engineering proceedings

MDPI

*Proceeding Paper*

# Human in the Loop: A Comparative Study of Human and Artificial Visual Search Strategies in a VR-Based Dynamic Environment for Defense Applications †

**Ferenc Samu *, Ádám Koleszár and Ágoston Pál Sándor**

Audi Hungaria Faculty of Engineering, University of Győr, Egyetem tér 1, 9026 Győr, Hungary; koleszar.adam@o365.sze.hu (Á.K.); sandor.agoston.pal@o365.sze.hu (Á.P.S.)

* Correspondence: samu.ferenc@o365.sze.hu

† Presented at the Sustainable Mobility and Transportation Symposium 2026, Győr, Hungary, 8 October 2026.

**Abstract**

In defense applications such as close-quarters combat and autonomous reconnaissance by unmanned systems (UAVs/UGVs), the rapid identification of concealed objects is a critical challenge. This pilot study compares human and AI-based search efficiency in a controlled 3D environment. An Unreal Engine 5 simulation was developed, with targets placed in demanding configurations such as occlusion, above eye level, and unusual viewing angles. The artificial agent was trained through Reinforcement Learning (RL), using the Environment Query System (EQS) for high-resolution spatial sampling and a Dynamic Behaviour Tree for adaptive control. Its performance was compared with human participants using Meta VR devices across three key performance indicators: movement trajectory, search time, and visual-focus heatmaps. The results highlight the contrast between the mathematical precision of RL-based algorithms and the context-driven intuition of human search, emphasizing the potential for optimizing object-search processes in defense-related applications.

**Keywords:** Unreal Engine 5; Meta VR; Environment Query System (EQS); reinforcement learning; Behaviour Tree; defense industry; autonomous reconnaissance

## 1. Introduction

An increasingly realistic direction in defense and emergency-response technology is the use of digitally scanned or reconstructed spaces as safe training grounds for artificial agents. If AI agents can be trained and validated in a faithful virtual counterpart of a real environment before deployment, parts of reconnaissance, search, and route-assessment tasks may be shifted away from human personnel, reducing both manpower demands and exposure to hazardous situations. This prospect is no longer purely speculative: recent work on sim-to-real transfer and visual domain randomization shows that virtual training environments remain central to embodied robotics research and deployment preparation [1,2].

Virtual environments are also already being used to prepare human first responders. Recent studies have proposed advanced VR training frameworks for first responders and simulation-based VR training for firefighters, showing that immersive digital spaces can support operational rehearsal without exposing trainees to real-world danger [3,4]. Building on this broader trajectory, the present pilot study investigates how an RL-based search agent and human participants differ in the same Unreal Engine 5 environment

Check for updates

Academic Editors: Gábor Dogossy, Szabolcs Kocsis-Szürke, Dániel Csikor, Dániel Serfőző, Miklós Kuczmann, András Lajos Nagy, László Lendvai and Ambrus Zelei

Published: 24 September 2026

**Copyright:** © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

during visually demanding target-search tasks. The goal is not only to compare search efficiency, but also to examine how structured AI control and human visual intuition diverge under comparable conditions.

## 2. Literature Review

This section situates the present study within four closely related research areas: autonomous target search and navigation, high-fidelity simulation and adaptive control, human visual search, and human-robot teaming. Rather than treating these domains separately, the review highlights their methodological overlap and their joint relevance for comparing human and artificial search behavior in the same virtual environment.

### *2.1. Autonomous Object Search and Navigation in Defense Applications*

In contemporary defense-related settings, particularly in Close-Quarters Battle (CQB) operations and the autonomous reconnaissance of unmanned systems (UAVs/UGVs), target search must be performed under dynamic and only partially observable conditions. Within this context, reinforcement learning has remained a major framework for autonomous navigation, and recent survey literature confirms its continued relevance in mobile-robot control [5]. At the same time, object-goal navigation has developed into a well-defined embodied-AI problem, with recent work systematizing the available datasets, metrics, and methodological approaches [6]. Recent target-driven methods, including training-free semantic-frontier strategies, further demonstrate that efficient object search in complex indoor scenes can be achieved without relying exclusively on end-to-end retraining [7]. Complementing this line of work, reinforcement-learning-based active-perception studies indicate that uncertainty reduction during navigation can itself be treated as a learnable component of the search process [8].

In real-world reconnaissance tasks, target identification frequently relies on state-of-the-art image-based computer vision models. The most prominent among these is the YOLO (You Only Look Once) architecture, which revolutionized real-time object detection by framing the task as a single regression problem rather than a complex multi-stage classification pipeline [9]. YOLO and its subsequent iterations achieve high accuracy and fast processing speeds by directly predicting bounding boxes and class probabilities from full images, making them standard for real-time UAV and UGV payloads [10].

However, the current study intentionally deviates from this image-based detection concept. Instead of simulating the full pixel-level processing pipeline required by YOLO, our approach abstracts perception using Unreal Engine's native sensory components (AI Perception). This deviation isolates the agent's navigational decision-making from the computational burden of pixel processing, allowing the study to focus purely on the spatial, logic-driven, and cognitive dimensions of search strategies—comparing human visual intuition against AI navigation rather than evaluating the accuracy of image recognition algorithms.

### *2.2. High-Fidelity Simulation Environments and Adaptive Control*

Because real-world experimentation is costly, time-consuming, and often unsafe, high-fidelity simulation continues to play a central role in the development of autonomous robotic systems. Recent survey work on sim-to-real transfer confirms that the gap between training and deployment remains a major research challenge in deep reinforcement learning for robotics [1]. More recent empirical studies have therefore focused on quantifying the benefits of domain randomization for visual transfer [2] and on developing offline 62 randomization strategies that improve policy robustness across environments [11]. In parallel with advances in transfer learning, research on control architectures has continued

to explore ways of combining structured decision logic with adaptive policy learning. In this regard, mixed deep reinforcement learning and Behavior Tree frameworks illustrate how reactive, interpretable control can be integrated with data-driven adaptation in intelligent-agent design [12].

### 2.3. Visual Search Strategies: Cognitive Human Intuition vs. Algorithmic Precision

From the perspective of human cognition, visual search remains strongly shaped by top-down goals, contextual expectations, and flexible attentional control. Recent experimental studies show, for example, that contextual cuing does not guide attention independently of task goals [13], and that the suppression of distracting features can be adaptively modulated according to the current search objective [14]. By contrast, embodied AI systems are commonly assessed through navigation success, exploration efficiency, and task completion, rather than through internal attentional strategies in the human sense [6]. This difference makes direct human-AI comparison methodologically challenging, but also scientifically valuable. In immersive three-dimensional settings, eye tracking in virtual reality offers a particularly suitable basis for such comparisons, as recent review literature shows that VR-based gaze measurement can support the analysis of trajectories, fixations, and heatmap-like attention patterns with increasing maturity and methodological precision [15].

### 2.4. The Research Gap and the Purpose of the Present Study

Despite substantial progress in embodied navigation, sim-to-real transfer, and human visual-behavior analysis, studies that directly compare human and AI search within the same controlled 3D environment remain limited. This gap is particularly relevant in light of recent human-robot teaming research, which emphasizes the importance of adaptive coordination [16] and identifies task recognition, shared situational awareness, and context-sensitive interaction as key requirements for effective teaming [17]. The present pilot study addresses this gap by comparing an RL-based search agent and human participants in the same Unreal Engine 5 environment, using movement trajectory, search time, and visual heatmaps as the primary performance indicators. In doing so, it seeks to connect embodied AI, VR-based behavioral measurement, and human-agent comparison within a single experimental framework.

Recent implementation-oriented studies further support the architectural choices adopted in this work. Grid/cell-based coverage formulations with actor–critic RL indicate that explicit spatial discretization and geometric obstacle exclusion can improve stability in full-area coverage learning for cluttered scenes [18]. UAV target-search studies likewise report that terminal-only sparse rewards often degrade training, whereas incremental reward shaping for progress and newly explored regions is important for stable policy acquisition [19]. Unreal-focused RL platform studies further show that numerical details, including observation normalization and safe handling of degenerate values, can critically affect training robustness [20]. Finally, recent BT–RL integration studies describe a recurrent trade-off: pure Behavior Trees are typically robust but less adaptive, pure RL is adaptive but less predictable, and hybrid control can offer a balanced compromise with improved recovery behavior [21,22].

## 3. Methodology

The artificial search system was implemented in Unreal Engine 5 as a hybrid architecture. Conventional AI modules provide perception, state management, room-level search sequencing, and task control, while the Learning Agents framework provides reinforcement learning (RL)-based movement optimization. This separation makes the

baseline behavior deterministic and interpretable, while still enabling policy improvement through experience.

### *3.1. AI Agent*

The robot is based on the default actor of the Unreal Engine third-person template, which already provides the visual body and locomotion animations. The actor class was extended with components required by the present search task and by the integration with the Behavior Tree and Learning Agents Manager. Movement is controlled by a dedicated AI Controller, which starts the Behavior Tree, updates Blackboard entries, and also hosts the robot's effective "eye."

### *3.2. AI Perception*

Environmental sensing is implemented with AI Perception rather than image-based computer vision. In the simulated environment, this mechanism acts as a perception envelope with a configured field of view. The robot can detect any non-occluded actor carrying the appropriate perceptual tag. This is how the target becomes visible to the agent.

### *3.3. Target Actor*

The target actor is intentionally minimal and contains only an AI Perception Stimuli Source component, making it detectable by the search robot without adding unrelated behavior.

### *3.4. Behavior Tree*

High-level decision-making is organized by a Behavior Tree. The tree receives runtime data from the AI Controller, stores state variables in the Blackboard, and selects actions according to room status, target status, and reset conditions. The Behavior Tree also supports RL by handling intermediate sequencing tasks while the policy controls local movement.

### *3.5. Blackboard*

The Blackboard acts as a shared working memory for the Behavior Tree. The main keys are:

- SelfActor: reference to the robot actor.
- FoundTarget: set when a target is detected.
- TargetToFollow: actor reference used by the success branch when a target is available.
- SearchLocation: candidate position returned by EQS.
- GoToNextRoom: indicates that the current room sweep is complete.
- CurrentRoom: room currently under investigation.
- DoorLocation: room entry point where search begins.
- OutOfBounds: indicates that the robot left the room before full coverage.
- TimeToRestart: restart intent after successful completion.

### *3.6. Environment Query System*

The Environment Query System (EQS) generates candidate navigation points inside the currently active room. Candidate points are filtered and weighted using three criteria: (i) reachability, so unreachable points are removed; (ii) distance preference, favoring efficient relocation; and (iii) occupancy/collision validity, preventing selections that intersect level geometry. Based on these weighted candidates, the Behavior Tree selects the next search point while excluding already covered regions via the Coverage Manager.

Let $\mathcal{P}_r = \{p_1, \ldots, p_N\}$ denote candidate points in room $r$, and let the valid set be

$$\mathcal{C}_t = \{p_i \in \mathcal{P}_{r_t} \mid \rho_i = 1,\ \kappa_i = 1\},$$

https://doi.org/10.3390/engproc2026157014

where $\rho_i \in \{0,1\}$ indicates path reachability and $\kappa_i \in \{0,1\}$ indicates collision validity. For each valid point, the EQS score is

$$S_i(t) = w_d\,\phi_d(p_i,t) + w_u\,\phi_u(p_i,t) + w_c\,\phi_c(p_i),$$

where $\phi_d$ is a distance-based preference term, $\phi_u$ favors unexplored regions, and $\phi_c$ encodes occupancy validity. The selected search location is

$$\text{SearchLocation}_t = \underset{p_i \in \mathcal{C}_t}{\text{argmax}} S_i(t),$$

or, in stochastic mode, a sample from the top-$K$ scored points.

### *3.7. Behavior Tree Execution Logic*

Execution follows a left-to-right selector hierarchy:

- If TimeToRestart is true, restart tasks are executed.
- Else, if TargetToFollow is valid, the success sequence runs: the robot moves to the target and waits for two seconds.
- Else, the tree evaluates room-level search:
  - If GoToNextRoom is true, the next room is selected and the robot moves to its entry location.
  - Otherwise, the search sequence runs EQS, updates SearchLocation, and delegates local motion to the RL policy, with periodic re-selection of search locations to complete room coverage.

Formally, if $\mathcal{B} = \{b_1, \ldots, b_m\}$ is the ordered branch set (left-to-right priority), branch selection at time $t$ is

$$b_t = \underset{b \in \mathcal{B}}{\text{argmax}}\ \pi(b)\,\mathbf{1}[c_b(x_t) = 1],$$

where $x_t$ is the Blackboard state, $c_b(\cdot)$ is the branch condition, $\mathbf{1}[\cdot]$ is the indicator function, and $\pi(b)$ is a strictly decreasing priority weight from left to right.

### *3.8. Learning Agents*

The Learning Agents subsystem provides the adaptive component of the architecture by implementing RL-based motion control.

#### 3.8.1. Learning Agents Interactor

The Interactor defines the observation and action interface. The observation vector contains:

- EQS target position,
- agent speed,
- distance-sensor line readings around the agent,
- forward direction,
- lateral direction,
- normalized step-progress value,
- percentage of already covered room points.

The action vector is continuous and contains three outputs: forward movement, right turn, and right strafe.

Using compact notation, the policy input-output mapping is

$$s_t = \left[\Delta p_t^{\text{eqs}},\ v_t,\ d_t^{(1)}, \ldots, d_t^{(M)},\ \hat{f}_t,\ \hat{l}_t,\ \eta_t,\ \zeta_t\right],$$

$$a_t = \pi_\theta(s_t) = [u_t,\ \omega_t,\ q_t] \in [-1,1]^3,$$

https://doi.org/10.3390/engproc2026157014

where $\Delta p_t^{\text{eqs}}$ is the relative EQS target vector, $v_t$ is speed, $d_t^{(j)}$ are distance-sensor readings, $\hat{f}_t$ and $\hat{l}_t$ are forward/lateral direction descriptors, $\eta_t$ is normalized step progress, and $\zeta_t$ is the covered-point ratio.

#### 3.8.2. Learning Agents Training Environment

The Training Environment defines rewards, episode termination, and reset logic. Reward shaping includes the following terms:

- Large positive reward when a target is found;
- Positive reward when moving closer to the EQS-selected point;
- Continuous time penalty;
- Negative reward when distance sensors indicate unsafe proximity to walls;
- Large negative reward if the robot leaves a room before complete coverage;
- Positive reward when newly visible EQS points are discovered;
- Large positive reward when a room is fully observed with no uncovered points remaining.

The shaped reward can be written as

$$\begin{aligned} r_t = {} & \alpha_f I_t^{\text{found}} + \alpha_p \Delta_t^{\text{eqs}} - \alpha_t \\ & - \alpha_w \sum_{j=1}^{M} \max\left(0, d_{\text{safe}} - d_t^{(j)}\right) - \alpha_o I_t^{\text{oob}} \\ & + \alpha_n n_t^{\text{new}} + \alpha_c I_t^{\text{cover}}, \end{aligned}$$

where $I_t^{\text{found}}$ indicates target detection, $\Delta_t^{\text{eqs}}$ is progress toward the current EQS point, $I_t^{\text{oob}}$ indicates out-of-bounds room exit, $n_t^{\text{new}}$ counts newly observed room points, and $I_t^{\text{cover}}$ indicates full room coverage.

Episodes also end periodically for training stability, and can terminate early on task-completion events. In the present setup, once all three targets are found, the run is restarted.

The optimization objective is

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T} \gamma^t r_t\right],$$

with discount factor $\gamma \in (0, 1)$.

At reset, the agent returns to its initial position and restarts the search from the beginning of the map while clearing target-discovery state.

#### 3.8.3. Learning Agents Manager

The Learning Agents Manager assembles and initializes the full RL stack: Interactor, Policy, Critic, Training Environment, Communicator, and Trainer. During runtime, learning updates are executed continuously (frame-wise). Once training is considered sufficient, the system is switched to inference mode, where the robot uses the learned policy without further parameter updates.

### *3.9. Coverage Manager*

The Coverage Manager implements spatial memory for room exploration. During runtime, it tracks which EQS room points are currently visible and removes newly observed points from the set of unexplored points. When this set becomes empty, the room is marked as fully covered.

If $\mathcal{U}_t$ is the unexplored-point set and $\mathcal{V}_t$ is the currently visible set, then

$$\mathcal{U}_{t+1} = \mathcal{U}_t \setminus \mathcal{V}_t,$$

https://doi.org/10.3390/engproc2026157014

and coverage ratio is

$$\zeta_t = 1 - \frac{|\mathcal{U}_t|}{|\mathcal{U}_0|}.$$

### *3.10. Room Manager*

The Room Manager stores room definitions and their entry locations, provides this data to the Behavior Tree, and manages room states (not visited, currently explored, completed). It selects the next room when the current one is finished.

Let each room state be $q_r \in \{0, 1, 2\}$ for not visited, active, and completed. A simple next-room policy is

$$r_{t+1} = \arg \min_{r \in \mathcal{R} : q_r \neq 2} \psi(r, t),$$

where $\psi(r, t)$ is a selection cost (e.g., travel distance to room entry, optionally with priority penalties).

### *3.11. Human Participants*

The human cohort in this pilot study consisted of three participants (mean age = 22.0 years, SD = 1.0). The participants represented diverse backgrounds regarding gaming and virtual reality (VR) familiarity: one participant (age 21) reported no regular gaming habits and no prior VR experience; the second participant (age 22) reported an average of 10 h of video gaming per week alongside prior VR experience; and the third participant (age 23) reported an average of 15 h of gaming per week, also with prior VR experience. Given the limited number of participants, this research is explicitly considered a pilot study. The descriptive statistics and behavioral metrics presented in the results are intended to yield preliminary observations and qualitative comparisons rather than broadly generalizable inferential conclusions.

## 4. Results

Before detailing the numerical outcomes, it is important to outline the evaluation procedure and its statistical considerations. The primary performance measures in this study include total search time, traveled path length, and visual-behavioral metrics (e.g., head rotation and look-vs-move offset). Given the pilot nature of this study, the evaluation was conducted with a limited number of human participants and AI configurations. Consequently, applying robust inferential statistical tests (such as ANOVA or *t*-tests) to this small sample size would yield underpowered and potentially misleading results. Therefore, the current evaluation deliberately relies on descriptive statistics to highlight qualitative trends and behavioral differences. Future research involving larger participant cohorts will incorporate appropriate inferential statistical methods to validate these preliminary observations and support broader generalizability.

Table 1 summarizes the two most directly comparable outcome measures across all sessions: total search time and traveled path length. Path length is reported in Unreal Engine world units, where 1 unit corresponds to 1 cm. All human participants and all AI configurations found all three target objects. Human participants completed the task faster in every recorded session, with total search times between 2 min 12.3 s and 3 min 48.3 s, whereas the AI runs ranged from 4 min 35.8 s to 13 min 21.4 s. At the same time, the AI runs covered substantially shorter paths than the human runs. The behavior-tree-only baseline completed the task in 4 min 35.8 s with a path length of 49,810 units, making it the fastest of the AI configurations in this experiment. The shortest AI path, however, was obtained by the 2 h RL configuration at 41,520 units, but with a slower completion time of 5 min 59.6 s. In this limited comparison, the behavior-tree-only baseline therefore appears

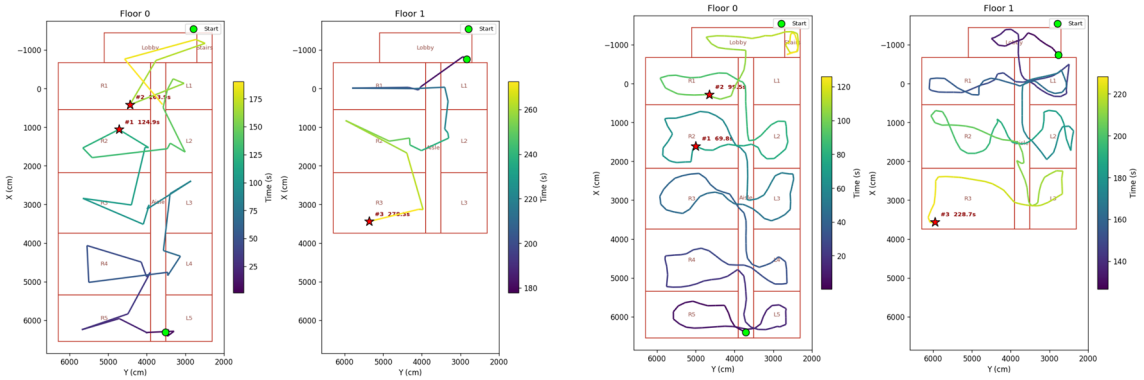
to be the most pragmatic AI solution when completion time is prioritized, while longer RL training improved path compactness rather than overall speed.

**Table 1.** Summary of total search time and path length.

| Session | Total Search Time | Path Length (Units) |
|---|---|---|
| AI—Behavior Tree only (no RL) | 4 m 35.8 s | 49,810 |
| AI—RL (45 min training) | 13 m 21.4 s | 53,960 |
| AI—RL (1.5 h training) | 6 m 56.2 s | 43,470 |
| AI—RL (2 h training) | 5 m 59.6 s | 41,520 |
| Human P1 | 3 m 48.3 s | 88,880 |
| Human P2 | 2 m 12.3 s | 72,560 |
| Human P3 | 2 m 24.7 s | 82,200 |

The additional behavioral metrics show a similarly clear contrast. Human participants exhibited substantially more head movement, with total head rotation between 8397° and 13,273° and mean head-turn rates between 58°/s and 64°/s, whereas the AI configurations ranged from 4569° to 7265° and from 9°/s to 21°/s. The median look-vs-move offset was also much smaller for humans (3°–31°) than for the AI runs (78°–92°), indicating tighter coupling between viewing direction and locomotion in the human trials. Dwell-time patterns also separated the AI configurations: the 45 min RL run concentrated 22% of its time in 0/Aisle and 19% in 0/L3, while the behavior-tree-only baseline showed a flatter profile, with its highest room share at 10% in 0/R2. Within the present dataset, this supports the interpretation that the behavior-tree baseline maintained more even spatial coverage, whereas early RL training produced more localized dwell behavior.

For a qualitative example, Figure 1 compares representative path visualizations for the behavior-tree AI run and Human P1.




(**a**) AI path example (Behavior Tree) (**b**) Human P1 path example

**Figure 1.** Example qualitative path comparison between AI and human runs.

## 5. Conclusions and Discussion

This pilot comparison showed a clear performance split. Human participants completed all sessions faster, while AI runs generally followed shorter paths. Among the evaluated AI configurations, the behavior-tree-only baseline achieved the best completion time, whereas longer RL training improved path compactness but did not yet yield the best end-to-end time. This pattern is consistent with the trade-offs discussed in recent BT–RL literature [19,20]: structured BT logic provides reliable high-level task flow, while RL contributes adaptive local behavior that may require longer and more stable training to translate into faster overall completion.

Several specific factors likely explain why the RL-enhanced agents did not outperform the pure Behavior Tree baseline in terms of absolute search time. First, the restricted training

duration (up to 2 h) may not have allowed the agent to perfectly optimize the balance between spatial exploration and traversal speed. Second, the design of the reward function—which heavily penalized unsafe proximity to walls and leaving a room before complete coverage—likely induced an overly cautious movement policy. While this resulted in highly efficient and compact paths, it constrained the raw speed of the agent. In contrast, the pure Behavior Tree baseline executed a deterministic, continuous search logic without learned hesitations, which ultimately yielded a faster overall completion time despite producing a longer, less refined trajectory.

In addition to reward shaping, the complexity of the continuous state-action space presents a significant challenge for the exploration-exploitation balance. Optimizing this balance across a relatively small training dataset means the RL agent prioritizes safe, localized exploitation over rapid, global exploration. AI-driven optimization techniques have demonstrated strong potential for improving performance and decision-making under uncertainty in complex, dynamic logistics systems [23]. This aligns conceptually with the adaptive requirements of autonomous agents operating in defense-oriented environments. Ultimately, the observed differences underscore a fundamental divergence highlighted in embodied AI literature: while artificial agents execute mathematically precise, exhaustive spatial coverage, human visual search is heavily guided by top-down cognitive intuition and contextual expectations.

From an engineering perspective, the present results support the use of a BT baseline for time-critical deployment, while continuing RL fine-tuning under robust reward shaping and numerically stable Unreal training pipelines [18–20]. Because the current dataset is limited in scale, any claim that hybrid control will eventually outperform the baseline should be treated as a testable hypothesis rather than a demonstrated conclusion.

From a practical perspective, the proposed approach holds significant implications for defense applications and autonomous reconnaissance. By integrating structured logic with adaptive learning, such hybrid systems can power decision-support technologies that reduce human exposure in hazardous close-quarters environments. However, the current study is subject to several limitations: the sample size of human participants is small, the evaluation was conducted within a single virtual scenario, and the RL training duration was restricted. Future research directions will address these limitations by involving larger participant cohorts to enable rigorous inferential statistical analysis, integrating precise eye-tracking technologies for a more granular measurement of visual attention, and evaluating alternative deep reinforcement learning algorithms and hybrid control architectures.

**Author Contributions:** Conceptualization, F.S., Á.K. and Á.P.S.; methodology, F.S. and Á.K.; software, F.S. and Á.K.; investigation, F.S. and Á.K.; resources, Á.K.; data curation, F.S.; formal analysis, Á.P.S.; visualization, Á.P.S.; writing—original draft preparation, Á.P.S.; writing—review and editing, Á.P.S., F.S. and Á.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

**Acknowledgments:** Project no. TKP2021-NVA-23 provided the institutional and professional framework for this research under the TKP2021-NVA scheme (Ministry of Technology and Industry of Hungary; National Research, Development and Innovation Fund). The authors declare that no financial funding was provided for the implementation of this specific research.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| AI | Artificial Intelligence |
| BT | Behavior Tree |
| CQB | Close-Quarters Battle |
| EQS | Environment Query System |
| RL | Reinforcement Learning |
| UAV | Unmanned Aerial Vehicle |
| UGV | Unmanned Ground Vehicle |
| VR | Virtual Reality |

## References


1. Zhao, W.; Pena Queralta, J.; Westerlund, T. Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: A Survey. In Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (SSCI), Canberra, Australia, 1–4 December 2020.
2. Alghonaim, R.; Johns, E. Benchmarking Domain Randomisation for Visual Sim-to-Real Transfer. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021.
3. Sofuoğlu, Z.; Sofuoğlu, T.; Pérez Robles, D.; Carjaval, F.R.; Didderen, E.; Veeneklaas, J.N.; Ba¸ser, A. First responders' innovative methodology and curriculum definition for advanced virtual reality training. *Health Sci. Q.* **2022**, *2*, 167–178. [CrossRef]
4. Hadj Sassi, M.S.; Battisti, F.; Carli, M. Simulation-based virtual reality training for firefighters. *Electron. Imaging* **2022**, *34*, 194-1–194-5. [CrossRef]
5. Jiang, H.; Wang, H.; Yau, W.-Y.; Wan, K.-W. A Brief Survey: Deep Reinforcement Learning in Mobile Robot Navigation. In Proceedings of the 2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA), Kristiansand, Norway, 9–13 November 2020.
6. Wang, D.; Chen, J.; Cheng, J. A Survey of Object Goal Navigation: Datasets, Metrics and Methods. In Proceedings of the 2023 IEEE International Conference on Mechatronics and Automation (ICMA), Harbin, China, 6–9 August 2023.
7. Chen, J.; Li, G.; Kumar, S.; Ghanem, B.; Yu, F. How To Not Train Your Dragon: Training-free Embodied Object Goal Navigation with Semantic Frontiers. In Proceedings of the Robotics: Science and Systems XIX, Daegu, Republic of Korea, 10–14 July 2023.
8. Mateus, M.G.; Grando, R.B.; Drews, P.L.J. Active Perception Applied To Unmanned Aerial Vehicles Through Deep Reinforcement Learning. In Proceedings of the 2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE), São Bernardo do Campo, Brazil, 18–21 October 2022.
9. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
10. Ma, C.; Fu, Y.; Wang, D.; Guo, R.; Zhao, X.; Fang, J. YOLO-UAV: Object Detection Method of Unmanned Aerial Vehicle Imagery Based on Efficient Multi-scale Feature Fusion. *IEEE Access* **2023**, *11*, 126857–126878. [CrossRef]
11. Tiboni, G.; Arndt, K.; Kyrki, V. DROPO: Sim-to-real transfer with offline domain randomization. *Robot. Auton. Syst.* **2023**, *166*, 104432. [CrossRef]
12. Li, L.; Wang, L.; Li, Y.; Sheng, J. Mixed Deep Reinforcement Learning-behavior Tree for Intelligent Agents Design. In Proceedings of the 13th International Conference on Agents and Artificial Intelligence, Vienna, Austria, 4–6 February 2021.
13. Luque, D.; Beesley, T.; Molinero, S.; Vadillo, M.A. Contextual cuing of visual search does not- guide attention automatically in the presence of top-down goals. *J. Exp. Psychol. Hum. Percept. Perform.* **2021**, *47*, 1080–1090. [CrossRef] [PubMed]
14. Forstinger, M.; Ansorge, U. Top-down suppression of negative features applies flexibly contingent on visual search goals. *Atten. Percept. Psychophys.* **2024**, *86*, 1120–1147. [CrossRef] [PubMed]
15. Adhanom, I.B.; MacNeilage, P.; Folmer, E. Eye Tracking in Virtual Reality: A Broad Review of Applications and Challenges. *Virtual Real.* **2023**, *27*, 1481–1505, Correction in *Virtual Real.* **2023**, *27*, 1569–1570. https://doi.org/10.1007/s10055-023-00781-4. [CrossRef]
16. Singh, S.; Heard, J. Human-Aware Reinforcement Learning for Adaptive Human Robot Teaming. In Proceedings of the 2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Sapporo, Japan, 7–10 March 2022.
17. Baskaran, P.; Adams, J.A. Multi-dimensional task recognition for human-robot teaming: Literature review. *Front. Robot. AI* **2023**, *10*, 1123374. [CrossRef] [PubMed]
18. Garrido-Castaneda, S.I.; Vasquez, J.I.; Antonio-Cruz, M. Coverage Path Planning Using Actor–Critic Deep Reinforcement Learning. *Sensors* **2025**, *25*, 1592. [CrossRef] [PubMed]
19. Feng, Z.; Na, X.; Hai, S.; Sun, Q.; Shi, J. Deep Reinforcement Learning for UAV Target Search and Continuous Tracking in Complex Environments with Gaussian Process Regression and Prior Policy Embedding. *Electronics* **2025**, *14*, 1330. [CrossRef]

20. Hu, T.; Fu, Q.; Pu, Z.; Wang, Y.; Qiu, T. Unreal-MAP: Unreal-Engine-Based General Platform for Multi-Agent Reinforcement Learning. *arXiv* **2025**, arXiv:2503.15947. [CrossRef]
21. Wagner, J. Improving Behavior Trees That Use Reinforcement Learning with Control Barrier Functions. Master's Thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2024.
22. Ahmad, F. Towards Self-Reliant Robots: Skill Learning, Failure Recovery, and Real-Time Adaptation: Integrating Behavior Trees, Reinforcement Learning, and Vision-Language Models for Robust Robotic Autonomy. Ph.D. Thesis, Department of Robotics and Semantic Systems, Lund University, Lund, Sweden, 2025.
23. Dejanović, M.; Panić, S.; Kontrec, N.; Đošić, D.; Milojević, S. Neural Network-Based Optimization of Repair Rate Estimation in Performance-Based Logistics Systems. *Information* **2025**, *16*, 1031. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.