The Visual Computer
https://doi.org/10.1007/s00371-020-01944-z

ORIGINAL ARTICLE

Single image deraining via deep shared pyramid network

Cong Wang[1] · Xiaoying Xing[2] · Guangle Yao[3] · Zhixun Su[1,4]

Abstract

Single image deraining is a highly ill-posed problem. Existing deep neural network-based algorithms usually use larger deep models to solve this problem, which is less effective and efficient. In this paper, we propose a deep neural network based on feature pyramid to solve image deraining. Our algorithm is motivated that the features at different pyramid levels share similar structures. Based on this property, we develop an effective deep neural network, where the deep models at different feature pyramid levels share the same weight parameters. In addition, we further develop a multi-stream dilation convolution to deal with complex rainy streaks. To preserve the image detail, we develop dense connections that can maintain important features from different levels. Our algorithm is trained in an end-to-end manner. Quantitative and qualitative experimental results demonstrate that the proposed method performs favorably against state-of-the-art deraining methods in terms of accuracy as well as model sizes. The source code and dataset will be available at https://supercong94.wixsite.com/supercong94.

Keywords Single image deraining · Deep neural network · Shared parameters · Pyramid network · Dense connections

1 Introduction

Lots of vision and multimedia systems usually rely on high-definition images or videos, e.g., object detection [30], object tracking [40], autonomous driving [1] and so on. However, the images and videos captured in a rainy environment usually contain significant rainy streaks, which fails most vision and multimedia tasks. Thus, it is necessary to develop algorithms that can automatically restore clear images from rainy images.

Image deraining has attracted much attention in the past years. A lot of methods have been proposed to solve this problem. The main success of these algorithms is due to the use of kinds of image priors [2,21,25] or deep neural networks [9,19,20,32,35]. Mathematically, a rainy image can be modeled as a linear combination of a rain streak component with a clean background image:

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

where O, B and R denote rainy image, clear image and rainy streaks, respectively. As only the rainy image is available, this problem is highly ill-posed.

To make this problem well proposed, numerous algorithms use prior knowledge about rainy streaks and clear images, e.g., low-rank prior [2], sparse representation [25], Gaussian mixture model [21], to constrain the solution space. Although these algorithms achieve promising performance, the prior knowledge used in these algorithms does not hold for some cases. Hence, more adaptive and efficient methods, which can deal with the problem of different rainy streaks in any case, are needed.

Motivated by the success of convolutional neural networks (CNNs) in many computer vision tasks, e.g., object detection [30], object tracking [40], semantic segmentation [24], super-resolution [5,6], style transfer [10,14], deblurring [33], dehazing [7,18,27,34,37–39], the CNNs have been developed to solve image deraining [8,9,19,20,32,35]. These deraining methods generally model the problem as a pixel-wise image regression process which directly learns to map an input rainy image to its clean one or a negative residual map in an end-to-end trainable CNNs. Among them, [8,9,19,20] proposed different network structures by consid-

✉ Cong Wang
supercong94@gmail.com

[1] Dalian University of Technology, Dalian, People's Republic of China

[2] Tsinghua University, Beijing, People's Republic of China

[3] Chengdu University of Technology, Chengdu, People's Republic of China

[4] Guilin University of Electronic Technology, Guilin, People's Republic of China

Published online: 01 August 2020

ering property of rain or rain streak feature. Different from designing network structure, [32,35] considered rain streak detection or estimating rain streak density into rain streak removal procedure. Although considerable progress has been made in comparison with traditional methods, existing algorithms [19,20,32,35] usually use larger deep models that are less effective and efficient. For example, Fig. 1 shows the results of ours and other state-of-the-art deraining methods. We can see that other state-of-the-art methods are inefficient and our algorithm generates better deraining performance utilizing small size network depending on the shared features that can be obtained through the different levels of the pyramid network.

Moreover, spatial pyramid features have been applied to many vision tasks and achieve excellent results. The pyramid manner usually utilizes max pooling or mean pooling operation to obtain. In [3,22,26,29], these methods take advantage of the pyramid to improve the performance of corresponding visual problems. However, these algorithms have a common disadvantage that the parameters in different levels of the pyramid are independent, which enlarges the model sizes. Hence, in this paper, by the proposed pyramid-based network, we have explored the dependency between the network levels which helps to shrinkage the proposed model size.

To overcome these problems, we develop an effective deep neural network based on the feature pyramid for image deraining. Our algorithm is motivated that the features at different image pyramid levels share similar structures. By assuming that the deep models at different pyramid levels share the same weight parameters, the proposed deep model is able to remove rain streaks and has a smaller model size. To deal with complex rainy streaks and preserve the image details, we develop a multi-stream dilation convolution (MSDC) and dense connections that can get a larger receptive field to obtain more rain streak information and maintain the important features from different levels, respectively, as shown in Fig. 1. By training in an end-to-end manner, the proposed algorithm performs favorably against state-of-the-art methods in terms of accuracy and model sizes.

The contributions of this paper are summarized as follows:

- We propose an effective deep neural network based on the feature pyramid for image deraining, where the deep models at different pyramid levels share the same weight parameters.
- We develop a multi-stream dilation convolution and dense connections that can maintain the important features from different levels to deal with complex rainy streaks and preserve the image details.
- We create a synthetic rainy Pascal VOC 2012 dataset to evaluate the improvement of performance by incorporating with deraining methods for the high-level vision tasks. As we know, the rainy evaluative dataset for high-level vision is firstly synthesized by us.
- We show that the proposed algorithm is able to remove rain streaks and preserve image details. Quantitative and qualitative experimental evaluations on both synthetic datasets and real-world datasets demonstrate that the proposed algorithm outperforms the state-of-the-art methods.

2 Related work

In this section, we present a brief review of the recent related works.

2.1 Single image deraining

As aforementioned, the single image deraining methods can be grouped into two categories: prior-based methods and deep learning-based methods.

Prior-Based Methods: Prior-based methods are the previous pioneers of the deraining problem. Kang et al. [15] assumed that rain streaks were a high-frequency structure and separated the rain streaks by utilizing sparse coding from HOG features in high-frequency layer. Kim et al. [16] directly regarded it as an image filtering problem and solved it by resorting to nonlocal mean smoothing. Luo et al. [25] proposed a discriminative sparse coding framework based on image patches and separated rain streaks from rain-free background images. Chen et al. [2] believed that the rain streak layer was of low rank and utilized generalized a low-rank model to separate rain streaks. Li et al. [21] developed a Gaussian mixture model to derain using layer prior.

Deep Learning-Based Methods: Recently, several deep learning-based deraining methods have achieved great success. Fu et al. [8,9] are the pioneers of applying the deep learning techniques to single image deraining. They decomposed rainy images into low- and high-frequency parts and mapped high-frequency parts to rain streaks by a deep residual network, and lastly, they utilize Eq. 1 to obtain a clean image. Yang et al. [32] proposed a joint rain streak detection and removal method. Considering the hazy condition into the rainy model, they applied a dehazing–deraining–dehazing algorithm to solve the complex situation. Li et al. [19] came up with a multi-scale nonlocal enhanced encoder–decoder network that mapped rainy images to clean images via learning the residual by considering the pixel-wise attention mechanism. Li et al. [20] recurrently utilized convolutional neural networks with dilation factors and squeeze-and-excitation [11] blocks to remove heavy rain streaks. Zhang et al. [35] proposed a multi-stream densely connected convolutional neural network to guide rain streak removal by estimating rain density.

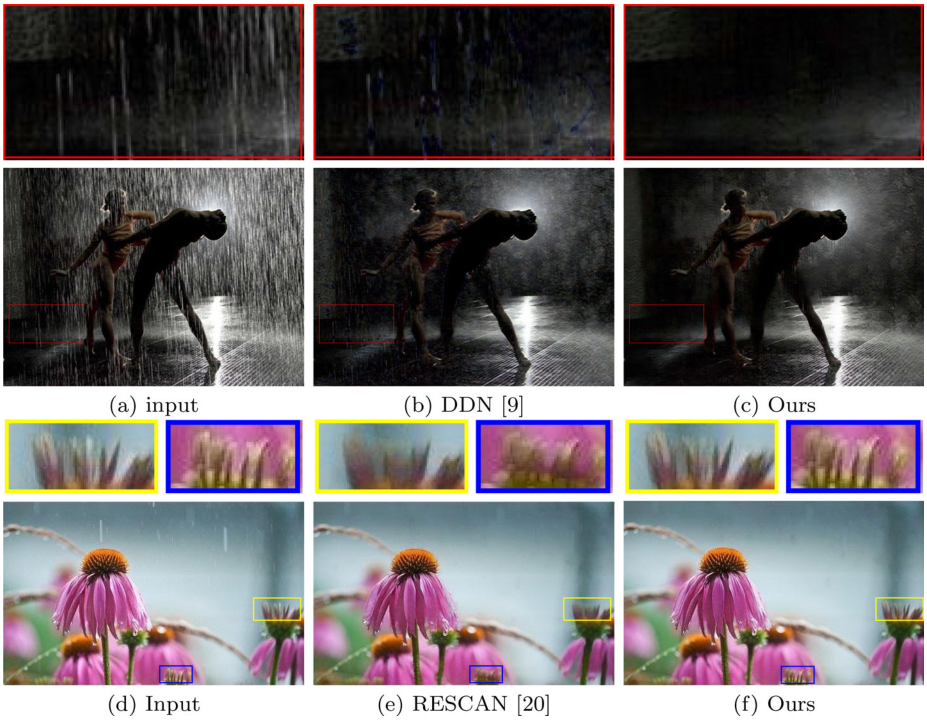


Fig. 1 Image deraining examples. The proposed algorithm is able to remove rain from the rainy images and generates better images with finer details. The DDN [9] does not have a large receptive field that results in residual rainy streaks. The RESCAN [20] neglects to fuse the features at different levels among layers so as to lose the image details

2.2 Pyramid network

Recently, the conventional spatial pyramid approaches have been combined successfully with neural network architectures to deal with various vision tasks. There are several networks based on spatial pyramid. Ranjan et al. [26] proposed a spatial pyramid network to estimate optical flow, where they utilized a coarse-to-fine manner to estimate large motions by warping one image. Different from standard minimization of an objective function at each pyramid level, they computed the flow update by training one deep neural convolutional network at per level of pyramid. Lin et al. [22] proposed a feature pyramid network for object detection that they constructed feature pyramids with marginal extra cost by using pyramidal hierarchy of deep networks. Chen et al. [3] came up with a cascaded pyramid network for multi-person pose estimation by designing the global-net and refine-net. In particular, in the deraining task, Wang et al. [4] proposed a deep pyramid model to solve the image deraining problem, where they did not consider the shared strategy, leading to bigger model size.

To sum up, although they have achieved well-pleasing performance, all of them have a common characteristic that parameters at different levels of the pyramid are independent. The feature similarity of the pyramid is not fully utilized so as to the heavy weight networks.

3 Proposed method

We develop an end-to-end convolutional neural network for single image deraining, which is a fully convolutional network that has been proved to be able to learn complex pixel-wise mappings from a large amount of input–output image pairs. The overall architecture of the proposed network is illustrated in Fig. 2. As features at different levels of the pyramid have similar structure (we discuss this in Sect. 5), we introduce shared parameter strategy to different levels of the pyramid. Each level of pyramid can learn different rain streak information so that the overall network are boosted to learn the most useful rain streak information.

3.1 Overall network framework

The overall network framework is shown in Fig. 2. As rainy streaks have simpler structure than clear images, the network learns the map from rainy images to rainy streaks and obtains final clean images by utilizing Eq. 1. To obtain more spatial contextual information, we develop a multi-stream dilation convolution (MSDC) as our basic component which will be introduced in detail in Sect. 3.2. Moreover, in order to boost the information flow along with features from different levels, we use dense connections to connect these layers at the same level of the pyramid. Several MSDCs and dense connections make up our network.

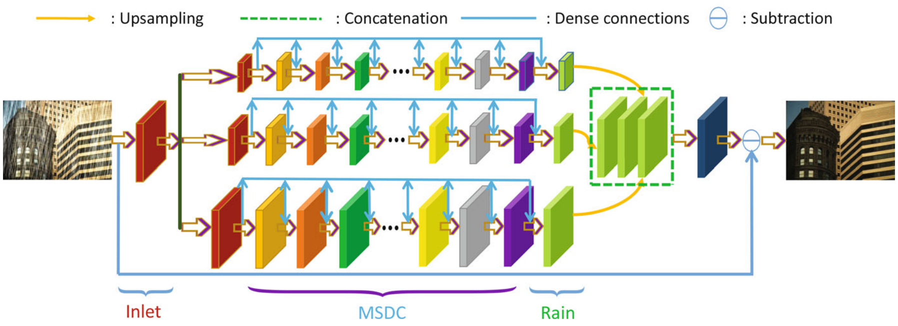


Fig. 2 Proposed network framework. The parameters are shared in the same color blocks at the same positional layer of different levels of pyramid. Inlet layer is to convert image space to feature space, i.e., F_0. MSDC denotes multi-stream dilation convolution shown in Fig. 3. Rain layer is estimated rainy streaks, i.e., $F_{i,rain}$

Mathematically, we describe this overall network as follows:

$$F_0 = \text{Conv}_{3\times3}(O), \tag{2}$$

where O and F_0 denote the input of rainy image and the shallow features, respectively. $\text{Conv}_{3\times3}$ denotes the convolution operation with the kernel size of 3×3 pixels. This operation is to convert image space into feature space.

Then the original features at different levels of the pyramid can be acquired by using max-pooling operation on shallow features:

$$F_{i,0} = \mathcal{P}_i(F_0), i = 1, 2, \ldots, 2^{K-1}. \tag{3}$$

Here, $\mathcal{P}_i$ denotes max-pooling operation, where its stride and the kernel size are i. K denotes the number of pyramid level, and $F_{i,0}$ denotes the input of ith level of the pyramid.

Dense connections are used to connect MSDCs densely to boost the information flow and preserve the image details:

$$F_{i,l} = \mathcal{M}_l(\text{Conv}_{1\times1}(\mathcal{C}[F_{i,l-1}, \ldots, F_{i,0}])), \tag{4}$$

where $\mathcal{C}$ denotes the concatenation operation and $\text{Conv}_{1\times1}$ denotes the convolution operation with kernel size of 1×1 pixels. $\mathcal{M}$ denotes MSDC operation in Fig. 3 and it will be described in Sect. 3.2 in details. $F_{i,l}$ denotes the output of lth MSDC operation at ith level of pyramid and $l = 1, 2, \ldots, L$.

The rainy streak layer $F_{i,\text{rain}}$ is obtained by cascading all MSDCs in order to obtain features at different levels:

$$F_{i,\text{rain}} = \text{Conv}_{1\times1}(\mathcal{C}[F_{i,L}, \ldots, F_{i,0}]), \tag{5}$$

The final estimated rain streak layer $\tilde{R}$ is that cascades all rain streaks at different levels by upsampling to the original size of rainy image:

$$\tilde{R} = \text{Conv}_{1\times1}(S_1(F_{1,\text{rain}}), \ldots, S_i(F_{i,\text{rain}})), \tag{6}$$

where S_i denotes the upsampling operation with the scale factor i.

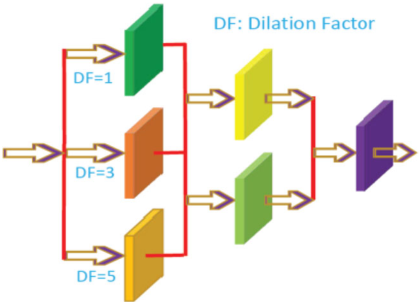


Fig. 3 Multi-stream dilation convolution (**MSDC**)

Finally, we obtain the estimated rain-free image $\tilde{B}$ via Eq. 1:

$$\tilde{B} = O - \tilde{R}, \tag{7}$$

3.2 Multi-stream dilation convolution

As the spatial contextual information is important for single image deraining [12], we use the multi-stream dilation convolution to capture the important features at different image pyramid levels. For large rainy streaks, the large receptive field is needed to capture the information, while small rainy streaks can be estimated well by a smaller receptive field. Based on this fact, we develop a multi-stream dilation convolution (**MSDC**) to achieve this goal. The detailed architecture is shown in Fig. 3.

The MSDC operation can be represented as:

$$D_r = \text{Conv}_r(I), r = 1, 3, 5, \tag{8}$$

where I and D_r denote the input feature and corresponding output, respectively. r denotes the dilation factor based on 3×3 convolution.

To effectively learn the rainy streak information, we fuse different layers by

$$D_{k,j} = \sigma(\text{Conv}_{1\times1}(\mathcal{C}[D_k, D_j])), \tag{9}$$

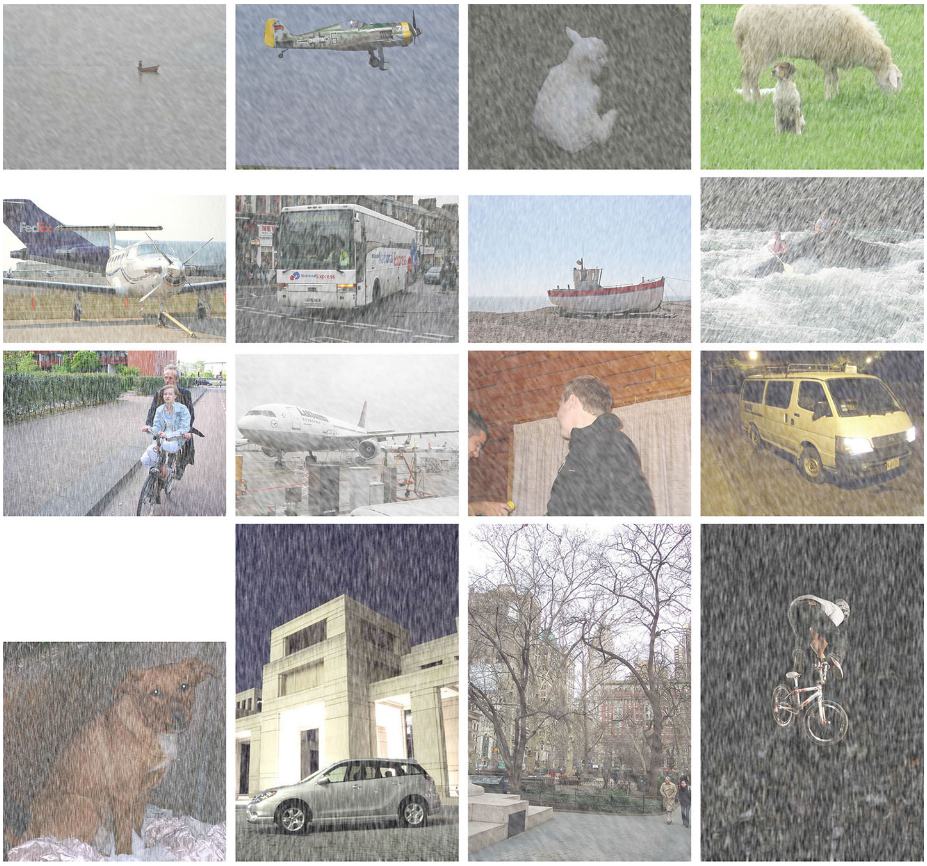


Fig. 4 Several examples in our created synthetic rainy dataset on Pascal VOC2012. They have dozens of different sizes, shapes and directions

where $D_{k,j}$ denotes the fusion output and σ denotes the activation function. Here, we select LeakyReLU with $\alpha = 0.2$ as σ.

Finally, the output of MSDC is:

$$\mathcal{MSDC} = \mathrm{Conv}_{1\times 1}(\mathcal{C}[D_{1,3}, D_{3,5}]), \tag{10}$$

Table 1 Quantitative experiments evaluated on three synthetic datasets

–	DSC ([25])		LP ([21])		DDN ([9])		JORDER ([32])		RESCAN ([20])		DID ([35])		$L_{10}C_6$		$L_{12}C_8$ (default)		$L_{14}C_{10}$	
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***	26.12	0.83	***26.21***	0.83	*26.90*	*0.85*	**27.56**	**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***	35.06	*0.97*	36.02	**0.98**	**36.84**	**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*	31.13	0.89	***31.57***	*0.90*	**32.50**	**0.91**
Parameters	–		–		58,175 (– 45%)		369,792 (– 91%)		54,735 (– 41%)		372,839 (– 91%)		14,865		32,075		59,453	

L_sC_t denotes the network that the number of MSDCs is s and the number of channels is t. The best, the second and third results are marked by bold, italic and bold italic, respectively. Please note that the JORDER [32] and DID [35] are trained by additional data (e.g., rain density level, rain mask annotation) provided by the datasets. The number in parentheses indicates the parameter reduction compared our default method, i.e., $L_{12}C_8$

Fig. 5 An example in synthetic datasets compared with prior-based deraining methods. The proposed method generates a much clear image

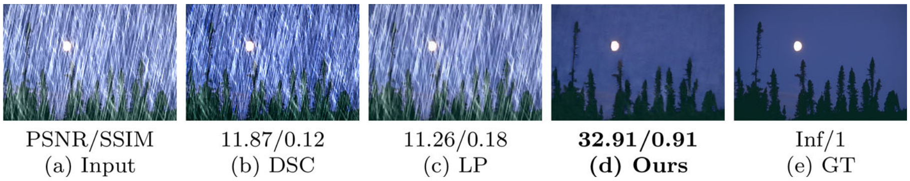


Fig. 6 Several examples in synthetic datasets compared with deep learning-based deraining methods

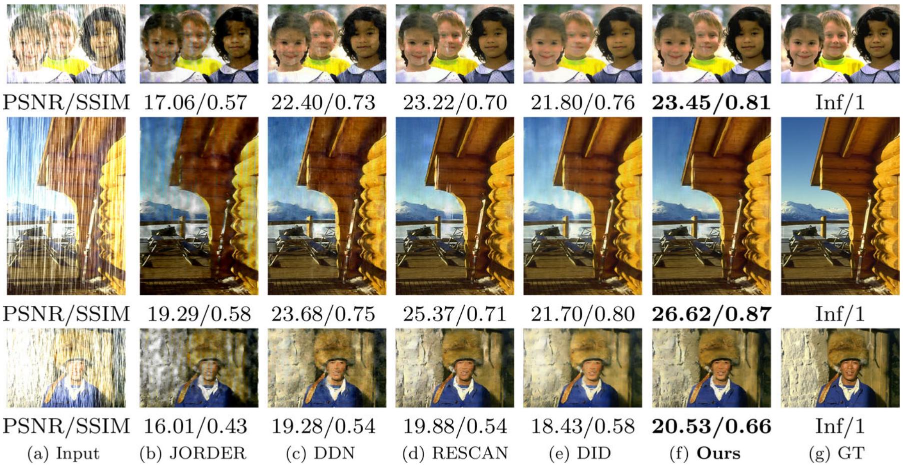


Fig. 7 A real-world example compared with prior-based deraining methods. The proposed method is able to remove rain and generates a much clear image

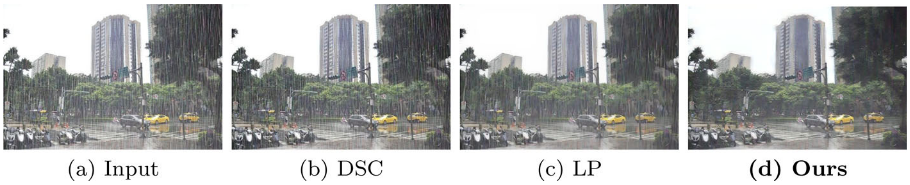
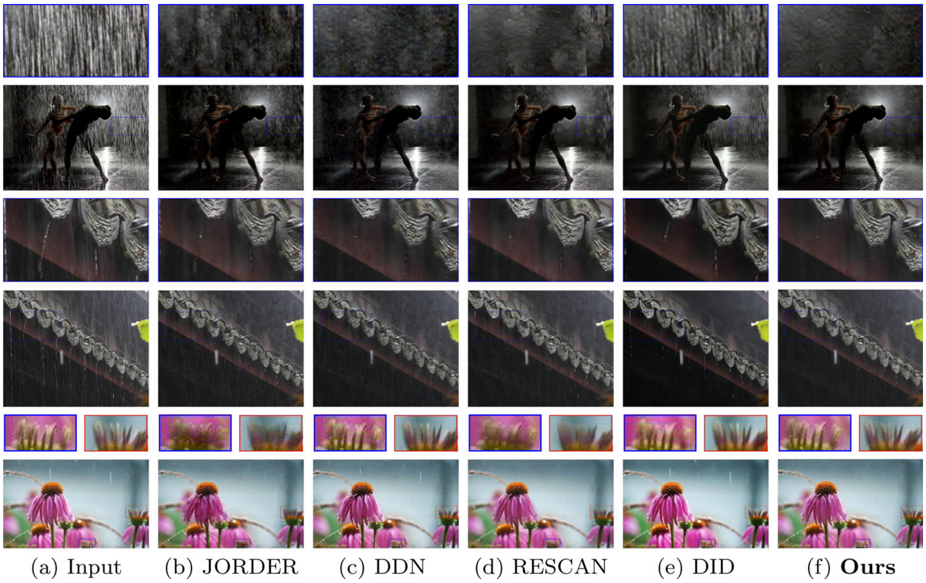


Fig. 8 Deraining results on real-world images. The proposed method is able to remove rain and generate much better results. Note that the DID [35] has a refinement processing after deraining. However, there still exist significant rain streaks in the restored images



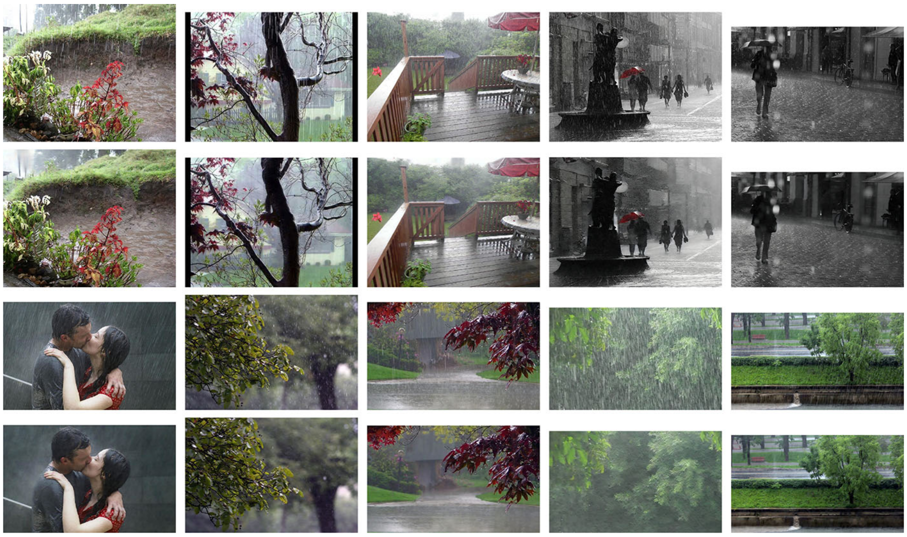


Fig. 9 More our deraining examples

3.3 Loss function

To train the proposed network, we use the mean square errors as the loss function which is defined as:

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

where H, W and C denote the height, width and channel number of a rain-free image, respectively; $\tilde{B}$ and B denote the estimated clean image and ground truth image, respectively.

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 [25] (ICCV15), LP [21] (CVPR16), DDN [9] (CVPR17), JORDER [32] (CVPR17), RESCAN [20] (ECCV18) and DID [35] (CVPR18).

4.1 Datasets and evaluation criteria

4.1.1 Synthetic datasets

We conduct deraining experiments on three widely used synthetic datasets: Rain100L [32], Rain100H [32] and Rain1200 [35]. These three datasets include various rain streaks that have 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 image for testing. It is ensured that all the testing datasets have different background images with training datasets. We select Rain100H as our ablation study analysis dataset.

4.1.2 Real-world datasets

Zhang et al. [36] and Yang et al. [32] also provide some real-world images, we use these images to evaluate the robustness on real-world images.

4.1.3 Our created rainy pascal VOC2012 dataset

We first create the rainy Pascal VOC2012 dataset to evaluate the improvement of performance by incorporating with deraining methods for the high-level vision tasks. The synthetic rainy images have dozens of different rainy streaks, including sizes, shapes and directions. Several samples are shown in Fig. 4.

4.1.4 Evaluation criteria

We use the peak signal-to-noise ratio (PSNR) [13] and structure similarity index (SSIM) [31] to evaluate the quality of the restored images on synthetic datasets. As there are no ground truth images for real-world images, we only show visual comparisons on the real-world datasets.

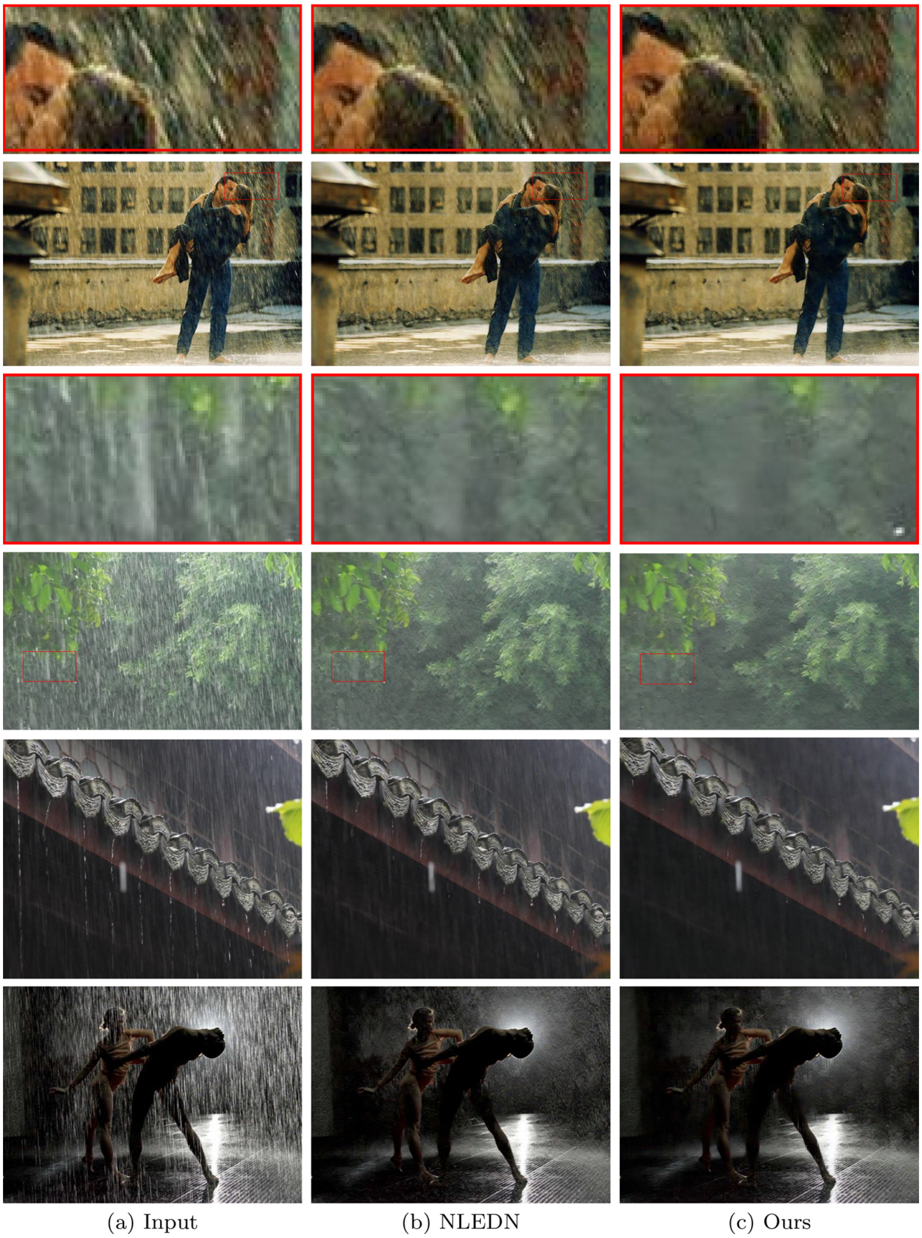
4.2 Experimental settings

We empirically set $L = 12$, $K = 3$ and the number of channels be to 8. We use the LeakyReLU with $\alpha = 0.2$ as the nonlinear activation function. We randomly crop image patches with size of 128 × 128 pixels from the training image datasets as the inputs and set the mini-batch size to be 10 to train the network. The ADAM [17] optimizer is used. The learning rate is initialized to be 0.001, and it will be divided

Table 2 Quantitative experiments evaluated on three synthetic datasets compared with NLEDN [19]

–	NLEDN ([19])		$L_{10}C_6$		$L_{12}C_8$ (default)		$L_{14}C_{10}$	
Dataset	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Rain100H	28.42	0.88	26.21	0.83	26.90	0.85	27.56	0.86
Rain100L	38.84	0.98	35.06	0.97	36.02	0.98	36.84	0.98
Rain1200	32.98	0.92	31.13	0.89	31.57	0.90	32.50	0.91
Parameters	1,005,379 (−97%)		14,865		32,075		59,453	

Number in parentheses indicates the parameter reduction compared our default method, i.e., $L_{12}C_8$


(a) Input

(b) NLEDN

(c) Ours

Fig. 10 Real-world examples compared with NLEDN [19]. We note that our method has better deraining performance than NLEDN [19], while decreasing 97% parameters

Table 3 Ablation study on basic component

Metric	Single	No dense	No dilation	Ours
PSNR	26.59	15.56	26.06	26.90
SSIM	0.84	0.33	0.83	0.85
Parameters	32,027	26,219	32,075	32,075

by 10 at 240K and 320K iterations. We train the network using 400K iterations on a PC with an NVIDIA GTX 1080Ti GPU. As our entire model is fully convolutional, the testing process only takes 0.024 seconds when handling a test image with 512 × 512 pixels on a PC with a GTX 1080Ti GPU.

4.3 Results on synthetic datasets

We compare our proposed network with six state-of-the-art methods, including two prior-based methods: DSC (ICCV15) [25] and LP (CVPR16) [21], and four deep learning-based methods: DDN (CVPR17) [9], JORDER (CVPR17) [32], RESCAN (ECCV18) [20] and DID (CVPR18) [35]. The results are shown in Table 1 and corresponding parameters are also specified. We can observe that the results of our method with the least parameters achieve better performances comparable to that of state-of-the-art methods. Note that the parameters are reduced by 91 percent compared with the newest state-of-the-art method: DID [35]. Our network is trained without any label; meanwhile, the JORDER [32] and DID [35] use the rainy streak mask and rainy streak density as label to guide the training, respectively. We also show the results on the version of our lightweight model, i.e., $L_{10}C_6$. This model has the advantage of the least parameters, while the decrease in performance is very small comparable with other state-of-the-art methods and even surpass them. Further, we also provide the version of our heavyweight model, i.e., $L_{14}C_{10}$, it is obvious that the model outperforms all state-of-the-art methods, while the parameters are almost the same with that of DDN [9] and RESCAN [20] and much less than JORDER [32] and DID [35].

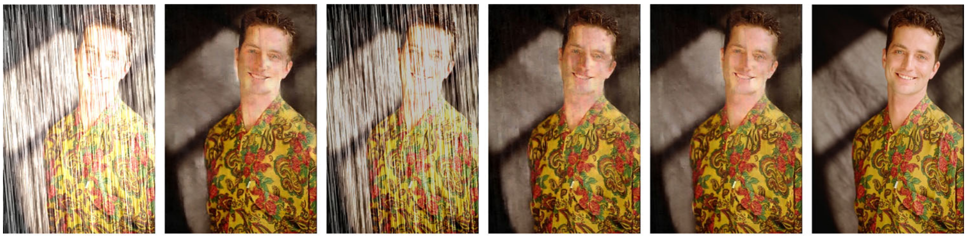
We also provide several examples as visual comparison. Figure 5 shows the results compared with prior-based methods [21,25]. It is obvious that our result is the best and the other results are unacceptable.

We further compare with deep learning-based methods, shown in Fig. 6. It can be observed that our results, shown in Fig. 6f, always obtain clearer texture information and have less artifacts. The other results either maintain residual rain streaks, e.g., Fig. 6b and c, or leave over more artifacts, e.g., Fig. 6d and e.

4.4 Results on real-world datasets

To verify the robustness, we compare our algorithm with state-of-the-art methods on real-world datasets. Firstly, we present one example compared with prior-based methods [21,25], illustrated in Fig. 7.

The other results hand down a mass of rainy streaks, while ours is the clearest and cleanest. Secondly, we display



PSNR/SSIM (a) Input | 24.14/0.78 (b) Single | 13.00/0.23 (c) No-dense | 23.78/0.76 (d) No-dilation | **25.13/0.82** (e) **Ours** | Inf/1 (f) GT

Fig. 11 Comparisons of the results by different baseline models. The proposed method generates a much better image as shown in (**e**)

Table 4 Results on different number of levels and dilation convolutions

	Metrics	$K = 2$	$K = 3$ (default)	$K = 4$
$D = 2$	PSNR	26.19	26.23	26.24
	SSIM	0.83	0.83	0.83
	Parameters	21,779	21,803	21,827
$D = 3$ (default)	PSNR	26.79	26.90	26.92
	SSIM	0.84	0.85	0.85
	Parameters	32,051	32,075	32,099
$D = 4$	PSNR	26.83	26.96	26.93
	SSIM	0.85	0.85	0.85
	Parameters	39,059	39,083	39,017

K and D denotes the levels of pyramid and dilation convolutions, respectively

Table 5 Results on different number of channels and MSDCs

	Metrics	$C = 6$	$C = 8$ (default)	$C = 10$
$L = 10$	PSNR	25.91	26.48	26.93
	SSIM	0.82	0.84	0.85
	Parameters	14,865	26,139	40,573
$L = 12$ (default)	PSNR	26.28	26.90	27.28
	SSIM	0.83	0.85	0.86
	Parameters	18,225	32,075	49,813
$L = 14$	PSNR	26.30	26.97	27.55
	SSIM	0.83	0.85	0.86
	Parameters	21,729	38,267	59,453

C and L denotes the number of channels and MSDCs, respectively

Table 6 Result on object detection and semantic segmentation are in the tables above and below, respectively

Metric	Rain	Derain
mIU	0.311	0.466
mAP	0.66	0.74

three examples compared with deep learning-based methods [9,20,32,35], shown in Fig. 8. Our results shown in Fig. 8f have the least artifacts. For the first example, the results of JORDER [32], shown in Fig. 8b, DDN [9], shown in Fig. 8c, and DID [35], shown in Fig. 8e, have lots of residual rainy streaks, while the result of RESCAN [20], shown in Fig. 8d, leave over many artifacts. For the second example, our result almost removes all the rain streaks, while the results of DDN [9], RESCAN [20] and DID [35], shown in Fig. 8c, d and e, respectively, hand down some rain streaks. Please note that the DID [35] has a refinement processing after deraining that it can be seen as dehazing procedure, so their results look like fog-free, while there exists fog in our result, because we do not have any post-processing. For the third example, our method gains better texture information in masked boxes, while JORDER [32] and RESCAN [20] lose the detail information. Moreover, DDN [9] and DID [35] maintain some rain streaks and our method is able to remove all rain streaks to obtain the cleanest image.

We provide more our deraining examples in Fig. 9. It can be seen that our method is able to process various rain streaks to generate better rain-free images.

4.5 Comparison with NLEDN

In particular, we compare our method with NLEDN [19] on synthetic datasets in Table 2. We note that our results are comparable with NLEDN, while the parameters are drastically reduced. Our model only has 32,075 parameters, while NLEDN has more than 1,000,000 parameters.

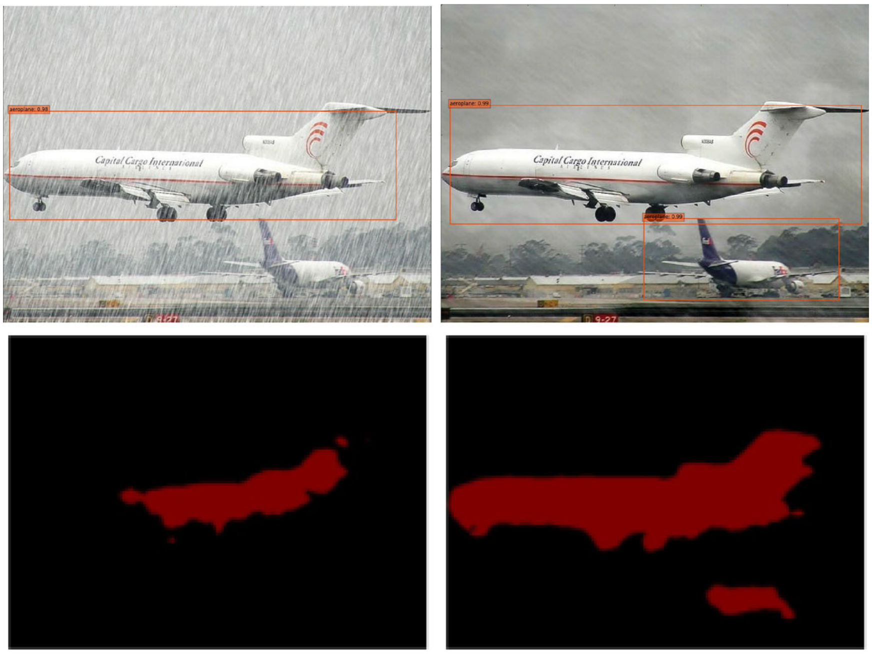
Moreover, we also provide several examples on real-world dataset compared with NLEDN [19] shown in Fig. 10. We can observe that our method is able to generate better and clearer deraining performance using only about 30,000 parameters, while NLEDN maintains a number of rain streaks and it has more than 1,000,000 parameters.

4.6 Ablation study

As our network consists of multi-stream dilation convolution, dense connections and multi-level shared pyramid, it is meaningful to discuss their effectiveness on image deraining. For simplicity, we use the following abbreviations for the baseline methods.

- **Single**: one-level pyramid network.
- **No dense**: our proposed network without dense connections.
- **No dilation**: our proposed network without dilation convolution.
- **Ours**: our proposed shared pyramid network that is of 3 levels.

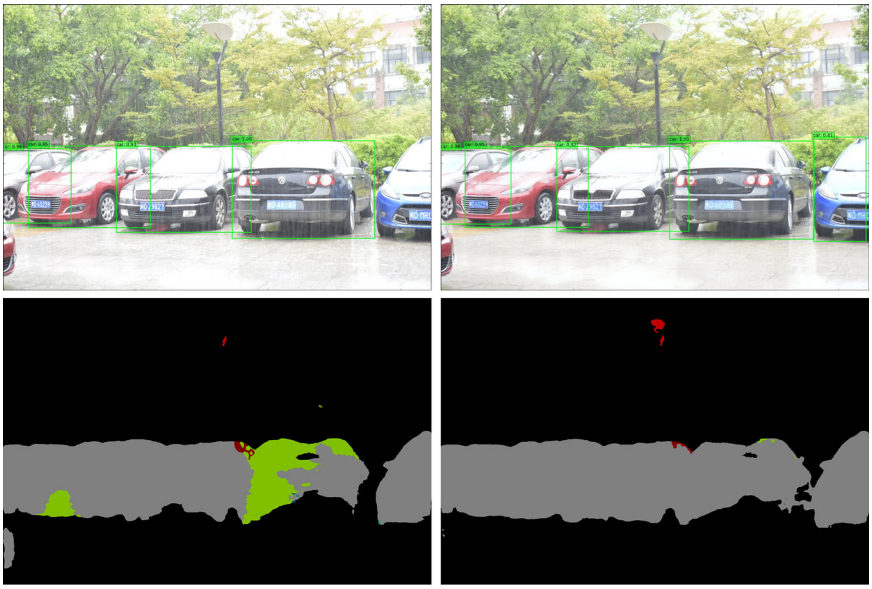
The results are shown in Table 3; compared with the one-level network, our sharing strategy improves the PSNR and SSIM by 0.31db and 1%, respectively, while the parameters have barely changed. This also demonstrates that our shared parameter strategy boosts the learning processing between different levels of the pyramid. So we believe that the shared parameter strategy is more worthy of promotion. In addition, it is observed the dense connections and dilation convolution also can be used to improve the performance. Specially, the dense connections greatly improve the expressive ability of the model. We also provide one example as visual comparison, shown in Fig. 11. Our proposed method obtains the highest PSNR and SSIM and has fewer artifacts compared with other methods.



(a) Rain

(b) De-rain

Fig. 12 Visual examples on object detection and semantic segmentation in synthetic dataset



(a) Rain

(b) De-rain

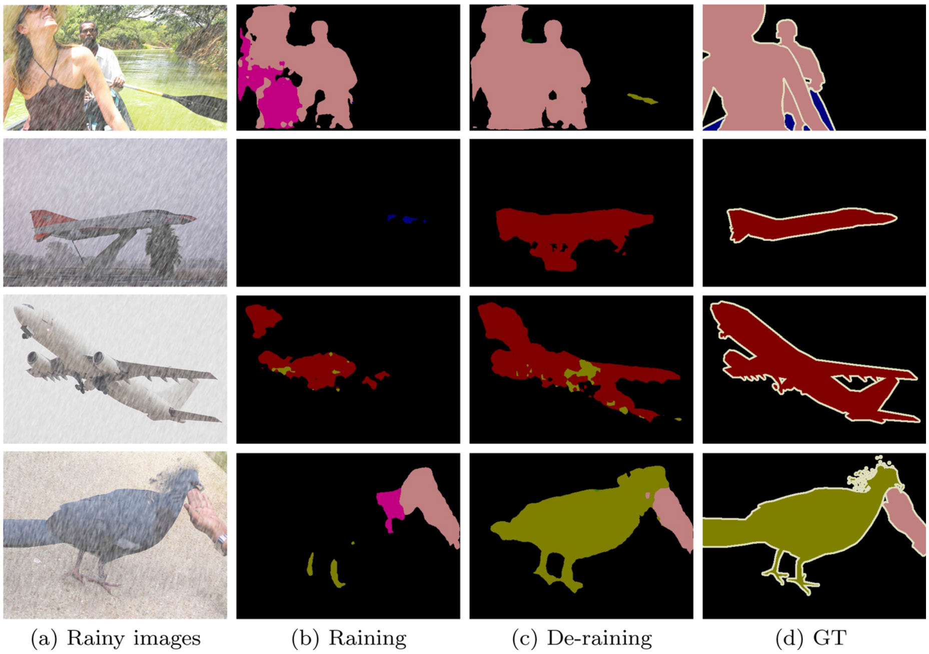
Fig. 13 Visual examples on object detection and semantic segmentation in real-world dataset

4.7 Effect of the pyramid levels and the numbers of dilation convolution

It is worth exploring the effect of different levels of the pyramid (K) and the numbers of dilation convolution of MSDC (D). Table 4 shows the results of different selection sets of the levels of the pyramid and the numbers of dilation convolution. It can be seen that the results are almost barely changed for the same number of dilation convolution. Moreover, there are obvious changes for the same levels of the pyramid with different numbers of dilation convolution. Although the parameters are less when $D = 2$, the results are unsatisfactory. While they have better performance when $D = 4$, the parameters are too many. We select $K = 3$ and $D = 3$ as our network set, because the parameters are less and the results are well-pleasing.

4.8 Analysis on the model size

In this section, we evaluate the effect of the model size. We show the results in Table 5. It can be observed that the performance of our method is improved with the channels and



(a) Rainy images (b) Raining (c) De-raining (d) GT

Fig. 14 Several preprocessing examples of semantic segmentation

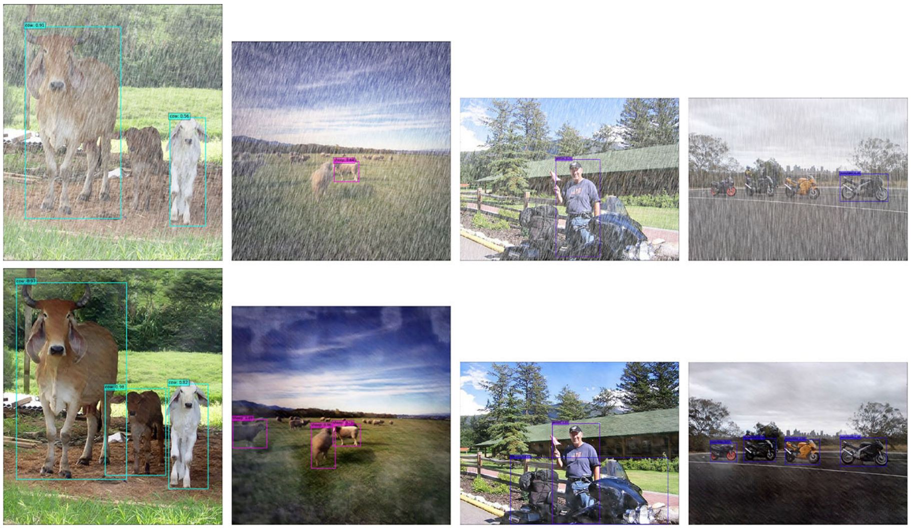


Fig. 15 Several preprocessing examples of object detection

the length of the network increasing, while the parameters are increased substantially. For lightweight network, i.e., $L = 10$ and $C = 6$, the result also is comparable with that of other state-of-the-art methods by combining with Table 1. Further, the results are far better than all state-of-the-art methods when the network has heavyweight parameters, i.e., $L = 14$ and $C = 10$. We select $L = 12$ and $C = 8$ as our network set; in this case, the network has fewer parameters and the results are satisfactory.

4.9 Applications on high-level vision tasks

Most CNN-based models for the high-level computer vision tasks, such as object detection and semantic segmentation, are trained in a good scenario. Therefore, rainy streaks will decrease the performance of these tasks. Figure 12a shows that under a rainy condition, SSD [23] fails to detect one of the planes, and FCN [28] cannot segment the planes. We created a Rainy Pascal VOC 2012 dataset, in which the images are synthesized to various rainy streaks with different sizes, directions and shapes. Incorporating our deraining method as a preprocess model for SSD and FCN, we conduct the experiments on this synthetic dataset, and the results are shown in Table 6. The mAP of SSD on the rainy validation set is 0.66, and after deraining, the mAP is improved to 0.74. And for FCN, the mIU is improved from 0.311 to 0.466 on the validation set. We also illustrate the results of detection and segmentation on synthetic and real-world images, respectively, in Figs. 12 and 13.

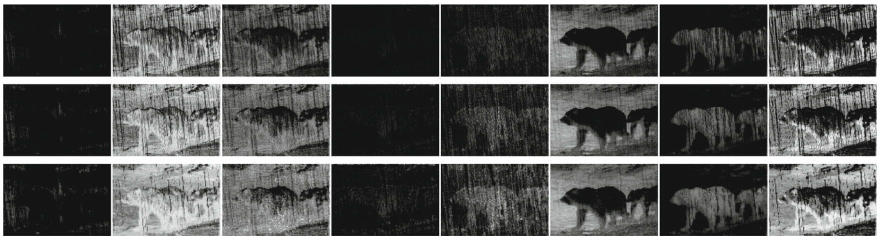


Fig. 16 In the shared parameter condition, the visualizations of feature maps on the $F_{1,0}$, $F_{2,0}$ and $F_{4,0}$ are from top to bottom. Our method is motivated that the features at different pyramid levels share similar structures. It can be observed that they indeed have similar structures at different pyramid levels

We also provide more examples in the applications of object detection and semantic segmentation in Figs. 14 and 15, respectively. It is obvious that these high-level tasks almost fail to work, while our method removes all of rain streaks and improves their accuracy with a large margin. So it is meaningful to develop a better deraining algorithm.

5 Visualization of feature maps

Our method is motivated by the fact that the features at different pyramid levels share similar structures. As shown in Fig. 16, it can be observed that the features at different pyramid levels indeed have the similar structures in the condition of shared parameters. This also supports our algorithm to work well and can boost the deraining by learning the features at different pyramid levels in the shared parameter condition.

6 Conclusions

In this paper, we have proposed a deep neural network based on a feature pyramid to solve image deraining. The proposed deep models at different feature pyramid levels share the same weight parameters. We further develop a multi-stream dilation convolution to deal with complex rainy streaks and propose dense connections to maintain the important features from different levels. By training in an end-to-end manner, the proposed method performs favorably against state-of-the-art deraining methods in terms of accuracy as well as model sizes.

Different loss functions may influence the deraining results and we will explore them in the future. Moreover, we only utilize simple rain model to solve the deraining tasks, while more complex rain models have been proposed and we will dig into more effective rain model and its principle to improve the deraining performance.

Funding This work was supported by the National Science and Technology Major Project [Grant Nos. 2018ZX04041001-007].

Compliance with ethical standards

Conflict of interest All authors declare that they have no conflict of interest.

References

1. Althoff, M., Stursberg, O., Buss, M.: Model-based probabilistic collision detection in autonomous driving. IEEE Trans. Intell. Transp. Syst. **10**(2), 299–310 (2009). https://doi.org/10.1109/TITS.2009.2018966
2. Chen, Y., Hsu, C.: A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In: ICCV, pp. 1968–1975 (2013). https://doi.org/10.1109/ICCV.2013.247
3. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. In: CVPR, pp. 7103–7112 (2018). 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
4. Cong, W., Wu, Y., Cai, Y., Yao, G., Su, Z., Wang, H.: Single image deraining via deep pyramid network with spatial contextual information aggregation. Appl. Intell. **50**, 1437–1447 (2020)
5. Cui, Z., Chang, H., Shan, S., Zhong, B., Chen, X.: Deep network cascade for image super-resolution. In: ECCV, pp. 49–64 (2014). https://doi.org/10.1007/978-3-319-10602-1_4
6. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell. **38**(2), 295–307 (2016). https://doi.org/10.1109/TPAMI.2015.2439281
7. Fan, X., Tang, X., Hou, M., Luo, Z.: Fast example searching for input-adaptive data-driven dehazing with gaussian process regression. Vis. Comput. **35**(4), 565–577 (2019). https://doi.org/10.1007/s00371-018-1485-y
8. Fu, X., Huang, J., Ding, X., Liao, Y., Paisley, J.: Clearing the skies: a deep network architecture for single-image rain removal. IEEE Trans. Image Process. **26**(6), 2944–2956 (2017). https://doi.org/10.1109/TIP.2017.2691802
9. Fu, X., Huang, J., Zeng, D., Huang, Y., Ding, X., Paisley, J.: Removing rain from single images via a deep detail network. In: CVPR, pp. 1715–1723 (2017). https://doi.org/10.1109/CVPR.2017.186
10. Gonzalez-Garcia, A., van de Weijer, J., Bengio, Y.: Image-to-image translation for cross-domain disentanglement. In: NeurIPS, pp. 1294–1305 (2018). URL http://papers.nips.cc/paper/7404-image-to-image-translation-for-cross-domain-disentanglement
11. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR, pp. 7132–7141 (2018). 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
12. Huang, D., Kang, L., Yang, M., Lin, C., Wang, Y.F.: Context-aware single image rain removal. In: ICME, pp. 164–169 (2012). https://doi.org/10.1109/ICME.2012.92
13. Huynh-Thu, Q., Ghanbari, M.: Scope of validity of psnr in image/video quality assessment. Electron. Lett. **44**(13), 800–801 (2008)
14. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV, pp. 694–711 (2016). https://doi.org/10.1007/978-3-319-46475-6_43

15. Kang, L., Lin, C., Fu, Y.: Automatic single-image-based rain streaks removal via image decomposition. IEEE Trans. Image Process. **21**(4), 1742–1755 (2012). https://doi.org/10.1109/TIP.2011.2179057
16. Kim, J., Lee, C., Sim, J., Kim, C.: Single-image deraining using an adaptive nonlocal means filter. In: ICIP, pp. 914–917 (2013). https://doi.org/10.1109/ICIP.2013.6738189
17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015). URL http://arxiv.org/abs/1412.6980
18. Li, B., Peng, X., Wang, Z., Xu, J., Feng, D.: Aod-net: All-in-one dehazing network. In: ICCV, pp. 4780–4788 (2017). https://doi.org/10.1109/ICCV.2017.511
19. Li, G., He, X., Zhang, W., Chang, H., Dong, L., Lin, L.: Non-locally enhanced encoder-decoder network for single image deraining. In: ACM MM, pp. 1056–1064 (2018). https://doi.org/10.1145/3240508.3240636
20. Li, X., Wu, J., Lin, Z., Liu, H., Zha, H.: Recurrent squeeze-and-excitation context aggregation net for single image deraining. In: ECCV, pp. 262–277 (2018). https://doi.org/10.1007/978-3-030-01234-2_16
21. Li, Y., Tan, R.T., Guo, X., Lu, J., Brown, M.S.: Rain streak removal using layer priors. In: CVPR, pp. 2736–2744 (2016). https://doi.org/10.1109/CVPR.2016.299
22. Lin, T., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: CVPR, pp. 936–944 (2017). https://doi.org/10.1109/CVPR.2017.106
23. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C., Berg, A.C.: SSD: single shot multibox detector. In: ECCV, pp. 21–37 (2016). https://doi.org/10.1007/978-3-319-46448-0_2
24. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR, pp. 3431–3440 (2015). https://doi.org/10.1109/CVPR.2015.7298965
25. Luo, Y., Xu, Y., Ji, H.: Removing rain from a single image via discriminative sparse coding. In: ICCV, pp. 3397–3405 (2015). https://doi.org/10.1109/ICCV.2015.388
26. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. In: CVPR, pp. 2720–2729 (2017). https://doi.org/10.1109/CVPR.2017.291
27. Ren, W., Liu, S., Zhang, H., Pan, J., Cao, X., Yang, M.: Single image dehazing via multi-scale convolutional neural networks. In: ECCV, pp. 154–169 (2016). https://doi.org/10.1007/978-3-319-46475-6_10
28. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(4), 640–651 (2017). https://doi.org/10.1109/TPAMI.2016.2572683
29. Wang, B., Chen, S., Wang, J., Hu, X.: Residual feature pyramid networks for salient object detection. Vis. Comput. (2019). https://doi.org/10.1007/s00371-019-01779-3
30. Wang, X., Shrivastava, A., Gupta, A.: A-fast-rcnn: Hard positive generation via adversary for object detection. In: CVPR, pp. 3039–3048 (2017). https://doi.org/10.1109/CVPR.2017.324
31. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004). https://doi.org/10.1109/TIP.2003.819861
32. Yang, W., Tan, R.T., Feng, J., Liu, J., Guo, Z., Yan, S.: Deep joint rain detection and removal from a single image. In: CVPR, pp. 1685–1694 (2017). https://doi.org/10.1109/CVPR.2017.183
33. Yuan, Q., Li, J., Zhang, L., Wu, Z., Liu, G.: Blind motion deblurring with cycle generative adversarial networks. Vis. Comput. (2019). https://doi.org/10.1007/s00371-019-01762-y
34. Zhang, H., Patel, V.M.: Densely connected pyramid dehazing network. In: CVPR, pp. 3194–3203 (2018). 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
35. Zhang, H., Patel, V.M.: Density-aware single image de-raining using a multi-stream dense network. In: CVPR, pp. 695–704 (2018). 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
36. Zhang, H., Sindagi, V., Patel, V.M.: Image de-raining using a conditional generative adversarial network. IEEE TCSVT (2019)
37. Zhang, S., He, F.: Drcdn: learning deep residual convolutional dehazing networks. Vis. Comput. (2019). https://doi.org/10.1007/s00371-019-01774-8
38. Zhang, S., He, F., Ren, W., Yao, J.: Joint learning of image detail and transmission map for single image dehazing. Vis. Comput. **36**(2), 305–316 (2020). https://doi.org/10.1007/s00371-018-1612-9
39. Zhang, S., Ren, W., Yao, J.: Feed-net: Fully end-to-end dehazing. In: IEEE ICME, pp. 1–6 (2018). https://doi.org/10.1109/ICME.2018.8486435
40. Zhang, Y., Wang, L., Qi, J., Wang, D., Feng, M., Lu, H.: Structured siamese network for real-time visual tracking. In: ECCV, pp. 355–370 (2018). https://doi.org/10.1007/978-3-030-01240-3_22

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Cong Wang received the B.S. degree from Inner Mongolia University, China, in 2017. He is currently pursuing the Master's Degree in computational mathematics with the School of Mathematical Sciences, Dalian University of Technology, China, under the supervision of Prof. Zhixun Su. His current research interests include deep learning, computer vision and image processing.

Xiaoying Xing is a junior student from Tsinghua University, China. She is currently pursuing the Bachelor's Degree in the department of automation. Her current research interests include machine learning, computer vision and image processing.

Guangle Yao received the Ph.D. degree from the University of Electronic Science and Technology of China. During his Ph.D. study, he was with the Institute of Optics and Electronics, Chinese Academy of Sciences. He is currently with the Chengdu University of Technology. His research interests include action recognition, object detection, visual tracking and deep learning.

Zhixun Su received the Ph.D. degree in computational mathematics from the Dalian University of Technology, China, in 1993, where he is currently a Professor. He is also a Visiting Professor with the Guilin University of Electronic Technology. His current research interests include computer graphics, computer vision, computational geometry and machine learning. He is currently a member of the Standing Committee of China Society of Computational Mathematics.