



## Towards Robust Rain Removal with Unet++

Boxia Hu<sup>1,2,\*</sup>, Yaqi Sun<sup>3</sup>, Yufei Yang<sup>1,4</sup>, Ze Ouyang<sup>3</sup> and Feng Zhang<sup>3</sup>

<sup>1</sup>School of Mathematics, Hunan University, Changsha, 410082, China

<sup>2</sup>College of Mathematics and Statistics, Hengyang Normal University, Hengyang, 421002, China

<sup>3</sup>College of Computer Science and Technology, Hengyang Normal University, Hengyang, 421002, China

<sup>4</sup>School of Mathematics, Changsha University, Changsha, 410022, China

\*Corresponding Author: Boxia Hu. Email: happylife008@163.com

Received: 07 September 2022; Accepted: 23 November 2022

**Abstract:** Image deraining has become a hot topic in the field of computer vision. It is the process of removing rain streaks from an image to reconstruct a high-quality background. This study aims at improving the performance of image rain streak removal and reducing the disruptive effects caused by rain. To better fit the rain removal task, an innovative image deraining method is proposed, where a kernel prediction network with Unet++ is designed and used to filter rainy images, and rainy-day images are used to estimate the pixel-level kernel for rain removal. To minimize the gap between synthetic and real data and improve the performance in real rainy image handling, a loss function and an effective data optimization method are suggested. In contrast with other methods, the loss function consists of Structural Similarity Index loss, edge loss, and L1 loss, and it is adopted to improve performance. The proposed algorithm can improve the Peak Signal-to-Noise ratio by 1.3% when compared to conventional approaches. Experimental results indicate that the proposed method can achieve a better efficiency and preserve more image structure than several classical methods.

**Keywords:** Rain removal; edge optimization; robust; Unet++

### 1 Introduction

Image rain removal is an image preprocessing method that deals with the inverse problem of removing the rain effect from an image, while highlighting its details to meet application-specific requirements and make it more suitable for human-machine recognition [1,2]. It is common to find rain in videos and images taken under bad weather conditions [3,4]. The presence of rain not only negatively affects the visual quality of an image or video, but also reduces the performance of application-specific tasks, such as object segmentation, recognition, tracking, and autonomous driving [5,6]. Therefore, video/image deraining is a crucial preprocessing step and important research area in the field of computer vision [7,8]. In the image rain removal task, many background elements and rain streaks are fused together, making them difficult to distinguish. When a convolutional neural network (CNN) is used to extract the rain trace information, it is easy to accidentally extract and remove



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

the background information. It is worth to mention that although generative adversarial networks (GANs) can reconstruct a more realistic image after rain removal [9,10], performance evaluation reveals that their performance is actually worse than the CNN-based reconstruction method. It can be inferred that many of the detail textures reconstructed by GANs are false and unreal.

In this paper, an innovative rain removal method is proposed, and a first attempt is made to investigate, improve and evaluate the robustness of rain removal methods. The main contributions are summarized as follows:

- An improved rain removal method is proposed based on Unet++ which achieves good performance in removing rain streaks from images and reducing their disruptive effects.
- A new loss function that includes Structural Similarity Index (SSIM) loss, edge loss, and L1 loss is adopted to improve the efficiency of the proposed approach for image deraining.

The remainder of this paper is organized as follows. In Section 2, we introduce some related work in this context. The framework of our method is presented in Section 3. We describe the proposed approach in Section 4. Experimental results are presented in Section 5, and conclusions are drawn in Section 6.

## 2 Related Work

Image deraining is the process of recovering a clear scene from the naturally blurred landscape due to bad weather conditions. Conventional image deraining algorithms generally rely on manual priors [11,12]. When the scene conditions do not meet these priors, such algorithms often produce an unrealistic output, which leads to non-ideal quality in the final restored image. In order to address this issue, with the great success of deep learning in the field of computer vision, numerous researchers have proposed various image rain removal methods based on CNNs [13,14]. A CNN is used to estimate the transmission map or directly predict the clear image [15,16]. These methods have been proved to be efficient and superior to priori-based methods, with significant performance improvement.

Single image rain removal algorithms are generally divided into conventional model-driven methods and data-driven deep learning methods [17,18]. Due to the strong automatic feature learning ability of deep networks, the single image rain removal algorithm based on deep learning has surpassed the conventional model-driven methods in recent years and caught the interest of field researchers [19,20]. To solve the issue of insufficient interpretability and incomplete integration with physical structures inside general rain streaks, Wang et al. proposed a model-driven deep neural network for single image rain removal [21]. Aiming to address real-world rain removal problems, namely the failure in handling real-world rainy scenes, Quan et al. proposed an innovative cascaded network architecture to remove rain streaks and rain drops [22]. It is well known that improving the performance of a single image rain removal algorithm based on deep learning mainly focuses on two aspects: the data quality of the rain map and the design of an image rain removal algorithm [23]. Numerous scholars have proposed new synthetic rain maps or rain map rendering methods to enhance the expression ability of rain map datasets. Wang et al. constructed the relatively real large-scale rain map dataset SPA-data through a semi-automatic method [24]. Ye et al. proposed a new framework which can jointly learn real rain generation and removal procedures [25]. Li et al. proposed a new deep network named REcurrent SE Context Aggregation Net (RESCAN). After decomposing the rain removal process into multiple stages, RESCAN incorporated a recurrent neural network (RNN) to preserve the useful information in previous stages and benefit the rain removal in later stages [26]. Guo et al. proposed a model-free deraining method named EfficientDeRain for the single-image deraining, which is over

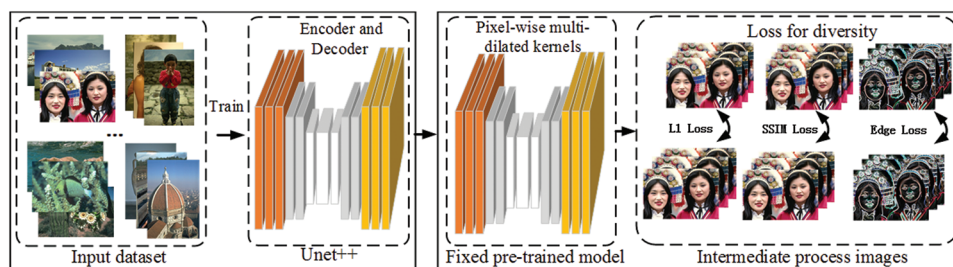
80 times faster than the state-of-the-art methods [27]. Li et al. developed a method named ESnet [28], which used layered Long Short-Term memory (LSTM) for recurrent deraining and fine-grained encoder feature refinement. Choose Rain100H as a testing dataset, Table 1 compares the outcomes of several popular neural network methods, including JORDER [9], PreNet [19], RCDnet [20], RESCAN [26], EfficientDeRain [27], and ESnet [28]. It can be seen from Table 1 that PreNet is able to achieve the best result in terms of PSNR, and RCDnet is able to achieve the best result in terms of SSIM. EfficientDeRain can achieve a good performance in terms of both PSNR and SSIM.

**Table 1:** Comparison of some common methods

Terms	Methods					
	JORDER [9]	PreNet [19]	RCDnet [20]	RESCAN [26]	EfficientDeRain [27]	ESnet [28]
PSNR	26.54	<b>30.5</b>	29.46	29.62	30.35	29.74
SSIM	0.8957	0.8974	<b>0.8994</b>	0.872	0.895	0.8968

### 3 Framework

The proposed deraining model is based on Unet++ and has two parts: training and optimization, as shown in Fig. 1.



**Figure 1:** Framework

Unet++ develops from Unet, which is closely similar to Fully Convolutional Networks (FCN) [29]. Unet was proposed shortly after FCN, but both were published in 2015 [30]. In the framework of the proposed approach, Unet++ is used to train the deraining model. With the increase in the number of network layers, the layer by layer downsampling will continuously incur information loss, so the small receptive field feature is used as supplement in order to increase the accuracy of location information. It can be used to achieve better results in rain removal as well. In the optimization step, a new loss function that aggregates three parts—SSIM loss, edge loss and L1 loss—is used in image deraining.

### 4 Deraining Model

#### 4.1 Pixel Level Extended Filtering Network with Unet++

Similar to Unet, the Unet++ model consists of two parts: left encoder part and right decoder part. The Unet++ architecture is composed of four repetitive structures: convolution, downsampling, upsampling, and skip sampling. The number of characteristic channels for each downsampling is

doubled. In the decoder part, similar to the coding layer, deconvolution also consists of four repeating structures. In Unet++, each point in horizontal direction is connected so that different feature levels can be captured [31]. The shallow layer is more sensitive to small targets while the deep layer is more sensitive to large targets, because the receptive fields at different depths have different sensitivities to targets of different sizes. The advantages of those operations can be integrated by concatenating the features.

Unet++ architecture is used to extract the features of the image. The proposed deraining network architecture is shown in Fig. 2. The *rainy day* image was used as input to estimate the pixel-level kernel  $G$  for rain removal, which is similar to the method in [32]. Through off-line training of clean image pairs on rainy days, the kernel prediction network can predict the spatial variable kernels of rain stripes with different thicknesses and intensities, while preserving the object boundary.

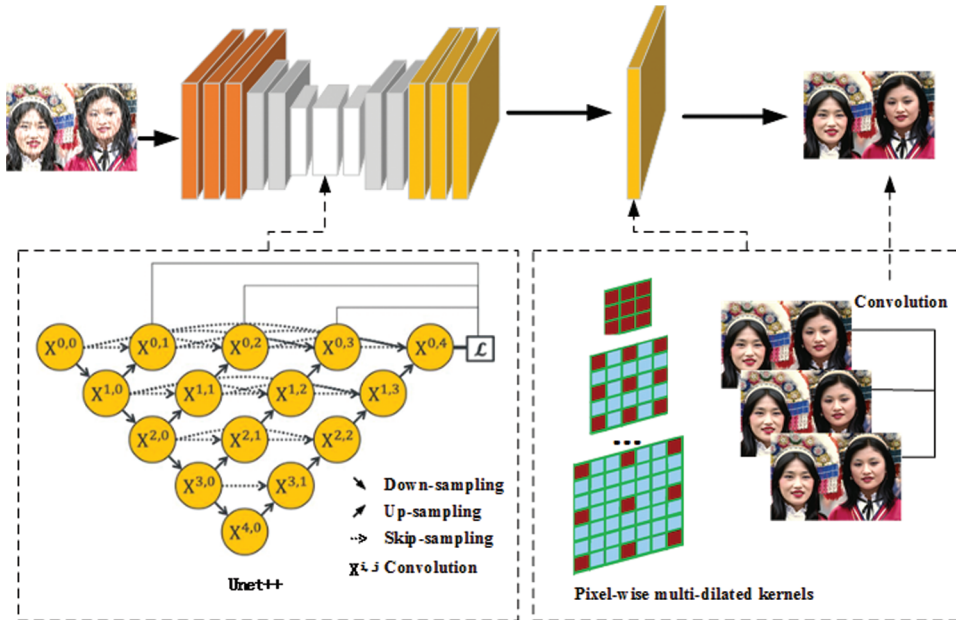


Figure 2: Rain removal network architecture

The pixel-level kernel can be estimated by:

$$G = UPN(\tilde{I}), \quad (1)$$

where  $\tilde{I}$  represents the input rainy image and  $UPN()$  denotes a network similar to UNet++. The pixel-level extended filtering network with Unet++ is shown in Fig. 2. Multiple magnified images are used for filtering and fusion. In order to enable the method to deal with multi-scale rain streaks without affecting efficiency, each prediction kernel is extended to three scales through extended convolution.

In order to reduce parameters and time costs, pixel-wise multi-dilated filtering for convolutional layer is suggested in [33,34]. This strategy is also adopted in the proposed approach. When the rain stripe covers a large area of the image, the large-scale kernel is used to effectively remove the rain from the relevant pixels that are far from the rain area. Through Eq. (2), the derained pixel can be predicted:

$$\hat{I}_l(g) = \sum_{s, h=g+s} G_g(s) \tilde{I}(h), \quad (2)$$



where  $g$  is defined as 2D coordinates for a pixel in the derained image and  $h$  is defined as 2D coordinates for a pixel in the rainy image,  $s$  is a step parameter which ranges from  $\left(-\frac{K-1}{2}, -\frac{K-1}{2}\right)$  to  $\left(\frac{K-1}{2}, \frac{K-1}{2}\right)$ , and  $l$  is a dilation scale factor. In the experiments, four scales are set, i.e.,  $l = 1, 2, 3$  and  $4$ . At last, a  $3 \times 3$  convolution kernel is used to fuse the four images and obtain the final output.

#### 4.2 Loss Function

Unlike some common methods which only consider one loss function, three loss functions are considered for training the proposed network, namely L1, SSIM and Edge loss functions. Given the derained image  $\tilde{I}$  and the clean image  $I$  as the real image, the proposed loss function is defined as:

$$\mathcal{L}(\tilde{I}, I) = \alpha \|\tilde{I} - I\|_1 - \beta SSIM(\tilde{I}, I) - \gamma \text{Edge}(\tilde{I}, I) \quad (3)$$

where  $\alpha = 0.9$ ,  $\beta = 0.1$  and  $\gamma = 0.15$  were empirically determined.

SSIM is an index to measure similarity of two images. Among the two images compared by SSIM, one is an original image and the other is a distorted image. The SSIM is defined as,

$$SSIM(X, Y) = \frac{(2u_X u_Y + C_1)(2\rho_{XY} + C_2)}{(u_X^2 + u_Y^2 + C_1)(\rho_X^2 + \rho_Y^2 + C_2)}, \quad (4)$$

where  $X$  and  $Y$  are two input images,  $u_X$  is the mean value of  $X$ ,  $u_Y$  is the mean value of  $Y$ ,  $\rho_X$  is the variance of  $X$ ,  $\rho_Y$  is the variance of  $Y$ ,  $\rho_{XY}$  is the covariance between  $X$  and  $Y$ , while  $C_1$  and  $C_2$  are two constants.

Typically, the purpose of edge detection is to significantly reduce the data size of the image while retaining the original image attributes. While numerous edge detection algorithms exist, Canny algorithm is very well-known and commonly used.

The Canny edge detection algorithm can be expressed as follows. Given an image  $I$ , it is first smoothed using a Gaussian filter. Then, the first-order partial derivative operator is used to find the partial derivative of the image along the horizontal direction  $E_x$  and the vertical direction  $E_y$ . The orientation of the gradient is calculated according to the following formula:

$$\theta = \arctan\left(\frac{E_y}{E_x}\right), \quad (5)$$

and the amplitude of the gradient is defined as:

$$E = \sqrt{E_x^2 + E_y^2}. \quad (6)$$

The non-maximum value of the gradient amplitude is suppressed, i.e., the maximum value of the local gradient is found.

#### 4.3 Algorithm

Our rain augmentation learning algorithm is shown in Algorithm 1.

In Algorithm 1, Dirichlet and Beta distributions are used to preprocess the weights. At each training iteration, we generate a rain map via function Rain\_aug and then augment it to the clean or rainy images. The new rainy images are then used for training the network and the fusion convolution layer.

**Algorithm 1:** Robust rain removal via Rain Augmentation

**Input:** Clean Images  $\Gamma$ , Real rain streak set  $\Lambda$ , Rainy image  $\Gamma^r$ ,  $UPN(\bullet)$ , fusion Convolution  $Conv(\cdot)$ , Loss function  $L$  and Operation set  $\Phi = \{rot, shear_{x/y}, trans_{x/y}, zoom_{x/y}\}$

**Output:** Pre-trained Network  $UPN(\bullet)$  and  $Conv(\bullet)$

Function  $Rain\_aug(\Lambda)$

Set a rain map  $\tau^0 \sim \Lambda$ ;

Set  $\tau^{aug} = \emptyset$ ;

Set augmentation weights  $(w_1, w_2, w_3, w_4) \sim D$ , Dirichlet distribution;

**For**  $i = 1$  to 4

    Set operations  $(\phi_1, \phi_2, \phi_3) \sim \Phi$ ;

    Combine via  $\phi_{1,2} = \phi_2\phi_1$  and  $\phi_{1,2,3} = \phi_3\phi_2\phi_1$ ;

    Update the Sample  $\Phi \sim (\phi_1, \phi_{1,2}, \phi_{1,2,3})$ ;

$\tau^{aug} = \tau^{aug} + w_i\phi(\tau^0)$ ;

    Set a blending weight  $w_i \sim B$ , Beta distribution;

    Return  $\Lambda = w\tau^0 + (1 - w)\tau^{aug}$ ;

**End For**;

**End Function**

**For**  $i = 1$  to  $|\Gamma^r|$

  Generate rain map via  $M = Rain\_aug(\Lambda)$ ;

  Set an image par via  $(I^r, I) \sim (\Gamma^r, \Gamma)$ ;

  Set  $R \sim (I^r, I)$  and  $I^m = M + R$ ;

  Predict pixel-wise kernels via  $G = UPN(I^m)$ ;

  Derain via Eq. (4) and  $\tilde{I} = Conv(\{\tilde{I}_l\})$ ,  $l = 1, 2, 3, 4$ ;

  Calculate Loss function in Eq. (3) and do iteration;

  Update parameters of  $UPN(\bullet)$  and  $Conv(\bullet)$ .

**End For**

## 5 Experimental Evaluations

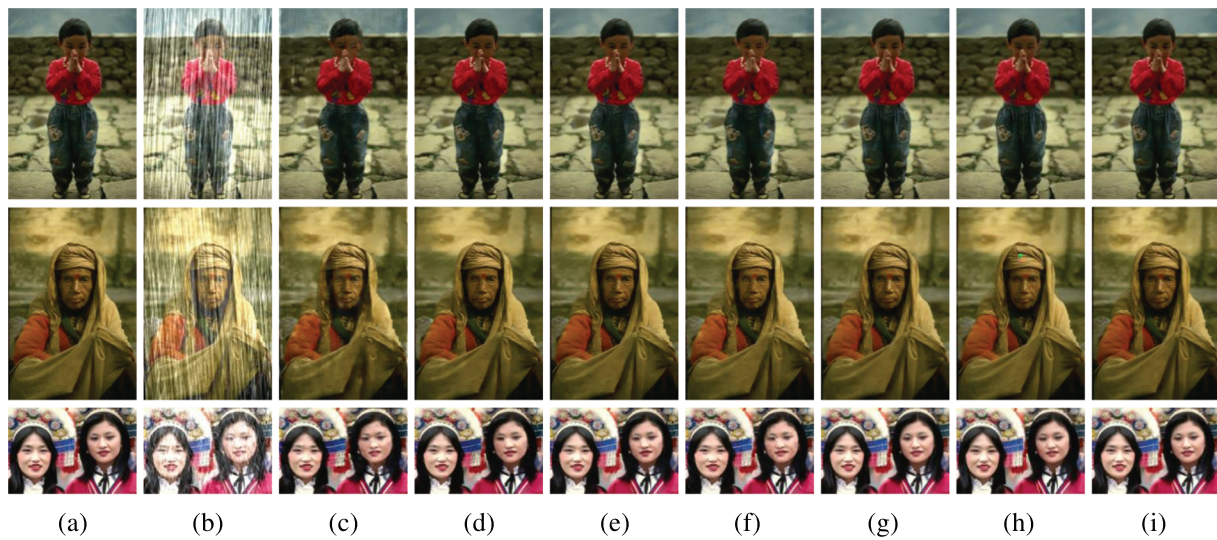
### 5.1 Evaluation Metrics

In our benchmark, two types of performance evaluation metrics are considered along with three datasets (Rain100H, Rain100L, Rain1400) [26,27]. Common quality measures including peak signal-to-noise ratio (PSNR) and SSIM are used for evaluating results fidelity.

### 5.2 Comparisons

The proposed approach will now be compared with several popular methods: JORDER [9], PreNet [19], RCDnet [20], RESCAN [26], EfficientDeRain [27] and ESnet [28], which are representative neural network methods. Some examples of deraining for person, landscape, and object images are shown in this section.

Fig. 3a shows three person images and Fig. 3b their rainy versions. One can see from Fig. 3 that the proposed method can recover the details of the human face and remove the rain from the image. Comparison results with regard to PSNR and SSIM are shown in Table 2. The best results are highlighted in bold. One can see from Table 2 that the proposed method obtained the best results.

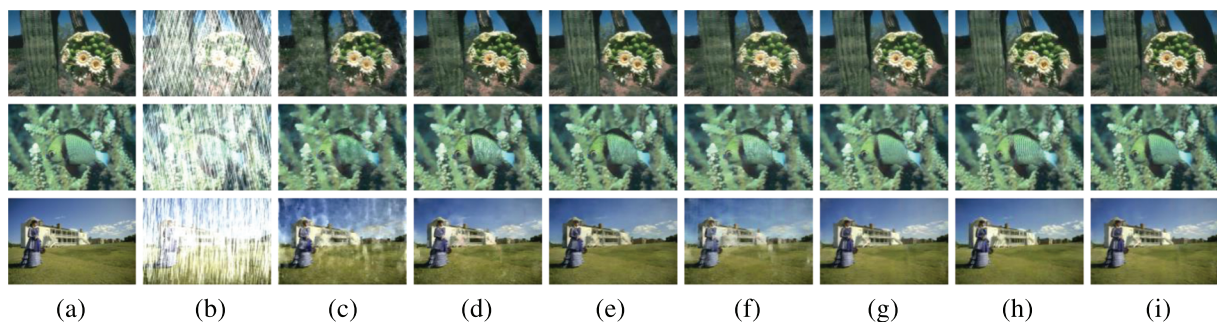


**Figure 3:** Qualitative results of different methods on several human images. From left to right: (a) Ground truth images, (b) Rainy image, (c)-(i) Rain removal results by JORDER, PreNet, RCDnet, RESCAN, EfficientDeRain, ESnet and proposed

**Table 2:** PSNR/SSIM results of different methods on the three images in Fig. 3b

Images	Methods													
	JORDER		PreNet		RCDnet		RESCAN		EfficientDeRain		ESnet		Proposed	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1	25.70	0.8127	26.44	0.8423	31.14	0.9159	28.66	0.8674	31.66	0.9208	25.37	0.8955	32.03	0.9285
2	25.35	0.7910	31.26	0.9193	31.93	0.9238	29.13	0.8770	31.46	0.9116	21.03	0.884	32.32	0.9377
3	23.65	0.8210	30.34	0.9301	26.47	0.8864	24.66	0.8241	32.10	0.9126	22.39	0.8137	32.79	0.9392

Fig. 4a shows different landscape images, and their rainy versions are shown in Fig. 4b. One can see from Fig. 4 that the proposed method can avoid mistakes in deraining. Comparison results in terms of PSNR and SSIM are shown in Table 3. The best results are highlighted in bold. One can see from Table 3 that the proposed method obtained the best results.

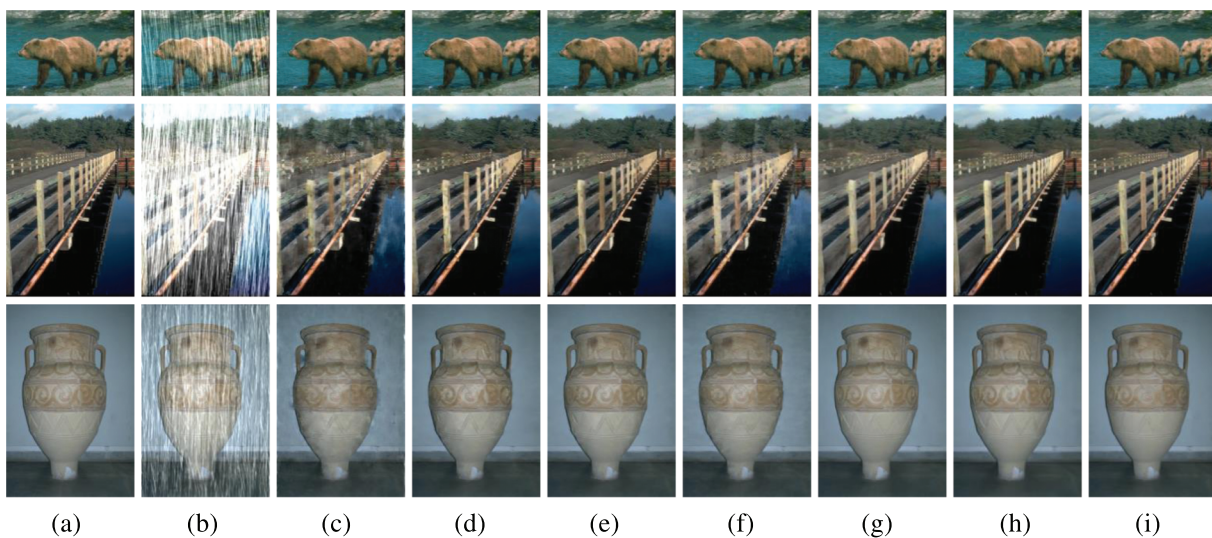


**Figure 4:** Qualitative results of different methods on several landscape images in Rain100H testing data set. From left to right: (a) Ground truth images, (b) Rainy image, (c)-(i) Rain removal results by JORDER, PreNet, RCDnet, RESCAN, EfficientDeRain, ESnet and proposed

**Table 3:** PSNR/SSIM results of different methods on the images in Fig. 4b

Images	Methods													
	JORDER		PreNet		RCDnet		RESCAN		EfficientDeRain		ESnet		Proposed	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1	20.85	0.6139	22.89	0.7544	24.45	0.7748	23.04	0.6941	24.79	0.7457	26.06	0.7091	<b>26.23</b>	<b>0.7867</b>
2	21.79	0.6958	24.15	0.8142	26.11	0.8459	24.13	0.7618	27.39	0.8592	23.29	0.8501	<b>27.9</b>	<b>0.8626</b>
3	18.12	0.6122	21.94	0.7811	23.54	0.7994	20.79	0.6910	24.00	0.7515	22.45	0.747	<b>24.52</b>	<b>0.8121</b>

Fig. 5a shows different object images, and their rainy images are shown in Fig. 5b. One can see from Fig. 5 that, in the proposed method, object characteristics (such as fur and edge) are better preserved after deraining. Comparison results of PSNR and SSIM values are shown in Table 4. The best results are highlighted in bold. One can see from Table 4 that the proposed method obtained the best results too.

**Figure 5:** Qualitative results of different methods on different objects. From left to right: (a) Ground truth images, (b) Rainy image, (c)–(i) Rain removal results by JORDER, PreNet, RCDnet, RESCAN, EfficientDeRain, ESnet and proposed**Table 4:** PSNR/SSIM results of different methods on the images in Fig. 5b

Images	Methods													
	JORDER		PreNet		RCDnet		RESCAN		EfficientDeRain		ESnet		Proposed	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1	29.06	0.9013	31.85	0.9456	33.35	0.9425	30.69	0.9027	35.79	0.9266	25.75	0.8114	<b>36.16</b>	<b>0.9497</b>
2	19.93	0.6571	23.58	0.8056	24.00	0.8105	21.48	0.7114	24.73	0.7940	23.46	0.8041	<b>25.12</b>	<b>0.8122</b>
3	29.62	0.8800	32.39	0.9370	34.77	0.9418	32.52	0.9140	34.69	0.9263	32.9	0.8947	<b>35.96</b>	<b>0.9488</b>

Rain100H was chosen as testing dataset. The results obtained by different methods are shown in Table 5. It can be noticed that the proposed method achieves the best results in terms of both PSNR and SSIM.

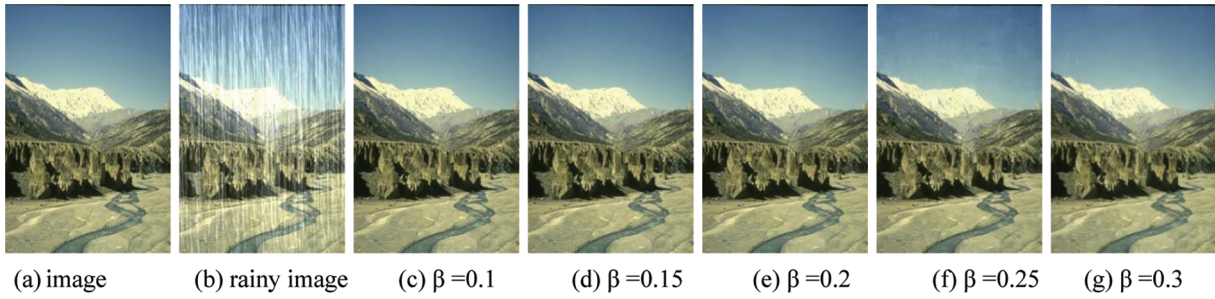
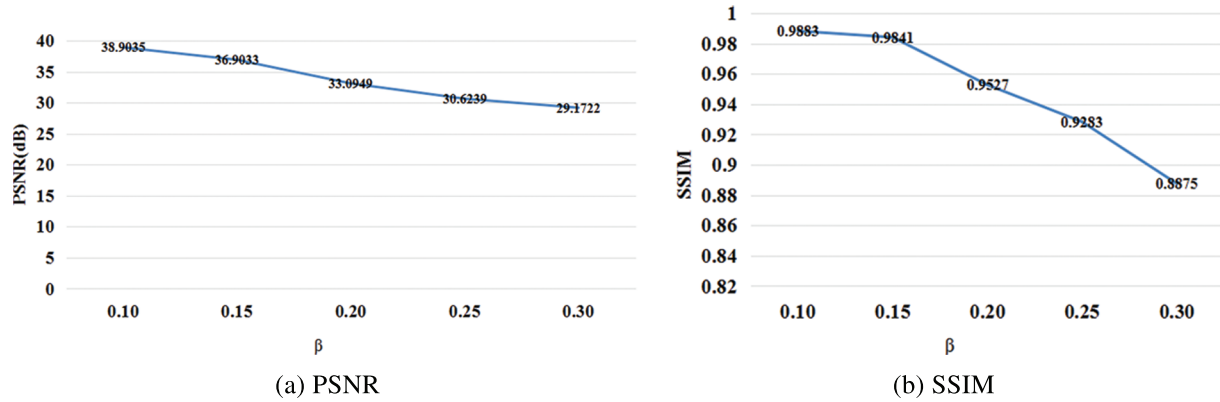


**Table 5:** Average PSNR/SSIM results of different methods on Rain100H testing dataset

Images set	Methods													
	JORDER		PreNet		RCDnet		RESCAN		EfficientDeRain		ESnet		Our	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1	26.54	0.8957	30.5	0.8974	29.46	0.8994	29.62	0.872	30.35	0.897	29.74	0.8968	<b>30.89</b>	<b>0.9087</b>

### 5.3 Ablation Studies

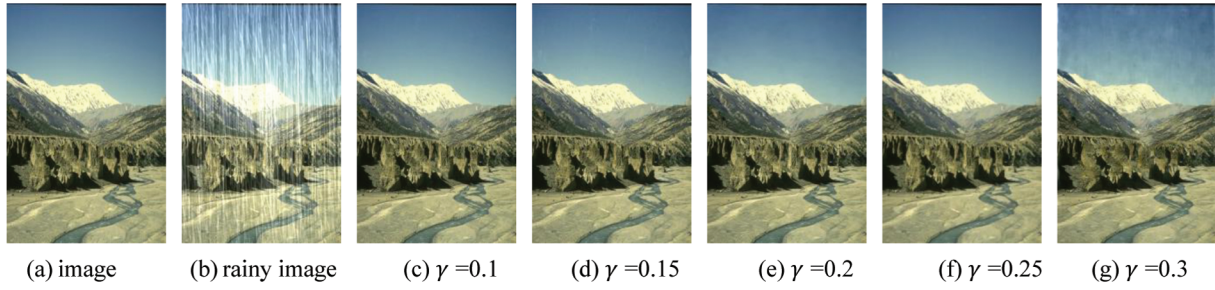
The loss function has three parameters:  $\alpha$ ,  $\beta$  and  $\gamma$ . We examine the impact of different strategies proposed in this paper for performance improvement. Further experiments are carried out, in which  $\beta$  and  $\gamma$  are modified while  $\alpha = 0.9$  is fixed. Fig. 6 demonstrates the effect of modifying  $\beta \in [0.1, 0.3]$ , while  $\gamma = 0.1$ . PSNR and SSIM results are shown in Fig. 7.

**Figure 6:** Result showing the effects of varying parameter  $\beta$ , while  $\alpha = 0.9$  and  $\gamma = 0.1$ **Figure 7:** Result showing the effects of varying  $\beta$ 

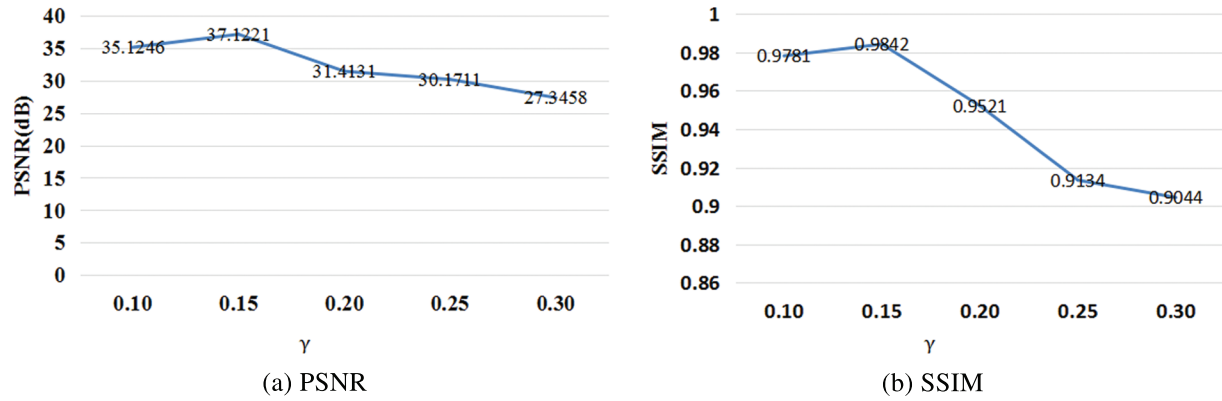
We can know from Fig. 7 that with the increasing noise in the parameter  $\beta$ , which represents the weight of SSIM loss, PSNR and SSIM values decrease. When  $\beta = 0.10$ , our method achieves the best PSNR and SSIM results.

Fig. 8 demonstrates the effect of modifying  $\gamma \in [0.1, 0.3]$ , and the corresponding PSNR and SSIM results are shown in Fig. 9.





**Figure 8:** Result showing the effects of varying parameter  $\gamma$  while  $\alpha = 0.9$  and  $\beta=0.1$



**Figure 9:** Result showing the effects of varying  $\gamma$

We can know from Fig. 9 that with the increasing noise in the parameter  $\gamma$ , which represents the weight of edge loss, PSNR and SSIM values fluctuate. When  $\gamma = 0.15$ , our method achieves the best PSNR and SSIM results.

Setting too large values for  $\beta$  or  $\gamma$  may lead to poor results. According to our experiments,  $\beta = 0.1$  and  $\gamma = 0.15$  achieve the best results.

## 6 Conclusions

A kernel prediction network with Unet++ is designed and used to filter rainy images. Then, to optimize the gap between synthetic and real data, a new loss function which consists of three parts, including SSIM loss, edge loss, and L1 loss, is further proposed along with an effective data optimization method that helps to improve the performance in real rainy image handling.

- Numerous examples of deraining for person, landscape, and object images demonstrated that our method is better than other common methods according to the PSNR and SSIM evaluation metrics.
- Experiments on datasets show that the proposed method outperforms the state-of-the-art methods under all evaluation metrics. It has improved nearly 1.3% from the conventional methods in terms of the PSNR evaluation metric.
- We also found that setting too large values for  $\beta$  or  $\gamma$  may lead to poor results. According to our experiments, the parameters in the loss function  $\alpha = 0.9$ ,  $\beta = 0.1$  and  $\gamma = 0.15$  achieve the best results.

Further research based on multi-head-attention in more complex environments and conditions is still necessary. This should be a next research priority.

**Funding Statement:** This work was supported by National Natural Science Foundation of China (61772179), Hunan Provincial Natural Science Foundation of China (2022JJ50016, 2020JJ4152), the Science and Technology Plan Project of Hunan Province (2016TP1020), Scientific Research Fund of Hunan Provincial Education Department (21B0649), Application-Oriented Characterized Disciplines, Double First-Class University Project of Hunan Province (Xiangjiaotong [2018]469), Discipline Special Research Projects of Hengyang Normal University (Grant No. XKZX21002).

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] H. Wang, Q. Xie, Y. Wu, Q. Zhao and D. Meng, "Single image rain streaks removal: A review and an exploration," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 4, pp. 853–872, 2020.
- [2] U. A. Bhatti, M. M. Nizamani and H. Mengxing, "Climate change threatens Pakistan's snow leopards," *Science*, vol. 377, no. 6606, pp. 585–586, 2022.
- [3] W. Yang, R. T. Tan, J. Feng, Z. Guo and S. Yan, "Joint rain detection and removal from a single image with contextualized deep networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 6, pp. 1377–1393, 2019.
- [4] W. Liu, J. Li, C. Shao, J. Ma, M. Huang *et al.*, "Robust zero watermarking algorithm for medical images using local binary pattern and discrete cosine transform," in *Proc. ICAIS*, Qingdao, SD, CN, pp. 350–362, 2022.
- [5] Y. Li, R. T. Tan, X. Guo, J. Lu and M. S. Brown, "Rain streak removal using layer priors," in *Proc. CVPR*, Las Vegas, NV, USA, pp. 2736–2744, 2016.
- [6] D. Yi, J. Li, Y. Fang, W. Cui, X. Xiao *et al.*, "A robust zero-watermarking algorithm based on PHTs-DCT for medical images in the encrypted domain," in *Proc. KES-InMed*, Singapore City, Singapore, SG, pp. 101–113, 2021.
- [7] H. Wang, Y. Wu, M. Li, Q. Zhao and D. Meng, "Survey on rain removal from videos or a single image," *Science China Information Sciences*, vol. 65, no. 1, pp. 1–23, 2022.
- [8] T. Li, J. Li, J. Liu, M. Huang, Y. W. Chen *et al.*, "Robust watermarking algorithm for medical images based on log-polar transform," *EURASIP Journal on Wireless Communications and Networking*, vol. 24, no. 1, pp. 1–11, 2022.
- [9] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo *et al.*, "Deep joint rain detection and removal from a single image," in *Proc. CVPR*, Honolulu, HI, USA, pp. 1357–1366, 2017.
- [10] Y. Fang, J. Liu, J. Li, D. Yi, W. Cui *et al.*, "A novel robust watermarking algorithm for encrypted medical image based on bandelet-DCT," in *Proc. KES-InMed*, Singapore City, Singapore, SG, pp. 61–73, 2021.
- [11] X. Fu, J. Huang, D. Zeng, Y. Huang and X. Ding, "Removing rain from single images via a deep detail network," in *Proc. CVPR*, Honolulu, HI, USA, pp. 3855–3863, 2017.
- [12] C. Zeng, J. Liu, J. Li, J. Cheng, J. Zhou *et al.*, "Multi-watermarking algorithm for medical image based on KAZE-DCT," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 10, pp. 1–9, 2022.
- [13] H. Zhang and V. M. Patel, "Density-aware single image deraining using a multi-stream dense network," in *Proc. CVPR*, Salt Lake City, UT, USA, pp. 695–704, 2018.
- [14] Y. Li, J. Li, C. Shao, U. A. Bhatti and J. Ma, "Robust multi-watermarking algorithm for medical images using patchwork-DCT," in *Proc. ICAIS*, Qingdao, SD, CN, pp. 386–399, 2022.
- [15] R. Qian, R. T. Tan, W. Yang and J. Liu, "Attentive generative adversarial network for raindrop removal from a single image," in *Proc. CVPR*, Salt Lake City, UT, USA, pp. 2482–2491, 2018.

- [16] U. A. Bhatti, L. Yuan, Z. Yu, J. Li, S. A. Nawaz *et al.*, “New watermarking algorithm utilizing quaternion Fourier transform with advanced scrambling and secure encryption,” *Multimedia Tools and Applications*, vol. 80, no. 9, pp. 13367–13387, 2021.
- [17] J. Chen, Z. He, D. Zhu, B. Hui, R. Yi *et al.*, “Mu-Net: Multi-path upsampling convolution network for medical image segmentation,” *CMES-Computer Modeling in Engineering & Sciences*, vol. 131, no. 1, pp. 73–95, 2022.
- [18] X. Xiao, J. Li, D. Yi, Y. Fang, W. Cui *et al.*, “Robust zero watermarking algorithm for encrypted medical images based on DWT-gabor,” in *Proc. KES-InMed*, Singapore City, Singapore, SG, pp. 75–86, 2021.
- [19] D. Ren, W. Zuo, Q. Hu, P. Zhu and D. Meng, “Progressive image deraining networks: A better and simpler baseline,” in *Proc. CVPR*, Long Beach, CA, USA, pp. 3937–3946, 2019.
- [20] H. Wang, Q. Xie, Q. Zhao and D. Meng, “A model-driven deep neural network for single image rain removal,” in *Proc. CVPR*, Seattle, WA, USA, pp. 3103–3112, 2020.
- [21] H. Wang, Z. Yue, Q. Xie, Q. Zhao, Y. Zheng *et al.*, “From rain generation to rain removal,” in *Proc. CVPR*, Los Alamitos, CA, USA, pp. 14791–14801, 2021.
- [22] R. Quan, X. Yu, Y. Liang and Y. Yang, “Removing raindrops and rain streaks in one go,” in *Proc. CVPR*, Los Alamitos, CA, USA, pp. 9147–9156, 2021.
- [23] L. Yu, Z. Qin, Y. Ding and Z. Qin, “MIA-UNet: Multi-scale iterative aggregation U-network for retinal vessel segmentation,” *CMES-Computer Modeling in Engineering & Sciences*, vol. 129, no. 2, pp. 805–828, 2021.
- [24] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang *et al.*, “Spatial attentive single-image deraining with a high quality real rain dataset,” in *Proc. CVPR*, Long Beach, CA, USA, pp. 12270–12279, 2019.
- [25] Y. Ye, Y. Chang, H. Zhou and L. Yan, “Closing the loop: Joint rain generation and removal via disentangled image translation,” in *Proc. CVPR*, Los Alamitos, CA, USA, pp. 2053–2062, 2021.
- [26] X. Li, J. Wu, Z. Lin, H. Liu and H. Zha, “Recurrent squeeze-and-excitation context aggregation net for single image deraining,” in *Proc. ECCV*, Munich, MU, GER, pp. 254–269, 2018.
- [27] Q. Guo, J. Sun, J. X. Felix, L. Ma, X. Xie *et al.*, “Efficientderain: Learning pixel-wise dilation filtering for high-efficiency single-image deraining,” in *Proc. AAAI*, Palo Alto, CA, USA, pp. 1487–1495, 2021.
- [28] Y. Li, Y. Monno and M. Okutomi, “Single image deraining network with rain embedding consistency and layered LSTM,” in *Proc. WACV*, Waikoloa, HI, USA, pp. 4060–4069, 2022.
- [29] O. Ronneberger, P. Fischer and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Proc. MICCAI*, Munich, MU, GER, pp. 234–241, 2015.
- [30] J. Long, E. Shelhamer and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. CVPR*, Boston, MA, USA, pp. 3431–3440, 2015.
- [31] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh and J. Liang, “Unet++: A nested u-net architecture for medical image segmentation,” in *Proc. MICCAI*, Granada, GRX, ES, pp. 3–11, 2018.
- [32] Y. Fisher and K. Vladlen, “Multi-scale context aggregation by dilated convolutions,” in *Proc. ICLR*, San Juan, PR, USA, pp. 1–13, 2016.
- [33] U. A. Bhatti, Z. Yu, J. Chanussot, Z. Zeeshan, L. Yuan *et al.*, “Local similarity-based spatial-spectral fusion hyperspectral image classification with deep CNN and gabor filtering,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, no. 9, pp. 1–15, 2021.
- [34] Z. Yue, J. Xie, Q. Zhao and D. Meng, “Semi-supervised video deraining with dynamical rain generator,” in *Proc. CVPR*, Los Alamitos, CA, USA, pp. 642–652, 2021.