex, interconnected tasks.

3. Sensors and Perception in Autonomous Vehicles

In the context of sensing in autonomous vehicles, it is sufficient to mention proprioceptive and exteroceptive sensors. Control by exteroceptive sensors is a somewhat recent trend. This has to some extent been due to the availability of faster computation and cheaper high-speed external sensors for environment perception. The sensor systems of autonomous vehicles operating in familiar as well as unfamiliar terrains are very different. Those vehicles operating in unknown environments are called pioneer systems. The function of artificial perception is to gather that information of the surroundings of a vehicle at any point in time. This data assists in checking the current status of a vehicle. Depending on this information, a vehicle can take correct decisions, or the perception information can be observed by human experts ([7]).

Sensors are an integral part of any intelligent system. In the case of autonomous vehicles, sensors are used for both perception and localization. The knowledge of the surroundings determines how the vehicle should navigate through the environment, and this is based on sensor readings. The primary sensors used for perception and navigation are vision cameras, LiDAR (Light Detection and Ranging), radar, ultrasonics, inertial measurement units (IMUs), and wheel encoders ([11]). Among the localization systems, the primary ones are CRNs (Coordinate Reference Networks), GPS (Global Positioning System), and LiDAR. A few vehicles and research projects use the V2X (Vehicle-to-Everything) communication too. In this section, we detail the various types of sensors and perception methods to understand the environment of an autonomous vehicle

3.1. Lidar Sensors

The cameras are perhaps the most versatile sensors one might put on board an autonomous vehicle. In fact, their data (images) recollect a great amount of (object) information that, consequently, allows to accomplish various tasks such as classification, object detection, semantic and instance segmentation and interpolation of objects of interest. The data acquired by cameras is also the principal or complementary input to most map-based localization or SLAM algorithms. A radar is the fastest and most robust sensor among the three for sensing information by processing the reflection of radio waves sent. However, they are among the worst sensors in terms of their marginal accuracy and the amount of information they can provide. LiDAR sensors are an intermediate solution in terms of the three main sensors in terms of accuracy and margin of error. They are also active sensors, which reproduce laser beams and precisely measure distances [12]. Shaik, Mahammad, et al. (2020) explore user privacy in decentralized identity management using ZKPs and anonymization.

Sensors in autonomous vehicles are classified into two classes: proprioceptive and exteroceptive, where the former only detects the vehicle and related events (e.g., temperature, speed, battery status, and so on). On the other hand, the exteroceptive sensors acquire environmental information that can be seen as landmarks in autonomous vehicle applications. As a result, they are also termed the vehicle's eyes on the road.Template:P.S. Many algorithms involved in autonomous vehicles work on and rely on the data provided by these sensors. These include, among others, perception, localization and path planning. Among the sensors often used in the context of autonomous vehicles, cameras, radar sensors and LiDAR sensors play fundamental roles, even if different applications and trade-offs suggest (also) to use (slightly) different sensors and, possibly, to integrate their data [11]. A brief overview of the main sensors is given in what follows.

3.2. Camera Systems

Autonomous vehicles need to perceive the road and surround environment in realtime in order to well navigate. In the last years, deep learningbased methods have significantly boosted object detection, identification, depth estimation, and semantic segmentation tasks. Convolutional neural networks (CNNs) are in particular subtracting a most effective way to perform these kind of tasks, overcoming more classic techniques such as HOG or sift. The recovered 3D bounding boxes for each detected object must be projected in a 2D image in order to be included as a new input for the system. To accomplish this, it is mandatory to have a geometric model of the camera included in the detections pipeline. Here we pose what is to our knowlegde the first attempt in the literature to perform this 2D projection using the stereographic projection in the spherical coordinates of the fisheye's view [13].

[14] [15]The camera system has become a key component in autonomous vehicles, being a cheap, compact, and lightweight sensor, providing rich color information to assess the correct state of traffic lights, road signs, traffic cones, and other participants, also providing important information regarding the surrounding environment in order to avoid potential collusions. Object detection systems are used as the basic unit to get a most complete understanding of what is going on in the shared environment, allowing the system to know in advance about potential collisions or obstacles. In order to provide orientation to the autonomous vehicle, several approaches can be found in the literature. Some of these approaches are mainly focused on object detection, others use also semantic segmentation annotations. However, if a complete and precise 3D scene understanding would be provided, additional relevant information for the autonomous vehicle can be obtained; for example, once the 3D bounding box of an object is obtained, the distance between the object and the autonomous vehicle can be easily calculated. The use of 2D object detection frameworks combined with 3D information provided by the external sensors or through a post-processing stage can allow to obtain a complete object pose in the 3D world.

4. Deep Learning for Autonomous Navigation

By designing larger and larger neural networks, it has been possible to achieve promising results in various learning tasks [16]. However, since the task is inherently sequential, these agents must quickly learn to generalize to unseen state-action pairs. Unfortunately, this type of learning does not scale well to high-dimensional input, and often requires a large amounts of human supervision to provide good training data. The traditional approach to a task as complex as navigating an environment from high-dimensional sensory input is to start from a model-free learning algorithm, which learns a policy from the environment's observations and rewards. However, there have recently been several attempts to approach this task in more of a model-based reinforcement learning fashion.

Autonomous navigation is crucial for robots to perform tasks in human environments where a priori maps and GPS signal are not accessible [17]. Emerging deep learning (DL) algorithms can be used to train and deploy networks for this purpose. This paper provides a comprehensive review of the existing deep learning models applied to robot navigation. We discuss decision-making solutions, the neural network architectures employed, and a survey on the sensors used in navigation. First, we review the conventional methodologies of navigation pipelines in robotic systems. Subsequently, we introduce deep learning-based perceptions and decision-making networks and discuss trained models used for robot navigation.

4.1. Convolutional Neural Networks

An important advantage of DNNs is their ability to accurately capture features at multiple scales in an image. This is achieved by design. The input passes through multiple stages of filtering, feature selection and ”pooling” to form a spatial hierarchy in the network. This property particularly suits image tasks, as sensitive features such as texture and finegrained shapes are accurately captured by lower-level filters, while higher-level ones focus on broader patterns and shapes. This makes it more conducive to predicting the 2.5D or 3D structure of the observed scene using monocular data. As a result, CNN based methods have been very effectively used for the subsequent task of depth estimation.

Obtaining an abstract representation of the environment from monocular image inputs is a crucial task in autonomous navigation systems [18]. Such a representation, often referred to as semantic segmentation, infers object pixels from within the vehicle's field of view. Semantic segmentation is attracting a considerable amount of interest in autonomous navigation. It solves the goal of perceiving obstacles in the scene from raw pixels, allowing for high-level decision-making processes to be designed on top of it. Furthermore, runtime performance can be optimized to be dependable for the learning process.

4.2. Recurrent Neural Networks

Vehicles often need to navigate to the goal while continuously avoiding the potential obstacles in the environment despite changes. This information for trajectory prediction may also depend on a large history of input–output sequence data. Due to its ability to capture the dynamic response of the system well for a large sequence of sequence data, the authors mainly investigate the use of an RNN [19]. The researchers implement a simple trajectory prediction task for the WiBo-NRS training architecture. The system uses two networks, that is, a WiBo and an RNN, to learn human–robot interaction from the recorded human demonstrations. The resulting network is used to predict the forthcoming motion (WiBo) and the final collision estimator (NRS).

Autonomous systems that operate in unstructured environments are becoming very popular. In this context, the task of autonomous navigation for vehicles is being vastly studied. Most of the recent studies report the use of the machine learning-based approach [20]. The use of the machine learning for autonomous vehicle navigation in unstructured environments provides a solution independent of any pre-defined features, or the navigable path, or the availability of the environment map. It provides highly discriminative features for capturing the information required for the trajectory prediction process. The most common and simple visualization tool used for visualization using 1-DoF (one-degree-of-freedom) is the normal example (NE) space [16]. The NE space provides a 3-segment curve. As speed is increased, the data is reduced to a 2-segment curve. The result is further simplified to a 1-segment curve at high speeds. For low-speed control, there is no need to predict. The motion pattern can be observed in the video frame. This paper demonstrates that RNN can be trained to effectively predict the vehicle's motion trajectory using WiBo and NRS. A special loss function and a new normalization method are used to train the trajectory prediction model.

5. Path Planning Algorithms

There are various other machine learning based methods (such as OracleNet, MPNet, MPath, Policy Transfer Imitation, and AutoRL), where the authors use past experience and learning from demonstration strategies to obtain a collision-free path plan for system navigation without any prior knowledge of the underlying environment [21]. OracleNet performes inference through a recurrence relation established in a continuous space-time representation that models the occupancy of the environment. MPNet attempts to estimate the probability distribution of the next state given the current state and goal employing a probabilistic feedforward neural network that considers the environment conceptually abstract, which implies faster inference. Another approach, called "Learning to Navigate in Cities Without a Map", reportedly uses imitation learning and reinforcement learning to learn to follow recorded paths from humans driving in a real car and to navigate to a goal using only past navigation as a source for learning. Finally, "Learning Model Predictive Control for Vision-Based Autonomous Driving on the F1/10" implements a vision-based path planner for real-world autonomous driving.

Prior work on path planning in robotics has focused on different algorithm design paradigms, targeting various aspects of robot navigation as well as exploration and search in unknown environments. The classic A* search algorithm is widely used for computing shortest paths in discrete grids, where the computation time of this search algorithm is highly sensitive to the size of the environment. The A* algorithm is unable to provide optimal solutions in continuous state spaces, which need to be embedded into discretized graph representations that can be time consuming [22]. Sampling-based algorithms, such as Probabilistic Roadmap Methods (PRM) and Rapidly-exploring Random Trees (RRT), are applicable for continuous domains and are well-suited for high-dimensional configuration spaces with constraints and nontrivial robot dynamics but are inherently offline in nature and thus not efficient to quickly solve for the next robot state practically.

5.1. A* Algorithm

The A* algorithm employs g(n) to represent the cost of the best path between O and node n found so far. While f(n) = g(n) + h(n) is the function representing the candidate for the cheapest path solution. h(n) is the heuristic function and its purpose is to estimate the cheapest cost of the path from current node n to the target node T. Different types of heuristics can be utilized, taking into consideration the potential of growth at any node. A simple choice is to apply the Manhattan method, for instance, and compute the estimated optimal cost of the global path between T and n. The Manhattan method produces the cost to be labeled by three never traveling diagonally, or dumb costs concerning the East, West, South, and North directions. Though a fast and reasonable heuristic, different sorts of effective heuristics can easily replace the Manhattan method. Furthermore, the greatest drawback of A* is the drive by the heuristic function and if this function is poorly chosen, then the effect of finding the solution will be poor. Selecting the efficient heuristic function is a major concern for reducing the search space to obtain success, accuracy and right solutions.

The A* algorithm [23] was first introduced in 1968 and is commonly used for collision-free navigation of mobile robots. A* is an extension of Dijkstra's algorithm with the advantage of exploiting a heuristic function, which makes it scalable for large environments. The input for A* is the grid map representing the world along with its legend, a graph representing the cells of the map, and origin and target nodes. A* generates a walk that begins at O and travels to T where the walk never overlaps with any grid cell.

5.2. Rapidly-exploring Random Trees (RRTs)

Full text is spelled out grammatically, coherent and informational.

Since the collision between two vehicle or between vehicle and obstacle leads to many accidents, it is extremely important to be able to identify obstacles and to perform path-planning in dynamic and unstructured environment [24]. It is very challenging that attaining maximum coverage and avoiding moving obstacles in different environment such as static, dynamic, uneven and structured environments is not easy due to the complexity in different types of environments. In order to overcome these difficulties in rapid, an efficient sampling-based path planner is introduced for large-scale or high-dimensional complex environments, known as rapidly-exploring random trees (RRTs) [25]. It is widely considered because the main work associated with Nearest Neighbor Search (NNS) and Collision Checking (CC) is offloaded to tree assembling which happens very differently from other path planning algorithms such as A*, D* or Dynamic Window Approach (DWA). A*. In RRT, a tree consists of nodes, which are the configuration of the robots and the edges which connects these nodes. Some advantages of RRTs over other sampling-based techniques, including PRM− and D* are that (I) once the tree is grown it guarantees to find the solution. However, on the basis of tree density the cost could be high or low, (II) tree primarily established to bias. RRT mainly founded to move quickly into unsearched areas, (III) for a given workspace dimension, RRT is spatial complexity is moderate and doesn't have to made observations in whole environment. It avoids large and computationally expensive observations [26]. With respect to these adavantages, by providing solution of the high-dimensional problems, many of the RRT modifications tries to implement successfully.

6. Simulators and Testing Environments

Real-Data to Sim Data A tracking and object classification pipeline implemented in used a simulation data-driven approach for synthetic data generation. The pipeline incorporates LIDAR and camera data and was mainly validated with real-world data recording at test-courses in Vigo, Spain, and Gteborg, Sweden. This approach could mark the beginning of a more converged workflow between reality-based data acquisition and synthetic data generation. It can be expected that the simulation data generation based on real-world data is an ideal use case for (possibly) unsupervised domain adaptation. i.e., the simulated images can be used to enhance a neural network trained for real-world test data. Other works leveraged the modularity of simulator frameworks to transfer learned representations from sensor data simulated in a basic digital twin to sensor data from a more detailed digital twin of a specific intersection or road section. This work represents a good way of combining the high-fidelity promise of game engines with the reproducibility of a simulation-based perception pipeline and the data-driven logic behind the perception module.

Machine learning-based approaches help in learning how to navigate in unstructured environments without explicitly hand crafting algorithms and representations [27]. Using machine learning, the onboard computer can process synthetic data generated by generic vehicle simulation software such as CarMaker. The synthetic data that is used for parameter tuning or for training machine learning models is simulated with animation software. Moreover, the recent release of open-source simulators designed for autonomous vehicle testing and research is of great interest to the development and research community. One example is AirSim developed by Microsoft that has an API integrated with the drone and racing car simulators [28]. A perception stack developed by Aptiv that has been fully integrated with Autoware was extensively used in [7]. The task of the perception stack is to make sense of the raw LIDAR, camera, and other sensor inputs and segment meaningful information from it for machine learning-based detectors, trackers, and classifiers.

6.1. CARLA Simulator

If the pedestrian reaches the sidewalk or the grass area it is removed from the simulation. Pedestrians are considered part of traffic. When a pedestrian appears in the close vicinity of the ego vehicle posing a potential danger, it issues a negative reward. To prevent the AV from traveling too fast reasons of safety and traffic law, the simulator controls the throttle of the ego vehicle. The car's speed should stay within the limits, otherwise a speed error reward is generated. In the case of an emergency braking situation, the AV receives an emergency stop command exceeding normal deceleration limits, resulting in a crash. Therefore, in addition to the maximum and minimum values of the legal advices, the speed regulation function limits its generation in the simulation environment.

The CARLA simulator [29] has been used for the training and evaluation of the full model. The CARLA simulator offers different environments, like the Town01 and the Town02, in which the AV system can interact with various traffic scenarios. Vehicles in the simulation environment vary from small cars over lorries to bicycles and pedestrians. All in all, CARLA provides many stimuli that an autonomous vehicle can expect in a real-world environment. Different sensors, like RGB cameras, depth cameras, and LiDAR devices, are available to observe the environment. These sensors deliver the perceptual input that the AV should understand. An OpenDrive file, describing the road network of the Town01 environment [30], was used to plan the ego vehicle's route. College Lane has been chosen, as it is a long straight road with complex crossroad structures, traffic lights, and other road users that challenge perception modules. Using the provided Python API, which allows to interact with the environment, it is possible to place an autonomous vehicle on this route and to feed the simulated sensors with these data.

7. Ethical Considerations in Autonomous Vehicle Development

It is expected that the positive societal and economic effects of autonomous vehicles (AVs) will be utmost for the general public. However, these presumed benefits of AVs are also accompanied by numerous problems and issues that need to be resolved. The ethical dilemmas do not just only refer to the end users and their ethical preferences, but also any participant in any interaction has the duty to behave ethically and respect the rules and duties that have been established in the society, informed by its values, religion, norms, and standards [31]. This is also valid for the AVs. They should be designed and developed having the right doctrines, norms, rights and duties, standards in the mind of that society in which they are supposed to operate and require manufacturer accountability, transparency and allocation of all intellectual safety and security resources that are necessary in order to minimize risks and contribute to the successful introduction of AVs to human society.

Autonomous vehicles (AVs) are seen as the advancement of vehicle control systems, using a set of technologies and machine learning methods to control different functions in the vehicle, offering solutions for technically complex and comprehensive situations that are beyond traditional vehicle dynamics control tools. The development of AV technologies will improve cars' safety, road use efficiency, and opportunities for the safe and comfortable mobility for all users, but will also generate new challenges to ensure the global benefits of the new transportation system. One further important issue is the technological development background includes various hidden but possible risks, for example ethical dilemmas, in autonomous vehicles (AV) development, proving a major challenge for industries, OEMs, suppliers, and the whole AV ecosystem [32].

8. Future Directions and Emerging Technologies

Current research in real-time segmentation is focused on tuning the speed–accuracy tradeoff across the range of inputs from inexpensive consumer camera hardware to event cameras. Technique advances in this field are expected in terms of attainable ego- and object-vehicle velocities as well as optical signal range, both for day and nighttime operation of autonomous vehicles. The primary objective of Segmentation in Autonomous Vehicle Navigation (SAVN) is to enable robust and accurate estimation of perceptually homogeneous space regions from raw sensory inputs, where the regions of the road surface and its adjacent environment are prone to be regarded as semantic classes [33]. Future research topics in this area include a robust transfer from synthetic and day-time downstream semantic quality metrics, especially for the scenario with long and fast moving vehicles. The key to enhancing current visual place recognition (VPR) algorithms in open-set deployment environments is to establish the simultaneous localization and mapping (SLAM) frameworks that are robust to the presence of open-world semantic classes. In the case of the highly unconstrained operating environment, the current probabilistic approaches are prone to fail and indications of the unneeded map updates may occur in the most unexpected places. Future research in VPR quality in open-set scenarios would better focus on uplifting the VPR quality directly by addressing the problem of deployment environmental segmentation. One potential future direction in this context is to replace the semantic-based segmented appearance features in the GeoTrack/Image-SDHIV with more generic appearance features extracted by the pretrained visual generation model along with the corresponding semantic features.

The current literatures provide the evolution of machine learning algorithms for autonomous vehicles in the past decade. There are some challenges and limitations need to be addressed for machine learning-based autonomous navigation in unstructured environments, such as machine learning-based algorithms for automotive perception, planning, and control to deal with complex environments, obstacle modelling and prediction to enhance safety and reaction, and system failures resilience for reducing accidents and economic costs [34]. What are the promising strategies in autonomous navigation research? What future research in machine learning and computer vision is demanded? To answer these questions we discuss the future directions of machine learning-based autonomous vehicle navigation in unstructured environments in some direction. In the automotive perception field, semantic segmentation by using ML-based segmentation is the most popular in automotive segmentation [35].

References:

1. [1] A. Asgharpoor Golroudbari and M. Hossein Sabour, "Recent Advancements in Deep Learning Applications and Methods for Autonomous Navigation: A Comprehensive Review," 2023. [PDF]
2. [2] X. Li, J. Su, Z. Yue, and F. Duan, "Adaptive Multi-ROI Agricultural Robot Navigation Line Extraction Based on Image Semantic Segmentation," 2022. ncbi.nlm.nih.gov
3. [3] X. Xiao, B. Liu, G. Warnell, and P. Stone, "Motion Planning and Control for Mobile Robot Navigation Using Machine Learning: a Survey," 2020. [PDF]
4. [4] X. Xiao, J. Biswas, and P. Stone, "Learning Inverse Kinodynamics for Accurate High-Speed Off-Road Navigation on Unstructured Terrain," 2021. [PDF]
5. [5] J. Hyun Lee, J. Choi, S. Ryu, H. Oh et al., "Learning Vehicle Dynamics from Cropped Image Patches for Robot Navigation in Unpaved Outdoor Terrains," 2023. [PDF]
6. [6] L. Ge, X. Zhou, Y. Li, and Y. Wang, "Deep reinforcement learning navigation via decision transformer in autonomous driving," 2024. ncbi.nlm.nih.gov
7. [7] D. Calogero Guastella and G. Muscato, "Learning-Based Methods of Perception and Navigation for Ground Vehicles in Unstructured Environments: A Review," 2020. ncbi.nlm.nih.gov
8. [8] J. Shabbir and T. Anwer, "A Survey of Deep Learning Techniques for Mobile Robot Applications," 2018. [PDF]

9. [9] Z. Li, S. Yuan, X. Yin, X. Li et al., "Research into Autonomous Vehicles Following and Obstacle Avoidance Based on Deep Reinforcement Learning Method under Map Constraints," 2023. ncbi.nlm.nih.gov
10. [10] J. Hossain, "Autonomous Driving with Deep Reinforcement Learning in CARLA Simulation," 2023. [PDF]
11. [11] D. Jong Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review," 2021. ncbi.nlm.nih.gov
12. Tatineni, Sumanth. "Beyond Accuracy: Understanding Model Performance on SQuAD 2.0 Challenges." *International Journal of Advanced Research in Engineering and Technology (IJARET)* 10.1 (2019): 566-581.
13. Shaik, Mahammad, et al. "Enhancing User Privacy in Decentralized Identity Management: A Comparative Analysis of Zero-Knowledge Proofs and Anonymization Techniques on Blockchain Infrastructures." *Journal of Science & Technology*1.1 (2020): 193-218.
14. Vemoori, Vamsi. "Comparative Assessment of Technological Advancements in Autonomous Vehicles, Electric Vehicles, and Hybrid Vehicles vis-à-vis Manual Vehicles: A Multi-Criteria Analysis Considering Environmental Sustainability, Economic Feasibility, and Regulatory Frameworks." *Journal of Artificial Intelligence Research* 1.1 (2021): 66-98.
15. [15] B. Shahian Jahromi, T. Tulabandhula, and S. Cetin, "Real-Time Hybrid Multi-Sensor Fusion Framework for Perception in Autonomous Vehicles," 2019. ncbi.nlm.nih.gov
16. [16] A. Nguyen and Q. Tran, "Autonomous Navigation with Mobile Robots using Deep Learning and the Robot Operating System," 2020. [PDF]
17. [17] T. Ganegedara, L. Ott, and F. Ramos, "Learning to Navigate by Growing Deep Networks," 2017. [PDF]
18. [18] S. Yang, S. Konam, C. Ma, S. Rosenthal et al., "Obstacle Avoidance through Deep Networks based Intermediate Perception," 2017. [PDF]
19. [19] Y. Ed-Doughmi, N. Idrissi, and Y. Hbali, "Real-Time System for Driver Fatigue Detection Based on a Recurrent Neuronal Network," 2020. ncbi.nlm.nih.gov
20. [20] H. Li, L. Zhou, and A. Knoll, "BCSSN: Bi-direction Compact Spatial Separable Network for Collision Avoidance in Autonomous Driving," 2023. [PDF]

21. [21] H. Liu, Y. Shen, S. Yu, Z. Gao et al., "Deep Reinforcement Learning for Mobile Robot Path Planning," 2024. [PDF]
22. [22] D. Rava and R. Xu, "Explained Variation under the Additive Hazards Model," 2020. [PDF]
23. [23] Y. Yao, N. He, and M. Zhang, "The Application of Internet of Things in Robot Route Planning Based on Multisource Information Fusion," 2022. ncbi.nlm.nih.gov
24. [24] T. W. Kang, J. G. Kang, and J. W. Jung, "A Bidirectional Interpolation Method for Post-Processing in Sampling-Based Robot Path Planning," 2021. ncbi.nlm.nih.gov
25. [25] X. Zhang, T. Zhu, L. Du, Y. Hu et al., "Local Path Planning of Autonomous Vehicle Based on an Improved Heuristic Bi-RRT Algorithm in Dynamic Obstacle Avoidance Environment," 2022. ncbi.nlm.nih.gov
26. [26] C. Wang, J. Wang, C. Li, D. Ho et al., "Safe and Robust Mobile Robot Navigation in Uneven Indoor Environments," 2019. ncbi.nlm.nih.gov
27. [27] L. Liu, S. Lu, R. Zhong, B. Wu et al., "Computing Systems for Autonomous Driving: State-of-the-Art and Challenges," 2020. [PDF]
28. [28] A. Santara, S. Rudra, S. Aditya Buridi, M. Kaushik et al., "MADRaS : Multi Agent Driving Simulator," 2020. [PDF]
29. [29] M. Haris and A. Glowacz, "Navigating an Automated Driving Vehicle via the Early Fusion of Multi-Modality," 2022. ncbi.nlm.nih.gov
30. [30] J. Hanhirova, A. Debner, M. Hyyppä, and V. Hirvisalo, "A machine learning environment for evaluating autonomous driving software," 2020. [PDF]
31. [31] T. LaCroix, "Moral Dilemmas for Moral Machines," 2022. [PDF]
32. [32] L. Luxmi Dhirani, N. Mukhtiar, B. Shankar Chowdhry, and T. Newe, "Ethical Dilemmas and Privacy Issues in Emerging Technologies: A Review," 2023. ncbi.nlm.nih.gov
33. [33] Y. Tang, C. Zhao, J. Wang, C. Zhang et al., "Perception and Navigation in Autonomous Systems in the Era of Learning: A Survey," 2020. [PDF]
34. [34] F. Islam, M. M Nabi, and J. E. Ball, "Off-Road Detection Analysis for Autonomous Ground Vehicles: A Review," 2022. ncbi.nlm.nih.gov
35. [35] J. Kinnari, A. Thomas, P. Lusk, K. Kondo et al., "SOS-SLAM: Segmentation for Open-Set SLAM in Unstructured Environments," 2024. [PDF]