Progressive Image Deraining Networks: A Better and Simpler Baseline
Supplementary Material

Dongwei Ren[1], Wangmeng Zuo[2], Qinghua Hu[1], Pengfei Zhu[1], and Deyu Meng[3]

[1]College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
[3]Xi'an Jiaotong University, Xi'an, China

In this supplementary file, we provide details of network architecture, ablation study on recursive ResBlocks and generalization evaluation, comparison on real rainy images and videos, more results on synthetic rainy images and more results on real-world rainy images.

1. Network Architecture

1.1. Architectures of PRN and PRN$_r$

PRN in Table s1 includes 1 convolution layer, 5 ResBlocks and 1 convolution layer, and in Table s2 PRN$_r$ has the similar architecture, but only 1 ResBlock is recursively unfolded 5 times.

Table s1. The architecture of PRN. Convolution is with the form Conv.(input channel, kernel size, padding size, stride, output channel).

Input: rainy image $\mathbf{y}$ ($m \times n \times 3$), initial $\mathbf{x}^0 = \mathbf{y}$		
for $t = 1$ to T		
	Concatenate($\mathbf{y}$, $\mathbf{x}^{t-1}$)	
	Layer 1	Conv.(6, 3, 1, 1, 32); ReLU;
	Layer 2	Conv.(32, 3, 1, 1, 32); ReLU;
	Layer 3	Conv.(32, 3, 1, 1, 32); ReLU;
		Add(*Layer1*, *Layer3*); ReLU;
	Layer 4	Conv.(32, 3, 1, 1, 32); ReLU;
	Layer 5	Conv.(32, 3, 1, 1, 32); ReLU;
		Add(*Layer3*, *Layer5*); ReLU;
	Layer 6	Conv.(32, 3, 1, 1, 32); ReLU;
	Layer 7	Conv.(32, 3, 1, 1, 32); ReLU;
		Add(*Layer5*, *Layer7*); ReLU;
	Layer 8	Conv.(32, 3, 1, 1, 32); ReLU;
	Layer 9	Conv.(32, 3, 1, 1, 32); ReLU;
		Add(*Layer7*, *Layer9*); ReLU;
	Layer 10	Conv.(32, 3, 1, 1, 32); ReLU;
	Layer 11	Conv.(32, 3, 1, 1, 32); ReLU;
		Add(*Layer9*, *Layer11*); ReLU;
	Layer 12	Conv.(32, 3, 1, 1, 3);
	Output: $\mathbf{x}^t$ ($m \times n \times 3$)	
end for		

Table s2. The architecture of PRN$_r$. Convolution is with the form Conv.(input channel, kernel size, padding size, stride, output channel).

Input: rainy image $\mathbf{y}$ ($m \times n \times 3$), initial $\mathbf{x}^0 = \mathbf{y}$	
for $t = 1$ to T	
Concatenate($\mathbf{y}$, $\mathbf{x}^{t-1}$)	
Layer 1	Conv.$(6, 3, 1, 1, 32)$; ReLU;
for $i = 1 : 5$	
Layer 2	Conv.$(32, 3, 1, 1, 32)$; ReLU;
Layer 3	Conv.$(32, 3, 1, 1, 32)$; ReLU;
	Add(*Layer1*, *Layer3*); ReLU;
end for	
Layer 4	Conv.$(32, 3, 1, 1, 3)$;
Output: $\mathbf{x}^t$ ($m \times n \times 3$)	
end for	

1.2. Architectures of PReNet and PReNet$_r$

The only difference of PReNet and PRN is the introduction of convolutional LSTM [5].

1.2.1 Convolutional LSTM

At stage t, LSTM receives both the features from ResBlocks $f_{\text{res}}(\mathbf{x}^{t-0.5})$ and recurrent states $\mathbf{s}^{t-1}$ from stage $t-1$. The LSTM includes an input gate $\mathbf{i}^t$, a forget gate $\mathbf{f}^t$, an output gate $\mathbf{o}^t$ and a cell state $\mathbf{c}^t$, and can be formally expressed as,

$$
\begin{aligned}
\mathbf{x}^t &= f_{\text{res}}(\mathbf{x}^{t-0.5}), \\
\mathbf{i}^t &= \sigma(\mathbf{W}_{ix} \otimes \mathbf{x}^t + \mathbf{W}_{is} \otimes \mathbf{s}^{t-1} + \mathbf{b}_i), \\
\mathbf{f}^t &= \sigma(\mathbf{W}_{fx} \otimes \mathbf{x}^t + \mathbf{W}_{fs} \otimes \mathbf{s}^{t-1} + \mathbf{b}_f), \\
\mathbf{o}^t &= \sigma(\mathbf{W}_{ox} \otimes \mathbf{x}^t + \mathbf{W}_{os} \otimes \mathbf{s}^{t-1} + \mathbf{b}_o), \\
\mathbf{g}^t &= tanh(\mathbf{W}_{cx} \otimes \mathbf{x}^t + \mathbf{W}_{cs} \otimes \mathbf{s}^{t-1} + \mathbf{b}_c), \\
\mathbf{c}^t &= \mathbf{f}^t \odot \mathbf{c}^{t-1} + \mathbf{i}^t \odot \mathbf{g}^t, \\
\mathbf{s}^t &= \mathbf{o}^t \odot tanh(\mathbf{c}^t),
\end{aligned}
\tag{1}
$$

where $\otimes$ is 2D convolution, $\odot$ is entry-wise product, σ is *sigmoid* function. All the convolutions in LSTM have 32 input channels and 32 output channels, and the kernel size is 3×3 along with 1×1 padding.

1.2.2 Architecture details

Table s3. The architecture of PReNet and PReNet$_r$. In f_{in}, f_{res} and f_{out}, PReNet and PReNet$_r$ share the same settings with PRN and PRN$_r$, respectively. The only difference is convolutional LSTM $f_{\text{recurrent}}$ Eqn. (1).

Input: rainy image $\mathbf{y}$ ($m \times n \times 3$), initial $\mathbf{x}^0 = \mathbf{y}$	
for $t = 1$ to T	
Concatenate($\mathbf{y}$, $\mathbf{x}^{t-1}$)	
Layer f_{in}	Conv.$(6, 3, 1, 1, 32)$; ReLU;
Layer $f_{recurrent}$	*Convolutional LSTM*;
Layer f_{res}	5 ResBlocks (**PReNet**) or recursive ResBlocks (**PReNet$_r$**) ;
Layer f_{out}	Conv.$(32, 3, 1, 1, 3)$;
Output: $\mathbf{x}^t$ ($m \times n \times 3$)	
end for	

2. Ablation Study

2.1. Effects of recursive ResBlocks

In Sec. 4.1.2 of the main manuscript, we have shown that by adopting intra-state recursive ResBlocks, the avarage PSNR and SSIM of PRN$_r$ and PReNet$_r$ are inferior to PRN and PReNet, respectively. But as shown in Fig. s1, the deraining results by PRN$_r$ and PReNet$_r$ are also visually plausible, and only seem to be a little brighter than those by PRN and PReNet. Considering the much smaller network sizes, we suggest to use PRN$_r$ and PReNet$_r$ in practical applications.

Rainy images | PRN | PReNet | PRN$_r$ | PReNet$_r$

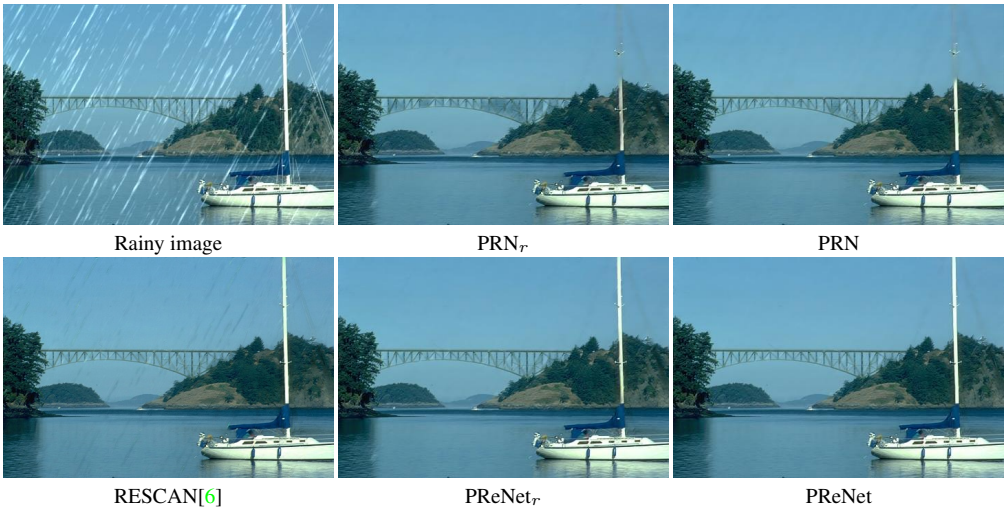
Figure s1. Effects of recursive ResBlocks. PRN and PReNet contain 5 ResBlocks. PRN$_r$ and PReNet$_r$ unfold 1 ResBlock 5 times.

2.2. Generalization Evaluation

To evaluate the generalization ability of PRN and PReNet, we use our models and RESCAN [3] trained for Rain100H [6] to process Rain100L [6] and Rain12 [4]. From Table s4, PReNet-H and PRN-H trained for Rain100H degrade obviously on Rain100L but generalize better on Rain12, partially due to that Rain12 contains heavy rain steaks. And our PReNet models perform better generalization ability than RESCAN [3]. The result of RESCAN has visible dark artifacts, while the results by our progressive networks are visually plausible, shown as in Fig. s2.

Table s4. Results on Rain100L and Rain12 by using Rain100H for training.

Model	RESCAN	PRN	PReNet	PRN$_r$	PReNet$_r$
Rain100L	32.02/0.949	31.92/0.958	34.89/0.971	31.13/0.951	33.77/0.965
Rain12	30.57/0.896	34.35/0.965	36.15/0.969	33.59/0.961	35.51/0.967



Rainy image | PRN$_r$ | PRN

RESCAN[6] | PReNet$_r$ | PReNet

Figure s2. Generalization evaluation by applying the models trained for Rain100H to directly process rainy images in Rain100L.

3. Comparison with State-of-the-arts

Here, we compare PReNet with state-of-the-arts on real rainy images and real rainy videos.

3.1. Evaluation on Real Rainy Images

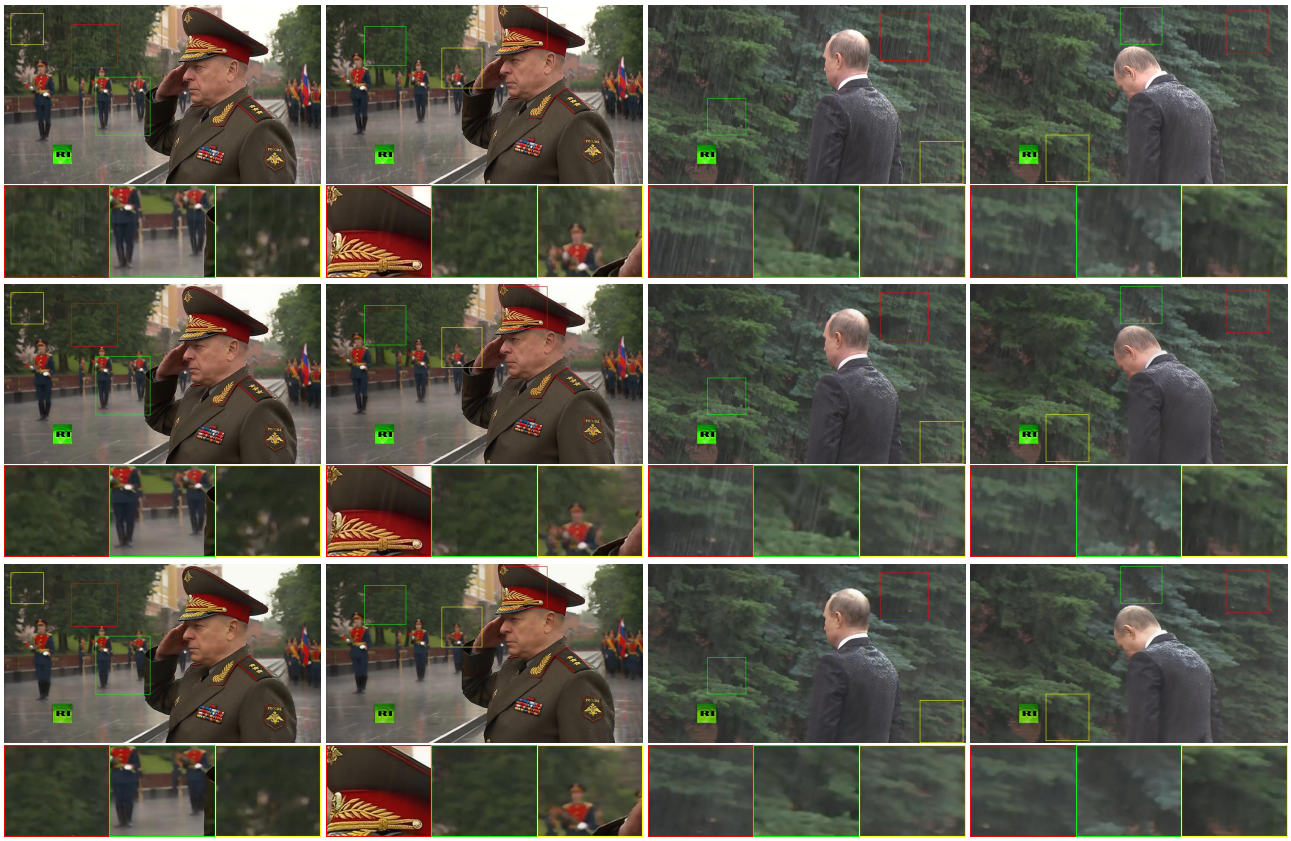
As shown in Fig. s3, PReNet is compared with GMM [4], DDN [1] and RESCAN [3]. On all the three images, PReNet can remove rain straks more clear and generate visually favorable deraining images.

Rainy images | GMM[4] | DDN[1] | RESCAN[6] | PReNet

Figure s3. Visual comparison on real rainy images.

3.2. Evaluation on Real Rainy Videos

Furthermore, PReNet is adopted to process a rainy video in a frame-by-frame manner, and is compared with state-of-the-art video deraining method, *i.e.*, FastDerain [2]. As shown in Fig. s4, for frame #510, both FastDerain and our PReNet can remove all the rain streaks, indicating the performance of PReNet even without the help of temporal consistency. However, FastDerain fails in switching frames, since it is developed by exploiting the consistency of adjacent frames. As a result, for frame #571, #572 and #640, rain streaks are remained in the results by FastDerain, while our PReNet performs favorably and is not affected by switching frames and accumulation error. Also by exploiting the temporal information, one potential furture work is to extend progressive networks to video deraining.



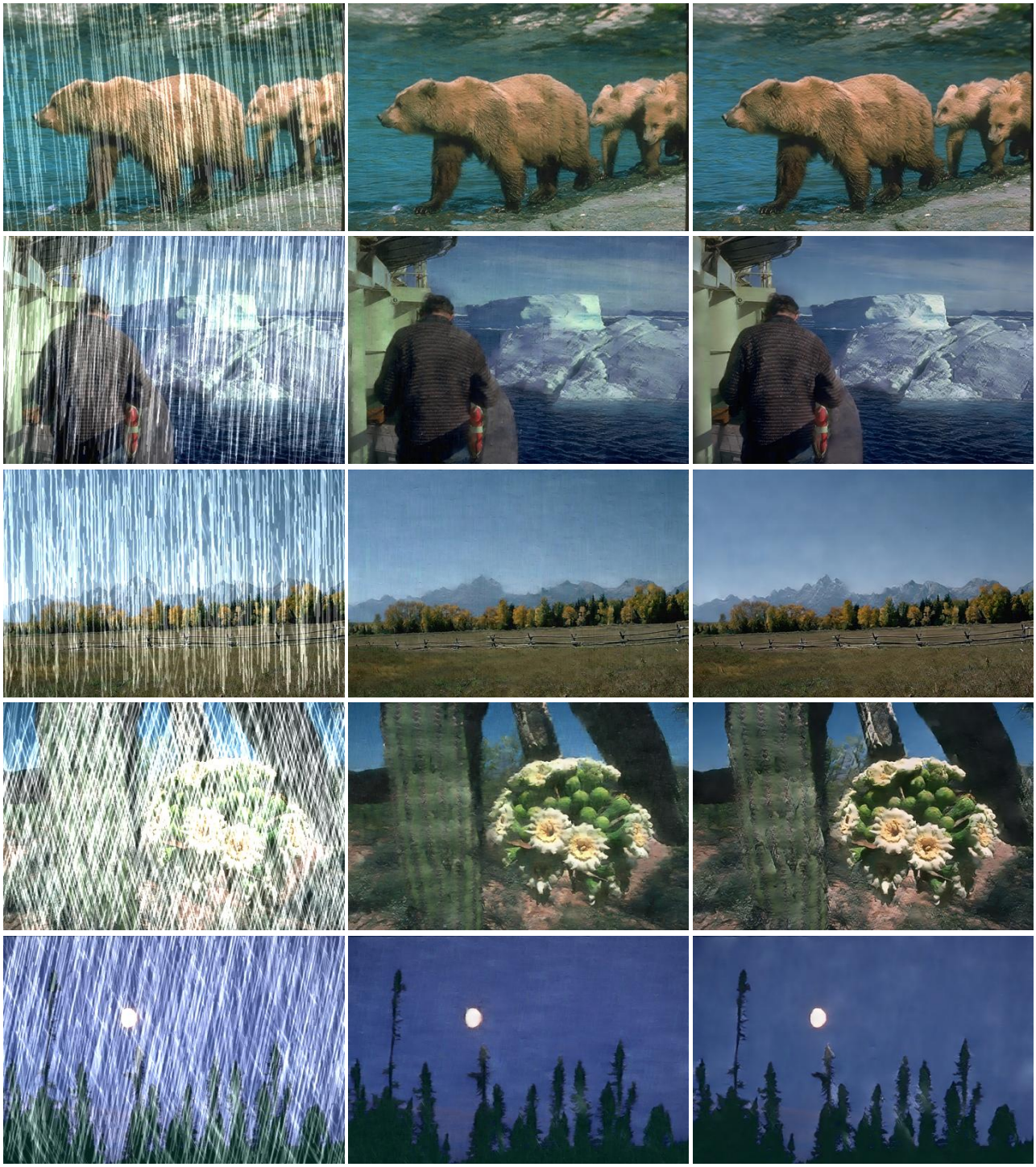
Frame #510 Frame #571 Frame #572 Frame #640

Figure s4. Visual quality comparison on a real rainy video. The first row is rainy frames, the second row is the results by FastDerain [2] and the third row is the results by PReNet.

4. More Results on Synthetic Rainy Images

4.1. More Results on Rain100H

The results by RESCAN [3] suffer from dark noises along rain direction, while our results are more visually pleasing.

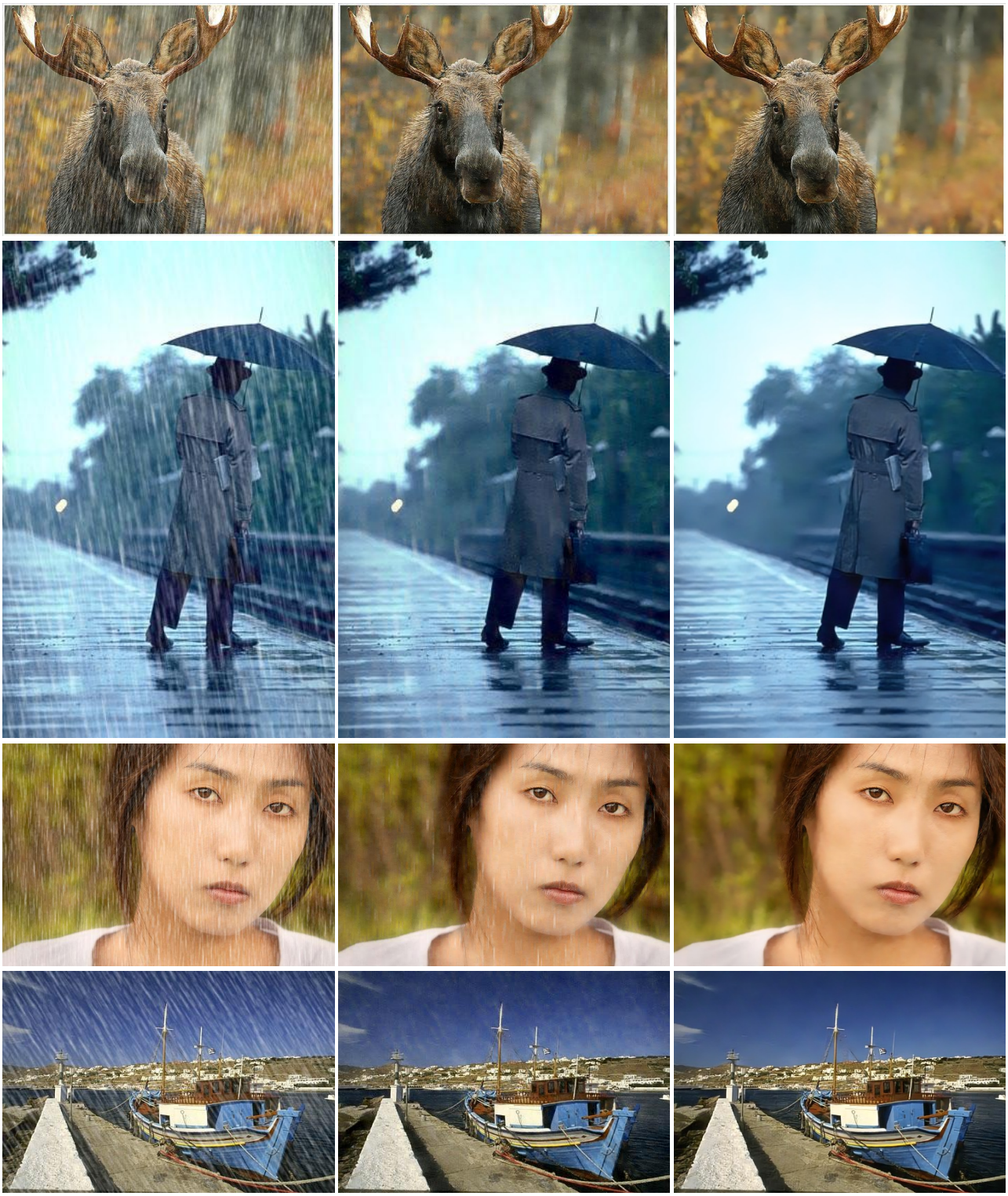


Rainy images RESCAN[3] PReNet

Figure s5. More results comparison on Rain100H.

4.2. More Results on Rain1400

The results by DDN [1] still have visible rain streaks.

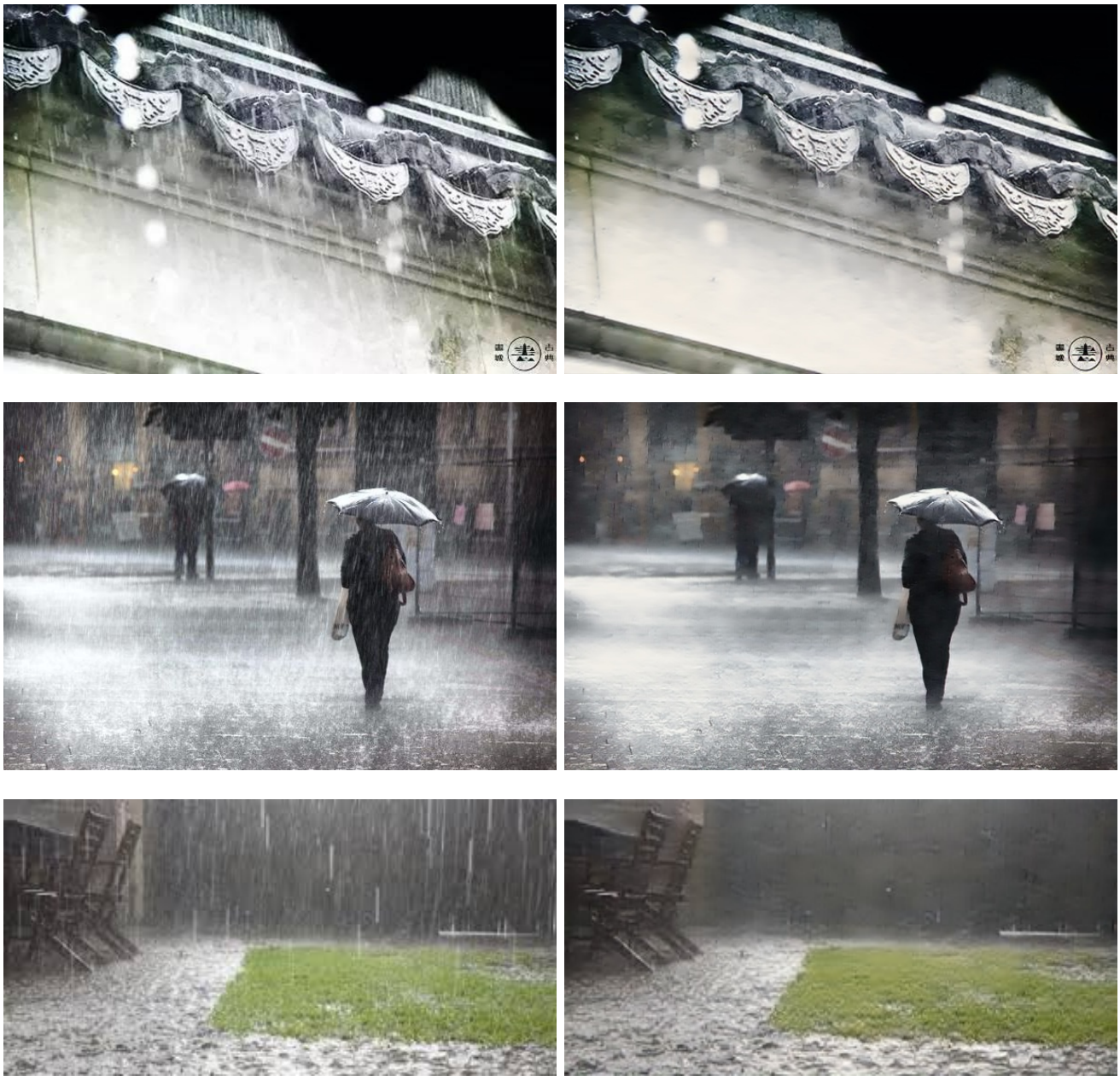


Rainy images DDN[1] PReNet

Figure s6. More results comparison on Rain1400.

5. More Results on Real Rainy Images

The left is rainy images, and the right is deraining results by our PReNet.




GLOBAL PHOTOGRAPHY
by Sherry Ott


GLOBAL PHOTOGRAPHY
by Sherry Ott


GLOBAL PHOTOGRAPHY






References

[1] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley. Removing rain from single images via a deep detail network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1715–1723, 2017. 4, 7

[2] T. Jiang, T. Huang, X. Zhao, L. Deng, and Y. Wang. A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 4, 5

[3] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *European Conference on Computer Vision*, pages 262–277. Springer, 2018. 3, 4, 6

[4] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown. Rain streak removal using layer priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2736–2744, 2016. 3, 4

[5] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015. 2

[6] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan. Deep joint rain detection and removal from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1357–1366, 2017. 3, 4