

BILATERAL RECURRENT NETWORK FOR SINGLE IMAGE DERAINING

Wei Shang, Pengfei Zhu, Dongwei Ren*, Hong Shi

¹College of Intelligence and Computing, Tianjin University, Tianjin, China

²Tianjin Key Lab of Machine Learning, China

ABSTRACT

Single image deraining has been widely studied in recent years. Motivated by residual learning, most deep learning based deraining approaches devote research attention to extracting rain streaks, usually yielding visual artifacts in final deraining images. To address this issue, we in this paper propose bilateral recurrent network (BRN) to simultaneously exploit rain streak layer and background image layer. Generally, we employ dual residual networks (ResNet) that are recursively unfolded to sequentially extract rain streaks and predict clean background image. Furthermore, we propose bilateral LSTMs into dual ResNets, which not only can respectively propagate deep features across multiple stages, but also bring the interplay between rain streak layer and background image layer. The experimental results demonstrate that our BRN notably outperforms state-of-the-art deep deraining networks on both synthetic datasets and real rainy images. All the source code and pre-trained models are available at <https://github.com/shangwei5/BRN>.

Index Terms— Image deraining, CNN, LSTM

1. INTRODUCTION

Removing rain streaks from a single image is a crucial task in computer vision systems, *e.g.*, surveillance, object detection and recognition in rainy outdoor scenes [1, 2]. Single image deraining is a very challenging ill-posed problem, and has received considerable research attention in recent years. Basically, image deraining can be regarded as a image decomposition problem, *i.e.*, a rainy image y should be decomposed into a rain streak layer r and a clean background image layer x . There are several conventional optimization based deraining methods [3–6] by studying the composition pattern of the rainy image and designing proper regularization priors.

With the great success of deep learning in low-level vision tasks [7–15], deep convolutional neural network (CNN)-based deraining methods [7, 16–18] also achieve significant performance improvements against conventional optimization based methods. Driven by the success of residual learning in low-level vision tasks [7, 16], most recent deraining networks only pay attention to studying rain streak layer. As

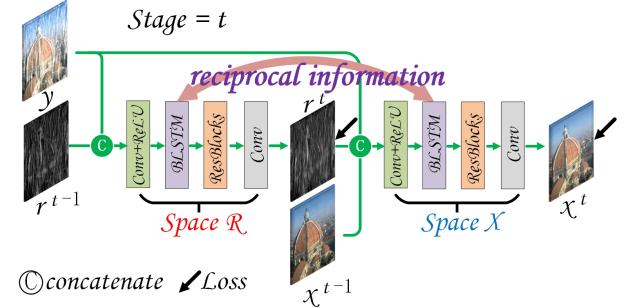


Fig. 1: Illustration of the proposed BRN. The deraining is tackled in T stages, where two recurrent ResNets are adopted to respectively extract rain streaks (\mathcal{F}_r) and generate clean background image (\mathcal{F}_x). The proposed BLSTMs not only propagate recurrent states across stages, but also bring the interplay between \mathcal{F}_x and \mathcal{F}_r , and its architecture is detailed in Fig. 2.

seminal deep deraining approaches, Fu *et al.* decompose a rainy image into a base layer and a detail layer, and utilize a 3-layer CNN [19] or a deeper residual network (ResNet) [20] to extract rain streaks from the detail layer. Yang *et al.* [17] propose a joint rain streak detection and removal framework using dilated convolution network. Moreover, various network architectures, *e.g.*, residual-guide fusion network [21], rain density aware multi-stream dense network [22] and squeeze-and-excitation context aggregation network [18] have been proposed to better extract rain streaks. Most recently, unsupervised learning [23] and semi-supervised learning [24] have also been studied for single image deraining. Despite of various network architectures and training strategies, these methods use deep network to learn a residual mapping from rainy image to rain streak layer.

By assuming the linear degradation model,

$$y = x + r, \quad (1)$$

the rain streaks r by residual mapping can be subtracted from rainy image y to predict clean background image x . However, for real world rainy images the composition of rain streak layers and background images become more complicated. Thus, rain streaks are very likely to be over-subtracted to yield visual artifacts (see Fig. 3), although these state-of-the-art deep deraining networks have achieved quantitative performance

*Corresponding author: rendongweihit@gmail.com

improvements.

To address this issue, we in this paper propose a bilateral recurrent network (BRN) to jointly exploit rain streak layer and clean background image layer. Specifically, we employ a ResNet (\mathcal{F}_r) to extract features of rain streak layer \mathbf{r} , which is then fed to another ResNet (\mathcal{F}_x) to predict clean background image \mathbf{x} , as shown in Fig. 1. By concatenating rainy image \mathbf{y} , current background image \mathbf{x} and rain streak layer \mathbf{r} from \mathcal{F}_r as the input, the composition pattern modeling is implicitly left to \mathcal{F}_x . Both \mathcal{F}_r and \mathcal{F}_x are recursively unfolded several times to benefit from progressive deraining in multiple stages [17, 18, 25]. Furthermore, we propose bilateral LSTMs (BLSTMs), which are incorporated into \mathcal{F}_r and \mathcal{F}_x , finally forming BRN. As shown in Figs. 1 and 2 for BLSTMs, not only deep features of rain streak layer and background image layer can be respectively propagated across multiple stages, but also the interplay between \mathcal{F}_r and \mathcal{F}_x would facilitate deraining performance. Extensive experiments have validated the superiority of BRN against state-of-the-art deep deraining networks [17, 18, 20] in generating visually favorable deraining images.

2. THE PROPOSED METHOD

Based on linear additive model, most recent deep deraining methods adopt neural network \mathcal{F} to extract rain streaks \mathbf{r} from rainy image \mathbf{y} , i.e., $\mathbf{r} = \mathcal{F}(\mathbf{y})$. Then, clean background image can be estimated using,

$$\mathbf{x} = \mathbf{y} - \mathcal{F}(\mathbf{y}). \quad (2)$$

To better extract rain streaks, various complicated architectures of \mathcal{F} have been proposed including residual-guide fusion network [21], rain density aware multi-stream dense network [22] and squeeze-and-excitation context aggregation network [18] etc. Albeit these state-of-the-art deraining networks have made significant improvements in terms of quantitative metrics, rain streaks are very likely to be over-subtracted, yielding dark artifacts along with the orientations of rain streaks in final deraining images (see Fig. 3).

One solution to this problem is to stack dual deep networks for rain streaks extraction and generating clean background image, respectively. And benefiting from the recursive computation to progressively remove rain streaks [17, 18], dual deep networks can be recursively unfolded T times. At stage t , it can be formulated as Eqn. (3) to sequentially extract rain streaks and generate clean background image.

$$\begin{aligned} \mathbf{r}^t &= \mathcal{F}_r(\mathbf{y}, \mathbf{r}^{t-1}), \\ \mathbf{x}^t &= \mathcal{F}_x(\mathbf{y}, \mathbf{r}^t, \mathbf{x}^{t-1}), \end{aligned} \quad (3)$$

where \mathcal{F}_r and \mathcal{F}_x are two coupled deep networks for extracting rain streaks and clean background image, respectively. Both \mathcal{F}_r and \mathcal{F}_x are specified as ResNets in this work, resulting in coupled ResNet (CRN) for image deraining. Motivated

by [18], two LSTMs [26, 27] can be individually incorporated into CRN to respectively propagate deep features of rain streak layer and clean background image layer across multiple stages. Moreover, we would like to take one step forward by asking a question: *Will it bring benefits by communicating between \mathcal{F}_r and \mathcal{F}_x ?* To answer this question, we first propose bilateral LSTMs.

2.1. Bilateral LSTMs

A standard LSTM [26, 27] consists of an input gate \mathbf{i} , a forget gate \mathbf{f} , an output gate \mathbf{o} and a hidden state \mathbf{h} . Besides propagating the hidden state \mathbf{h} across stages, respectively, we propose to bring the interplay between these two LSTMs, forming BLSTMs. As shown in Fig. 2, BLSTMs provide reciprocal information between \mathbf{r} and \mathbf{x} at every single recursion. In particular, the hidden state \mathbf{h}_r^t in \mathcal{F}_r is not only propagated across stages to facilitate rain streaks extraction, but also is fed to the BLSTM in \mathcal{F}_x , and vice versa.

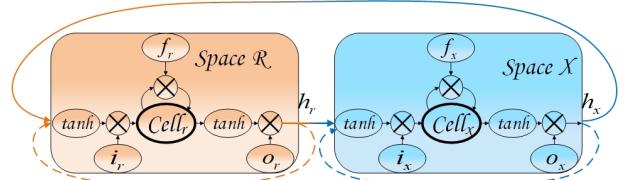


Fig. 2: The architecture of BLSTMs, where the hidden state \mathbf{h} is not only respectively propagated through space \mathcal{X} and space \mathcal{R} (dashed lines), but also brings the interplay between space \mathcal{R} and space \mathcal{X} (solid lines).

Taking BLSTM in \mathcal{F}_r as an example, BLSTM at stage t receives the features \mathbf{z}_r^t from the input layer $f_r(\cdot)$, recurrent state \mathbf{h}_r^{t-1} of rain streak layer and recurrent state \mathbf{h}_x^{t-1} of clean background image layer as the input. And then BLSTM in \mathcal{F}_r can be formally expressed as,

$$\begin{aligned} \mathbf{z}_r^t &= f_r(\mathbf{y}, \mathbf{r}^{t-1}), \\ \mathbf{i}_r^t &= \sigma(\mathbf{W}_{riz} \otimes \mathbf{z}_r^t + \mathbf{W}_{rih_x} \otimes \mathbf{h}_x^{t-1} + \mathbf{W}_{rih_r} \otimes \mathbf{h}_r^{t-1} + \mathbf{b}_{ri}), \\ \mathbf{f}_r^t &= \sigma(\mathbf{W}_{rfz} \otimes \mathbf{z}_r^t + \mathbf{W}_{rfh_x} \otimes \mathbf{h}_x^{t-1} + \mathbf{W}_{rfh_r} \otimes \mathbf{h}_r^{t-1} + \mathbf{b}_{rf}), \\ \mathbf{o}_r^t &= \sigma(\mathbf{W}_{roz} \otimes \mathbf{z}_r^t + \mathbf{W}_{roh_x} \otimes \mathbf{h}_x^{t-1} + \mathbf{W}_{roh_r} \otimes \mathbf{h}_r^{t-1} + \mathbf{b}_{ro}), \\ \mathbf{g}_r^t &= \tanh(\mathbf{W}_{rgz} \otimes \mathbf{z}_r^t + \mathbf{W}_{rgh_x} \otimes \mathbf{h}_x^{t-1} + \mathbf{W}_{rgh_r} \otimes \mathbf{h}_r^{t-1} + \mathbf{b}_{rg}), \\ \mathbf{c}_r^t &= \mathbf{f}_r^t \odot \mathbf{c}_r^{t-1} + \mathbf{i}_r^t \odot \mathbf{g}_r^t, \\ \mathbf{h}_r^t &= \mathbf{o}_r^t \odot \tanh(\mathbf{c}_r^t), \end{aligned} \quad (4)$$

where \otimes is 2D convolution, \odot is entry-wise product, σ is sigmoid function, \mathbf{W} and \mathbf{b} are corresponding convolutional matrix and bias vector. Due to limited space, we cannot provide details of BLSTM in \mathcal{F}_x , whose architecture is almost same with that in \mathcal{F}_r by exchanging subscripts r and x . And the minor distinctions are two fold: (i) $\mathbf{z}_x^t = f_x(\mathbf{y}, \mathbf{x}^{t-1}, \mathbf{r}^t)$ is used as input, and (ii) recurrent state \mathbf{h}_x^t instead of \mathbf{h}_r^{t-1} is taken into account.

Table 1: Average PSNR and SSIM comparison on synthetic datasets, including Rain100H [17], Rain100L [17] and Rain12 [4]. Red, blue and purple colors are used to indicate top 1st, 2nd and 3rd rank, respectively.

Method	GMM [4]	DDN [20]	ResGuideNet [21]	JORDER [17]	RESCAN [18]	CRN	BRN _{x→r}	BRN _{r→x}	BRN
Rain100H	15.05/0.425	21.92/0.764	25.25/0.841	26.54/0.835	28.64/0.864	29.10/0.897	29.50/0.901	29.16/0.898	29.58/0.902
Rain100L	28.66/0.865	32.16/0.936	33.16/0.963	36.61/0.974	—	37.52/0.980	37.65/0.980	37.40/0.979	37.82/0.981
Rain12	32.02/0.855	31.78/0.900	29.45/0.938	33.92/0.953	—	36.58/0.959	36.63/0.959	36.54/0.959	36.70/0.959

2.2. Bilateral Recurrent Network

Generally, we incorporate BLSTMs into CRN to answer whether the interplay between \mathcal{F}_x and \mathcal{F}_r would benefit extracting rain streaks and generating visually favorable deraining images.

2.2.1. Network Architecture

As shown in Fig. 1, CRN with \mathcal{F}_r and \mathcal{F}_x are recursively unfolded T times. BLSTMs are incorporated into \mathcal{F}_r and \mathcal{F}_x to respectively propagate deep features of \mathcal{F}_r and \mathcal{F}_x across stages and the interplay between \mathcal{F}_r and \mathcal{F}_x , forming BRN. At stage t , BRN can be formulated as,

$$\begin{aligned} \mathbf{r}^t &= \mathcal{F}_r(\mathbf{y}, \mathbf{r}^{t-1}, (\mathbf{h}_r^{t-1}, \mathbf{h}_x^{t-1})), \\ \mathbf{x}^t &= \mathcal{F}_x(\mathbf{y}, \mathbf{x}^{t-1}, \mathbf{r}^t, (\mathbf{h}_r^t, \mathbf{h}_x^{t-1})), \end{aligned} \quad (5)$$

where \mathbf{h}_r and \mathbf{h}_x are hidden states for rain streak layer and clean background image layer in BLSTMs, respectively. By simultaneously considering rain streak layer and background image layer, the composition pattern of rainy image can be implicitly learned by \mathcal{F}_x . In \mathcal{F}_x and \mathcal{F}_r , all convolutions are with 3×3 kernel size, 1×1 padding size and stride 1.

Implementation of \mathcal{F}_r : \mathcal{F}_r includes 1 input layer, BLSTMs, 3 ResBlocks and 1 output layer. Specifically, the input layer with 1 convolution takes 6-channel concatenation of RGB rainy image \mathbf{y} and current rain streak layer \mathbf{r}^{t-1} as input. 3 ResBlocks are used to extract deep features, and finally 1 convolutional layer is adopted to output 3-channel RGB deraining image. Except convolutions in input and output layers, all the other convolutions have 32 input channels and 32 output channels.

Implementation of \mathcal{F}_x : \mathcal{F}_x has the similar architecture as \mathcal{F}_r . The distinctions are two-fold: (i) 5 ResBlocks are adopted in \mathcal{F}_x , since background images are usually with richer structures and textures. (ii) The input channels of the input layer are 9, since it is the concatenation of RGB rainy image \mathbf{y} , background image \mathbf{x}^{t-1} and rain streaks \mathbf{r}^t .

2.2.2. Loss function

As for training BRN, we propose to impose recursive supervision at each stage. For BRN with T stages, we have T estimated background images and T extracted rain streak layers, i.e., $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^T$ and $\mathbf{r}^1, \mathbf{r}^2, \dots, \mathbf{r}^T$. And the recursive supervision is,

$$\mathcal{L}_x = \sum_t^T \lambda_t \ell(\mathbf{x}^t, \mathbf{x}^{gt}), \quad (6)$$

$$\mathcal{L}_r = \sum_{t=1}^T \lambda_t \ell(\mathbf{r}^t, \mathbf{r}^{gt}), \quad (7)$$

where $\ell(\cdot, \cdot)$ measures the difference between the output of stage t and the corresponding ground-truth, and λ_t is a trade-off parameter. As the final loss function, \mathcal{L}_r and \mathcal{L}_x should be jointly considered and balanced by hyper-parameters α and β ,

$$\mathcal{L} = \alpha * \mathcal{L}_r + \beta * \mathcal{L}_x. \quad (8)$$

As for the choice of loss function $\ell(\cdot, \cdot)$, there are several combinations in recent works, e.g., MSE+SSIM [21] and ℓ_1 +SSIM [2]. In this work, for an image \mathbf{a} and its groundtruth \mathbf{a}^{gt} , we only adopt negative SSIM loss [28],

$$\ell(\mathbf{a}, \mathbf{a}^{gt}) = -\text{SSIM}(\mathbf{a}, \mathbf{a}^{gt}). \quad (9)$$

3. EXPERIMENTAL RESULTS

In this section, BRN is evaluated on three benchmark datasets and real world rainy images. More results can be found at <https://github.com/shangwei5/BRN>. All the BRN models in this paper have $T = 4$ stages, and the trade-off parameters in the loss function are set as $\lambda_1 = \lambda_2 = \lambda_3 = 0.5$, $\lambda_4 = 1.5$, and $\alpha = 0.45$, $\beta = 0.55$. BRN is implemented using Pytorch, and the experiments are conducted on a PC with 4 NVIDIA TITAN Xp GPUs. As for the settings in training, the patch size is 100×100 , and the batch size is 12. The training is optimized using ADAM [29] algorithm with initial learning rate 1×10^{-3} , and ends with 100 epochs. When reaching 30, 50 and 80 epochs, the learning rate is multiplied by 0.2.

3.1. Evaluation on Synthetic Datasets

We evaluate BRN on three benchmark datasets, i.e., Rain100H [17], Rain100L [17] and Rain12 [4], and compare it with several state-of-the-art deraining methods including conventional optimization-based method: GMM [4] and state-of-the-art deep CNN-based methods: DDN [20], ResGuideNet [21], JORDER [17] and RESCAN [18].

For heavy rainy images (Rain100H) and light rainy images (Rain100L), the models are respectively trained, and the models for light rain are used to process Rain12. Since the source code of ResGuideNet is not available, we borrow the quantitative results from [21]. As for JORDER, we directly calculate average PSNR and SSIM on deraining results provided by the authors. We retrain RESCAN [18] for Rain100H with the default settings. It is worth noting that RESCAN and our BRN are trained on strict 1254 rainy

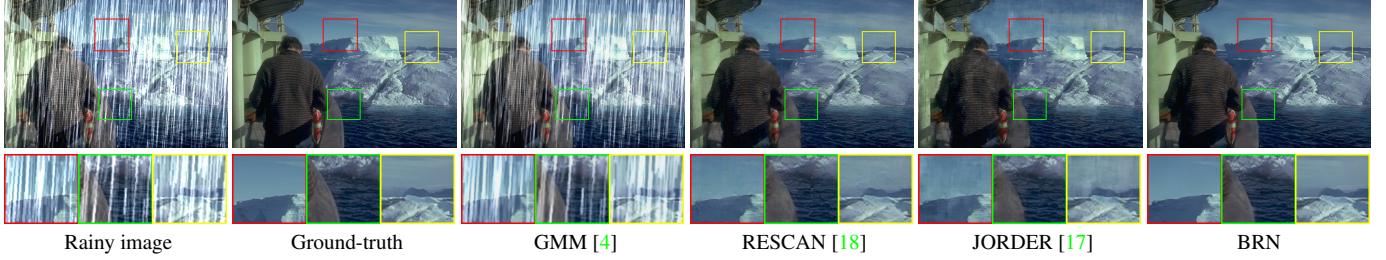


Fig. 3: Qualitative comparison on Rain100H dataset [17].

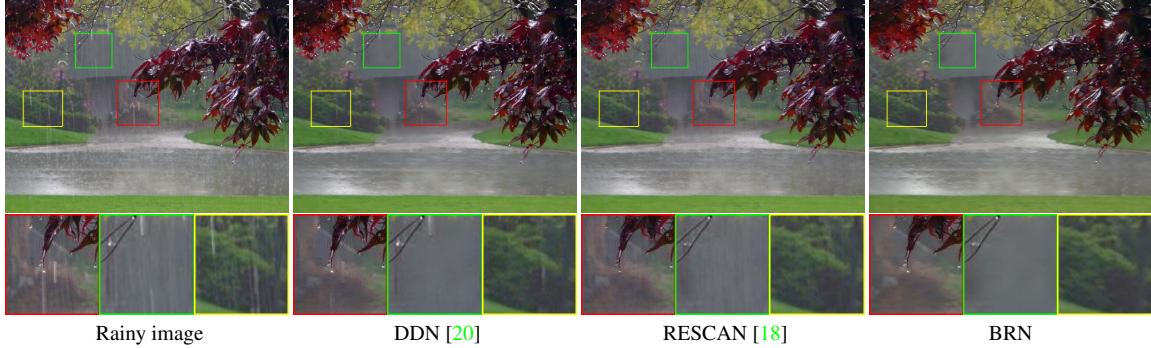


Fig. 4: Qualitative comparison on real world rainy images.

images for Rain100H, since among full 1800 training samples, there are 546 rainy images having the same background contents with those in the testing dataset. These methods are quantitatively evaluated using two popular metrics in low level vision tasks, *i.e.*, SSIM [28] and PSNR. The comparison results are reported in Table 1, from which we can see that our BRN is notably superior to existing methods. Especially for Rain100H with heavy rain streaks, the performance gain by our BRN is very significant. From the visual quality comparison in Fig. 3, our BRN can generate visually plausible deraining images, while the results generated by the other deraining approaches suffer from visible dark artifacts due to the over-subtracted rain streaks.

3.2. Evaluation on Real Rainy Images

BRN is further evaluated on real-world rainy images, and is compared with two state-of-the-art deep deraining networks, *i.e.*, DDN [20] and RESCAN [18]. As shown in Fig. 4, BRN performs better than DDN [20] and RESCAN [18] in removing rain streaks and generating visually favorable deraining images. The deraining results by both DDN and RESCAN have visible rain streaks, while our BRN succeed in removing rain streaks, and can better recover texture details in background images.

3.3. Effectiveness of BLSTMs

To validate the effectiveness of BLSTMs in BRN, we validate the performance of three variants of BRN, including CRN (with two individual LSTMs) and two monodirectional

BRN models, *i.e.*, $\text{BRN}_{r \rightarrow x}$ (communication from \mathcal{F}_r to \mathcal{F}_x) and $\text{BRN}_{x \rightarrow r}$ (communication from \mathcal{F}_x to \mathcal{F}_r). From Table 1, one can see that BRN can achieve higher average PSNR and SSIM metrics than CRN, $\text{BRN}_{r \rightarrow x}$ and $\text{BRN}_{x \rightarrow r}$ on all these three benchmark datasets, validating the effectiveness of BLSTMs.

4. CONCLUSION

In this paper, we proposed a novel bilateral recurrent network for effective single image deraining, where the proposed BLSTMs are effective in propagating reciprocal information between rain streak layer and clean background image layer. Benefiting from bilateral recurrent network, both rain streak layer and clean background image layer can be well exploited, leading to visually favorable deraining results. Extensive experimental results validate the superiority of BRN against state-of-the-art deep deraining networks. Moreover, the proposed bilateral modeling for dual layers can be applied to other two-layer image decomposition problems in the future work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants (Nos. 61801326, 61876127 and 61732011), and was supported by SenseTime Research Fund for Young Scholars. We gratefully acknowledge the support of NVIDIA Corporation with the donation of GPU.

5. REFERENCES

- [1] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu, “Automatic single-image-based rain streaks removal via image decomposition,” *IEEE TIP*, 2012. 1
- [2] Xueyang Fu, Borong Liang, Yue Huang, Xinghao Ding, and John Paisley, “Lightweight pyramid networks for image deraining,” *IEEE TNNLS*, 2019. 1, 3
- [3] Yi-Lei Chen and Chiou-Ting Hsu, “A generalized low-rank appearance model for spatio-temporally correlated rain streaks,” in *ICCV*, 2013. 1
- [4] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown, “Rain streak removal using layer priors,” in *CVPR*, 2016. 1, 3, 4
- [5] Yu Luo, Yong Xu, and Hui Ji, “Removing rain from a single image via discriminative sparse coding,” in *ICCV*, 2015. 1
- [6] S. Gu, D. Meng, W. Zuo, and L. Zhang, “Joint convolutional analysis and synthesis sparse representation for single image layer separation,” in *ICCV*, 2017. 1
- [7] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang, “Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising,” *IEEE TIP*, 2017. 1
- [8] Jun Xu, Lei Zhang, David Zhang, and Xiangchu Feng, “Multi-channel weighted nuclear norm minimization for real color image denoising,” in *ICCV*, 2017. 1
- [9] Dongwei Ren, Wangmeng Zuo, David Zhang, Lei Zhang, and Ming-Hsuan Yang, “Simultaneous fidelity and regularization learning for image restoration,” *IEEE TPAMI*, 2019. 1
- [10] Jun Xu, Lei Zhang, and David Zhang, “External prior guided internal prior learning for real-world noisy image denoising,” *IEEE TIP*, 2018. 1
- [11] Chunwei Tian, Yong Xu, and Wangmeng Zuo, “Image denoising using deep cnn with batch renormalization,” *Neural Networks*, 2020. 1
- [12] Jun Xu, Lei Zhang, Wangmeng Zuo, David Zhang, and Xiangchu Feng, “Patch group based nonlocal self-similarity prior learning for image denoising,” in *ICCV*, 2015. 1
- [13] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang, “Image super-resolution using deep convolutional networks,” *IEEE TPAMI*, 2016. 1
- [14] Jiawei Zhang, Jinshan Pan, Jimmy Ren, Yibing Song, Linchao Bao, Rynson W.H. Lau, and Ming-Hsuan Yang, “Dynamic scene deblurring using spatially variant recurrent neural networks,” in *CVPR*, 2018. 1
- [15] Dongwei Ren, Kai Zhang, Qilong Wang, Qinghua Hu, and Wangmeng Zuo, “Neural blind deconvolution using deep priors,” *arXiv preprint arXiv:1908.02197*, 2019. 1
- [16] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu, “Residual dense network for image super-resolution,” in *CVPR*, 2018. 1
- [17] 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 *CVPR*, 2017. 1, 2, 3, 4
- [18] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha, “Recurrent squeeze-and-excitation context aggregation net for single image deraining,” in *ECCV*, 2018. 1, 2, 3, 4
- [19] Xueyang Fu, Jabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley, “Clearing the skies: A deep network architecture for single-image rain removal,” *IEEE TIP*, 2017. 1
- [20] Xueyang Fu, Jabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley, “Removing rain from single images via a deep detail network,” in *CVPR*, 2017. 1, 2, 3, 4
- [21] 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
- [22] He Zhang and Vishal M Patel, “Density-aware single image deraining using a multi-stream dense network,” in *ICCV*, 2018. 1, 2
- [23] Hongyuan Zhu, Xi Peng, Joey Tianyi Zhou, Songfan Yang, Vijay Chandrasekh, Liyuan Li, and Joo-Hwee Lim, “Single image rain removal with unpaired information: A differentiable programming perspective,” in *AAAI*, 2019. 1
- [24] Wei Wei, Deyu Meng, Qian Zhao, Zongben Xu, and Ying Wu, “Semi-supervised transfer learning for image rain removal,” in *CVPR*, 2019. 1
- [25] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng, “Progressive image deraining networks: a better and simpler baseline,” in *CVPR*, 2019. 2
- [26] Sepp Hochreiter and Jürgen Schmidhuber, “Long short-term memory,” *Neural computation*, 1997. 2
- [27] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo, “Convolutional lstm network: A machine learning approach for precipitation nowcasting,” in *NIPS*, 2015. 2
- [28] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE TIP*, 2004. 3, 4
- [29] Diederik P Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” in *ICLR*, 2015. 3