Signal Processing: Image Communication 78 (2019) 206–215

Contents lists available at ScienceDirect

Signal Processing: Image Communication

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

Learning a multi-level guided residual network for single image deraining[☆]

Cong Wang[a], Man Zhang[a], Zhixun Su[a,b,*], Yutong Wu[a], Guangle Yao[c], Hongyan Wang[a]

[a] School of Mathematical Sciences, Dalian University of Technology, No. 2 Linggong Road, Ganjingzi District, Dalian, China
[b] Guangxi Experiment Center of Information Science, Guilin University of Electronic Technology, No. 1 Jinji Road, Qixing District, Guilin, China
[c] School of Cyber Security, Chengdu University of Technology, No. 1 Erxianqiaodongsan Road, Chenghua District, Chengdu, China

ARTICLE INFO

Keywords:
Deraining
Convolutional neural network
Fusion connections
Multi-level
Guided learning

ABSTRACT

Rainy images severely degrade visibility and make many computer vision algorithms invalid. Hence, it is necessary to remove rain streaks from a single image. In this paper, we propose a novel end-to-end deep learning based deraining method. Previous methods neglect the correlation between different layers with different receptive fields that loss a lot of important information. To better solve the problem, we develop a multi-level guided residual block that is the basic unit of our network. In this block, we utilize multi-level dilation convolutions to obtain different receptive fields and the layer with smaller receptive fields to guide the learning of larger receptive fields. Moreover, in order to reduce the model sizes, the parameters are shared among all multi-level guided residual blocks. Experiments illustrate that guided learning improves the deraining performance and the shared parameters strategy is also feasible. Quantitative and qualitative experimental results demonstrate the superiority of the proposed method compared with several state-of-the-art deraining methods.

1. Introduction

Image processing is an important research field and usually acts as preprocessing of numerous applications in artificial intelligence domain. Rainy images often degrade the visibility and make the background scene misty, which will seriously influence the accuracy of many computer vision systems, e.g., object detection, object tracking, video surveillance and so on. So deraining becomes more and more important and it is necessary to propose an effective deraining algorithm.

Usually, rainy images O can be modeled as the linear combination of rain-free image B and rain streaks R:

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

Eq. (1) is an ill-posed problem that there are numerous solutions of B, R for a given O, theoretically. In the past decades, in order to get a better solution, many researchers have proposed some priors about rain streaks or rainy images to restrain the solution space. Among these priors, sparse coding [1], low-rank representation [2] and the Gaussian mixture model [3] are widely used. And some researchers can also directly regard deraining as an image filtering problem and solve by resorting to nonlocal mean smoothing [4]. However, since these models are based on handcrafted low-level features and fixed prior rain streaks assumptions, they can only cope with raindrops of specific shapes, scales and density, and can easily lead to the destruction of image details which are similar to rain streaks. Hence, a more effective deraining method, which can process various rain streaks and preserve the image details, is needed.

Recently, due to its powerful ability of feature representation, convolutional neural network (deep learning) have achieved great success in many computer vision domains, e.g. object detection [5], object tracking [6], semantic segmentation [7], debluring [8], dehazing [9–12], super resolution [13–15] and also de-raining [16–22]. These deep learning based deraining methods have gained a huge improvement and been demonstrated to be more effective than the traditional methods. They usually obtain the rain-free images from a designed end-to-end network by learning the negative residuals or adversarial learning. Although a lot of work has been done and these deep-learning based methods work very well, there are still existing several unsolved shortcomings. Our summary is as follows.

Firstly, spatial contextual information acts as an important role for single image deraining [23] that rain streaks with different sizes need different spatial context to process. Several methods [17] only design forwarded residual networks to learn the rain streaks from high-frequency detail information, which neglect the spatial contextual information that greatly limits the robustness of the algorithms. Secondly, although some methods use the sum among multi-stream

[☆] No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.image.2019.07.003.

[*] Corresponding author at: School of Mathematical Sciences, Dalian University of Technology, No. 2 Linggong Road, Ganjingzi District, Dalian, China.
E-mail address: zxsu@dlut.edu.cn (Z. Su).

https://doi.org/10.1016/j.image.2019.07.003
Received 4 June 2019; Received in revised form 28 June 2019; Accepted 3 July 2019
Available online 11 July 2019

dilation convolutions [18] or multi-kernels with different sizes [22] to obtain the spatial contextual information, the inner correlation between features at different levels is ignored. There are no methods in the literature considering them into deraining.

To resolve these issues, we propose a novel end-to-end network for single image deraining. Specifically, we propose a multi-level guided residual block (MLGRB) that makes up our final network. On the one hand, the MLGRB utilizes several dilation convolutions with exponential factors to acquire spatial contextual information with different levels that process rain streaks with different sizes. On the other hand, we take the inner correlation between different levels into consideration for deraining. The layer with smaller receptive fields guides the learning of an adjacent layer with larger receptive fields by using 1 × 1 convolution to further process the rain streaks information better. By guided learning, the network has more stronger rain streaks representation ability.

Here, our contributions are:

- We develop a multi-level guided residual block and experiments demonstrate that guided learning can enhance further the rain streaks representation ability.
- We demonstrate the flexibility of our algorithm in several aspects by conducting a lot of comparative experiments.
- Quantitative and qualitative experimental evaluations on both synthetic datasets and real-world datasets show that our proposed network outperforms the state-of-the-art methods.

2. Related work

Existing deraining methods can be divided into two categories, including video-based methods and single image-based methods. Compared with single image deraining, video-based methods [24,24–27] are easier, because they can leverage temporal information by analyzing the difference between adjacent frames. In this paper, we focus on single image deraining.

As aforementioned, single image based deraining methods can be divided into two categories, including prior based methods and deep-learning based methods. In this section, we provide a brief review of these methods.

Prior Based Methods: Deraining starts from traditional methods. Kang et al. [28] assume that rain streaks are high frequency structure and separate the rain streaks by utilizing sparse coding from HOG features in the high-frequency layer. Luo et al. [1] propose a discriminative sparse coding framework based on image patches and separate rain streaks from rain-free background images. Chen et al. [2] believe that rain streaks layer is low-rank and utilize a generalized low-rank model to separate rain streaks. Wang et al. [29] design a hierarchical approach for rain or snow removing in a single color image.

Deep-learning Based Methods: Recently, several deep learning based deraining methods achieve promising performance. Fu et al. [16,17] firstly introduce deep-learning methods to single image deraining. They decompose rainy images into low- and high-frequency parts by the guided filter and map high-frequency parts to rain streaks by a residual network, lastly utilize Eq. (1) to obtain the clean image. Yang et al. [18] propose a deep-learning frame, where they jointly detect and remove rain streaks using a recurrent contextual convolutional neural network. Li et al. [21] come up with a non-local enhanced encoder–decoder network that maps rainy images to clean image via learning the residual. Fan et al. [30] develop a residual-guide network for single image deraining. Li et al. [20] propose a recurrent squeeze-and-excitation [31] context aggregation net for single image deraining. Zhang et al. [22] present a density-aware guided multi-stream connected network to jointly estimate rain density and clean images.

3. Proposed method

Fig. 1 illustrates our proposed overall network framework, which consists of several of our developed multi-level guided residual block (MLGRB) shown in Fig. 2. The MLGRB acquires more spatial contextual information by utilizing dilation convolutions with exponential factors. And the layer with smaller receptive field guides the learning of the layer with the larger receptive field in order to learn more the rain streaks information from the former. Moreover, to reduce the model sizes, we utilize the shared parameters strategy that the parameters are shared among all MLGRBs. We will demonstrate that the shared parameters strategy is feasible while the deraining results are still satisfactory. All MLGRBs are connected by fusion connections that can boost the information flow along with features from different levels. We will introduce the proposed network and the developed multi-level guided residual block detailedly in the next sections.

3.1. Overall network framework

Our proposed network is illustrated in Fig. 1. As rain streaks have simpler structure than background images, they are easier to learn. Hence, the network maps rainy images to rain streaks and then obtain final clean images via Eq. (1). The basic unit of the network is the multi-level guided residual block (MLGRB) and fusion connections are utilized to inner-connect the MLGRBs in order to boost the information flow along with features from different levels.

We describe the overall network mathematically:

$$F_0 = Conv(O), \tag{2}$$

where O denotes a rainy image. $Conv$ denotes 3 × 3 convolution. The operation is to convert image space into feature space.

The converted features are flowed into a series of MLGRBs for further extracting rain streaks information:

$$F_i = MLGRB_i(F_{i-1}), i = 1, 2, \dots, N. \tag{3}$$

where $MLGRB_i$ denotes ith MLGRB and F_i denotes the corresponding output. N denotes the number of MLGRBs. Please note that the parameters are shared among all MLGRBs.

All the MLGRBs are fused to boost the information flow along with features from different levels:

$$F_{fusion} = Conv_{1\times1}(Cat[F_N, F_{N-1}, \dots, F_0]), \tag{4}$$

where F_{fusion} denotes the fusion output. Cat and $Conv_{1\times1}$ denote the concatenation at the dimension of channel and 1 × 1 convolution operation, respectively.

Then, the rain streaks $\hat{S}$ can be gained:

$$\hat{S} = Conv(F_{fusion}), \tag{5}$$

Finally, we obtain the final estimated clean image $\hat{B}$ via Eq. (1):

$$\hat{B} = O - \hat{S}, \tag{6}$$

3.2. Multi-level guided residual block

As spatial contextual information is important for single image deraining [23], multi-level dilation convolutions are utilized that can enlarger receptive fields, while keeping the number of parameters unchanged. Moreover, in order to learn the rain streaks information better, we develop guided learning between two adjacent layers that layer with a smaller receptive field guide the learning of the layer with the larger receptive field. In the guided learning, both 1 × 1 convolution and element-wise sum are available. We select 1 × 1 convolution to guide the learning because element-wise sum does not learn the fusion that which parts are effective for different layers. However, the 1 × 1 convolution learns the fusion between different layers to boost the guide learning. More discussion about guided learning can be found in

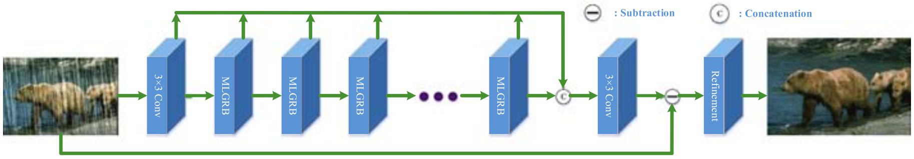


Fig. 1. Overall network framework. MLGRB is shown in Fig. 2 The parameters in all MLGRBs are shared.

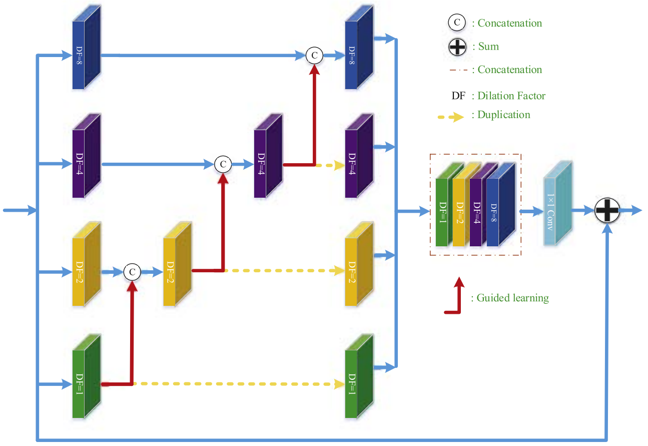


Fig. 2. Multi-level Guided Residual Block (MLGRB). Take $L = 4$ as an example.

Section 4.4.1. The proposed multi-level guided residual block is shown in Fig. 2 and we express it mathematically:

Firstly, multi-level dilation convolutions are to acquire more spatial contextual information:

$$y_r = Conv_r(x), r = 1, 2, \dots, 2^{L-1}. \tag{7}$$

where x denotes the input signal and y_r is the output signal. $Conv_r$ denotes the dilation convolution with dilation factor r. L denotes the number of levels in the multi-level guided residual block.

Then, the guided learning between two adjacent layers is developed where the layer with smaller receptive field guide the learning of the layer with larger receptive field:

$$z_r = \begin{cases} y_r, & r = 1 \\ Conv_r(Conv_{1\times1}(Cat[y_{r/2}, y_r])), & r = 2, \dots, 2^{L-1} \end{cases}, \tag{8}$$

Lastly, all the levels are fused and added the original input signal x in order to learn the residual:

$$MLGRB = Conv_{1\times1}(Cat[z_{2^{L-1}}, z_{2^{L-2}}, \dots, z_1]) + x. \tag{9}$$

where $MLGRB$ denotes the output of the multi-level guided residual block.

3.3. Refinement process

We find that there are existing some artifacts in the final estimated images, see Fig. 8. To solve this problem, we use a simper convolutional network to refine the final results. The refinement stage firstly encodes estimated rain-free images to high-level features gradually with the channels increasing, and then decodes the high-level features to reconstruct the refined rain-free images. Although this encode–decode designment is simple, it indeed boosts to obtain the better deraining results, illustrated in Section 4.4.1. The refinement stage is defined as: $\sigma(Conv(3, C)) - \sigma(Conv(C, 2\times C)) - \sigma(Conv(2\times C, 4\times C))-$ $\sigma(Conv(4\times C, 2\times C)) - Conv(2\times C, 3)$, where $Conv(p, q)$ denotes the 3×3 convolution with p input channels and q output channels. C denotes the number of channels. Here we select LeakyReLU with $\alpha = 0.2$ as σ.

3.4. Loss function

As our network has two stages that one is deraining procedure and another is refinement process, the loss function also contains two parts.

For the deraining procedure, we use L_2 as error metric:

$$\mathcal{L}_{rain} = \frac{1}{HWC} \sum_{h=1}^{H} \sum_{w=1}^{W} \sum_{c=1}^{C} \| \hat{\boldsymbol{R}}_{h,w,c} - \boldsymbol{R}_{h,w,c} \|_2^2, \tag{10}$$

where H, W and C denote the height, width and channel number of a rain streaks, respectively. $\hat{\boldsymbol{R}}$ and $\boldsymbol{R}$ denote estimated clean image and background image, respectively. Actually, $\boldsymbol{R}$ can be obtained via Eq. (1): $\boldsymbol{R} = \boldsymbol{O} - \boldsymbol{B}$.

For the refinement process, we use L_2 as error metric:

$$\mathcal{L}_{refine} = \frac{1}{HWC} \sum_{h=1}^{H} \sum_{w=1}^{W} \sum_{c=1}^{C} \| \hat{\boldsymbol{B}}_{h,w,c} - \boldsymbol{B}_{h,w,c} \|_2^2, \tag{11}$$

So the integral loss function is:

$$\mathcal{L} = \mathcal{L}_{rain} + \mathcal{L}_{refine}. \tag{12}$$

4. Experimental results

In this section, we demonstrate the effectiveness of the proposed method by conducting various experiments on three synthetic datasets and a real-world dataset. All the results are compared with six state-of-the-art methods: DSC [1] (ICCV15), LP [3] (CVPR16), DDN [17] (CVPR17), JORDER [18] (CVPR17), RESCAN [20] (ECCV18), DID [22] (CVPR18).

4.1. Experiment settings

Synthetic Datasets We conduct deraining experiments on three widely used synthetic datasets: Rain100L [18], Rain100H [18] and Rain1200 [22]. These three datasets include various rain streaks with different sizes, shapes and directions. Rain100H and Rain100L have 1800 images for training and 200 images for testing, respectively. Rain1200 has 12000 images for training and 1200 images for testing. It is ensured that all the testing datasets have different background images with training datasets. We select Rain100H as our analysis dataset.

Real-world Datasets: Zhang et al. [19] and Yang et al. [18] also provide some real-world images, we use these images to evaluate out the robustness on real-word images.

Quality Measurements: Peak signal to noise ratio (PSNR) [32] and structure similarity index (SSIM) [33] are widely used in image restoration, which evaluates the quality of restored results with ground-truth. We also use them as our measurement criteria on synthetic datasets. As it is difficult to acquire the ground-truth for real-world images, we only evaluate the performance on the real-world dataset visually.

Training Details: We set $N = 10$, $L = 4$ empirically and the reason will be given in next sections. The number of channels, i.e. C, is 20 and the non-linear activation is LeakyReLU with $\alpha = 0.2$ for all convolution layers. We randomly crop 100×100 patch pairs from training image datasets as inputs with a mini-batch size of 10 to train our network. ADAM [34] is used as the optimization algorithm with an initialized learning rate of 0.001, and divide it by 10 at 240 K and 320 K

Table 1
Quantitative experiments evaluated on three synthetic datasets. The best and the second best results are boldfaced and underlined, respectively.

Dataset	DSC [1]		LP [3]		DDN [17]		JORDER [18]		RESCAN [20]		DID [22]		Ours	
	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	26.12	0.83	**27.52**	**0.86**
Rain100L	24.16	0.87	29.11	0.88	34.85	0.95	36.11	0.97	36.12	0.97	36.14	0.96	**36.97**	**0.98**
Rain1200	21.44	0.79	22.46	0.80	30.95	0.86	29.75	0.87	32.35	0.89	29.65	0.90	**32.36**	**0.91**

iterations, and terminate training after 400 K iterations. We use PyTorch to perform all experiments on a NVIDIA GTX 1080Ti GPU. The network is trained end-to-end. As our entire model is fully convolutional, the testing process only takes 0.014 s when handling a test image with 512 × 512 pixels on a PC with a GTX 1080Ti GPU.

4.2. Results on synthetic datasets

Quantitative comparisons between the proposed method with six state-of-the-art deraining methods on three synthetic datasets are shown in Table 1. There are two prior based methods, including DSC [1] and LP [3], and four deep-learning based methods, including DDN [17], JORDER [18], RESCAN [20] and DID [22].

As shown in Table 1, compared with prior based methods, there has been a huge improvement in our results. For deep-learning based methods, the proposed method also obtains the highest evaluative criteria on three used widely datasets.

Further, we provide several visual examples to compare. Firstly, we compare prior based methods in Fig. 3. It can be observed that the results of prior based methods, DSC [1] and LP [3], are unacceptable, while our results are clearest and cleanest. Secondly, we compare deep learning based methods in Fig. 4. We can see that the results of deep learning based methods always exist some remaining rain streaks or artifacts. However, our results obtain the best performance that is the clearest and cleanest.

4.3. Results on real-world datasets

To further verify the effectiveness of the proposed method, we provide several real-world examples. Firstly, we provide some real-world examples compared with prior based methods [1,3], shown in Fig. 5. It is obvious that our method gains the best performance. While the other results, shown in Fig. 5(b) and (c), have a large amount of residual rain streaks. Then, we provide several examples compared with deep-learning based methods [17,18,20,22]. For the first two examples, our results in Fig. 6(f) obtain clearer texture information in the masked boxes and cleaner from the global perspectives. For the latter two examples, our results in Fig. 6(f) gain the best performance and have fewer artifacts, while the other results remain a lot of rain streaks. So our method obtains the best performance in provided real-world examples. We also present more of our deraining results in real-world dataset in Fig. 7.

Lastly, we conduct a user study on real-world data. There are 84 real-world images from the previously released dataset [18] and [19] and by searching ‘rain’ in Google Images. Based on the output results of DDN [17], JORDER [18], RESCAN [20] and DID [22] on these 84 real images, we invite 10 people to select the one with the best deraining results. Results are shown in Table 2 and we can see that our deraining results obtain the best reputation in the number of voting and selecting images. It can well demonstrate our superiority over other methods on real images.

4.4. Internal analysis

In this section, we give a number of internal analysis of our proposed network. It mainly contains following five parts: ablation study on different components of our proposed network, the effect on different levels, the effect on the number of MLGRBs, the effect on model sizes and the effect on whether the parameters are shared.

Table 2
User study on real-world data. ‘Selected’ represents the number of most voted selection.

Meassure	Not sure	DDN [17]	JORDER [18]	RESCAN [20]	DID [22]	Ours
Voted	157	111	104	128	147	193
Selected	16	7	6	11	17	27

4.4.1. Ablation study

Firstly, we discuss the effectiveness of our proposed each component, including, fusion connections, dilation, guided learning and refinement process. Their abbreviation is provided as follows and the results are shown in Table 3. Symmetry skip connections are usually utilized in many low-level visions, which transform low-level features to high-level features in order to enable the computation of long-range spatial dependencies as well as efficient usage of the feature activation of proceeding layers. However, our proposed fusion connections are to cascade all layers to fuse features at every level. So it is meaningful to discuss these connection styles. R_1 and R_7 denote symmetry skip connections and our proposed fusion connections, respectively. We can see that our proposed fusion connections are more effective that the PSNR is improved by 0.04. R_2, R_3 and R_4 denote whether having dilation convolutions or guided learning. R_7 denotes the proposed network with dilation convolution and guided learning. It can be observed that our used guided learning and dilation convolution improve the deraining results. Compared without dilation convolution and guided learning, the proposed network, i.e. R_7, improves the PSNR and SSIM by 0.13 dB and 2%, respectively. The promotion is obvious and also illustrates the effectiveness of our proposed two components. Moreover, the feature maps at the first MLGRB of R_2, R_3 and R_4 are visualized in Fig. 9. The last row denotes our final network and we can see that it learns rain streaks information better and the background and rain streaks information is also more vivid. This also illustrates that the dilation convolution and guided learning are useful for our deraining results. Moreover, we discuss the two forms of guided learning in MLGRB: element-wise sum (R_5) and 1 × 1 convolution fusion (R_7). Although the element-wise sum also can guide the learning from the layer with a smaller receptive field to the layer with the larger receptive field, it does not learn the fusion that which parts are effective for different layers. However, the 1 × 1 convolution learns the fusion between different layers to boost the guided learning. The experimental results also verify our design scheme in R_5 and R_7. R_6 and R_7 compare the effect whether having refinement process. The comparison shows that the refinement is indeed effective and the PSNR and SSIM are improved by 0.34 dB and 2%, respectively. From the above, our proposed each component is effective and boost the deraining results better.

- R_1 : The proposed method using symmetry skip connections.
- R_2 : The proposed method without dilation convolutions and guide learning.
- R_3 : The proposed method without guide learning.
- R_4 : The proposed method without dilation convolutions.
- R_5 : Replacing the 1 × 1 convolution with element-wise sum in MLGRB.
- R_6 : The proposed method without refinement process.
- R_7 : Our proposed network with dilation convolutions, guide learning, refinement process and fusion connections.

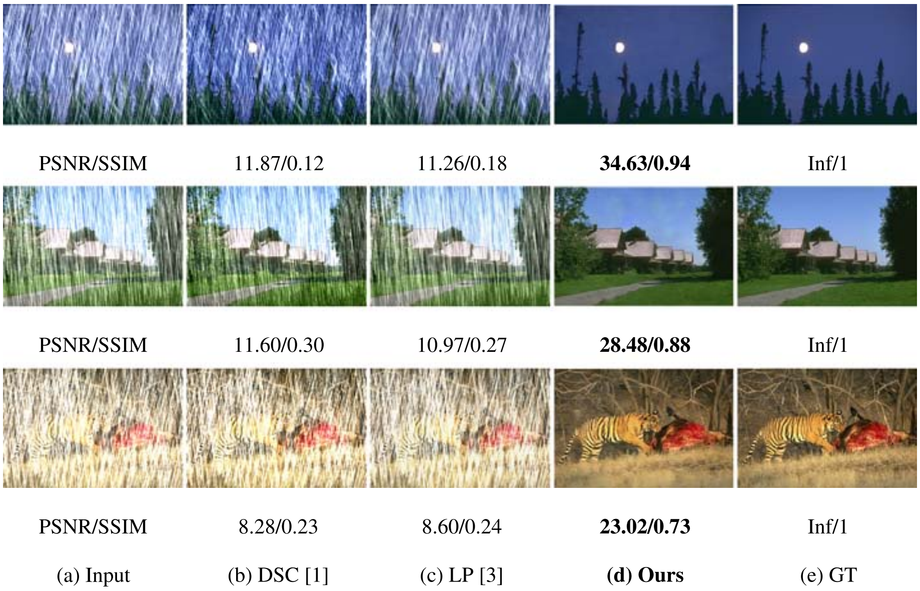


Fig. 3. Several examples in synthetic datasets compared with priors based deraining methods. It is obvious that our method is much better than other methods.

Table 3
Ablation study on different components of our proposed network. The √ symbol denotes that the corresponding component is adopted.

Experiments	R_1	R_2	R_3	R_4	R_5	R_6	R_7
Symmetry skip connections	√						
Fusion connections		√	√	√	√	√	√
Dilation convolution	√		√		√	√	√
Guided learning	√			√	√	√	√
Element-wise sum					√		
Refinement	√	√	√	√	√		√
PSNR	27.48	27.39	27.11	27.01	26.91	27.18	27.52
SSIM	0.86	0.84	0.85	0.85	0.85	0.84	0.86

Table 4
The effect of levels. L denotes the number of levels.

Metric	L = 2	L = 3	L = 4	L = 5	L = 6
PSNR	26.44	27.12	27.52	27.48	28.26
SSIM	0.84	0.85	0.86	0.86	0.86

Furthermore, we provide one visual example on ablation study in Fig. 8. The result of proposed network shown in Fig. 8(g) gains the highest PSNR and SSIM. And the other results have either artifacts or remaining rain streaks.

4.4.2. *The effect on levels*

We discuss the effect on levels in the section and the results are illustrated in Table 4. It can be observed that when L less than 4, the PSNR is increasing as the L becomes big and the value of PSNR becomes small when L is more than 4. The PSNR is the highest when $L = 4$. So we select $L = 4$ as our network settings.

4.4.3. *The effect on the number of MLGRBs*

We further analyze the effect on the number of MLGRBs and the curves of PSNR and SSIM are illustrated in Fig. 10(a) and (b), respectively. We can see that the results are better when $N = 10$, and the SSIM declines when N increases to 12. And the PSNR and SSIM are increasing as the N becomes large when N is less than 10. Considering the results are good enough and outperform all state-of-the-art methods when $N = 10$, so we set $N = 10$ as our network settings.

Table 5
Results on model sizes. C denotes the number of channels.

Metric	C = 12	C = 16	C = 20	C = 24	C = 28
PSNR	25.59	27.21	27.52	27.82	28.19
SSIM	0.84	0.85	0.86	0.87	0.88
Parameters	37,026	65,110	101,066	144,894	196,594

4.4.4. *The effect on model sizes*

We also discuss the effect on model sizes. We believe a good model should be flexible that small model sizes also have a good deraining performance and the results should become better with model sizes increasing. For this purpose, we carry out several experiments on different model sizes. We change model sizes by increasing or decreasing the number of channels. The results are shown in Table 5 and corresponding curves of PSNR and SSIM are illustrated in Fig. 11. We can observe that the results become better with the model sizes increasing. When the model sizes are small, i.e., $C = 12$, the results are comparable with other state-of-the-art methods and even outperform most methods by combining with Table 1. Moreover, the results are far more than all state-of-the-art methods when $C = 28$ and the number of parameters only has 196,594. We also provide a visual example as a comparison in Fig. 12. It is obvious that the deraining performance gets better with the model sizes increasing.

4.4.5. *The effect on whether parameters are shared*

At last, we discuss the effect on whether parameters are shared. To reduce the model sizes, we set the parameters to be shared in every MLGRBs. It is meaningful to analyze the change when the parameters are independent. For this purpose, we conduct the experiments when the parameters are independent and the results are shown in Table 6. As we can see that the results have a big improvement when the parameters are independent and the parameters are three times as big as the shared condition. Our model is flexible that the results are better and the parameters are less when the parameter are shared and the results are far better to the other state-of-the-art methods then the parameters are independent. So we conclude that the shared parameters strategy is also feasible. Furthermore, we present a visual example to compare the two conditions in Fig. 13. We can see that the deraining result has a huge improvement compared with the shared condition.

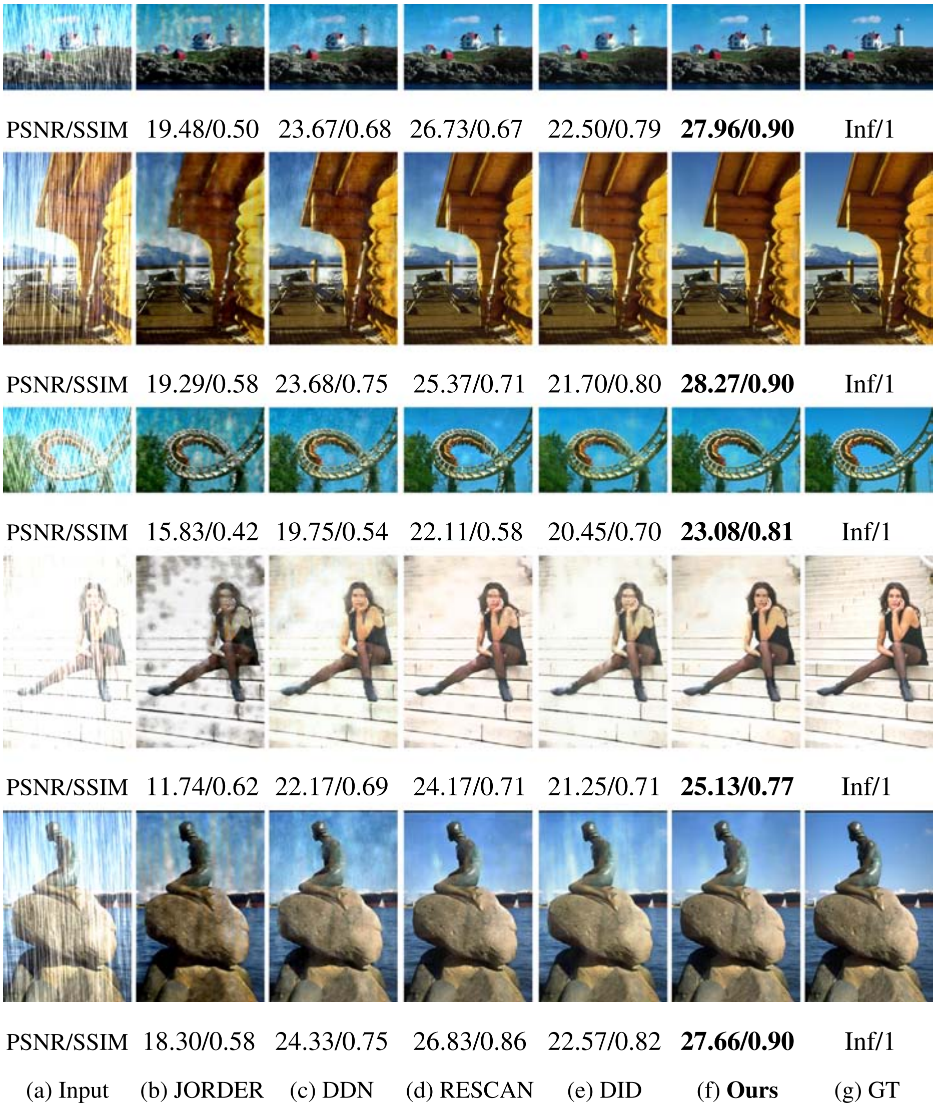

PSNR/SSIM 19.48/0.50 23.67/0.68 26.73/0.67 22.50/0.79 **27.96/0.90** Inf/1
PSNR/SSIM 19.29/0.58 23.68/0.75 25.37/0.71 21.70/0.80 **28.27/0.90** Inf/1
PSNR/SSIM 15.83/0.42 19.75/0.54 22.11/0.58 20.45/0.70 **23.08/0.81** Inf/1
PSNR/SSIM 11.74/0.62 22.17/0.69 24.17/0.71 21.25/0.71 **25.13/0.77** Inf/1
PSNR/SSIM 18.30/0.58 24.33/0.75 26.83/0.86 22.57/0.82 **27.66/0.90** Inf/1
(a) Input
(b) JORDER
(c) DDN
(d) RESCAN
(e) DID
(f) **Ours**
(g) GT

Fig. 4. Several examples in synthetic datasets compared with deep-learning based deraining methods. Our results shown in (f) are the best than others. And the results of JORDER [18] always restore to be darker.

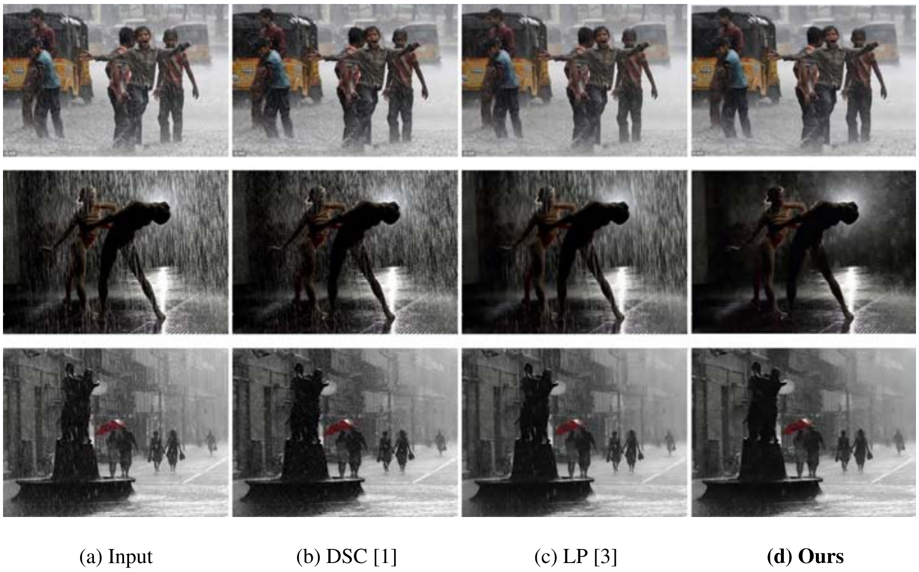


Fig. 5. Several examples in real-world datasets compared with priors based deraining methods. It is obvious that our method is much better than other methods.

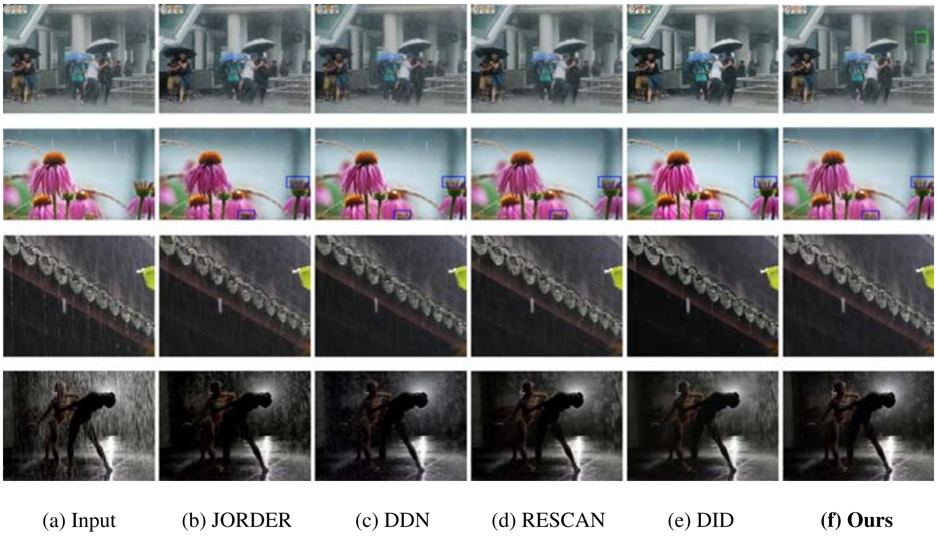


Fig. 6. Several examples in real-world datasets compared with deep-learning based deraining methods. Our results shown in (f) have less artifacts and obtain clearer texture information than others.

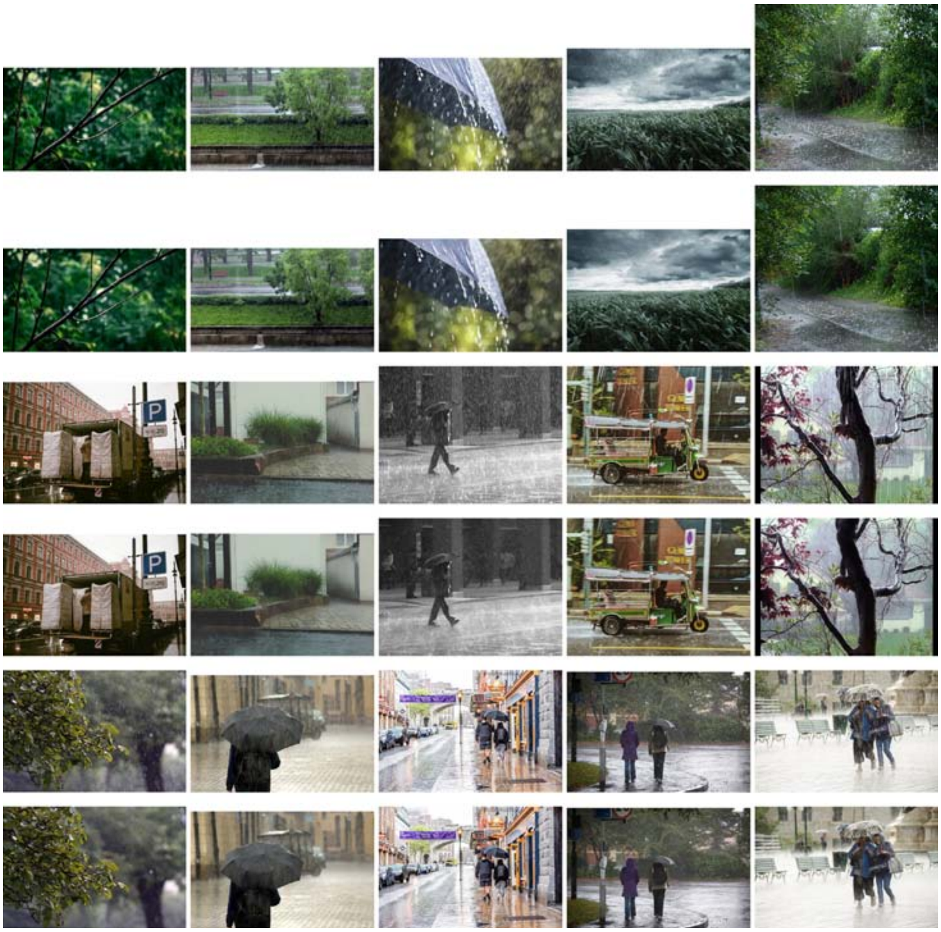
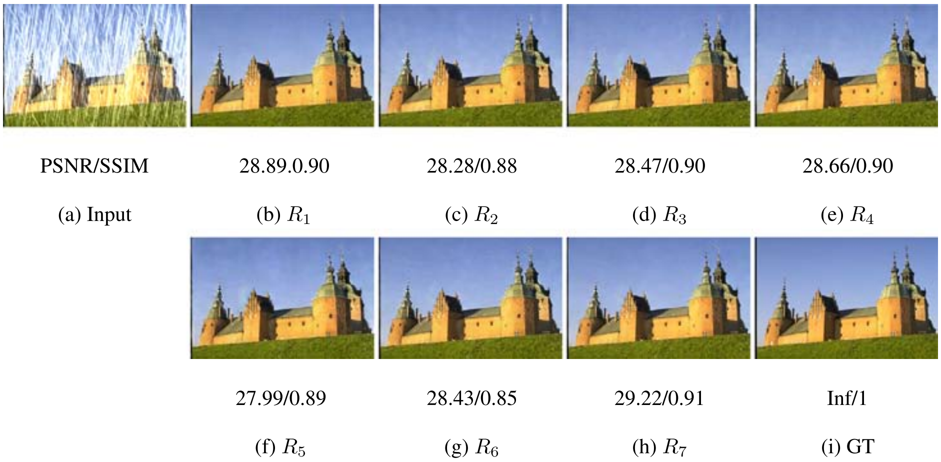


Fig. 7. More our deraining results in real-world dataset.

5. Conclusion

In this paper, we propose a novel end-to-end deep learning based deraining method. The method utilizes dilation convolutions to obtain more spatial contextual information to capture rain streaks with different sizes. The guided learning is proposed that the layer with a smaller receptive field is to guide the layer with the larger receptive field in order to learn rain streaks better. Experiments illustrate that the dilation convolution and guide learning improves the deraining performance. Quantitative and qualitative experimental results demonstrate the superiority of the proposed method compared with several state-of-the-art deraining methods on Rain100H, Rain100L and Rain1200 datasets.


PSNR/SSIM 28.89.0.90 28.28/0.88 28.47/0.90 28.66/0.90

(a) Input (b) R_1 (c) R_2 (d) R_3 (e) R_4

27.99/0.89 28.43/0.85 29.22/0.91 Inf/1

(f) R_5 (g) R_6 (h) R_7 (i) GT

Fig. 8. An example on ablation study.

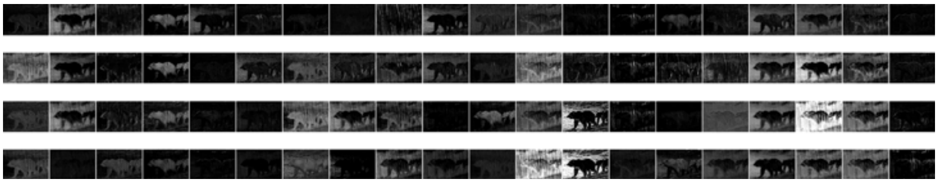
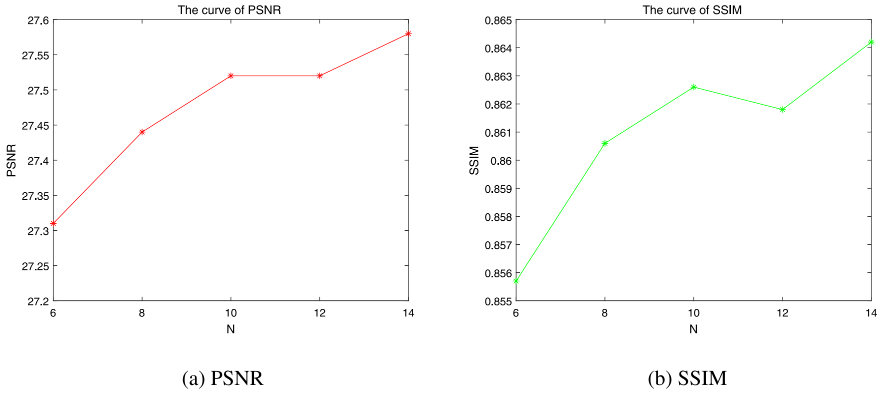
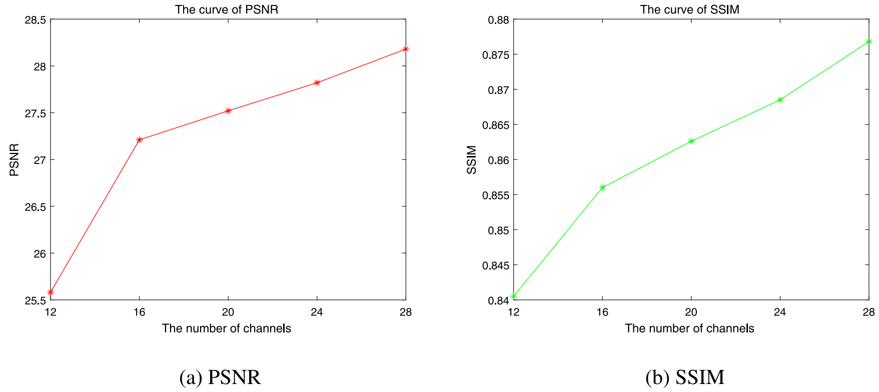


Fig. 9. The visualization of feature maps in the first MLGRB on different networks settings. From top to bottom, the proposed method without dilation convolution and guide learning, without dilation convolution, without guide learning and with dilation convolution and guide learning, respectively.



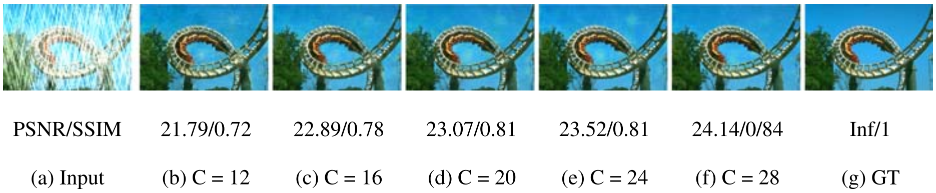
(a) PSNR (b) SSIM

Fig. 10. The curves of the number of MLGRBs.



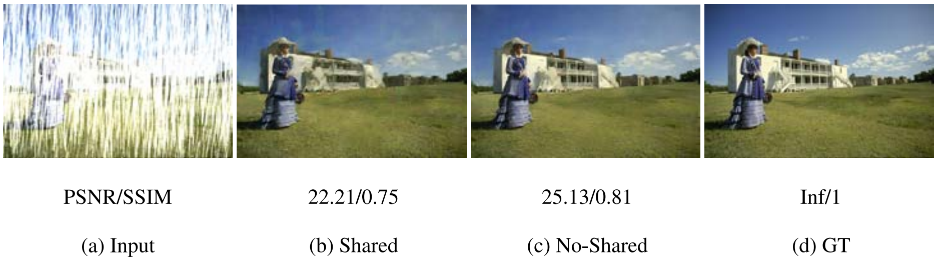
(a) PSNR (b) SSIM

Fig. 11. The curves of model sizes.



PSNR/SSIM	21.79/0.72	22.89/0.78	23.07/0.81	23.52/0.81	24.14/0/84	Inf/1
(a) Input	(b) C = 12	(c) C = 16	(d) C = 20	(e) C = 24	(f) C = 28	(g) GT

Fig. 12. An example on model sizes.



PSNR/SSIM	22.21/0.75	25.13/0.81	Inf/1
(a) Input	(b) Shared	(c) No-Shared	(d) GT

Fig. 13. An example on whether parameters are shared.

Table 6
Results on whether parameters are shared.

Metric	Shared	No-shared
PSNR	27.52	28.91
SSIM	0.86	0.89
Parameters	101,066	345,846

Acknowledgments

This work was supported by the Natural Science Foundation of China [grant numbers 61572099]; Major National Science and Technology Project of China [grant number 2018ZX04011001-007].

References

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

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

[3] Y. Li, R.T. Tan, X. Guo, J. Lu, M.S. Brown, Rain streak removal using layer priors, in: CVPR, 2016, pp. 2736–2744, http://dx.doi.org/10.1109/CVPR.2016.299.

[4] J. Kim, C. Lee, J. Sim, C. Kim, Single-image deraining using an adaptive nonlocal means filter, in: ICIP, 2013, pp. 914–917, http://dx.doi.org/10.1109/ICIP.2013.6738189.

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

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

[7] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: CVPR, 2015, pp. 3431–3440, http://dx.doi.org/10.1109/CVPR.2015.7298965.

[8] 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.

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

[10] 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, http://dx.doi.org/10.1109/TIP.2016.2598681.

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

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

[13] 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, http://dx.doi.org/10.1109/TPAMI.2015.2439281.

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

[15] J. Yu, Y. Fan, J. Yang, N. Xu, Z. Wang, X. Wang, T. Huang, Wide Activation for Efficient and Accurate Image Super-Resolution, Vol. abs/1808.08718, 2018, arXiv:1808.08718. URL http://arxiv.org/abs/1808.08718.

[16] 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, http://dx.doi.org/10.1109/TIP.2017.2691802.

[17] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, J. Paisley, Removing rain from single images via a deep detail network, in: CVPR, 2017, pp. 1715–1723, http://dx.doi.org/10.1109/CVPR.2017.186.

[18] 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, http://dx.doi.org/10.1109/CVPR.2017.183.

[19] 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.

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

[21] G. Li, X. He, W. Zhang, H. Chang, L. Dong, L. Lin, Non-locally enhanced encoder–decoder network for single image de-raining, in: ACM MM, 2018, pp. 1056–1064, http://dx.doi.org/10.1145/3240508.3240636, URL http://doi.acm.org/10.1145/3240508.3240636.

[22] H. Zhang, V.M. Patel, Density-aware single image de-raining using a multi-stream dense network, in: CVPR, 2018, pp. 695–704, http://dx.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.

[23] D. Huang, L. Kang, M. Yang, C. Lin, Y.F. Wang, Context-aware single image rain removal, in: ICME, 2012, pp. 164–169, http://dx.doi.org/10.1109/ICME.2012.92.

[24] K. Garg, S.K. Nayar, Detection and removal of rain from videos, in: CVPR, 2004, pp. 528–535, http://dx.doi.org/10.1109/CVPR.2004.79.

[25] 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, http://dx.doi.org/10.1007/978-3-540-89689-0_49.

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

[27] A.K. Tripathi, S. Mukhopadhyay, Removal of rain from videos: a review, Signal, Image Video Process. 8 (8) (2014) 1421–1430, http://dx.doi.org/10.1007/s11760-012-0373-6.

[28] 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, http://dx.doi.org/10.1109/TIP.2011.2179057.

[29] 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, http://dx.doi.org/10.1109/TIP.2017.2708502.

[30] Z. Fan, H. Wu, X. Fu, Y. Huang, X. Ding, Residual-guide network for single image deraining, in: ACM MM, 2018, pp. 1751–1759, http://dx.doi.org/10.1145/3240508.3240694.

[31] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: CVPR, 2018, pp. 7132–7141, http://dx.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.

[32] Q. Huynh-Thu, M. Ghanbari, Scope of validity of psnr in image/video quality assessment, Electron. Lett. 44 (13) (2008) 800–801.

[33] 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, http://dx.doi.org/10.1109/TIP.2003.819861.

[34] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: CoRR, Vol. abs/1412.6980, 2014, arXiv:1412.6980. URL http://arxiv.org/abs/1412.6980.