Soft Computing (2022) 26:3261–3272
https://doi.org/10.1007/s00500-022-06840-3

DATA ANALYTICS AND MACHINE LEARNING

Boosting denoisers with reinforcement learning for image restoration

Jie Zhang[1,2] · Qiyuan Zhang[3] · Xixuan Zhao[1,2] · Jiangming Kan[1,2]

Accepted: 21 January 2022 / Published online: 20 February 2022

Abstract
Learning-based image restoration approaches typically learn to map distorted images to clean images. To remove multiple combined distortions with unknown mixture ratios, most of the existing methods have focused on the development of different deep neural network architectures and novel loss functions. Although these methods have proved their effectiveness on image restoration tasks, they require expensive training data and produce results in a noninterpretable way. In this work, we present a deep reinforcement learning (DRL) based method to restore the distorted images, which casts an image restoration Problem as a Partially Observable Markov Decision Process (POMDP) where actions are defined as multiple pixel-wise image denoising operations. In our method, each agent possesses a pixel, the agent learns to adjust the corresponding pixel value by determining the proper combination of the actions. We also develop a novel exploration scheme such that similar actions have similar value, thereby avoiding overfitting in state-action value estimation. Through extensive experiments, we show that our method can restore images with multiple combined distortions and our DRL approach performs comparable or better performance against previous learning-based approaches. By visualizing the process of weighting multiple pixel-wise operations, we can identify what combination of operations is employed for each pixel at each stage. We believe our work takes a step toward the explainability and interpretability of learning-based image restoration methods.

1 Introduction

Image restoration has always been a hot topic in computer vision. Both the traditional filtering methods and the deep learning algorithms, which have attracted much attention in recent years, have achieved high achievements in image restoration (Dabov et al. 2007; Buades et al. 2005; Rudin et al. 1992; Chen et al. 2015a; Burger et al. 2012). Due to the development of neural networks, deep learning has not only made great success in image recognition and detection but also achieved remarkable achievements in low-level tasks, such as image denoising and image enhancement. However, image restoration methods, which are based on deep learning, often train a single and large network, which requires a large amount of training data, also with a large number of parameters, so it makes these methods computationally expensive and consumes resources. We can't help asking, since large neural networks are relatively complex, can we combine simple networks or traditional filtering algorithms with general restoration effects into an algorithm with strong recovery effects through a certain method, and using a small amount of data and calculation to realize image restoration.

Ensemble learning (Polikar 2012) provides ideas for us. In ensemble learning, the weak classifiers can be integrated into a strong classifier by boosting. Therefore, in image restoration, it might be feasible to combine multiple algorithms which are weak restoration performance into an algorithm with excellent restoration performance. Coincidentally, there are few recent articles using deep reinforcement learning to do this kind of work.

Yu et al. (2018) tried this idea for the first time. They perform a method called RL-Restore. In their previous

✉ Xixuan Zhao
zhaoxixuan@bjfu.edu.cn

Jie Zhang
jiezhang@bjfu.edu.cn

Qiyuan Zhang
zhangqiyuan19@hit.edu.cn

Jiangming Kan
kanjm@bjfu.edu.cn

[1] School of Technology, Beijing Forestry University, No. 35 Tsinghua East Road, Haidian District, Beijing 100083, China

[2] Key Laboratory of State Forestry Administration on Forestry Equipment and Automation, No. 35 Tsinghua East Road, Haidian District, Beijing 100083, China

[3] School of Mechatronics Engineering, Harbin Institute of Technology, Xi Da Zhi Jie, Nangang District, Harbin 150001, China

experiments, they found that for a contaminated picture with multiple noises, even if the type of pollution is known, the order of the denoise methods used affects the quality of the final restoration greatly. This is an exciting discovery. Through this discovery, they convert the restoration problem of multiple distortion images into an MDP problem. They construct a toolbox that contains multiple small and simple denoising networks and use deep reinforcement learning to decide the optimal order of small neural networks using. In their experiments, they found that the restoration effect of their method is better than that of the large-scale neural network, and the parameters of it are far less than those based on deep learning. However, Suganuma et al. (2019) proved that although the image restored by the RL-Restore method (Yu et al. 2018) has a relatively good restoration effect, The accuracy of its recognition will be greatly reduced in the subsequent recognition task, which obviously should not be. And an article (Xie et al. 2019) explains the reason why the accuracy of the image restored by the neural network is declined during recognition at pixel-wise.

Furuta et al. (2019) proposed an RL-based image restoration method at pixel-wise. Similar to the above, They also used a toolbox, but it contains a variety of traditional filtering algorithms. They modeled the problem as a MARL problem, that is, each pixel is regarded as an agent, and the value of each pixel is changed by using the filtering algorithms which are determined by the policy of deep reinforcement learning, so as to achieve image restoration in pixel-wise. However, this method only aims at the restoration of a single noisy image, and the A3C algorithm used in this method is an on-policy RL algorithm, so the sample is inefficient.

In addition, the above methods are all performed in the discrete action space, that is, only one denoiser is used in each step of the processing. The multi-noise image contains a variety of noises. If only one denoiser is used in each step, it will inevitably not be able to complete the restoration task in a few time steps. But if the step is too long, using the RL algorithm to solve the problem seems meaningless. Moreover, a pixel is not independent. Whether using a small denoising network or a traditional denoising method, a change in one pixel will inevitably cause changes in the surrounding neighboring pixels of it, which add a new challenge to the restoration of multi-noise images.

The method we proposed in this paper examines the problem from a new perspective for image denoising at the pixel-wise, that is, there is a certain connection between the pixel value of a damaged image after filtering and the pixel value of the real image. For a certain pixel of the real image, its pixel value may be similar to the pixel value of the image after filtering by a certain type, or it may be similar to the pixel value synthesized by weight after multiple filtering processing. Therefore, from this point of view, our method changes the pixel value in a way of weight synthesis. Our method performs a variety of traditional filtering operations on noisy image synchronously, then use a deep reinforcement learning algorithm to learn a policy which gives each pixel a group of weights of filters, and uses these weights to fuse images (which processed by traditional denoisers) into a clear image. We set the weight to the action of our policy, thus our agent will be in the continuous action space. In addition, by assigning weights to each filtered image and synthesizing a clear image according to the weights, a pixel can be changed without affecting surrounding pixels, and the coupling problem between adjacent pixels can be avoided.

We model our task as a POMDP problem and use the policy gradient RL method to solve the task of continuous action space. In addition, each pixel requires to change, we define each pixel as an agent, so our problem will be transformed into a MARL problem under POMDP.

The main framework of this method is shown in Fig. 1, it consists of two parts: One is a toolbox that contains several

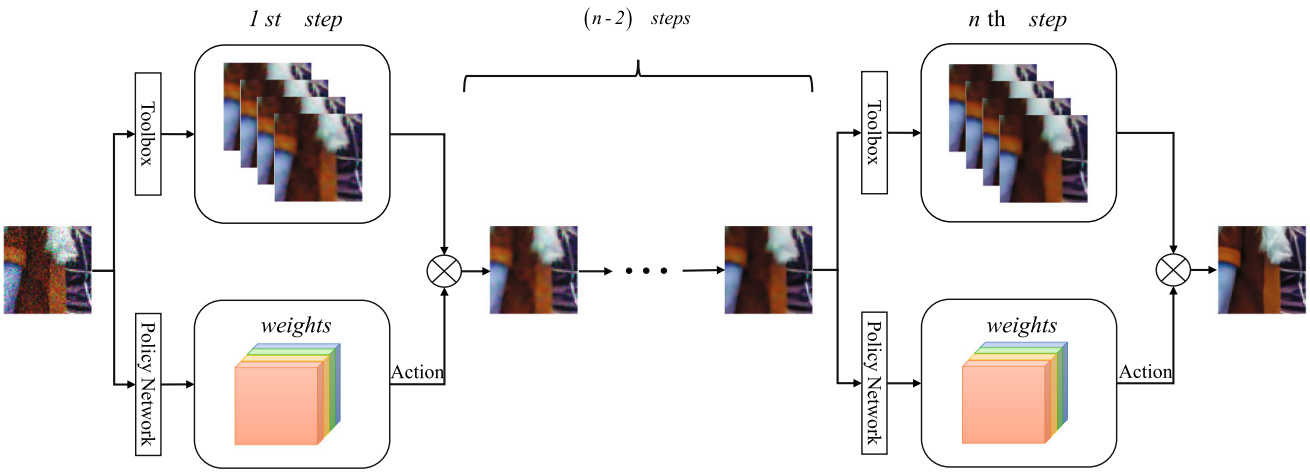

Fig. 1 Illustration of our method

traditional image filters. Another is an agent that dynamically chooses the action which changes the weights of each pixel at each step for a group of filters. The main contributions of this paper are:

- We redefine the restoration task of multi-noise images as a MARL problem under POMDP and proposes an integrated denoising method.
- We propose a multi-noise image restoration method at pixel-wise in the continuous action space.
- We solve the problems of the deterministic policy gradient method in the continuous action space, which caused by insufficient state information under POMDP.

2 Related work

2.1 Multiple degradation for a single image

It is not uncommon to use traditional filtering methods and CNN-based methods to process a single image contaminated by a single pollution source. Whether it is a single processing task for various single images such as denoising, anti-artifacts, or color enhancement, the effects of traditional filtering methods and methods based on deep learning are also obvious to all. However, for the contaminated images in the real environment, the pollution is often not from a single source. For such tasks, traditional filtering methods appear to be inadequate. CNN-based methods, such as Kim et al. (2016) used a 20-layer neural network and proposed a VDSR multi-scale single image super-resolution reconstruction method. Zhang et al. (2017) proposed a 20-layer CNN network that can handle multiple recovery tasks at the same time. However, these CNN-based processing methods and Guo and Chao (2017), etc., do not consider the problem of mixed pollution, that is, the situation where a single image contains multiple losses at the same time. In addition, since a large-scale neural network is required to process complex tasks, its network parameters are many and the calculations are more complicated. Although methods such as Chen et al. (2015b) and Han et al. (2015) can compress large networks, there are still many parameters after compression for the neural networks needs to perform a lot of recursive operations.

2.2 Deep reinforcement learning for image processing

Deep reinforcement learning algorithms have also achieved success in some image processing fields. Park et al. (2018) proposed an image color enhancement method based on deep reinforcement learning. They convert the image color enhancement process into an MDP process, and define the output action as a global color enhancement operation, then use the deep reinforcement learning algorithm to learn the best global enhanced action sequence. Cao et al. (2017) took advantage of reinforcement learning and proposed a super-resolution reconstruction method of Attention-aware Face Hallucination. Li et al. (2018) applied the deep reinforcement learning method to the image cropping task. This method formulates the image cropping task as a sequential decision, and proposed an Aesthetics Aware Reinforcement Learning (A2-RL) framework to solve the aesthetic problem in image cropping. Li et al. (2020) combines deep reinforcement learning at pixel-wise to achieve MRI image reconstruction. Li and Zhang (2019) proposed an automatic thumbnail generation method based on deep reinforcement learning. Liao et al. (2020) implemented an image segmentation task based on reinforcement learning and cross-entropy at pixel-wise.

2.3 POMDP

The real environment is often not fully observable. For the agent, the state it can observe is generally limited. Partially Observable Markov Decision Process (POMDP) is a general Markov decision process. In the POMDP model, the agent must make use of the limited information in the environment to make decisions, but the observed information is incomplete, so in practice, POMDP is usually computationally difficult to solve. Using value iteration to solve is a method of approximately solving POMDP (White and Scherer 1989), but these methods will turn the complexity of the entire problem into an exponential function based on the value iteration algorithm, which may cause a dimensional explosion. Therefore, methods such as (Koller and Parr 2013; Guestrin et al. 2001) decompose the entire problem to reduce the scale of the problem. In addition, in recent years, the use of learning-based algorithms to solve them has also achieved good results (Bertsekas and Tsitsiklis 1995; Lin and Mitchell 1992), especially the emergence of RNN, which makes it possible for agents to make decisions based on historical information.

2.4 MARL

Multi-Agent Reinforcement Learning (MARL) (Tan 1993) is an important branch of the multi-agent system. There are at least two agents in the multi-agent reinforcement learning system. Unlike the single-agent reinforcement learning, each agent is not only affected by the environment but also affected by other agents. MARL is used to solve the sequential decision-making problem of multiple agents in the same environment. Each agent needs to interact with the environment and other agents to make it achieve more rewards (Lowe et al. 2017). Compared with the single-agent system, the multi-agent system has the following characteristics: (1) the state transition of the multi-agent system depends on the actions of all agents. (2) In a multi-agent system, the rewards

received by each agent are not only related to its actions, but also related to the actions of other agents. Due to the above two characteristics, the task of solving multi-agent systems is more complicated and difficult. Generally speaking, multi-agent reinforcement learning algorithms are mainly divided into three categories: full cooperation, full competition, and combining them for different application tasks (Yang et al. 2018). The basic algorithms for solving multi-agent reinforcement learning include MiniMax-Q (Littman 1994), NashQ (Singsanga et al. 2010), FFQ (Littman 2001), WoLF-PHC (Bowling and Veloso 2001), and other mainstream methods include MADDPG (Rashid et al. 2018), QMIX, MFMRL (Buşoniu et al. 2010) and so on.

3 Problem statement

A human expert removes multiple combined distortions by applying a set of image denoising operations. To imitate this process, we formulate image restoration as a problem of finding an optimal operation combination of denoising actions.

Let I_t^i be the i-th pixel of the modified image $I_t \in \mathbb{R}^{H\times W\times C}$ that has N pixels ($i = 1, \cdots, N$). Here, $N := H \cdot W$, I_0 is denoted as the original distorted image, H, W and C are its height, width and the number of channels, respectively.Since different areas of the image may be distorted by multiple noises, it is necessary to restore the image at the pixel level. Each pixel corresponds to an agent, each agent $a \in A \equiv \{1, \cdots, N\}$ receives the local observation $o_t^a \in \mathcal{O}$ provided by an observation function $O(I_t, a)$ and takes an action $u_t^a \in U$ according to a stationary policy $\pi^a(u_t^a|o_t^a)$. Here, the action $\mathbf{u}_t^a \in \mathbb{R}^M$ denotes the attention weights on the pixel-wise outputs of the toolbox containing M parallel image denoising operations, U is a probability simplex. After adjusting its corresponding pixel value I_t^a, each agent obtains a reward r_t^a that measures how much the modified pixel value I_{t+1}^a has improved compared to the previous one. Given the input image I_t and the joint action $\mathbf{u}_t := \left[u_t^1, \ldots, u_t^N\right]$ at time step t, the environment change to the next state I_{t+1} according to the state transition probability $P(I(t+1)|I(t), \mathbf{u}_t)$. All agents work together to enhance the image in an iterative way, and terminate this process when the maximum time step T is achieved.

The goal of the RL-based image restoration problem is to learn the optimal joint policies $\boldsymbol{ß} = \left(\pi^1, \ldots, \pi^N\right)$ that maximize the mean of the total expected rewards at all pixels:

$$\max_{\boldsymbol{ß}} \mathbb{E}_{\tau\sim p_{\boldsymbol{ß}}(\tau)}\left[\sum_{t=0}^{T}\gamma^t\frac{1}{N}\sum_{a=1}^{N}r_t^a\right]$$
$$\text{s.t.}\quad \sum_{k=1}^{M}u_t^{a,k}=1,\quad u_t^a\sim\pi^a\left(u_t^a|o_t^a\right),\quad a=1,\ldots,N. \tag{1}$$

where γ is the discounted factor, the trajectory $\tau := \left\{o_0^1, u_0^1, \ldots, o_0^N, u_0^N, \ldots, o_T^1, u_T^1, \ldots, o_T^N, u_T^N\right\}$. the induced trajectory distribution $p_{\boldsymbol{\pi}}(\tau)$ is given by

$$p_{\boldsymbol{\pi}}(\tau) = p(I_0)\prod_{t=0}^{T}\left(\prod_{a=1}^{N}\pi\left(u_t^a|o_t\right)\right) P\left(I(t+1)|I(t), \mathbf{u}_t\right), \tag{2}$$

The common approach is to divide this decision-making problem into N independent subproblems and train N networks, where the i-th policy learns to maximize the expected discounted cumulative rewards at the i-th pixel $J\left(\pi^i\right) = \mathbb{E}_{\tau\sim p_{\boldsymbol{ß}}(\tau)}\left[\sum_{t=0}^{T}\gamma^t r_t^i\right]$. However, this method is not suitable for situations where the size of an input image in the test is different from the one in the training, and it becomes computationally impractical as the number of agents increases to thousands. Moreover, since the pixel-wise outputs of the toolbox are invisible to each agent, the policy $\pi(\cdot|o_t)$ leads to the poor performance of image restoration in this partially observable setting. In the next section, we propose a sample efficient and computationally tractable RL method to solve these issues.

4 Learning to restore the distorted images

In RL based pixel-wise image restoration, the input information o_t^i each agent i receives at step t consists of the pixel value I_t^i and its neighborhood pixels provided by a observation function $O(I_t, i)$, based on which the corresponding policy performs inference. The field-of-view of the observation function has an important influence on restoring images, the small field contains little useful information, but the large field will include redundant observation that is useless to the i-th agent, leading to high computational burden. Rather than designing the observation function in a hand-crafted way, we use convolutional blocks to provide the agent with its neighborhood information. Another advantage of convolutional blocks is that all the N agents can share their parameter, leading to the high-efficient computation.

Partial observability arises from two sources including a restricted field-of-view and the invisibility of the output of the toolbox to all agents. Each agent i should learn to form memories based on interactions with the environment to handle partially observed problems, thus the optimal policy of agents in principle require to access to the historical experience $h_t = \{I_0, \mathbf{u}_0, \ldots, I_{t-1}, \mathbf{u}_{t-1}, I_t\}$. Here, we use Gate Recurrent Unit (GRU) network to effectivelyextract this his-

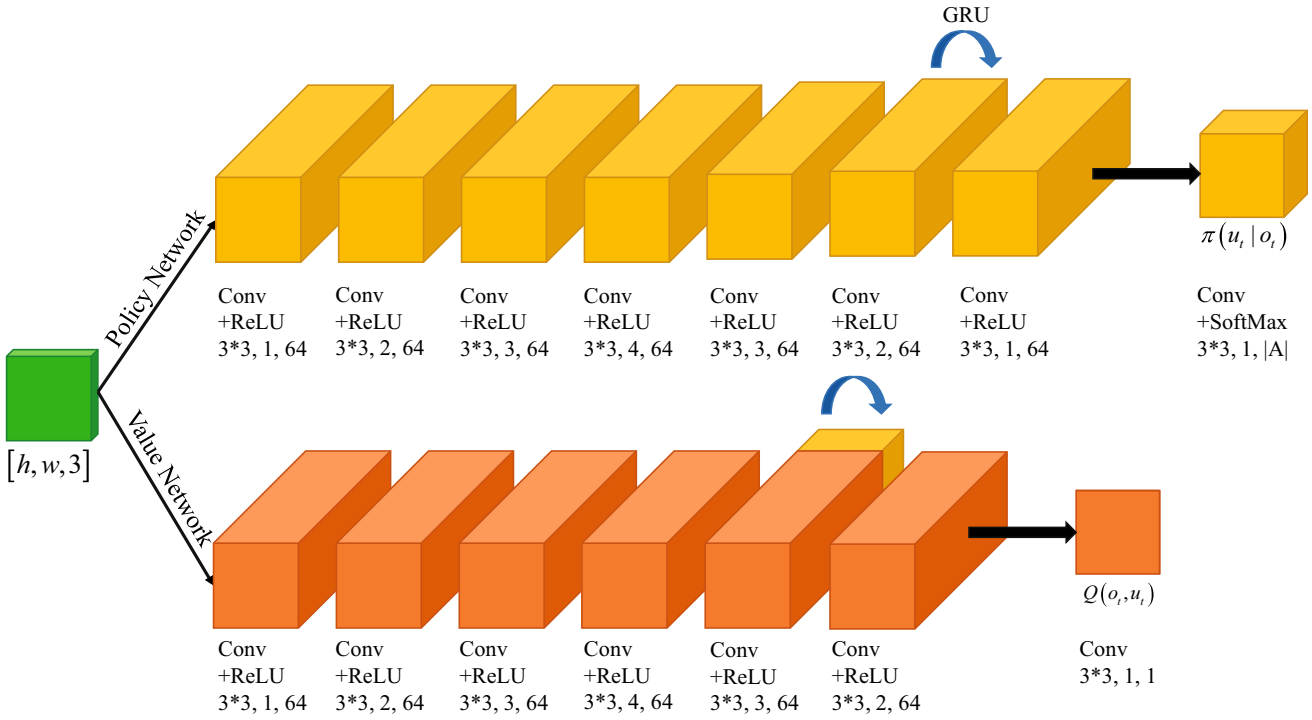

Fig. 2 The network architecture of our method

torical information in their recurrent state, which is given by

$$h_t = \mathrm{GRU}\left(h_{t-1}, I_t, \mathbf{u}_{t-1}\right), \tag{3}$$

where h_{-1} and $\mathbf{u}_{-1}$ are the zero start state. The attention weights $\mathbf{u}_t \in \mathbb{R}^{N\times M}$ on the pixel-wise output of the toolbox are calculated as follow

$$\begin{aligned} u_t^{i,m} &= \frac{\exp\left(\mathbf{z}_t^{i,m}\right)}{\sum_{k=1}^{M}\exp\left(\mathbf{z}_t^{i,k}\right)}, \\ \mathbf{z}_t &= \mathcal{F}\left(h_t\right), \quad i = 1, \ldots, N. \end{aligned} \tag{4}$$

where $\mathcal{F}(\cdot)$ is the convolution operator. The modified image at step $t+1$ is given by

$$I_{t+1}^{i} = \sum_{m=1}^{M} u_t^{i,m}\bar{I}_t^{m,i}, \quad \bar{I}_t^{m} = g_m\left(I_t\right) \in \mathbb{R}^{N\times C}, \tag{5}$$

where $g_m(\cdot)$ is the m-th image denoising operation in the toolbox. Therefore the entire architecture of the policy network in Fig. 2 includes three modules: it uses convolutional blocks to learn low-level features. The GRU block combines these features with historical information extracted from past experience to learn high-level features, based on which all agent make decisions.

A major challenge of training deterministic policies in image restoration is exploration. One choice to improve the exploration ability of policies is to construct an exploration policy, which is represented by a Gaussian noise source and a deterministic neural network that transform a draw from that noise source, i.e., $\mathbf{u}_t = \mathbf{ß}_\phi(h_t, \varepsilon)$ with $\varepsilon \sim \mathcal{N}(0, \delta^2)$. Specially, the modified action of the i-th agent is given by

$$\begin{aligned} \tilde{u}^{i,m} &= \frac{\exp\left(\mathbf{z}^{i,m} + \varepsilon^m\right)}{\sum_{k=1}^{M}\exp\left(\mathbf{z}^{i,k} + \varepsilon^k\right)}, \\ \varepsilon_k &\sim \mathrm{clip}\left(\mathcal{N}\left(0, \sigma^2\right), -c, c\right). \end{aligned} \tag{6}$$

where the added noise is clipped to keep the modified action close to the original one. Further, we can easily obtain that

$$\frac{\exp\left(\mathbf{z}^{i,m} - c\right)}{\sum_{k=1}^{M}\exp\left(\mathbf{z}^{i,k} + c\right)} < \tilde{u}^{i,m} < \frac{\exp\left(\mathbf{z}^{i,m} + c\right)}{\sum_{k=1}^{M}\exp\left(\mathbf{z}^{i,k} - c\right)}, \tag{7}$$

then the modified action $\tilde{u}^{i,m}$ is a random variable with support in $\left(\exp\left(-2c\right)u^{i,m}, \exp\left(2c\right)u^{i,m}\right)$.

Similar to TD3 algorithm, we use parameterized function approximators for both the Q-function Q_θ and policy π_ϕ, and then alternatively performs policy evaluation and policy improvement.

$$J_Q\left(\theta_s\right) = \mathbb{E}\left[N^{-1}\sum_{i=1}^{N}\left(Q_{\theta_s}\left(h_t^i, u_t^i\right)\Big|_{u_t^i=\pi_\phi\left(h_t^i\right)} - y_t^i\right)^2\right],$$

$$y_t^i = r_t^i + \gamma \min_{s=1,2} Q_{\bar{\theta}_s}\left(h_{t+1}^i, \pi_{\bar{\phi}}\left(h_{t+1}^i, \varepsilon\right)\right),$$
$$\nabla_\phi J(\phi) = \mathbb{E}\left[N^{-1}\sum_{i=1}^{N} \nabla_{u_t^i} Q_{\theta_1}\left(h_t^i, u_t^i\right)\Big|_{u_t^i=\pi_\phi(h_t^i)} \nabla_\phi \pi_\phi\left(h_t^i\right)\right]. \tag{8}$$

where $\bar{\phi}$ and $\bar{\theta}_s$, $s = 1, 2$ are delayed parameters, and fitting the value of the modified action can alleviate the narrow peak of overfitting to the value estimation, decreasing the variance of the target Q. The pseudo code of our algorithm is shown in the Algorithm 1.

5 Experiment

5.1 Toolbox

Similar to Furuta et al. 2019, our experiment also uses a toolbox that contains multiple traditional filters for image denoising. In order to compare with Pixel-RL (Furuta et al. 2019) fairly, the toolbox designed in our experiment is the same as Furuta et al. (2019) except for the "do-nothing" operation. Since the method of our experiment is to use a variety of traditional denoising algorithms, which combined with deep reinforcement learning and tries to integrate a variety of weak filters into a strong filter to achieve denoising driven by knowledge and data, the "do-nothing" operation is meaningless for our experiment, so it is removed. The traditional filters and their parameters in our toolbox are shown in Table 1:

5.2 Reward

The goal of reinforcement learning is to obtain the largest cumulative reward, and the reward determines the quality of the policy adopted by the agent. In this paper, the image quality of each step is used to determine the reward. The reward is defined by $r_t = P_{t+1} - P_t$, where P_{t+1} is the peak signal-to-noise ratio (PSNR) value between the image processed in the tth step and the real image. Therefore, the cumulative reward is defined as $R_{ij}|_{i\in(0,\ h),\, j\in(0,\ w)} = \sum_{t=0}^{T-1} r_t = P_T - P_0$, which indicates that the cumulative reward is related to the PSNR value of the last processed image and the PSNR value of the initial state image.

Table 1 Tools in toolbox

Tools	Filter size	Parameters
Gaussian filter	5*5	$\sigma = 0.5$
Gaussian filter	5*5	$\sigma = 1.5$
Bilateral filter	5*5	$\sigma_c = 0.1,\ \sigma_s = 5$
Bilateral filter	5*5	$\sigma_c = 1.0,\ \sigma_s = 5$
Median filter	5*5	–
Box filter	5*5	–
Pixel value − = 1	–	–
Pixel value + = 1	–	–

5.3 Image restoration dataset

We use the same dataset—BSD68 dataset (Mairal et al. 2007)—as in Furuta et al. (2019), and preprocess the dataset in the following operation. Firstly, the dataset is down-sampled, and then the sampled image is divided into 63*63 sub-images. In our experiment, 3584 images were generated for the dataset as the ground truth of the training data. The two most common noises in the original images are Gaussian noise and Poisson noise. Therefore, we randomly added Gaussian noise and Poisson noise in random proportion to the ground truth dataset to form the noise dataset. This operation ensures the authenticity of the training data since the ratio of Gaussian to Poisson noise added to each image is random.

Also, we used the DIV2K dataset (Agustsson and Timofte 2017) which has been preprocessed in Yu et al. (2018) as the ground truth of the training data. We generated two sets of noise images for this dataset, one refers to Yu et al. (2018) and generates the DIV2K-Mild dataset, and the other dataset, the DIV2K-Mixed dataset, is generated using the same processing with the BSD68 dataset. All of them have 3584 images.

The above three sets of datasets are all generated by artificially adding noise. To verify the effect of our method on real noise images obtained by different camera parameters, we also use the Mi3 dataset in the RENOIR dataset (Anaya and Barbu 2018). The RENOIR data set is established by taking a low ISO image as ground truth and a high ISO image as a noise image for the same scene, and adjusting camera parameters, such as exposure time, to make the brightness of the two images consistent. In our experiment, we select the first 40 scenes of the Mi3 dataset in RENOIR as the experimental data, and each scene contains 2 high-noise images and 2 low-noise images. We select a low-noise image in each scene as a ground-truth image, the images with different ISO parameters as noisy images, and use the aforementioned method to process 3584 images as well. Thus, two sets of ISO noise datasets, the RENOIR-Low dataset and RENOIR-High dataset, are obtained.

In order to reveal the effect of each step in our method, Figs. 3 and 4 show the results of each step of the test images on the DIV2K dataset and the BSD68 dataset. It can be seen that the image of each step is greatly improved, which means our method can restore the noise image efficiently.

We use PSNR (Peak Signal to Noise Ratio) and SSIM (Structural Similarity) to evaluate the image quality in our

Algorithm 1

Initialize critic networks Q_{θ_1}, Q_{θ_2} and actor network π_ϕ with random parameters θ_1, θ_2, ϕ
Initialize target networks $\bar{\theta}_1 \leftarrow \theta_1, \bar{\theta}_2 \leftarrow \theta_2, \bar{\phi} \leftarrow \phi$
Initialize replay buffer $\mathcal{B}$
for episodes=1, M **do**
  Initialize empty history h_0 and action u_0.
  Receive the orignal distored image I_0.
  for t = 1, T **do**
    $h_t \leftarrow I_t,\ \mathbf{u}_{t-1},\ h_{t-1}$
    Select raw action with exploration noise $\mathbf{z}_t = \pi_\phi(h_t) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2)$.
    Make the $action \in (0,\ 1)$, $\mathbf{u}_t = \text{softmax}(\mathbf{z}_t)$.
  end for
  Store transition sequence $(I_0,\ \mathbf{u}_1,\ r_1,\ I_1,\ \cdots,\ I_T)$ in $\mathcal{B}$.
  Sample a mini-batch of N episodes $I_0^i,\ \mathbf{u}_1^i,\ r_1^i,\ I_1^i,\ \cdots,\ I_T^i$ from $\mathcal{B}$.
  Construct histories $h_t^i = (o_1^i,\ u_1^i,\ \cdots,\ u_{t-1}^i,\ o_t^i)$.
  Compute target values for each sample episode $y_1^i,\ \cdots,\ y_T^i$ using recurrent target networks
  $y_t^i = r_t^i + \gamma min_{s=1,2} Q_{\theta_s'}\left(h_{t+1}^i, \pi_{\bar{\phi}}\left(h_{t+1}^i\right)\right)$
  Update critics $\theta_{s=1,2} \leftarrow min_{\theta_s} N^{-1} \sum \left(y_t^i - Q_{\theta_s}\left(h_t^i,\ u_t^i\right)|_{u_t^i=\pi_\phi(h_t^i)}\right)$
  if t mod d **then**
    Update ϕ by the deterministic policy gradient:
    $\nabla_\phi J(\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}\left(h_t^i,\ u_t^i\right)|_{u_t^i=\pi_\phi(h_t^i)} \nabla_\phi \pi_\phi(h_t^i)$
  end if
end for

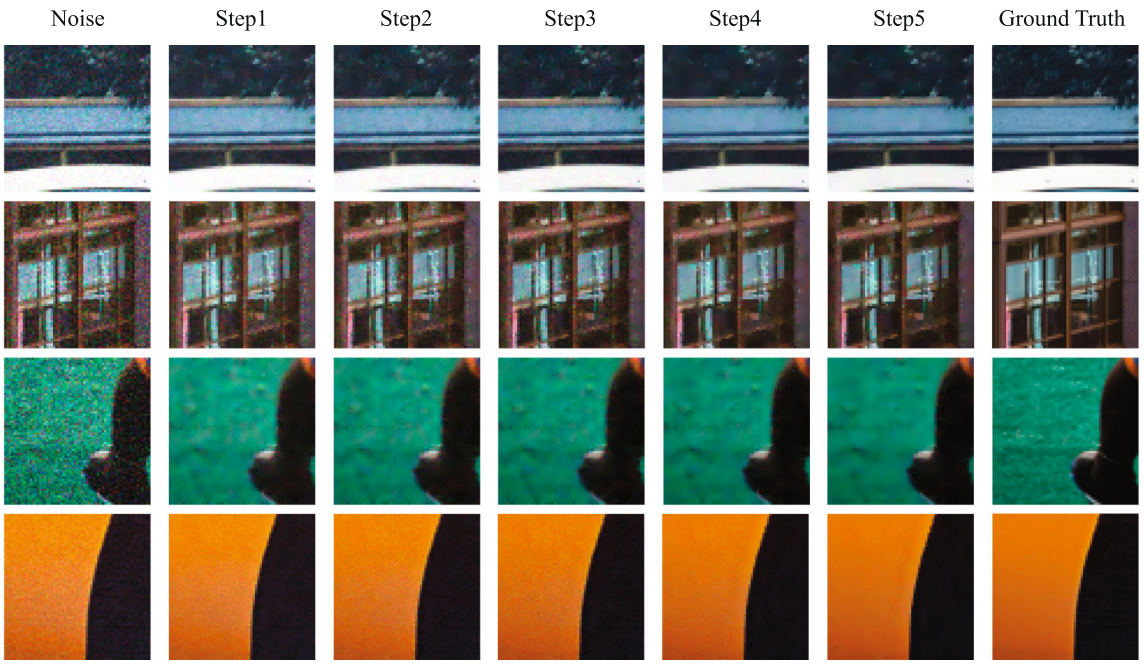

Fig. 3 The test result of DIV2K dataset in each step

experiments. First, we compare the image results and the PSNR of our method with the Pixel-RL method in the test images of the DIV2K dataset and BSD68 dataset, which shows in Figs. 5 and 6. The results show that our method has stronger effects and clearer details than Pixel-RL.

We also compare our method with the Pixel-RL method on 5 sets of datasets. The evaluation indicators are the average PSNR value and SSIM value during the training processing. To clarify the advantage of pixel-wise image denoising operation, we compare the proposed method with a baseline that uses one filter randomly sampled from the toolbox to restore distorted images at each step. Since multiple combined distortions are introduced into the images, this baseline performs worse than a pixel-wise combination of denoising actions obtained from our random policy. In Pixel-RL, in order to further improve the recovery effect, the RMC (Reward Map

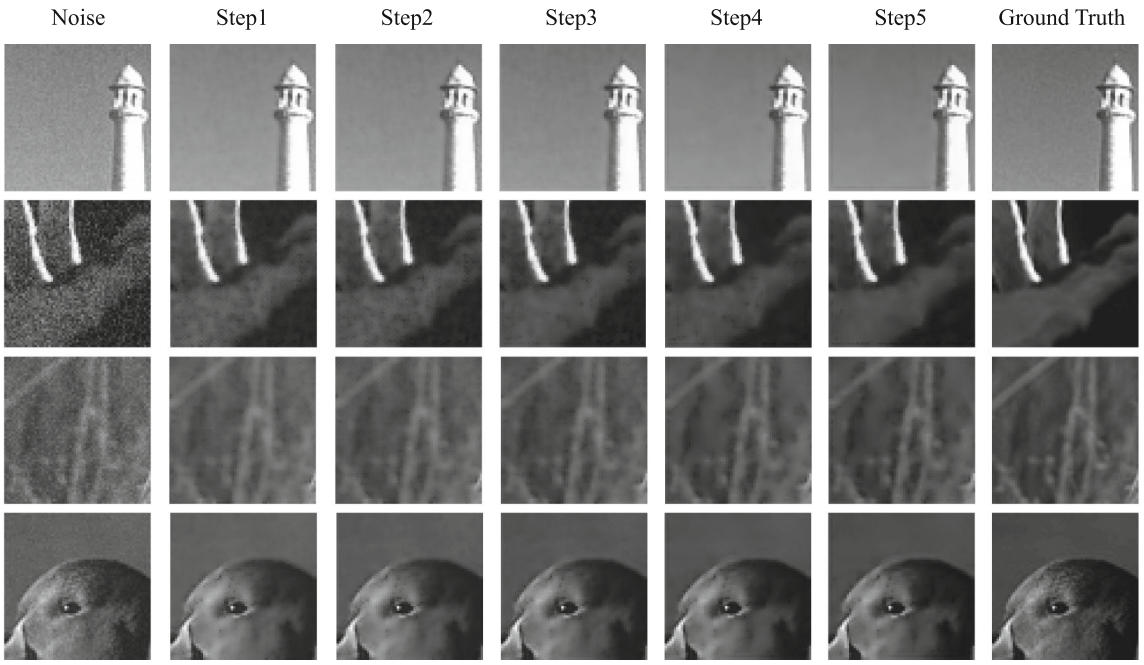

Fig. 4 The test result of BSD68 dataset in each step

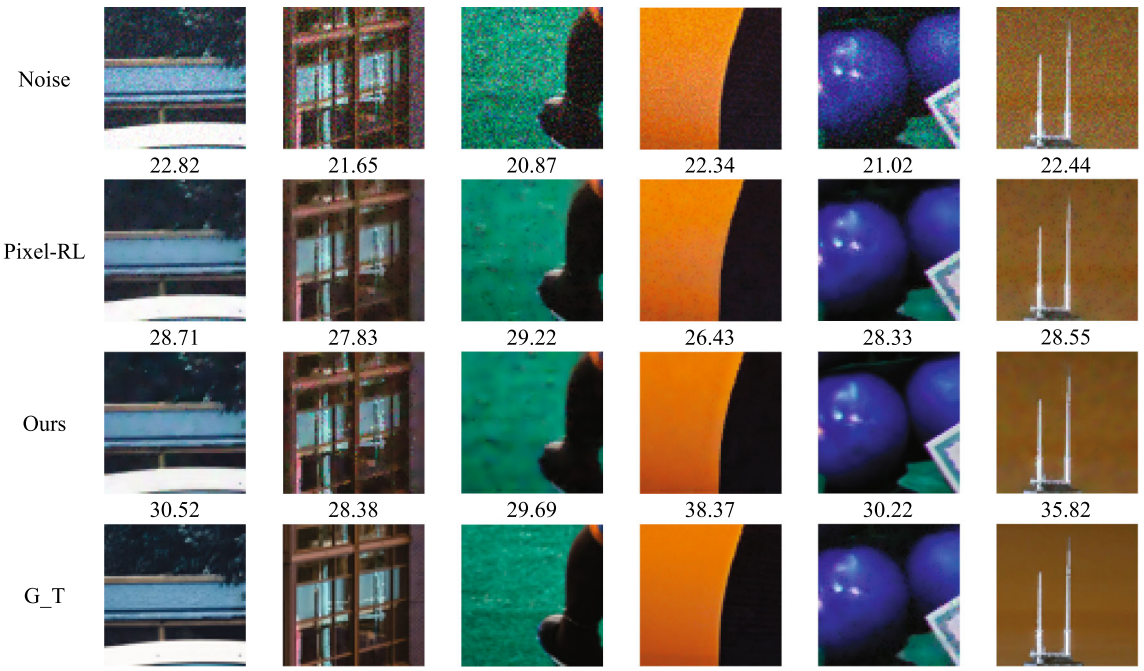

Fig. 5 The result of comparison in DIV2K dataset

Convolution) method is added by the authors. Therefore, we compare our method with the two Pixel-RL methods (Pixel-RL-w/o- RMC and Pixel-RL-RMC). In addition, in order to explore the impact of adding noise to the policy on our method, we regard our method which not adding noise as an ablation experiment. The experimental results are shown in Table 2:

The hyperparameters of this experiment are specifically set as follows: the maximum step size of training is 1e6, and the size of batch-size is 6. During agent training, the learning rate is 7e−4, the maximum step = 5, the optimizer uses Adam optimizer (Kingma and Ba 2014), buffer-size = 1e5, and the update frequency $C = 1000$.

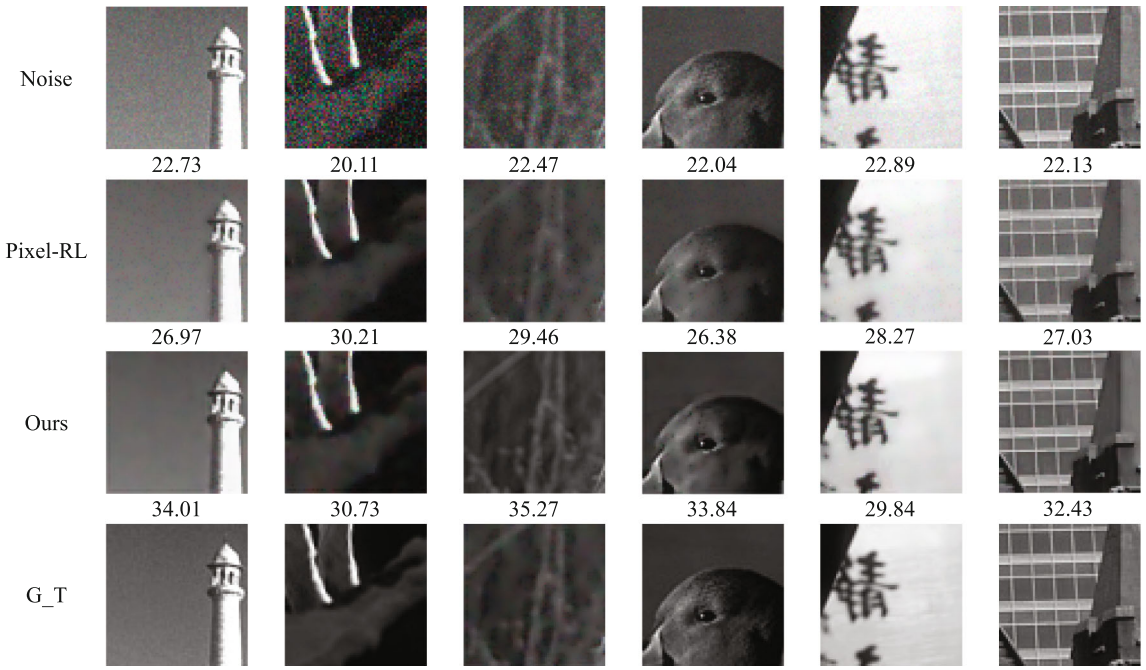

Fig. 6 The result of comparison in BSD68 dataset

Table 2 The experimental results of five training datasets

Datasets	BSD68		DIV2K-Mild		DIV2K-Mixed		RENOIR-Low		RENOIR-High	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Raw	19.64	0.688	23.85	0.660	19.68	0.679	26.54	0.878	23.66	0.810
Baseline	22.81	0.755	24.49	0.722	22.63	0.739	30.02	0.830	28.42	0.796
Random	23.19	0.781	25.59	0.749	22.97	0.777	32.12	0.967	29.89	0.962
Pixel-RL-w/o-RMC	28.59	0.905	27.29	0.804	28.02	0.889	**33.87**	0.919	32.17	0.900
Pixel-RL-RMC	28.48	0.900	**27.39**	**0.806**	28.06	0.888	33.74	0.918	**32.48**	0.906
Ours-w/o-noise	29.11	0.904	26.44	0.789	28.22	0.909	32.98	0.966	**31.27**	0.961
Ours	**29.81**	**0.910**	26.50	0.799	**29.48**	**0.921**	**32.98**	**0.970**	31.22	**0.964**

Best results of different algorithms on the dataset

It can be seen from the experimental results in Table 2 that our method greatly improves the image quality compared with the original PSNR and SSIM values of the noise image. First, the comparison with the baseline verifies the necessity of pixel-level restoration. Then, using the same BSD68 dataset as in Buades et al. (2005), our method outperforms the Pixel-RL method in both PSNR and SSIM indicators. Similarly, in the DIV2K-Mixed dataset with artificially added mixed noise, our method is also superior to the Pixel-RL method in these two indicators. Except that the SSIM of the DIV2K-Mild dataset of our method is slightly lower than the Pixel-RL method, the SSIM of the other four groups is better than the Pixel-RL algorithm, that is, our method is superior in terms of guaranteeing image similarity (SSIM) than Pixel-RL. Although the average PSNR value of the other three data sets is slightly lower than that of the Pixel-RL algorithm, it can also be seen from the comparison result with Random policy that the method proposed in this paper has sufficient advantages in terms of integrated denoiser. In addition, the ablation experiment without adding noise shows that the method of adding noise to the policy increases the agent's exploration ability while significantly improving the quality of image denoising.

Table 3 The test efficiency of Pixel-RL and our method

Algorithm	Params (M)	Time (s)
Pixel-RL	1.78	0.036
Ours	1.81	0.040

As Pixel-RL needs to load a pre-trained model during the training process., when the pre-trained model is removed, the

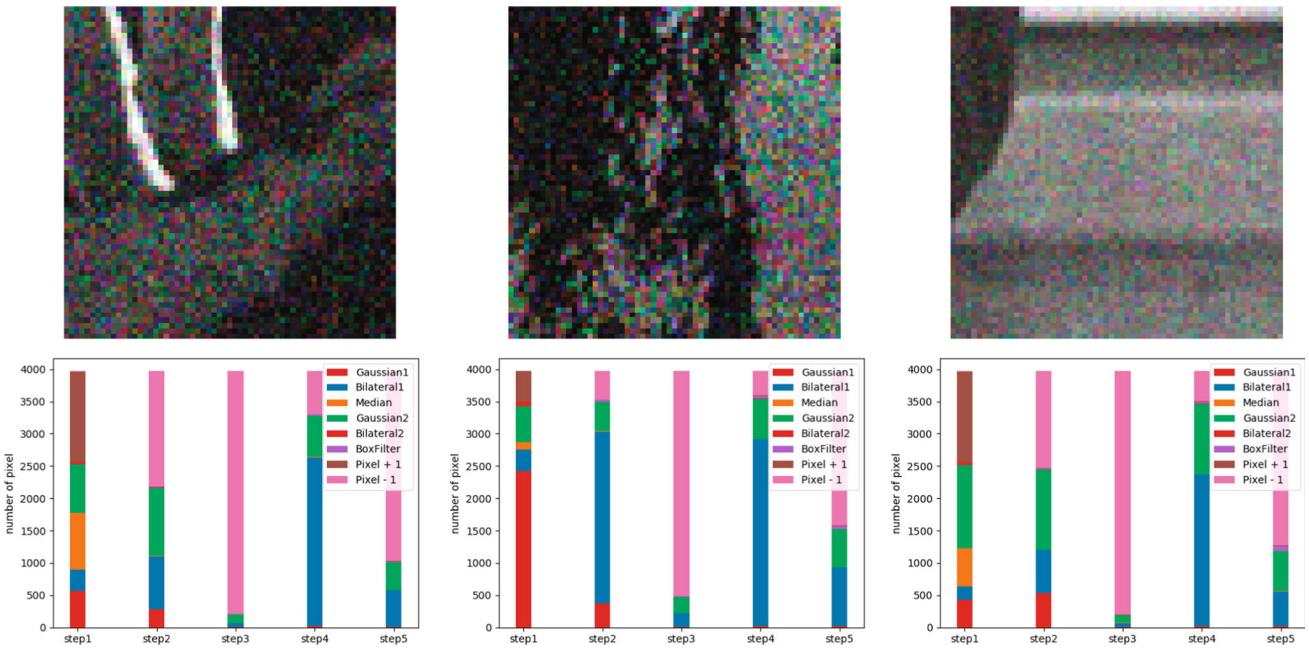

Fig. 7 The visualized restoration processing of three test images

training time of the algorithm will increase significantly, and it will converge at about 15e5 steps. Our method does not need to load the pre-trained model, and can reach convergence within 1e6 steps, which is a 1/3 step reduction. Also, we compared the size of the two methods' models and the average test time of the two methods on 100 test images, the results are shown in Table 3. We can see from the results that the test efficiency between Pixel-RL and our method is almost near. Since the two methods use similar network structures, but the output dimensionality of our method is higher, it shows that the test efficiency of our method is better than that of the Pixel-RL method. In addition, our method only uses 3854 images for training, which realizes the task of image restoration under small sample conditions. Compared with the 25,296 training images of Pixel-RL, the method in this paper also has higher training efficiency.

In order to explore the specific actions performed by our method, we visualized the restoration process of the images at each step. Figure 7 is the result of three random test images on the BSD68 dataset. The result is represented by a stacked bar graph. In the stacked bar chart, each bar represents the proportion of actions taken at a certain step in the process. We can see that, at the beginning of the restoration, since the image contains multiple noises, it is more important to choose a suitable filter. After that, the filtered image is fine-tuned through the increase or decrease of the pixel value, and finally the other noise generated in the previous steps is removed by the subsequent operation again, so as to achieve the restoration task of the multi-noise image. The interpretable details are described below: At step = 1 or 2, since the image is contaminated by a variety of noises, choosing suitable filters can make the policy obtain greater rewards. At step = 3, since the image will become blurred after being processed by filters, the reward will not increase but may decrease if the policy continues to select the filters. Therefore, the policy chooses the pixel value operation ($+1$ or -1) to further improve image clarity. At step=4, the policy mainly selects the bilateral filter, for it not only reduces the luminosity and color difference between pixels caused by the $+1$ or -1 operation, but also retains the edge information of the image. At step = 5, since the noise of the image has been basically processed, the policy considers more about improving the clarity of the image, so more -1 operations are selected.

6 Conclusion

We propose a method, which integrates a variety of traditional denoisers into a strong denoiser to restore the images which contain more than one type of noise and traditional denoisers cannot directly restoration. We redefine this problem as MARL problem on the condition of POMDP. To solve this new problem, we propose a method which combine the recurrent neural network with the off-policy RL algorithm and optimized the exploration of agent in pixel-wise. Through a variety of parallel processing of the image and a learned policy based on RL, each pixel is given a weight, and an image that is closest to the true pixel value is merged according to the weight. Several experiments showed that our method not only achieves the recovery task of damaged images very

well, but also requires only a few samples to achieve the recovery effect. However, this method still has limitations. The quality of the final restored image is largely limited by the effect of the traditional filter in toolbox, so how to adaptively change the traditional filter during the training process to make it well adapted to various environmental conditions will be important.

Acknowledgements We are very grateful to the anonymous reviewers for their constructive comments on improving this paper.

Author Contributions Jie Zhang was responsible for experimental design, implementation, and paper editing. Qiyuan Zhang participated in the realization of the experimental process. Xixuan Zhao participated in the editing of the article, and Jiangming Kan guided the experiment.

Funding This work was supported by the Key-Area Research and Development Program of Guangdong Pro-vince, Grant No. 2019B02 0223003.

Declarations

Conflict of interest All the authors declare no conflict of interests.

Ethical approval This paper does not contain any studies with human participants or animals performed by any of the authors.

Informed consent This paper does not contain any studies with human participants performed by any of the authors, so there is no informed consent involved.

References

Agustsson E, Timofte R (2017) Ntire 2017 challenge on single image super-resolution: dataset and study. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 126–135

Anaya J, Barbu A (2018) RENOIR-a dataset for real low-light image noise reduction. J Vis Commun Image Represent 51:144–154

Bertsekas DP, Tsitsiklis JN (1995) Neuro-dynamic programming: an overview. In: Proceedings of 1995 34th IEEE conference on decision and control. IEEE, pp 560–564

Bowling M, Veloso M (2001) Rational and convergent learning in stochastic games. In: International joint conference on artificial intelligence. Lawrence Erlbaum Associates Ltd, pp 1021–1026

Buades A, Coll B, Morel J-M (2005) A non-local algorithm for image denoising. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). IEEE, pp 60–65

Burger HC, Schuler CJ, Harmeling S (2012) Image denoising: Can plain neural networks compete with BM3D? In: 2012 IEEE conference on computer vision and pattern recognition. IEEE, pp 2392–2399

Buşoniu L, Babuška R, De Schutter B (2010) Multi-agent reinforcement learning: an overview. In: Innovations in multi-agent systems and applications-1. Springer, pp 183–221

Cao Q, Lin L, Shi Y (2017) Attention-aware face hallucination via deep reinforcement learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 690–698

Chen Y, Yu W, Pock T (2015) On learning optimized reaction diffusion processes for effective image restoration. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5261–5269

Chen W, Wilson J, Tyree S (2015) Compressing neural networks with the hashing trick. In: International conference on machine learning, pp 2285–2294

Dabov K, Foi A, Egiazarian K (2007) Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans Image Process 16:2080–2095

Furuta R, Inoue N, Yamasaki T (2019) Pixelrl: Fully convolutional network with reinforcement learning for image processing. IEEE Trans Multimed 22(7):1704–1719

Guo J, Chao H (2017) One-to-many network for visually pleasing compression artifacts reduction. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3038–3047

Guestrin C, Koller D, Parr R (2001) Solving factored POMDPs with linear value functions. In: 17th international joint conference on artificial intelligence (IJCAI-01) workshop on planning under uncertainty and incomplete information. Citeseer, pp 67–75

Han S, Mao H, Dally WJ (2015) Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding. ArXiv preprint arXiv:1510.00149

Kim J, Lee JK, Lee KM (2016) Accurate image super-resolution using very deep convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1646–1654

Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. ArXiv preprint arXiv:1412.6980

Koller D, Parr R (2013) Policy iteration for factored MDPs. ArXiv preprint arXiv:1301.3869

Li D, Wu H, Zhang J, Huang K (2018) A2-RL: Aesthetics aware reinforcement learning for image cropping. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8193–8201

Li W, Feng X, An H (2020) MRI reconstruction with interpretable pixel-wise operations using reinforcement learning. In: Proceedings of the AAAI conference on artificial intelligence, pp 792–799

Li Z, Zhang X (2019) Deep reinforcement learning for automatic thumbnail generation. In: International conference on multimedia modeling. Springer, pp 41–53

Liao X, Li W, Xu Q (2020) Iteratively-refined interactive 3D medical image segmentation with multi-agent reinforcement learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9394–9402

Lin L-J, Mitchell TM (1992) Memory approaches to reinforcement learning in non-Markovian domains. Carnegie-Mellon University. Department of Computer Science

Littman ML (1994) Markov games as a framework for multi-agent reinforcement learning. In: Machine learning proceedings. Elsevier, pp 157–163

Littman ML (2001) Friend-or-foe Q-learning in general-sum games. In: ICML, pp 322–328

Lowe R, Wu YI, Tamar A (2017) Multi-agent actor-critic for mixed cooperative-competitive environments. In: Advances in neural information processing systems, pp 6379–6390

Mairal J, Elad M, Sapiro G (2007) Sparse representation for color image restoration. IEEE Trans Image Process 17:53–69

Park J, Lee J-Y, Yoo D, So Kweon I (2018) Distort-and-recover: color enhancement using deep reinforcement learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5928–5936

Polikar R (2012) Ensemble learning. In: Ensemble machine learning. Springer, pp 1–34

Rudin LI, Osher S, Fatemi E (1992) Nonlinear total variation based noise removal algorithms. Phys Nonlinear Phenom 60:259–268

Rashid T, Samvelyan M, De Witt CS (2018) QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. ArXiv preprint arXiv:1803.11485

Singsanga S, Hattagam W, Tat EH (2010) Packet forwarding in overlay wireless sensor networks using NashQ reinforcement learning. In: 2010 6th international conference on intelligent sensors, sensor networks and information processing. IEEE, pp 85–90

Suganuma M, Liu X, Okatani T (2019) Attention-based adaptive selection of operations for image restoration in the presence of unknown combined distortions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 9039–9048

Tan M (1993) Multi-agent reinforcement learning: independent vs. cooperative agents. In: Proceedings of the 10th international conference on machine learning, pp 330–337

White CC III, Scherer WT (1989) Solution procedures for partially observed Markov decision processes. Oper Res 37:791–797

Xie C, Wu Y, Maaten LV, Yuille AL, He K (2019) Feature denoising for improving adversarial robustness. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 501–509

Yang Y, Luo R, Li M (2018) Mean field multi-agent reinforcement learning. ArXiv preprint ArXiv:1802.05438

Yu K, Dong C, Lin L, Loy CC (2018) Crafting a toolchain for image restoration by deep reinforcement learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2443–2452

Zhang K, Zuo W, Chen Y (2017) Beyond a gaussian denoiser: residual learning of deep CNN for image denoising. IEEE Trans Image Process 26:3142–3155

Zhang K, Yang Z, Başar T (2019) Multi-agent reinforcement learning: a selective overview of theories and algorithms. ArXiv preprint arXiv:1911.10635