Journal of Industrial Engineering and Applied Science
Journal Home: http://jieas.suaspress.org/ | CODEN: JIEAAE
Vol. 2, No. 6, 2024 | ISSN 3005-6071 (Print) | ISSN 3005-608X (Online)

Advances in Deep Reinforcement Learning for Computer Vision Applications

LI, Zhengyang [1*]

[1] DigiPen Institute of Technology, USA

**** LI, Zhengyang is the corresponding author, E-mail: levey.lee@gmail.com***

Abstract: Deep Reinforcement Learning (DRL) has become very popular for computer vision (CV), solving mostly visually complex environments with decision making and dynamic adaption to different situations. This article provides an introduction to the basic concepts of deep reinforcement learning with a special focus on their applications in computer vision tasks, including challenging problems and emerging solutions. It reviews various DRL algorithms such as Q-learning, policy gradient methods, and Actor-Critic models, explaining much of the modifications that are done to make it work on high-dimensional visual problems. Different applications of DRL in major CV applications like object detection, image segmentation, target tracking and image generation are reviewed to demonstrate the power as well as limitations of DRL in practice. More importantly, the novel paradigms such as hierarchical policy learning, adaptive reward design multi-task reinforcement learning and domain adaptation can be viewed as promising premises to improve model efficiency and generalizability in multiple scenarios. Finally, this paper mentions a few of the existing challenges like computational power cost and sample efficiency; as well as future paths for enhancing that can widen devout reinforcement learning in computer vision. Through this comprehensive overview, we aim to shed light on the promising synergies between DRL and CV, while identifying key areas for future research and application.

Keywords: Deep Reinforcement Learning, Computer Vision, Object Detection, Q-learning.

Disciplines: Computer Science. **Subjects:** Computer Vision.

DOI: https://doi.org/10.70393/6a69656173.323234 **ARK:** https://n2t.net/ark:/40704/JIEAS.v2n6a03

1 INTRODUCTION

Reinforcement Learning (RL) is a type of machine learning where an agent learns the best possible actions to take by interacting with the environment in order to maximize cumulative rewards over time. DRL, which combines deep learning and reinforcement learning, enables an agent to solve high-dimensional problems using the power of deep neural networks[1]. In the field of computer vision, many complex tasks—such as object tracking, autonomous driving, and robotic manipulation—require sequential decision-making, adaptability to dynamic conditions, and real-time performance. DRL is uniquely suited to address these challenges by enabling continuous learning and strategy optimization as the environment changes, making it a significant area of research in modern AI applications. This paper aims to provide a comprehensive overview of the foundational concepts, applications, challenges, and innovations of DRL in the CV domain, ultimately guiding future research and development in the field.

2 FOUNDATIONS OF DEEP REINFORCEMENT LEARNING

2.1 CORE FRAMEWORK OF RL

The basic framework of reinforcement learning (RL) can be represented by a Markov Decision Process (MDP), which consists of state (S), action (A), reward (R), and transition probability (P)[2]. At each time step, the agent selects an action $a_t \in A$ in state $s_t \in S$ and determines the probability distribution of action a based on policy $\pi(a \mid s)$. The environment then provides a reward r_{t+1} and moves to the next state s_{t+1}. The goal of reinforcement learning is to learn the optimal policy π^* by maximizing the cumulative reward $R = \sum_{t=0}^{\infty} \gamma^t r_{t+1}$, where γ is the discount factor, controlling the impact of future rewards on current decisions. This core framework is widely applied for learning optimal strategies in dynamic environments.

2.2 Combining Deep Learning with RL

Deep learning, with its capability of extracting complex features through deep neural networks (DNNs)[3], has enabled the development of deep reinforcement learning , enhancing RL's applicability in complex visual tasks[4]. For example, in Deep Q-Networks (DQN), convolutional neural networks (CNNs) process image inputs to extract abstract representations of states, followed by Q-learning for optimizing action selection[5]. The non-linear properties of deep learning enable more complex modeling of the environment, allowing agents to efficiently search for optimal strategies in high-dimensional state spaces. This structure, combining feature extraction with policy optimization, effectively overcomes the traditional RL bottleneck in handling high-dimensional inputs[6].

2.3 Algorithm Overview

In DRL, different algorithms are suited for different types of tasks, including:

(1) Q-learning and Its Deep Variants

Q-learning is a value-based algorithm that guides action selection by learning the value function $Q(s,a)$ for state-action pairs. The Q-learning update formula is:

$$Q(s_t,a_t) = Q(s_t,a_t) + \alpha\left(r_{t+1} + \gamma \max_{a'} Q(s_{t+1},a') - Q(s_t,a_t)\right)$$

where α is the learning rate, and γ is the discount factor. In DQN, Q-learning is combined with CNNs to process high-dimensional image inputs, enabling agents to learn effective strategies for complex visual tasks[7]. Double DQN reduces overestimation by separating action selection from value evaluation, making the model more stable[8].

(2) Policy Gradient Methods

Policy gradient methods directly optimize the policy $\pi(a|s,\theta)$, where θ denotes the policy parameters. The goal is to maximize the expected cumulative reward $J(\theta) = \mathbb{E}_\pi[R]$, with the policy gradient update formula as:

$$\nabla_\theta J(\theta) = \mathbb{E}_\pi[\nabla_\theta \log \pi(a|s,\theta) Q_\pi(s,a)]$$

Policy gradient methods are suitable for continuous action space tasks and provide higher efficiency in handling high-dimensional continuous spaces by directly optimizing the policy.

(3) Actor-Critic Methods

Actor-Critic algorithms combine the strengths of value function methods and policy optimization. The actor updates the policy to select optimal actions, while the critic estimates the value of these actions. The update formula is:

$$\nabla_{\theta_{\text{actor}}} J = \mathbb{E}_\pi[\nabla_{\theta_{\text{actor}}} \log \pi(a|s,\theta_{\text{actor}}) \cdot A_\pi(s,a)]$$

where $A_\pi(s,a) = Q(s,a) - V(s)$ is the advantage function, used to assess the quality of actions. Improved algorithms like Proximal Policy Optimization (PPO) and Soft Actor-Critic (SAC) perform well in enhancing policy stability and are suitable for complex computer vision tasks[9].

Here is an enhanced description of the A3C algorithm along with its formula:

Asynchronous Advantage Actor-Critic

The Asynchronous Advantage Actor-Critic (A3C) algorithm extends traditional actor-critic methods by employing multiple agents to interact with different copies of the environment asynchronously. Each agent updates a shared global model, which accelerates learning and allows for better exploration across large state spaces.

In A3C, each agent learns an advantage function to improve the stability of policy updates, computed as the difference between the estimated state-action value $Q(s,a)$ and the value function $V(s)$. The advantage function $A(s,a)$ for a given state-action pair is:

$$A(s,a) = Q(s,a) - V(s)$$

The actor (policy) and critic (value) are updated based on the agent's collected experience. The policy update formula for the actor is:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(a|s) A(s,a)$$

where α is the learning rate, $\pi_\theta(a|s)$ represents the policy, and $\nabla_\theta \log \pi_\theta(a|s)$ is the policy gradient, adjusted by the advantage function $A(s,a)$.

The critic's update formula minimizes the mean-squared error of the value function:

$$\phi \leftarrow \phi - \beta \nabla_\phi \left(R - V_\phi(s)\right)^2$$

where β is the learning rate for the critic, R represents the observed cumulative reward, and $V_\phi(s)$ is the value function parameterized by ϕ.

A3C is effective in tasks requiring high exploration and can handle large-scale problems in diverse environments, leveraging asynchronous updates for efficient learning.

3 MAJOR APPLICATIONS IN COMPUTER VISION

3.1 Object Detection and Recognition

Object detection and recognition is one of the core tasks in the field of computer vision. Traditional detection methods typically require analyzing every region of an image, whereas RL guides the agent to focus on informative regions, reducing redundant computations and improving detection efficiency[10]. In multi-object detection tasks, the reward mechanism directs the agent to learn to focus on specific regions, enabling effective localization and recognition of multiple objects in complex backgrounds. This method is

Journal of Industrial Engineering and Applied Science
Journal Home: http://jieas.suaspress.org/ | CODEN: JIEAAE
Vol. 2, No. 6, 2024 | ISSN 3005-6071 (Print) | ISSN 3005-608X (Online)

particularly important in autonomous driving and robot navigation, as it allows for quick responses to dynamic environmental changes and improves recognition accuracy.

3.2 Image Segmentation and Semantic Understanding

A second class of vision tasks that need accurate image processing is all about image segmentation and semantic understanding. By adapting their decision strategy during segmentation, RL methods achieve more active and less context-free policy learning on image segmentation with a strong focus on correct object boundary delineation. Specifically for sophisticated scenes, RL can gradually optimize segmentation strategies to further improve segmentation performance. High-precision segmentation can be very beneficial to improve the understanding of a scene, and this technology has been widely used in autonomous driving scenarios and medical image processing.

3.3 Object Tracking

Object tracking refers to the task of identifying and tracking the position of moving objects in video sequences. In complex backgrounds, RL can learn adaptive tracking strategies, maintaining high-precision tracking of the target in dynamic scenes.

3.4 Image Generation and Inpainting

Image generation and Image Inpainting represent another key application of Deep Reinforcement Learning in computer vision. Traditional image generation methods typically rely on Generative Adversarial Networks (GANs) or autoregressive models, while in the DRL framework, agents learn to generate images or complete missing parts of an image based on reward signals. The overall architecture of GAN is shown in Fig. 1, where the dual-network structure, consisting of a generator and a discriminator, forms the core of the GAN model. Particularly in image restoration, RL explores different repair strategies, progressively improving the quality of the restored image by generating content that matches the original style. This approach has significant potential in areas such as cultural heritage restoration and video editing.

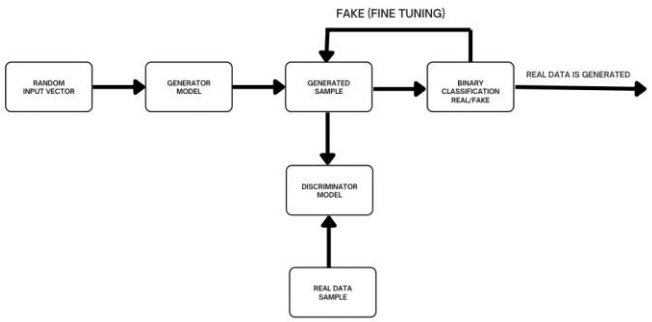

Fig. 1 GAN's overall architecture

4 Innovative Approaches in Deep Reinforcement Learning

4.1 Hierarchical Reinforcement Learning (HRL)

Traditional RL can struggle with complex tasks due to issues like slow convergence. Hierarchical Reinforcement Learning (HRL) addresses this by breaking down complex tasks into sub-tasks through a layered structure, where each level of the hierarchy is responsible for a specific sub-task. For example, in multi-object detection, lower-level strategies handle individual object recognition, while higher-level strategies coordinate the results from multiple objects. HRL enhances task execution efficiency and reduces the complexity of policy search, making it particularly suited for high-dimensional visual tasks.

4.2 Adaptive Reward Design

Reward design plays a critical role in RL algorithm convergence and learning efficiency. Traditional fixed rewards may cause slow learning, especially in environments with sparse rewards. Adaptive reward design introduces dynamic mechanisms that adjust reward signals based on agent performance, accelerating the agent's approach to the optimal strategy. For example, in target tracking tasks, RL models dynamically adjust rewards, providing higher rewards when the agent is near a fast-moving target to maintain tracking accuracy. Adaptive reward design not only accelerates convergence but also significantly improves performance in complex scenarios.

4.3 Multi-task RL

In traditional RL, models are usually trained for a single task. Multi-task RL, however, aims to enable agents to learn to solve multiple tasks, enhancing the model's versatility and adaptability. Through multi-task learning, agents can share features and strategies between tasks, improving learning efficiency for new tasks. For instance, an agent trained on multiple tasks like object detection, image segmentation, and target tracking can quickly adapt and make effective decisions in combined task environments. Multi-task RL shows immense potential in improving the generalization of models.

4.4 Domain Adaptation

Data distributions may be different in other application scenarios, and since traditional RL model needs to be retrained when transferring the same model from one environment to another, it introduces more computation cost. This leads to the development of adaptable reinforcement learning models that can transfer learnings in domains other than where they were trained, which is made possible by Domain Adaptation techniques also promising high performance across a number of domains as the training and

test environments could look very different. For example, domain adaptation techniques enable adaptation of a robot vision system trained in simulation very quickly to changes in the real world. It improves the real-world practicality and robustness of DRL models.

4.5 Summary of Key Algorithms and Performance Comparison

4.5.1 Classic Algorithm Comparison

In DRL, classical algorithms like DQN, PPO, and SAC each have unique characteristics. DQN excels in discrete action spaces, making it suitable for low-dimensional tasks with high sample efficiency; PPO introduces probability limits in the policy optimization process, significantly improving stability, making it ideal for tasks requiring high stability; SAC combines entropy regularization to encourage exploration and achieves stable convergence in continuous action space tasks. The following summarizes the core formulas for these algorithms.

(1) Deep Q-Network (DQN)

DQN is a value-based algorithm suitable for discrete action spaces, with the following update formula

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha \left(r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right)$$

By using convolutional neural networks (CNN) to process high-dimensional image inputs, DQN efficiently learns optimal policies for complex visual tasks[11].

(2) Proximal Policy Optimization (PPO)

PPO is a policy gradient-based method, where the target function during updates is:

$$L_{PPO}(\theta) = E_t \left[\min \left(r_t(\theta) A_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) A_t \right) \right]$$

Here, $r_t(\theta)$ is the ratio of the current policy to the old policy, and ϵ defines the clipping range. PPO improves policy stability by limiting the update size, making it well-suited for tasks requiring high stability in computer vision.

(3) Soft Actor-Critic (SAC)

SAC is an entropy-regularized algorithm that maximizes cumulative rewards while promoting exploration by increasing the policy's entropy, with the following update formula:

$$J(\pi) = \sum_{t=0}^{T} E_\pi \left[r(s_t, a_t) + \alpha H(\pi(\cdot|s_t)) \right]$$

where α is the entropy coefficient, and $H(\pi(\cdot|s_t))$ represents the entropy of the policy. SAC performs well in continuous action spaces, making it ideal for complex visual tasks.

4.5.2 Parameter Optimization in Model Training

Hyperparameters such as learning rate, discount factor, and reward factor significantly affect the final performance of the model during training. For instance, adjusting the learning rate and discount factor in DQN can improve its adaptability to high-dimensional state spaces, while the entropy coefficient in PPO determines the level of exploration. Choosing the appropriate hyperparameter combination is crucial for improving convergence speed and stability.

4.5.3 Algorithm Applicability Analysis

In terms of practicality, certain algorithms work better than others for a task. DQN has good performance in discrete space tasks such as image segmentation and object recognition; PPO is suitable for stable and continuous space tasks, such as target tracking; SAC performs better in highly exploratory environments that are a continuous action space. This difference lays out a theoretical basis for the algorithms to be chosen in computer vision concerning various needs, thereby optimizing algorithm choice.

5 OVERVIEW AND PERFORMANCE COMPARISON OF KEY ALGORITHMS

5.1 Comparison of Classical Algorithms: DQN, PPO, and SAC

Deep Reinforcement Learning has brought significant advancements in addressing complex decision-making tasks in computer vision (CV). Here, we systematically compare three widely used DRL algorithms: Deep Q-Networks (DQN), Proximal Policy Optimization (PPO), and Soft Actor-Critic (SAC), each exhibiting unique characteristics and strengths.

DQN is a value-based algorithm that uses Q-learning with deep neural networks to approximate action-value functions. Formally, DQN aims to maximize the expected cumulative reward $E\left[\sum_{t=0}^{T} \gamma^t r_t\right]$, where r_t is the reward at time step t, and γ is the discount factor. The main objective is to minimize the temporal difference (TD) error:

$$L(\theta) = E_{(s,a,r,s') \sim \mathcal{D}} \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right)^2 \right]$$

where θ are the parameters of the Q-network, θ^- represents the parameters of a target network, and $\mathcal{D}$ is the experience replay buffer. DQN has demonstrated success in various tasks but struggles in high-dimensional continuous action spaces, which are common in CV.

PPO is a policy-gradient-based algorithm that optimizes the policy by maximizing a clipped objective function. PPO effectively balances exploration and exploitation by constraining policy updates, reducing the risk of performance collapse. The objective function is given as:

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

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio, ϵ is the clipping threshold, and $\hat{A}_t$ is the estimated advantage function. PPO is particularly suitable for continuous control tasks in CV due to its stability and efficiency.

SAC is an entropy-regularized actor-critic algorithm, where a soft Q-learning objective is used to encourage exploration by adding an entropy term to the reward. SAC maximizes the expected cumulative reward with entropy regularization:

$$L(\theta) = E_{(s,a,r,s')\sim\mathcal{D}}\left[(r + \gamma\mathbb{E}_{a'\sim\pi}\left[Q(s',a') - \alpha\log\pi(a'|s')\right] - Q(s,a))^2\right]$$

where α is a temperature parameter that controls the trade-off between exploration and exploitation. SAC has shown superior performance in tasks requiring exploration in large, continuous action spaces, making it highly applicable for dynamic CV applications.

5.2 Hyperparameter Optimization in Model Training

The performance of DQN, PPO, and SAC significantly depends on the choice of hyperparameters during training, which influences the stability and convergence speed of each model. Key hyperparameters to consider include the learning rate η, discount factor γ, batch size N, and target network update frequency for DQN, among others.

DQN: Sensitive to the learning rate η and the size of the experience replay buffer $\mathcal{D}$. A smaller η can lead to stable learning but slower convergence. Additionally, an optimal target network update frequency can reduce the risk of diverging from the optimal Q-values.

PPO: Highly dependent on the clipping parameter ϵ and the batch size Nused in the gradient update. Larger N values can enhance training stability but increase computational costs, especially in large CV applications.

SAC: The temperature parameter α is critical in SAC, as it controls the entropy bonus and, therefore, the degree of exploration. High values of α encourage exploration but may lead to suboptimal performance in deterministic tasks, whereas lower values emphasize exploitation.

Proper tuning of these hyperparameters can dramatically improve the performance of DRL algorithms in CV tasks, as a well-optimized model can learn faster and achieve higher rewards, while poorly chosen parameters may lead to unstable or suboptimal policies.

5.3 Applicability Analysis of Algorithms Based on Computer Vision Task Requirements

Since applications of computer vision span from image classification to robotic control, the use of RL must be tailored to fit the particular subtask. Different DRL algorithms have different strengths that may line up better with the structure, complexity and control of specific CV tasks. In this work, we examine the fitness of DQN, PPO and SAC for these classes of CV tasks, providing some insights over their suitability and drawbacks.

DQN (Deep Q-Networks): DQN is ideal for discrete action space problems, which provided a solution for image classification and object detection as it is easy to constrain the actions into discrete choices like classifying or localizing an object. An effective strategy based on maximizing near-term visual information gain, Q-learning performs well in static and structured environments. Nonetheless, DQN faces limitations when applied to high-dimensional or continuous action-space applications such as robotic control and visual navigation that demand rich motor control and real-time adaptation. DQN is inherently limited because it works with discrete actions so it struggles at dynamic or continuous state representations and in more complex control problems its granularity is often insufficient.

PPO (Proximal Policy Optimization): PPO offers a balance between exploration and exploitation, making it highly suitable for tasks requiring continuous control over complex environments, such as autonomous driving, continuous visual tracking, and robotic object manipulation. These applications often require precise adjustments and smooth actions, where sudden, large updates in policy could destabilize performance. PPO's clipped objective function promotes incremental policy updates, which enhances learning stability and resilience in environments with frequent variations. In CV tasks with extensive action spaces and a need for continuous adjustments, PPO demonstrates robust performance and generalizes well across various tasks. Additionally, its simplicity and efficiency make it ideal for large-scale tasks in high-dimensional state spaces, where stability and reliability are critical.

SAC (Soft Actor-Critic): SAC excels in tasks demanding both high exploration and fine-grained control, especially where environments are unpredictable, dynamic, or stochastic. Applications such as visual navigation, complex robotic manipulation, and autonomous drones rely on policies that can adapt to varying states and uncertain environments, where SAC's entropy-driven objective encourages policies that remain adaptable and exploratory. By maximizing both reward and entropy, SAC promotes diverse action strategies that can handle sparse rewards and complex visual cues, maintaining robustness in environments with a high degree of randomness. Its unique balance between exploration and exploitation makes SAC advantageous for CV tasks where reliable performance under uncertainty is essential, enabling adaptive, resilient decision-making in the face of noisy or incomplete visual data.

Journal of Industrial Engineering and Applied Science
Journal Home: http://jieas.suaspress.org/ | CODEN: JIEAAE
Vol. 2, No. 6, 2024 | ISSN 3005-6071 (Print) | ISSN 3005-608X (Online)

6 CV DATASETS AND BENCHMARK TESTING

6.1 COMMON DATASETS

Datasets are essential in building and validating DRL models for CV since they constitute the raw data used to train and evaluate algorithms. Some of these are COCO (Common Objects in Context), ImageNet, and Cityscapes which each have their own usecase when it comes to CV work.

Well known for its large scale, COCO has more than 330k images and over 2.5 million object instances labeled in 80 categories. This is especially beneficial for object detection, segmentation, and captioning tasks. COCO provides variety of objects and images together which allows DRL models to learn on a very diverse environments. For example, in COCO you can have a image with overlapping objects using the variety present in COCO, it will help to train robust models as well since this mesh of images will be kind of more real life data.

ImageNet, another foundational dataset, is renowned for its massive scale, with over 14 million labeled images spanning 1,000 object categories. ImageNet is widely used for image classification tasks and serves as a benchmark for evaluating the performance of deep neural networks, especially in large-scale visual recognition. ImageNet's extensive labeling enables DRL models to learn a broad range of visual features, which are crucial for high-level recognition tasks and transfer learning to other domains.

Cityscapes offers a dataset tailored for urban scene understanding, featuring high-resolution images from 50 cities. It focuses on semantic segmentation and pixel-level understanding of street scenes, making it especially valuable for applications like autonomous driving. Cityscapes contains annotated data for road infrastructure, traffic signs, and pedestrians, which are key components for intelligent transportation systems. DRL can leverage this dataset to enhance models' decision-making in real-time applications, such as obstacle avoidance and path planning in dynamic environments.

By using these datasets, researchers can evaluate the generalization abilities of DRL models, compare different algorithms, and benchmark performance across tasks. They provide the diversity and complexity needed to push DRL models to handle real-world computer vision problems.

6.2 BENCHMARK TESTING

Benchmark testing is essential for evaluating the performance of DRL models in CV applications. The primary goal is to assess how well an algorithm performs across a variety of tasks under realistic conditions. In the context of DRL, this testing typically focuses on several key performance metrics: time efficiency, accuracy, and robustness.

Time efficiency refers to the ability of a model to make decisions and process information in real-time, which is critical for applications in fields like autonomous driving and robotics. DRL algorithms often require many iterations to converge to an optimal policy, and finding the balance between exploration and exploitation becomes crucial for ensuring timely responses in dynamic environments.

Accuracy is a core metric, particularly in tasks like object detection and segmentation. Here, the DRL model is evaluated based on its ability to correctly identify objects, delineate regions of interest, and predict visual cues from input data. For instance, in an object detection task, accuracy is commonly assessed using the mean average precision (mAP) metric, which averages the precision at different recall levels.

Mathematically, mAP is defined as:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i$$

Where N is the number of classes, and AP_i is the average precision for class i, which can be calculated as:

$$AP_i = \frac{\sum_r P(r) \cdot \Delta r}{\sum_r P(r)}$$

Where $P(r)$ is the precision at recall r, and Δr represents the change in recall.

Robustness refers to the ability of a DRL model to maintain consistent performance across a variety of challenging conditions. This can include handling noisy input data, dealing with changes in lighting, or adapting to previously unseen environments. Robustness testing often involves evaluating how well the model generalizes to new, unseen data or performs in less ideal conditions, such as in adversarial environments or with missing input features.

Benchmarking DRL models in CV tasks often involves comparing several different algorithms and architectures, ranging from traditional Q-learning methods to more advanced techniques like Proximal Policy Optimization (PPO) or Soft Actor-Critic (SAC). Through such comparisons, researchers can gain insights into which approaches are best suited for particular types of visual tasks, balancing trade-offs between performance metrics such as speed, accuracy, and stability.

6.3 EVALUATION METRIC ANALYSIS

The choice of evaluation metrics is crucial for accurately assessing DRL models in computer vision applications. The metrics must align with the specific task requirements, as different tasks emphasize different aspects of model performance. The most common evaluation metrics used for assessing DRL models in CV are Precision, Recall, and Mean Average Precision (mAP), though other metrics, such as IoU (Intersection over Union), F1 Score, and Tracking Accuracy, are also commonly used.

Precision measures the proportion of positive predictions that are actually correct. It is calculated as:

$$Precision = \frac{TP}{TP + FP}$$

Where TP is the number of true positives and FP is the number of false positives. In object detection, high precision means that the model does not produce many false positives, which is crucial for tasks like autonomous driving where misidentifying objects can have severe consequences.

Recall assesses the proportion of actual positive instances that are correctly identified by the model. It is given by:

$$Recall = \frac{TP}{TP + FN}$$

Where FN is the number of false negatives. Recall is particularly important in applications like medical image analysis, where missing a positive diagnosis can be fatal.

Mean Average Precision (mAP), as mentioned earlier, is an aggregate metric used to evaluate detection tasks. It provides a more comprehensive view of a model's performance across varying levels of recall, ensuring that the model balances precision and recall effectively. For example, mAP values that are consistently high across different recall thresholds indicate that the model is both precise and able to detect a wide range of object instances.

For tasks like target tracking, the performance metrics become more specialized. Stability and consistency in real-time tracking are critical, as the model needs to continuously track a moving object under varying conditions. In these cases, metrics like Tracking Accuracy (TA) and Normalized Object Tracking Precision (NOTP) are commonly used.

$$TA = \frac{\text{number of correct track assignments}}{\text{total number of objects in the sequence}}$$

Other advanced evaluation metrics, like IoU for segmentation tasks, are used to measure the overlap between predicted and ground-truth regions. In segmentation, a higher IoU indicates that the model is able to correctly segment the relevant regions from the background. Mathematically, IoU is defined as:

$$IoU = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$

Selecting the right evaluation metric based on task characteristics ensures that the model performs well in its intended application. For example, in autonomous driving, both mAP (for object detection) and tracking accuracy are essential to ensure that objects are detected reliably and tracked consistently across frames. In medical image analysis, precision and recall are key to minimizing false positives and false negatives in disease detection.

By conducting thorough evaluations across different metrics, researchers can more effectively assess DRL models' capabilities, refine their performance, and guide future developments in the field of computer vision.

7 CASE ANALYSIS: CUTTING-EDGE APPLICATIONS OF RL IN CV

7.1 Adaptive Robot Vision Systems

Adaptive robot vision systems represent one of the most promising applications of Deep Reinforcement Learning in real-world robotics. The primary goal in such systems is to enable robots to navigate autonomously in dynamic and often unpredictable environments. By using reward-based mechanisms, robots can continuously learn from interactions with their surroundings, adapting their strategies to handle a wide variety of obstacles, changes in lighting, or environmental uncertainties.

In particular, DRL empowers robots to build comprehensive environmental maps, recognize obstacles, and make intelligent decisions regarding pathfinding. This ability is particularly beneficial in complex settings such as automated warehouses, where robots must navigate narrow aisles while interacting with moving human workers and other machines. Moreover, DRL can improve the efficiency of object recognition systems by adjusting the robot's visual perception to match the task at hand. For example, a robot might learn to differentiate between objects based on priority or contextual relevance, whether it's sorting packages in a factory or identifying household items in a smart home environment.

7.2 Smart Surveillance Systems

The DRL in smart surveillance systems is changing the way public spaces are monitored, security is managed, and incidents responded to directly. Conventional surveillance systems are usually dependent on tailored algorithms that are unlikely to respond accordingly, or timely, during the evolution of events. In contrast, DRL feeds into the entire surveillance system with new adaptability that helps the system to adapt themselves according to new faced situation without human intervention.

A prominent example of DRL in surveillance is anomaly detection, where the system constantly evolves and discerns between normal and abnormal patterns from video streams. With the help of massive amounts of surveillance footage, it is able to learn suspicious activities, such as theft, fights or accidents visually. By refining detection methods and adjusting policies to minimize false positives and false negatives, DRL agents can learn optimal strategies for detecting these anomalies.

Traditional surveillance systems face challenges in realistic environments, like crowded public spaces or highly variable traffic condition. DRL partly mitigates this issue by allowing surveillance systems to modify their decision-making behavior based on constantly changing conditions. As a concrete use case, the RL agent can optimize how to allocate its focus among multiple traffic lanes for a traffic monitoring system depending on which one has more vehicles and is likely congested enabling better detection of vehicle flow as well as identification of incidents. In the same

way, smart surveillance systems based on DRL can alter their monitoring of active hotspot based on what is happening in a certain area to watch for any breaches in public safety.

The ways of their operation can have social effects as well such as improved response time and greater threat detection capabilities. In cases when something goes wrong, such type of intelligent surveillance system can enable human operators to view real-time data so they can decide how to respond, whether that is dispatching security personnel or directing medical teams to a critical area.

7.3 CV Applications in Games

Deep reinforcement learning with computer vision has also been an important trend in the gaming domain, where agents need to extract information from visual perception based on complicated and dynamic environment quorums to perform appropriate actions. CV techniques allow RL agents to convert high-dimensional visual data — character positioning, terrain details, and movement patterns into lower dimensional representations that provide context for digital brains when making decisions — for games modeled with real-time strategies in visually-rich worlds. The integration of CV and DRL creates agents capable of perceiving game states through raw visual inputs while dynamically reacting to a continually changing game environment.

For example, in classic Atari games, RL agents equipped with convolutional neural networks interpret pixel-based visual inputs to determine optimal actions, such as moving or shooting, in response to the game state. In this setting, CNNs allow agents to directly process raw image data, learning patterns from visual cues without manual feature engineering[12]. This approach, as seen in "Playing Atari with Deep Reinforcement Learning," showcases how CV enables RL agents to succeed across diverse game environments using only visual input.

Similarly, in AlphaGo, the RL agent processes the Go board as a visual input to evaluate possible moves and plan complex, multi-step strategies. By leveraging CNNs to analyze board states, the agent learns optimal play strategies without human knowledge, setting a new benchmark in game AI[13].

In 3D multiplayer games like Quake III Arena, CNNs help RL agents process three-dimensional environments, recognizing other players' positions and complex spatial layouts. By utilizing population-based reinforcement learning, these agents achieve human-level performance in navigating and competing in highly dynamic environments.

In the popular mobile game Honor of Kings, a hierarchical macro-strategy model integrates CNNs to analyze the game map, detecting ally and enemy positions. This enables the agent to make high-level strategic decisions, such as when to attack, defend, or cooperate with other agents, showcasing the power of combining CV with DRL in multiplayer online battle arenas (MOBA) to create sophisticated AI strategies[14].

Furthermore these games allow to test how robust RL algorithms are in changing and ambiguous environments. In competitive video games, the environment and opponent strategies are rapidly changing which requires DRL models to adapt quickly to new situations. Somewhere in here is where the shined ability of DRL to explore broad areas of strategy and evolve over time. The same adaptability of RL agents can be easily applied to many types of CV tasks as well, like in the case of various autonomous driving and robotic control problems.

7.4 Medical Image Analysis

Medical image analysis is one of the most critical and high-risk areas where DRL can make a significant impact. In medical environments, accuracy and interpretability are paramount, especially when the system's decisions affect patient health outcomes. Due to the life-or-death nature of many medical conditions, any errors in diagnosis or treatment recommendations based on medical imagery can have severe consequences. DRL has demonstrated its potential in a variety of medical applications, such as disease detection, organ segmentation, and tumor classification, by offering automated, precise, and consistent analysis of medical imagery, which is often too complex for traditional methods.

DRL has been applied in this field for automating image segmentation problems, where the objective is to accurately segment and localize the regions of interest from healthy surrounding tissues (e.g. This is all the more important in time-critical medical environments — radiology departments, for example, where a prompt and correct diagnosis is paramount for enhancing patient outcomes. The DRL mechanism can also be validated to enhance segmentation strategies since DRL agents receive rewards when identifying abnormalities more accurately in medical images such as CT scans, MRIs and X-rays, therefore ensuring precise abnormalities identification is essential for trustworthy and timely clinical decisions.

8 CURRENT CHALLENGES AND FUTURE TRENDS

8.1 Model Training Efficiency and Computational Resource Demands

Training Deep Reinforcement Learning models for high-dimensional tasks, particularly in visual domains like robotics, autonomous driving, and computer vision , demands considerable computational powerp[15]. High-resolution images and complex environments generate massive data loads, requiring extensive GPU or TPU resources, which in turn drive up both time and cost. As models become more intricate, they also require careful tuning of hyperparameters and significant memory capacity, adding to the complexity and expense of deployment.

Improving training efficiency in DRL has direct positive implications for computer vision . Faster training cycles and lower resource consumption would make it more feasible to

deploy DRL-based CV solutions in real-time applications, such as facial recognition, autonomous navigation, or medical image analysis[16]. For example, in CV tasks that involve processing large volumes of visual data, such as in surveillance or remote sensing, improved DRL training efficiency can accelerate model development and deployment, enabling quicker adaptation to new environments or tasks.

In the future, optimizing model training efficiency and minimizing resource usage will become essential for making DRL scalable and accessible across various industries[17]. Emerging research is expected to emphasize techniques such as model compression, where large models are streamlined without sacrificing performance, and distributed learning, which leverages multiple processors to speed up training. Additionally, advancements in hardware, such as the development of specialized AI processors, could further reduce resource demands and make DRL solutions more feasible for broader applications, including those that heavily rely on CV tasks.

8.2 Sample Efficiency and Exploration Strategies

Real-world applications of DRL often involve environments where data samples are sparse, making it challenging to collect enough information for effective training. For instance, in fields like healthcare, autonomous navigation, or finance, real-world interactions may be limited due to safety concerns, high costs, or data availability restrictions[18]. Improving sample efficiency is therefore crucial to ensure that DRL models can learn effectively from limited experiences or interactions.

In order to fix this, future work may harness self-supervised learning, whereby the model creates its own training signal by predicting or inferring missing parts of information based on data it already possesses. Better exploration strategy (curiosity-driven exploration, intrinsic motivation) is another area with a great promise that helps the models to visit less traveled states. These developments will create adaptive and robust systems in data-constraining, expensive-to-collect environments by improving the ability of DRL models to overcome sparse rewards and rare data.

9 CONCLUSION

Due to the diversity of the domains to which DRL is applied, this paper presents a comprehensive review of recent developments in Deep Reinforcement Learning for Computer Vision applications covering some fundamental aspects on what gives it foundation explain key algorithms and its application over various computer vision tasks. Specifically, the author spend much of the paper discussing mainstream algorithms — Q-learning, policy gradient and Actor-Critic methods — and how effective they are for high-dimensional visual environments. Key CV applications, including object detection, image segmentation, target tracking, and image generation, are discussed. The paper enlist novel strategies including hierarchical policy learning, adaptive reward design, multi-task learning and domain adaptation to improve efficiency and generalizability of the models. The paper elucidates the enormous potential of DRL on CV tasks along with inherent challenges in terms of resource consumption, sample efficiency, and real-time action production, while outlining corresponding countermeasures. This review serves twofold – first, it presents an overview of the works involved in the intersection of DRL and CV and second, it provides potential future research directions and application areas.

ACKNOWLEDGMENTS

The authors thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

FUNDING

Not applicable.

INSTITUTIONAL REVIEW BOARD STATEMENT

Not applicable.

INFORMED CONSENT STATEMENT

Not applicable.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

CONFLICT OF INTEREST

The 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.

PUBLISHER'S NOTE

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

AUTHOR CONTRIBUTIONS

Not applicable.

ABOUT THE AUTHORS

LI, Zhengyang

DigiPen Institute of Technology, Redmond, WA, USA.

REFERENCES

[1] Swaminathan, M., Bhatti, O. W., Guo, Y., Huang, E., & Akinwande, O. (2022). Bayesian learning for uncertainty quantification, optimization, and inverse design. IEEE Transactions on Microwave Theory and Techniques, 70(11), 4620-4634.

[2] Garcia, F., & Rachelson, E. (2013). Markov decision processes. Markov Decision Processes in Artificial Intelligence, 1-38.

[3] Sze, V., Chen, Y. H., Yang, T. J., & Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. Proceedings of the IEEE, 105(12), 2295-2329.

[4] Yukun, S. (2019). Deep learning applications in the medical image recognition. American Journal of Computer Science and Technology, 2(2), 22-26.

[5] Chauhan, R., Ghanshala, K. K., & Joshi, R. C. (2018, December). Convolutional neural network (CNN) for image detection and recognition. In 2018 first international conference on secure cyber computing and communication (ICSCCC) (pp. 278-282). IEEE.

[6] Legenstein, R., Wilbert, N., & Wiskott, L. (2010). Reinforcement learning on slow features of high-dimensional input streams. PLoS computational biology, 6(8), e1000894.

[7] Pan, J., Wang, X., Cheng, Y., & Yu, Q. (2018). Multisource transfer double DQN based on actor learning. IEEE transactions on neural networks and learning systems, 29(6), 2227-2238.

[8] Gu, S., Lillicrap, T., Sutskever, I., & Levine, S. (2016, June). Continuous deep q-learning with model-based acceleration. In International conference on machine learning (pp. 2829-2838). PMLR.

[9] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

[10] Sun, Y., Salami Pargoo, N., Jin, P., & Ortiz, J. (2024, October). Optimizing Autonomous Driving for Safety: A Human-Centric Approach with LLM-Enhanced RLHF. In Companion of the 2024 on ACM International Joint Conference on Pervasive and Ubiquitous Computing (pp. 76-80).

[11] Cao, J., Xu, R., Lin, X., Qin, F., Peng, Y., & Shao, Y. (2023). Adaptive receptive field U-shaped temporal convolutional network for vulgar action segmentation. Neural Computing and Applications, 35(13), 9593-9606.

[12] Chen, B., Qin, F., Shao, Y., Cao, J., Peng, Y., & Ge, R. (2023). Fine-grained imbalanced leukocyte classification with global-local attention transformer. Journal of King Saud University-Computer and Information Sciences, 35(8), 101661.

[13] Jiang, L., Yang, X., Yu, C., Wu, Z., & Wang, Y. (2024, July). Advanced AI framework for enhanced detection and assessment of abdominal trauma: Integrating 3D segmentation with 2D CNN and RNN models. In 2024 3rd International Conference on Robotics, Artificial Intelligence and Intelligent Control (RAIIC) (pp. 337-340). IEEE.

[14] Gu, Y., Liu, B., Zhang, T., Sha, X., & Chen, S. (2024). Architectural Synergies in Bi-Modal and Bi-Contrastive Learning. IEEE Access.

[15] Plaat, A., Kosters, W., & Preuss, M. (2020). Deep model-based reinforcement learning for high-dimensional problems, a survey. arXiv preprint arXiv:2008.05598.

[16] Rao, N., Chu, S. L., Jing, Z., Kuang, H., Tang, Y., & Dong, Z. (2022, June). Exploring information retrieval for personalized teaching support. In International Conference on Human-Computer Interaction (pp. 191-197). Cham: Springer Nature Switzerland.

[17] Sami, H., Otrok, H., Bentahar, J., & Mourad, A. (2021). AI-based resource provisioning of IoE services in 6G: A deep reinforcement learning approach. IEEE Transactions on Network and Service Management, 18(3), 3527-3540.

[18] He, Z. (2024). An Empirical Analysis of the Factors Influencing Commodity Housing Prices in China Based on Econometrics. In INTERNET FINANCE AND DIGITAL ECONOMY: Advances in Digital Economy and Data Analysis Technology The 2nd International Conference on Internet Finance and Digital Economy, Kuala Lumpur, Malaysia, 19–21 August 2022 (pp. 99-112).

[19] Yu, P., Cui, V. Y., & Guan, J. (2021, March). Text classification by using natural language processing. In Journal of Physics: Conference Series (Vol. 1802, No. 4, p. 042010). IOP Publishing.

[20] Lin, W. (2024). A Review of Multimodal Interaction Technologies in Virtual Meetings. Journal of Computer Technology and Applied Mathematics, 1(4), 60-68.

[21] Lin, W. (2024). A Systematic Review of Computer Vision-Based Virtual Conference Assistants and Gesture Recognition. Journal of Computer Technology and Applied Mathematics, 1(4), 28-35.

[22] Sun, Y., & Ortiz, J. (2024). Machine Learning-Driven Pedestrian Recognition and Behavior Prediction for Enhancing Public Safety in Smart Cities. Journal of Artificial Intelligence and Information, 1, 51-57.

[23] Luo, M., Zhang, W., Song, T., Li, K., Zhu, H., Du, B., & Wen, H. (2021, January). Rebalancing expanding EV sharing systems with deep reinforcement learning. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence (pp. 1338-1344).

[24] Luo, M., Du, B., Zhang, W., Song, T., Li, K., Zhu, H., ... & Wen, H. (2023). Fleet rebalancing for expanding shared e-Mobility systems: A multi-agent deep reinforcement learning approach. IEEE Transactions on Intelligent Transportation Systems, 24(4), 3868-3881.

[25] Zhu, H., Luo, Y., Liu, Q., Fan, H., Song, T., Yu, C. W., & Du, B. (2019). Multistep flow prediction on car-sharing systems: A multi-graph convolutional neural network with attention mechanism. International Journal of Software Engineering and Knowledge Engineering, 29(11n12), 1727–1740.

[26] Yaonian Zhong, Enhancing the Heat Dissipation Efficiency of Computing Units Within Autonomous Driving Systems and Electric Vehicles, J. World Journal of Innovation and Modern Technology, 2024, 7 (5): 100-104.