Neurocomputing 467 (2022) 242–255

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Single image rain removal using recurrent scale-guide networks

Cong Wang [a,*], Honghe Zhu [b], Wanshu Fan [c], Xiao-Ming Wu [a], Junyang Chen [d]

[a] Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
[b] School of Mathematical Sciences, Dalian University of Technology, Dalian, China
[c] School of software Engineering, Dalian University, Dalian, China
[d] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China

ARTICLE INFO

Article history:
Received 6 September 2020
Revised 17 July 2021
Accepted 2 October 2021
Available online 7 October 2021
Communicated by Zidong Wang

Keywords:
Deraining
Deep-learning
Scale-guide
Recurrent networks

ABSTRACT

Recently, removing rain streaks from a single image has attracted a lot of attention because rain streaks can severely degrade the perceptual quality of the image and cause many practical vision systems to fail. Single image deraining can be served as a pre-processing step to improve the performance of high-level vision tasks such as object detection and video surveillance. In this paper, we propose recurrent scale-guide networks for single image deraining. Although the multi-scale strategy has been successfully applied to many computer vision problems, the correlation between different scales has not been explored in most existing methods. To overcome this deficiency, we propose two types of scale-guide blocks and develop two combinations between the blocks. One type of scale-guide block is that small scale guides the large, and the other is that large scale guides the small. Moreover, we extend the single-stage deraining model to the multi-stage recurrent framework and introduce the Long Short-Term Memory (LSTM) to link every stage. Extensive experiments verify that the scale-guide manner boosts the deraining performance and the recurrent style improves the deraining results. Experimental results demonstrate that the proposed method outperforms other state-of-the-art deraining methods on three widely used datasets: Rain100H, Rain100L, and Rain1200. The source codes can be found at https://supercong94.wixsite.com/supercong94.

1. Introduction

Images and videos taken under severe weather conditions such as rain and snow usually suffer from noticeable degradation of scene visibility. These degraded images severely affect the performance of many practical vision systems (e.g., image recognition systems, intelligent monitoring systems, object tracking systems, etc.) since they always require high-quality images for scene understanding and event detection. Single image deraining aims to recover a clean background image from the observed rainy image. It is a challenging task of image restoration, which has attracted much attention in the past decade. Hence, designing an efficient rain removal algorithm is a significant issue in computer vision and image processing.

In general, existing deraining methods can be roughly divided into two categories: video-based approaches and single image-based approaches. Video-based approaches [1–8] can utilize temporal information by analyzing the difference between adjacent frames, which makes the deraining problem much easier.In this paper, we explore the more challenging problem: single image deraining that aims to remove rain streaks from a single image.

Mathematically, a rainy image can be modeled as the linear combination of the rain-free image B and the rain streaks R:

$$O = B + R, \tag{1}$$

where O denotes the observed rainy image, B denotes the rain-free image, and R denotes the rain streaks. Eq. (1) is a highly ill-posed problem because many different pairs of B and R give rise to the same O, theoretically.

To make this problem well-posed, researchers attempt to design various priors about rainy images or rain streaks to constrain the solution space. Among them, image decomposition [11], sparse coding [9], low-rank representation [12], Gaussian mixture model [10], and bi-layer optimization model [13] are the most widely used priors in early deraining methods. Although these prior-based methods have shown good performance, they only work in certain conditions. This is because these priors are mainly based on some empirical observations of image properties, which may not hold in some complex cases, as illustrated in Fig. 1.

* Corresponding author.
E-mail address: supercong94@gmail.com (C. Wang).

https://doi.org/10.1016/j.neucom.2021.10.029
0925-2312/

Fig. 1. One example result on the real-world dataset compared with two prior-based deraining methods, DSC [9] and LP [10]. Prior-based methods fail to work in this case. The proposed method removes the majority of the rain streaks, while the prior-based methods generate unsatisfa.ctory results.

Recently, Convolutional Neural Networks (CNNs) have achieved tremendous progress in many computer vision tasks, e.g., object detection [14,15], object tracking [16], pose estimation [17], label classification [18], domain adaption [19,20], optical flow [21], semantic segmentation [22], photometric stereo [23–28], specular highlight detection and removal [29–31], deblurring [32], dehazing [33–37], super-resolution [38–40] and deraining [41–71]. Benefiting from the powerful ability of feature representation of CNNs, CNN-based deraining methods have achieved better performance than prior-based methods. These methods focus on designing various deep networks to learn the transformation from the input image to the rain streaks and then obtain the rain-free image via Eq. (1). Among them, light-weight models [57,51] are proposed to improve the computational efficiency, and multi-stream deraining net [47] is developed to guide the deraining process by estimating the rain density. Multi-stream [43,47], channel attention via squeeze-and-excitation [72], pixel-wise attention using nonlocal mean network [73], and adversarial learning [44,54,49] are also proposed to solve various deraining problems. Although tremendous improvements have been made, there are some important issues remained to be explored.

The multi-scale strategy aims to utilize images or features at different scales to exploit information from different levels, which is often implemented by pooling operations with different kernels and strides. Although this manner is simple, it achieves satisfactory results in many cases. Recently, some researchers combined the properties of traditional spatial pyramid architectures and convolutional neural networks to deal with various practical vision tasks. However, little research effort has been devoted to using the multi-scale strategy for the image deraining problem. Most existing methods [43,47,49–52] neglect information at different scales that are worth exploring and do not consider how to apply the multi-scale strategy to the deraining task. Moreover, how to explore the correlation between different scales also remains to be studied for the deraining task.

To address these issues, we propose a novel deep-learning-based method, which is based on recurrent networks. Firstly, we introduce two types of scale-guide blocks. One is that the small scale guides the large scale, and the other is that the large scale guides the small scale. The scale guidance can link the features between different scales, and most importantly, it can boost the deraining performance. Secondly, we extend the single network to the recurrent networks, using Long Short-Term Memory (LSTM) to link every stage. On the premise of not increasing the size of the model, the parameters are shared between different stages.

Our main contributions are as follows:

- We propose two types of scale-guide blocks. One is that small scale guides the large, and the other is that large scale guides the small. The scale guidance manner links the features between different scales and boosts the deraining performance. To the best of our knowledge, this is the first work to consider the scale guidance manner in the multi-scale methods for single image deraining.
- We conduct quantitative and qualitative experiments on both synthesized datasets and real-world rainy images to demonstrate that the proposed method performs favorably against state-of-the-art deraining methods.

2. Related work

In this section, we briefly review state-of-the-art deraining methods and then introduce the multi-scale manner applied in some vision tasks.

2.1. Single image deraining

Most of the existing single image deraining methods can be roughly divided into two categories: prior-based methods and deep-learning-based methods.

2.1.1. Prior-based methods

Single image deraining is a classical ill-posed problem, most early deraining methods attempt to explore additional prior information to make the problem more tractable. Kang et al. [11] regard rain streaks as high-frequency structures and decompose the rain image into the low-frequency and high-frequency layer and apply dictionary learning to remove rain streaks in the high-frequency layer. Similarly, Luo et al. [9] propose a discriminative sparse coding framework based on image patches and separate rain streaks from background images. Chen and Hsu [12] present a low-rank model from matrix to tensor structure in order to capture the correlated rain streaks and use this model to remove rains streaks from images in a unified way. Kim et al. [74] detect the rain streaks and remove them by the non-local mean filter. Wang et al. [75] design a 3-layer hierarchical scheme to remove rain or snow from a single image. Zhu et al. [13] further propose a joint optimization process by considering the directions of the rain streaks into the deraining problem.

2.1.2. Deep-learning-based methods

In recent years, several deep-learning-based deraining methods have made remarkable progress in image deraining problem. In [42], deep-learning is firstly introduced to solve single image deraining problem. In this work, a input rainy image is decomposed into a background-based layer and a detail layer separately, and then the rain streaks are removed from the detail layer by a learnable CNN-based nonlinear mapping. Yang et al. [43] propose a recurrent network to jointly detect and remove rain streaks step by step. Considering the hazy condition in the rainy model, they apply a dehazing-deraining-dehazing algorithm to solve the complex situation. Li et al. [46] design a multi-scale non-local enhanced encoder-decoder network, which maps rainy images to clean ones by considering pixel-wise attention mechanism. Fan

et al. [51] develop a lightweight residual-guide network to recover the clean background image. Li et al. [45] recurrently utilize the convolutional neural network with dilation factors and squeeze-and-excitation [72] blocks to remove heavy rain streaks. Zhang et al. [47] propose a multi-stream densely connected convolutional neural network (DID-MDN) for jointly rain-density estimation and deraining. Pan et al. [50] present a dual convolutional neural network to learn jointly the rain streaks and rain-free images based on the physical model Eq. (1). Ren et al. [53] consider network architectures, inputs, outputs, and loss functions and provide a better baseline deraining network. Wang et al. [52] propose a spatial attentive network to remove the rain streaks in a local-to-global manner. Moreover, they also develop a semi-automatic method that incorporates temporal priors and human supervision to generate the clean image from each input sequence of the real-world rain images. Specially, Li et al. [76] give a comprehensive benchmark analysis among the existing single image deraining algorithms. Moreover, the generative adversarial networks [44,49,54] are also applied to the deraining tasks.

2.2. Multi-scale manner

Multi-scale manner has been applied to several computer vision tasks, such as object detection [77], pose estimation [17], optical flow [21], scene parsing [78], and visual recognition [79]. The pooling operation is often used to obtain multi-scale feature information. Although the multi-scale manner presents strong effectiveness in these tasks, the correlation between different scales is always ignored. Different scales have different contextual information. How to effectively utilize this information needs to be explored.

In this paper, the scale-guide model is proposed to consider the correlation between different scales for image deraining.

3. Proposed method

In this section, we introduce the details of the proposed method. We describe the overall network framework in Section 3.1. Section 3.2 presents the scale-guide block, including S2L block (small scale guides the large) and L2S block (large scale guides the small). We provide more analysis on various recurrent units in Section 3.3 and present the loss function in Section 3.4.

3.1. Overall network framework

Fig. 2 illustrates our proposed overall network framework, which is the multi-stage networks. We use Long Short-Term Memory (LSTM) to link every stage, while the parameters are shared for all stages to improve the deraining performance with the same model size. For every stage, the network consists of an encoder stage and a decoder stage with LSTM. For the encoder stage, firstly, the input rainy image is converted into the feature space, i.e., the enter layer in Fig. 2. Then, the features are encoded by several of our proposed scale-guide blocks (SGBs) with several pooling operations. We will give a detailed analysis of the scale-guide block in Section 3.2. For the decoder stage, a symmetric structure is adopted. Skip connections between encoder and decoder are applied to enable the computation of long-range spatial dependencies as well as efficient usage of the feature activation of proceeding layers. As the rain streaks have relatively simple structures, they are easier to learn than the background images. The proposed network aims to learn a non-linear f, which directly describes the mapping relationship between the rainy image and the rain streaks. When the rain streaks are obtained, the final rain-free image can be computed by subtraction operation via Eq. (1). Figs. 3 and 4.

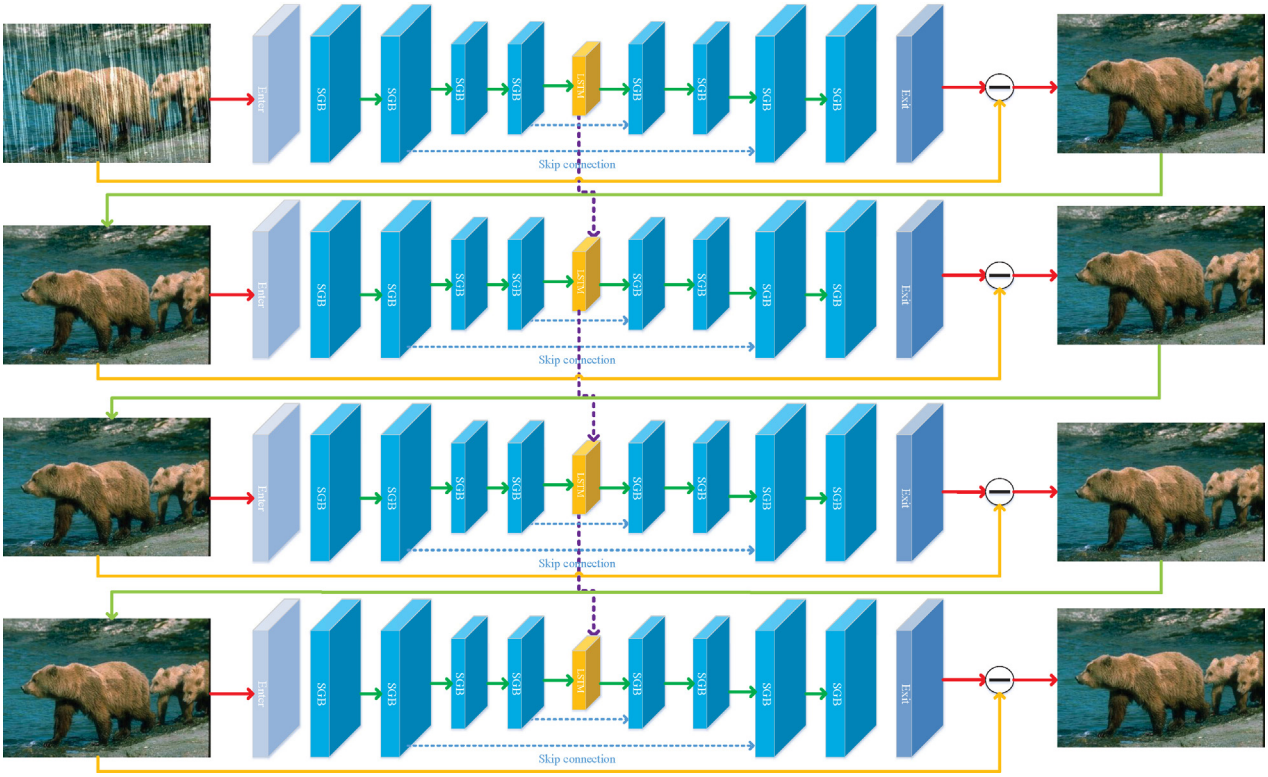


Fig. 2. Overall network framework. SGB denotes the scale-guide block that is the concatenation of S2L (Fig. 3) and L2S (Fig. 4) blocks..

3.2. Scale-guide block

Many works have shown that the multi-scale method is effective for various computer vision tasks. However, most existing multi-scale strategies only implement the multi-scale manner by fusing the output features of different scales, e.g., [36] [44]. These methods do not analyze the correlation between different scales, which should be explored. Hence, it is necessary to introduce the multi-scale learning manner to handle the single image deraining problem. Furthermore, we visualize the feature maps at different scales and their fusion outputs. We find that they have different rain streak structures at different scales and they can obtain better rain streak structures after fusing these convolutional features guided by scale manner. More analysis can be found in Section 4.5.1.

To the best of our knowledge, there is no literature paying attention to the correlation between different scales. While the relevancy between different scales has not been explored, It should be introduced into deep-learning-based methods as a guide function.

In this paper we design two types of scale-guide block: (a) small scale guides the large, (b) large scale guides the small, which are denoted as S2L block and L2S block, respectively.

3.2.1. S2L block

We propose the S2L block that small scale guides the large. Features at different scales have different structure and these features should be fused to further enhance rainy features. Mathematically, the S2L block is formulated as follows. Firstly, we utilize the *Pooling* operation with different size of kernels and strides to obtain the multi-scale features:

$$y_i = \mathcal{P}_i(x), \quad i = 1, 2, 4, \tag{2}$$

where x, y_i denote the input and output, respectively. $\mathcal{P}_i$ denotes the pooling operation with $i \times i$ kernels and strides.

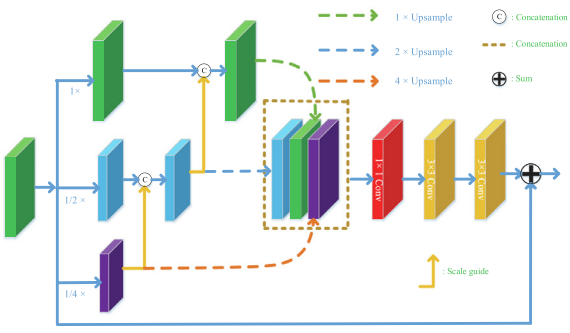


Fig. 3. Small scale guides the large (S2L Block).

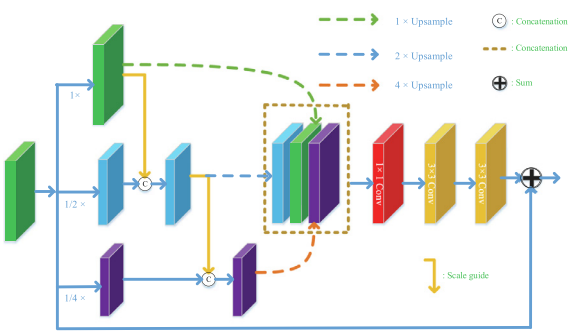


Fig. 4. Large scale guides the small (L2S Block).

Secondly, the guided learning is developed by using 1×1 convolution:

$$z_r = \begin{cases} y_r, & r = 4 \\ \mathcal{H}(\mathcal{H}_{1\times1}(\mathcal{C}[\mathcal{U}(y_{2^r}), y_r])), & r = 1, 2. \end{cases}, \tag{3}$$

where $\mathcal{C}$ denotes concatenation operation at the channel dimension. $\mathcal{H}_{1\times1}$ and $\mathcal{H}$ denotes 1×1 and 3×3 convolution followed a nonlinear activation, respectively. $\mathcal{U}$ denotes the upsampling operation for resizing the small scale feature maps to large scale feature maps. $z_r$is the fusion result of guided learning.

Lastly, all the scales are fused and fed into two convolution layers, and then we add the original input signal x to learn the residual:

$$S2L = \mathcal{H}(\mathcal{H}(\mathcal{H}_{1\times1}(\mathcal{C}[z_4, z_2, z_1]))) + x, \tag{4}$$

where $S2L$ denotes the output of S2L block.

3.2.2. L2S block

The other type of block is that large scale guides the small, i.e., L2S. Although the structure of L2S is similar to S2L, we believe that the two types of scale-guide block can learn different features. So, we also introduce the L2S as another scale-guide manner.

We express the L2S block mathematically:

Firstly, we utilize the *Pooling* operation with different size of kernels and strides to obtain the multi-scale features:

$$y_i = \mathcal{P}_i(x), \quad i = 1, 2, 4. \tag{5}$$

Then the guided learning is developed by using 1×1 convolution:

$$z_r = \begin{cases} y_r, & r = 1 \\ \mathcal{H}\left(\mathcal{H}_{1\times1}\left(\mathcal{C}\left[y_{r/2}, \mathcal{D}(y_r)\right]\right)\right), & r = 2, 4. \end{cases}, \tag{6}$$

where $\mathcal{D}$ denotes the downsampling operation for resizing the large scale feature maps to small scale feature maps.

Lastly, all the scales are fused and fed into two convolution layers, and then we add the original input signal x to learn the residual:

$$L2S = \mathcal{H}(\mathcal{H}(\mathcal{H}_{1\times1}(\mathcal{C}[z_1, \mathcal{U}(z_2), \mathcal{U}(z_4)]))) + x, \tag{7}$$

where $L2S$ denotes the output of the L2S block.

3.2.3. The concatenation between S2L and L2S block

Except for the two types of scale guide block above, we also develop another two types of concatenation between S2L and L2S block, i.e., S2L-L2S and L2S-S2L. One main purpose of developing the two concatenations is that the network can adaptively learn rain streaks information by combining the two types of blocks. By combining the two blocks, the network is able to achieve better rain streaks expression than with either one type of block, i.e., S2L or L2S block. As the two types of scale-guide block can learn different features, the concatenation of them can boost the deraining performance.

The S2L-L2S is the concatenation of the S2L and L2S blocks:

$$y = L2S(S2L(x)). \tag{8}$$

The L2S-S2L is the concatenation of the L2S and S2L blocks:

$$y = S2L(L2S(x)). \tag{9}$$

3.3. Various recurrent units

For the deraining task, we explore three different recurrent unit variants, including ConvRNN [80], ConvGRU [81], and ConvLSTM

[82]. In the following, we provide a detailed illustration of these three recurrent units. Here, we denote x_s^{-1} and x_{s-1}^0 as the feature map of the previous layer in the s-th stage and the hidden layer in (s-1)-th stage. W and U are convolutional kernels.

3.3.1. ConvRNN

ConvRNN [80] is the simplest convolutional unit among these three recurrent units. Its convolutional version can be formulated as:

$$x_s = tanh(W * x_s^{-1} + U * x_{s-1}^0 + b), \tag{10}$$

where $*$ denotes the convolution operation.

3.3.2. ConvGRU

Gated Recurrent Unit (GRU) [81] is one of the most commonly used recurrent units in sequential models.

$$\begin{aligned} z_s &= \sigma(W_z * x_s^{-1} + U_z * x_{s-1}^0 + b_z), \\ r_s &= \sigma(W_r * x_s^{-1} + U_r * x_{s-1}^0 + b_r), \\ n_s &= tanh(W_n * x_s^{-1} + U_n * (r_s \odot x_{s-1}^0) + b_n), \\ x_s^0 &= (1 - z_s) \odot x_{s-1}^0 + z_s \odot n_s. \end{aligned} \tag{11}$$

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

3.3.3. ConvLSTM

The last recurrent unit we investigated is LSTM [82]. Unlike ConvRNN and ConvGRU, ConvLSTM cell has one more input which is the cell state c_s. For the layer of the s-th stage, given the current input x_s^{-1}, the previous cell state c_{s-1} and the previously hidden state x_{j-1}^0, the current cell state c_s and the current hidden state x_s^0 can be computed as:

$$\begin{aligned} f_s &= \sigma(W_f * x_s^{-1} + U_f * x_{s-1}^0 + b_f), i_s \\ &= \sigma(W_i * x_s^{-1} + U_i * x_{s-1}^0 + b_i), o_s \\ &= \sigma(W_o * x_s^{-1} + U_o * x_{s-1}^0 + b_o), t_s \\ &= tanh(W_t * x_s^{-1} + U_t * x_{s-1}^0 + b_t), c_s = f_s \odot c_{s-1} + i_s \odot t_s, x_s^0 \\ &= o_s \odot tanh(c_s). \end{aligned} \tag{12}$$

3.4. Loss function

As our network has multi stages, the loss function also contains multi parts.

At the t stage, the error metric $\mathcal{L}^t$ is:

$$\mathcal{L}^t = ||\widehat{\mathbf{B}}^t - \mathbf{B}||_1, \tag{13}$$

where $\widehat{\mathbf{B}}^t$ and $\mathbf{B}$ denote the estimated clean image at t stage and the background image, respectively. Actually, according to the rainy image decomposition (Eq. (1)), this loss is equivalent to $||\widehat{\mathbf{R}} - \mathbf{R}||_1$, where $\widehat{\mathbf{R}}$ and $\mathbf{R}$ denote the estimated rain streaks and the corresponding ground truth, respectively.

The overall loss function $\mathcal{L}$ is the sum of all stages:

$$\mathcal{L} = \sum_{t=1}^{T} \mathcal{L}^t, \tag{14}$$

where T is the number of all stages.

4. Experiments

In this section, we conduct extensive experiments to demonstrate the effectiveness of the proposed method on both synthetic datasets and real-world rainy images compared with eight state-of-the-art methods, including two prior-based methods, DSC [9] (ICCV15) and LP [10] (CVPR16), six deep-learning based methods, DDN [42] (CVPR17), JORDER [43] (CVPR17), RESCAN [45] (ECCV18), REHEN [83] (ACM MM19), PreNet [53] (CVPR19), and SSIR [84] (CVPR19). The major experimental settings and criterions of quality evaluation are presented in Section 4.1. The implementing details are shown in Section 4.2. Quantitative comparisons on synthetic datasets are reported in Section 4.3, and visual comparisons on real-world datasets are provided in Section 4.4. The inner analysis is discussed in Section 4.5.

4.1. Experiment settings

Synthetic datasets. We evaluate the performance of the proposed method on three widely used synthetic datasets: Rain100L [43], Rain100H [43] and Rain1200 [47]. Rain100L and Rain100H contain 1800 image pairs for training and additional 200 image pairs for testing. Rain1200 has 12000 images for training and 1200 images for testing. These three datasets include various rain streaks with different sizes, shapes, and directions.

In Section 4.5, we select Rain100H as our analysis dataset.
Real-world testing images. Parts of the real-world rainy images are provided by the previous works [43,44,76]. The other challenging ones are downloaded from the Internet. For these real-world rainy images, the rain streaks are different from orientation to density. We use them to illustrate the effectiveness of our method.
Evaluation metrics. To evaluate the quality of the restored results in terms of ground-truth images for a specific algorithm, we use peak signal to noise ratio (PSNR) [85] and structure similarity index (SSIM) [86] as the evaluation metrics on the synthetic datasets. However, since there is no ground-truth (GT) for the real-world images, we only evaluate the performance on the real-world images visually.

4.2. Implementing details

Our networks are trained on three Nvidia GTX 1080Ti GUPs based on Pytorch. The number of channels is chosen as 20, the nonlinear activation we used is LeakyReLU with $\alpha = 0.2$ for all convolution layers except for the last layer. For the last layer, the channel is 3 without any activation function. The patch size is 128×128, and the batch size is 32. We use ADAM [87] to optimize our networks parameters and the parameters are shared among different stages. The initial learning rate is 0.0005, and the rate will be divided by 10 at $60K$ and $80K$ iterations, and the model's training terminates after $100K$ iterations. T is set to be 4 and the reason is given in Section 4.5.4.

4.3. Results on synthetic datasets

There are two prior based methods, DSC [9] and LP [10], and six deep-learning based methods, DDN [42], JORDER [43], RESCAN [45], REHEN [83], PreNet [53], and SSIR [84].

As shown in Table 1, our method significantly outperforms the two prior methods in terms of both PSNR and SSIM. Compared with deep-learning-based methods, our method also obtains the highest scores of the evaluative criteria on three widely used datasets. We also note that the semi-supervised method, i.e., SSIR [84], is less effective on the synthetic datasets. To sum up, our method has a significant improvement compared with previous state-of-the-art methods.

We also provide several visual examples to compare the proposed algorithm with these state-of-the-art methods. Fig. 5 shows three synthetic examples compared with two prior-based methods. DSC [9] and LP [10] fail to generate desirable results, while our method is able to effectively remove the rain streaks and gen-

Table 1
Quantitative experiments evaluated on three synthetic datasets. The best results are highlighted in boldface.

	DSC [9] ICCV'15		LP [10] CVPR'16		DDN [42] CVPR'17		JORDER [43] CVPR'17		RESCAN [45] ECCV'18		REHEN [83] ACM MM'19		PreNet [53] CVPR'19		SSIR [84] CVPR'19		Ours	
Dataset	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Rain100H	15.66	0.42	14.26	0.54	22.26	0.69	23.45	0.74	25.92	0.84	27.52	0.86	27.89	0.89	22.47	0.71	**28.98**	**0.90**
Rain100L	24.16	0.87	29.11	0.88	34.85	0.95	36.11	0.97	36.12	0.97	37.91	0.98	36.69	0.98	32.37	0.92	**38.78**	**0.99**
Rain1200	21.44	0.79	22.46	0.80	30.95	0.86	29.75	0.87	32.35	0.89	32.51	0.91	32.38	0.92	29.32	0.89	**32.93**	**0.93**

erates the cleanest deraining images. Moreover, we also compare our method with six deep-learning-based methods in Fig. 6. It can be seen that the deraining results of other deep-learning-based methods always hand down some rain streaks or artifacts and also blur the background information. In comparison, our method can remove the rain streaks and generate the cleanest results.

4.4. Results on real-world datasets

To further verify the effectiveness of the proposed method, we evaluate the performance on some real-world rainy images. The results of various deraining methods are shown in Fig. 7–9. Fig. 7 shows several real-world examples compared with two prior-based methods [9,10]. For the first two examples, we can observe that our method removes the majority of the rain streaks, while other prior-based methods remain a mass of rain streaks. For the last two examples in Fig. 7, our method restores the cleanest and clearest images, while the results of prior-based methods are unacceptable. It also illustrates that the prior-based methods do not work in most conditions, while our method, a deep-learning-based method, achieves better deraining performance in most circumstances.

Furthermore, we present some examples compared with deep-learning-based methods. In Fig. 8, our method is able to preserve details and generate better background information and effectively remove rain-streaks, while other methods tend to remove some important texture details or remain some rain streaks or blur the background. In Fig. 9, our method obtains the cleanest and clearest deraining images. Moreover, we also provide several real-world rainy examples of [76] in Fig. 10. We can observe that the proposed method obtains better deraining performance.

Overall, the results evaluated on the real-world images captured from different rain conditions demonstrate the effectiveness of our method.

4.5. Inner analysis

In this section, we provide more analysis of our proposed deep model. We first conduct an ablation study on the scale and the guide manner in Section 4.5.1. Then we analyze the selection of the scale-guide style in Section 4.5.2 and the selection of the recurrent unit in Section 4.5.3. Finally, the number of the stages of recurrent networks is discussed in Section 4.5.4.

4.5.1. Ablation study on the multi-scale manner and the scale-guide learning

It is meaningful to explore the effectiveness of each component in the proposed network, including the multi-scale manner and the scale-guide learning. To this end, we conduct a detailed ablation study.

First, we explore the meaning of scale-guide learning. We visualize the feature maps of different layers and scales of S2L block in Fig. 11. We can see that there are different rain streak structures at different scales before fusing them.

Moreover, an important point to be noted is that these layers can not capture better rain streak information, where they do not pay more attention to rain streak information that the network needs to learn. However, the fusion output with different scales can express better rain streaks structures. This is because that the fusion between scale guide manners can learn better rain

Fig. 5. Results on the synthetic datasets compared with prior-based deraining methods. The results of DSC [9] and LP [10] are unsatisfying. The proposed method generates much clearer images.

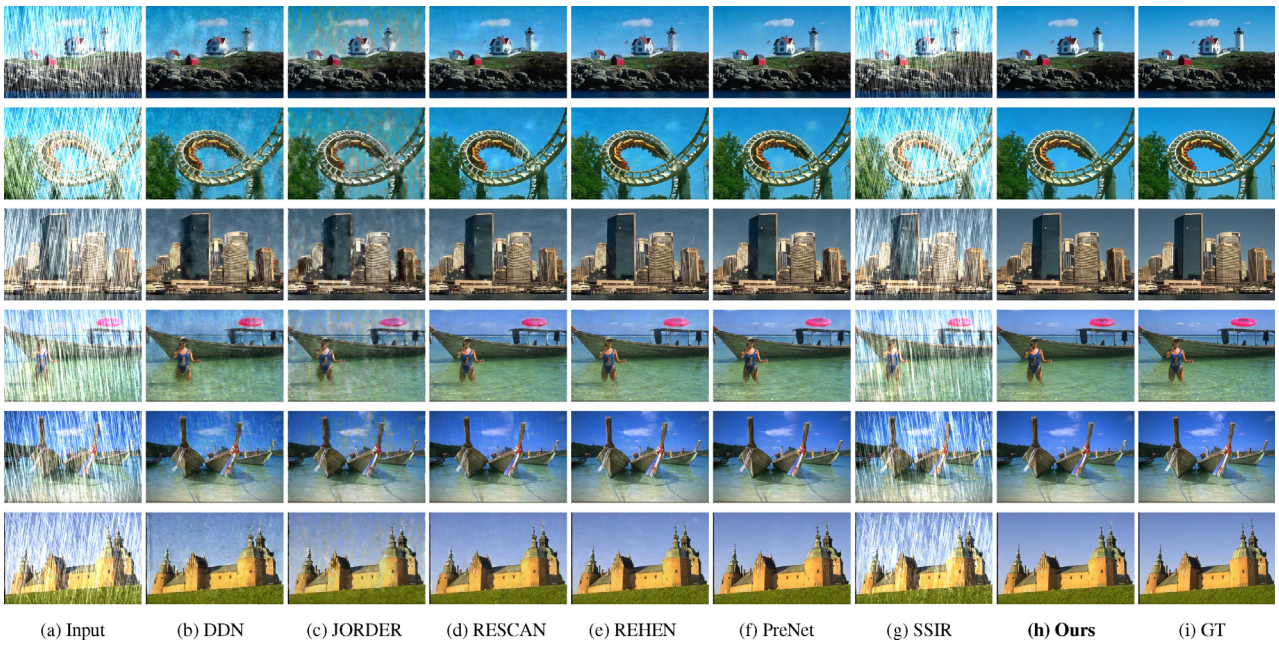


Fig. 6. Results on the synthetic datasets compared with deep-learning-based deraining methods. The proposed method generates much cleaner images, while the other methods either retain some rain streaks or blur the background.

Fig. 7. Several examples on the real-world datasets compared with prior-based deraining methods. The results of DSC [9] and LP [10] still contain some rain streaks. In comparison, our method generates much .clearer results.

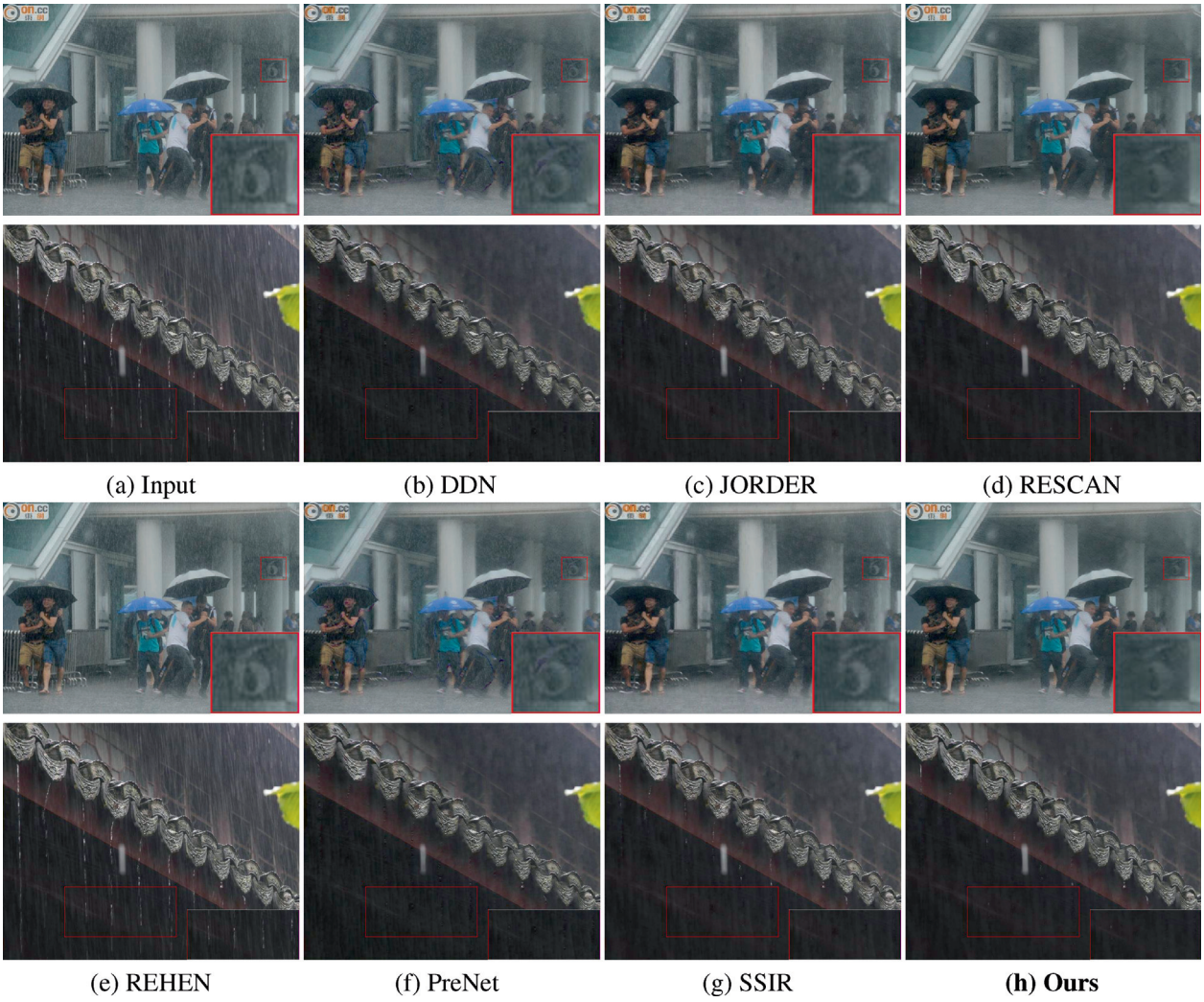


Fig. 8. Two examples on the real-world datasets compared with deep-learning-based deraining methods.

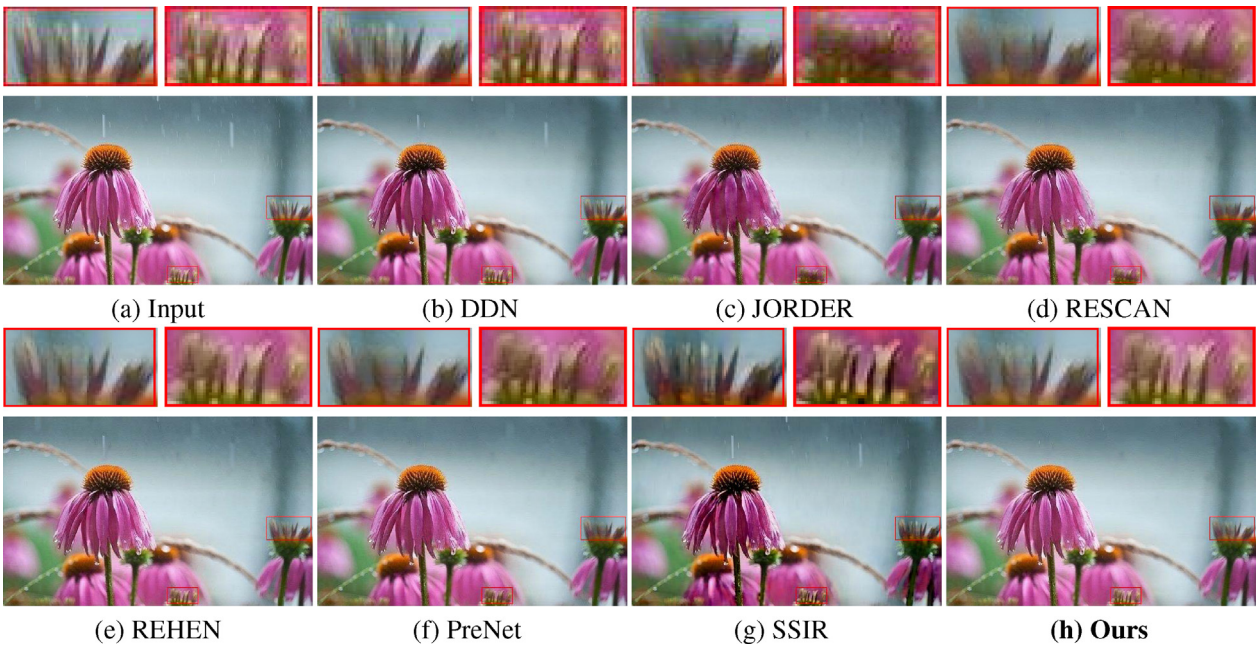


Fig. 9. One example on the real-world datasets compared with deep-learning-based deraining methods.

Fig. 10. Examples on the real-world datasets [76] compared with deep-learning-based deraining methods pro.posed in 2019.

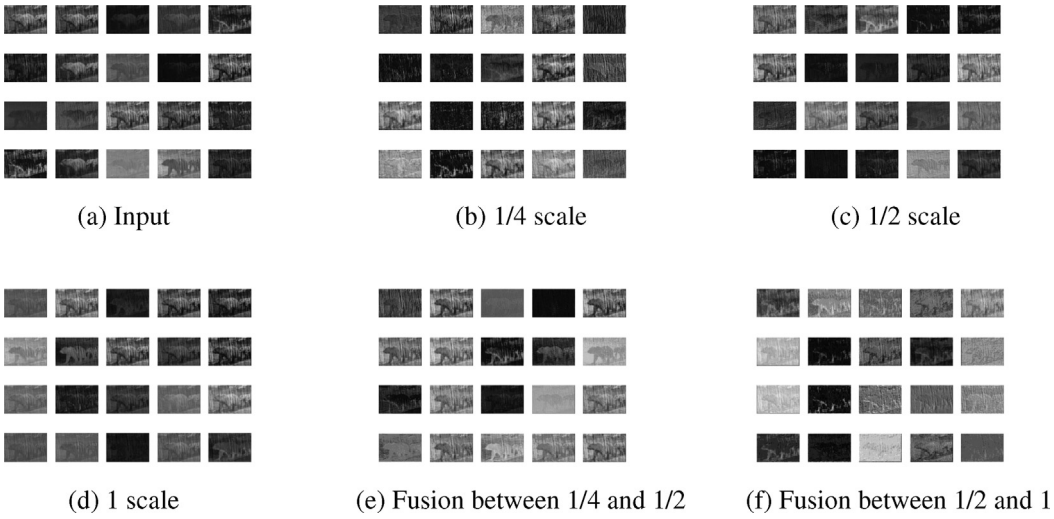


Fig. 11. The visual feature maps of the S2L block.

Table 2
Ablation study on the different components of our method.

Metric	*ResBlock*	*MSBlock*	*SGB*
PSNR	23.97	28.44	28.59
SSIM	0.7630	0.8875	0.8892

streak information, which is meaningful for the network to achieve better rain streaks removal.

Second, quantitative evaluations about the multi-scale manners are reported in Table 2.

- *ResBlock*: The proposed method without the multi-scale manner, i.e., every unit is a residual block.

Fig. 12. One visual example on the ablation study of the multi-scale manner and the scale-guide learning.

Table 3
Effect of the scale-guide style. Best results are marked in bold.

Metric	S2L	L2S	L2S-S2L	S2L-L2S
PSNR	27.90	27.87	28.49	**28.59**
SSIM	0.8776	0.8762	0.8874	**0.8892**

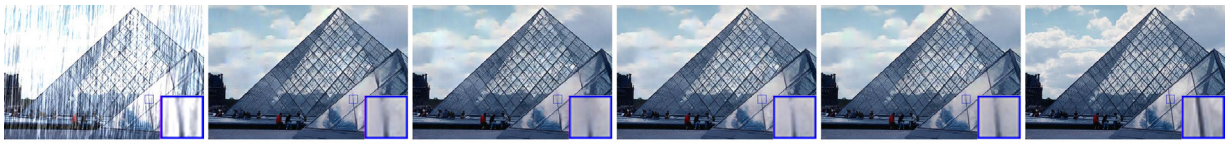


Fig. 13. One visual example on the guide styles. We can observe that S2L-L2S obtains the best deraining result and restore better texture information.

Table 4
Quantitative experiments evaluated on different recurrent units.

Metric	ConvGRU	ConvRNN	**ConvLSTM**
PSNR	28.88	28.82	**28.98**
SSIM	0.8946	0.8942	**0.8968**

- *MSBlock*: The proposed method with multi-scale but without scale-guide learning.
- *SGB*: The proposed method with multi-scale manner and scale-guide learning.

It can be seen that the multi-scale method greatly improves the deraining results. It also illustrates that the multi-scale method is indeed effective for the deraining task. We also observe that our proposed scale-guide manner boosts the deraining performance compared with the method without the guide. Overall, these comparisons firmly indicate the proposed multi-scale manner and scale-guide learning are beneficial for performance improvement.

Moreover, we also present a visual example in Fig. 12. It is obvious that the result shown in Fig. 12 (b) is unacceptable for the method without the multi-scale manner. The deraining result of the method with multi-scale learning and without scale-guide learning shown in Fig. 12 (c) is visually a little better, but it is still unpleasing and some textures are damaged. Details can be better observed via masked boxes. In contrast, our proposed method shown in Fig. 12 (d) is capable of removing rain streaks while preserving the background details.

4.5.2. The selection of the scale-guide style

Different scale-guide blocks are described in Section 3.2. It is meaningful to discuss their representative ability of rain streaks. Without loss of generality, we discuss and analyze when the number of stage is one. We conduct several experiments on these scale-guide blocks and the comparative quantitative results are shown in Table 3. It can be seen that our designed S2L is more effective than L2S, i.e., the style of small scale guides the large performs better than the style of large scale guides the small. Moreover, as can be seen in Table 3, S2L-L2S is better than L2S-S2L in term of PSNR and SSIM and has 0.7 dB PSNR and 1.2% SSIM improvement compared with S2L. So S2L-L2S has stronger representative ability of the rain streaks and we select S2L-L2S as our scale-guide block (SGB).

We also present one example to compare these scale-guide blocks visually in Fig. 13. Our default scale-guide block S2L-L2S gets the highest PSNR and SSIM, and preserves better details than other scale-guide styles in the masked box.

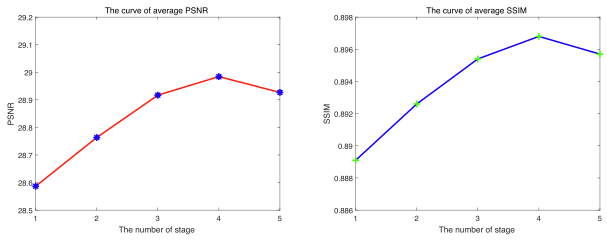
4.5.3. The selection of the recurrent unit

There are three recurrent units that can be selected: ConvRNN [80], ConvGRU [81] and ConvLSTM [82]. It needs to explore which is the best recurrent unit for our method. We conduct three experiments on the recurrent units, as shown in Table 4. We can see that the ConvLSTM perform better than the other recurrent units: ConvGRU and ConvRNN. So, we select ConvLSTM as our default recurrent unit.

Fig. 14. Comparison of recurrent units..

Table 5
Quantitative experiments for analyzing the number of stage. T denotes the number of stage.

Metric	T = 1	T = 2	T = 3	**T = 4**	T = 5
PSNR	28.59	28.76	28.92	**28.98**	28.93
SSIM	0.8891	0.8927	0.8854	**0.8968**	0.8957



(a) PSNR (b) SSIM

Fig. 15. Effect of the number of stages on Rain100H.

(a) (b) (c)

Fig. 17. Application to semantic segmentation. (a) A real-world rainy image. (b) The segmentation result of the rainy image. (c) The segmentation result of the derained image generated by our method.

PSNR/SSIM 26.94/0.88 27.38/0.89 27.11/0.89 **28.04/0.90** 27.60/0.89 Inf/1
(a) Input (b) T = 1 (c) T = 2 (d) T = 3 **(e) T = 4** (f) T = 5 (g) GT

Fig. 16. One visual example w.r.t. different number of stages.

We also present a visual example to compare these three recurrent styles in Fig. 14. It can be clearly observed that the ConvLSTM significantly outperforms the other recurrent units in terms of the PSNR and SSIM.

4.5.4. The selection of the number of stages

In our network, the number of stages of the recurrent networks is critical for the deraining results. We carry out some experiments to choose a suitable number of stages. Table 5 reports the results on the number of stages. We can see that the multi-stage method improves the deraining performance compared with the single-stage and the results get better along with the number of stages increasing when the number of stages is smaller than 4. And it obtains the highest PSNR and SSIM when the number of stages is 4. The result becomes worse when the number of stages changes from 4 to 5. So we select 4 as the number of the recurrent stage. The curves of PSNR and SSIM on different number of stages are presented in Fig. 15.

We also provide a visual example to compare the performance of the stage in Fig. 16. We can see that it obtains better texture information when the stage is 4.

4.6. Application to high-level task

In Fig. 17, we provide an application of deraining on downstreaming task, semantic segmentation. We can observe that the proposed deraining method is able to improve the segmentation accuracy, whereas the segmentation on the rainy image without deraining has more pixels wrongly classified.

5. Conclusion

In this paper, we have proposed a deep-learning-based method to tackle single image deraining. The main contribution of this paper is that we propose a more effective multi-scale manner, i.e., scale-guide learning. We find an interesting fact that feature maps at different scales have different rain streak structures. Based on the finding, we consider the correlation between different scales so that the network can learn better rain streaks expression. We present two types of scale-guide blocks, i.e., S2L and L2S, and develop their combinations, i.e., S2L-L2S and L2S-S2L. Experiments demonstrate the guide manner boosts the deraining performance and our designed S2L-L2S is able to achieve superior deraining results. Moreover, the multi-stage networks improve the deraining results compared with the single-stage ones and LSTM is a better recurrent unit than others. Furthermore, we conduct an application on semantic segmentation, which shows the deraining result can improve the segmentation performance. Quantitative and qualitative experimental results demonstrate the superiority of the proposed method compared with several state-of-the-art deraining methods.

CRediT authorship contribution statement

Cong Wang: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Honghe Zhu:** Visualization, Writing – review & editing. **Wanshu Fan:** Writing – review & editing. **Xiao-Ming Wu:** Writing – review & editing. **Junyang Chen:** Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, athttps://doi.org/10.1016/j.neucom.2021.10.029.

References

[1] K. Garg, S.K. Nayar, Detection and removal of rain from videos, CVPR (2004) 528–535, https://doi.org/10.1109/CVPR.2004.79.

[2] N. Brewer, N. Liu, Using the shape characteristics of rain to identify and remove rain from video, in: Structural, Syntactic, and Statistical Pattern Recognition, 2008, pp. 451–458. doi:10.1007/978-3-540-89689-0_49..

[3] V. Santhaseelan, V.K. Asari, Utilizing local phase information to remove rain from video, Int. J. Comput. Vision 112 (1) (2015) 71–89, https://doi.org/10.1007/s11263-014-0759-8.

[4] A.K. Tripathi, S. Mukhopadhyay, Removal of rain from videos: a review, SIViP 8 (8) (2014) 1421–1430, https://doi.org/10.1007/s11760-012-0373-6.

[5] M. Li, Q. Xie, Q. Zhao, W. Wei, S. Gu, J. Tao, D. Meng, Video rain streak removal by multiscale convolutional sparse coding, CVPR (2018) 6644–6653, https://doi.org/10.1109/CVPR.2018.00695, URL: http://openaccess.thecvf.com/content_cvpr_2018/html/Li_Video_Rain_Streak_CVPR_2018_paper.html.

[6] J. Chen, C. Tan, J. Hou, L. Chau, H. Li, Robust video content alignment and compensation for rain removal in a CNN framework, CVPR (2018) 6286–6295, https://doi.org/10.1109/CVPR.2018.00658, URL: http://openaccess.thecvf.com/content_cvpr_2018/html/Chen_Robust_Video_content_cvpr_2018_paper.html.

[7] J. Liu, W. Yang, S. Yang, Z. Guo, Erase or fill? deep joint recurrent rain removal and reconstruction in videos, CVPR (2018) 3233–3242, https://doi.org/10.1109/CVPR.2018.00341, URL: http://openaccess.thecvf.com/content_cvpr_2018/html/Liu_Erase_or_Fill_CVPR_2018_paper.html.

[8] W. Yang, J. Liu, J. Feng, Frame-consistent recurrent video deraining with dual-level flow, CVPR (2019), https://doi.org/10.1109/CVPR.2019.00176, URL: http://openaccess.thecvf.com/content_CVPR_2019/html/Yang_Frame-Consistent_Recurrent_Video_Deraining_With_Dual-Level_Flow_CVPR_2019_paper.html.

[9] Y. Luo, Y. Xu, H. Ji, Removing rain from a single image via discriminative sparse coding, in: ICCV, 2015, pp. 3397–3405. 10.1109/ICCV.2015.388. 10.1109/ICCV.2015.388..

[10] Y. Li, R.T. Tan, X. Guo, J. Lu, M.S. Brown, Rain streak removal using layer priors, CVPR (2016) 2736–2744, https://doi.org/10.1109/CVPR.2016.299.

[11] L. Kang, C. Lin, Y. Fu, Automatic single-image-based rain streaks removal via image decomposition, IEEE Trans. Image Process. 21 (4) (2012) 1742–1755, https://doi.org/10.1109/TIP.2011.2179057.

[12] Y. Chen, C. Hsu, A generalized low-rank appearance model for spatio-temporally correlated rain streaks, in: ICCV, 2013, pp. 1968–1975. 10.1109/ICCV.2013.247. 10.1109/ICCV.2013.247..

[13] L. Zhu, C. Fu, D. Lischinski, P. Heng, Joint bi-layer optimization for single-image rain streak removal, in: ICCV, 2017, pp. 2545–2553. 10.1109/ICCV.2017.276. 10.1109/ICCV.2017.276..

[14] X. Wang, A. Shrivastava, A. Gupta, A-fast-rcnn: Hard positive generation via adversary for object detection, in: CVPR, 2017, pp. 3039–3048. 10.1109/CVPR.2017.324. 10.1109/CVPR.2017.324.

[15] Y. Wu, S. Feng, X. Huang, Z. Wu, L4net: An anchor-free generic object detector with attention mechanism for autonomous driving, IET Comput. Vision 15 (1) (2021) 36–46, https://doi.org/10.1049/cvi2.12015.

[16] Y. Zhang, L. Wang, J. Qi, D. Wang, M. Feng, H. Lu, Structured siamese network for real-time visual tracking, ECCV (2018) 355–370, https://doi.org/10.1007/978-3-030-01240-3_22.

[17] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, J. Sun, Cascaded pyramid network for multi-person pose estimation, CVPR (2018) 7103–7112, https://doi.org/10.1109/CVPR.2018.00742, URL: http://openaccess.thecvf.com/content_cvpr_2018/html/Chen_Cascaded_Pyramid_Network_CVPR_2018_paper.html.

[18] Y. Wu, H. Liu, S. Feng, Y. Jin, G. Lyu, Z. Wu, GM-MLIC: graph matching based multi-label image classification, IJCAI (2021).

[19] W. Wang, Z. Wang, H. Li, J. Zhou, Z. Ding, Adaptive local neighbors for transfer discriminative feature learning, ECAI (2020) 1595–1602, https://doi.org/10.3233/FAIA200269.

[20] W. Wang, S. Chen, Y. Xiang, J. Sun, H. Li, Z. Wang, F. Sun, Z. Ding, B. Li, Sparsely-labeled source assisted domain adaptation, Pattern Recogn. 112 (2021), https://doi.org/10.1016/j.patcog.2020.107803 107803.

[21] A. Ranjan, M.J. Black, Optical flow estimation using a spatial pyramid network, CVPR (2017) 2720–2729, https://doi.org/10.1109/CVPR.2017.291.

[22] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, CVPR (2015) 3431–3440, https://doi.org/10.1109/CVPR.2015.7298965.

[23] Y. Ju, L. Qi, J. He, X. Dong, F. Gao, J. Dong, Mps-net: Learning to recover surface normal for multispectral photometric stereo, Neurocomputing 375 (2020) 62–70.

[24] Y. Ju, J. Dong, S. Chen, Recovering surface normal and arbitrary images: A dual regression network for photometric stereo, IEEE Trans. Image Process. 30 (2021) 3676–3690.

[25] Y. Ju, K.-M. Lam, Y. Chen, L. Qi, J. Dong, Pay attention to devils: A photometric stereo network for better details, in: IJCAI, 2020, pp. 694–700..

[26] Y. Ju, X. Dong, Y. Wang, L. Qi, J. Dong, A dual-cue network for multispectral photometric stereo, Pattern Recogn. 100 (2020) 107162.

[27] Y. Ju, M. Jian, J. Dong, K. Lam, Learning photometric stereo via manifold-based mapping, in: VCIP, IEEE, 2020, pp. 411–414. 10.1109/VCIP49819.2020.9301860. 10.1109/VCIP49819.2020.9301860..

[28] Y. Ju, M. Jian, S. Guo, Y. Wang, Z. Huiyu, J. Dong, Incorporating lambertian priors into surface normals measurement, IEEE Transactions on Instrumentation and Measurement 10.1109/TIM.2021.3096282..

[29] G. Fu, Q. Zhang, L. Zhu, P. Li, C. Xiao, A multi-task network for joint specular highlight detection and removal, in: CVPR, IEEE, 2021, pp. 7752–7761..

[30] G. Fu, Q. Zhang, Q. Lin, L. Zhu, C. Xiao, Learning to detect specular highlights from real-world images, in: ACM MM, ACM, 2020, pp. 1873–1881..

[31] G. Fu, Q. Zhang, C. Song, Q. Lin, C. Xiao, Specular highlight removal for real-world images, Comput. Graphics Forum 38 (7) (2019) 253–263.

[32] J. Mustaniemi, J. Kannala, S. Särkkä, J. Matas, J. Heikkilä, Inertial-aided motion deblurring with deep networks, in: CoRR, Vol. abs/1810.00986, 2018. arXiv:1810.00986. url:http://arxiv.org/abs/1810.00986.

[33] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, M. Yang, Single image dehazing via multi-scale convolutional neural networks, ECCV (2016) 154–169, https://doi.org/10.1007/978-3-319-46475-6_10.

[34] B. Cai, X. Xu, K. Jia, C. Qing, D. Tao, Dehazenet: An end-to-end system for single image haze removal, IEEE Trans. Image Process. 25 (11) (2016) 5187–5198, https://doi.org/10.1109/TIP.2016.2598681.

[35] B. Li, X. Peng, Z. Wang, J. Xu, D. Feng, Aod-net: All-in-one dehazing network, in: ICCV, 2017, pp. 4780–4788. doi:10.1109/ICCV.2017.511..

[36] H. Zhang, V.M. Patel, Densely connected pyramid dehazing network, CVPR (2018) 3194–3203, https://doi.org/10.1109/CVPR.2018.00337, URL: http://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_Densely_Connected_Pyramid_CVPR_2018_paper.html.

[37] C. Wang, W. Fan, Y. Wu, Z. Su, Weakly supervised single image dehazing, J. Vis. Commun. Image Represent. 72 (2020), https://doi.org/10.1016/j.jvcir.2020.102897 102897.

[38] C. Dong, C.C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2) (2016) 295–307, https://doi.org/10.1109/TPAMI.2015.2439281.

[39] Z. Cui, H. Chang, S. Shan, B. Zhong, X. Chen, Deep network cascade for image super-resolution, ECCV (2014) 49–64, https://doi.org/10.1007/978-3-319-10602-1_4.

[40] K. Jiang, Z. Wang, P. Yi, J. Jiang, Hierarchical dense recursive network for image super-resolution, Pattern Recogn. 107 (2020) 107475.

[41] X. Fu, J. Huang, X. Ding, Y. Liao, J. Paisley, Clearing the skies: a deep network architecture for single-image rain removal, IEEE Trans. Image Process. 26 (6) (2017) 2944–2956, https://doi.org/10.1109/TIP.2017.2691802.

[42] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, J. Paisley, Removing rain from single images via a deep detail network, CVPR (2017) 1715–1723, https://doi.org/10.1109/CVPR.2017.186.

[43] W. Yang, R.T. Tan, J. Feng, J. Liu, Z. Guo, S. Yan, Deep joint rain detection and removal from a single image, in: CVPR, 2017, pp. 1685–1694. 10.1109/CVPR.2017.183. 10.1109/CVPR.2017.183.

[44] H. Zhang, V. Sindagi, V.M. Patel, Image de-raining using a conditional generative adversarial network, in: CoRR, Vol. abs/1701.05957, 2017. arXiv:1701.05957. url:http://arxiv.org/abs/1701.05957..

[45] X. Li, J. Wu, Z. Lin, H. Liu, H. Zha, Recurrent squeeze-and-excitation context aggregation net for single image deraining, ECCV (2018) 262–277, https://doi.org/10.1007/978-3-030-01234-2_16.

[46] G. Li, X. He, W. Zhang, H. Chang, L. Dong, L. Lin, Non-locally enhanced encoder-decoder network for single image de-raining, ACM MM (2018) 1056–1064, https://doi.org/10.1145/3240508.3240636.

[47] H. Zhang, V.M. Patel, Density-aware single image de-raining using a multi-stream dense network, CVPR (2018) 695–704, https://doi.org/10.1109/CVPR.2018.00079, URL: http://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_Density-Aware_Single_Image_CVPR_2018_paper.html.

[48] X. Fu, Q. Qi, Y. Huang, X. Ding, F. Wu, J.W. Paisley, A deep tree-structured fusion model for single image deraining, CoRR abs/1811.08632. arXiv:1811.08632. url:http://arxiv.org/abs/1811.08632.

[49] J. Pan, Y. Liu, J. Dong, J. Zhang, J.S.J. Ren, J. Tang, Y. Tai, M. Yang, Physics-based generative adversarial models for image restoration and beyond, IEEE Transactions on Pattern Analysis and Machine Intelligence 10.1109/TPAMI.2020.2969348. URL: http://arxiv.org/abs/1808.00605..

[50] J. Pan, S. Liu, D. Sun, J. Zhang, Y. Liu, J.S.J. Ren, Z. Li, J. Tang, H. Lu, Y. Tai, M. Yang, Learning dual convolutional neural networks for low-level vision, CVPR (2018) 3070–3079, https://doi.org/10.1109/CVPR.2018.00324, URL: http://openaccess.thecvf.com/content_cvpr_2018/html/Pan_Learning_Dual_Convolutional_CVPR_2018_paper.html.

[51] Z. Fan, H. Wu, X. Fu, Y. Huang, X. Ding, Residual-guide network for single image deraining, ACM MM (2018) 1751–1759, https://doi.org/10.1145/3240508.3240694.

[52] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, R.W. Lau, Spatial attentive single-image deraining with a high quality real rain dataset, CVPR (2019) 12270–12279, https://doi.org/10.1109/CVPR.2019.01255, URL: http://openaccess.thecvf.com/content_CVPR_2019/html/Wang_Spatial_Attentive_Single-Image_Deraining_With_a_High_Quality_Real_Rain_CVPR_2019_paper.html.

[53] D. Ren, W. Zuo, Q. Hu, P. Zhu, D. Meng, Progressive image deraining networks: A better and simpler baseline, CVPR, IEEE (2019) 3937–3946, https://doi.org/10.1109/CVPR.2019.00406, URL: http://openaccess.thecvf.com/content_CVPR_2019/html/Ren_Progressive_Image_Deraining_Networks_A_Better_and_Simpler_Baseline_CVPR_2019_paper.html.

[54] R. Li, L.-F. Cheong, R.T. Tan, Heavy rain image restoration: Integrating physics model and conditional adversarial learning, CVPR (2019) 1633–1642, https://doi.org/10.1109/CVPR.2019.00173, URL: http://openaccess.

thecvf.com/content_CVPR_2019/html/Li_Heavy_Rain_Image_Restoration_Integrating_Physics_Model_and_Conditional_Adversarial_CVPR_2019_paper.html.
[55] C. Wang, M. Zhang, Z. Su, Y. Wu, G. Yao, H. Wang, Learning a multi-level guided residual network for single image deraining, Signal Process. Image Commun. 78 (2019) 206–215, https://doi.org/10.1016/j.image.2019.07.003.
[56] W. Yang, R.T. Tan, J. Feng, J. Liu, S. Yan, Z. Guo, Joint rain detection and removal from a single image with contextualized deep networks, IEEE Trans. Pattern Anal. Mach. Intell. 42 (6) (2020) 1377–1393, https://doi.org/10.1109/TPAMI.2019.2895793.
[57] X. Fu, B. Liang, Y. Huang, X. Ding, J. Paisley, Lightweight pyramid networks for image deraining, IEEE Trans. Neural Networks Learn. Syst. 31 (6) (2020) 1794–1807, https://doi.org/10.1109/TNNLS.2019.2926481.
[58] K. Jiang, Z. Wang, P. Yi, C. Chen, Z. Han, T. Lu, B. Huang, J. Jiang, Decomposition makes better rain removal: An improved attention-guided deraining network, IEEE Trans. Circuits Syst. Video Technol. (2020), https://doi.org/10.1109/TCSVT.2020.3044887, 1–1.
[59] C. Wang, Y. Wu, Y. Cai, G. Yao, Z. Su, H. Wang, Single image deraining via deep pyramid network with spatial contextual information aggregation, Appl. Intell. 50 (5) (2020) 1437–1447, https://doi.org/10.1007/s10489-019-01567-5.
[60] C. Wang, M. Zhang, Z. Su, G. Yao, Y. Wang, X. Sun, X. Luo, From coarse to fine: A stage-wise deraining net, IEEE Access 7 (2019) 84420–84428, https://doi.org/10.1109/ACCESS.2019.2922549.
[61] C. Wang, H. Wang, Z. Su, Y. Yang, Embedding non-local mean in squeeze-and-excitation network for single image deraining, ICMEW, IEEE (2019) 264–269, https://doi.org/10.1109/ICMEW.2019.00-76.
[62] C. Wang, M. Zhang, Z. Su, G. Yao, Densely connected multi-scale de-raining net, Multim. Tools Appl. 79 (27–28) (2020) 19595–19614, https://doi.org/10.1007/s11042-020-08855-0.
[63] C. Wang, W. Fan, H. Zhu, Z. Su, Single image deraining via nonlocal squeeze-and-excitation enhancing network, Appl. Intell. 50 (9) (2020) 2932–2944, https://doi.org/10.1007/s10489-020-01693-5.
[64] C. Wang, X. Xing, G. Yao, Z. Su, Single image deraining via deep shared pyramid network, Vis. Comput. 37 (7) (2021) 1851–1865, https://doi.org/10.1007/s00371-020-01944-z.
[65] H. Zhu, C. Wang, Y. Zhang, Z. Su, G. Zhao, Physical model guided deep image deraining, ICME, IEEE (2020) 1–6, https://doi.org/10.1109/ICME46284.2020.9102878.
[66] C. Wang, X. Xing, Y. Wu, Z. Su, J. Chen, DCSFN: deep cross-scale fusion network for single image rain removal, in: MM, ACM, 2020, pp. 1643–1651. 10.1145/3394171.3413820. 10.1145/3394171.3413820..
[67] C. Wang, Y. Wu, Z. Su, J. Chen, Joint self-attention and scale-aggregation for self-calibrated deraining network, in: MM, ACM, 2020, pp. 2517–2525. 10.1145/3394171.3413559. 10.1145/3394171.3413559..
[68] X. Chen, Y. Huang, L. Xu, Multi-scale hourglass hierarchical fusion network for single image deraining, in: CVPRW, IEEE, 2021..
[69] Z. Wang, C. Wang, Z. Su, J. Chen, Dense feature pyramid grids network for single image deraining, ICASSP (2021) 2025–2029, https://doi.org/10.1109/ICASSP39728.2021.9415034.
[70] G. Yao, C. Wang, Y. Wu, Y. Wang, Pyramid fully residual network for single image de-raining, Neurocomputing 456 (2021) 168–178, https://doi.org/10.1016/j.neucom.2021.05.086.
[71] S. Li, W. Ren, F. Wang, I.B. Araujo, E.K. Tokuda, R.H. Junior, R.M.C. Jr., Z. Wang, X. Cao, A comprehensive benchmark analysis of single image deraining: Current challenges and future perspectives, Int. J. Comput. Vis. 129 (4) (2021) 1301–1322. 10.1007/s11263-020-01416-w. 10.1007/s11263-020-01416-w..
[72] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, CVPR (2018) 7132–7141, https://doi.org/10.1109/CVPR.2018.00745, URL: http://openaccess.thecvf.com/content_cvpr_2018/html/Hu_Squeeze-and-Excitation_Networks_CVPR_2018_paper.html.
[73] X. Wang, R.B. Girshick, A. Gupta, K. He, Non-local neural networks, CVPR (2018) 7794–7803, https://doi.org/10.1109/CVPR.2018.00813, URL: http://openaccess.thecvf.com/content_cvpr_2018/html/Wang_Non-Local_Neural_Networks_CVPR_2018_paper.html.
[74] J. Kim, C. Lee, J. Sim, C. Kim, Single-image deraining using an adaptive nonlocal means filter, ICIP (2013) 914–917, https://doi.org/10.1109/ICIP.2013.6738189.
[75] Y. Wang, S. Liu, C. Chen, B. Zeng, A hierarchical approach for rain or snow removing in a single color image, IEEE Trans. Image Process. 26 (8) (2017) 3936–3950, https://doi.org/10.1109/TIP.2017.2708502.
[76] S. Li, I.B. Araujo, W. Ren, Z. Wang, E.K. Tokuda, R.H. Junior, R. Cesar-Junior, J. Zhang, X. Guo, X. Cao, Single image deraining: A comprehensive benchmark analysis, in: CVPR, 2019, pp. 3838–3847. 10.1109/CVPR.2019.00396. url:http://openaccess.thecvf.com/content_CVPR_2019/html/Li_Single_Image_Deraining_A_Comprehensive_Benchmark_Analysis_CVPR_2019_paper.html.
[77] T. Lin, P. Dollár, R.B. Girshick, K. He, B. Hariharan, S.J. Belongie, Feature pyramid networks for object detection, CVPR (2017) 936–944, https://doi.org/10.1109/CVPR.2017.106.
[78] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, CVPR (2017) 6230–6239, https://doi.org/10.1109/CVPR.2017.660.
[79] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, ECCV (2014) 346–361, https://doi.org/10.1007/978-3-319-10578-9_23.
[80] D.P. Mandic, J. Chambers, Recurrent neural networks for prediction: Learning algorithms, architectures and stability, Adaptive Learning Systems for Signal Processing Communications Control..
[81] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, EMNLP (2014) 1724–1734, URL: http://aclweb.org/anthology/D/D14/D14-1179.pdf.
[82] W. Zaremba, I. Sutskever, O. Vinyals, Recurrent neural network regularization, CoRR abs/1409.2329. arXiv:1409.2329. url:http://arxiv.org/abs/1409.2329.
[83] Y. Yang, H. Lu, Single image deraining via recurrent hierarchy enhancement network, in: ACM MM, 2019, pp. 1814–1822. doi:10.1145/3343031.3351149..
[84] W. Wei, D. Meng, Q. Zhao, Z. Xu, Y. Wu, Semi-supervised transfer learning for image rain removal, CVPR (2019) 3877–3886, https://doi.org/10.1109/CVPR.2019.00400, URL: http://openaccess.thecvf.com/content_CVPR_2019/html/Wei_Semi-Supervised_Transfer_Learning_for_Image_Rain_Removal_CVPR_2019_paper.html.
[85] Q. Huynh-Thu, M. Ghanbari, Scope of validity of psnr in image/video quality assessment, Electron. Lett. 44 (13) (2008) 800–801, https://doi.org/10.1049/el:20080522, URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4550695.
[86] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612, https://doi.org/10.1109/TIP.2003.819861.
[87] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: ICLR, 2015. url:http://arxiv.org/abs/1412.6980.

Cong Wang is currently a Ph.D. student at the Department of Computing of The Hong Kong Polytechnic University. He received the Master's Degree in Computational Mathematics from Dalian University of Technology in 2020 and the Bachelor's Degree in Mathematics and Applied Mathematics from Inner Mongolia University in 2017. His research interests are computer vision and deep learning.

Honghe Zhu received B.S. degree from Northeast Normal University, China, in 2019. She is currently a master student in Dalian University of Technology, China. Her research interests include computer vision and deep learning.

Wanshu Fan is working in Dalian University. She received the Ph.D. degree in computational mathematics from the Dalian University of Technology in 2020. Her research interests include computer vision, image processing and deep learning.

Xiao-Ming Wu is currently an assistant professor at the Department of Computing, The Hong Kong Polytechnic University. She works on machine learning and artificial intelligence. Her research has contributed to new theoretical insights of machine learning algorithms and novel methods for artificial intelligence applications in various fields. She publishes in leading venues including NeurIPS, CVPR, ACL, EMNLP, KDD, IJCAI and AAAI. She obtained her PhD degree from the Department of Electrical Engineering, Columbia University in 2016. Prior to that, she obtained her MPhil Degree from the Chinese University of Hong Kong and her Bachelor's and Master's degrees from Peking University.

Junyang Chen received a Ph.D. degree in computer and information science from University of Macau, Macau, China, in 2020. He is currently an Assistant Professor with the College of Computer Science and Software Engineering, Shenzhen University, China. His research interests include graph neural networks, text mining, and recommender systems.