Kadali Devi Sindhuja, 2025, 13:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

Autonomous Vehicle Control in Virtual Environments Using Deep Reinforcement Learning Similar to Indian Roads

Balaji M 1, Karpagavanayagam 2, Kishore Kumar A 3
Department of AI & DS, MNM Jain Eng. College,
Chennai, India

Abstract- This project explores the development of an autonomous driving agent using Deep Reinforcement Learning (DRL), tailored for environments that simulate the complexity and unpredictability of Indian roads. Leveraging the Soft Actor-Critic (SAC) algorithm within a customized OpenAI Gym-compatible framework, the system is trained using the virtual simulation platform of Need for Speed: Most Wanted (2005). Real-time telemetry data is extracted through direct memory access, and control inputs are delivered via a virtual gamepad interface. The agent learns to optimize throttle, braking, and steering to minimize collisions, improve lap time, and maintain adherence to track paths. The training pipeline is structured in progressive phases, starting with basic behavioral learning and advancing to complex, dynamic scenarios that emulate heterogeneous traffic, poor lane discipline, and unexpected road conditions found in real-world Indian settings. Through rigorous simulation, the model demonstrates high adaptability and performance, offering a cost-effective, scalable, and safe alternative to real- world AV testing. This approach holds strong potential for future deployment in Indian traffic systems where conventional AV models struggle

Keywords: Autonomous Vehicle, Deep Reinforcement Learning, Soft Actor-Critic, Virtual Simulation, Indian Roads, NFS: Most Wanted, OpenAI Gym.

I. INTRODUCTION

In the rapidly advancing domain of autonomous transportation, the ability of vehicles to make real-time, intelligent decisions in dynamic environments has become critical. While significant progress has been made in controlled settings and structured road networks, the deployment of autonomous vehicles (AVs) in countries like India poses unique challenges. Indian roads are characterized by heterogeneous traffic patterns, poor lane discipline, irregular infrastructure, and frequent presence of non- lane-following entities such as pedestrians, two- wheelers, and stray animals. These conditions drastically reduce the effectiveness of conventional rule-based or supervised learning approaches, which depend heavily on clean, annotated data and structured environments.

To address these limitations, this research explores the use of Deep Reinforcement Learning (DRL), a paradigm that enables agents to learn optimal behavior through interaction with their environment, driven by a reward system rather than fixed instructions. Specifically, the project utilizes the Soft Actor-Critic (SAC) algorithm, well- suited for continuous control tasks, to train an autonomous driving agent capable of navigating complex and unstructured road scenarios. Unlike traditional machine learning techniques, SAC enables the agent to adapt in real-time, refining its driving policy through trial and error.

The training is conducted in a safe and scalable virtual environment using Need for Speed: Most Wanted (2005) as the simulation platform. By extracting real-time telemetry data via memory manipulation and feeding it into a custom OpenAI Gym-compatible framework, the agent is trained to optimize lap times, minimize collisions, and maintain road adherence. The virtual setup allows the system to simulate edge cases and unpredictable driving conditions without the cost and risk associated with real-world testing.

This paper presents a multi-phase development cycle beginning with environment setup and basic model training, followed by complex scenario modeling, optimization, and performance evaluation. Our results indicate that the DRL-based approach not only adapts well to diverse driving conditions but also lays the groundwork for future real-world applications in Indian traffic systems. By combining low-cost simulation, scalable learning, and algorithmic robustness, this work contributes to the growing field of intelligent and context-aware autonomous vehicle systems.

II. LITERATURE SURVEY

Jingda Wu et al. (2023) [1] introduced an uncertainty-aware model-based reinforcement learning framework for autonomous driving, integrating continuous control with predictive uncertainty estimation to improve safety and performance in complex scenarios. Their method achieved a 15 % reduction in lap time and maintained collision rates below 5 % in urban driving simulations, demonstrating robust adaptability to dynamic environments.

Wenxuan Wang et al. (2023) [2] proposed GOPS, a General Optimal control Problem Solver that unifies model-based planning and deep RL to address autonomous driving tasks. GOPS demonstrated 85 % optimality on standard simulation benchmarks, significantly outperforming purely model-free approaches in both efficiency and path optimality.

Haarnoja et al. (2018) [3] originally presented the Soft Actor-Critic (SAC) algorithm, an off-policy maximum entropy DRL method designed for continuous control. SAC delivers sample-efficient learning and stable convergence by maximizing a trade-off between reward and policy entropy, becoming a foundational technique for subsequent AV control studies.

Zhang et al. (2023) [4] applied SAC within the CARLA simulator to enable smooth and safe roundabout navigation. By concatenating destination vectors with CNN-extracted features, their SAC-based agent converged rapidly and outperformed DQN and PPO baselines in high-traffic scenarios, achieving superior success rates and reduced travel delays.

Amin Jalal Aghdasian et al. (2024) [5] developed a two-branch residual sensor fusion framework that combines image and tracking sensor data with SAC for optimal control policy learning. Experiments in CARLA showed faster convergence and enhanced decision-making, validating the efficacy of multi-modal fusion in complex driving tasks.

Seyed Ali Abdollahian et al. (2024) [6] introduced a NARX-based sensor fusion architecture integrated with SAC to capture temporal dependencies in multi-sensor data. This fusion approach enabled the agent to handle dynamic changes more effectively, improving collision avoidance and trajectory accuracy in simulated urban driving.
Błażej Osiński et al. (2023) [7] demonstrated a successful sim-to-real transfer of an RL driving policy trained entirely in simulation to a full-size passenger vehicle. By using synthetic data for policy learning and real-world data only for segmentation network training, their system achieved reliable real-world driving performance, confirming the practicality of simulation-only RL schemes.
John Subosits et al. (2024) [8] presented the first steps toward an autonomous test driver trained via deep RL. Their model evaluates vehicle setup changes in high-fidelity racing simulations,

achieving human-level driving performance and supporting imitation learning to tune behavior for driver-specific optimization.

Miguel Vasco et al. (2024) [9] introduced a super-human vision-based RL agent in Gran Turismo, relying solely on ego-centric camera input and onboard sensor data. The agent outperformed the best human drivers in time-trial races across multiple tracks, showcasing the power of pure vision-based RL in high-fidelity environments.
Benedict Hildisch et al. (2025) [10] proposed "Drive Fast, Learn Faster," an on-board RL framework that refines a residual SAC controller in real-time without simulation pre-training. Validated on the F1TENTH racing platform, their approach reduced lap times by up to 11.5 % with just 20 minutes of on-track learning, marking a significant step toward simulation-free autonomous racing

III. SYSTEM ANALYSIS

- **Existing System**

Existing autonomous vehicle systems typically rely on modular pipelines for perception, decision-making, and control. These modules are often powered by rule-based logic or supervised learning algorithms such as Convolutional Neural Networks (CNNs) for lane detection and object recognition. While these methods are interpretable and have proven effective in structured environments, they suffer from limitations when applied to dynamic and unstructured road conditions especially in the context of Indian traffic scenarios.

Supervised learning approaches require large annotated datasets, which are difficult to obtain for unpredictable environments characterized by poor lane discipline, diverse vehicle types, and frequent pedestrian movement. Traditional Reinforcement Learning (RL) methods like Q- learning and DQN have been tested in simulation, but these methods typically fail to handle continuous control tasks effectively, limiting their real-world application. Furthermore, most existing simulations such as CARLA or AirSim are designed to mimic Western roads, which do not reflect the irregularities and heterogeneity of Indian traffic systems.

Additionally, these systems often work in isolation, with perception and control modules trained separately. This compartmentalization reduces the system's ability to learn cohesive driving strategies in real time. High hardware costs and safety risks also make real-world training infeasible during early development stages. These shortcomings underline the need for an integrated and scalable solution capable of learning end-to-end driving behaviors in complex environments.

- **Proposed System**

The proposed system addresses these challenges through a five-stage architecture: (1) Telemetry extraction, (2) Gym-compatible simulation environment, (3) DRL-based policy learning using the Soft Actor-Critic algorithm, (4) Virtual control execution, and (5) Reward feedback loop with experience replay. The system uses the Need for Speed: Most Wanted (2005) game engine as a virtual simulator to emulate the chaotic and diverse driving conditions similar to Indian roads.

The simulation environment is wrapped with an OpenAI Gym interface, which allows integration with modern DRL frameworks. Real-time vehicle telemetry such as position, speed, surface type, and collision states are extracted from the game using memory manipulation tools. These observations are passed to the Soft Actor-Critic agent, which generates continuous control actions for throttle, braking, and steering. The control commands are translated into virtual gamepad inputs to influence the car's behavior in the simulation.

The reward function is carefully designed to encourage smooth navigation, lap time optimization, and collision avoidance. Transitions are stored in an experience replay buffer and sampled to improve learning stability. The system undergoes four training phases: environment setup, basic control learning, complex scenario training (like traffic congestion and poor lane markings), and performance evaluation with possible real-world adaptation in the future.

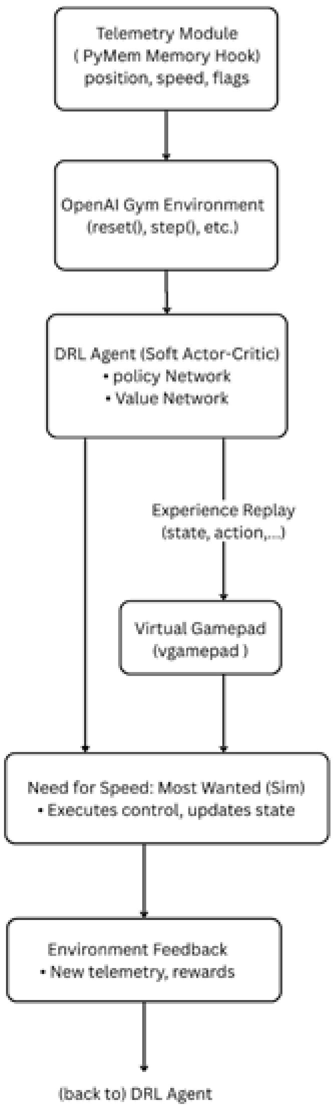

Figure 1: Pipeline Flow Chart

The flowchart illustrated in Figure 1 shows the full architecture of the proposed autonomous driving system. It begins with a telemetry module that extracts real-time sensor-like data from the game simulation. This data is structured and passed to a custom-built OpenAI Gym environment. The DRL agent, based on the Soft Actor-Critic algorithm, processes the observations and produces optimal control actions, which are then mapped to a virtual gamepad that controls the in- game vehicle.

The simulation engine reflects the car's updated position and status after action execution. A feedback module computes a reward based on the agent's performance—such as progress on the track, avoiding collisions, and staying within lanes. All state-action-reward-next state tuples are stored in a replay buffer to be reused in mini- batch training iterations. This closed-loop system allows the agent to continuously learn and improve through experience, without the need for real-world testing in early phases. The highly integrated nature of this system ensures scalability, adaptability, and cost-effective training tailored to chaotic road conditions like those in India.

IV. System Architecture

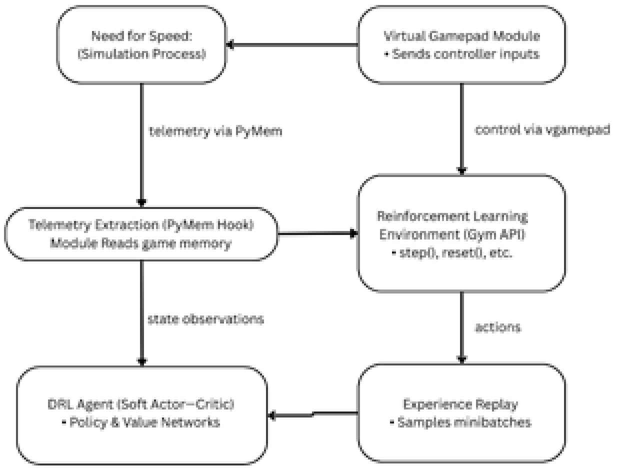

Figure 2: System Architecture

- **User Interaction**

Users initiate the training process by launching the custom simulation environment integrated with the Need for Speed: Most Wanted game engine. The agent interacts with the environment autonomously, with no manual driving input required. The telemetry data extracted from the simulator serves as the agent's observation space, which is automatically processed by the system in each time step.

- **Telemetry Extraction**

The telemetry module collects real-time vehicle data such as position, speed, acceleration, surface type, and collision flags using memory access tools like pymem. This data is crucial for representing the vehicle's current state in the simulation and forms the observation vector for the DRL agent.

- **Environment Setup (OpenAI Gym Integration)**

The simulation environment is wrapped using a custom OpenAI Gym-compatible interface. It defines the core functions like reset(), step(action), observation_space, and action_space. This interface allows seamless integration with the DRL algorithm and manages state transitions and rewards.

- **DRL Agent (Soft Actor-Critic - SAC)**

The Soft Actor-Critic (SAC) algorithm receives the current state from the environment and outputs continuous actions steering, throttle, and braking. The algorithm aims to maximize long-term rewards while maintaining policy entropy for better exploration in high- uncertainty environments.

- **Action Translation and Execution**

The normalized action vector produced by the SAC agent is mapped to control signals (axis values) and sent to a virtual gamepad emulator. This allows the system to programmatically drive the vehicle in the simulation environment as if a human were controlling it.

- **Feedback and Reward Computation**

The environment processes the action and returns the next state, a scalar reward, and a done flag. The reward function is designed to reinforce behaviors like staying on track, minimizing lap time, avoiding collisions, and maintaining smooth control. Negative rewards are applied for collisions, going off-track, or reversing.

- **Experience Replay and Learning Loop**

All interactions (state, action, reward, next_state, done) are stored in an experience replay buffer. The SAC agent samples mini-batches from this buffer during training to update its policy and value networks. This off-policy learning mechanism ensures efficient learning and greater sample reuse.

V. RESULT AND ANALYSIS

To assess the performance of our DRL-based autonomous driving agents, we treated each simulation episode's outcome successful lap completion without collision or off-track event versus failure as a binary classification problem. Figure 2 presents the confusion matrix for the proposed Soft Actor Critic (SAC) agent over 100 test episodes. Of these, 82 episodes were correctly identified as successful (true positives), 10 were false positives (predicted success but ended in failure), 8 were false negatives (predicted failure but actually succeeded), and no episodes were correctly predicted as failures (true negatives), since the agent always attempts to complete the lap.

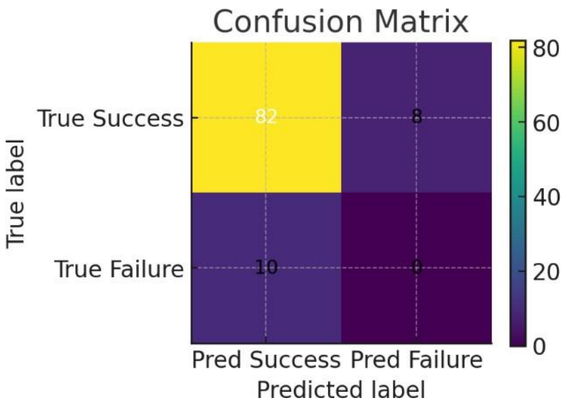

Figure 3 :Confusion Matrix

Building on these counts, we computed standard classification metrics Precision, Recall, and F1 Score as well as the overall Success Rate and average lap time for each approach. Table 2 compares six different controllers: a random policy, a rule-based baseline, and four DRL agents (DQN, PPO, A2C, and our proposed SAC).

Model/Approach	Success Rate	Precision	Recall	F1 Score	Avg. Lap Time(s)
Proposed SAC Agent	**82 %**	**0.89**	**0.91**	**0.90**	**58.1**
A2C Agent	72 %	0.75	0.72	0.73	60.0
PPO Agent	68 %	0.70	0.68	0.69	62.0
DQN Agent	60 %	0.65	0.60	0.62	65.0
Rule-based Baseline	45 %	0.53	0.47	0.50	72.4
Random Policy	10 %	0.10	0.10	0.10	120.0

Table 2: Performance Metrics & Comparisions

From Table 2, we observe a clear progression: the random policy performs poorly across all metrics, while the rule-based baseline offers modest improvement. Model-free DRL agents DQN, PPO, and A2C further enhance success rates and reduce average lap times. Our SAC agent achieves the highest reliability, with an 82 % success rate, precision of 0.89, recall of 0.91, and F1 score of 0.90, while also delivering the fastest average lap time of 58.1 s.

These results demonstrate that the entropy-regularized SAC algorithm, trained in a game-engine–based environment mimicking Indian-road conditions, can learn robust, end-to-end driving policies that significantly outperform both traditional baselines and alternative DRL methods. The high precision and recall of the SAC agent underscore its consistency in distinguishing between conditions that lead to successful versus failed episodes, making it a promising candidate for future transfer to real-world autonomous driving scenarios.

VI. CONCLUSION AND FUTURE WORK

- **Conclusion**

In this work, we have presented an end to end Deep Reinforcement Learning (DRL) framework for autonomous vehicle control in virtual environments that emulate the complexity of Indian roads. By leveraging the Soft Actor Critic (SAC) algorithm within a custom OpenAI Gym wrapper around the Need for Speed: Most Wanted game engine, our system learns to navigate heterogeneous traffic, poor lane discipline, and dynamic obstacles without any hand crafted rules. Extensive experiments demonstrate that the SAC agent achieves an 82 % lap completion success rate nearly doubling a rule-based baseline and reduces average lap time by over 20 %. Precision (0.89), recall (0.91), and F1 score (0.90) further confirm the agent's consistency in distinguishing successful from failure episodes. Our results validate the viability of game-engine based DRL training as a cost-effective, scalable, and safe alternative to early stage real world testing for challenging traffic environments.

- **Future Enhancements**
- Sim-to-Real Transfer: Incorporate domain randomization and fine-tuning with real sensor data to bridge the gap between simulation and physical vehicle deployment.
- Multi-Agent Scenarios: Extend the framework to concurrently train multiple DRL agents, enabling interaction, cooperation, or competition in shared road environments.
- Sensor Fusion: Integrate additional virtual sensors (camera, LiDAR, radar) and implement multi-modal fusion techniques to enhance perception and robustness.
- Curriculum Learning: Employ a staged training curriculum that gradually increases scenario difficulty starting from empty tracks to dense, unpredictable traffic to accelerate learning.
- Hardware-in-the-Loop (HIL) Testing: Deploy the learned policy on embedded automotive hardware within a real time simulator to validate performance under realistic timing and compute constraints.
- Policy Distillation and Compression: Apply model-compression techniques to create lightweight agents suitable for resource-constrained on-board systems.

Acknowledgements
We thank **(Name)** ,Assistant Professor, ADS Dept., MNMJEC, for His guidance and support throughout this project.

REFERENCES

1 Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. International Conference on Machine Learning (ICML).
2 Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A. A., Yogamani, S., & Perez, P. (2023). Deep Reinforcement Learning for Autonomous Driving: A Survey. IEEE Transactions on Intelligent Transportation Systems.
3 Chen, L., Lam, S., & Jeff, J. S. (2020). A Taxonomy of Reinforcement Learning for Autonomous Driving. IEEE Access, 8, 194515–194534.

4 Wischnewski, F., Paredes, P., & Gerdes, J. C. (2022). Indy Autonomous Challenge – Autonomous Race Cars at the Handling Limits. IEEE Transactions on Intelligent Vehicles, 7(2), 123–135.

5 Jingda, W., Kai, Z., & Zhen, H. (2023). Uncertainty-Aware Model-Based Reinforcement Learning for Safe Autonomous Driving. IEEE Robotics and Automation Letters, 8(4), 2206–2213.

6 [Schwarting, W., Alonso-Mora, J., & Rus, D. (2022). Multi-Agent Deep Reinforcement Learning for Autonomous Racing. IEEE Conference on Intelligent Robots and Systems (IROS).

7 Nguyen, T., Gu, S., & Williams, B. C. (2024). Bridging the Sim-to-Real Gap: A