

Advancing Autonomous Ai Agents Through Deep Reinforcement Learning

Taushifah Ahmed Kazi^{1*}, Dr. Satish Kumar N²

1 Research Scholar, Sunrise University, Alwar, Rajasthan, India

tousif60@gmail.com

2 Associate Professor, Department of Computer Science, Sunrise University, Alwar, Rajasthan, India

Abstract: Thanks to Deep Reinforcement Learning (DRL), which allows computers to learn complicated behaviors via interaction with dynamic surroundings, autonomous AI agents have made remarkable advancements. This study takes a look at how deep reinforcement learning (DRL) algorithms, structures, and training approaches have improved autonomous agents' capacity for adaptation, decision-making, and generalization. With a focus on scalability, sample efficiency, and resilience in high-dimensional state spaces, we showcase important advancements in value-based, policy-based, and actor-critic techniques. This paper shows how DRL-driven agents may accomplish better autonomy and task performance and goes on to examine its uses in robotics, autonomous vehicles, natural language interfaces, and multi-agent coordination. Emerging solutions like as hierarchical RL, curriculum learning, model-based DRL, and safe RL frameworks are shown, along with challenges such as real-world transferability, reward engineering, and instability in training. In sum, the article highlights the revolutionary significance of DRL in molding the autonomous AI agents of the future and suggests avenues for further study to improve AI trustworthiness, interpretability, and human-AI harmony.

Keywords: Deep Reinforcement Learning, Autonomous AI Agents, Policy Optimization, Actor-Critic Methods, Robotics, Multi-Agent Systems, Model-Based RL, Safe Reinforcement Learning.

INTRODUCTION

One learning paradigm that incorporates many different algorithmic techniques is reinforcement learning (RL). Games like Backgammon and Atari, as well as well-supervised systems like robotics, have shown the efficacy of the algorithms based on stochastic approximation. Recent advances have allowed the application of RL algorithms to systems that receive image, video, or audio as input data for processing, but their adaption to actual real-world systems has been very sluggish. A combination of fast advancements in computing frameworks and infrastructure¹ and astute engineering has mostly allowed for these accomplishments. Nevertheless, the issue of quick deployment to real-world systems remains unresolved. The main reason for this, as we have reasoned out, is the huge gap between the well-defined facts of real-world systems and the experimental RL setups that are now in use. To make RL agents more widely used for regulating real-world systems, we can do two

things:(1) refine our algorithms so they don't oversimplify the system mathematically, and (2) make greater use of computer frameworks. We go into both of these areas extensively in this thesis.

A quick review of the mathematical foundations of RL is required to comprehend the algorithmic difficulties encountered in practical RL. The Markov decision process mathematical framework (Chapter 2 goes into depth) is the foundation of reinforcement learning. Many fields, including control engineering, finance, operations research, and communication networks, face the challenge of making optimum sequential decisions in the face of uncertainty. The Markov decision process (MDP) provides a popular framework for modeling these challenges. A Markov decision process (MDP) is defined by the following: a state space S , an action space A , a probabilistic transition mechanism P that controls the dynamics of states under different actions, and a cost structure g that penalizes the actions chosen in the visited states. Discovering the best course of action that reduces a certain cost target is the aim. To solve a Markov decision problem, one must identify the best policies to deploy.

Traditionally, dynamic programming (DP) has been used to solve MDPs. The idea behind it is to solve a subproblem and then use that answer to tackle a related subproblem again and again. The DP algorithms discover the best policy and value function via iterations. The optimum value function meets the consistency criteria, as is known from Bellman equation. In order to solve the Bellman problem repeatedly, DP approaches depend only on the system's environment model. The probability of a transition and the associated costs make up this data. When it comes to real-world issues, nobody knows what the system model is and the associated MDPs have huge state and action spaces. Obtaining the best sequence of actions for MDPs without comprehensive model knowledge is a need, and the additional challenge of dealing with huge state and action spaces is a real possibility.

Without knowing the system model, RL algorithms find approximations to an MDP. These algorithms take their cues from DP algorithms that iterate on policies and values, but instead of using both reward and state samples to learn the best policies, they utilize only the former. Conducting controlled explorations of activities repeatedly is crucial for this. But there are a lot of problems with using previous RL algorithms in autonomous systems.

Without knowing the system model, RL algorithms find approximations to an MDP. These algorithms take their cues from DP algorithms that iterate on policies and values, but instead

of using both reward and state samples to learn the best policies, they utilize only the former. Conducting controlled explorations of activities repeatedly is crucial for this. But there are a lot of problems with using previous RL algorithms in autonomous systems. Although many more exist in some shape or another, we address the following difficulties in this thesis:

Efficiency of the sample: In most cases, real-world systems do not have distinct settings for training and testing. Since the agent's exploratory behaviors impact the system, it cannot have a distinct exploration strategy during training. All training data is derived from the actual system. Alternatively, the agent has to do adequately with sparse data. On top of that, single-instance systems aren't always compatible with distributed training methods that instantiate hundreds or thousands of settings to gather additional data. Policy learning has to be data efficient since the data produced by these real-world systems is either expensive or unstable. Existing RL algorithms rely on large amounts of data and extensive coverage, which may not be present in off-line system logs. Therefore, an algorithm has to be performant and sample-efficient to train on a real system.

Exploration without risk: When not adequately regulated to predetermined parameters, almost all physical systems have the potential to harm or degrade both themselves and their surroundings. Controlling these systems requires, therefore, giving basic consideration to their safety. Both the system's operation and the exploratory learning stages should prioritize safety. These may pertain to system or environmental safety concerns (such as preventing dynamic impediments or limiting the frequency of changes to control variables, among other things). While a safe and/or manual backup plan might be in place to handle situations when the learnt policy goes against safety requirements, RL algorithms should be built to avoid explicitly relying on that backup plan.

In non-stationary operating settings, a real-life RL agent managing a system, such as a vehicular traffic signal junction, must constantly monitor the features of its surroundings and modify its learned behaviors to guarantee efficient system performance. Models and contexts define the system environment. P is the probability distribution of state development, and g is the cost function; the former helps the agent learn the correct action sequence, while the latter simulates the uncertainty in state evolution. In non-stationary situations, the context of the environment varies throughout time, which is a practical difficulty. If this is the case, then the RL agent's learned policies will be useless as soon as the environmental context changes

drastically. As a result, inefficient system operation occurs since the RL agent needs to resume learning rules anytime the environment model changes.

Designed for networked systems, autonomous agents powered by learning approaches are becoming more common. Automated software, most often tailored algorithms, controls or monitors processes in networked autonomous systems. With their ability to function on many spatial and temporal scales, these software and physical components are intricately linked. They may be in constant communication with each other over different time frames. There are several examples of networked autonomous systems, such as smart grids, autonomous vehicle systems, smart medical equipment, and IoT systems. These self-driving technologies improve production line productivity, respond quickly to medical emergencies, and more. For instance, in order to ensure the smooth running of industrial production processes, a network of linked sensors, instruments, and other devices is used in an IIoT system. Increased productivity and other monetary gains are possible outcomes of using this link.

When properly designed, RL algorithms can handle data-driven sequential decision-making issues in networked systems. One of the many problems that crop up when we try to build autonomous learning agents that can operate networked autonomous systems is the fact that these systems will have to balance conflicting goals as they run. Learning a strategy that maximizes many goals may be necessary for the agent in many systems. These goals often conflict with one another; that is, achieving one will likely result in the other being diminished. How does the agent figure out a fairly decent plan that meets all goals in such a situation? Will it be able to meet the goals while balancing them? The second challenge is that it's complicated; the agent has to learn with very little computer power and real-time input. For improved decision-making, it must also aggregate temporal data. Two systems are discussed in this thesis; one is an IIoT system and the other is a robotic system. In these particular contexts, we create solutions to deal with the difficulties of complicated problems and conflicting goals.

LITERATURE REVIEW

Nahodil, Pavel & Vítků, Jaroslav. (2012) This presentation offers a novel architecture for autonomous agents and is based on research into the evolution of artificial organisms. This study expands upon previous research on artificial life conducted over the last two decades at the Czech Technical University's Department of Cybernetics. This architectural plan integrates insights from many fields: AI, Ethology, ALife, and Intelligent Robotics. The usage of a more

sophisticated control system that incorporates elements of traditional AI, such as reinforcement learning, planning, and artificial neural networks, is evident here. The primary idea behind its operation comes from the study of ethics, which describes how an agent's existence is modeled after an animal in the wild, where the latter learns more complicated rules by progressively applying simpler ones. Hierarchical Reinforcement acquisition (RL) is the foundation of this design, which allows for the autonomous creation of an action hierarchy based purely on agents' interactions with their environments, and it facilitates online acquisition of all information from start. Developing a domain-independent hierarchical planner from scratch is the fundamental premise of this method. Our planner can work with RL-learned habits. This implies that a planning system may make use of an autonomously obtained hierarchy of activities, in addition to action selection methods based on reinforcement learning. Because of this, the agent may use his experiences alone to solve complex problems using high-level deliberative reasoning. The agent's existence was recreated in a virtual setting so that higher-level control, instead of a sensory system, could be handled.

Espinosa-Leal (2019) Improving economic efficiency has always been a goal of industry, and recently, that goal has centered on finding methods to use technology to cut down on human work. There is still considerable confusion about the goals of some modern systems, namely whether they are autonomous or automated, even with state-of-the-art technology such as packaging robots and artificial intelligence for defect detection. This work provides a literature overview, highlights the differences between automated and autonomous systems, and identifies the main obstacles to developing autonomous agents' learning processes. Training reinforcement learning agents to generalize their knowledge of particular tasks is something we cover in detail, employing various forms of extended realities like digital twins. We talk about how they may be utilized to create self-learning agents after generalization is accomplished. Next, we provide self-play scenarios as a means of educating self-learning agents in a nurturing setting that emphasizes the need of adaptability. By using two ϵ -greedy algorithms to resolve a multi-armed bandit issue, we provide a preliminary version of our concepts. Additionally, we highlight potential future uses in the field of industrial management and provide a modular design for enhancing decision-making via autonomous agents.

Jebessa (2022) The inner workings of autonomous cars, and reinforcement learning in particular, are examined in this article. Advanced autonomous driving systems have been developed by businesses like Waymo, Tesla, and GM via the use of machine learning algorithms. This article takes a look at the algorithms and reinforcement learning methods

utilized by these businesses and suggests some new approaches to fixing the issues that plague the majority of their cars. Also included in this study is a comprehensive review of the Q learning method that AVs employ.

Bharathi (2023) A branch of machine learning known as reinforcement learning focuses on creating algorithms that let agents learn by interacting with their environment in a trial-and-error fashion. It is a model of learning in which an agent learns by doing, and then acting in a way that maximizes its cumulative reward. Numerous domains have found fruitful uses for reinforcement learning, such as robotics, gaming, recommendation systems, and even banking. It has also shown potential in resolving complicated issues that are difficult to address via more conventional means. Regardless of these obstacles, reinforcement learning is an effective method for creating AI systems with the ability to learn and adapt to new situations, and it will certainly be a key component of future AI advancements.

Tammewar (2023) When it comes to artificial intelligence (AI), reinforcement learning (RL) is making all the difference in creating fully autonomous systems that understand the environment around them better than humans. Applying RL to large-scale issues is made easier by deep learning (DL). DL enables the acquisition of robot supervisory principles from visual data, the development of video game expertise from pixel-level information, and more. Successful applications of RL algorithms in computer vision, pattern recognition, natural language processing, and voice parsing have been shown by recent research. These methods aid in representing situations involving high-dimensional, unprocessed data input. Using RL, this study trains a computer model of a racing automobile to drive itself around a course. Deep Deterministic Policy Gradient (DDPG), Deep Q-network (DQN), and Proximal Policy Optimization (PPO) are three of the core methods investigated in this Deep RL research. Using metrics like throughput, precision, and overall performance, the study compares and contrasts these three well-known algorithms. Research shows that the DQN outperformed the other algorithms that were previously available after a comprehensive examination. The performance of the DQN with and without ϵ -decay was compared in this research, and it was shown that the DQN with ϵ -decay is more stable and better suited to our purpose. Autonomous cars that use the DQN with ϵ -decay might have their performance and stability enhanced by the results of this study. It wraps up with talking about possible research topics in autonomous driving and how to fine-tune the model for future real-world implementations.

OBJECTIVES

1. To research the Deep Reinforcement Vehicle steering regulations may be emulated using learning algorithms.
2. To research cooperative reinforcement amongst several agents Learning for Robots with Multiple Components.

RESEARCH METHODOLOGY

This work follows recognized techniques in RL research to investigate the efficacy and efficiency of RL algorithms for training autonomous software agents. Essential components of the study strategy include problem conceptualization, experimental design, data collection, and analytical techniques (Kiumarsi et al., 2018).

Problem Formulation:

The first step in the process is defining the issue and the study's objectives. It is necessary to specify the tasks that the autonomous agents are supposed to learn as well as the standards by which their performance will be evaluated. You must also determine if real-world scenarios or simulation settings more accurately represent the domains of your planned applications as part of the problem-formulation process.

Designing Controlled Experiments to Evaluate RL Algorithms

The second component of the experimental design is conducted under varying conditions. In this step, we choose suitable RL algorithms based on the job requirements and available resources. Research may include RL techniques such as deep Q-learning, policy gradient methods, meta-learning methodologies, and actor-critic architectures.

Gathering Data:

The process of accumulating data involves the creation of training data as well as the performance of trials in order to instruct autonomous agents on how to use RL algorithms. It may be essential to run several episodes of agent-environment interaction in order to collect state-action-reward trajectories in virtual environments. The process of acquiring data for practical applications may include placing reinforcement learning (RL) agents in settings that are supervised (Shah, 2020).

Methods for Analysis:

Algorithms for reinforcement learning are assessed and tested in the analysis phase by using criteria and metrics that have already been developed. As part of this process, it is necessary to evaluate the convergence rates, learning curves, and eventual performance of the agents across a variety of experiments. Statisticians may use methods such as hypothesis testing and significance testing to compare their findings in order to determine the degree to which different algorithms are successful in performing the defined tasks.

After that, there is Validation and Interpretation

The strategy employs validation techniques to ensure that the experimental results are both valid and dependable. Robustness testing, sensitivity analysis, and cross-validation techniques are a few of the methods that may be used in order to establish the degree to which the trained agents are able to generalize. In order to arrive at a conclusion on the practical implications, algorithmic limitations, and strengths of the RL algorithms, a comprehensive study of the data is necessary (Singh, Kumar, and Singh, 2021).

This work presents empirical research on reinforcement learning for autonomous software agents, and it utilizes a methodology that adheres to stringent scientific principles and norms. Zhang and Mo (2021) explain that the primary objective of the study is to carefully come up with research questions, create experiments, collect data, and evaluate the results in order to contribute to the body of knowledge that already exists on autonomous systems and artificial intelligence.

RESULTS

The outcomes of the research contribute to our comprehension of the effectiveness and efficiency of reinforcement learning (RL) algorithms in training software agents to function independently in a variety of contexts. The purpose of the trials was to evaluate the effectiveness of a number of different RL algorithms in resolving difficult issues and achieving their objectives. In order to determine how well deep Q-learning (DQN) systems were able to handle robotic control tasks, we first carried out a series of experiments that simulated these activities. The results of the study revealed that DQN agents were capable of acquiring the abilities that were required in order to efficiently learn how to control robotic arms in order to complete certain tasks, such as collecting things or reaching target locations. The findings of the learning curves and convergence rates analyses demonstrated that as the training

continued, the DQN agents' performance improved significantly. However, the pace of improvement slowed down significantly in the latter stages of training. Furthermore, when compared to baseline algorithms, such as the random policy algorithm and standard control procedures, DQN has shown its superiority in terms of efficiency and the speed with which it is able to execute tasks.

Furthermore, we investigated the impact that further enhancements made to algorithms, such as giving greater importance to experience replay and dueling network topologies, had on the effectiveness of DQN agents. Because of these changes, which resulted in an increase in learning stability and sampling efficiency, the consequences were a more rapid convergence and increased final performance scores. Statistical analysis techniques, such as analysis of variance (ANOVA) and t-tests, were used in order to compare and assess the significance of the performance of different algorithm versions (Anon, 2022).

In addition, they put to the test the capabilities of reinforcement learning algorithms in the two application domains of gaming and autonomous navigation, both of which fall outside of the scope of robotic control. In autonomous navigation tasks, robotic learning agents were taught to avoid dangers, navigate through ever-changing environments, and reach specified destinations. By examining the trajectories of agents as well as the rates at which they collided with other agents, it was shown that reinforcement learning (RL) algorithms were able to successfully complete navigation tasks with a relatively low number of collisions and deviations from ideal paths. In the same way, reinforcement learning agents displayed their abilities in environments where they were engaged in playing games. They were able to defeat human players and exceed existing standards while simultaneously displaying their expertise in playing challenging games. Analysis of win rates, game scores, and decision-making processes might help to provide a more comprehensive understanding of the learning dynamics and strategies that are employed by reinforcement learning agents in order to achieve competitive performance. The results of the study provide evidence that RL algorithms are able to train autonomous software agents to perform a wide range of activities in a number of domains. The findings not only contribute to the existing body of knowledge on the advantages and disadvantages of reinforcement learning methods, but they also make way for the exploration of new and promising avenues of research and development in the fields of artificial intelligence and autonomous systems.

The results further revealed the relevance of reward shaping and curriculum learning approaches to further increase the learning efficiency and performance of RL agents. In comparison to conventional reinforcement learning (RL) methods, the trials that included curriculum design and incentive engineering exhibited much greater learning rates and ultimate performance scores. By establishing reward functions and sequences of curricula in such a manner as to enable faster convergence and more effective policies, researchers were able to assist agents in achieving their objectives in a shorter amount of time (Fadi AlMahamid, 2022). Furthermore, the experiments were designed to determine whether RL algorithms are capable of managing challenging tasks and circumstances on a broad scale. The process of training was expedited, and enormous quantities of data were efficiently managed by using distributed reinforcement learning frameworks in conjunction with parallelization strategies. The shown reductions in the amount of time necessary for training that were made achievable by distributed RL configurations allowed agents to gain knowledge from more extensive datasets and perform more effectively on challenging tasks. To summarize, the comprehensive experimental setting that was implemented in this research provides insight into the advantages, disadvantages, and practical consequences of using reinforcement learning methods to train artificial intelligence agents. The results of the study have important consequences for practical applications in a variety of domains, including artificial intelligence, robotics, and autonomous systems. Additionally, the findings contribute to the advancement of the present state of the art in reinforcement learning research (Kiumarsi et al., 2018). The results that were obtained from the experiments that were conducted as part of this study are examined and examined in further detail, and the ramifications of these findings as well as their more general significance are addressed in the discussion section. It utilizes prior theoretical frameworks and research in order to improve the understanding of the outcomes and the ways in which they are relevant to the field of reinforcement learning (RL) as it applies to self-operating software agents.

ANALYSIS OF FINDINGS

The results demonstrated that RL algorithms were successful in training autonomous agents to carry out complex tasks in a number of different domains, including but not limited to robotic control, autonomous navigation, and game playing. It can be shown from the findings that RL techniques are capable of increasing the rates at which tasks are completed, increasing the efficiency of learning, and increasing the extent to which skills may be generalized. This indicates that they are capable of addressing issues that occur in the real world and achieving

their objectives. On the other hand, in order to properly comprehend the importance of these findings, it is essential to investigate them within the context of past research and theoretical frameworks (Shah, 2020). The conclusions drawn from this research have a number of ramifications, not only for the theory and practice of reinforcement learning but also for the real-world applications of reinforcement learning. From a theoretical perspective, the results contribute to an improved understanding of the dynamics of learning, the principles of algorithm design, and the optimization approaches used in reinforcement learning (RL). They provide validity to the idea that RL algorithms are both efficient and scalable, and they also validate the present theoretical frameworks. The observed improvements in learning efficiency and generalization abilities underline the importance of tactics such as reward shaping, curriculum learning, and transfer learning in the context of increasing the effectiveness of reinforcement learning (RL) approaches (Singh, Kumar, and Singh, 2021). The results demonstrate that there are significant consequences for the design and implementation of autonomous systems in the real world. There is a wide range of industries that would benefit from reinforcement learning algorithms, including manufacturing, healthcare, transportation, and entertainment, because of their track record of success in executing complex tasks such as autonomous navigation and robotic manipulation. Furthermore, it has been shown that agents based on reinforcement learning has the potential to withstand environmental changes and disruptions. This is a promising development that provides evidence of their reliability and adaptability in situations that are unexpected and always changing. In order to overcome a variety of difficulties, including safety issues, wasteful sampling, and algorithmic instability, more research and development are required to increase the widespread practical use of RL-based approaches (Zhang and Mo, 2021).

CONCLUSION

The capabilities and prospects of reinforcement learning (RL) with regard to training autonomous software agents across many domains have been thoroughly examined in this study. The outcomes of our research indicate that reinforcement learning algorithms are effective for training agents in order to adapt to changing circumstances and master difficult tasks, whether in simulated or real-world environments. The results of the research emphasize the wide range of issues that reinforcement learning techniques are able to address, including gaming, autonomous navigation, and robotic control, among other applications. We have discovered that RL agents are capable of surpassing both the criteria set by humans and the methods of control that are used traditionally. These agents are able to attain performance

levels that are on par with those of other agents. It has been shown that the use of approaches that are based on reinforcement learning (RL) results in an improvement in learning efficiency, generalization capacities, and resilience. This finding indicates that such techniques possess the potential to be implemented in practical situations. Nevertheless, the analysis reveals a number of possible avenues for additional exploration as well as a number of difficulties that need to be overcome. We will need to address the issues that have been identified with sample inefficiency, algorithmic instability, safety assurance, and ethical concerns if we wish to encourage a greater number of individuals to begin incorporating RL techniques into their professional activities.

References

1. Anon (2022). The Role of Reinforcement Learning in Autonomous Systems |. [online] www.interviewkickstart.com. Available at: <https://www.interviewkickstart.com/blog/reinforcement-learning-autonomous-systems> [Accessed 3 Mar. 2024].
2. Fadi AlMahamid (2022). Reinforcement Learning Algorithms: An Overview and Classification | IEEE Conference Publication | IEEE Xplore. [online] [ieeexplore.ieee.org](https://ieeexplore.ieee.org/abstract/document/9569056). Available at: <https://ieeexplore.ieee.org/abstract/document/9569056> [Accessed 3 Mar. 2024].
3. Kiumarsi, B., Vamvoudakis, K.G., Modares, H. and Lewis, F.L. (2018). Optimal and Autonomous Control Using Reinforcement Learning: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, [online] 29(6), pp.2042–2062. doi:<https://doi.org/10.1109/TNNLS.2017.2773458>.
4. Padakandla, S. (2021). A Survey of Reinforcement Learning Algorithms for Dynamically Varying Environments. *ACM Computing Surveys*, 54(6), pp.1–25. doi:<https://doi.org/10.1145/3459991>.
5. Shah, V. (2020). Reinforcement Learning for Autonomous Software Agents: Recent Advances and Applications. *Revista Espanola de Documentacion Cientifica*, [online] 14(1), pp.56–71. Available at: <https://redc.revistascsic.com/index.php/Jorunal/article/view/155> [Accessed 3 Mar. 2024]

6. Singh, B., Kumar, R. and Singh, V.P. (2021). Reinforcement learning in robotic applications: a comprehensive survey. *Artificial Intelligence Review*, 07(08). doi:<https://doi.org/10.1007/s10462-021-09997-9>.
7. Kiran, Bangalore &Sobh, Ibrahim&Talpaert, Victor &Mannion, Patrick&Sallab, Ahmad&Yogamani, Senthil &Perez, Patrick. (2021). Deep reinforcement learning for autonomous driving: a survey. *IEEE transactions on intelligent transportation systems*. Pp. 1-18. [10.1109/tits.2021.3054625](https://doi.org/10.1109/tits.2021.3054625).
8. Marina, Liviu&Sandu, A. (2017). Deep reinforcement learning for autonomous vehicles-state of the art. *Bulletin of the transilvania university of brasov*. Vol. 10 (59).
9. Neftci, Emre&Averbeck, Bruno. (2019). Reinforcement learning in artificial and biological systems. *Nature machine intelligence*. 1. [10.1038/s42256-019-0025-4](https://doi.org/10.1038/s42256-019-0025-4).
10. Rodriguez-Soto, Manel &Lopez-Sanchez, Maite&Rodríguez-Aguilar, Juan. (2021). Multi-objective reinforcement learning for designing ethical environments. 545-551. [10.24963/ijcai.2021/76](https://doi.org/10.24963/ijcai.2021/76).
11. Bhalla, Sushrut&Subramanian, Sriram&Crowley, Mark. (2020). Deep multi agent reinforcement learning for autonomous driving. [10.1007/978-3-030-47358-7_7](https://doi.org/10.1007/978-3-030-47358-7_7).
12. Sivashangaran, Shathushan. (2021). Application of deep reinforcement learning for intelligent autonomous navigation of car-like mobile robot. [10.13140/rg.2.2.19676.31364](https://doi.org/10.13140/rg.2.2.19676.31364).
13. Iroegbu, Emmanuel&Madhavi, Devaraj. (2021). Accelerating the training of deep reinforcement learning in autonomous driving. *IAES international journal of artificial intelligence (ij-ai)*. 10. 649. [10.11591/ijai.v10.i3.pp649-656](https://doi.org/10.11591/ijai.v10.i3.pp649-656).