DUAL RECURSIVE NETWORK FOR FAST IMAGE DERAINING

Linwen Cai[1], *Si-Yao Li*[3], *Dongwei Ren**[2], *Ping Wang*[1]

[1]School of Mathematics, Tianjin University, Tianjin, China
[2]College of Computing and Intelligence, Tianjin University, Tianjin, China
[3]SenseTime Research, Beijing, China

ABSTRACT

Recent years have witnessed the great progress on deep image deraining networks. On one hand, deraining performance has been significantly improved by designing complex network architectures, yielding high computational cost. On the other hand, several lightweight networks try to improve computational efficiency, but at the cost of notable degrading deraining performance. In this paper, we propose a dual recursive network (DRN) for fast image deraining as well as comparable or superior deraining performance compared with state-of-the-art approaches. Specifically, our DRN only utilizes a 4-layer residual network (ResNet), which is recursively unfolded to remove rain streaks in multiple stages. Meanwhile, the intermediate 2-layer residual block (ResBlock) can be recursively computed, forming the dual recursive network. Experimental results show that DRN is very computationally efficient and can achieve favorable deraining results on both synthetic and real rainy images. The source codes and pre-trained models are available at https://github.com/csdwren/DRN.

***Index Terms*—** Image deraining, deep learning, recursive computation

1. INTRODUCTION

Rain is a common weather condition, and rain streaks can severaly damage the image quality when taking outdoor scenarios. Removing rain streaks not only enhances the visually perceptual quality of images, but also benefits various high level vision tasks [1, 2], such as object detection and image recognition. Thus, it has received considerable research attention to improve deraining performance in recent years [3, 4, 5, 2, 6, 7, 8]. Meanwhile, when applying on mobile imaging devices, it is also a valuable research topic to reduce model size and enhance the computational efficiency [8, 2].

With the great progress of deep learning in image processing tasks[9, 10, 11, 12, 13], e.g., image denoising and superresolution, deep learning-based image deraining methods have also been developed and achieved better performance than conventional optimization-based deraining methods [14, 15, 1, 3]. Given a rainy image $\mathbf{y}$, it is an ill-posed problem to seperate clean background image $\mathbf{x}$ and rain streak layer $\mathbf{r}$. Instead of designing degradation models and regularizations, deep learning-based image deraining methods try to learn the composition patterns of clean background images and rain layers from sufficient training data.

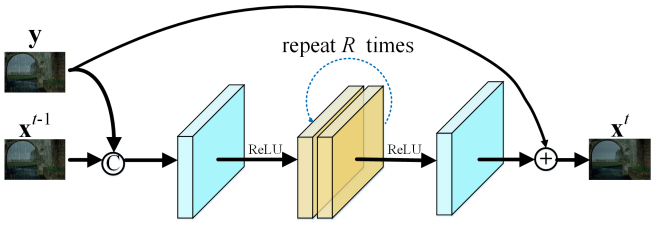

Fig. 1. Dual Recursive Network at stage t. DRN only includes 4 convolutional layers: 1 convolutional layer to receive the 6-channel concatenation of $\mathbf{x}^{t-1}$ and $\mathbf{y}$, and output 16 channel features; 2 convolutional layers in ResBlock have 16 channels; 1 convolutional layer receives 16 channel features and outputs 3 channel image that is then added to $\mathbf{y}$ to generate deraining result. The ResBlock is repeated R times, and the overall network is recursively unfolded T times.

On one hand, more research attentions have been paid on designing network architectures for better deraining quality. At first, Fu et al. [4] propose to decompose a rainy image into a base structure layer and a high frequency detail layer, and then utilize a 3-layer convolutional neural network (CNN) to extract rain streaks from the high-frequency detail layer. Subsequently, Fu et al. [5] employ a deep residual network (ResNet) [16] to substitute CNN for better extracting rain streaks from detail layers. In [6], Yang et al. design a complex CNN architecture to jointly detect and remove rain streaks, in which dilation filters [17] are utilized to take advantage of larger receptive field. In [7], squeeze-and-excitation blocks and dilation CNN are incoporated into recurrent network to remove rain streaks gradually. Recently, rain densities are jointly estimated to facilitate removing rain streaks [18] . The deraining performance is signifantly improved by these networks, but are facing higher computational cost due to complex network architectures with more network parameters.

* Corresponding author: Dongwei Ren (rendongweihit@gmail.com)

On the other hand, to meet the requirements of mobile devices, there are two lightweight deraining networks trying to reduce the computational cost. In [8], the authours propose a residual guidance network with dense connection to reduce network size. In [2], a Laplacian pyramid framework is utilized, in which deraining is tackled in multiple image resolutions, and the network size is further reduced. However, these lightweight deraining networks are at the cost of notable deraining performance degradation than state-of-the-art methods [6].

In this paper, we propose a dual recursive network (DRN) for fast image deraining, which can achieve superior or comparable deraining performance compared with state-of-the-art methods but significantly reduce network parameters. Specifically, the proposed DRN utilizes a ResNet with only 4 convolutional layers. The ResNet can be recursively unfolded to gradually remove rain streaks in multiple stages. To further strengthen the model capability, recursive computation is also employed in each stage. In particular, the intermediate 2-layer residual block (ResBlock) can be repeately unfolded several times. The network parameters of DRN only comes from 4-layer CNN, which is significantly less than state-of-the-art deraining networks [5, 6, 7], and is comparable with lightweight networks [8, 2].

The experimental results on synthetic datasets validate that our DRN can achieve superior and comparable deraining results than state-of-the-art methods [5, 6], and performs much better than lightweight networks [8, 2]. Our pre-trained DRN can also be well generallized to real-world rainy images to generate visually favorable deraining results.

2. THE PROPOSED METHOD

2.1. Motivation

With the powerful modeling capacity of deep CNN, image deraining performance has been significantly improved by deeper and complex networks. And to further enhance the deraining ability, there are some attempts to recurrently reuse deraining networks, e.g., JORDER [6], in which deraining results are recursively fed to deraining network to progressively remove rain streaks. However, the base deraining network is with complex architecture and more parameters. Motivated by the recursive unfolding strategy, we in this paper propose to repeatly unfold a simple ResNet to reduce network parameters. Furthremore, we propose to further take advantage of recursive computation in each stage, i.e., recursively computing the intermediate ResBlock several times, forming the Dual Recursive Network (DRN).

The proposed DRN has 3 advantages: (i) The network parameters only come from 4 convolutional layers, and thus DRN has very smaller network size than state-of-the-art deraining networks [5, 6, 7]. (ii) The recursive computation of simple ResNet is also very computationally efficient. (ii) DRN performs much better than the existing lightweight deraining networks [8, 2], and is comparable or superior than state-of-the art deraining networks [6, 5].

2.2. Network architecture

As shown in Fig. 1, our DRN only consists of 4 convolutional layers, i.e., 1 convolutional layer f_{in} to receive input, 2-layer ResBlock with recursive computation $f_{\text{recursive}}$ to extract features and 1 convolutional layer f_{out} to output clean images. Following [8, 5, 6], we also employ the residual learning. Thus the inference of DRN at stage t can be formulated as

$$\mathbf{x}^t = \mathbf{y} + f_{\text{out}}(f_{\text{recursive}}(f_{\text{in}}(\mathbf{x}^{t-1}, \mathbf{y}))). \tag{1}$$

The maximum stage number is T. All the convolutional layers have 3×3 kernel with 1×1 padding. In the following, we give the implementation details of each layer.

Input layer: Given a 3-channel RGB rainy image $\mathbf{y}$, the network takes the concatenation of 3-channel RGB rainy image $\mathbf{y}$ and 3-channel RGB deraining result $\mathbf{x}^{t-1}$ as input. Especially, $\mathbf{x}^0$ is set as $\mathbf{y}$. The convolution layer outputs 16-channel features, and is followed by ReLU.

Recursive ResBlock: Then, the features are fed to a 2-layer ResBlock. Instead of one forward pass, the ResBlock can be recursively computed R times. Following [16], the ResBlock has 2 convolutional layers followed by ReLU. And to make the dimension consistent in recursive computation, both the input and output of ResBlock have 16 channels.

Output layer: The output layer only contains one convolutional layer without activation. It takes 16 channel features from ResBlock as input, and outputs 3-channel image, which is added to $\mathbf{y}$ to generate the final deraining result.

2.3. Training

For DRN with T stages, we have T outputs, i.e., $\mathbf{x}^1$, $\mathbf{x}^2$,..., $\mathbf{x}^T$. Due to the intra-stage recursive computation, it is hard to train DRN by only imposing the supervision on the final output $\mathbf{x}^T$. Therefore, we propose to impose the supervision loss function on output of each stage, i.e.,

$$\mathcal{L} = \Sigma_{t=1}^{T} \lambda_t \ell\left(\mathbf{x}^t, \mathbf{x}^{GT}\right), \tag{2}$$

where λ_t is the positive trade-off parameter, $\mathbf{x}^{GT}$ is the ground-truth background image, and $\ell\left(\mathbf{x}^t, \mathbf{x}^{GT}\right)$ measures the loss values between $\mathbf{x}^t$ and $\mathbf{x}^{GT}$.

As for the choice of loss function $\ell\left(\mathbf{x}^t, \mathbf{x}^{GT}\right)$, several hybrid loss combinations are suggested in image deraining methods [2, 8, 6, 7]. However, in this paper we suggest that the negative SSIM loss [19] function is sufficient to train DRN for image deraining, i.e.,

$$\ell\left(\mathbf{x}^t, \mathbf{x}^{GT}\right) = -\text{SSIM}\left(\mathbf{x}^t, \mathbf{x}^{GT}\right). \tag{3}$$

Table 1. Average PSNR/SSIM comparison on synthetic datasets, including Rain100H [6], Rain100L [6] and Rain12 [15]. Red and blue colors are used to indicate top 1st and 2nd rank, respectively.

	GMM[15]	DDN[5]	JORDER[6]	ResGuideNet[8]	LPNet[2]	DRN
Rain100H	15.05/0.425	21.92/0.764	26.54/0.835	25.25/0.841	23.73/0.810	26.99/0.861
Rain100L	28.66/0.865	32.16/0.936	36.61/0.974	33.16/0.963	34.26/0.950	35.74/0.970
Rain12	32.02/0.855	31.78/0.900	33.92/0.953	29.45/0.938	35.35/0.950	36.15/0.960
Parameters	——	57,369	369,792	37,065	7,548	10,595

3. EXPERIMENTAL RESULTS

In this section, the proposed DRN is evaluated on three synthetic datasets and real rainy images. In comparison experiments, DRN is implemented with $T = 7$ and $R = 7$. As for the trade-off parameters in loss function (2), we set $\lambda_1 = 0.1$, $\lambda_2 = 0.2$, $\lambda_3 = 0.3$, $\lambda_4 = 0.4$, $\lambda_5 = 0.5$, $\lambda_6 = 0.6$ and $\lambda_7 = 1.7$, since the latter outputs should be paid more strength. DRN is implemented using Pytorch and is trained on a PC equipped with two NVIDIA GTX 1080Ti GPUs. We train all the models using ADAM [20] with batch size fixed as 16. The learning rate is set as 1×10^{-3}, and when reaching 30, 50 and 80 epoches, the learning rate is decayed by multiplying 0.5. The training ends after 100 epoches.

3.1. Evaluation on synthetic datasets

The DRN network is evaluated on three datasets, i.e., Rain100H [6], Rain100L [6] and Rain12 [15], and is comparad with 5 competing deraining approaches, including conventional optimization-based method: GMM [15], two state-of-the-art deep CNN-based models: DDN [5] and JORDER [6], and two lightweight networks: LPNet [2] and ResGuideNet [8]. Following the experimental settings [8, 6, 2], DRN models for Rain100H and Rain100L are trained, respectively. The model on Rain100L is applied to process the images in Rain12.

Table 2. Comparison of running time (*sec*.)

Image size	DDN[5]	JORDER[6]	DRN
500×500	0.407	0.179	0.107
1024×1024	0.754	0.815	0.381

Table 1 reports the quantitative comparison metrics along with network parameters. Since the source codes of ResGuideNet [8] and LPNet [2] are not available, we directly copy their reported metrics. Meanwhile, the deraining results by JORDER [6] are provided, based on which we can compute average PSNR and SSIM, which are consistent with those reported in ResGuideNet [8] and LPNet [2], confirming these borrowed metrics of ResGuideNet and LPNet. From the average PSNR and SSIM, our DRN performs better than these lightweight networks, and is comparable with state-of-the-art network JORDER. We should note that our DRN can achieve higher quantative metrics on Rain12, although it is a little inferior to JORDER on Rain100L. Since the training dataset of Rain100L only has 200 rainy images, it is possible that JORDER is overfitted to this dataset, while our DRN has better generatilization ability. The visual quality by DRN is also favorable than the competing methods, as shown in Fig. 2.

In Table 2, we report the running time of these competing networks, whose souce code or testing code are released. The running time is recorded on a NVIDIA GTX 1080Ti GPU. Our DRN has much fewer parameters than DDN and JORDER, and is also more computationally efficient than these methods.

3.2. Evaluation on real rainy images

On real rainy images, DRN is compared with DDN [5] and JORDER [6]. As shown in Fig. 3, our DRN performs better in removing rain streaks on the first rainy image, while both the results by DDN and JORDER have visible rain streaks. On the second image, both DRN and JORDER can remove rain streaks clearly. But JORDER over-suppresses texture details (see the red and green close-ups). Our DRN can better preserve texture details as well as effectively removing rain streaks.

3.3. Effects of recursive numbers in DRN

There are many settings of T and R. We have conducted comprehensive ablation studies, and found that DRN models with $T = R$ perform better than other settings. Due to the limited space, we only discuss the selection of recursive numbers with $T = R$. As in Table 3, we train four DRN models for Rain100H [6] with $T = 5, 6, 7, 8$, respectively. It is resonable to see that both PSNR and SSIM increase along with the recursive numbers, and the improvment becomes marginal. From the visual quality in Fig. 4, all the DRN models can achieve favorable deraining results. Considering the performance improvement and computational cost, we suggest to adopt $\text{DRN}_{7,7}$ in practical applications.

Table 3. The effects of recursive numbers in DRN on Rain100H [6]

	$\text{DRN}_{5,5}$	$\text{DRN}_{6,6}$	$\text{DRN}_{7,7}$	$\text{DRN}_{8,8}$
PSNR	26.74	26.87	26.99	27.03
SSIM	0.859	0.861	0.861	0.864
Time (*sec*.)	0.044	0.062	0.079	0.103

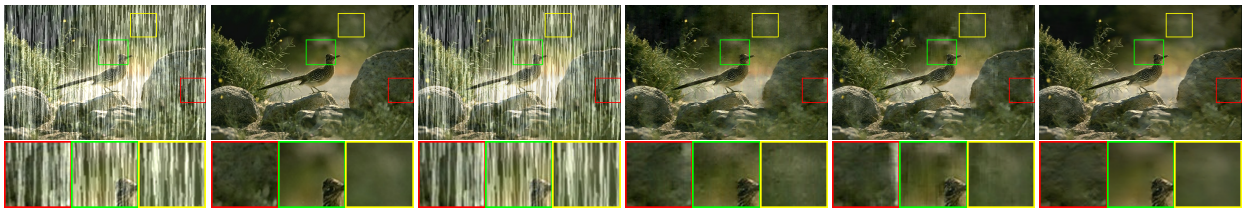

Fig. 2. Visual quality comparison on Rain100H dataset.

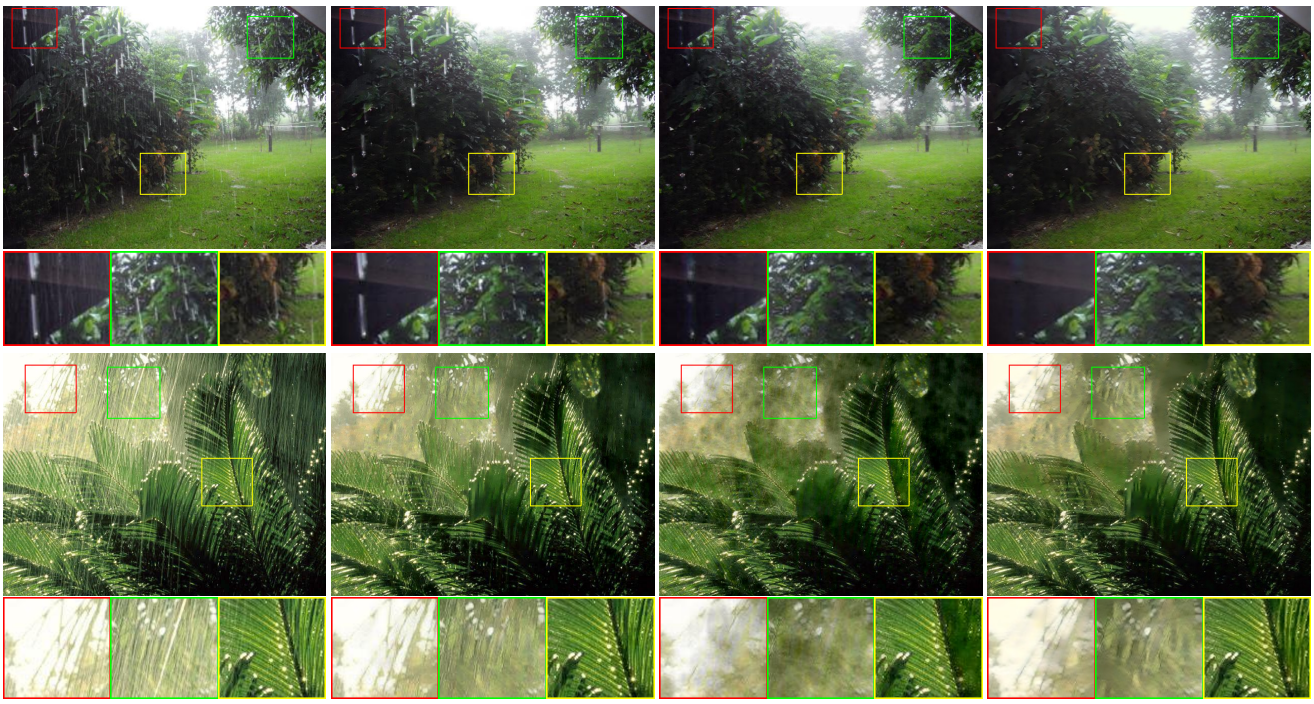

Fig. 3. Visual quality comparison on real world rainy images.

Fig. 4. Visual quality comparison of DRN with different recursive numbers.

4. CONCLUSION

In this paper, we proposed a dual recursive network for fast image deraining. A 4-layer ResNet is recursively unfolded, leading to small network size and satisfying deraining performance. Experimental results have demonstrated that DRN can achieve comparable or superior deraining results compared with state-of-the-art deraining networks, and performs much better than the existing lightweight deraining networks. The proposed dual recursive network can also be applied to other image restoration tasks for balancing the restoration qualtiy and computational cost.

5. ACKNOWLEDGEMENTS

This work was partially supported by the National Natural Science Foundation of China (Nos. 61801326, 91746107).

6. REFERENCES

[1] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1742, 2012. 1

[2] Xueyang Fu, Borong Liang, Yue Huang, Xinghao Ding, and John Paisley, "Lightweight pyramid networks for image deraining," *arXiv preprint arXiv:1805.06173*, 2018. 1, 2, 3

[3] Shuhang Gu, Deyu Meng, Wangmeng Zuo, and Lei Zhang, "Joint convolutional analysis and synthesis sparse representation for single image layer separation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1717–1725. 1

[4] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2944–2956, 2017. 1

[5] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley, "Removing rain from single images via a deep detail network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1715–1723. 1, 2, 3, 4

[6] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan, "Deep joint rain detection and removal from a single image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1357–1366. 1, 2, 3, 4

[7] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha, "Recurrent squeeze-and-excitation context aggregation net for single image deraining," in *European Conference on Computer Vision*. Springer, 2018, pp. 262–277. 1, 2

[8] Zhiwen Fan, Huafeng Wu, Xueyang Fu, Yue Hunag, and Xinghao Ding, "Residual-guide feature fusion network for single image deraining," in *ACM Multimedia*, 2018. 1, 2, 3

[9] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017. 1

[10] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016. 1

[11] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al., "Photo-realistic single image super-resolution using a generative adversarial network.," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1

[12] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1637–1645. 1

[13] Ying Tai, Jian Yang, and Xiaoming Liu, "Image super-resolution via deep recursive residual network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, vol. 1, p. 5. 1

[14] Yi-Lei Chen and Chiou-Ting Hsu, "A generalized low-rank appearance model for spatio-temporally correlated rain streaks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1968–1975. 1

[15] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown, "Rain streak removal using layer priors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2736–2744. 1, 3, 4

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. 1, 2

[17] Fisher Yu and Vladlen Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015. 1

[18] He Zhang and Vishal M Patel, "Density-aware single image de-raining using a multi-stream dense network," in *Proceedings of the IEEE International Conference on Computer Vision*, 2018. 1

[19] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004. 2

[20] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representation*, 2015. 3