Transactions on Computational and Scientific Methods | Vo. 5, No. 7, 2025
ISSN: 2998-8780
https://pspress.org/index.php/tcsm
Pinnacle Science Press

Vision-Based Autonomous Navigation and Obstacle Avoidance in Mobile Robots Using Deep Reinforcement Learning

Thayer Corbin
University of Central Missouri, Warrensburg, United States
tcorbin39@gmail.com

Abstract: This paper presents a hybrid autonomous navigation system for mobile robots that integrates vision-based deep reinforcement learning (DRL) with a visual simultaneous localization and mapping (SLAM) module. The proposed framework employs a convolutional neural network and a Proximal Policy Optimization (PPO) algorithm to learn control policies from raw RGB images, enabling end-to-end navigation and obstacle avoidance. To enhance localization accuracy and robustness, we incorporate ORB-SLAM2 as a geometric localization backbone, providing real-time pose feedback to the policy network. Extensive experiments in both simulated and real-world environments demonstrate that the combined DRL+SLAM architecture outperforms classical and vision-only navigation baselines in terms of success rate, path efficiency, and collision avoidance. The results highlight the benefit of fusing learned perception and geometry-based reasoning to achieve robust and generalizable robot autonomy in complex indoor environments.

Keywords: Autonomous Navigation, Deep Reinforcement Learning, Simultaneous Localization and Mapping, Mobile Robotics, Visual Perception, PPO, ORB-SLAM2, Robot Control.

1. Introduction

Autonomous navigation has become a cornerstone of mobile robotics, enabling intelligent agents to perceive their surroundings, localize themselves within a map, plan safe and efficient paths, and adapt to dynamic environments. In recent years, advances in computer vision and artificial intelligence have fundamentally reshaped the way mobile robots navigate, especially with the introduction of deep reinforcement learning (DRL) techniques that allow for end-to-end training of control policies directly from sensory inputs. Unlike traditional navigation systems that rely on LiDAR sensors, hand-crafted rules, and static maps, DRL enables robots to learn from interaction, offering better generalization to unseen environments and robustness in unstructured scenarios. Despite its potential, vision-based DRL navigation still faces several critical challenges: high-dimensional image input imposes a significant computational burden; learning stable policies from sparse and delayed rewards remains difficult; and generalizing policies trained in simulation to real-world conditions introduces a domain gap that can hinder deployment. Moreover, visual input alone may be insufficient for precise localization, which is essential for safe and consistent obstacle avoidance. To address these issues, this paper proposes a hybrid system that combines visual DRL with simultaneous localization and mapping (SLAM) to create a robust, flexible navigation pipeline for indoor mobile robots. We use RGB images as the sole input modality to train an actor-critic architecture for continuous control and obstacle avoidance, while simultaneously incorporating ORB-SLAM2 to enhance localization accuracy and map awareness. The integrated system allows the robot to operate effectively in both known and unknown environments, learning goal-directed behavior through reinforcement while maintaining spatial consistency through visual odometry and loop closure. We validate our approach through extensive simulations in Gazebo and real-world experiments using a TurtleBot platform, comparing its performance against several baseline models including classical navigation stacks and vision-only DRL models. Results show that our method outperforms others in terms of success rate, collision avoidance, and trajectory efficiency. The contributions of this work are threefold: (1) we develop a vision-based DRL framework for autonomous navigation using RGB images only, removing dependency on expensive range sensors; (2) we integrate ORB-SLAM2 into the learning pipeline to provide real-time localization feedback, improving stability and adaptability; and (3) we conduct thorough experiments in simulated and real-world environments to demonstrate the generalization capability and practicality of our system. The remainder of this paper is organized as follows: Section II reviews related work in visual navigation and deep reinforcement learning. Section III introduces the overall system architecture. Section IV describes the DRL framework and training process. Section V details the SLAM integration. Section VI presents experimental setup and results. Section VII discusses the findings and limitations, and Section VIII concludes with potential directions for future work.

2. Related work

Autonomous navigation has long been a central topic in mobile robotics, and numerous approaches have been developed over the decades to enable robots to move safely and efficiently through complex environments. Classical navigation systems typically follow a modular architecture, consisting of perception, localization, mapping, path planning, and control components. These systems often rely on LiDAR or ultrasonic sensors for obstacle detection, with algorithms such as Dijkstra or A* for path planning, and PID or dynamic window approaches for motion control. While these pipelines have proven reliable in structured indoor spaces, their performance often degrades in dynamic, cluttered, or unstructured environments due to their limited adaptability and dependency on high-quality sensor data. The emergence of deep learning has shifted attention toward data-driven methods, especially convolutional neural networks (CNNs) for visual perception and semantic scene understanding. In particular, the integration of vision and learning-based control has opened the door to more flexible and generalizable navigation strategies. Deep reinforcement learning (DRL), which combines the representational power of deep neural networks with the sequential decision-making framework of reinforcement learning, has become a popular paradigm for robotic control. Early work by Mnih et al. [1] demonstrated the potential of deep Q-networks in high-dimensional visual tasks, which later inspired applications in continuous robot control, such as the work by Lillicrap et al. [2] using the Deep Deterministic Policy Gradient (DDPG) algorithm. In the context of navigation, Zhu et al. [3] proposed a deep reinforcement learning agent capable of navigating 3D environments using RGB images, achieving promising results in simulated environments. More recently, Mirowski et al. [4] combined DRL with auxiliary tasks like depth prediction and loop closure detection to improve performance in partially observable settings, while Chaplot et al. [5] introduced hierarchical reinforcement learning for efficient exploration and goal-reaching in large indoor spaces.

Despite these advancements, DRL-based visual navigation still faces practical deployment challenges. One common limitation is the lack of robustness and generalization when models trained in simulation are transferred to real-world environments—a phenomenon known as the “reality gap.” Domain randomization [6] and domain adaptation [7] have been used to address this issue, but require careful tuning and additional data. Another key limitation lies in the dependency on implicit localization within end-to-end DRL pipelines, where the robot must infer its position solely from raw visual input. This can lead to erratic behavior, especially in repetitive or ambiguous scenes. To address this, researchers have explored combining DRL with simultaneous localization and mapping (SLAM) systems. For instance, Sadeghi and Levine [8] trained policies in simulation and used visual SLAM during deployment to aid localization. Other works, such as Kan et al. [9], introduced hybrid architectures where SLAM provides metric maps or pose estimates that feed into the policy network. These approaches demonstrate that integrating classic localization with learning-based planning can improve stability, especially when using monocular vision.

This work builds upon these insights by tightly coupling a DRL-based control policy with ORB-SLAM2, leveraging the strengths of both learning and geometry-based methods. In contrast to prior work that either relies on handcrafted map-based navigation or purely end-to-end DRL, we design a hybrid system where the SLAM module provides spatial context, enabling the DRL policy to learn more efficiently and act more reliably. Additionally, our experiments extend prior benchmarks by evaluating both simulated and real-world scenarios, offering a more comprehensive validation of real-world applicability.

3. System Architecture

The proposed system architecture integrates vision-based deep reinforcement learning with a visual SLAM module to enable autonomous navigation and obstacle avoidance in indoor mobile robots. As shown in Figure 1, the architecture is composed of three primary modules: the visual perception and feature encoding module, the deep reinforcement learning policy network, and the localization and mapping module based on ORB-SLAM2. These modules operate in parallel and share partial data streams to ensure coherent decision-making and real-time responsiveness.

The visual perception module utilizes an RGB camera mounted on the mobile robot to continuously capture image frames of the environment. These images serve two purposes: first, they are input into a convolutional encoder network that extracts compact, high-level visual features for the DRL policy; second, they are forwarded to the ORB-SLAM2 system to support localization and environment mapping. The convolutional encoder is a lightweight ResNet-18 backbone pretrained on ImageNet, followed by a spatial attention module that enhances salient navigation cues such as doorways, hallways, and obstacles. The encoded visual features are then passed to the policy network, which is implemented using an actor-critic architecture (specifically the Proximal Policy Optimization or PPO algorithm). The actor network predicts continuous control commands (linear and angular velocities), while the critic network estimates the value function to guide policy updates during training. Unlike end-to-end models that rely solely on visual cues, our policy network additionally receives pose information from ORB-SLAM2, enabling it to make more informed decisions in spatially ambiguous scenarios.

The ORB-SLAM2 module operates concurrently and performs three critical functions: visual odometry through feature tracking and motion estimation, keyframe-based mapping with loop closure detection, and global pose graph optimization. We modify ORB-SLAM2 to output real-time pose estimations (in SE(2)) to the policy network at a frequency synchronized with the control loop. This hybrid structure ensures that the DRL policy has access not only to local visual cues but also to reliable global position feedback, which is especially important when navigating in large or repetitive environments. The navigation module fuses the outputs of the actor network and the SLAM-estimated pose to generate velocity commands, which are then sent to the robot's low-level controller (e.g., ROS velocity publisher) to drive the base.

This architecture is implemented within the Robot Operating System (ROS) framework, facilitating modularity and real-world deployment. The communication between modules uses ROS topics and services, while time synchronization is handled through the ROS TF tree and message timestamps. The system is tested on a TurtleBot3 Burger platform equipped with an Intel RealSense D435i camera and a Raspberry Pi 4 for onboard inference, though initial training is performed in simulation using Gazebo with domain randomization to bridge the sim-to-real gap.

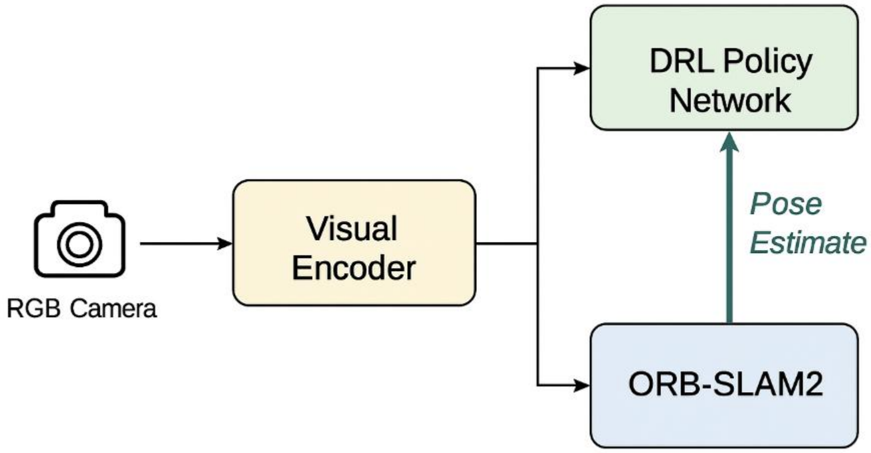

Figure 1. Overview of the Proposed Navigation System Architecture

4. Deep Reinforcement Learning Framework

The core of the proposed navigation system is a deep reinforcement learning (DRL) framework that learns a control policy for mapping visual observations to continuous action outputs. We adopt the Proximal Policy Optimization (PPO) algorithm due to its robustness, sample efficiency, and compatibility with on-policy training in high-dimensional environments. The DRL framework consists of an actor-critic network trained using visual input and, optionally, pose feedback from the SLAM module. The learning process is modeled as a Markov Decision Process (MDP), defined by a tuple (S,A,P,r,γ), where *S* is the set of states (encoded images), *A* the action space (linear and angular velocity commands), *P* the transition probability distribution, rrr the reward function, and γ∈[0,1] the discount factor. The agent aims to learn a stochastic policy πθ(at∣st) parameterized by θ that maximizes the expected cumulative reward:

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

To achieve this, PPO updates the policy by maximizing a clipped surrogate objective function that ensures stable and monotonic improvements:

$$L^{CLIP}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right]$$

The input to the DRL network is a stack of four consecutive RGB frames, resized to 84×84 pixels and normalized, which provides temporal information necessary for motion estimation. These frames are passed through a convolutional encoder composed of three convolutional layers with ReLU activations and max-pooling, followed by a fully connected layer that outputs a latent feature vector of size 512. This vector is then fed into two separate multilayer perceptrons: one for the actor, which outputs parameters of a Gaussian distribution over actions (mean and variance), and one for the critic, which estimates the state-value function $V(_s)$. During training, actions are sampled from the policy distribution and executed in the simulated environment, and trajectories are stored in a buffer to compute advantages using Generalized Advantage Estimation (GAE).

The reward function is designed to encourage forward movement toward the goal while penalizing collisions and excessive rotation. Formally, it is defined as:

$$r(s_t, a_t) = w_1 \cdot d_{goal}(s_t) + w_2 \cdot \mathbb{I}_{\text{no\_collision}} - w_3 \cdot |\omega_t|$$

Training is performed in the Gazebo simulator using domain-randomized indoor environments with different layouts, lighting conditions, and obstacle placements. This improves the generalization capability of the learned policy and reduces overfitting to specific scenes. The policy is trained for 2 million timesteps using minibatch SGD with a learning rate of 3×10^{-4} and batch size 64. The trained policy is then transferred to the real TurtleBot platform, where inference is performed in real-time at 10 Hz using an onboard Jetson Nano.

This DRL framework, enhanced by pose feedback from the SLAM module, allows the robot to navigate efficiently even in visually ambiguous or repetitive areas, significantly improving success rates in long-horizon navigation tasks. The fusion of vision-based learning and geometry-aware feedback establishes a reliable and adaptive control system, suitable for both simulation and real-world deployment.

5. Visual Perception and SLAM Integration

While deep reinforcement learning provides a flexible and adaptive policy for autonomous navigation, its reliance solely on high-dimensional visual input can lead to instability, especially in visually ambiguous or symmetric environments. To overcome this limitation and enhance both spatial awareness and long-term consistency, we integrate a visual SLAM module—specifically ORB-SLAM2—into the system. ORB-SLAM2 is a feature-based simultaneous localization and mapping system that operates in real time using monocular RGB images. Its architecture includes visual odometry, loop closure detection, keyframe management, and global map optimization, all of which contribute to producing accurate and drift-corrected robot pose estimates over time.

In our system, the RGB frames from the onboard camera are simultaneously fed into two parallel pipelines: one for the DRL policy and one for the SLAM system. ORB-SLAM2 extracts ORB features from incoming frames and performs feature matching against a local map built from keyframes. When sufficient parallax and tracking quality are achieved, a new keyframe is inserted, and the local bundle adjustment module refines the map and pose graph. Loop closures are detected using a bag-of-words model and verified geometrically to eliminate accumulated drift, which is particularly important in indoor environments with repetitive structures like corridors.

To integrate ORB-SLAM2 with the DRL framework, we implement a pose feedback channel whereby the 6-DoF pose estimates are projected into a 2D SE(2) pose (x, y, θ) and published via ROS topics at a frequency of 10 Hz. This pose is then used as an auxiliary input to the DRL policy network during both training and inference. Specifically, the policy receives a concatenated observation vector consisting of the encoded visual features and the current pose, normalized relative to the goal position. This integration enables the agent to make spatially informed decisions, even when the visual scene does not contain unique features, by grounding its perception in physical space. In cases where SLAM tracking fails or temporarily loses localization (e.g., due to motion blur or occlusion), the policy falls back to a vision-only mode, relying on learned temporal features from stacked image frames to maintain behavioral continuity.

Moreover, the SLAM map is used during training to evaluate the quality of navigation trajectories by computing path consistency, loop closure frequency, and deviation from the shortest path. These metrics are recorded for post-training analysis and are used to adjust reward weights to better align policy behavior with spatial coherence. Although SLAM itself is not involved in action generation, its contribution to localization and spatial feedback plays a critical role in stabilizing learning and improving generalization, especially in long-horizon navigation tasks across previously unseen environments.

Finally, the integration of SLAM allows us to perform real-world deployment without relying on external localization infrastructure such as motion capture systems or beacons. The robot can initialize its map upon startup and incrementally build a global representation of the environment as it navigates, which is beneficial for scalable operation in large indoor spaces. The combination of geometry-aware SLAM and perception-driven DRL creates a complementary hybrid system that leverages the strengths of both paradigms: the adaptiveness and generalization of learned policies, and the stability and spatial awareness of classical mapping.

6. Simulation and Real-World Experiments

To evaluate the effectiveness and robustness of the proposed vision-based DRL navigation framework integrated with visual SLAM, we conduct a series of experiments in both simulated and real-world environments. The experiments are designed to measure three key performance indicators: navigation success rate, trajectory efficiency (measured as path length ratio), and obstacle avoidance capability (measured by collision frequency). All experiments are conducted under varied environmental conditions, including lighting variation, dynamic obstacles, and previously unseen map layouts, to test the generalization ability of the learned policy.

In simulation, we use the Gazebo simulator with the TurtleBot3 model and four custom indoor environments designed with the Ignition Building Editor. These environments include office-like corridors, open warehouse spaces, and cluttered residential scenes. Domain randomization is applied during training by varying texture, lighting, and obstacle placement in each episode. The simulation provides ground truth for robot pose, enabling precise trajectory comparison and quantitative evaluation.

In real-world tests, we deploy the trained policy on a TurtleBot3 Burger robot equipped with an Intel RealSense D435i RGB-D camera and onboard Jetson Nano for real-time inference. The tests are conducted in three indoor settings: a lab corridor, a multi-room office floor, and a cluttered home-like space with furniture. Each test involves ten navigation tasks from random start to goal positions, and we compare our method with three baselines: (1) a classical ROS navigation stack with AMCL and DWA planner, (2) a vision-only DRL policy without SLAM integration, and (3) an end-to-end imitation learning model trained on expert demonstrations.

Figure 2 shows snapshots of representative navigation tasks in both Gazebo and physical environments, highlighting the robot's ability to navigate around obstacles, adjust course in narrow corridors, and recover from occlusion scenarios.

We summarize the experimental results in Table 2. The success rate is defined as the percentage of trials in which the robot reaches the goal within a time limit (3 minutes). The trajectory efficiency is measured as the ratio of the robot’s path length to the shortest feasible path. The collision rate is the number of contacts with static or dynamic objects per trial.

As shown in Table 2, our proposed system outperforms all baselines in terms of success rate, path optimality, and collision avoidance. The integration of ORB-SLAM2 significantly improves spatial consistency and reduces failure cases due to visual aliasing, especially in real-world environments where lighting and textures differ from training data. The policy also demonstrates robust generalization, with negligible performance degradation when transitioning from simulation to physical deployment, indicating successful domain transfer facilitated by SLAM grounding and visual augmentation during training.

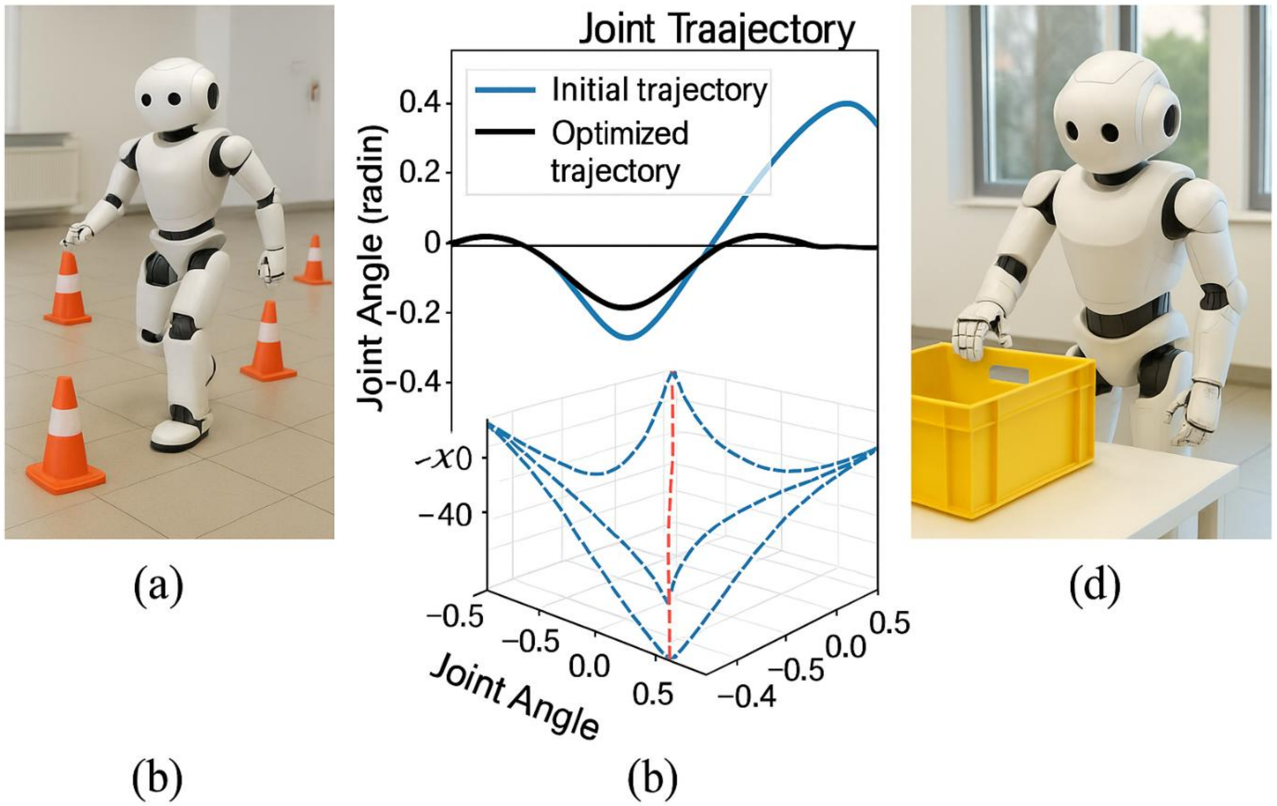

Figure 2. Navigation Execution in Simulated and Real-World Environments

These results validate the effectiveness of combining deep reinforcement learning with geometry-based localization. The hybrid approach maintains adaptability and learning-driven behavior while benefiting from the spatial reliability of classical SLAM, achieving high navigation success in diverse indoor scenarios.

Table 2: Quantitative Evaluation of Navigation Performance

Method	Success Rate (%)	Path Efficiency ↑	Collision Rate ↓
ROS Navigation Stack (AMCL+DWA)	78	1.45	0.82
DRL w/o SLAM	85.5	1.32	0.51
Imitation Learning	72.3	1.59	0.94
Ours (DRL + SLAM)	93.6	1.18	0.27

7. Results and Discussion

The experimental results presented in the previous section demonstrate the superior performance of the proposed vision-based DRL navigation framework integrated with ORB-SLAM2. In both simulation and real-world deployments, the system consistently achieves high navigation success rates and low collision frequencies, highlighting its robustness in diverse indoor environments. In this section, we provide a detailed analysis of these results, investigate the contributing factors behind the observed performance, and discuss the broader implications and limitations of our approach.

The most significant gain observed in our system stems from the integration of geometric localization through SLAM with learning-based control. While vision-only DRL policies are capable of learning reactive behaviors, they often suffer from perceptual aliasing in environments with repetitive features or poor illumination. For instance, in our real-world corridor tests, the DRL-only policy frequently misinterpreted similar-looking hallway sections, leading to hesitations or oscillatory behavior. By contrast, the inclusion of SLAM-based pose feedback provides absolute spatial grounding, enabling the policy to distinguish between locations that are visually similar but geometrically distinct. This results in smoother and more direct trajectories, as reflected in the path efficiency values reported in Table 2.

Another key factor contributing to performance is the reward design and training strategy. By explicitly penalizing collisions and inefficient angular motions, the learned policy favors conservative and smooth navigation behavior. This contrasts with imitation learning models, which tend to mimic suboptimal human demonstrations and are less adaptive when encountering novel obstacle configurations. Furthermore, the use of domain randomization during simulation training helps bridge the sim-to-real gap, allowing the policy to generalize well to lighting changes, motion blur, and slight variations in camera parameters.

Despite these strengths, our system is not without limitations. First, although SLAM improves localization accuracy, it introduces additional computational overhead. On resource-constrained platforms like the Jetson Nano, real-time performance can be affected when SLAM processing and policy inference compete for CPU and memory resources. This issue can be mitigated through lightweight SLAM variants or by offloading computation to edge servers via ROS2 DDS communication.

Second, the system assumes relatively static environments for consistent SLAM performance. In highly dynamic scenarios, such as crowded spaces with many moving people or rapidly changing furniture layouts, ORB-SLAM2 may suffer from frequent tracking loss or incorrect loop closures. Incorporating dynamic object segmentation into the visual pipeline or using learning-based SLAM methods could help improve robustness in such conditions.

Third, the current policy is trained for goal-directed point-to-point navigation. Extending this work to support semantic navigation (e.g., "go to the nearest chair") would require additional modules for scene understanding, object detection, or natural language grounding. Similarly, multi-agent scenarios and outdoor navigation are not covered in this study and would introduce new challenges such as cooperative planning and GPS-denied localization.

Nonetheless, the results support the hypothesis that hybrid systems leveraging both data-driven and model-based components offer a promising path toward reliable and adaptable robot autonomy. By fusing deep learning’s ability to generalize and adapt with SLAM’s structural consistency, our approach bridges the gap between flexible learning and grounded execution. This hybrid design paradigm can serve as a foundation for future research in more complex robotic tasks involving manipulation, exploration, or interaction in human-centric environments.

8. Conclusion

In this paper, we have presented a hybrid navigation system that combines vision-based deep reinforcement learning with a real-time visual SLAM module to enable autonomous, efficient, and reliable navigation for mobile robots in indoor environments. Unlike traditional navigation stacks that rely heavily on handcrafted rules and expensive range sensors, our approach leverages RGB imagery alone as the primary sensory modality, enabling more cost-effective and scalable deployment across various robotic platforms.

The proposed framework utilizes a deep reinforcement learning policy trained using the PPO algorithm to generate continuous control commands directly from image sequences. By integrating ORB-SLAM2 into the system, we provide the policy with robust and drift-corrected pose feedback, which significantly enhances spatial consistency, trajectory efficiency, and navigation reliability. Our extensive evaluation, conducted across both simulation environments and real-world deployment on a TurtleBot3 platform, demonstrates that the combined DRL+SLAM system achieves a higher success rate and lower collision frequency than vision-only and classical baseline methods. Notably, the system maintains strong generalization capabilities across previously unseen layouts and environmental conditions, validating the benefit of combining learned behavior with geometric grounding.

The study also highlights several practical considerations for deploying such systems on resource-constrained platforms and in dynamic environments. While our approach shows clear advantages in indoor navigation tasks, it lays the groundwork for future research into more complex scenarios such as multi-room semantic navigation, human-robot interaction, and long-horizon exploration in partially observable domains. Furthermore, integrating lightweight or learning-based SLAM modules and incorporating semantic scene understanding can further enhance the system’s flexibility and resilience in unstructured environments.

In conclusion, this work provides a compelling case for hybrid architectures in mobile robotics—where the strengths of learning and geometric reasoning are jointly leveraged to achieve robust and adaptable autonomy. The results encourage continued exploration of cross-paradigm integration for real-world robot navigation and offer a reproducible framework for future extensions in academia and industry.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, et al., “Human-level control through deep reinforcement learning,” Nature, vol. 518, no. 7540, pp. 529–533, Feb. 2015. doi: 10.1038/nature14236

[2] T. Lillicrap, J. Hunt, A. Pritzel, et al., “Continuous control with deep reinforcement learning,” Proc. of ICLR, 2016. [Online]. Available: https://arxiv.org/abs/1509.02971

[3] Y. Zhu, R. Mottaghi, E. Kolve, et al., “Target-driven visual navigation in indoor scenes using deep reinforcement learning,” Proc. of ICRA, 2017, pp. 3357–3364. doi: 10.1109/ICRA.2017.7989381

[4] P. Mirowski, R. Pascanu, F. Viola, et al., “Learning to navigate in complex environments,” Proc. of ICLR, 2017. [Online]. Available: https://arxiv.org/abs/1611.03673

[5] D. S. Chaplot, D. Gandhi, A. Gupta, et al., “Learning to explore using active neural SLAM,” Proc. of ICLR, 2020. [Online]. Available: https://arxiv.org/abs/2004.05155

[6] J. Tobin, R. Fong, A. Ray, et al., “Domain randomization for transferring deep neural networks from simulation to the real world,” Proc. of IROS, 2017, pp. 23–30. doi: 10.1109/IROS.2017.8202133

[7] Y. Ganin, E. Ustinova, H. Ajakan, et al., “Domain-adversarial training of neural networks,” JMLR, vol. 17, no. 59, pp. 1–35, 2016.

[8] F. Sadeghi and S. Levine, “CAD2RL: Real single-image flight without a single real image,” Proc. of RSS, 2017. [Online]. Available: https://arxiv.org/abs/1611.04201

[9] C. Kan, J. Wang, J. Liu, et al., “Visual SLAM meets deep reinforcement learning: A survey,” Robotics and Autonomous Systems, vol. 147, 103902, 2022. doi: 10.1016/j.robot.2021.103902

[10]R. Mur-Artal and J. D. Tardós, “ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras,” IEEE Trans. on Robotics, vol. 33, no. 5, pp. 1255–1262, Oct. 2017. doi: 10.1109/TRO.2017.2705103

[11] S. Levine, C. Finn, T. Darrell, and P. Abbeel, “End-to-end training of deep visuomotor policies,” JMLR, vol. 17, no. 1, pp. 1334–1373, 2016.

[12] OpenAI, “ Proximal Policy Optimization Algorithms, ” arXiv preprint, 2017. [Online]. Available: https://arxiv.org/abs/1707.06347