

Research Article

Multistage Feature Complimentary Network for Single-Image Deraining

Kangying Wang  and Minghui Wang 

College of Computer Science, Sichuan University, Chengdu, Sichuan 610065, China

Correspondence should be addressed to Kangying Wang; 2019223049279@stu.scu.edu.cn

Received 12 November 2021; Accepted 29 November 2021; Published 16 December 2021

Academic Editor: Kaijian Xia

Copyright © 2021 Kangying Wang and Minghui Wang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Rain will cause the occlusion and blur of background and target objects and affect the image visual effect and subsequent image analysis. Aiming at the problem of insufficient rain removal in the current rain removal algorithm, in order to improve the accuracy of computer vision algorithm in the process of rain removal, this paper proposes a multistage framework based on progressive restoration combined with recurrent neural network and feature complementarity technology to remove rain streak from single images. Firstly, the encoder-decoder subnetwork is adapted to learn multiscale information and extract richer rain features. Secondly, the original resolution image restored by decoder is used to preserve refined image details. Finally, we use the effective information of the previous stage to guide the rain removal of the next stage by the recurrent neural network. The final experimental results show that a multistage feature complementarity network performs well on both synthetic rainy data sets and real-world rainy data sets can remove rain more completely, preserve more background details, and achieve better visual effects compared with some popular single-image deraining methods.

1. Introduction

When the computer vision systems work outdoors, they often suffer the effects of poor weather. Especially in rainy days, rain generally results in collecting images quality degradation and important information damage. This will degrade the performance of the computer vision systems or even break down and make it difficult to fully play a role in tasks such as object tracking [1] and object detection [2]. It is thus important to restore clean background images from degraded and rainy images.

In recent decades, many researchers have been paying attention to the deraining research and proposing a large number of methods, which mainly can be divided into two major directions: on the one hand, removing the rain in a video; on the other hand, deraining in a single image. Removing the rain in the video can use the temporal information between the frames of the video. Single rainy images not only lack the temporal information of frames but also have complex rain streaks with different sizes and directions

in a single rainy image. So it is more challenging to research single-image deraining.

Single-image deraining algorithms mainly include model-based methods and data-driven methods. Kang et al. [3] firstly extract the high-frequency features from the rainy images by a bilateral filter and then decompose the extracted high-frequency features into a rainy part and a nonrainy part by dictionary learning and sparse coding. Finally, the high-frequency features of the nonrainy part are combined with the low-frequency features to obtain the final deraining result. Kim et al. [4] research on the typical characteristics of rain streaks and design nonlocal means (NL-means) to remove rain streaks. Zhang et al. [5] are inspired by sparse coding and low-rank representation and propose a general convolution filter, a single-image deraining algorithm based on convolution coding.

In recent years, with the continuously improving computing capacity of computer hardware, deep learning technology has been popular with the researchers of computer vision gradually. In the current research on single-

image deraining, the algorithms by deep learning dominate the single-image deraining method research. Representative methods mainly include the convolutional neural network by combining rain streaks detection and deraining architecture by Yang et al. [6], which can effectively handle the scenario of heavy rain, overlapping rain streaks, and rain accumulation. Fu et al. [7] decompose each input image into low-frequency background features and high-frequency detail features by a low-pass filter and then input the high-frequency detail features to the CNN to restore the clean background image. About the training of high-frequency detail features, Fu et al. [8] are inspired by the deep residual network and introduce residual learning to predict residual, which improves the efficiency of network training greatly. Although the deraining algorithms based on deep learning have achieved great success, there is still plenty of room to improve.

By studying a large number of proposed methods, most of the methods they proposed adapt a one-stage design, which idea of architecture design comes from high-level visual tasks, such as residual learning [9] and dense skip connection [10]. But there are some general problems: blur background texture, inadequate deraining resulting in some rain remains, and others. On multistage methods, they have usually broken down the whole process into multiple stages to remove rain progressively. Since the rain removal information in the earlier stage is instructive for the rain removal in the later stage, it is necessary for these different stages to cooperate to complete the rain removal, which is exactly the problem not taken into account by the representative multistage rain removal methods [6, 11].

To remove rain streaks and preserve more background details, we are inspired by deep learning and progressive restoration [12, 13] and propose a multistage framework decomposing challenging restoration tasks into smaller subtasks. In our network, each subtask corresponds to each stage, and each stage has a different design. We use an encoder-decoder network [14] to learn the features of rain streaks with different sizes and directions and use a recurrent neural network to learn the cross-stage complementary features. Finally, we combine the features with the original resolution rain image to preserve richer local details.

2. Deraining Method

Many deraining algorithms assume that rain streaks are sparse and have similar characters in falling directions and shapes. However, the rain streaks in the real world are usually very complicated and are not necessarily satisfied with this hypothesis. Even in synthetic rainy images, the rain may have different directions and sizes and overlap with each other, so it is difficult to remove all rain streaks at once. The close and large rain streaks are often removed in most methods, while the small rain streaks in the distance are ignored. Because, in reality, the deraining model is unpredictable and is difficult to remove rain by the one-stage method, even parallel subnetwork [12].

2.1. Deraining Model. The model research of deraining generally decomposes the rain image O into a linear combination of nonrain streaks background layer B and rain streaks layer R as follows:

$$O = B + R. \quad (1)$$

Nonrain streak background layer B can be obtained by removing the rain streaks layer R from the rain image O .

Based on the deraining model in formula (1), in order to reduce the complexity of the model, we regard the similar rain streaks as one layer and then decompose the rain image into a combination of multilayer rain streaks layers and a clean background layer. The rain image model can be reformulated as follows:

$$O = B + \sum_{s=1}^S R_s, \quad (2)$$

where R_s represents the s -th rain streak layer in the stage s and S is the total number of stages.

2.2. Network Structure. In order to better adapt to the image rain removal task and model lightweight, the original basic components of the U-Net required are retained in the encoder-decoder network: there are two 3×3 conv in each layer of the encoder part (ReLU as the activation function), and 2×2 max pooling is used for downsampling. There are still two 3×3 conv in each layer of the decoder (ReLU as the activation function), and the upsampling is 2×2 deconv (transpose convolution). The encoder and decoder of the same layer use skip connection for feature concatenation.

On this basis, in order to better adapt to the multistage single-image rain removal task proposed in this paper, the original five-layer U-Net structure is reduced to three layers, which will minimize the amount of parameters in each stage of the network, and considering that the rain streak has different distributions in direction, color, and shape, the BN layer is removed (the normalization characteristics of BN layer are inconsistent with the features of rain map model proposed in this paper), which will also enhance the operation efficiency of the algorithm.

For the single-image rain removal task, the features of the rain streak are repetitive. By introducing RNN, the rain streak features extracted by ConvGRU in the encoder-decoder prediction process of the previous stage are fully utilized in the encoder or decoder prediction process of the later stage, which can make these features work together to capture the global features of the rain streak. In other words, ConvGRU is introduced to extract the rain streak feature information flow in the spatial dimension, so that these relevant context features have recurrent dependencies, so the dependent features can work together to extract the global streak features.

We propose a multistage rain removal network with feature aggregation in this paper shown in Figure 1, inspired by progressive restoration [13] to remove rain streak progressively. In the previous stage, we use encoder-decoder subnetwork [14] to encode multiscale information

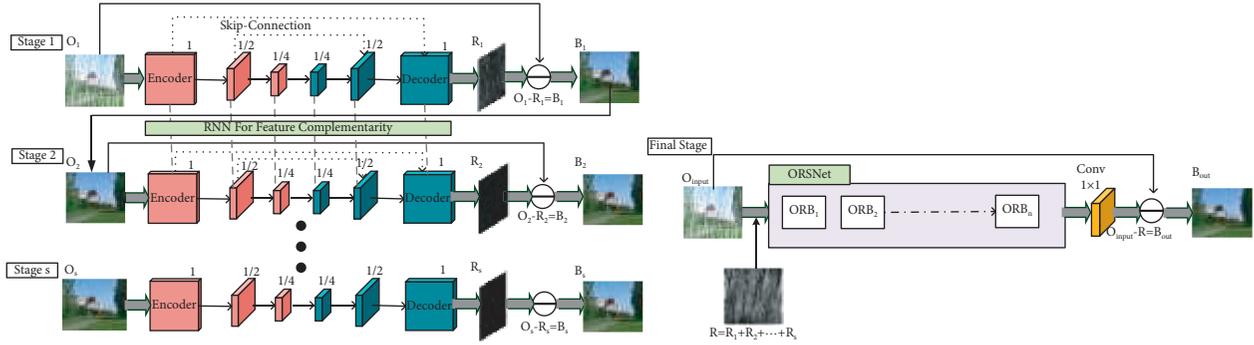


FIGURE 1: The structure of multistage feature aggregation network.

effectively. In the final stage, an original resolution subnetwork (ORS) is introduced to preserve the refine texture required in the final output image needed.

In the first stage of the network (RNN is not considered here), we can output a 256×256 feature map at the top layer of the decoder, which is recorded as R_1 . R_1 is combined with the rain map input O_1 in this stage to output the rain removed image, which is recorded as B_1 . This process completes the rain removal work in the first stage, that is, $B_1 = O_1 - R_1$.

After completing the rain removal work in the first stage, the rain removal output image B_1 is taken as the input of the next stage. For example, the input in stage s ($s > 1$) is the output B_{s-1} of stage $s-1$, which is recorded as O_s . In stage s , the rain map to be predicted is still input to the encoder-decoder subnetwork for further rain streak feature extraction. But different from the first stage, the convolution operation of each layer in the encoder and decoder network of stage s will cooperate with the rain streak feature extracted by RNN in the previous stage encoder-decoder subnetwork to capture the global rain line feature of the image. In other words, by introducing the gated recurrent unit (ConvGRU) in RNN to capture the rain streak feature information flow in the spatial dimension, these relevant context texture features have recurrent dependencies, so the dependent features can work together to better extract the global texture features.

Finally, the rain streak features have been fully extracted by combining the encoder-decoder network and RNN in the s stages before the last stage, but considering that due to the repeated downsampling operation in the encoder, they tend to lose spatial details. In order to preserve the fine details from the input image to the output image, in the last stage, we introduce the original resolution subnetwork and input the final predicted rain streak feature map into the original resolution subnetwork to generate rich spatial high-resolution features, so as to make up for the loss of spatial information. At the end of the whole network, the high-resolution rain line features obtained from the original resolution subnetwork are combined with the original rain map to obtain the final rain removal image.

Instead of using decomposition methods with artificial prior information to solve the problem of formula (2), the model intends to learn a function f , which maps the observed rain image O to the rain streak image R directly. Because the rain streak layer R is sparser than the observed rain image O

and has a simpler texture. Then we can obtain a nonrain streak background image B by subtracting the rain streak image R from the observed rain image O . The above function can be expressed as a deep neural network, which is learned by optimizing the loss function $\|f(O) - R\|_F^2$.

2.3. Multiscale Feature Extraction. For the single-image deraining task, the multiscale information from the input image has been proved important for the rain streaks recognition and removal task [14]. Because there may be more than one rain streak in the rain image, extracting the features of multiple scales and directions is more helpful for the description of rain. A module that can efficiently extract rain streak features is particularly important for the whole network. Therefore, we propose an encoder-decoder subnetwork based on U-Net to capture multiscale information of rain streaks. The encoder-decoder subnetwork is shown in Figure 2. For example, in the first stage of the network (the rain streak feature information of the previous stage supplemented by RNN in the subsequent stage is not considered here), if the input picture scale is 256×256 , the first convolution operation in the encoder will extract an original scale (256×256) rain feature. After two times of pooling (downsampling) and convolution, the feature map of 256×256 will become 128×128 and 64×64 . The deconvolution is applied in the decoder section to upsampling, so the previous 64×64 is upsampled to obtain a new 128×128 feature map. Then, the new 128×128 feature map is connected with another 128×128 feature map predicted by encoder, which we can get a 128×128 feature map of information fusion. After convolute the fused feature map, we get the final feature prediction map of a certain layer at the output of the decoder. Further using the same operation, we can output a 256×256 feature map at the top layer of the decoder. At the end of the decoder, through a 1×1 convolution kernel to recover the RGB channel of the color image or the gray channel of the gray image, which is recorded as R_1 . R_1 combined with the rain map input O_1 in this stage to output the rain removed image, which is recorded as B_1 . This process completes the rain removal work in the first stage, that is, $B_1 = O_1 - R_1$. In addition, the gray arrows in Figure 2 represent skip connections and aggregation operation.

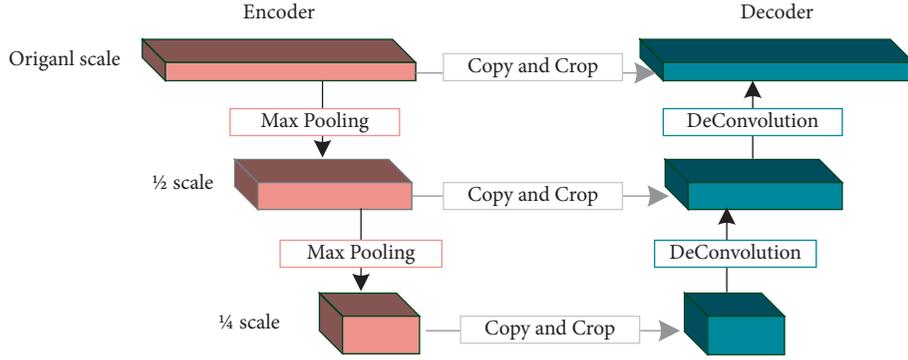


FIGURE 2: Encoder-decoder subnetwork.

Compared with the previous model [15], there is a big difference between our encoder-decoder network used in this module and others in that we do not adopt the batch normalization (BN) layer [9]. Although BN is widely used in deep neural network training and can reduce internal covariate shift of feature maps. Each scalar feature is normalized and has zero mean and unit variance by applying BN. However, these features are independent of each other and have the same distribution. In formula (2), the different layers of rain streak have different distributions in directions, colors, and shapes, and the same to each scalar feature of different rain streak layers. Therefore, BN contradicts the characteristics of our proposed deraining model; we remove BN from our model.

2.4. Multistage Feature Complement. Rain streaks may have different directions and overlap with each other, so it is difficult to remove them all at once. We send the preliminary results of rain streaks removal, their feature representation, and each predicted rain layer to the next stage for further refined restoration. Because the main information with obvious rain streaks (the closer and larger rain streaks) has been removed, the predictor of the next stage is better able to remove the remaining rain streaks [16]. This argument can be understood in different senses: (1) in the case of overlapping rain streaks mentioned previously, removing the nearest (and therefore brightest) rain streaks can reveal the darker rain streaks below the nearest rain streaks and (2) in the heavy rainy image, most rain streaks have relatively similar characteristics. After removing dominant rain streaks with similar direction or size, the other rain streaks will be detected and removed easily, which is inconsistent with the global pattern in size or direction.

Therefore, combined with the recurrent architecture, the process of rain streaks removal can be decomposed into multiple stages, which can be expressed as follows:

$$O_1 = O, \quad (3)$$

$$R_s = U_{\text{Codec}}(O_s), \quad 1 \leq s \leq S, \quad (4)$$

$$O_{s+1} = O_s - R_s, \quad 1 \leq s \leq S, \quad (5)$$

$$R = \sum_{s=1}^S R_s, \quad (6)$$

where S is the max number of the stage, R_s is the output of the s -th stage, O_{s+1} is the output image by the s -th stage rain streaks removal, and Codec means encoder-decoder subnetwork.

The above rain streaks removal model has been adopted in [6, 11]. But their methods only regard the recurrent structure as the same network with shared weights. Simultaneously, they only use the output feature map of the current stage as the input of the next stage and ignore the complementarity of different stages in their works. In our work, we use a multistage network structure to simulate the inverse process of rain formation to restore the nonrain streak image. With the gradual restoration of each stage, the interference degree of rain streaks to the image gradually decreases. So this is the reason why we introduce the recurrent neural network into our model. Formula (4) can be further expressed as follows:

$$R_s = U_{\text{Codec}}(O_s), s = 1; R_s = U_{\text{Codec+RNN}}(O_s, x_{s-1}), \quad 1 < s \leq S, \quad (7)$$

where the O_s represents the rain map input in the stage s . x_{s-1} generally refers to the different feature maps extracted by the encoder-decoder network in the convolution layer of different scales in the previous stage of the s stage. In the downsampling stage of the encoder, we will extract the rain line features of the original size, 1/2 size, and 1/4 of the input image. Similarly, three rain streak feature maps of different scales will be extracted in the upsampling process of the decoder. We use ConvGRU to cooperate with the encoder-decoder subnetwork in the next stage to guide the same layer encoder or decoder to predict the rain streak feature.

ConvGRU is a popular current unit in sequence model. Our network uses a convolution based on ConvGRU in the model. x_s^j represents feature maps of the s -th stage and the j -th layer and can be described by x_{s-1}^j (the feature map of the upper stage) and x_s^{j-1} (the feature map of the current stage) as follows:

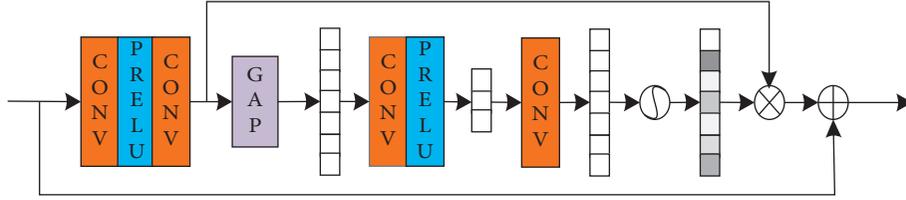


FIGURE 3: The structure of the original resolution model.

$$z_s^j = \sigma(W_z^j \# x_s^{j-1} + U_z^j \# x_{s-1}^j + b_z^j), \quad (8)$$

$$r_s^j = \sigma(W_r^j \# x_s^{j-1} + U_r^j \# x_{s-1}^j + b_r^j), \quad (9)$$

$$n_s^j = \tanh(W_n^j \# x_s^{j-1} + U_n^j \# (r_s^j \odot x_{s-1}^j) + b_n^j), \quad (10)$$

$$x_s^j = (1 - z_s^j) \odot x_{s-1}^j + z_s^j \odot n_s^j, \quad (11)$$

where σ is the sigmoid function $\sigma(x) = 1/(1 + \exp(-x))$ and \odot represent element multiplication.

2.5. Original Resolution Network. In the s stages before the implementation of the last stage, the rain streak features have been fully extracted by combining the encoder-decoder network and RNN. Although the model generates multiscale information effectively, it will tend to lose some spatial detail information by downsampling repeatedly. In order to preserve the refine details from the input image to the output image, we introduce the original resolution network (ORSNet) in the final stage.

We input the final predicted rain streak feature map into the original resolution subnetwork to generate rich spatial high-resolution features, so as to make up for the loss of spatial information. At the end of the whole network, the high-resolution rain line features obtained from the original resolution subnetwork are combined with the original rain map to obtain the final rain removal image.

ORSNet generates rich high-resolution spatial features without downsampling operations, which consist of many original resolution blocks (ORBs). The structure of the ORB is shown in Figure 3, where GAP represents global mean pooling.

3. Experiment

3.1. Training and Data Sets. To verify the effectiveness of our method, we test on many data sets, such as Rain100L, Rain100H, and Rain12 data sets. The three data sets are synthetic rainy image data sets and are used in deraining research popular. The typical characteristics of the Rain100L data set are thinner rain streaks and smaller raindrops. The rain streaks of the Rain100H data set have large sizes and different directions. The Rain12 data set is the synthetic rainy images but is the most similar to real-world rainy images by using rendering techniques in the process of synthetic. In terms of the real rainy image data set, we adopt the data sets Rain100L and Rain100H proposed by Yang et al. [6] to verify the deraining affection of our model in the real scene.

TABLE 1: Comparison of SSIM quality evaluation on three data sets.

Data sets	Yang et al. [6]	Fu et al. [8]	Zhang et al. [14]	Ours
Rain100L	0.9701	0.8447	0.8516	0.9729
Rain100H	0.7629	0.6954	0.7822	0.8683
Rain12	0.9568	0.9407	0.9130	0.9592

TABLE 2: Comparison of PSNR quality evaluation on three data sets.

Data sets	Yang et al. [6]	Fu et al. [8]	Zhang et al. [14]	Ours
Rain100L	35.96	32.57	25.84	36.09
Rain100H	24.15	23.60	23.94	24.82

Rain100L and Rain100H respectively contain 2,000 pairs of synthetic images (synthetic image pairs are composed of rainy images and their corresponding nonrainy images), from which 1,800 pairs of synthetic images are selected as the training data set and the remaining 200 pairs of synthetic images are selected as the test data set. There are only 12 pairs of synthetic rainy images in Rain12, so Rain12 is only used as a test data set in the experiment.

In our baseline, we reduced the number of layers of the original U-Net to 3, so that the encoder can obtain the feature map of the original scale, 1/2 scale, and 1/4 scale in the downsampling process. Similarly, only two deconvolution operations are required in the upsampling process. For the nonlinear operation, we use ReLU. We use Adam optimizer with a batch size of 8 for training on the NVIDIA 2080TI GPU; for optimization, the ADAM [17] algorithm is adopted with a start learning rate of 5×10^{-3} . During training, the learning rate is divided by 10 at every 20,000 steps. And the network is trained for 30 epochs with the above settings.

3.2. Comparison and Analysis of Experiment Results. To verify the deraining affection of the model we proposed, we compare the proposed method with JORDER [6], DDN [8], DIDMDN [14], and another representative single-image rain streaks removal method on synthetic data sets. Since there are corresponding nonrainy images in the synthetic data set, the index can use structural similarity metric (SSIM) and peak signal-to-noise ratio (PSNR) to evaluate the image quality and compare the deraining effects of different algorithms objectively. However, lacking corresponding real nonrainy images, the quality of rain removal is evaluated by a subjective visual affection on the test of the real rainy image data set.

3.2.1. Synthetic Data Sets. In Tables 1 and 2, the comparison results of SSIM and PSNR between the proposed method and other three popular methods [6, 8, 14] in three synthetic

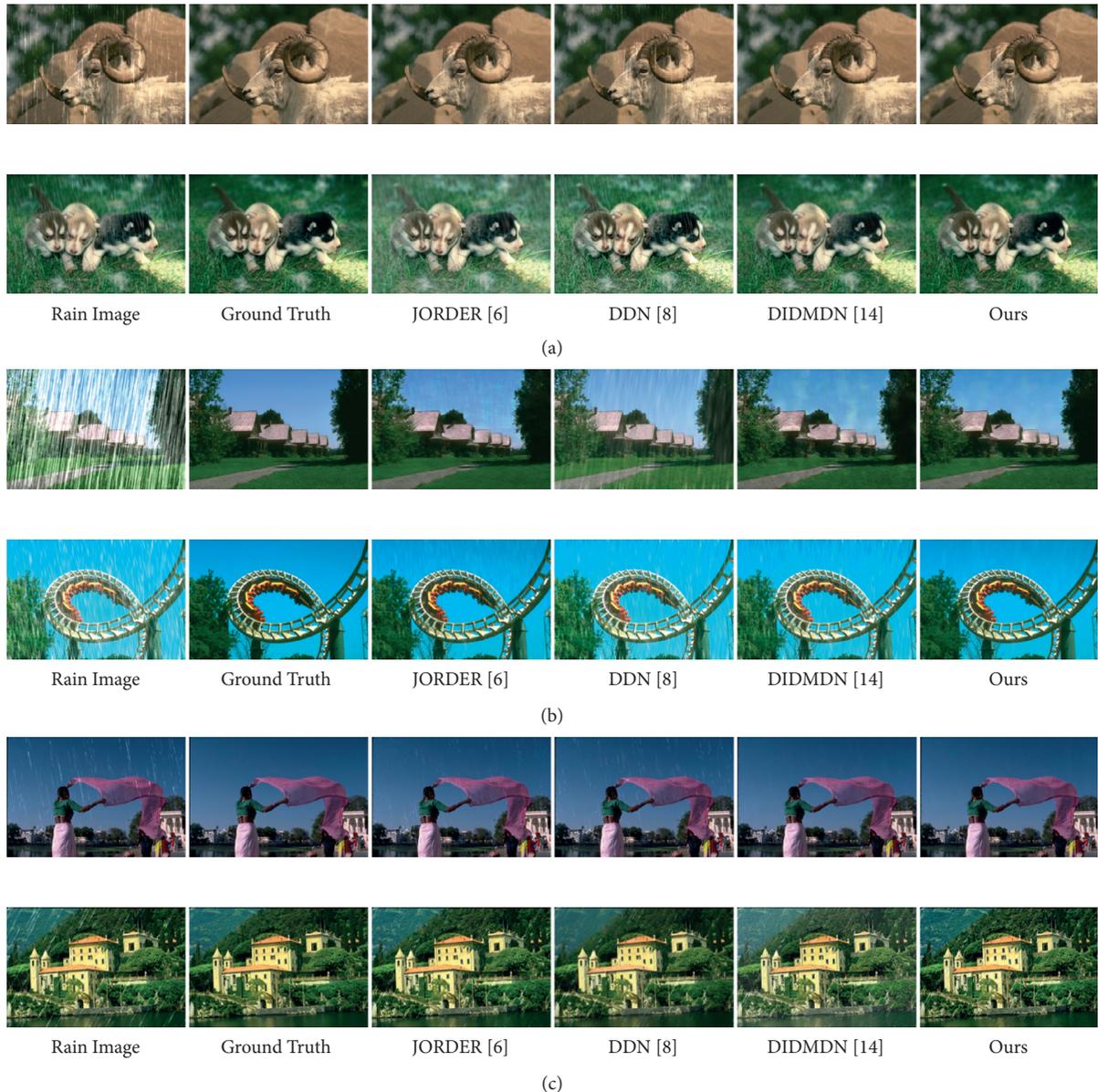


FIGURE 4: Contrast diagram of rain removal effect of each method in synthetic image: (a) the experiment results of Rain100L data set, (b) The experiment results of Rain100H data set, and (c) the experiment results of Rain12 data set.

data sets are shown separately. As we see from Table 1, the index SSIM of the method we proposed is superior to other methods on three data sets, especially in the Rain100H data set. It means that the method we proposed can remove rain from heavy rain, overlapping rain, and other complex conditions. In Table 2, the PSNR index of our method performs better than the method proposed in Fu et al. [8] and Zhang and Patel [14] on Rain100L and Rain100H data sets obviously but only slightly superior to the method in Yang et al. [6]. The fog removal algorithm is added in the process of rain removal in Yang et al. [6] to restore nonrainy image effectively.

Figure 4 shows the comparison of rain streaks removal effects, in which Figures 4(a)–4(c) are the experiments on Rain100L, Rain100H, and Rain12, respectively. From

Figure 4, we can know that the methods in [6] and [8] both have the obvious problem of rain streaks remaining and blur images in restored images. The method in [14] is more thorough in removing rain streaks but lost some details of goat horns. Our method removes all rain streaks and preserves sufficient image details. As seen from Figure 4(b) of Rain100H experiment results: compared with [8], the method in [6] removed more rain streaks, but there were still some rain streaks remaining and lost some refine details. The method in [14] could well identify and deal with heavy rain, but the image is a blurred image after removing rain streaks. Our method has no rain streaks remaining, and the restored image is clear. From Figure 4(c) of results on Rain12, the method in [14] still has obvious rain streaks remain in the skirt part of the image, while there is no obvious rain streaks



FIGURE 5: Comparison diagram of rain removal effect of each method on a real image.

remain in [6, 8], but there are still problems of detail loss and blur. However, the method in this paper almost removes all rain streaks and the image is clear.

Compared with single-stage rain removal networks such as DDN [8] and DIDMDN [14], although better spatial image details can be retained in the single-scale pipeline architecture, due to the limited receptive field, they cannot well detect the features of rain streak from different shapes and sizes, so that they always cannot fully remove rain streak. In contrast, our method recurrently uses the encoder-decoder subnetwork through the multistage network structure, which can better detect and remove all kinds of rain streaks. Compared with Jordan [6], which is a multistage rain removal network, Jordan [6] does not consider the relationship between rain streak features in each stage. Furthermore, our network saves the extracted rain streak multiscale information in the encoder-decoder subnetwork through RNN and uses it to guide the rain streak detection and removal in the later stage, which is very effective for fully removing rain streaks. In addition, in order to make up for the image information lost in the upsampling process in the encoder, the rain removal image output by the final stage through the original resolution subnetwork to ensure the information integrity and clarity of the image, which makes the method comparable to the single-scale pipeline architecture in spatial image details.

3.2.2. Real Scene Data Sets. To verify the practicability of the proposed method, three other popular rain removal methods and the method we proposed are tested on realistic rain image data sets. Since there are no nonrainy images for comparison in real scenes, the evaluation index of real rainy images is subjective visual effects. Figure 5 shows the rain removal effect of each method on the real rainy image. Compared with other methods, the method we proposed removes rain streaks more thoroughly and retains more details, resulting in the optimal visual effect after removing, which further verifies the practicability of the method proposed in this paper.

4. Conclusion and Discussion

A new single-image deraining method named multistage feature complimentary network based on progressive restoration has been proposed. The result shows that the method we proposed

can effectively deal with different rainfall scenarios; especially in the case of large rainfall and overlapping rain streaks, it can give full play to the performance of the network. Although the network has done the lightweight operation suitable for the rain removal task on the basic U-Net architecture, the network still appears redundant when the rain streaks almost are small scale or the difficulty of rain line removal is not high. The data sets used in the training of the network are pictures in sunny weather, so the difficulty of extracting rain streak features will be reduced. If the rain information is no longer the highlighted white rain bar in the common rain removal data set at night, fog, or under the reflection of light, the rain removal effect of the algorithm will not be ideal.

Because the number of improved network layers is still large, and the calculation speed is not well suitable for real-time applications, there are still many problems to be solved from the perfect real-time rain removal. Whether it is video image or single-image rain removal, there is not a large number of successful precedents in practical application. Due to the complexity of rainfall, it is currently impossible to use a network framework to carry out rain removal under various circumstances. Therefore, for different tasks, how to effectively integrate rain removal algorithms to improve the accuracy of computer vision algorithms is worth studying. For the landing application of rain removal algorithms, it involves the compression and acceleration of network models, which is also a hot research topic in image rain removal in recent years.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] Y. Sun and Y. Shi, "Adaptive decision multi-layer features fusion for object tracking," *Journal of Electronics & Information Technology*, vol. 41, no. 10, pp. 2464–2470, 2019.

- [2] Z. Guo, P. Song, and Y. Zhang, "Remote sensing image aircraft detection based on deep convolutional neural Network," *Journal of Electronics & Information Technology*, vol. 40, no. 11, pp. 2684–2690, 2018.
- [3] L. W. Kang, C. W. Lin, and Y. H. Fu, "Automatic single image-based rain streaks removal via image decomposition," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1742–1755, 2012.
- [4] J.-H. Kim, J.-Y. Sim, and C.-S. Kim, "Video deraining and desnowing using temporal correlation and low-rank matrix completion," *IEEE Transactions on Image Processing*, vol. 24, no. 9, pp. 2658–2670, 2015.
- [5] X. Zhang, H. Li, Y. Qi, W. K. Leow, and T. K. Ng, "Rain removal in video by combining temporal and chromatic properties," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 461–464, IEEE Press, Toronto, Ontario, Canada, July 2006.
- [6] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Deep joint rain detection and removal from a single image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1357–1366, IEEE Press, Honolulu, HI, USA, July 2017.
- [7] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: a deep network architecture for single-image rain removal," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2944–2956, 2016.
- [8] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1715–1723, IEEE Press, Honolulu, HI, USA, July 2017.
- [9] K. Jiang, Z. Wang, Yi Peng et al., "Multi-scale progressive fusion network for single image deraining," in *Proceedings of the CVPR*, Seattle, WA, USA, June 2020.
- [10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densel connected convolutional networks," in *Proceedings of the CVPR*, Honolulu, Hawaii, July 2017.
- [11] R. Li, L. F. Cheong, and R. T. Tan, "Single image deraining using scale-aware multistage recurrent network," 2019, <https://arxiv.org/abs/1712.06830>.
- [12] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha, "Recurrent squeeze-and-excitation context aggregation net for single image deraining," in *Proceedings of the IEEE European Conference Computer Vision*, Munich, Germany, September 2018.
- [13] S. Waqas Zamir, A. Arora, S. Khan et al., "CycleISP: real image restoration via improved data synthesis," in *Proceedings of the CVPR*, Seattle, WA, USA, June 2020.
- [14] H. Zhang and V. M. Patel, "Density-aware single image deraining using a multi-stream dense network," in *Proceedings of the IEEE Int'l Conference Computer Vision and Pattern Recognition*, Salt Lake City, Utah, USA, June 2018.
- [15] M. Lan and C. Li, "A hybrid neural network for image restoration," *In Computer Engineering and Applications*, vol. 54, no. 9, pp. 201–206, 2018.
- [16] Y. Zheng, X. Yu, M. Liu, and S. Zhang, "Residual multiscale based single image deraining," in *Proceedings of the BMVC*, Cardiff, UK, September 2019.
- [17] D. P. Kingma and B. Jimmy, "Adam: a method for stochastic optimization," 2014, <https://arxiv.org/abs/1412.6980>.