

Article

# Multi-Frame Star Image Denoising Algorithm Based on Deep Reinforcement Learning and Mixed Poisson–Gaussian Likelihood

Ming Xie , Zhenduo Zhang, Wenbo Zheng, Ying Li \* and Kai Cao

Navigation College, Dalian Maritime University, Dalian 116026, China; mingxie@dlnu.edu.cn (M.X.); zhangzhenduo@dlnu.edu.cn (Z.Z.); zwb123456789@dlnu.edu.cn (W.Z.); caokai@dlnu.edu.cn (K.C.)

\* Correspondence: yldmu@dlnu.edu.cn

Received: 26 September 2020; Accepted: 20 October 2020; Published: 22 October 2020



**Abstract:** Mixed Poisson–Gaussian noise exists in the star images and is difficult to be effectively suppressed via maximum likelihood estimation (MLE) method due to its complicated likelihood function. In this article, the MLE method is incorporated with a state-of-the-art machine learning algorithm in order to achieve accurate restoration results. By applying the mixed Poisson–Gaussian likelihood function as the reward function of a reinforcement learning algorithm, an agent is able to form the restored image that achieves the maximum value of the complex likelihood function through the Markov Decision Process (MDP). In order to provide the appropriate parameter settings of the denoising model, the key hyperparameters of the model and their influences on denoising results are tested through simulated experiments. The model is then compared with two existing star image denoising methods so as to verify its performance. The experiment results indicate that this algorithm based on reinforcement learning is able to suppress the mixed Poisson–Gaussian noise in the star image more accurately than the traditional MLE method, as well as the method based on the deep convolutional neural network (DCNN).

**Keywords:** star image; image denoising; reinforcement learning; maximum likelihood estimation; mixed Poisson–Gaussian likelihood

## 1. Introduction

A star image is obtained from star sensor, which is a high-accuracy attitude determination instrument. The three-axis attitude and spatial position of the star sensor can be calculated based on the reference of stars recognized in the digitized star image. Being able to accurately collect and process a star image at all times, however, is one of the challenges in the application of a star sensor due to the brightness of the sky's background and the complicated mixture of noise [1]. These conditions lead to a low Signal-to-Noise Ratio (SNR) in the star image obtained in the daytime, and thus have negative influences on the calculation of attitude and position. Similarly, it is an important basis for the star sensor to work efficiently that noise is effectively suppressed and the star targets are accurately recognized in the star image. Researchers have proposed various types of algorithms to suppress the effects of salt-and-pepper noise, strip noise, speckle noise, and defective pixels [1–5]. Nevertheless, mixed Poisson–Gaussian noise remains after these denoising methods are applied and affects the extraction of the true value of star points [6].

The traditional method suppresses the mixed Poisson–Gaussian noise simply by taking the arithmetic mean of multiple frames of star images. However, the restoration through such methods is not likely to be accurate without a large number of star images that are taken in a short period or by a stationary star sensor [7]. Considering the movement in the carrier of the star sensor, such conditions could sometimes be unrealistic. Maximum likelihood estimation (MLE) is one of widely used methods

that can restore a star image with limited number of frames. In the MLE method, the distribution model of the noise is constructed as a likelihood function, and the image that is most likely to produce the given observations is termed as the restored image. However, mixed Poisson–Gaussian noise is still difficult to be suppressed through the MLE method due to its complicated likelihood function. This will be discussed in detail in the next section.

Towards the goal of high-accuracy star image restoration, this study integrated deep reinforcement learning techniques with the MLE method and developed an iterative denoising algorithm that can accurately suppress mixed Poisson–Gaussian noise. Generally, an agent proposes “restored images”, which are evaluated through a modified likelihood function. Borrowing the idea of the MLE method, the mission of the agent is to find the image that returns the highest probability according to the modified likelihood function. Based on the gradient descent algorithm in the deep reinforcement learning method, the agent efficiently learns to propose a more appropriate “restored image” until it gets close enough to the ground truth. Thus, the aim and contribution of this study is to build a fully automated denoising algorithm that can accurately suppress the mixed Poisson–Gaussian noise in a star image.

## 2. Related Works

### 2.1. The Application of MLE in Star Image Denoising

The MLE method regards the image denoising problem as the optimization process of the likelihood function. Moreover, other regularization terms can be added to the likelihood function as the constraints on the estimator and form a denoising algorithm together with the likelihood function. Based on the MLE method and the image blur model with Gaussian noise, Katsaggelos proposed an image restoration algorithm using blur coefficients identification and expectation-maximization (EM) [8]. Llacer and Nunez applied the MLE method to restore the astronomic images obtained from the Hubble Space Telescope and proposed an iterative MLE and Bayesian algorithm [9]. Similarly, Synder discussed the convolution of astronomic image obtained from Charge Coupled Device (CCD) camera and suggested that the restoration of such astronomic image could potentially be achieved via EM method [10]. Benvenuto further explored the image denoising problem specifically for star images. They gave an approximate model of the flux of photons and verified the existence of its solution [11]. Li et al. constructed the likelihood function for multi-frame Adaptive Optics (AO) image according to the Poisson distribution model, based on which they proposed an AO image denoising method that achieved accurate results [12]. However, this algorithm is not able to restore the image with mixed Poisson–Gaussian noise, which has a more complicated distribution model. Zhang et al. approximated the mixed Poisson–Gaussian noise model using generalized Anscombe transformation approximation Fourier ptychographic (GATFP), and then solved the transformed likelihood function with the MLE method [6]. Although the GATFP method is able to optimize the complicated distribution model of mixed Poisson–Gaussian noise, it cannot avoid the error introduced during the approximation.

### 2.2. Reinforcement Learning and Its Application in Image Processing

Derived from behavioral psychology, reinforcement learning is a classical topic in the studies of artificial intelligence (AI) [13,14]. Various algorithms have been developed for reinforcement learning, but they share a major process in common: an agent learns through the interactions with certain environment, and gradually finds an optimal solution or approach. The Markov Decision Process (MDP) is a commonly used method of reinforcement learning [15]. In MDP, an agent takes limited choices of actions to update the state of the environment, then receives encouragement or punishment according to a reward function that evaluates the chosen action, as well as how it changed the state [16]. As it can be inferred from the learning process, MDP is usually accomplished through an iterative algorithm.

Due to the development of AI technology in the last decade, the machine learning algorithm has been applied in many subjects, including object detection [17–19], data mining [20,21], image processing [22,23], etc. Typically, the convolutional neural network (CNN) has been successfully applied in various image processing tasks, including deblurring [24–26], denoising [27–29], JPEG artifacts reduction [30–32], and super-resolution [33–40], owing to its ability to efficiently handle imagery information. Incorporated with the deep learning algorithm, traditional reinforcement learning also rapidly developed [41,42] and showed its potential of application in image processing [43,44]. Yu et al., designed a toolbox for image processing, and then developed an image restoration algorithm based on reinforcement learning using the tools within the toolbox [44]. Despite its various applications in image processing, the machine learning method is not widely utilized in star image restoration. This is partly because the star image restoration problem is different from the general image restoration problem: the goal is to restore the value of a star point as close to the true value as possible, rather than make the image visually fine. A deep convolutional neural network (DCNN) model proposed by Liu et al. is one of the limited examples of applying deep learning method in star image denoising [45]. Liu et al. trained a supervised DCNN model with simulated star image and achieved better restoration than traditional method.

### 2.3. Mixed Poisson–Gaussian Noise in Star Image

Gaussian noise in the star image can be introduced by many aspects including the noise of circuit components, the temperature change in image sensor, etc. The intrinsic noise introduced during the photon counting process of the photosensitive component, however, depends on the signal value, thus follows Poisson distribution. Because the measurements from all pixels are independent, for a star image with  $M \times M$  pixels and  $N$  observations, the likelihood function that describe the possibility of receiving the known  $N$  observations is given as Equation (1) [46]:

$$p(z_q|b_q) = \prod_{q=1}^N \prod_{i=1}^{M^2} \left( \sum_{j=1}^{+\infty} \frac{e^{-[b_q]_i} [b_q]_i^j}{j!} \frac{e^{-\frac{1}{2\sigma^2}([z_q]_i - j)^2}}{\sqrt{2\pi\sigma^2}} \right) \quad (1)$$

where  $z_q$  is the noise image,  $b_q$  is the ground truth,  $\sigma$  is the variance of Gaussian distribution, subscript  $i$  is an index of the number of pixels,  $q$  is an index of the number of observations, and  $j$  is an index of the possible observation of the gray value. As discussed above, MLE method is intended to find a restore image  $b'_q$  that returns the highest value of the likelihood function.

Taking the negative logarithm of the likelihood function makes it easier to solve this problem. Thus, the optimized likelihood function is given as:

$$\min f(z_q|b'_q) = -\log\left(\prod_{q=1}^N (p(z_q|b'_q))\right) = -\sum_{q=1}^N \sum_{i=1}^{M^2} \log\left(\sum_{j=1}^{+\infty} \frac{e^{-[b'_q]_i} [b'_q]_i^j}{j!} \frac{e^{-\frac{1}{2\sigma^2}([z_q]_i - j)^2}}{\sqrt{2\pi\sigma^2}}\right) \quad (2)$$

Nevertheless, it is still extremely difficult to directly solve this likelihood function. Marnissi et al., further optimized the likelihood function using generalized Anscombe transformation (GAT) approximation, and form the likelihood function as Equation (3) [47]:

$$p(z_q|b_q) = \prod_{i=1}^{M^2} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(z_q - 2\sqrt{[b_q]_i + \frac{3}{8} + \sigma^2}\right)^2\right) \quad (3)$$

As indicated in Equation (3), GAT method simplifies the mixed Poisson–Gaussian likelihood as a Gaussian-like distribution. Although it is able to provide an approximate solution to the likelihood function, the tails of variance stabilized coefficients distribution are still empirically longer than normality [48] (which means the variance of the transformed noise still depends on signal intensity). In this study, the mixed Poisson–Gaussian likelihood function is directly introduced in the reinforcement

learning algorithm without approximation. Therefore, it is expected that the likelihood function given by Equation (2) can be solved more accurately, and the mixed Poisson–Gaussian noise in the star image can be suppressed more effectively.

### 3. Methodology

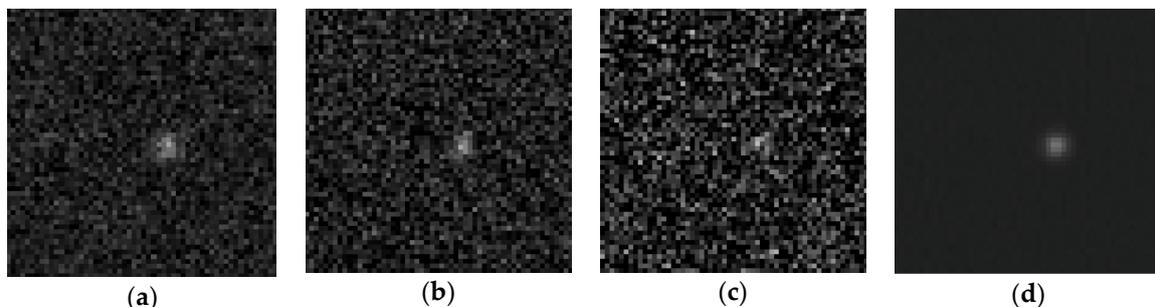
#### 3.1. Dataset

In this study, simulated star images are produced as the dataset for training and testing the model. Simulated star image has been successfully utilized to train machine learning model owing to its ability of generating a variety of star images under different conditions at very low cost [45]. Xu et al., studied the stellar radiation model and proposed a star image simulation method based on calibration coefficient, which relates to stellar magnitude  $m_v$  and color temperature  $T$  [49]. Generally, the total energy of photoelectrons generated by an observed star can be calculated by Planck’s black body radiation law as:

$$E_s(\lambda, T) = \frac{2\pi hc^2}{\lambda^5 [\exp(hc/\lambda k_B T) - 1]} \quad (4)$$

where  $\lambda$  is wavelength,  $k_B$  is Boltzmann constant,  $h$  is Planck constant,  $E_s$  is the stellar radiation energy per unit area, per unit time, and per incremental wavelength ( $\text{W}/\text{m}^3$ ).

The energy of star point is then dispersed into several pixels based on point spread function (PSF) of the optical system. After that, the star points are integrated with the background simulated based on MODTRAN software to generate the simulated star image. Finally, mixed Poisson–Gaussian noise is added to the image. This simulation method is adopted in this study to produce the training dataset. More specifically, three levels of noise images are produced in the dataset as mild noise, moderate noise, and severe noise, in which the variances of Gaussian noise are 5, 10, and 20, respectively. The noise in each image is randomly generated and independent from each other in order to avoid having the same noisy instantiation among the different images in the dataset. The simulated image of ground truth and three sets of star images with different amount of noise is shown as Figure 1.

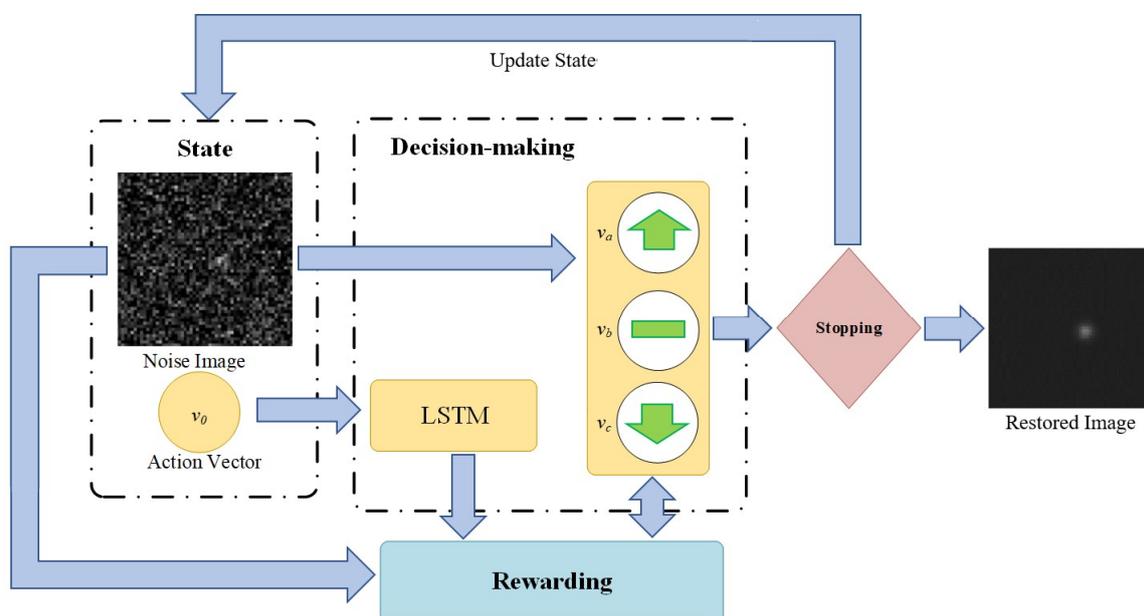


**Figure 1.** Simulated star images with: (a) mild noise, (b) moderate noise, (c) severe noise, (d) ground truth.

Two thousand different simulated star images are produced as the images of ground truth. For each star image of the ground truth, 10 noise images are produced for each type of noise. Thus, there are 6000 sets of noise images that contain three levels of noise and are produced based on 2000 different images of ground truth. Each set includes 10 frames of noise image, thus the training and testing for certain image with each ground truth image that contains each type of noise is carried out based on 10 frames of noise images. The image sets are randomly shuffled for model training. Seventy percent of the images are used for training, 10% of the images are used for validation, and 20% of the images are used for testing. The star images with different types of noise are distributed in the training, validation, and testing dataset as evenly as possible. All of the tests on hyperparameters are carried out separately with the testing images of these three types of noise.

### 3.2. Reinforcement Learning Algorithm

Different from existing star image denoising methods (such as GAT) that try to approximately solve the likelihood function given by Equation (2), we intend to solve the denoising problem through MDP, which include an environment part (state), a decision-making part (agent), a rewarding part, and a stopping part (Figure 2). The decision-making part receives the noise image and treats it as the initial state, makes a sequence of actions on the image and produces a processed image. The rewarding part mainly consists of a reward function that returns a higher value when the processed image is closer to the ground truth. In the case of the star image denoising in this study, the reward function is directly derived from the likelihood function that is given by Equation (2). The stopping part evaluates the processed image and determines whether it is close enough to the ground truth. If the processed image does not satisfy the stopping part, it will be sent back to the decision-making part as an updated state. Otherwise, the stopping part will end the iterations and return the processed image as the final restored image. The detail settings of these parts are discussed as follows.



**Figure 2.** Flowchart of the star image denoising algorithm based on reinforcement learning.

#### 3.2.1. State and Initial State

The state is referred as a set of input information for the decision-making part. In this study, the state consists of two sections: (1) the current input image, on which the selected actions will be directly applied; (2) the historical action vector, which is the action selected in the previous iteration. In the iterative image denoising model proposed by Yu et al., a past historical action vector is also added as a reference to the action selection at current iteration [44]. They suggest that the restoration results are improved compared with using current input image only.

The input image of the initial state is set as the arithmetic mean of the multi-frame star images. Because the average of multi-frame star images is usually not too far away from the ground truth (as discussed in the introduction, this is the traditional way to restore star image), using it as the initial input image reduces the numbers of iterations (NOI), and also avoids the local optimum. The action vector of the initial state is set as a zero vector, since there is no action taken before the initial state.

### 3.2.2. Decision-Making Part

The decision-making part receives the input information and performs predefined choices of actions on the image. There are two modules in the decision-making part corresponding to the two sections of input state. The first module processes the input image. The star images are 8-bit or 16-bit grayscale images, of which the gray value is determined by the number of photons counted by the star sensor. Therefore, the decisions for any given pixels can be simply defined as: (1) increasing gray value by  $w$ ; (2) decreasing gray value by  $w$ ; (3) stay the same. The parameter  $w$  is the searching radius, which is (similar as the learning rate in deep neural network) a hyperparameter that can be adjusted based on the property of image. For the 8-bit grayscale images with mild noise, a searching radius can be directly set as 1, since the initial input image is usually not too far away from the ground truth. For the 16-bit gray-scale images or images with severe noise, searching radius could be set as a variable that is larger in the first few iterations, and then gradually decreases to 1 as the iterations go on. Increasing the initial searching radius (ISR) can reduce NOI when the amount of noise is relatively large, but it has to decrease to 1 in order to produce accurate restored image.

The second module analyzes the historical action vectors based on long short-term memory (LSTM) [50]. LSTM is a special type of recurrent neural network (RNN), of which the output of current state partly depends on the previous input. Compared with regular RNN, LSTM includes an additional hidden state that decides whether the previous input should be “remembered” or “forgotten” based on a gating function. By abandoning unnecessary inputs, LSTM overcomes the gradient vanishing/explosion problem in regular RNN, and achieves better performance in processing long sequence data [51]. LSTM is introduced in the model because the iterative actions performed by the decision-making part form a sequence dataset, which can be effectively processed by LSTM. By storing historical images and corresponding actions, LSTM provides additional information and references for current action selection, and enable the model to learn from previous actions.

### 3.2.3. Rewarding Part

Theoretically, any functions that fulfill the requirement that returns a higher value as it is getting closer to the ground truth could potentially be a reward function. However, the choice of reward function significantly influences the performance of the restoration.

In the traditional reinforcement learning algorithm for image denoising, reward function is usually derived from Peak Signal-to-Noise Ratio (PSNR) or Structural Similarity (SSIM) [44]. In this study, the likelihood function of mixed Poisson–Gaussian noise, which was approximately solved in the previous studies [5,47,48], is introduced as the reward function without approximation. Specifically, the reward function is showing as follows:

$$\max f(z_q|b'_q) = \log\left(\prod_{q=1}^N (p(z_q|b'_q))\right) = \sum_{q=1}^N \sum_{i=1}^{M^2} \log\left(\sum_{j=1}^{+\infty} \frac{e^{-[b'_q]_i} [b'_q]_i^j}{j!} \frac{e^{-\frac{1}{2\sigma^2}([z_q]_i - j)^2}}{\sqrt{2\pi\sigma^2}}\right) \quad (5)$$

There are two advantages of applying this likelihood function as reward function: (1) it returns highest value at the ground truth, which fulfills the requirement of reward function for reinforcement learning; (2) by directly applying the likelihood function as the reward function rather than making an approximate solution (like GAT method), it is expected to achieve more accurate restoration on the star image.

In practice, the variance of the Gaussian noise is unknown, but is required to calculate the rewards. This term needs to be calculated from the variance of the marginal pixels that are far away from stars. Because the signal for the background is expected to be close to 0 and Poisson noise is related to signal intensity, ideally the Poisson noise in the marginal pixels are ignorable, and the variance in the marginal pixels are mostly contributed by that of the Gaussian noise.

It should also be noted that there is an infinite term in the function. This term refers as the probability of the showing certain gray value at a given pixel. Therefore, it usually returns high value when the gray value is close to the ground truth, and then decreases to almost 0 when the difference is too large (it is less likely that the observed value has too much difference from the ground truth). Therefore, in our algorithm, this term is not taken into account when it is smaller than  $10^{-8}$ —the probability less than that has little influence on the reward function and is ignorable. Moreover, in practice, the differences between the possibilities calculated from Equation (5) could be very small (especially when the variance of the noise is relatively large and the curve of the likelihood function is “flat”), which means the differences between the rewards of actions are not obvious. As a result, the decision-making part may fail to determine the appropriate action due to the vague rewarding. In order to solve this problem, the values returned from the likelihood function may need to be normalized to a larger scale, so as to magnify the differences between the rewards of different states and actions.

#### 3.2.4. Stopping Part

An automatic stopping part is designed in the algorithm in order to determine whether the processed image is close enough to the ground truth, and if yes, automatically stop the iterations. According to the decision-making part and reward function described in the previous sections, it can be easily inferred that when the processed image reaches the ground truth, the decision-making part would choose to stay at the same gray value, because either increasing or decreasing the gray value would return negative rewards. Thus, stopping index (SI), defined as the numbers of times that the decision-making part has continuously chosen to stay at the same gray value, is designed as the main algorithm of the stopping part. SI is also a hyperparameter that should be adjusted based on property of the star images. For the reason that the decision-making part could possibly choose random actions, increasing SI in some extent can improve the accuracy of the restoration results, but at the same time, increase number of iterations (NOI). For the same reason, the maximum number of iterations (MNI) may also need to be defined, so as to avoid infinite loop. The detail test on SI and MNI will be discussed in the next section.

### 3.3. Implementation Details

The algorithm in this study is built using Tensorflow backend [52] and in Python 3.6 environment. Deep Q-learning [41] is adopted in this study for the training process. The  $\epsilon$ -greedy is set relatively high as 0.95, because the rewarding function is continuous and has only single peak, which means taking the action with higher rewarding is a better choice than taking a random action for the most of the cases. Adam [53] optimizer is applied in the model and the batch size is set at 32. The hyperbolic tangent function is applied as the gating function of LSTM.

In order to find the favorable choice of hyperparameters, two sets of experiments were conducted to test the hyperparameters introduced above. For the tests on SI, the searching radius was set as a constant of 1, and MNI was set as a constant of 500. Six experiments were conducted for SIs, varying from 2 to 7 with an increment of 1. For the tests on the ISR, the SI is set as a constant of 3, and the MNI was set as a constant of 500. Four experiments were conducted for the test on ISR. In one of the experiments, the searching radius is set as a constant of 1. While in the other three experiments, the ISR is set at 5, 10, and 20, respectively, and then decreased at a factor of 0.5 (rounding up) every 10 iterations. The working flow and specific implementations for the algorithm proposed in this study is summarized in Algorithm 1.

**Algorithm 1.** Steps for the proposed denoising algorithm

---

**Step 1:** Create initial input image as the arithmetic mean of  $N$  frames of noise image; define initial action vector  $v_0$  as a zero vector; initialize stopping counter  $c_s = 0$ ; setup hyperparameter:  $SI, MNI, ISR$  \* Initialize operation

**Step 2:** Calculate the variance of Gaussian noise using the boundary pixels

**Step 3:** Define action table ( $a_1 = +w, a_2 = -w, a_3 = no\ action$ , initialize  $w$  equal to  $ISR$ ); initialize action-value function  $Q$  with random weight  $\theta$  \* Setup deep Q-learning

**Step 4:** for each of the  $M \times M$  pixels: initialize sequence of state  $\{b_i\}$  \* Start iteration

Iterate through  $i = 1, 2, \dots, MNI$ :

Process  $v_{i-1}$  through LSTM and obtain  $v'_{i-1}$

With probability  $\varepsilon$  select  $v_i = \operatorname{argmax} Q(b_{i-1}; v'_{i-1}; \theta)$

Otherwise select a random action  $v_i$  from action table

Execute action  $v_i$ , update environment  $b_i = b_{i-1} + v_i$ , and observed reward  $r_i$

Update action-value function  $Q$  with the observed  $(b_i; v_i; r_i)$

After every 10 iterations  $w \leftarrow \text{rounding up}(w/2)$

if  $a_3$  is chosen in this loop then:  $c_s \leftarrow c_s + 1$

else:  $c_s \leftarrow 0$

end if

if  $c_s \geq SI$  or  $I > MNI$  then: output the state  $b_i$  as the restored result, end the iteration \* Automatic stopping

end if

end for

---

#### 4. Results

Hyperparameters were tested and calibrated so as to provide appropriate settings to the model.  $SI$ ,  $MNI$  and  $ISR$  were tested for simulated star images of different amount of noise. Accuracy and running time are the two aspects tested in the experiments. The running time is evaluated by NOI—obviously, more iterations are more time-consuming. Concerning the goal of star image restoration, the accuracy was evaluated by mean squared error (MSE), PSNR, and SSIM, which are defined in Equations (6)–(8) as follow:

$$MSE = \frac{1}{M^2} \sum_{i=1}^{M^2} ([b'_q]_i - [b_q]_i)^2 \quad (6)$$

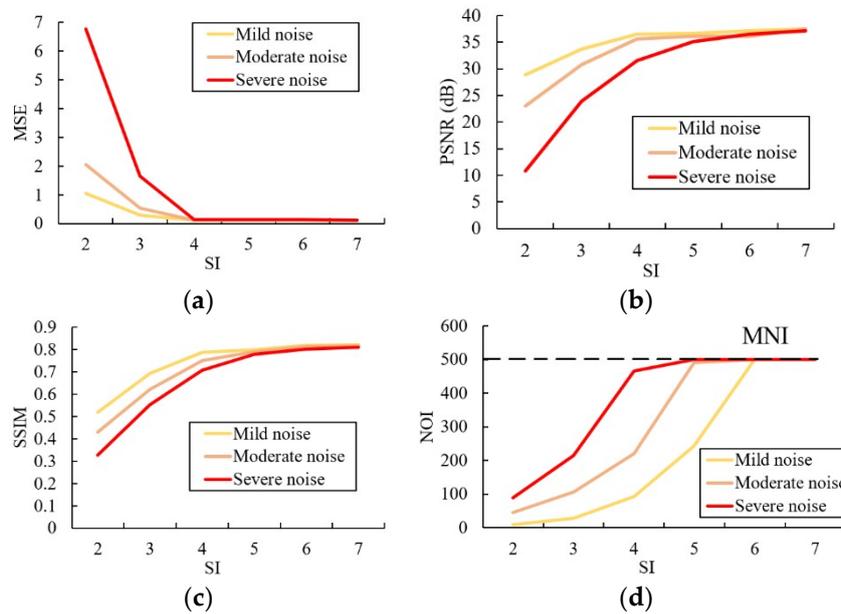
$$PSNR = 10 \log_{10} \left( \frac{MAX^2}{MSE} \right) \quad (7)$$

$$SSIM = \frac{(2\mu_b\mu_{b'} + (0.01MAX)^2)(2\sigma_{bb'} + (0.03MAX)^2)}{(\mu_b^2 + \mu_{b'}^2 + (0.01MAX)^2)(\sigma_b^2 + \sigma_{b'}^2 + (0.03MAX)^2)} \quad (8)$$

where  $MAX$  is the possible maximum gray value (which is 255 for 8-bit and 65,535 for 16-bit grayscale image),  $b$  represents the ground truth and  $b'$  represents the restore image (also in subscript),  $\mu$  is the average and  $\sigma$  is the variance,  $\sigma_{bb'}$  is the covariance of restore image and ground truth. These three evaluation indicators are calculated for each set of noise images, and then averaged with other sets of noise images with the same type of noise. As described in Section 3.1, there are 1200 sets of noise images with three different levels of noise utilized as the testing data. Thus, the three evaluation indicators are averaged among 400 tests (which is the value range of  $q$  in Equation (6)) in this study.

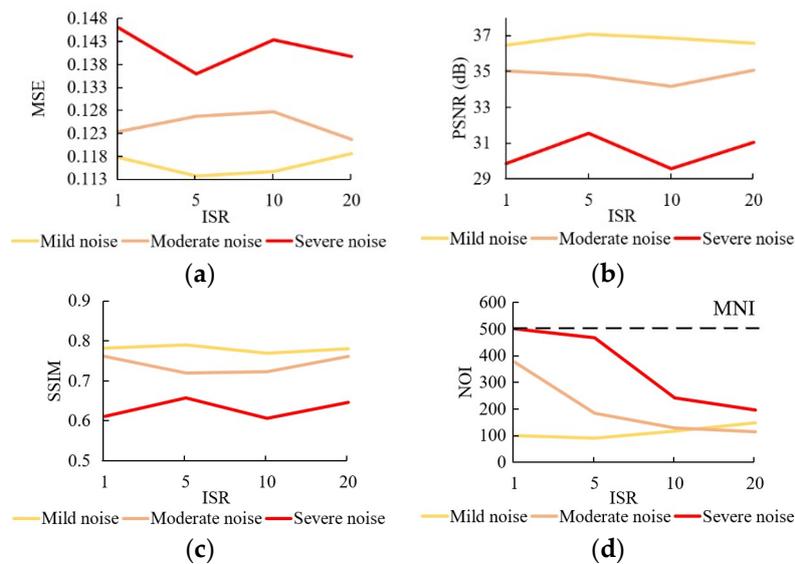
The results of experiments on  $SI$  and  $MNI$  are shown in Figure 3. It turns out that the three evaluation indicators share the similar trend with the change of  $SI$ . Specifically, the accuracy of the model improves significantly in a limited range with the increase of  $SI$  regardless of the amount of noise in the star images. That means increasing  $SI$  in a certain range can improve the quality of restoration results. However, as shown in the same figure, the increase of the  $SI$  also leads to more NOI (longer running time). When the  $SI$  exceeds certain threshold, the model ends up in infinite loop and

has to be stopped by MNI. In that vein, the appropriate SI setting should be large enough to achieve accurate restoration, but small enough to avoid unnecessary iterations.



**Figure 3.** Test results of (a) mean square error (MSE); (b) peak signal-to-noise ratio (PSNR); (c) structure similarity (SSIM); (d) number of iterations (NOI) at different settings of stopping index (SI) for three levels of noise.

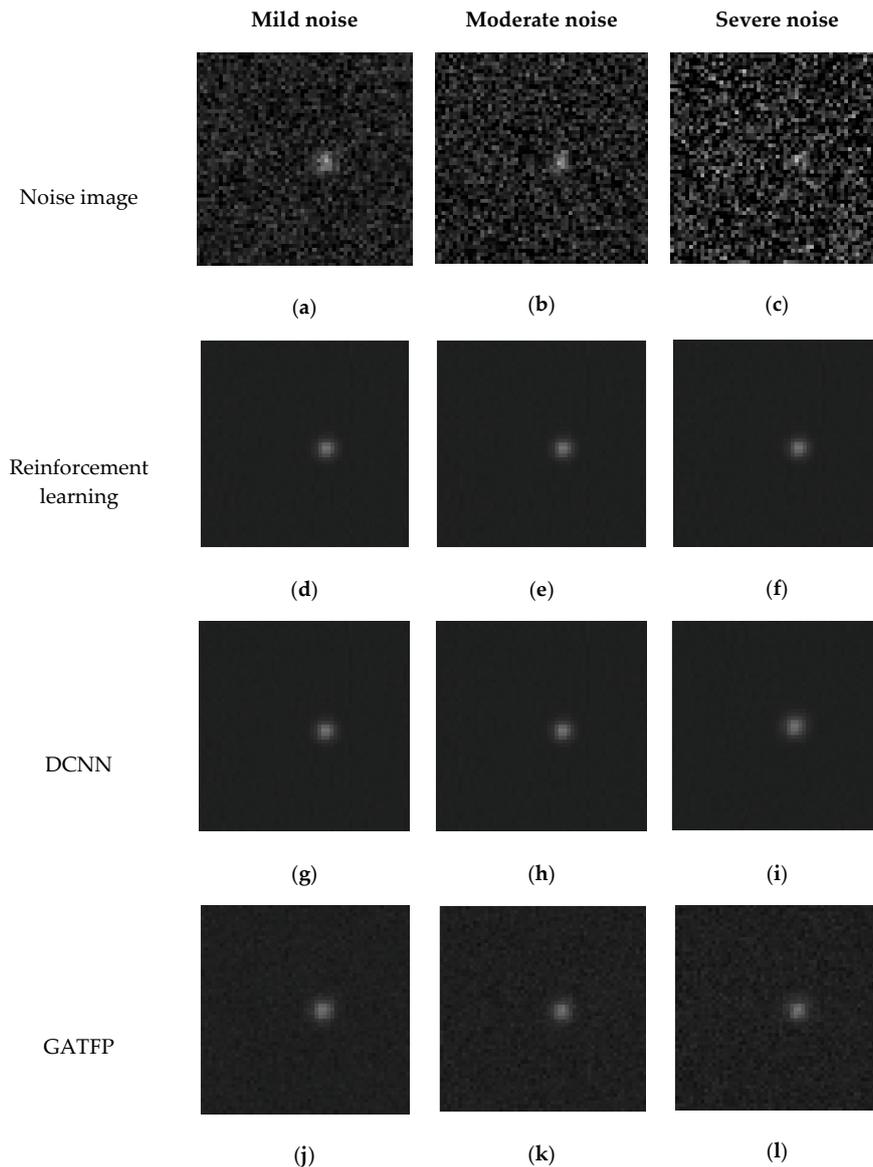
The results of experiments on the ISR are shown in Figure 4. It turns out that ISR does not have significant influence on the accuracy of restoration results, but affects the running time. Moreover, its influences on running time vary for star images with different amounts of noise: for the star images with mild noise, increasing the ISR does not reduce running time; for the star images with moderate and severe noise, however, the NOI significantly reduced with a larger ISR.



**Figure 4.** Test results of (a) MSE; (b) PSNR; (c) SSIM; (d) NOI at different settings of initial searching radius (ISR) for three levels of noise.

The performance of the star image denoising method proposed in this study is compared with GATFP method proposed by Zhang et al. [6] and DCNN algorithm proposed by Liu et al. [45] on

the same testing dataset. For the GATFP method, the denoising process is conducted based on the optimized likelihood function given in Equation (3). The implementation of the DCNN model is a setup based on the structure described in Reference [45], which includes 17 convolutional layers and 256 neurons in each layer. Rectified linear unit (ReLU) [54] is applied as the activation function between the layers. The performance of these denoising methods is also evaluated with the three evaluation indicators described above. The arithmetic means of the evaluation indicators for all the testing images in each category of noise are compared in Table 1 for all the three denoising methods. Typical examples of restored images are also shown in Figure 5.



**Figure 5.** Restoration results on the noise images produced based on the same ground truth (Figure 1): (a) noise image with mild noise, (b) noise image with moderate noise, (c) noise image with severe noise, (d) denoising result using reinforcement learning on mild noise, (e) denoising result using reinforcement learning on moderate noise, (f) denoising result using reinforcement learning on severe noise, (g) denoising result using DCNN [45] on mild noise, (h) denoising result using DCNN [45] on moderate noise, (i) denoising result using DCNN [45] on severe noise, (j) denoising result using GATFP [6] on mild noise, (k) denoising result using GATFP [6] on moderate noise, (l) denoising result using GATFP [6] on severe noise.

**Table 1.** Arithmetic means of evaluation indicators for all the testing images with three levels of noise using three different denoising methods.

	Mild Noise			Moderate Noise			Severe Noise		
	MSE	PSNR(dB)	SSIM	MSE	PSNR(dB)	SSIM	MSE	PSNR(dB)	SSIM
Reinforcement learning	0.1147	37.57	0.8215	0.1149	37.10	0.8150	0.1440	31.58	0.7060
DCNN [45]	0.1238	35.43	0.7776	0.1260	35.06	0.7602	0.1587	30.81	0.6816
GATFP [6]	0.1402	31.18	0.7153	0.1638	27.94	0.6384	0.1925	23.05	0.5483

## 5. Discussion

### 5.1. Hyperparameter Settings

#### 5.1.1. SI and MNI

As indicated in the experiment results, SI influences both accuracy and running time. When the SI is set at 2 or 3, the accuracy is relatively low because the decision-making part randomly chooses to stay at the same value. But this possibility quickly decreases as the SI increases to 4. That means the accuracy of the denoising model can be improved by increasing SI in a limited extent: as indicated in Figure 3, the increase of SI does not improve the model accuracy when it is set beyond 5.

NOI also increases with SI. In fact, when SI is larger than 6, the model easily gets into infinite loop and has to be stopped by MNI. This is because in order to stop the iterations without MNI, the decision-making part has to avoid randomly choosing to increase or decrease the gray value for number of times equals to SI after reaching the appropriate restored value—that possibility also decreases with the increase of SI.

It is also notable that the SI has little relation with the amount of noise or searching radius (with adequate iterations). Because the amount of noise decreases with the iterations, and SI does not start to work until the decision-making part reaches to the appropriate restored value. In that vein, 4 or 5 seems to be the appropriate value for SI regardless of the amount of noise in the star image. The MNI setting, however, depends on the amount of noise in the image, since NOI increases with the amount of noise. A short experiment under low SI (for example, 3 or 4) may be necessary to figure out approximately how many iterations the decision-making part needs to form the restored image before formally running the model in order to find the appropriate setting of MNI.

#### 5.1.2. Initial Searching Radius

As indicated in the experiment results, ISR does not affect the accuracy significantly, since it will decrease to 1 in the end. But it affects the NOI. For the images with mild noise, an ISR larger than 1 is unnecessary, and sometimes even causes more iterations, especially when ISR is larger than the noise. For the images with moderate and severe noise, a larger ISR reduces NOI significantly, and therefore, restores the images faster. In practice, however, the quality of observed star image is unknown. In that vein, searching radius is a hyperparameter that needs to be calibrated with actual input star image.

According to discussion of the experiments above, the recommended setting of the hyperparameter for the noise images with different amount of noise using the hyperparameter settings shown in Table 2. This setting is also applied to the training of the reinforcement learning model when comparing its performance with other two methods.

**Table 2.** Hyperparameter settings for the star images of three different amount of noise.

	Mild Noise	Moderate Noise	Severe Noise
Stopping index (SI)	4	4	4
Initial searching radius (ISR)	1	5	10
Maximum number of iterations (MNI)	300	500	500

## 5.2. Comparisons with Existing Methods

The algorithm proposed in this article is compared with GATFP [6] and DCNN [45] methods. The performances of the two methods on the dataset used in study generally confirm with those presented in the previous studies. As indicated in Table 1 and Figure 5, these denoising methods are able to suppress the noise in a star image to some extent. The three denoising methods have better performance on the suppression of mild noise than that of moderate noise restoration, which is better than that of extreme noise. This means the images with less noise are naturally easier to be accurately denoised. Although it can hardly be identified from the restored image (Figure 5) by human eyes, the three methods have different performances in term of retrieving the true value of star target. As shown in Table 1, all the three of the evaluation indicators shows that the reinforcement learning method proposed in this study provides more accurate restoration results than that of the DCNN method, which is more accurate than that of GATFP method. That means the algorithm proposed in this article is able to achieve more accurate restoration than the existing star image denoising methods.

The fact that the two machine learning methods achieve better restoration results, especially when the star images are more distorted, is possibly because they overcome the error introduced by the approximation in GAT method. Moreover, the method proposed in this article can perform more accurate restorations than that based on DCNN. It should be noted that the advantage in accuracy of the restoration results is probably caused by the reward function applied in the reinforcement learning algorithm, which is directly linked with the distribution pattern of the noise in the star image. Although the reinforcement learning algorithm proposed in this study is designed more specifically for star image with mixed Poisson–Gaussian noise, it certainly has the potential to be applied to suppress other type of image noise, as long as the corresponding reward function is defined.

In comparing the two machine learning models, the DCNN model [45] includes a series of convolutional layers for image feature extraction and a deep neural network (DNN) for denoising calculation, while the reinforcement learning model presented in this study is an iterative MDP model that is integrated with DNN for action-value regression. The different structures of the two models lead to different training processes. The weights and bias of the neurons are trained in DCNN model, while the optimal denoising process is trained through iterative trials in reinforcement learning model. As a result, the reinforcement learning model is much simpler than the DCNN model in terms of computational complexity. As shown in Table 3, the parameters that need to be trained in the reinforcement learning model is less than one half of those in DCNN model, and the number of computations in training the reinforcement learning model is only about one fourth of that in training DCNN model. This results generally conforms with a comparison between reinforcement learning model and DCNN model by Yu et al. [44].

**Table 3.** Comparison of the computational complexity between the reinforcement learning model proposed in this study and DCNN model [45].

	Reinforcement Learning	DCNN [45]
Parameters ( $\times 10^5$ )	1.10	2.47
Calculations ( $\times 10^8$ )	2.26	9.02

Moreover, large numbers of noise images for different ground truth are necessary to train DCNN model, while the reinforcement learning model requires much less training dataset: theoretically, the agent can finally propose the restored image even with only one set of noise images with adequate numbers of trials. Thus, the star image denoising model proposed in this study depends on less training dataset, and may be more practical to extract the true value of star target, especially when the observations are limited.

## 6. Conclusions and Future Studies

A novel approach for star image denoising based on reinforcement learning is presented in this study. Compared with the existing methods based on DCNN, this model learns to dynamically search for the appropriate restored image based on reward function derived from mixed Poisson–Gaussian likelihood function. Thus, the mechanism behind our algorithm is more intuitive, and leads to more accurate restoration, than those that are based on DCNN. Compared with MLE methods that try to approximately solve the likelihood function (such as GAT), the same likelihood function is applied as the reward function without approximation. In that way, the likelihood function is solved by the computer through the MDP more accurately.

The limitation of this algorithm is that we didn't add any kind of convolution layer to the input image. As a result, when the star image is rather huge, the restoration process becomes time-consuming (the accuracy is not affected though). The reason we did not apply CNN in the algorithm is that the reward function derived from the likelihood function only works for the gray value of the star image, but not the image feature as captured by the CNN. An appropriate way to integrate CNN in the algorithm, and thus further improve its efficiency, could be a topic of our study in the future.

It is also a potential topic to extend the application of the reinforcement learning model proposed in this study into the denoising practice of broader types of images, which could have different numbers of color depths. Processing the noise image depth by depth may not be a favorable approach, since it disregards the correlations between the color depths. This problem may also be solved by a properly designed CNN that can extract the features of a noise image.

**Author Contributions:** Conceptualization, M.X., Z.Z. and Y.L.; methodology, M.X.; software, M.X. and K.C.; validation, M.X., W.Z. and K.C.; formal analysis, M.X.; investigation, M.X.; data curation, M.X. and W.Z.; writing—original draft preparation, M.X. and W.Z.; writing—review and editing, M.X., Z.Z., W.Z., Y.L., and K.C.; visualization, M.X.; supervision, Y.L. and Z.Z.; project administration, Y.L.; funding acquisition, Y.L. and M.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded in part by China National Key R&D Program grant number 2018YFB1600400 and 2020YFE0201500, and in part by China Postdoctoral Science Foundation, grant number 2020M670730.

**Acknowledgments:** The authors would also like to thank Qinglai Yu, Zhanjun Ma from Dalian Maritime University for their help in preparing the training dataset. The authors are grateful to the anonymous reviewers for their constructive comments on this work.

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

## References

1. Wang, W.; Wei, X.; Lin, J.; Wang, G. Noise suppression algorithm of short-wave infrared star image for daytime star sensor. *Infrared Phys. Technol.* **2017**, *85*, 382–394. [[CrossRef](#)]
2. Jin, B.; Park, N.; George, K.M.; Choi, M.; Yeary, M.B. Modeling and analysis of soft-test/repair for CCD-based digital X-ray systems. *IEEE Trans. Instrum. Meas.* **2003**, *52*, 1713–1721. [[CrossRef](#)]
3. Hwang, H.; Haddad, R. Adaptive median filters: New algorithms and results. *IEEE Trans. Image Process.* **1995**, *4*, 499–502. [[CrossRef](#)]
4. Yugander, P.; Tejaswini, C.H.; Meenakshi, J.; Samapath kumar, K.; Suresh Varma, B.V.N.; Jagannath, M. MR image enhancement using adaptive weighted mean filtering and homomorphic filtering. In Proceedings of the International Conference on Computational Intelligence and Data Science, Gurgaon, India, 20–23 February 2019; Procedia Computer Science. Elsevier: Amsterdam, The Netherlands, 2020. [[CrossRef](#)]
5. Zhou, Z.; Lam, E.Y.; Lee, C. Nonlocal means filtering based speckle removal utilizing the maximum a posteriori estimation and the total variation image prior. *IEEE Access* **2019**, *7*, 99231–99243. [[CrossRef](#)]
6. Zhang, Y.; Song, P.; Dai, Q. Fourier ptychographic microscopy using a generalized Anscombe transform approximation of the mixed Poisson–Gaussian likelihood. *Opt. Express* **2017**, *25*, 168–179. [[CrossRef](#)]
7. Bonavito, N.L.; Dorband, J.E.; Busse, T. Maximum entropy restoration of blurred and oversaturated Hubble space telescope imagery. *Appl. Opt.* **1993**, *32*, 5768–5774. [[CrossRef](#)]
8. Katsaggelos, A.K.; Lay, K.T. Maximum likelihood blur identification and image restoration using the EM algorithm. *IEEE Trans. Signal Process.* **1991**, *39*, 729–733. [[CrossRef](#)]

9. Llacer, J.; Nuiez, J. Iterative maximum-likelihood and Bayesian algorithms for image reconstruction in astronomy. In *Restoration of Hubble Space Telescope Images*; White, R.L., Allen, R.J., Eds.; Space Telescope Science Institute: Baltimore, MD, USA, 1990; pp. 62–69.
10. Snyder, D.L.; Hammoud, A.M.; White, R.L. Image recovery from data acquired with a charge-coupled-device camera. *J. Opt. Soc. Am. A* **1993**, *10*, 1014–1023. [[CrossRef](#)]
11. Benvenuto, F.; Camera, A.L.; Theys, C.; Ferrari, A.; Lanteri, H.; Bertero, M. The study of an iterative method for the reconstruction of images corrupted by Poisson and Gaussian noise. *Inverse Probl.* **2008**, *24*, 1–20. [[CrossRef](#)]
12. Li, D.; Sun, C.; Yang, J.; Liu, H.; Peng, J.; Zhang, L. Robust Multi-Frame Adaptive Optics Image Restoration Algorithm Using Maximum Likelihood Estimation with Poisson Statistics. *Sensors* **2017**, *17*, 785. [[CrossRef](#)]
13. Barto, A.G. Reinforcement Learning. In *Neural Systems for Control*; Omidvar, O., Elliott, D.L., Eds.; Academic Press: Cambridge, MA, USA, 1997; pp. 7–30.
14. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 1998.
15. Bellman, R.E. A Markov Decision Process. *J. Math. Fluid Mech.* **1957**, *6*, 679–684.
16. Puterman, M.L. *Markov Decision Processes: Discrete Stochastic Programming*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2014.
17. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
18. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37.
19. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster-RCNN: Towards real-time object detection with region proposal networks. In Proceedings of the Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 265–283.
20. Zhang, P.; Cheng, H.; Zou, B.; Dai, P.; Ye, C. Load data mining based on deep learning method. In Proceedings of the 3rd International Conference on Computer Science and Application Engineering, Sanya, China, 22–24 October 2019.
21. Nguyen, G.; Dlugolinsky, S.; Bobak, M.; Tran, V.; Garcia, A.L.; Heredia, I.; Malik, P.; Hluchy, L. Machine learning and deep learning for frameworks and libraries for large-scale data mining: A survey. *Artif. Intell. Rev.* **2019**, *52*, 77–124. [[CrossRef](#)]
22. Zhang, K.; Zou, W.; Zhang, L. FFDNet: Toward a Fast and Flexible Solution for CNN based Image Denoising. *IEEE Trans. Image Process.* **2018**, *27*, 4608–4622. [[CrossRef](#)]
23. Jiao, L.; Zhao, J. A survey on the new generation of deep learning in image processing. *IEEE Access* **2019**, *7*, 172231–172263. [[CrossRef](#)]
24. Nah, J.; Kim, T.H.; Lee, K.M. Deep multi-scale convolutional neural network for dynamic scene deblurring. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 257–265.
25. Sun, J.; Cao, W.; Xu, Z.; Ponce, J. Learning a convolutional neural network for non-uniform motion blur removal. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 769–777.
26. Xu, L.; Tao, X.; Jia, J. Inverse kernels for fast spatial deconvolution. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 5–12 September 2014; pp. 33–48.
27. Chen, Y.; Yu, W.; Pock, T. On learning optimized reaction diffusion processes for effective image restoration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5261–5269.
28. Lefkimiatis, S. Non-local color image denoising with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3587–3596.
29. Zhang, K.; Zou, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a Gaussian denoiser: Residual learning of Deep CNN for image denoising. *IEEE Trans. Image Process.* **2017**, *26*, 3142–3155. [[CrossRef](#)]

30. Dong, C.; Deng, Y.; Loy, C.C.; Tang, X. Compression artifacts reduction by a deep convolutional network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 576–584.
31. Guo, J.; Chao, H. Building dual-domain representations for compression artifacts reduction. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 629–644.
32. Wang, Z.; Liu, D.; Chang, S.; Ling, Q.; Yang, Y.; Huang, T.S. D3: Deep dual-domain based fast restoration of JPEG-compressed images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2764–2772.
33. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *38*, 295–307. [[CrossRef](#)]
34. Hui, T.W.; Loy, C.C.; Tang, X. Depth map super-resolution by deep multi-scale guidance. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 3539–3569.
35. Kim, J.; Lee, J.K.; Lee, K.M. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1646–1654.
36. Kim, J.; Lee, J.K.; Lee, K.M. Deeply-recursive convolutional network for image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1637–1645.
37. Haris, M.; Shakhnarovich, G.; Ukita, N. Deep back-projection networks for super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1664–1673.
38. Tai, Y.; Yang, J.; Liu, X. Image super-resolution via deep recursive residual network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2790–2798.
39. Tai, Y.; Yang, J.; Liu, X.; Xu, C. Memnet: A persistent memory network for image restoration. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4549–4557.
40. Wang, X.; Yu, K.; Dong, C.; Loy, C.C. Recovering realistic texture in image super-resolution by deep spatial feature transform. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 606–615.
41. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [[CrossRef](#)]
42. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M. Mastering the game of go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489. [[CrossRef](#)]
43. Peng, Y.; Zhang, J.; Ye, Z. Deep reinforcement learning for image hashing. *IEEE Trans. Multimed.* **2020**, *22*, 99–111. [[CrossRef](#)]
44. Yu, K.; Dong, C.; Lin, L.; Loy, C.C. Crafting a toolchain for image restoration by deep reinforcement learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2443–2452.
45. Liu, Y.; Zhao, C.; Xu, Q. Neural network-based noise suppression algorithm for star images captured during daylight hours. *Acta Opt. Sin.* **2019**, *39*, 1–8.
46. Chouzenoux, E.; Jeziarska, A.; Pesquet, J.C.; Talbot, H. A convex approach for image restoration with exact Poisson–Gaussian likelihood. *SIAM J. Imaging Sci.* **2015**, *8*, 2662–2682. [[CrossRef](#)]
47. Marnissi, Y.; Zheng, Y.; Pesquet, J.C. Fast variational Bayesian signal recovery in the presence of Poisson–Gaussian noise. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Shanghai, China, 20–25 March 2016; pp. 3964–3968.
48. Zhang, J.; Hiraoka, K.; Jin, X. Quantile analysis of image sensor noise distribution. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Brisbane, QLD, Australia, 19–24 April 2015; pp. 1598–1602.
49. Xu, Q.; Zhao, C.; Li, X. Stellar radiation modeling and image simulation for airborne daytime star sensor. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Shanghai, China, 20–25 March 2016; pp. 630–635.

50. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
51. Sundermeyer, M.; Schluter, R.; Ney, H. LSTM neural networks for language modeling. In Proceedings of the InterSpeech, Portland, OR, USA, 9–13 September 2012.
52. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Ghemawat, S.; Irving, G.; Isard, M.; Kudlur, M.; et al. TensorFlow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating System Design and Implementation (OSDI), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
53. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the IEEE International Conference on Learning Representation, San Diego, CA, USA, 7–9 May 2015.
54. Nair, V.; Hinton, G.E. Rectified linear units improve restricted Boltzmann machines. In Proceedings of the International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; pp. 807–814.

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).