DSR-Net: Dynamic Star Map Denoising Algorithm Based on Deep Reinforcement Learning

Yifan Zhao[1], Shiji Song[2,*], Shaochen Jiang[3]

{zhaoyifan@stu.xju.edu.cn[1], shijis@mail.tsinghua.edu.cn[2], scj@xju.edu.cn[3]}

School of Computer Science and Technology, Xinjiang University, Xinjiang, China[1]
School of Information Science and Technology, Tsinghua University, Beijing, China[2]
School of Computer Science and Technology, Xinjiang University, Xinjiang, China[3]
*corresponding author

Abstract. As astronomical observation technology continues to progress, obtaining high-quality star maps provides us with valuable opportunities to explore the universe. However, The acquired star maps are often affected by various random noises, including speckle noise, Poisson noise, Impulse noise, Thermal noise, Reynolds noise, and Gaussian noise, etc. These noises degrade the image quality and limit the efficiency of scientific research. Traditional denoising methods are often limited in their effectiveness when faced with such complex noise and lack the ability to model the temporal features of dynamic star maps, making it difficult to handle the sparsity and complex background of dynamic star maps. Therefore, this paper introduces DSR-Net, a deep reinforcement learning-based dynamic star map denoising algorithm. The algorithm combines Convolutional Gated Recurrent Units (ConvGRU) and Region-based Reward Convolution (Rrc) modules, enabling it to capture the dynamic changes of star maps, effectively remove noise, and preserve important details in the star maps. Experimental results show that DSR-Net outperforms traditional denoising methods on multiple real dynamic star map datasets, providing an effective solution for improving the quality of star maps.

Keywords: Star image, Image denoising, Deep reinforcement learning.

1 Introduction

High-quality star maps provide us with valuable opportunities to explore the universe. Since humans first observed the sky using early astronomical telescopes, advancements in telescope technology, such as the Hubble and James Webb Space Telescopes, have revealed many cosmic mysteries. However, the challenge in astronomy is not only the development of advanced telescopes but also the optimization of image processing workflows to enhance image quality and reduce noise interference. Common enhancement methods, such as extended exposure times and stack imaging, often introduce more noise and complexity, limiting research efficiency. Acquired star maps are often affected by various types of random noise, including Gaussian noise, Poisson noise, and speckle noise. The random distribution of these noises increases the difficulty of denoising, and single denoising methods are often not effective for all types of noise, thus affecting the clarity and quality of star maps. Therefore, dynamic star map denoising becomes a key technology for improving star map quality, as it can suppress noise interference while preserving details, facilitating scientific exploration [1].

CONF-MLA 2024, November 21, Adana, Turkey

DOI 10.4108/eai.21-11-2024.2354638

Deep learning has made remarkable progress in image classification, segmentation, enhancement, and generation, driven by the advancement of machine learning techniques.

In the field of image enhancement, Transformers capture global dependencies using self-attention mechanisms, which helps with denoising, while convolutional neural networks (CNN) identify noise by extracting local features through multiple convolution layers [2]. However, the sparsity, complex backgrounds, and weak signals of star map images present challenges for traditional deep learning methods (such as CNNs and Transformers) in star map enhancement tasks. To address these issues, deep reinforcement learning (DRL) offers an adaptive approach, enhancing star map quality by learning optimal strategies and adjusting processing methods in real-time [3]. In addition, existing methods generally lack the ability to model the temporal features of dynamic star maps, making it difficult to effectively address the complexity of star maps in dynamic environments [1]. Based on this, a DRL method is proposed to reduce observation time and lower noise in astronomical images, which is of great significance for upcoming large-scale astronomical surveys.

This paper, based on a fully convolutional A3C network structure, introduces Convolutional Gated Recurrent Units (ConvGRU) [4] and a Region-based Reward Convolution module, and proposes a new dynamic star map denoising algorithm, DSR-Net. This algorithm effectively captures the dynamic changes of star maps in time series, enabling the agent to better understand local region features. The key contributions of this paper are outlined as follows:

- A region-based reward convolution module for dynamic star map denoising is proposed, targeting pixel-level denoising. This allows the agent to focus on the value and strategy of neighboring pixels, thereby improving the understanding of local region features.
- The Convolutional Gated Recurrent Unit is incorporated into the network to enhance the agent's ability to perceive dynamic changes in star maps over time, improving the denoising effect.
- Experiments were conducted on multiple real dynamic star map datasets. The experimental results show that the proposed model exceeds existing traditional denoising methods, yielding superior denoising results.

2 Literature Review

This chapter reviews the development of deep reinforcement learning and the current research status of image denoising.

2.1 Deep Reinforcement Learning

A key milestone in deep reinforcement learning is the DQN algorithm introduced by DeepMind. By using neural networks to approximate the Q-function, DQN integrates deep learning with reinforcement learning, achieving significant progress in decision-making within high-dimensional state spaces and exceeding human-level performance in Atari game tasks. The success of DQN has led to the development of numerous deep reinforcement learning (DRL) algorithms, which have gradually been applied to image processing tasks. In image denoising tasks, Yu et al. [5] proposed the RL-Restore method, which allows agents to choose the best restoration toolchain from a lightweight CNN toolkit, effectively improving image quality. Burak et al. [6] developed a conditional reinforcement learning agent that selects different resolutions based on the image content to optimize detection results, thus saving the need for high-resolution images. Cao et al. [7] introduced the Attention-FH framework, which enhances local regions to progressively improve the resolution of facial images. However, star map data is scarce and highly susceptible to noise, lighting, and other factors. The star map denoising task requires higher precision in both detail preservation and noise suppression. Compared to conventional image processing tasks, star map processing requires specially designed and optimized reinforcement learning algorithms to address its unique denoising needs and features.

2.2 Image Denoising

Image denoising focuses on eliminating noise from images to enhance their quality and make them clearer for further processing and analysis. Denoising methods are divided into non-learning methods and learning-based methods. Among non-learning methods, histogram equalization [8] enhances contrast by uniformly distributing intensity but may introduce noise; Gaussian filtering [9] denoises by weighted averaging, but may blur details; total variation denoising [10] smooths the image while preserving edges but sometimes produces artifacts. Lately, research on image denoising has focused on learning-based methods, particularly the application of convolutional neural networks (CNNs) in denoising and deblurring [11]. Reinforcement learning (RL) has also developed in this area. Guo et al. [12] proposed a pixel-level cross-modal denoising RL method, where each pixel is treated as an agent that learns an optimal strategy. Zhang et al. [13] modeled low-light enhancement as a Markov decision process for low-light images, achieving flexible weak light enhancement. In star map denoising tasks, Xie et al. [14] combined maximum likelihood estimation (MLE) and machine learning algorithms to improve the denoising effect of star maps.

3 Methods

In this chapter, the focus is primarily on presenting the comprehensive framework of DSR-Net.

3.1 Overall Architecture

In the star map denoising task, the noise distribution typically exhibits high locality and non-uniformity, with different regions potentially being affected by various types of noise (e.g., Gaussian noise, background noise, etc.). Due to this characteristic, image-level denoising decisions often struggle to balance detail preservation and noise suppression. Therefore, this paper proposes a network architecture specifically for star map denoising—DSR-Net, based on the fully convolutional A3C framework [15]. Through asynchronous parallel computation of the fully convolutional A3C network, the training is accelerated, and by combining the Actor-Critic structure with pixel-level denoising decision-making ability [16], DSR-Net can more flexibly handle local noise features in the star map. In this framework, the entire star map denoising task is formulated as a Markov Decision Process (MDP). For the current time step t, let the star map being processed consist of N images (i=1,2,...,N), where each pixel i is the current state si(t) of the agent i. The reward r(t) is determined by assessing the difference between the original image and the image after processing, while the action ai(t) involves selecting the denoising filter operation for the image, such as Gaussian blur, bilateral filter, or median filter. The gradient update of the policy network and value network in the A3C architecture is the core and key part [17]. Let the parameters of the two networks be θp and θv.

The input to both networks is the current state si(t), and the value network outcomes the value V(si(t)) of the current state, which represents the cumulative future reward that can be obtained starting from state si(t). This is used to evaluate the quality of the current star map noise processing state. The gradient update of θv is computed as follows:

$$R_i^{(t)} = r_i^{(t)} + \gamma r_i^{(t+1)} + \gamma^2 r_i^{(t+2)} + \cdots + \gamma^{n-1} r_i^{(t+n-1)} + \gamma^n V\left(s_i^{(t+n)}\right) \tag{1}$$

$$d\theta_v = \nabla_{\theta_v}\left(R_i^{(t)} - V\left(s_i^{(t)}\right)\right)^2 \tag{2}$$

The policy network represents the probability of executing each denoising action given the current star map state. Through policy gradient, the network learns how to select denoising operations in order to maximize long-term rewards. The policy π(ai(t)| si(t)) used to generate actions is updated through the gradient of θp, and the gradient update of θp is calculated as follows:

$$A\left(a_i^{(t)}, s_i^{(t)}\right) = R_i^{(t)} - V\left(s_i^{(t)}\right) \tag{3}$$

$$d\theta_p = -\nabla_{\theta_p} \log \pi\left(a_i^{(t)} | s_i^{(t)}\right) A\left(a_i^{(t)}, s_i^{(t)}\right) \tag{4}$$

The goal of DSR-Net is to learn the optimal policy π = (π₁, ..., πₙ), which maximizes the expected total reward for pixel i:

$$\pi_i^* = \arg \underset{\pi_i}{max}\, \mathrm{E}_{\pi_i}\left[\sum_{t=0}^{\infty} \gamma^t r_i^{(t)}\right] \tag{5}$$

Considering that star maps typically contain various celestial bodies and complex backgrounds, processing individual pixels may not fully account for the relationships between these elements, potentially limiting the training effectiveness of the network agent and affecting overall quality. Additionally, the temporal nature of star maps makes capturing dynamic changes particularly important. Therefore, the introduction of Convolutional Gated Recurrent Units (ConvGRU) effectively captures the dynamic changes of star maps in time series, and the proposed Region-based Reward Convolution (Rrc) enables the agent to better understand the features of local regions. Figure 1 shows the overall framework of DSR-Net.

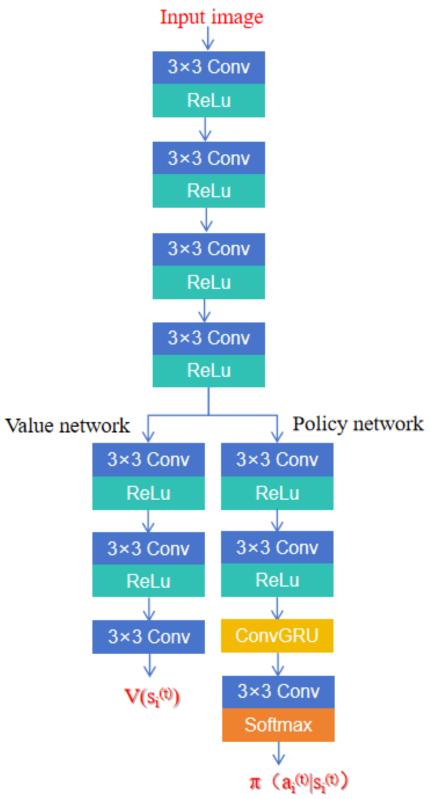

Fig. 1. The overall framework of DSR-Net

3.2 Convolutional Gated Recurrent Units

The introduction of ConvGRU allows the model to more flexibly address the challenges posed by temporal sensitivity and environmental changes when processing star map sequences [18]. Since the features of celestial bodies and noise distribution in star maps may change over time and under different observational conditions, the use of ConvGRU enables dynamic state adjustment through its gating mechanism, allowing the network to retain critical information while effectively filtering out noise. The update gate ensures that historical information is not entirely lost; when star map features remain consistent, the historical information helps enhance the denoising effect.

$$z_i^{(t)} = \sigma\left(\mathrm{Conv}\left(s_i^{(t)}, W_z\right) + \mathrm{Conv}\left(h_i^{(t-1)}, U_z\right)\right) \tag{6}$$

The gating mechanism zi(t) controls the combination ratio between the previous hidden state hi(t−1) and the current input state si(t), determining how much information from the previous hidden state hi(t−1) should be retained and how much of the current input si(t) should be integrated when updating the hidden state. The reset gate helps prevent the accumulation of historical noise information, allowing the model to focus more on the important information at the current time step.

$$r_i^{(t)} = \sigma\left(\mathrm{Conv}\left(s_i^{(t)}, W_r\right) + \mathrm{Conv}\left(h_i^{(t-1)}, U_r\right)\right) \tag{7}$$

The reset gate ri(t) determines which parts of the previous hidden state hi(t−1) should be ignored, allowing the network to selectively "forget" part of the past memory when generating the current hidden state. The candidate hidden state is generated based on the current input and important historical information, enabling the model to selectively retain or forget historical information as needed.

$$\tilde{h}_i^{(t)} = \tanh\left(\mathrm{Conv}\left(s_i^{(t)}, W\right) + \mathrm{Conv}\left(r_i^{(t)} \odot\ h_i^{(t-1)}, U\right)\right) \tag{8}$$

The final hidden state is computed based on both the current input and historical information, and is used for subsequent decision-making.

$$h_i^{(t)} = \left(1 - z_i^{(t)}\right) \odot h_i^{(t-1)} + z_i^{(t)} \odot \tilde{h}_i^{(t)} \tag{9}$$

By applying ConvGRU at each time step, the agent can consider the contextual information from the previous frame while processing the current image, allowing for more effective decision-making in reward feedback and ultimately improving the overall performance of star map denoising.

3.3 Region-based Reward Convolution

Combined with the fully convolutional A3C framework, the network achieves a large receptive field. With the help of the dilated convolution idea in fully convolutional networks, The A3C policy network and value network in addition to focusing on the i-th pixel si(t), also consider the value and policy generated by neighboring pixels for the i-th pixel. Simply put, the action ai(t) can influence the pixels in the local region centered on pixel i. Therefore, the expected total reward for agent i at time t can be redefined as:

$$R_i^{(t)} = r_i^{(t)} + \gamma \sum_{j \in N(i)} w_{i-j} V\left(s_j^{(t+1)}\right) \tag{10}$$

w can be viewed as the weight of the convolution kernel, which represents how much the influence of neighboring pixel values V should be considered in the following time step. w is learned as a parameter alongside the policy network and value network. Since each pixel in the star map has a two-dimensional coordinate, the second component in equation (10) is a two-dimensional convolution. Therefore, using matrix notation, the R(t) for n steps can be defined as:

$$R^{(t)} = r^{(t)} + \gamma w \cdot r^{(t+1)} + \gamma^2 w^2 \cdot r^{(t+2)} + \cdots + \gamma^{n-1} w^{n-1} \cdot r^{(t+n-1)} + \gamma^n w^n \cdot V\left(s^{(t+n)}\right) \tag{11}$$

Where R(t), V(s(t)), and r(t) are matrices, with elements Ri(t), V(si(t)), and ri(t) correspondingly. w * r represents n-fold convolution filtering of the reward r. The gradient updates for the policy network and value network can be expressed as follows:

$$d\theta_v = \nabla_{\theta_v} \frac{1}{N} 1^T \left\{\left(R^{(t)} - V\left(s^{(t)}\right)\right) \odot \left(R^{(t)} - V\left(s^{(t)}\right)\right)\right\} 1 \tag{12}$$

$$d\theta_p = -\nabla\theta_p \frac{1}{N} 1^T \left\{\log \pi_{\theta_p}\left(a^{(t)} | s^{(t)}\right) \odot A\left(a^{(t)}, s^{(t)}\right)\right\} 1 \tag{13}$$

Where π(a(t)|s(t)) and s(t) are matrices, with elements si(t) and π(ai(t)|si(t)) respectively. 1 is a vector with all elements equal to 1, and ⊙ denotes element-wise multiplication. The gradient update for w can also be represented in matrix form as:

$$dw = -\nabla_w \frac{1}{N}\sum_{i=1}^{N} \log \pi\left(a_i^{(t)} | s_i^{(t)}\right)\left(R_i^{(t)} - V\left(s_i^{(t)}\right)\right) + \nabla_w \frac{1}{N}\sum_{i=1}^{N} \left(R_i^{(t)} - V\left(s_i^{(t)}\right)\right)^2 \quad (14)$$

$$= -\nabla_w \frac{1}{N} 1^T \{\log \pi\left(a^{(t)} | s^{(t)}\right) \odot A\left(a^{(t)}, s^{(t)}\right)\} 1$$

$$+\nabla_w \frac{1}{N} 1^T \left\{\left(R^{(t)} - V\left(s^{(t)}\right)\right) \odot \left(R^{(t)} - V\left(s^{(t)}\right)\right)\right\} 1 \quad (15)$$

Like common policy gradient algorithms, the first component in equation (14) aims to increase the expected total reward. The second component ensures that during the convolution process, Ri does not deviate from the predicted V(si(t)).

4 Experiments and Results

4.1 Dataset

In this experiment, star maps captured by the PST3S2H4 star sensor were used. These star maps were obtained through high-precision turntables, covering the northern hemisphere's star maps and their orientations, and providing data captured at different angular velocities of the star sensor. Due to factors such as seasons and weather, the acquired star maps have high star magnitudes and contain rich and complex noise, making them suitable for star map denoising tasks. In the experiment, star maps with dynamic angular velocities of 1.0°/s, 2.0°/s, and 3.0°/s were selected, with all star maps having a size of 1292×1040 and an update rate of 10Hz. For each condition, 1000 star maps were chosen, with 800 used for training and 200 for testing.

4.2 Implementation Details

The experiment was conducted on a Tesla T4 GPU with the torch 1.13.1+cu117 platform. The images were standardized to a 256x256 input size to fit the features of the star maps. The batch size was set to 64, and the training images were processed using various data augmentation techniques, including horizontal flipping, translation, and rotation. The full convolutional A3C model was trained using the ADAM optimizer with a polynomial learning rate scheduling strategy. The learning rate started at 1.0 × 10−3 and was gradually adjusted based on the training progress. The max_episode was set to 30,000, with a tmax of 5 for each episode. The full convolutional A3C weights were initialized using pretrained weights that are publicly available, and the filter size of w was set to 33 × 33 to accommodate the local features and noise patterns of the star maps.

4.3 Results and Discussion

Under the same conditions, GATFP [19], Open Operation [20], and DSR-Net were used for star map denoising, and the denoising accuracy was evaluated through Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM).

Comparative experiment

Table 1. Arithmetic Mean of Evaluation Metrics for Comparative Experiments under Three Different Dynamic Conditions

Method	Angular velocity 1.0 °/s		Angular velocity 2.0 °/s		Angular velocity 3.0 °/s	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
GATFP[19]	36.645	0.8785	36.566	0.8762	36.550	0.8737
Open Operation[20]	41.742	0.9539	41.653	0.9513	39.286	0.9288
DSR-Net	46.237	0.9803	46.139	0.9799	46.101	0.9786

Table 1 presents the experimental results of different models on dynamic star map datasets with varying motion conditions. In terms of both PSNR and SSIM, DSR-Net outperforms the other two methods across all tested datasets, with the best performance observed on the dataset with a dynamic angular velocity of 1.0 °/s, achieving a PSNR value of 46.237 and an SSIM value of 0.9803. Compared to Open Operation [20], DSR-Net improves the PSNR and SSIM scores by 4.495 and 0.0264, respectively. This indicates that DSR-Net, based on reinforcement learning, demonstrates stronger adaptability and achieves better denoising results under varying dynamic conditions in the star map denoising task.

The advantage of DSR-Net lies primarily in its dynamic reinforcement learning strategy, which allows the model to learn noise characteristics under different dynamic environments and gradually optimize the denoising process. In contrast to traditional methods, DSR-Net not only emphasizes the local features of each pixel but also dynamically modifies its denoising strategy to more effectively align with the noise patterns and energy distribution of star points in the star maps. Especially under high noise complexity conditions, DSR-Net, through flexible strategy selection and efficient training, can effectively suppress noise interference and retain more star point details, ultimately achieving superior denoising performance.

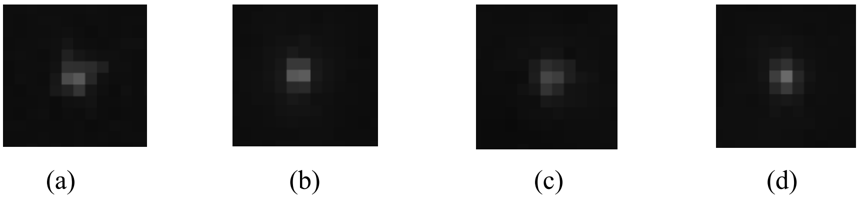

Fig. 2. Local star points of the star map under 1°/s dynamic after denoising by different methods: (a) Real star map local star points; (b) Local star points of the star map after denoising with GATFP [19]; (c) Local star points of the star map after denoising with Open Operation [20]; (d) Local star points of the star map after denoising with DSR-Net.

Figure 2 shows the local star point results of star maps after denoising using different methods under a 1°/s dynamic condition. As seen from the figure, DSR-Net is able to more accurately restore the energy distribution characteristics of the star points, particularly excelling in the edges and details of the stars, which significantly improves the denoised image quality. Compared to GATFP [19] and Open Operation [20], DSR-Net not only effectively suppresses noise interference during denoising but also preserves more star point details, resulting in clearer and more realistic local features of the star map. Overall, DSR-Net shows considerably superior denoising performance compared to the other two methods, proving its superior capability in complex dynamic conditions.

Ablation Experiment

Table 2. Arithmetic mean of evaluation metrics under three different dynamic conditions for the ablation experiment.

Net			Angular velocity 1.0 °/s		Angular velocity 2.0 °/s		Angular velocity 3.0 °/s	
A3C	+ConvGRU	+Rrc	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
√			44.387	0.9738	44.381	0.9738	44.317	0.9733
√	√		44.551	0.9744	44.442	0.9739	44.397	0.738
√	√	√	46.237	0.9803	46.139	0.9799	46.101	0.9786

Table 2 presents the ablation experiment results of the proposed method on different dynamic star map datasets. As shown in the table, the reinforcement learning model combining the ConvGRU and Rrc modules achieves the best performance in both SSIM and PSNR metrics, delivering the most effective denoising results. Specifically, the ConvGRU module, with its gating mechanism, effectively captures the spatiotemporal dynamics in the star maps, particularly maintaining superior denoising performance under varying observational conditions and the temporal nature of the star maps. The Rrc module, on the other hand, helps the model focus more on the noise characteristics in local regions through Region-based Reward Convolution, further enhancing the model's adaptability to complex noise environments. The experimental results show that integrating the ConvGRU and Rrc modules greatly enhances the model's performance in dynamic star map processing, playing a crucial role in capturing the spatiotemporal features of the star maps and handling complex noise. This design enables the model to more accurately process the noise in dynamic star maps, ultimately improving the denoising effect.

5 Conclusion

The proposed DSR-Net method is based on a reinforcement learning framework and has been innovatively designed for the star map denoising task. By combining ConvGRU and the Region-based Reward Convolution (Rrc) module, DSR-Net effectively handles the dynamic changes and complex noise distributions in star maps. The introduction of ConvGRU enhances the model's ability to process temporal features, while the Rrc module improves the learning of local features, significantly improving denoising performance in complex noise environments. DSR-Net has been evaluated on three real star map datasets with different dynamics, and the results show that DSR-Net exceeds existing methods in denoising performance, demonstrating its effectiveness and superiority in the star map denoising task.

References

[1] Sen S, Agarwal S, Chakraborty P, et al. Astronomical big data processing using machine learning: A comprehensive review[J]. Experimental Astronomy, 2022, 53(1): 1-43.

[2] Liu J, Zang H, Cheng L, et al. A Transformer-based multimodal-learning framework using sky images for ultra-short-term solar irradiance forecasting[J]. Applied Energy, 2023, 342: 121160.

[3] Jia P, Jia Q, Jiang T, et al. Observation strategy optimization for distributed telescope arrays with deep reinforcement learning[J]. The Astronomical Journal, 2023, 165(6): 233.

[4] Zhu H, Yi X, Li X, et al. ConvGRU-based Multi-scale Frequency Fusion Network for PAN-MS Joint Classification[J]. IEEE Transactions on Geoscience and Remote Sensing, 2024.

[5] Yu K, Dong C, Lin L, et al. Crafting a toolchain for image restoration by deep reinforcement learning[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 2443-2452.

[6] Uzkent B, Yeh C, Ermon S. Efficient object detection in large images using deep reinforcement learning[C]//Proceedings of the IEEE/CVF winter conference on applications of computer vision. 2020: 1824-1833.

[7] Yang D, Roth H, Xu Z, et al. Searching learning strategy with reinforcement learning for 3D medical image segmentation[C]//Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part II 22. Springer International Publishing, 2019: 3-11.

[8] Rao B S. Dynamic histogram equalization for contrast enhancement for digital images[J]. Applied Soft Computing, 2020, 89: 106114.

[9] Mafi M, Martin H, Cabrerizo M, et al. A comprehensive survey on impulse and Gaussian denoising filters for digital images[J]. Signal Processing, 2019, 157: 236-260.

[10] Chatterjee S, Goswami S. New risk bounds for 2d total variation denoising[J]. IEEE Transactions on Information Theory, 2021, 67(6): 4060-4091.

[11] Tian C, Xu Y, Fei L, et al. Enhanced CNN for image denoising[J]. CAAI Transactions on Intelligence Technology, 2019, 4(1): 17-23.

[12] Guo Y, Gao Y, Hu B, et al. CMID: Crossmodal Image Denoising via Pixel-Wise Deep Reinforcement Learning[J]. Sensors, 2023, 24(1): 42.

[13] Zhang R, Guo L, Huang S, et al. Rellie: Deep reinforcement learning for customized low-light image enhancement[C]//Proceedings of the 29th ACM international conference on multimedia. 2021: 2429-2437.

[14] Xie M, Zhang Z, Zheng W, et al. Multi-Frame Star Image Denoising Algorithm Based on Deep Reinforcement Learning and Mixed Poisson–Gaussian Likelihood[J]. Sensors, 2020, 20(21): 5983.

[15] Satish V, Mahler J, Goldberg K. On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks[J]. IEEE Robotics and Automation Letters, 2019, 4(2): 1357-1364.

[16] Hou Y, Xu J, Liu M, et al. NLH: A blind pixel-level non-local method for real-world image denoising[J]. IEEE Transactions on Image Processing, 2020, 29: 5121-5135.

[17] Wang X, Wang S, Liang X, et al. Deep reinforcement learning: A survey[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 35(4): 5064-5078.

[18] He W, Xiong T, Wang H, et al. Radar echo spatiotemporal sequence prediction using an improved ConvGRU deep learning model[J]. Atmosphere, 2022, 13(1): 88.

[19] Zhang Y, Song P, Dai Q. Fourier ptychographic microscopy using a generalized Anscombe transform approximation of the mixed Poisson-Gaussian likelihood[J]. Optics express, 2017, 25(1): 168-179.

[20] Jiang H, Fan X Y. Centroid locating for star image object by FPGA[J]. Advanced Materials Research, 2012, 403: 1379-1383.