electronics

MDPI

Article

Active Visual Perception Enhancement Method Based on Deep Reinforcement Learning

Zhonglin Yang [1,2], Hao Fang [1], Huanyu Liu [1,*], Junbao Li [1], Yutong Jiang [2] and Mengqi Zhu [2]

[1] School of Cyberspace Security, Harbin Institute of Technology, Harbin 150000, China; 22b903105@stu.hit.edu.cn (Z.Y.); 22s103145@stu.hit.edu.cn (H.F.); lijunbao@hit.edu.cn (J.L.)
[2] Information and Control Technology Department, China North Vehicle Research Institute, Beijing 100072, China; jiangyutong@bit.edu.cn (Y.J.); 3220235095@bit.edu.cn (M.Z.)
* Correspondence: liuhuanyu@hit.edu.cn

Abstract: Traditional object detection methods using static cameras are constrained by their limited perspectives, hampering the effective detection of low-confidence targets. To address this challenge, this study introduces a deep reinforcement learning-based visual perception enhancement technique. This approach leverages pan–tilt–zoom (PTZ) cameras to achieve active vision, enabling them to autonomously make decisions and actions tailored to the current scene and object detection outcomes. This optimization enhances both the object detection process and information acquisition, significantly boosting the intelligent perception capabilities of PTZ cameras. Experimental findings demonstrate the robust generalization capabilities of this method across various object detection algorithms, resulting in an average confidence level improvement of 23.80%.

Keywords: deep reinforcement learning; object detection; active vision; PTZ camera

Citation: Yang, Z.; Fang, H.; Liu, H.; Li, J.; Jiang, Y.; Zhu, M. Active Visual Perception Enhancement Method Based on Deep Reinforcement Learning. *Electronics* **2024**, *13*, 1654. https://doi.org/10.3390/electronics13091654

Academic Editor: George A. Tsihrintzis

Received: 12 March 2024
Revised: 12 April 2024
Accepted: 22 April 2024
Published: 25 April 2024

1. Introduction

In recent years, remarkable advancements have been achieved in the realm of object detection algorithms, which have emerged as a crucial technology in video surveillance systems, especially for security monitoring [1]. These algorithms play a pivotal role in real-time monitoring and event response. Nevertheless, when confronted with distant, small-sized, and low-resolution targets, conventional object detection algorithms may exhibit reduced performance [2]. This can lead to significantly lower detection confidence levels, which are imperative to address in security monitoring scenarios. The limitations of traditional static camera-based object detection are apparent due to the fixed perspective, making it challenging to effectively monitor targets with low confidence on a large scale and for extended durations [3], which hinders subsequent analysis. Hence, the exploration of active vision techniques using pan–tilt–zoom (PTZ) cameras, with their highly adaptable pan, tilt, and zoom capabilities, is imperative. Active vision mimics the human eye's functionality, allowing individuals to make judgments based on external situations and adjust their perspective as the object of interest moves [4]. To this end, we propose a reinforcement learning-based visual perception enhancement approach. This method leverages PTZ cameras to implement active vision, enabling them to autonomously make decisions and actions based on the current scene and object detection results. It optimizes the process of object detection and information acquisition, thereby enhancing the intelligent perception capabilities of PTZ cameras.

In this article, a novel approach is introduced that integrates reinforcement learning into PTZ camera control, conferring intelligent decision-making capabilities on these cameras when encountering targets with low detection confidence. By harnessing reinforcement learning, PTZ cameras gain autonomy to take appropriate actions, such as adjusting their field of view and zooming in on ambiguous targets to acquire a clearer perspective. Furthermore, this article presents a simulation environment tailored for reinforcement

learning with PTZ cameras and conducts a comprehensive set of experiments to assess the effectiveness of the proposed method in enhancing the intelligence of these cameras.

Specifically, the primary contributions of this article are outlined as follows:

1. We successfully applied reinforcement learning to PTZ camera control, designing and implementing a comprehensive reinforcement learning framework. This framework enables PTZ cameras to autonomously adopt control strategies through interactive learning with their environment, significantly enhancing their intelligent perception capabilities.
2. We introduce the Comprehensive Object Perception Reward Function (COMPRF), a novel approach that significantly improves the intelligence of PTZ cameras in terms of object detection performance.
3. A simulation environment, based on the Unity3D engine, is proposed for simulating PTZ cameras. This simulation environment is fully compatible with the OpenAI Gym [5] interface, accurately simulating camera control tasks in real-world scenarios. It seamlessly receives action commands and returns images along with the object detection results observed by the camera.

2. Related Work

In recent years, reinforcement learning has emerged as one of the three primary machine learning technologies [6] alongside supervised and unsupervised learning, thanks to its exceptional exploratory and self-learning capabilities. Its evolution has encompassed traditional reinforcement learning algorithms like Q-learning [7] to more advanced deep reinforcement learning algorithms, such as DQN [8], and recent innovations like proximal policy optimization (PPO) [9] and asynchronous advantage actor-critic (A3C) [10]. Reinforcement learning has proven successful in addressing numerous robotic and control tasks, eliminating the need for tedious traditional control techniques. Current research has also explored its application in camera control. Sandha et al. [11] introduced an end-to-end deep reinforcement learning approach that directly utilizes raw input images to govern PTZ camera parameters. However, this method suffers from high coupling, necessitating retraining for novel targets. Nikolaos et al. [12] proposed leveraging reinforcement learning to automate camera control, enhancing movie shooting outcomes. Nevertheless, the simulation environment utilized in their work remains relatively straightforward, exhibiting a notable discrepancy from real-world conditions.

In recent years, object detection algorithms have made remarkable strides in the realm of computer vision, with notable works such as YOLOX [13], YOLOF [14], YOLOv8 [15], DETR [16], SiamEXTR [17], and Dynamic R-CNN [18] demonstrating superior performance. Presently, there are also efforts to integrate object detection algorithms with reinforcement learning techniques, aiming to achieve active camera control. Jin et al. [19] employed object detection models to identify object information, generating control values for multiple cameras, thereby facilitating precise monitoring of objects of interest or abnormal behaviors. Kim et al. [20] leveraged object position and size information from video analysis systems to automatically control PTZ cameras, enhancing the accuracy of identifying abnormal behavior in video surveillance systems. Fahim et al. [21] utilized an object detection model to detect targets in surveillance video frames, transforming them into a reinforcement learning state. By adaptively adjusting the camera's position and scaling level, they achieved tracking of existing targets and searching for new ones. Hao et al. [22] delved into enhancing underwater images to improve human observation of underwater scenes. A reinforcement learning-based human visual perception-driven image enhancement paradigm for underwater scenes was proposed. However, the aforementioned works primarily focus on the domain of object tracking and image enhancement, whereas this article's emphasis lies in enhancing the confidence performance of object detection through active camera control.

3. Methods

3.1. Reinforcement Learning

In reinforcement learning, Markov decision processes $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ are widely employed to model the interaction between agents and the environment. The state space is represented by $\mathcal{S}$ and the action space by $\mathcal{A}$. At each discrete time step t, the agent takes action $a_t \in \mathcal{A}$ based on the current state $s_t \in \mathcal{S}$. Subsequently, it transitions to the next state $s_{t+1} \in \mathcal{S}$ guided by the state transition matrix. As a result of this action, the agent receives rewards r_t determined by the reward function $\mathcal{R}$.

The objective of an intelligent agent is to optimize returns, which are defined as follows:

$$R_t = \sum_{n=0}^{\infty} \gamma^n r_{t+n}$$

where t denotes the time step, and $\gamma \in [0, 1)$ represents the attenuation factor. The mapping from each state $\mathbf{s}$ to an action $\mathbf{a}$ is encoded by a policy function $\pi : \mathbf{a} \sim \pi(\cdot \mid \mathbf{s})$. Each strategy π is associated with a corresponding action value function:

$$Q^{\pi}(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\pi}[R_t \mid s_t = \mathbf{s}, a_t = \mathbf{a}]$$

This represents an anticipation of potential returns upon executing an action a_t in a given state s_t. The action value function of a strategy π can also be determined by utilizing the Bellman expectation equation:

$$Q^{\pi}(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\pi}[r_{t+1} + \gamma Q^{\pi}(s_{t+1}, a_{t+1}) \mid s_t = \mathbf{s}, a_t = \mathbf{a}]$$

Traditional reinforcement learning algorithms rely on tabular representations of action value functions, which can be highly limiting in scenarios with vast state and action spaces. These limitations manifest in the immense size of the tables required, leading to challenges in storage and updating. To address these challenges, deep neural networks have emerged as a promising approach to mapping environmental states to action values. Among these advancements, the Deep Q-Network (DQN) algorithm, a deep learning-based extension of Q-learning, has ushered in a new era in deep reinforcement learning. At the core of the DQN algorithm lies the estimation of value functions through a deep neural network. This network takes states as inputs and produces the values for each potential action. Typically, the input layer represents the state, while the output layer comprises nodes corresponding to the number of actions. Through backpropagation algorithms, the network updates its parameters to approximate the true action values.

To train the network effectively, the DQN algorithm employs a technique known as experience replay. This approach involves storing the agent's interaction experiences with the environment, encompassing states, actions, rewards, and subsequent states. During the training process, the DQN algorithm randomly selects a batch of these experiences from the replay memory and utilizes them to update the network parameters. This random sampling technique enhances training stability and improves sample utilization efficiency. Furthermore, the DQN algorithm incorporates a target network in the parameter update process. This target network shares the same structure as the main network but does not undergo real-time parameter updates during training. Instead, its parameters are periodically synchronized with the main network. By employing the target network, the DQN algorithm ensures smoother and more stable training, thereby minimizing instability during the learning process.

The DQN algorithm has exhibited outstanding performance across numerous reinforcement learning tasks, making it a valuable tool for addressing high-dimensional and intricate problems. Building on the DQN algorithm, the present work aims to further delve into and enhance reinforcement learning methods for PTZ cameras, promising even more robust and efficient solutions.

3.2. Network Structure

Reinforcement learning agents leverage neural networks, denoted by $f_W(\cdot)$, parameterized by W to guide their decision-making. These agents receive input vectors s $\in R^3$,

reflecting the current environmental state, and promptly predict the values of potential actions Q. Specifically, they output an m-dimensional vector $y = f_W(s) \in R^m$, where m represents the total number of actions available to the agent. The intricacies of action space and state space will be thoroughly unpacked in the subsequent sections.

The network structure depicted in Figure 1 is specifically designed for estimating action values Q. Initially, a video frame image is processed through an object detection model, extracting crucial state information about the monitored object. This information serves as the input for the network. Throughout all fully connected layers, the ReLU activation function is employed. Notably, the final layer remains unactivated, as it is tasked with directly predicting the values of potential actions Q.

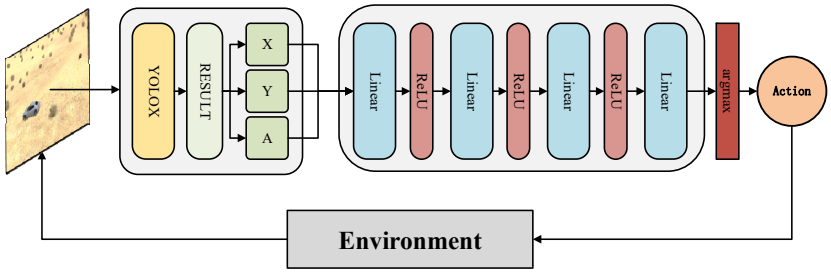

Figure 1. Network structure.

3.3. Intelligent Perception Enhancement Method Based on Deep Reinforcement Learning

The overall framework diagram of the method proposed in this article is presented in Figure 2. Initially, the intelligent agent samples actions from the action space. The selected action is then transmitted to the simulation environment, where it is executed accordingly. Upon completion of the action, the current video frame is captured and fed into the target detection model for processing. This process extracts the coordinates and size information of the monitored object. Subsequently, the detection results are further processed to derive the current state information. The agent then stores this state in an experience replay pool for sampling, learning, and updating network parameters. A key aspect of this design is the decoupling achieved through the introduction of an object detection model. This allows for direct substitution of the object detection model to adapt to different application scenarios or monitoring targets, eliminating the need for retraining. Additionally, this architecture facilitates the migration of models to hardware devices.

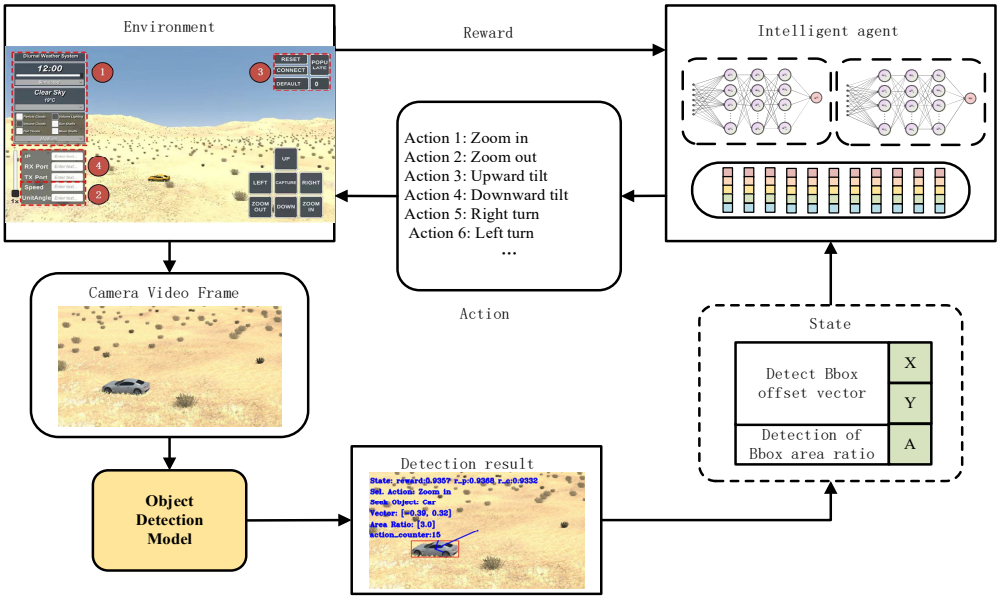

Figure 2. Overall architecture.

3.3.1. Action Space

A PTZ camera is a surveillance device that offers directional focal length adjustment capabilities. By adjusting its direction, angle, and focal length, it can achieve a more flexible field of view. Consequently, the action space for this camera is defined as a discrete one, encompassing various actions that can be performed.

$$\mathcal{A} = [a_{\text{Right}}, a_{\text{Left}}, a_{\text{Up}}, a_{\text{Down}}, a_{\text{ZoomIn}}, a_{\text{ZoomOut}}, a_{\text{Stay}}]$$

Within the context of this study, each distinct action represents the execution of a corresponding action command for a duration of T_{action} seconds. For instance, in our example, when $T_{\text{action}} = 2$, a_{Right} signifies that the PTZ camera will execute a right turn command for a period of 2 s.

However, executing the same action for the same duration at different camera focal lengths can pose challenges. Specifically, when the focal length is increased, a single camera action can result in a significant shift in the field of view, potentially leading to the loss of the target or motion oscillation. To mitigate this issue and because the effectiveness of the same discrete action varies at different focal lengths, the execution time T_{action} of each discrete action is determined as a function that takes into account the camera's zoom factor Z_{ratio}:

$$T_{\text{action}}^{(\text{reduction})} = \frac{1}{Z_{\text{ratio}}}$$

When the focal length is large, it diminishes the amplitude of the camera's movement, whereas a smaller focal length results in a relatively larger amplitude, thereby significantly mitigating the issue of target loss resulting from the camera's actions.

3.3.2. State Space

In reinforcement learning, the agent determines its next action by relying on the state information received from the environment. The state space is defined as follows:

$$\mathcal{S} = [s_x, s_y, s_a]$$

Among them, $[s_x, s_y]$ represents the offset vector between the center of the target detection bounding box and the center of the camera's field of view. This target detection bounding box corresponds to the monitoring target with the highest confidence level. This offset vector is designated as follows:

$$s_x = \frac{\text{Target}_x - \text{Observation}_{\text{x}}}{\text{Observation}_{\text{width}}/2}$$

$$s_y = \frac{\text{Target}_y - \text{Observation}_{\text{y}}}{\text{Observation}_{\text{height}}/2}$$

Target_x, Target_y represents the coordinate position of the center point of the bounding box (Bbox) of the monitoring target with the highest confidence within the current entire field of view. $\text{Observation}_{\text{x}}$, $\text{Observation}_{\text{y}}$ denotes the coordinate position of the center point of the current field of view. Additionally, $\text{Observation}_{\text{width}}$ and $\text{Observation}_{\text{height}}$ stand for the width and height of the current field of view, serving to normalize the offset vector.

In the state space, s_a signifies the ratio of the area of the target detection bounding box (Bbox) of the monitoring target with the highest confidence within the field of view to the overall field of view of the camera, multiplied by a specific coefficient. This can be expressed as follows:

$$s_a = \frac{Area_{bbox}}{Area_{\text{observation}}} \times \text{G}$$

Among them, $Area_{bbox}$ represents the area occupied by the monitoring target's bounding box (Bbox) with the highest confidence in the current scene, while $Area_{\text{observation}}$ denotes

the field of view of the current camera. Furthermore, G serves as a multiplication coefficient, whose purpose is to prevent the area ratio value from being excessively small and thus insignificant as a feature within the state space.

3.3.3. Reward Function

This article introduces a comprehensive target perception reward function, COMPRF, that incorporates both target detection confidence information and target position size data. This innovative reward function aims to encourage the intelligent agent to prioritize enhancing the accuracy of object detection while simultaneously adjusting the PTZ camera's direction, angle, and focal length to center the monitoring target within the camera's field of view at an optimal size. By doing so, it enhances the agent's decision-making stability and precision. The reward function is formally defined as follows:

$$r_t = w_1 \times r_{\mathrm{pos}_t} + w_2 \times r_{\mathrm{con_t}}$$

In the formula, w_1 and w_2 are the reward weights, here set $w_1 = 0.7$. $w_2 = 0.3$ r_{pos_t} and r_{con_t} are defined in formulas respectively.

$$r_{\mathrm{pos}_t} = 1 - \frac{1}{4}\left({s_x}^2 + {s_y}^2 + 2 \times A_{\mathrm{e}}^2\right)$$

$$r_{\mathrm{con}t} = \max_{\mathrm{i} \in O}(\mathrm{Confidence_i})$$

Among them, O is the set of all monitoring targets in the current field of view, and $\mathrm{Confidence_i}$ is the confidence value of the corresponding target. A_{e} is defined in formula.

$$A_{\mathrm{e}} = \begin{cases} \frac{a - a_{\mathrm{e}}}{a_{\mathrm{e}}}, \ if \ a < a_{\mathrm{e}} \\ \frac{a - a_{\mathrm{e}}}{1 - a_{\mathrm{e}}}, if \ a \geq a_{\mathrm{e}} \end{cases}$$

Among them, $a = A/G$, which is the ratio of the Bbox area of the monitoring target with the highest confidence to the overall field of view area of the camera. This ratio, denoted as a_{e}, is crucial in ensuring accurate target detection. Set to 0.03 in this context, representing the expected area ratio, which strikes a balance between maintaining high detection confidence and obtaining a broader field of view.

The comprehensive consideration of both the confidence level of object detection and the location information of the target is pivotal. The confidence level serves as a direct indicator of the algorithm's trustworthiness in its detection results. By rewarding higher confidence outcomes, we steer the camera toward actions that enhance this confidence. Concurrently, incorporating target location information into the reward system incentivizes the camera to center the target within its field of view, optimizing information capture. For moving targets, this optimization enhances both detection coverage and visibility.

Furthermore, this article introduces a threshold mechanism to refine the reward system. Specifically, a reward threshold, denoted as r_{thres}, is established. When the single-step reward value falls below this threshold, it is disregarded. This approach prevents the agent from receiving indiscriminate positive feedback, which could hinder its ability to learn correct behaviors. This thresholding mechanism ensures a more focused and effective learning process. This mechanism can be expressed in the following form:

$$r_t^{(\mathrm{clipped})} = \begin{cases} 0, \ \text{if } r_t < r_{\mathrm{thres}} \\ r_t, \ \text{otherwise} \end{cases}$$

Simultaneously, if a target detection confidence value exceeding 0.9 is maintained for 20 consecutive actions, the current game will prematurely conclude. The total number of actions, denoted as s_t, executed prior to termination will be recorded. Additionally, a task completion reward value, designated as r_{complete}, will be appended to the cumulative

reward value of the current game, represented by $R = \sum_{n=1}^{s_t} r_n$. This augmentation yields the final total reward value. Conversely, if the confidence criterion is not met, no additional reward will be awarded. The computation of the final reward is outlined in equation.

$$R_{\text{total}} = R + r_{\text{complete}}$$

Among them,

$$r_{\text{complete}} = \begin{cases} 300 - s_t, \ if \ s_t < 256 \\ 0, \ if \ s_t \geq 256 \end{cases}$$

The purpose of introducing this reward value is to expedite the intelligent agent's ability to optimize target detection confidence performance, guiding it to choose the most efficient execution path possible for completing the task.

3.4. Simulation Environment

This article utilizes the Unity3D engine to create a bespoke simulation environment named CamSim, depicted in Figure 3. This environment accurately simulates the effects of various PTZ camera control commands. The simulation environment boasts the following functionalities:

1. Day, night, and weather adjustments: The simulation environment faithfully replicates changing conditions throughout the day, including shifts in time and random weather patterns. These changes can be modulated by adjusting the simulation environment's time flow rate, which operates at a faster pace compared to real-world conditions. This accelerated pace enables a comprehensive evaluation of reinforcement learning algorithms' performance across diverse temporal and atmospheric scenarios.
2. Camera parameter customization: Users can tailor the rotation angle and speed of simulated PTZ cameras based on their specific hardware capabilities. This customization ensures that the simulation accurately reflects the observation capabilities and response speeds of various PTZ cameras in real-world settings, thereby enhancing the precision of reinforcement learning algorithm evaluation and training.
3. Experimental object and trajectory generation: The simulation environment incorporates seven distinct car object types. Users can select specific car types or opt for random generation to cater to their experimental or training needs. Additionally, the simulation environment provides preset trajectories and speed profiles, allowing users to combine them freely to create diverse scenarios tailored for evaluating and training reinforcement learning algorithms.
4. UDP communication support: The simulation environment boasts a robust UDP communication function that facilitates the use of customized IP addresses and ports. This functionality enables seamless contact with external systems for data transmission, environmental status detection, control operations, and various other tasks.

This article establishes a robust communication mechanism between the agent and the Unity3D simulation environment, enabling intelligent control of PTZ cameras. The Gym environment has been thoroughly rewritten to facilitate interaction with the Unity3D simulation environment through UDP communication. Specifically, the interfaces of the Gym environment, including functions like step and reset, have been revised to incorporate internal statements that send instructions via UDP. The proxy relays control instructions to the Unity3D simulation environment by invoking these Gym interface functions.

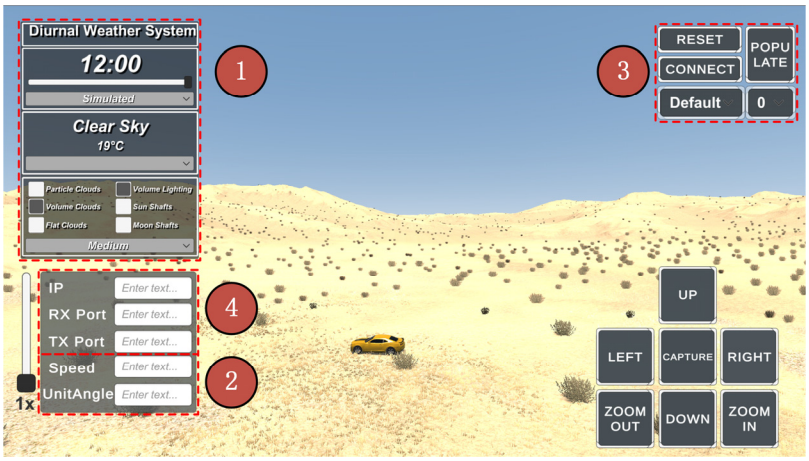

Figure 3. Simulation environment CamSim.

Within the communication strategy between the agent and the simulation environment, a bidirectional UDP-based communication mechanism has been implemented, as depicted in Figure 4. Here are the specific steps involved in the communication process:

1. The intelligent agent transmits action commands to the simulation environment via the Gym interface. Based on its current strategy, the agent selects appropriate actions and sends these commands to the simulation environment using UDP. These commands specify the control actions for the PTZ camera. Once the instruction is dispatched, the agent transitions to a waiting state, suspending further actions until it receives a completion signal from the simulation environment.
2. Execution of actions within the simulation environment: Upon receiving the action instructions from the agent, the simulation environment interprets them and directs the PTZ camera in the scene to perform the corresponding actions.
3. Completion signal from the simulated environment: Once the PTZ camera has executed the designated action, the simulated environment transmits a completion signal to the agent, indicating that it can proceed with subsequent operations. Concurrently, the simulation environment captures the current video frame image and forwards it to the agent.
4. Agents receive observation data: Upon receiving the action signal, the agent exits the waiting state and proceeds with subsequent operations. Simultaneously, the video frame images are processed using the YOLOX model interface, enabling further elaboration to extract state space information and reward values.

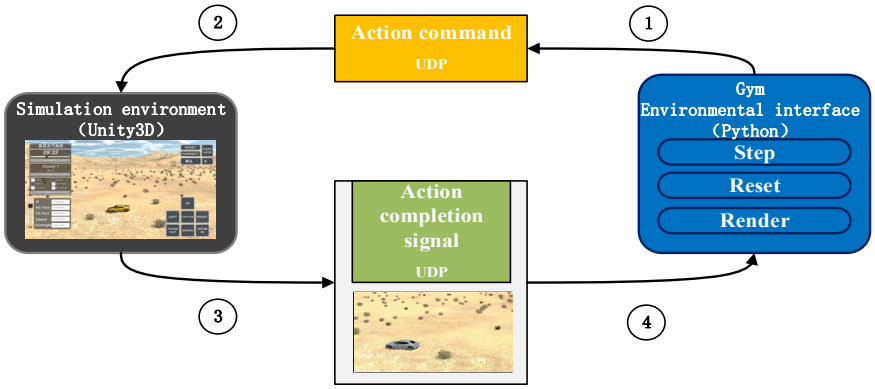

Figure 4. Bidirectional communication mechanism.

This communication mechanism facilitates the control of the PTZ camera within a simulation environment, allowing feedback about the action results to be returned to the intelligent agent to reinforce learning training and evaluation.

4. Experimental Results and Analysis

To assess the effectiveness and robustness of the method introduced in this article, a comprehensive series of experiments were designed and conducted. This section delves into the experimental setup, including confidence experiments, three distinct types of scene generalization experiments, and ablation experiments, offering a detailed overview of the procedures and findings.

4.1. Experimental Setup

The intelligent agent model underwent a total of 158,720 steps of training on the workstation 12th Gen Intel(R) Core (TM) i7-12700K and NVIDIA GeForce RTX 3060 (Lenovo Group, Beijing, China). Throughout the training process, the model network was optimized using the Adam optimizer, with a learning rate of 0.001. The experience replay pool maintained 20,000 samples, from which 256 samples were randomly selected prior to each optimization iteration. The attenuation factor γ was set to 0.995, and the target network update frequency was configured as 100. Additionally, an Epsilon-greedy exploration strategy was employed where $\varepsilon = 0.1$.

The simulation environment was capable of executing a maximum of 256 actions per episode, and an episode would terminate prematurely if a reward value exceeding 0.9 was achieved for 20 consecutive actions.

4.2. Confidence Experiment

To assess the impact of enhancing confidence, we selected the widely used object detection algorithms, namely YOLOX, YOLOv8, SSD, and Cascade RCNN, as the detection methods for the experiment. These object detection models were built utilizing the MMdetection framework [23] and underwent training on a self-constructed target dataset.

A total of 150 experimental scenarios were created within the simulation environment, with randomized target positions in each scenario. The conclusive experimental outcomes are summarized in Table 1.

Table 1. Experimental results of confidence of object detection model.

Method	Confidence	Method + Ours	Confidence
YOLOX	0.723 ± 0.121	YOLOX + Ours	0.940 ± 0.013
YOLOv8	0.833 ± 0.161	YOLOv8 + Ours	0.985 ± 0.019
SSD	0.718 ± 0.190	SSD + Ours	0.996 ± 0.017
Cascade RCNN	0.652 ± 0.138	Cascade RCNN + Ours	0.957 ± 0.028

The results indicate that our proposed method effectively enhances the confidence level of low-confidence targets compared to the original object detection algorithm. Experimental findings demonstrate the robust generalization capabilities of this method across various object detection algorithms, resulting in an average confidence level improvement of 23.80%.

To further illustrate this, we selected ten sets of YOLOX data for display (Figure 5).

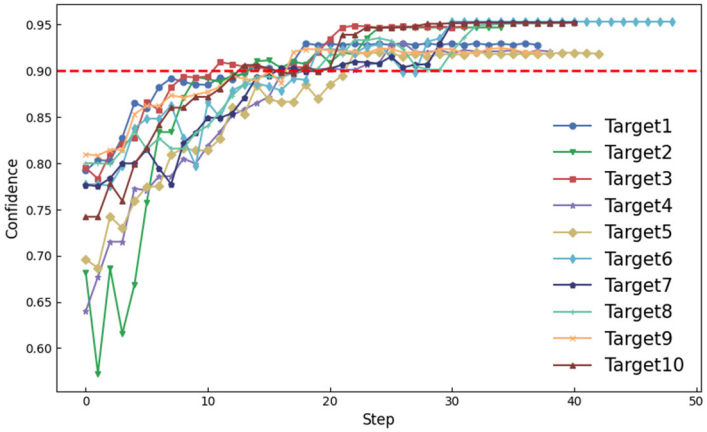

Figure 5. Confidence variation curve of YOLOX object detection algorithm.

4.3. Scene Generalization Experiment

Additionally, we designed two scenarios to evaluate the robustness of our method. The experimental environment settings for these scenarios are depicted in Figure 6.

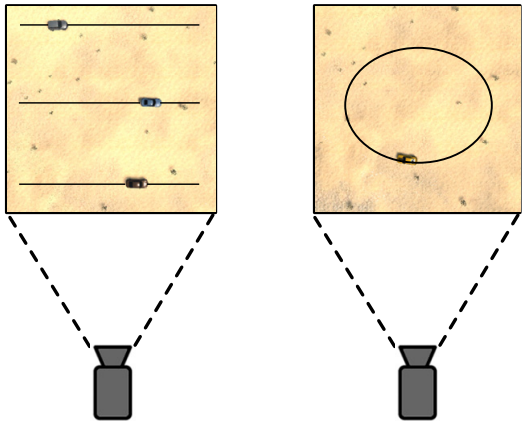

Figure 6. Scenario generalization experimental environment setup.

1. Distance scene: involves three driving routes for the vehicle, each corresponding to a different distance level: far, middle, and near, while assuming a zero vehicle target speed.
2. Sports scene: involves a designated car driving route where the route is divided into seven distinct speed levels.

These scenarios were crafted to thoroughly test the adaptability and performance of our proposed method under varying conditions.

The results of the two experimental scenarios are presented in Table 2, demonstrating that the proposed method achieves reward values approaching the maximum theoretical reward across three distinct distance routes and the initial five speed grades. While the reward value begins to decrease from the sixth speed grade, it remains at a high level, indicating the strong generalization ability of the method across various scenarios. This underscores the high feasibility of the approach in practical applications.

Table 2. Experimental results of scenario generalization.

Scene	Scene Subdivision	Bonus Value
Distance scene	Far	291.056 ± 1.516
	Medium	291.552 ± 1.425
	Near	293.531 ± 1.268
Speed scene	Speed rating 1	289.187 ± 3.371
	Speed rating 2	290.744 ± 3.291
	Speed rating 3	286.934 ± 4.344
	Speed rating 4	286.928 ± 4.810
	Speed rating 5	282.227 ± 9.795
	Speed rating 6	260.014 ± 27.862
	Speed rating 7	242.048 ± 28.010

4.4. False Target Experiment

At the far end of the scene, a deceptive, false target is positioned. The experimental environment settings are depicted in Figure 7.

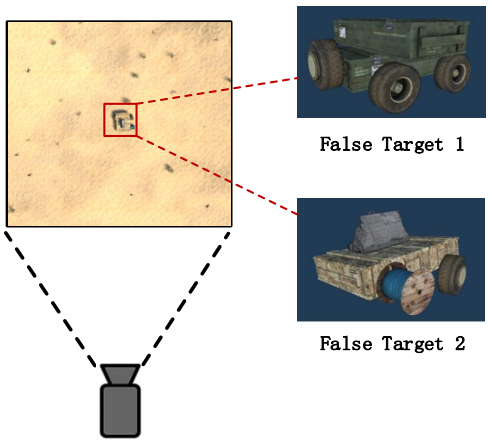

Figure 7. Setup of false target experiment environment.

This study defines and experiments with two distinct types of false targets. The experimental outcomes for both types of false targets are summarized in Table 3.

Table 3. Experimental results of false target scenario.

False Target	The False Target, Whether Can Judge
False target 1	Yes
False target 2	Yes

Initially, when the focal length is short, the PTZ camera mistakenly identifies both types of false targets as legitimate monitoring objects. However, as the focal length increases, the camera gains a clearer view of the targets' details and characteristics. Gradually, it becomes apparent that the current target being tracked is a false one, as illustrated in Figure 8. This demonstrates the method's ability to distinguish between genuine monitoring objects and deceptive false targets. Complete example is shown in Figure A1.

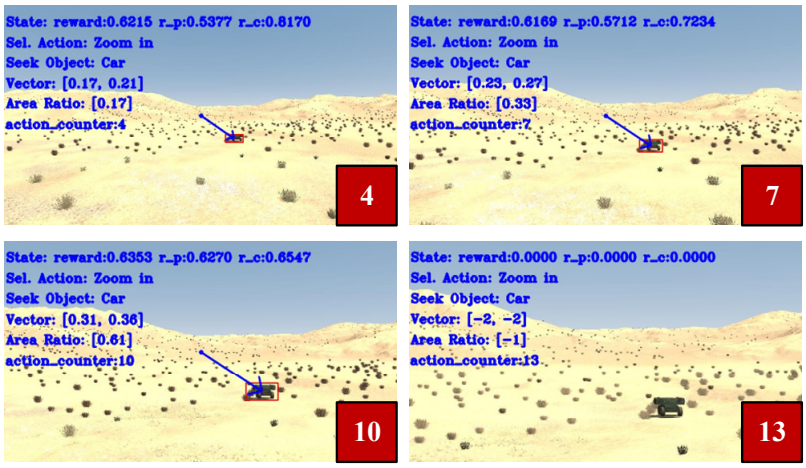

Figure 8. Examples of a fake target scenario experiment.

4.5. Ablation Experiment

To validate the effectiveness of our method, we conducted a series of ablation experiments. These experiments involved removing various components from the reward system: remove the reward threshold r_{thres} (Without_rt), remove r_{pos_t} from the reward (Without_rp), and remove r_{con_t} from the reward (Without_rc). We then evaluated the modified models in diverse task scenarios, comparing their performance against the original model. Notably, to ensure consistency, all modified trained models underwent the same number of training iterations as the original model.

Due to alterations in the reward system, we adopted a new metric to assess performance: the success rate. This metric measures the percentage of total moves that successfully complete a task, where the confidence level of the target exceeds 90% per game and remains above this threshold for 20 consecutive moves. We conducted ten experiments across various scenarios, and the results are summarized in Table 4.

Table 4. Results of ablation experiment.

Scene	Rt	rp	rc	The Success Rate
Near	√	√	√	100%
		√	√	100%
	√		√	10%
	√	√		80%
Medium	√	√	√	100%
		√	√	60%
	√		√	0%
	√	√		70%
Far	√	√	√	100%
		√	√	70%
	√		√	0%
	√	√		80%
move	√	√	√	100%
		√	√	0%
	√		√	30%
	√	√		30%
Target lost	√	√	√	100%
		√	√	0%
	√		√	0%
	√	√		60%

Drawing from the aforementioned results, several key conclusions emerge. Firstly, the Without_rp model, devoid of the reward component pertaining to the monitored object's location and size, exhibited significantly inferior performance across diverse scenarios. Its success rate hovered around 30% and, in most instances, plummeted to 0%. This underscores the critical role of location and size information in providing a unified and definitive basis for the agent's decision-making. Consequently, the model's performance in the three scenarios proved far less stable than the original. The reward aspect associated with the monitored object's location and size encourages the camera to prioritize centering the target within the field of view at an appropriate scale, thereby minimizing the risk of target loss when tracking moving objects. This accounts for the Without_rp model's ineffectiveness in handling moving targets.

Furthermore, the Without_rc model, deprived of the reward component linked to confidence, also demonstrated reduced effectiveness. The absence of this reward prevents the agent from learning actions that enhance the test model's confidence, often leading to choices that fail to meet the confidence threshold required for successful task completion. These findings further highlight the significance of both reward function components, which play pivotal roles in guiding the agent's decision-making.

Secondly, the Without_rt model, stripped of the reward threshold, exhibited inferior performance compared to the original in most scenarios. This indicates that the threshold mechanism serves a crucial screening function in managing reward allocation, disregarding unduly low rewards. Consequently, the agent is incentivized to prioritize strategies offering higher reward values. This underscores the importance of rewarding agents solely for actions that closely align with the desired outcome.

5. Conclusions

In this study, we have successfully implemented the active vision functionality of a PTZ camera through the integration of depth reinforcement learning techniques. This allows the camera to autonomously make decisions based on real-time scene analysis and target detection results, significantly enhancing its perception of targets with low confidence levels. A key innovation is the introduction of an integrated target perception reward function that incorporates both target detection confidence and positional information. This approach has yielded remarkable improvements in target detection confidence by an average improvement of 23.80%. To facilitate the training and evaluation of our proposed method, we have developed a simulation environment, CamSim, using Unity3D, which offers a robust platform for further research and application development.

We believe that these advancements hold significant potential for enhancing camera technology and its practical applications, providing vital support for areas such as intelligent monitoring and autonomous driving. However, it is worth noting that our study still faces some challenges. One such limitation lies in the design of the state space, where there is room for incorporating additional feature information to enrich the decision-making process. Furthermore, addressing the oscillation phenomenon observed at high speeds remains a crucial area for future research because it holds the key to enhancing the method's stability and overall performance.

Author Contributions: Writing—original draft preparation, Z.Y.; writing—review and editing, H.F.; supervision, H.L. and M.Z.; project administration, J.L.; funding acquisition, J.L. and Y.J. All authors have read and agreed to the published version of the manuscript.

Funding: Supported by the National Natural Science Foundation of China (Grant No. 62271166) and Interdisciplinary Research Foundation of HIT, No. IR2021104.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: Author Zhonglin Yang, Yutong Jiang and Mengqi Zhu was employed by the company China North Vehicle Research Institute. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A

Figure A1. Complete example.

References

1. Tsakanikas, V.; Dagiuklas, T. Video Surveillance Systems-Current Status and Future Trends. *Comput. Electr. Eng.* **2018**, *70*, 736–753. [CrossRef]
2. Zhang, J.; Meng, Y.; Chen, Z. A Small Target Detection Method Based on Deep Learning with Considerate Feature and Effectively Expanded Sample Size. *IEEE Access* **2021**, *9*, 96559–96572. [CrossRef]
3. Wang, S.; Tian, Y.; Xu, Y. Automatic Control of PTZ Camera Based on Object Detection and Scene Partition. In Proceedings of the 2015 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Ningbo, China, 19–22 September 2015; pp. 1–6.
4. Wang, X.; Van De Weem, J.; Jonker, P. An Advanced Active Vision System Imitating Human Eye Movements. In Proceedings of the 2013 16th International Conference on Advanced Robotics (ICAR), Montevideo, Uruguay, 25–29 November 2013; pp. 1–6.
5. Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; Zaremba, W. OpenAI Gym 2016. *arXiv* **2016**, arXiv:1606.01540. [CrossRef]
6. Xu, Z.; Wang, S.; Xu, G.; Liu, Y.; Yu, M.; Zhang, H.; Lukasiewicz, T.; Gu, J. Automatic Data Augmentation for Medical Image Segmentation Using Adaptive Sequence-Length Based Deep Reinforcement Learning. *Comput. Biol. Med.* **2024**, *169*, 107877. [CrossRef]

7. López Diez, P.; Sundgaard, J.V.; Margeta, J.; Diab, K.; Patou, F.; Paulsen, R.R. Deep Reinforcement Learning and Convolutional Autoencoders for Anomaly Detection of Congenital Inner Ear Malformations in Clinical CT Images. *Comput. Med. Imaging Graph.* **2024**, *113*, 102343. [CrossRef] [PubMed]
8. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning. *arXiv* **2013**, arXiv:1312.5602.
9. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347.
10. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.P.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous Methods for Deep Reinforcement Learning. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016.
11. Sandha, S.S.; Balaji, B.; Garcia, L.; Srivastava, M. Eagle: End-to-End Deep Reinforcement Learning Based Autonomous Control of PTZ Cameras. In Proceedings of the 8th ACM/IEEE Conference on Internet of Things Design and Implementation, San Antonio, TX, USA, 9 May 2023; pp. 144–157.
12. Passalis, N.; Tefas, A. Deep Reinforcement Learning for Controlling Frontal Person Close-up Shooting. *Neurocomputing* **2019**, *335*, 37–47. [CrossRef]
13. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
14. Chen, Q.; Wang, Y.; Yang, T.; Zhang, X.; Cheng, J.; Sun, J. You Only Look One-Level Feature. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13034–13043.
15. Reis, D.; Kupec, J.; Hong, J.; Daoudi, A. Real-Time Flying Object Detection with YOLOv8. *arXiv* **2023**, arXiv:2305.09972.
16. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In *European Conference on Computer Vision*; Springer International Publishing: Cham, Switzerland, 2020.
17. Wang, Y.; Chen, Z.; Sun, M.; Sun, Q. Enhancing Active Disturbance Rejection Design via Deep Reinforcement Learning and Its Application to Autonomous Vehicle. *Expert Syst. Appl.* **2024**, *239*, 122433. [CrossRef]
18. Zhang, H.; Chang, H.; Ma, B.; Wang, N.; Chen, X. Dynamic R-CNN: Towards High Quality Object Detection via Dynamic Training. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020.
19. Kim, D.; Yang, C.M. Reinforcement Learning-Based Multiple Camera Collaboration Control Scheme. In Proceedings of the 2022 Thirteenth International Conference on Ubiquitous and Future Networks (ICUFN), Barcelona, Spain, 5 July 2022; pp. 414–416.
20. Kim, D.; Kim, K.; Park, S. Automatic PTZ Camera Control Based on Deep-Q Network in Video Surveillance System. In Proceedings of the 2019 International Conference on Electronics, Information, and Communication (ICEIC), Auckland, New Zealand, 22–25 January 2019; pp. 1–3.
21. Fahim, A.; Papalexakis, E.; Krishnamurthy, S.V.; Roy Chowdhury, A.K.; Kaplan, L.; Abdelzaher, T. AcTrak: Controlling a Steerable Surveillance Camera Using Reinforcement Learning. *ACM Trans. Cyber-Phys. Syst.* **2023**, *7*, 1–27. [CrossRef]
22. Wang, H.; Sun, S.; Chang, L.; Li, H.; Zhang, W.; Frery, A.C.; Ren, P. INSPIRATION: A Reinforcement Learning-Based Human Visual Perception-Driven Image Enhancement Paradigm for Underwater Scenes. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108411. [CrossRef]
23. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv* **2019**, arXiv:1906.07155.