

Article

Removing Rain Streaks from Visual Image Using a Combination of Bilateral Filter and Generative Adversarial Network

Yue Yang ^{1,2,3,*}, Minglong Xu ^{1,2,3}, Chuang Chen ^{1,2,3} and Fan Xue ^{1,2,3}¹ School of Automation, China University of Geosciences, Wuhan 430074, China² Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems, Wuhan 430074, China³ Engineering Research Center of Intelligent Technology for Geo-Exploration, Ministry of Education, Wuhan 430074, China

* Correspondence: yangyue@cug.edu.cn

Abstract: Images acquired using vision sensors are easily affected by environmental limitations, especially rain streaks. These streaks will seriously reduce image quality, which, in turn, reduces the accuracy of the algorithms that use the resulting images in vision sensor systems. In this paper, we proposed a method that combined the bilateral filter with the generative adversarial network to eliminate the interference of rain streaks. Unlike other methods that use all the information in an image as the input to the generative adversarial network, we used a bilateral filter to preprocess and separate the high frequency part of the original image. The generator for the high-frequency layer of the image was designed to generate an image with no rain streaks. The high-frequency information of the image was used in a high-frequency global discriminator designed to measure the authenticity of the generated image from multiple perspectives. We also designed a loss function based on the structural similarity index to further improve the effect of removal of the rain streaks. An ablation experiment proved the validity of the method. We also compared images in synthetic and real-world datasets. Our method could retain more image information, and the generated image was clearer.

Keywords: vision sensors; rain streaks removal; bilateral filter; generative adversarial network



Citation: Yang, Y.; Xu, M.; Chen, C.; Xue, F. Removing Rain Streaks from Visual Image Using a Combination of Bilateral Filter and Generative Adversarial Network. *Appl. Sci.* **2023**, *13*, 6387. <https://doi.org/10.3390/app13116387>

Academic Editors: Dongpo Xu, Huisheng Zhang and Jie Yang

Received: 23 April 2023

Revised: 17 May 2023

Accepted: 22 May 2023

Published: 23 May 2023



Copyright: © 2023 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Vision systems are widely used in such fields as intelligent transportation, security monitoring, and object detection [1]. However, most vision systems are installed in outdoor environments and are easily affected by weather conditions. Images acquired by vision sensors on rainy days often contain rain streaks that degrade the quality of the image, where this, in turn, degrades the accuracy of the recognition and detection of the visual system [2]. Therefore, to improve the quality of imaging and algorithmic stability of the visual sensor system in rainy weather, it is important to study methods to remove rain streaks from images.

The approaches taken in the last two decades for rain streaks removal can be divided into two main types: methods of rain streaks removal based on multi-frame images and those based on single-frame images. The former is mainly used using inter-frame information [3–5], which makes it easy to remove rain streaks. However, in practice, in visual systems, the frame images are usually not continuous, and no prior information can be used. Methods based on multi-frame images struggle to solve the problem if the input image is only a single frame. In this paper, we, therefore, study methods to remove rain streaks from images based on single-frame images. Traditional methods in this vein include dictionary learning [6], sparse representation [7,8], and the Gaussian mixture model [9]. As an important technology in the image field, a convolutional neural network (CNN) has also

been applied to rain streaks removal by directly learning the nonlinear mapping between input and output to improve the quality of the image [10].

Traditional single-frame image-based methods to remove rain streaks are generally dependent on a model of rainy images that is challenging to build. It is also difficult to separate the clear background image from the rain streaks in the image, which prevents the complete removal of the streaks. The relevant methods generally also have high algorithmic complexity. A large number of features are duplicated or redundant between the rainy image and the image without streaks. Most single-frame image-based methods of removing rain streaks that use deep learning employ direct mapping to learn the mapping relationships, which enhances the difficulty and complexity of network learning [11,12]. The currently used loss function is also not strongly constrained, which may lead to image blurring or the incomplete removal of the rain streaks.

Therefore, we propose a method to improve the image quality [13]. It involves first using a bilateral filter to separate the high-frequency part of the rainy image to eliminate the interference from its low-frequency part. A generator is then used to process only the high-frequency part, thereby simplifying the input information. The loss of structural similarity index (SSIM) is proposed as an additional constraint to enhance the effect of removing rain streaks. Finally, the high-frequency global (H-G) discriminator is used to comprehensively evaluate the high-frequency and global parts of the generated image.

The main contributions of this paper are as follows:

- We used a bilateral filter to separate the high-frequency part of the original image.
- An image with no rain streaks, generated by the high-frequency layer of the image. We measured the authenticity of the generated image by a high-frequency global discriminator. The generator for the high-frequency layer of the image was designed to generate an image without rain streaks. We designed a high-frequency global discriminator to measure the authenticity of the generated image from multiple perspectives.
- We proposed the novel loss function based on the structural similarity index to further improve the effect of the rain streaks.

Section 2 compares related methods for removing rain streaks based on single-frame images, consisting of image decomposition and deep learning. Section 3 proposes a novel method that combines the bilateral filter and the GAN and proposes a network. Section 4 presents a series of experiments on datasets and evaluates the effect of rain streaks' removal of the synthetic dataset, both on qualitative and quantitative metrics.

2. Methods for Removing Rain Streaks Based on Single-Frame Images

Current methods on rain streaks' removal from single-frame images are two main categories. One comprises methods based on image decomposition that involve using a model to describe the rain streaks, background images, and an optimization algorithm. The other comprises single-frame image-based methods that use deep learning, which solves the problem by constructing and training a network.

2.1. Single-Frame Image-Based Methods for Removing Rain Streaks Based on Image Decomposition

In earlier works, Kang et al. adopted the idea of layer separation to separate the components of the high-frequency part, representing rain through dictionary learning to obtain an image without streaks. Dictionary learning and sparse coding consumed most of execution time and increased computational complexity [6]. Luo et al. improved this to propose a discriminative sparse coding method based on an analysis of the components of the image's morphology. The aim is to learn the dictionary using the rainy image and that without streaks. Discriminant sparse coding is then used to separate the background image to remove the rain streaks. Discriminant sparse coding separated the background image to remove the rain streaks. The ambiguities between the low-pass frequency components of the image layer and the rain layer cannot be solved [7]. Deng et al. [8] proposed a unidirectional global sparse model that used the alternating direction method of multipliers to remove rain streaks and optimize the complexity of the algorithm. However, when the

number of rain streaks in the image was high, the effect of removal was not satisfactory. The method proposed by Li et al. [9] used the layer separation idea based on the Gaussian mixture model, which could model the background and rain streaks separately for adapting to multiple directions and scales of rain streaks.

2.2. Single-Frame Image-Based Methods for Removing Rain Streaks Based on Deep Learning

Due to the excellent performance of CNN in the field of image processing, its application is becoming more and more extensive. Eigen et al. [14] first applied it to the image to remove rain streaks. The basic idea was to train a CNN on pairs of rainy images and the corresponding ground truth images. The method was particularly effective for sparse and light rain, but it could not produce clean results for dense and heavy rain. Fu et al. [15,16] designed a deep CNN that learned the mapping function between clean and rainy image detail layers on the high-frequency detail content and improved the effect of removal of streaks by modifying the objective function and using image enhancement methods. Yang et al. used multi-stream networks to learn the characteristics of different types of rain streaks for rain streak detection and removal. They proposed a recurrent rain detection and removal network that removed rain streaks and cleared up the rain accumulation iteratively and progressively to handle rain streak accumulation in various shapes and directions of overlapping rain streaks [17]. The same idea was also used by Zhang et al. [18], who proposed a de-raining network called DID-MDN. The difference was that it consisted of two modules: a residual-aware rain-density classifier and multi-stream densely connected de-raining network. However, it was difficult to strictly distinguish between different degrees of rain streaks, resulting in unsatisfactory results. Xia et al. [19] directly learned the residuals between the detail layer of the rainy image and the detail layer of the truth image and improved a simplified residual dense network to de-rain and reduce runtime. Li et al. [20] combined the SE blocks to assign different weights to different rain streak layers and divided the rain removal into multiple stages. Moreover, they combined it with a recurrent neural network to remove rain streaks.

In recent years, generative adversarial networks (GANs) have been widely used in image restoration [21] and data classification [22,23]. The removal of rain streaks from images can also be seen as a recovery problem, and thus many scholars have taken this approach. Zhang et al. used a conditional GAN and designed a loss function that made the de-rained image indistinguishable from its corresponding ground truth clean image. They proposed a multi-scale discriminator to leverage features from different scales to determine whether the de-rained image was real or fake [24]. In [25], a network based on cycle GAN was designed that used dual learning to remove rain streaks, with the background being preserved, and eased the training of both generators by adding a reverse mapping. Additionally, they reduced the mapping range from input to output and made the mapping process easier with residual image learning. The method proposed in [26] used the GAN to inject an attention map into the network to design an attentive GAN. Xiang et al. [27] proposed a feature-supervised GAN that added features of truth images to the generated network for supervision. Based on conditional GAN, Sharma et al. [28] combined the spatial and frequency domain features of rain images to remove rain streaks. Recently, Jin et al. [29] decomposed the input image into background space and rain space from the perspective of feature decomposition, and the two branches were optimized against each other. Chen et al. [30] replaced low-quality features with high-quality features inspired by spirit of closed-loop feedback in the automatic control field to solve the model errors of the conditional generator in a deraining network. Zou et al. proposed a model named Dreaming to Prune Image Deraining Networks to invert the pre-trained model and constrain the orthogonality of their degradation representations to reconstruct diverse and in-distribution rainy data [31]. Chen et al. presented an effective sparse Transformer network to solve image deraining. Based on the observation that vanilla self-attention in the Transformer may suffer from the global interaction of irrelevant information, they developed the top-k sparse attention to keep useful self-attention values for feature aggregation [32].

Single-frame image-based methods of removing rain streaks from images that use image decomposition do not yield satisfactory results and sometimes lose detailed information of the image. The relevant algorithms are also complex. In the deep-learning-based approach, the method in [24,25] directly uses all information as input to the GAN, and the characteristics of the high-frequency portion of the image are not considered. The attentive GAN in [26] was mainly applied to removing raindrops and was not suitable for rain streaks. Raindrops in an image are visual effects caused by rain falling on the glass or the camera lens, and this can be usually avoided. Images acquired in rainy weather contain rain streaks. The method in [27] had some rain streaks, even when there were more rain streaks in the image. We also compare this method in later experimental sections. Therefore, in this paper, we propose a method that combines the bilateral filter and the GAN to enhance the effect of removal of rain streaks from images.

3. Proposed Method

The proposed method combined the bilateral filter and the GAN, as shown in Figure 1. It consisted of two main parts: a rain streaks removal network and an H-G discriminator.

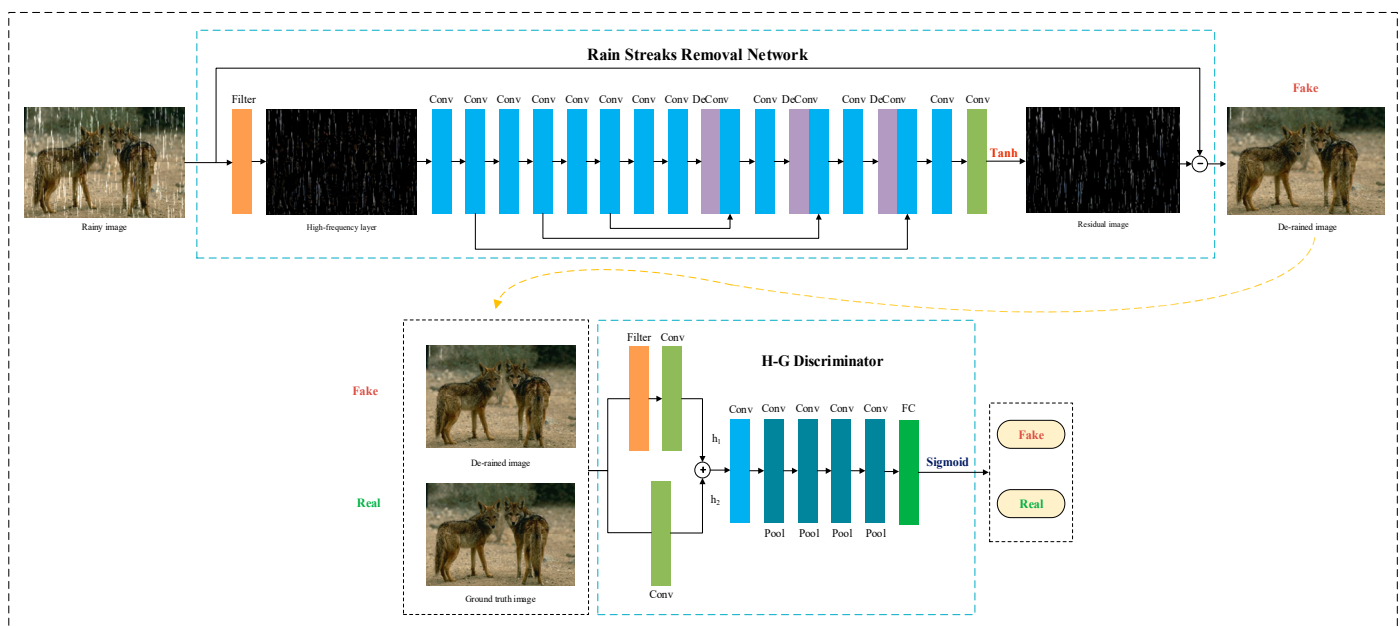


Figure 1. The architecture of the proposed network.

The rain streaks removal network generated an image without streaks as close to the ground truth as possible by the constraint of the loss function. The role of H-G discriminator was to make the generated image fake and the ground truth image true. These two networks were in opposition to each other, so that the H-G discriminator could not judge the true or false of the generated de-rained image, thus achieving a balance. Finally, the optimal solution of the generated image without rain streaks (de-rained image) was obtained.

3.1. Analysis of Filter

Generally, the input is a rainy image X and the output is the de-rained image $\bar{Y}$ (where Y is as close as possible to the truth image, i.e., $\bar{Y} \approx Y$). Therefore, our goals are as follows:

$$X \rightarrow Y. \quad (1)$$

We analyze the relationship between X and Y . Figure 2 shows that there is a large amount of duplicate information in X and Y . We thus need only generate the truth residual image S :

$$S = |Y - X|. \quad (2)$$

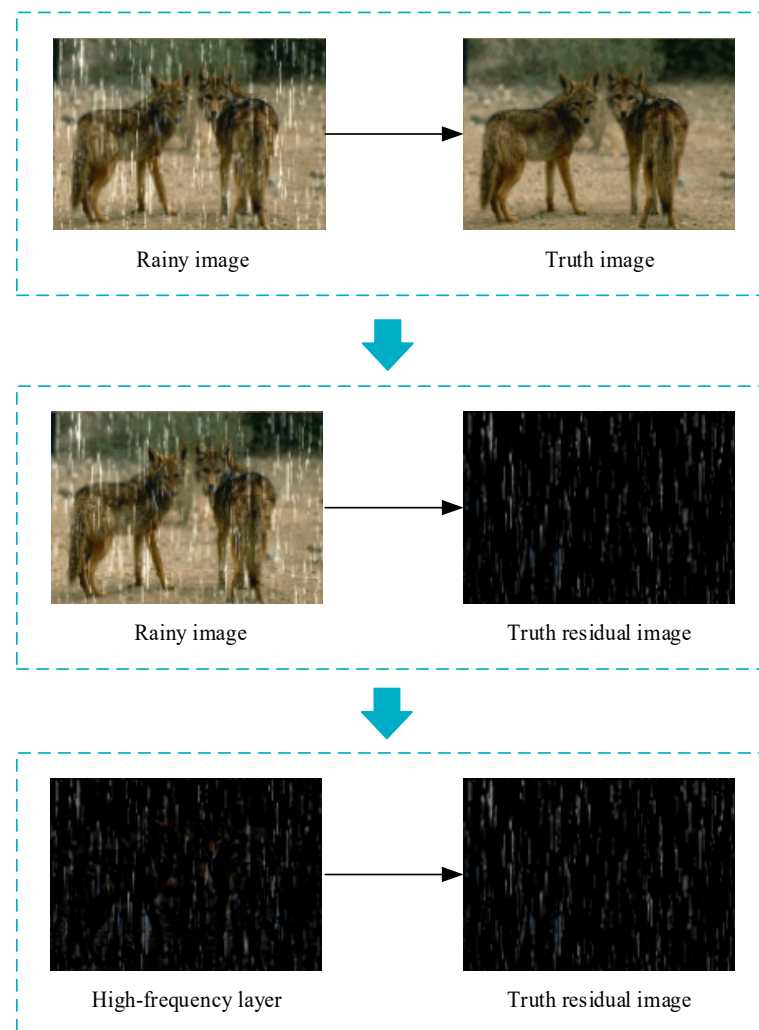


Figure 2. The architecture of the proposed network.

The generator learns the mapping from X to S , thereby improving learning ability and reducing the range of mapping of the network. Our goals can be written as:

$$X \rightarrow S. \quad (3)$$

Furthermore, we simplify the input image. The rainy image can be divided into two parts, which is:

$$X = \tilde{X} + \hat{X}, \quad (4)$$

where $\hat{X}$ represents the low-frequency layer of the input image, and $\tilde{X}$ represents its high-frequency layer.

Images acquired on rainy days contain rain streaks as white marks mainly in the high-frequency part, as shown in Figure 3. This can be further reduced compared with X as input.

The low-frequency part can be obtained by the low-pass filter (in this paper, we use a bilateral filter, and, in the experimental section, we compare several different low-pass filters and analyze the reasons for choosing the bilateral filter), which is:

$$\hat{X} = f_{\text{bilateral}}(X), \quad (5)$$

where $f_{\text{bilateral}}$ represents a bilateral filter.

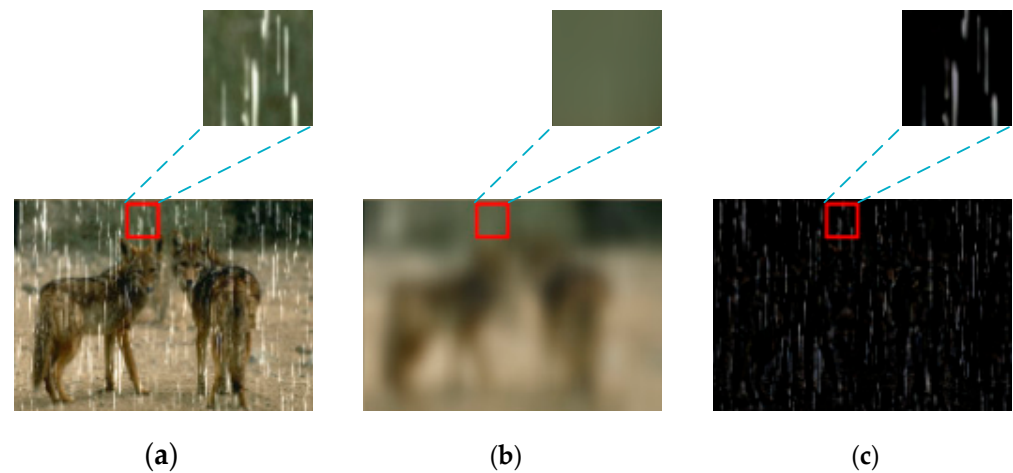


Figure 3. There is an image after the rainy image is subjected to bilateral filter: (a) Rainy image; (b) Low-frequency layer; (c) High-frequency layer.

Then, the high-frequency layer can be obtained by (4) and (5):

$$\tilde{X} = X - f_{\text{bilateral}}(X). \quad (6)$$

Finally, our goal can be expressed as:

$$\tilde{X} \rightarrow S. \quad (7)$$

3.2. Figures, Tables, and Schemes

We used the above analysis to design our rain streaks removal network. As shown in Figure 1, it used a bilateral filter to decompose the high-frequency part of the input image and used it as input. Moreover, the network first generated a residual image to learn the residual and better remove the rain streaks. The rainy image was eventually subtracted from the residual image to obtain the image without streaks.

The good performance of U-Net has been applied in image segmentation [33]. In recent years, some studies have applied it in the task of image rain streaks removal. It can be divided into two parts, a contracting path for feature extraction by downsampling and an expansive path for gradual recovery of image scale by upsampling. The skip connection structure connects the downsampling layer to the upsampling layer, so that the features extracted from the downsampling layer can be directly passed to the upsampling layer. Inspired by this, in this paper, we designed the rain streaks removal network based on U-Net. The structure of the rain streaks removal network is shown in Table 1. It contained 12 convolutional layers and three deconvolutional layers. The first layer used a convolutional kernel of size 5×5 , and the other layers use one of size 3×3 . The larger convolutional kernel could afford a larger perceptual field to ensure the extraction of more features of the original image. The strides of the third, fifth, and seventh convolutional layers were two, and thus the size of the feature map changed from I to $I/4$, $I/16$, and $I/64$. Correspondingly, after three iterations of deconvolutional operations on the 9th, 11th, and 13th layers, the size of the feature map changed back to the original. The input to layer 10 was the concatenation of the outputs of layers 9 and 6; layer 12 was the concatenation of the outputs of layers 11 and 4; and the input to layer 14 was the concatenation of the outputs of layers 13 and 3. This preserved the image original information and improved the correlation between contexts. Layers 1 to 14 used the leaky rectified linear units (LReLU) activation function, and the last layer used the Tanh function to map the result to the interval $[-1, 1]$.

Table 1. The specific structure of Rain Streaks Removal Network.

No.	Layer	K ¹	ST ²	In-Out Channel	BN	A ³	F ⁴	Output
1	Conv	5 × 5	1	3-64	Yes	LReLU	I ⁵	320 × 480 × 64
2	Conv ⁶	3 × 3	1	64-64	Yes	LReLU	I	320 × 480 × 64
3	Conv	3 × 3	2	64-64	Yes	LReLU	I/4	160 × 240 × 64
4	Conv	3 × 3	1	64-128	Yes	LReLU	I/4	160 × 240 × 128
5	Conv	3 × 3	2	128-128	Yes	LReLU	I/16	80 × 120 × 128
6	Conv	3 × 3	1	128-256	Yes	LReLU	I/16	80 × 120 × 256
7	Conv	3 × 3	2	256-256	Yes	LReLU	I/64	40 × 60 × 256
8	Conv	3 × 3	1	256-512	Yes	LReLU	I/64	40 × 60 × 512
9	DeConv ⁷	3 × 3	2	512-256	No	LReLU	I/16	80 × 120 × 256
10	Conv	3 × 3	1	512-256	Yes	LReLU	I/16	80 × 120 × 256
11	DeConv	3 × 3	2	256-128	No	LReLU	I/4	160 × 240 × 128
12	Conv	3 × 3	1	256-128	Yes	LReLU	I/4	160 × 240 × 128
13	DeConv	3 × 3	2	128-64	No	LReLU	I	320 × 480 × 64
14	Conv	3 × 3	1	128-64	Yes	LReLU	I	320 × 480 × 64
15	Conv	3 × 3	1	64-3	No	Tanh	I	320 × 480 × 32

¹ K = kernel_size, ² ST = stride, ³ A = activation function, ⁴ F = feature map, ⁵ I = input image, ⁶ Conv = convolutional layer, ⁷ Deconv = Deconvolutional layer.

The high-frequency part of the rainy image is used as input to the CNN to generate the residual image:

$$\bar{S} = F_G^\theta(\tilde{X}), \quad (8)$$

where F_G^θ represents the part of the convolutional network in the rain streaks removal network. Finally, we can obtain the de-rained image $\bar{Y}$ according to (2):

$$\bar{Y} = X - \bar{S}. \quad (9)$$

3.3. Details of the H-G Discriminator

As shown in Figure 1, the H-G discriminator also used the high-frequency information of the image to impose discriminant constraints on the generated image from the perspectives of both the high-frequency and the global parts. Similarly, a bilateral filter was used to extract the high-frequency part of the image. Two parallel convolutional layers were used to extract the high-frequency features and global features, and feature fusion was then carried out. The result was sent to the network for processing.

The structure of the H-G discriminator is shown in Table 2. Both 1.1 and 1.2 represent the two parallel convolutional layers that extracted the high-frequency and the global features. Their strides were three and five. The global feature was the main part of the image, due to which a large convolutional kernel was used for it. The network in its later part had four convolutional and pooling layers that changed the size of the feature map to 1/256. The input to layer 2-1 was the sum of the outputs of layers 1.2 and 1.1, with weights h_1 and h_2 , respectively. The sigmoid activation function was used in the last layer to map the result to the interval [0, 1].

On the whole, the goal of the H-G discriminator as follows:

$$\begin{aligned} F_D^\eta(\bar{Y}, \tilde{\bar{Y}}) &\rightarrow 0 \\ F_D^\eta(Y, \tilde{Y}) &\rightarrow 1 \end{aligned} \quad (10)$$

where F_D^η represents the H-G discriminator. $\tilde{\bar{Y}}$ represents the high-frequency part of $\bar{Y}$, and $\tilde{Y}$ represents the high-frequency part of Y .

Table 2. The specific structure of H-G Discriminator.

No.	Layer	K ¹	ST ²	In-Out Channel	BN	A ³	F ⁴	Output
1-1	Conv	5 × 5	1	3-16	Yes	ReLU	I ⁵	320 × 480 × 16
1-2	Conv ⁶	3 × 3	1	3-16	Yes	ReLU	I	320 × 480 × 16
2-1	Conv	3 × 3	1	16-16	Yes	ReLU	I	320 × 480 × 16
2-2	Max-Pool ⁷	2 × 2	2	16-16	No	/	I/4	160 × 240 × 16
3-1	Conv	3 × 3	1	16-32	Yes	ReLU	I/4	160 × 240 × 32
3-2	Max-Pool	2 × 2	2	32-32	No	/	I/16	80 × 120 × 32
4-1	Conv	3 × 3	1	32-64	Yes	ReLU	I/16	80 × 120 × 64
4-2	Max-Pool	2 × 2	2	64-64	No	/	I/64	40 × 60 × 64
5-1	Conv	3 × 3	1	64-128	Yes	ReLU	I/64	40 × 60 × 28
5-2	Max-Pool	2 × 2	2	128-128	No	/	I/256	20 × 30 × 64
6	FC	/	/	128-1	No	Sigmoid	1	

¹ K = kernel_size, ² ST = stride, ³ A = activation function, ⁴ F = feature map, ⁵ I = input image, ⁶ Conv = convolutional layer, ⁷ Max-Pool = Maximum Pooling.

3.4. Loss Function

Our entire network is based on GAN. In an image-to-image task, the objective function of the GAN is:

$$\min_{\theta} \max_{\eta} L(F_G^{\theta}, F_D^{\eta}) = E_{X, Y \sim p(x, y)} [\log F_D^{\eta}(X, Y)] + E_{X \sim p(x)} [\log (1 - F_D^{\eta}(F_G^{\theta}(X)))] \quad (11)$$

where θ represents the parameters of the rain streaks removal network, and η represents parameters related to the H-G discriminator.

The loss of the generator part is as follows:

$$L_{\text{gen}} = \log(1 - F_D^{\eta}(F_G^{\theta}(X))). \quad (12)$$

Additionally, the loss of the discriminator part as follows:

$$L_{\text{dis}} = -[\log(F_D^{\eta}(Y)) + \log(1 - F_D^{\eta}(\bar{Y}))]. \quad (13)$$

Training the network to remove rain streaks from the image that relied only on adversarial loss was often unable to achieve a satisfactory effect, and the output image was fuzzy. Therefore, we used an additional loss to further constrain the output image to improve the effect of the removal of the streaks.

Given X , the difference in pixels between the predicted image $\bar{Y}$ and the truth image Y is minimal and is recorded as pixel-pixel loss L_{pix} . It can be expressed by the Frobenius norm of the matrix:

$$L_{\text{pix}} = \frac{1}{N} \sum_{i=1}^N \|Y^{(i)} - \bar{Y}^{(i)}\|_F, \quad (14)$$

where N represents the batch size used for training.

The SSIM indicates the degree of similarity between images, and it can evaluate the quality of the image obtained after the removal of rain streaks. Its value is in the interval $[0, 1]$. The larger the value the closer the generated image is to the ground truth image. It is expressed as F_{SSIM} .

As our network first generates the residual image and then generates the final de-rained image, we combined the SSIM of the residual image and the de-rained image as the total loss of SSIM, i.e.,

$$L_{\text{SSIM}} = [1 - F_{\text{SSIM}}(\bar{Y}, Y)] + [1 - F_{\text{SSIM}}(\bar{S}, S)] = 2 - F_{\text{SSIM}}(\bar{Y}, Y) - F_{\text{SSIM}}(\bar{S}, S). \quad (15)$$

Therefore, the total loss of the rain streaks removal network can be expressed as:

$$L_{\text{GEN}} = L_{\text{pix}} + L_{\text{SSIM}} + \alpha L_{\text{gen}}, \quad (16)$$

where α represents the weight of L_{gen} .

As shown in (10), the H-G discriminator adds high-frequency information of the image. Combining it with (13), the total loss of the H-G discriminator can be expressed as:

$$L_{\text{DIS}} = -\left[\log\left(F_D^{\eta}\left(Y, \tilde{Y}\right)\right) + \log\left(1 - F_D^{\eta}\left(\bar{Y}, \tilde{\bar{Y}}\right)\right)\right]. \quad (17)$$

4. Experiment

To evaluate our network, we compared it with four advanced methods in this field: the Gaussian Mixture Model (GMM [9]), Unidirectional Global Sparse Model (UGSM [8]), Deep Convolutional Neural Network (DCNN [16]), and Feature Supervised GAN (FS-GAN [28]).

We present a series of experiments on a synthetic dataset (on the basis of the truth image, the rainy image is synthesized by artificially adding rain streaks through related operations) and a real-world dataset (image containing rain streaks captured with camera on rainy day). The former contained images from the Rain100 L dataset provided in [18], and the latter contained images from the dataset provided in [9]. We trained and tested the proposed algorithm using TensorFlow in the Python environment on an NVIDIA GeForce GTX 1070 with 8 GB of GPU memory. During training, the filtering window of the bilateral filter was 50, and the standard variance of the color space and the coordinate space was set to 30. The h_1 and h_2 in the H-G discriminator were set to 0.2 and 0.8, respectively. The weight in (16) was set to 0.005. Both the rain streaks removal network and the H-G discriminator used the Adam algorithm, and the initial momentum was set to 0.9. The batch size and patch size were set to 2 and 64, respectively. The initial learning rate was 0.01, and the number of iterations was 150 k. After 100 k iterations, the learning rate was reduced to 1/10 of the original. Training stopped once the maximum number of iterations was reached.

The synthetic dataset consisted of 1800 images and was labeled with rain-density level. Three rain-density labels were present in the dataset (as shown in Figure 4 light, medium and heavy). There were approximately 600 images of rain streaks with different orientations and scales per rain density level in the dataset. The images with different backgrounds of the synthetic dataset are shown in Figure 4.

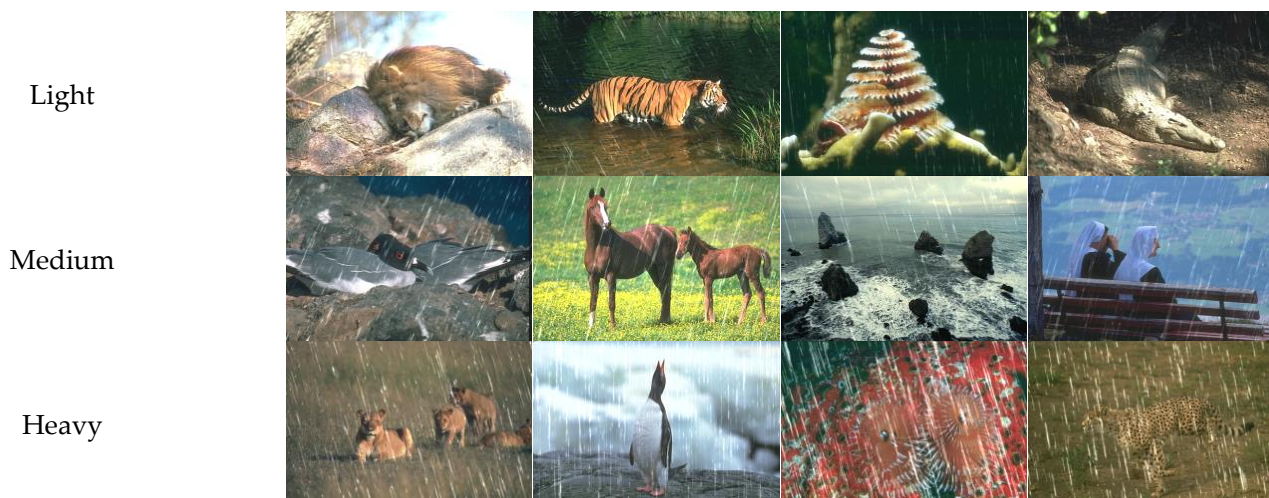


Figure 4. The partial images of the synthetic dataset.

The real dataset consisted of 59 images obtained on a rainy day, shown in Figure 5.

Truth



Figure 5. The partial images of the real dataset.

4.1. Experiments on Synthetic Dataset

We evaluated the effect of rain streak removal in the synthetic dataset both on qualitative and quantitative metrics. The results with the Rain100 L dataset are shown in Figure 6.



Figure 6. The de-rained results on Rain100 L dataset. From the first column to the sixth column: Rainy image, GMM [9], UGSM [8], DCNN [16], FS-GAN [28], Ours.

When there were few rain streaks in an image (as shown in the fifth row of Figure 6), all methods achieved results. When the rain streaks were medium in number (as shown in the first row of Figure 6), the GMM and UGSM yielded worse results than the other methods. In the case of a large number of rain streaks in the image (as shown in the third row of Figure 6), our method, which yielded images barely containing any visible streaks, delivered the best visual effect. The images generated by the GMM contained prominent streaks, and those obtained by the DCNN and FS-GAN contained blur in regions containing the rain streaks. In particular, when the streaks were similar to the background image, such as in the image of white sky region in the second row of Figure 6, the image obtained using the DCNN and FS-GAN contained residual streaks. In comparison, our method removed sparse rain streaks to a greater extent and retained background information, as shown in the fourth row of Figure 6. We also tested the results of some nighttime rain image experiments, as shown in the last three rows of Figure 6. Our approach yielded the best results.

We used Mean Squared Error (MSE) [34], the Structural Similarity Index Method (SSIM) [35], the Peak Signal to Noise Ratio (PSNR) [36], Visual Information Fidelity (VIF) [37], and Learned Perceptual Image Patch Similarity (LPIPS) [38] as performance metrics for quantitative evaluations.

MSE is the most common estimator of image quality measurement metric. PSNR is used to calculate the ratio between the maximum possible signal power and the power of the distorting noise, which affects the quality of its representation. The ratio between two images is computed in decibel form. SSIM is a perception-based model that considers image degradation as the change in perception in structural information. VIF is derived from the quantification of two mutual information quantities, measured for the goal of the quality assessment of image. LPIPS is a perceptual distance that measures how similar two images are in a way that coincides with human judgment.

Table 3 shows that our method achieved the best values in the MSE, PSNR, SSIM, VIF, and LPIPS. The results were normalized and shown in a histogram in Figure 7. A lower value of the MSE and higher values of the SSIM, PSNR, and VIF indicated images of a higher quality. The results on the synthetic dataset thus verified that our method could adapt to different scenes featuring rain and yielded a better visual effect and higher image quality than the other methods.

Table 3. The average value of MSE, SSIM, PSNR, VIF, and LPIPS on Rain100 L dataset.

	Rainy Image	GMM [9]	UGSM [8]	DCNN [16]	FS-GAN [28]	Ours
MSE	0.0044	0.0013	0.0008	0.0014	0.0009	0.0006
SSIM	0.8292	0.8803	0.9236	0.9352	0.9143	0.9614
PSNR	24.77	29.55	31.63	28.97	31.61	33.64
VIF	0.4832	0.4935	0.5418	0.5542	0.5111	0.5916
LPIPS	0.2386	0.1672	0.1058	0.0516	0.0721	0.0350

4.2. Experiments on Real-World Dataset

The results of the methods on the real-world dataset are shown in Figure 8. To compare the results of processing more clearly, we also zoomed in on some areas of the images. The red frame represents the original image and the green frame the enlarged image. Our model could be well adapted to the rainy images under different conditions. When there were few rain streaks, our method and the DCNN method were better able to remove them. However, in the enlarged area, the image generated by the DCNN lost important details from the original. When the rain streaks were large in number, our method removed most of them, while each of the other methods left a certain residue in the resulting image. Our method thus removed more rain streaks and retained more image detail, and the resulting image was thus clearer.

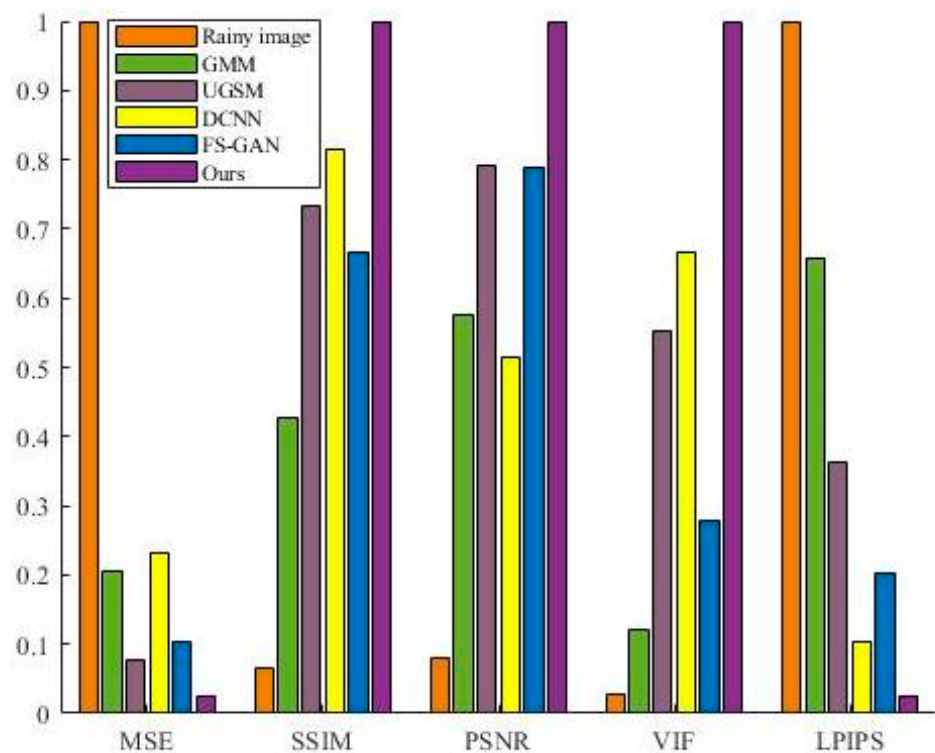


Figure 7. Display results of histogram of MSE, SSIM, PSNR, VIF, and LPIPS.

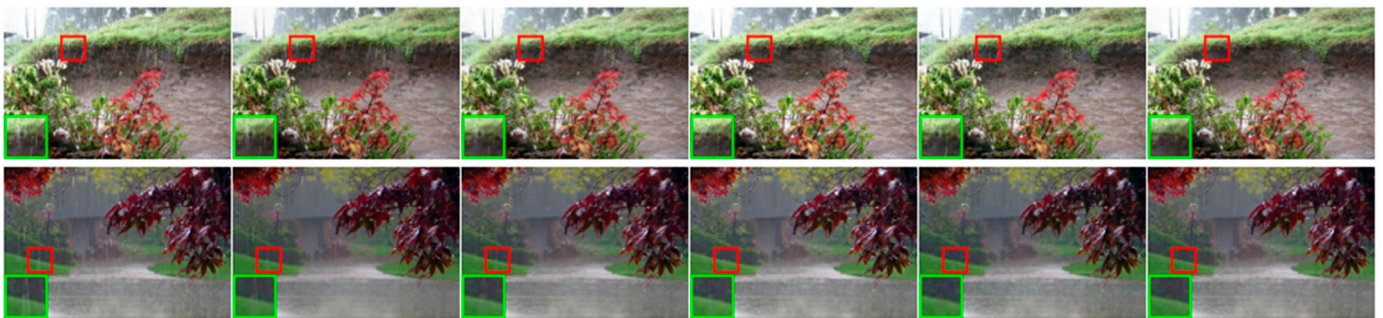


Figure 8. The de-rained results on real-world dataset. From the first column to the sixth column: Rainy image, GMM [9], UGSM [8], DCNN [16], FS-GAN [28], Ours.

4.3. Selection of Low-Pass Filters

In this section, we present a series of experiments on the effect of different low-pass filters. We use the Means Filter, Median Filter, Gaussian Filter, Non-local Means (NLM) Filter [39], and Bilateral Filter [40] on rain streak removal experiments.

The results of different filtering methods on a rainy image are shown in Figure 9. The contour information of the NLM filter and the bilateral filter are similar in the high-frequency part. In contrast, the other three filters contain more background information. We obtained the high-frequency part of the rain image by different filters with different window sizes and input them into the subsequent GAN. Then, we obtained the de-rained image from the proposed network and calculated the SSIM value for evaluation. The results are shown in Figure 10 (since the median filter and Gaussian filter had odd numbers of filter windows, their true filter windows were 19, 29, 39, and 49 when the window size was 20, 30, 40, and 50, respectively). The bilateral filter achieved the optimal SSIM value, and, at the filter window size of 50, the SSIM value hardly changed anymore. Therefore, in order to obtain the optimal experimental results, we chose to use a bilateral filter in this paper.

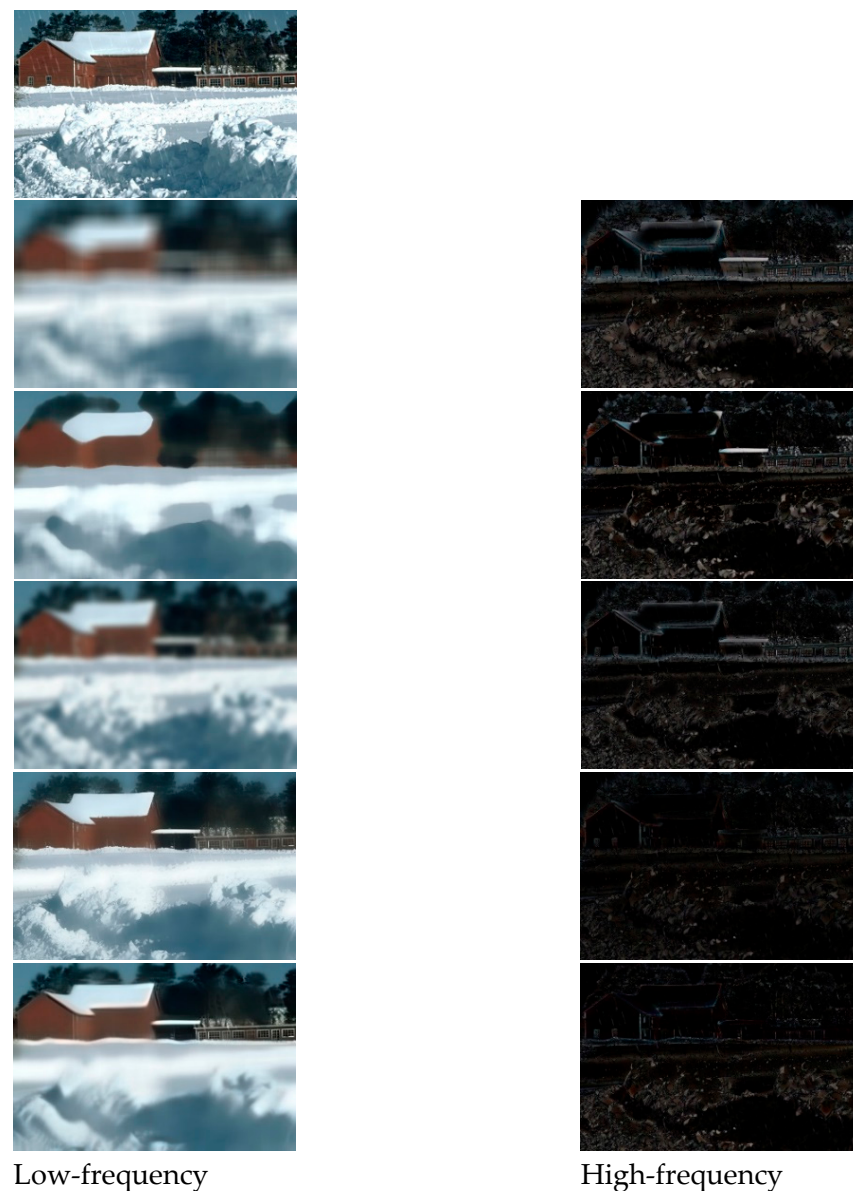


Figure 9. An example of the filtering results of different low-pass filters. The first line is the original image with rain, and the remaining lines are filtered from top to bottom to means filter, median filter, Gaussian filter, NLM filter, and bilateral filter.

4.4. Ablation Experiments

In this section, we analyze the effects of the bilateral filter and the proposed SSIM loss. The results of each section are shown in Figure 11. Where GAN indicates that the removal of bilateral filter and L_{SSIM} based on our proposed network; GAN + L_{SSIM} indicates that only L_{SSIM} is used; and GAN + Filter indicates that only bilateral filter is used.

We zoomed in on the two localized parts of the figure, and GAN removed some evident bright white streaks. After adding L_{SSIM} , the detailed information of the image is preserved, as shown in Figure 11d, but there is still a small amount of rain streaks, which can be a comprehensive measure of the similarity of the image, and retains as much detail information in the image as possible; after adding the bilateral filter, the rain streaks are all removed, but some detail information is lost. As shown in Figure 11e, it allows our network to exclude the background information and focus more on the removal of rain streaks to the maximum extent. By adding L_{SSIM} and a bilateral filter at the same time, the rain streak removal effect is better and closest to the ground truth image, as shown in Figure 11e. PSNR and SSIM also reached optimal values. The ablation experiments show

that our method not only removes rain streaks but also preserves the details of the image and generates a clearer image.

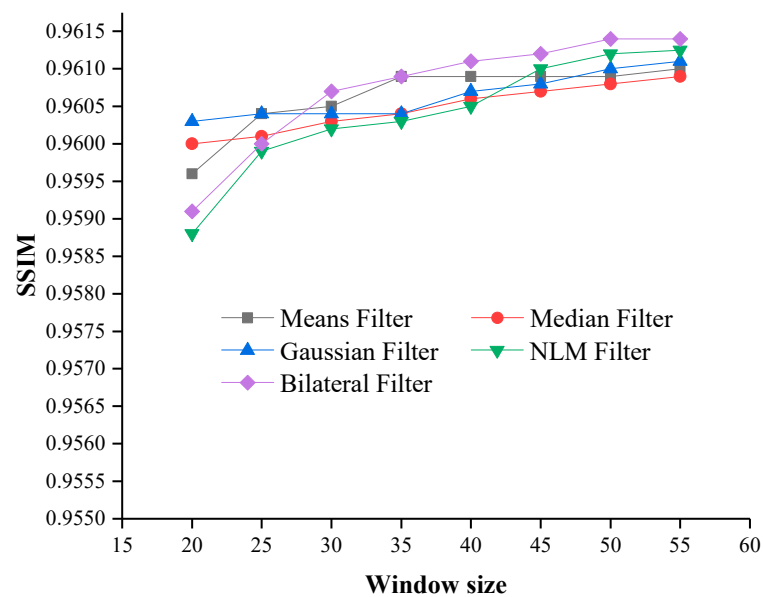


Figure 10. SSIM values of the final de-rained image obtained with different filters in different window sizes.

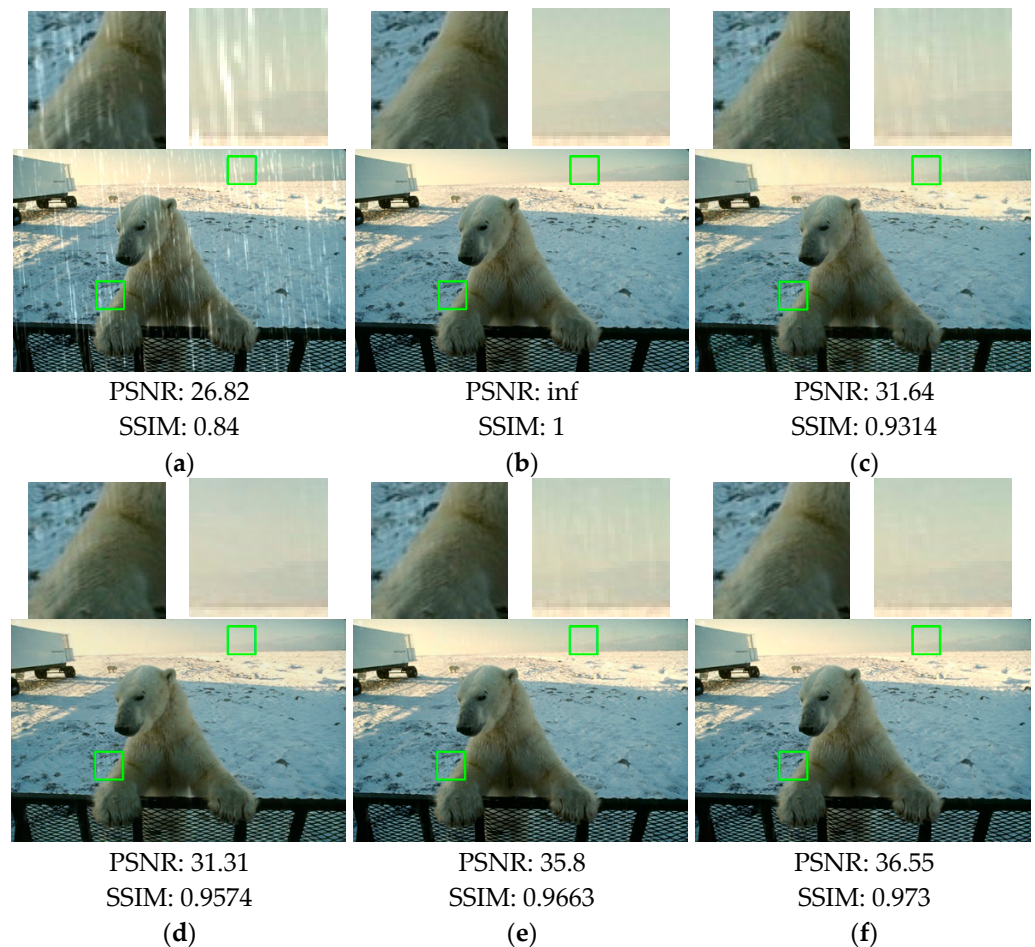


Figure 11. The de-rained results of ablation experiments. (a) Input image; (b) Ground truth image; (c) GAN; (d) GAN + LSSIM; (e) GAN + Filter; (f) Ours (GAN + LSSIM + Filter).

4.5. Runtime

We also tested the average running time of different methods. The resolution of the images was 320×480 , and the number of test images was 20.

The experimental results are shown in Table 4. GMM had the longest running time due to its high algorithmic complexity. Our method ran for 0.3 s, with a filter window size of 30, which achieved an optimal result. In addition, in cases where the image could be filtered in advance, our method only took 0.04 s to process an image without filtering time.

Table 4. Comparison of the average running time of different methods.

	GMM [9]	UGSM [8]	FS-GAN [28] (GPU)	Ours (GPU)
Time (s)	439.2	1.1	0.5	0.3

5. Conclusions

We proposed a method to remove rain streaks from images. Our method combined the bilateral filter and the generative adversarial network. Its rain streak removal network could restrain and train the high-frequency part of the image to preserve the background information. We also designed a loss function as an additional constraint on training. Our H-G discriminator comprehensively evaluated the generated image to render it as close as possible to the original, streak-free image. We assessed our method, in comparison with advanced methods, on two public datasets. Our method was universal and could better remove streaks from image than the other methods. The final results showed that our proposed method performed well in real-world and synthetic datasets, with a 0.9614 SSIM and a 33.64 PSNR value improvement. Our method could adapt to different scenes and yielded a better visual effect and higher image quality than other methods.

The proposed method also had the potential for use in the de-noising of images obtained by visual sensors to improve the stability and accuracy of the overall system algorithm. For the future work in this paper, we plan to apply more strategies to optimize the computational complexity of the algorithm and further shorten its running time. Furthermore, we plan to apply the proposed network to applications, including multi-object recognition, detection tasks, and vision tracking systems.

Author Contributions: Writing—original draft, Y.Y.; writing—review and editing, M.X. and Y.Y.; data curation, M.X.; investigation, C.C.; supervision, F.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partly supported by the National Natural Science Foundation of China, under Grant No. 41874212 and No. 61503350, and the Natural Science Foundation of Hubei Province of China, under Grant No. 2021CFB527.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We thank the authors of [9,18] for providing us with their experimental datasets, we also thank authors of [8,9,16,28] for providing program codes for our comparative testing.

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

References

1. Garg, K.; Nayar, S.K. Detection and removal of rain from videos. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, Washington, DC, USA, 27 June–2 July 2004; p. I-I.
2. Garg, K.; Nayar, S.K. Vision and rain. *Int. J. Comput. Vis.* **2007**, *75*, 3–27. [[CrossRef](#)]
3. Kim, J.H.; Sim, J.Y.; Kim, C.S. Video deraining and desnowing using temporal correlation and low-rank matrix completion. *IEEE Trans. Image Process.* **2015**, *24*, 2658–2670. [[CrossRef](#)] [[PubMed](#)]
4. You, S.; Tan, R.T.; Kawakami, R.; Mukaigawa, Y.; Ikeuchi, K. Adherent raindrop modeling, detection and removal in video. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 1721–1733. [[CrossRef](#)] [[PubMed](#)]

5. Bossu, J.; Hautière, N.; Tarel, J.P. Rain or snow detection in image sequences through use of a histogram of orientation of streaks. *Int. J. Comput. Vis.* **2011**, *93*, 348–367. [\[CrossRef\]](#)
6. Kang, L.W.; Lin, C.W.; Fu, Y.H. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Trans. Image Process.* **2011**, *21*, 1742–1755. [\[CrossRef\]](#)
7. Luo, Y.; Xu, Y.; Ji, H. Removing Rain from a Single Image via Discriminative Sparse Coding. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 13–16 December 2015; pp. 3397–3405.
8. Deng, L.J.; Huang, T.Z.; Zhao, X.L.; Jiang, T.X. A directional global sparse model for single image rain removal. *Appl. Math. Model.* **2018**, *59*, 662–679. [\[CrossRef\]](#)
9. Li, Y.; Tan, R.T.; Guo, X.; Lu, J.; Brown, M.S. Rain Streak Removal Using Layer Priors. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2736–2744.
10. Wu, C.; Ju, B.; Wu, Y.; Lin, X.; Xiong, N.; Xu, G.; Li, H.; Liang, X. UAV autonomous target search based on deep reinforcement learning in complex disaster scene. *IEEE Access* **2019**, *7*, 117227–117245. [\[CrossRef\]](#)
11. Palevičius, P.; Pal, M.; Landauskas, M.; Orinaitė, U.; Timofejeva, I.; Ragulskis, M. Automatic Detection of Cracks on Concrete Surfaces in the Presence of Shadows. *Sensors* **2022**, *22*, 3662. [\[CrossRef\]](#)
12. Pal, M.; Palevičius, P.; Landauskas, M.; Orinaitė, U.; Timofejeva, I.; Ragulskis, M. An Overview of Challenges Associated with Automatic Detection of Concrete Cracks in the Presence of Shadows. *Appl. Sci.* **2021**, *11*, 11396. [\[CrossRef\]](#)
13. He, R.; Xiong, N.; Yang, L.T.; Park, J.H. Using multi-modal semantic association rules to fuse keywords and visual features automatically for web image retrieval. *Inf. Fusion* **2011**, *12*, 223–230. [\[CrossRef\]](#)
14. Eigen, D.; Krishnan, D.; Fergus, R. Restoring an image taken through a window covered with dirt or rain. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 633–640.
15. Fu, X.; Huang, J.; Zeng, D.; Huang, Y.; Ding, X.; Paisley, J. Removing Rain from Single Images via a Deep Detail Network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1715–1723.
16. Fu, X.; Huang, J.; Ding, X.; Liao, Y.; Paisley, J. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Trans. Image Process.* **2017**, *26*, 2944–2956. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Yang, W.; Tan, R.T.; Feng, J.; Liu, J.; Guo, Z.; Yan, S. Deep joint rain detection and removal from a single image. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1357–1366.
18. Zhang, H.; Patel, V.M. Density-aware single image de-raining using a multi-stream dense network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 695–704.
19. Xia, H.; Zhuge, R.; Li, H.; Song, S.; Jiang, F.; Xu, M. Single Image Rain Removal via a Simplified Residual Dense Network. *IEEE Access* **2018**, *6*, 66522–66535. [\[CrossRef\]](#)
20. Li, X.; Wu, J.; Lin, Z.; Liu, H.; Zha, H. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 254–269.
21. Isola, P.; Zhu, J.-Y.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5967–5976.
22. Ding, H.; Sun, Y.; Wang, Z.; Huang, N.; Shen, Z.; Cui, X. RGAN-EL: A GAN and ensemble learning-based hybrid approach for imbalanced data classification. *Inf. Process. Manag.* **2023**, *60*, 103235. [\[CrossRef\]](#)
23. Zheng, Y.J.; Gao, C.C.; Huang, Y.J.; Sheng, W.G.; Wang, Z. Evolutionary ensemble generative adversarial learning for identifying terrorists among high-speed rail passengers. *Expert Syst. Appl.* **2022**, *210*, 118430. [\[CrossRef\]](#)
24. Zhang, H.; Sindagi, V.; Patel, V.M. Image De-Raining Using a Conditional Generative Adversarial Network. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *30*, 3943–3956. [\[CrossRef\]](#)
25. Pu, J.; Chen, X.; Zhang, L.; Zhou, Q.; Zhao, Y. Removing rain based on a cycle generative adversarial network. In Proceedings of the 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), Wuhan, China, 31 May–2 June 2018; pp. 2158–2297.
26. Qian, R.; Tan, R.T.; Yang, W.; Su, J.; Liu, J. Attentive Generative Adversarial Network for Raindrop Removal from a Single Image. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2482–2491.
27. Xiang, P.; Wang, L.; Wu, F.; Cheng, J.; Zhou, M. Single-image de-raining with feature-supervised generative adversarial network. *IEEE Signal Process. Lett.* **2019**, *26*, 650–654. [\[CrossRef\]](#)
28. Sharma, P.K.; Jain, P.; Sur, A. Dual-Domain Single image de-raining using conditional generative adversarial network. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 2796–2800.
29. Jin, X.; Chen, Z.; Li, W. AI-GAN: Asynchronous interactive generative adversarial network for single image rain removal. *Pattern Recognit.* **2020**, *100*, 107143. [\[CrossRef\]](#)
30. Chen, C.; Hao, L. Robust Representation Learning with Feedback for Single Image Deraining. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Online, 19–25 June 2021; pp. 7738–7747.
31. Zou, W.; Wang, Y.; Fu, X.; Cao, Y. Dreaming to Prune Image Deraining Networks. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022; pp. 6013–6022.
32. Chen, X.; Li, H.; Li, M.; Pan, J. Learning a Sparse Transformer Network for Effective Image Deraining. *arXiv* **2023**, arXiv:2303.11950.

33. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
34. Bauer, E.; Kohavi, R. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Mach. Learn.* **1999**, *36*, 105–139. [[CrossRef](#)]
35. Sara, U.; Akter, M.; Uddin, M. Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study. *J. Comput. Commun.* **2019**, *7*, 8–18. [[CrossRef](#)]
36. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]
37. Sheikh, H.R.; Bovik, A.C. Image information and visual quality. *IEEE Trans. Image Process.* **2006**, *15*, 430–444. [[CrossRef](#)] [[PubMed](#)]
38. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 586–595.
39. Buades, A.; Coll, B.; Morel, J.M. A non-local algorithm for image denoising. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; pp. 60–65.
40. Elad, M. On the origin of the bilateral filter and ways to improve it. *IEEE Trans. Image Process.* **2002**, *11*, 1141–1151. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.