17th International Conference on Machine Vision Applications (MVA)
Fully Online, July 25–27, 2021.

Recurrent RLCN-Guided Attention Network for Single Image Deraining

Yizhou Li[1], Yusuke Monno[2], Masatoshi Okutomi[3]
Tokyo Institute of Technology
Tokyo, Japan
{yli[1], ymonno[2]}@ok.sc.e.titech.ac.jp, mxo[3]@sc.e.titech.ac.jp

Abstract

Single image deraining is an important yet challenging task due to the ill-posed nature of the problem to derive the rain-free clean image from a rainy image. In this paper, we propose Recurrent RLCN-Guided Attention Network (RRANet) for single image deraining. Our main technical contributions lie in threefold: (i) We propose rectified local contrast normalization (RLCN) to apply to the input rainy image to effectively mark candidates of rain regions. (ii) We propose RLCN-guided attention module (RLCN-GAM) to learn an effective attention map for the deraining without the necessity of ground-truth rain masks. (iii) We incorporate RLCN-GAM into a recurrent neural network to progressively derive the rainy-to-clean image mapping. The quantitative and qualitative evaluations using representative deraining benchmark datasets demonstrate that our proposed RRANet outperforms existing state-of-the-art deraining methods, where it is particularly noteworthy that our method clearly achieves the best performance on a real-world dataset.

1 Introduction

Single image deraining is a highly ill-posed task to remove the rain from a single rainy image and has been treated as a significant process, since the rain-degraded images may disturb many high-level computer vision tasks, such as object detection [10], video surveillance [11], and autonomous driving [1]. Various deraining methods based on a physical or a subjective prior on rain streaks have been proposed [2, 9, 24]. However, these prior-based methods have limited applicability for the images captured under complex real rainy situations. On the other hand, data-driven learning-based deraining methods based on a convolutional neural network (CNN) have recently demonstrated their superior performance for synthetic and real-world benchmark datasets (see [7, 22] for reviews).

In this paper, we propose Recurrent RLCN-Guided Attenion Network (RRANet) based on an encoder-decoder architecture to deal with single image deraining. Firstly, we propose rectified local contrast normalization (RLCN) to apply to the rainy image and exploit the calculated RLCN image as an additional deraining network input and a guide to learn an attention map, where RLCN is applied to extract the pixels having a higher pixel intensity than the average pixel intensity of neighborhood pixels in the local window. We find that, since rain streaks usually show a higher pixel intensity, the extracted pixels by RLCN cover almost all candidates of the rain regions. Meanwhile, with the contrast normalization within a local window, weak rain regions are extracted as well as strong rain regions (see Fig. 1(b)), contributing to a highly capable network.

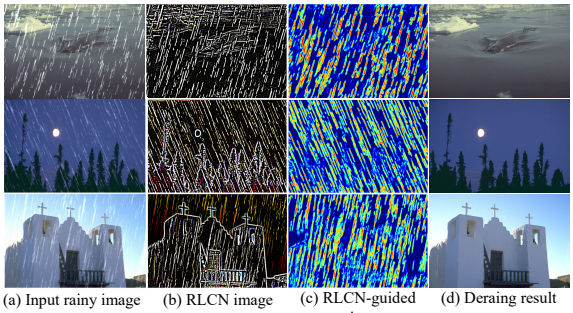

Figure 1: Three examples of (a) input rainy images, (b) our proposed RLCN images, (c) our proposed RLCN-guided attention maps, and (d) our deraining results

Secondly, we propose an RLCN-guided attention module (RLCN-GAM) which utilizes the RLCN image as a guide to learn an attention map for the deraining. RLCN-GAM is able to generate an effective attention map, where the rain streaks and their surrounding regions have high attention scores, while the other regions have low attention scores (see Fig. 1(c)). Unlike the methods in [17, 20, 21] which learn a rain mask or rain attention map directly using ground-truth rain mask supervision, our proposed RLCN-GAM does not require any supervision, enabling better generalization for real-world rainy situations.

Finally, inspired by [8, 13, 14], we introduce an RNN framework to divide the network training into multiple stages and progressively remove the rain. In order to effectively adopt the RNN into an encoder-decoder architecture, we propose to introduce a residual block with an Long short-term memory (LSTM) layer into the encoder part of the network to effectively encode the features using the RLCN image. We also apply our proposed RLCN-GAM for every recurrent stage.

In the experiments, we compare our proposed RRANet with state-of-the-art single image deraining methods, including some of the above-mentioned methods [8, 13, 14, 17, 20], using six representative deraining datasets for synthetic or real situations. The experimental results demonstrate that RRANet outperforms the state-of-the-art meth-

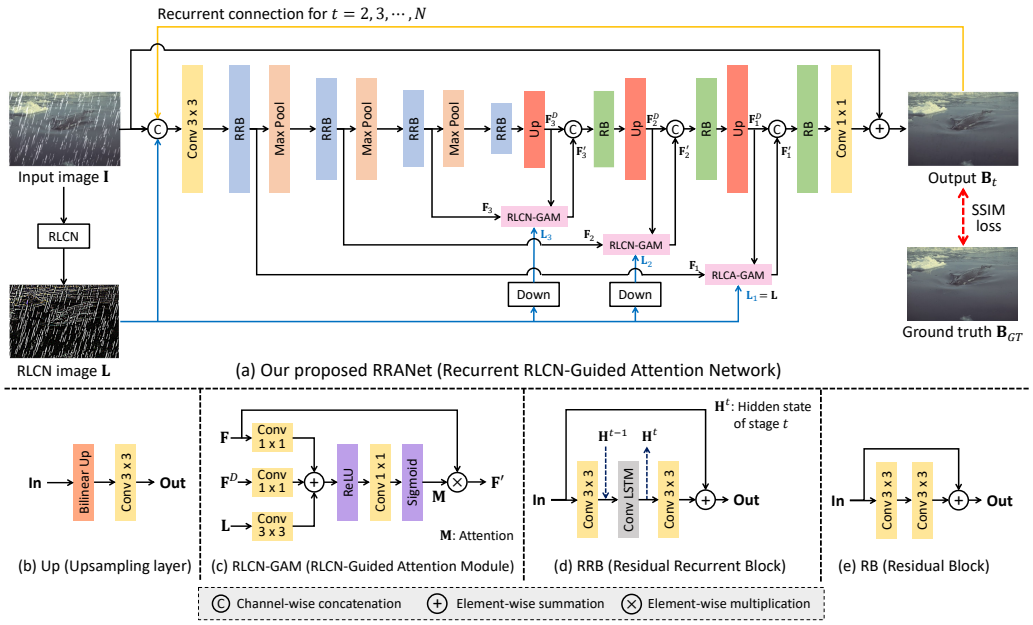

Figure 2: Overview of our proposed RRANet and its components.

ods. In particular, it clearly achieves the best performance on a real-world dataset, exhibiting a strong generalization capability under intricate real-world scenarios.

2 Proposed RRANet

2.1 Overall Architecture

Figure 2(a) illustrates the overall network architecture of our proposed RRANet. RRANet is based on a recurrent encoder-decoder architecture, where the whole deraining process is separated into multiple stages. For the encoder-decoder architecture, we apply U-Net [15] with long-range (encoder-to-decoder) and short-range (residual block) shortcut connections.

We first calculate the RLCN image $\mathbf{L}$ from the input rainy image $\mathbf{I}$ to extract the information suggesting rain regions, which will be detailed in Sec. 2.2. For each recurrent stage t of the network, the channel-wise concatenation of $\mathbf{I}$, $\mathbf{L}$, and $\mathbf{B}_{t-1}$, which denotes the output image of the previous stage $t-1$, is used as the network input, where we set $\mathbf{B}_0 = \mathbf{I}$ for the first stage. For the encoder part, the input firstly passes through a 3×3 convolutional layer. As shown in Fig. 2(d), the extracted features are then encoded with a residual recurrent block (RRB), whose details will be introduced in Sec. 2.4. The output features of RRB is then downsampled with a 2×2 max pooling, which are send to the next scale of the encoder.

For the decoder part, the output features of the last RRB are fed to an upsampling layer as shown in Fig. 2(b). Then, as shown in Fig. 2(c), the decoded features $\mathbf{F}^D$, the features from the encoder of the same scale $\mathbf{F}$, and the downsampled RLCN image is send to the proposed RLCN-GAM to generate a spatial attention map $\mathbf{M}$ to derive attentive mapping regions and generate the re-weighted features $\mathbf{F}' = \mathbf{M} \circ \mathbf{F}$, where details will be introduced in Sec. 2.3. The features $\mathbf{F}'$ is concatenated with $\mathbf{F}^D$ and send to a residual block (RB). The output features of RB are then fed to the next upsampling layer and these processes are repeated three times to recover the original spatial resolution.

The output feature maps of the last RB in the decoder part then passes through a 1×1 convolutional layer to generate the negative rain layer, which is added with the input rainy image $\mathbf{I}$ to derive the derained background image $\mathbf{B}_t$ of the current stage t. Then, the current output $\mathbf{B}_t$ is used as the input of the next stage $t+1$ and the overall network flow is repeated until the iteration reaches the maximum recurrent time N. For the loss function to optimize the whole network parameters, we simply apply the negative structural similarity index measure (SSIM) loss using the last stage output $\mathbf{B}_{t=N}$, which can be represented as

$$L = -SSIM(\mathbf{B}_{t=N}, \mathbf{B}_{GT}) \tag{1}$$

where $\mathbf{B}_{GT}$ denotes the groundtruth rain-free image.

2.2 RLCN

We propose RLCN based on the LCN [6] as

$$\mathbf{L}(i,j,c) = \frac{\max\left(\mathbf{I}_c(i,j) - \mu_{\mathbf{I}_c}(i,j), 0\right)}{\sigma_{\mathbf{I}_c}(i,j) + \epsilon}, \tag{2}$$

where $\mathbf{L}$ is the RLCN image, (i,j) denotes the pixel coordinate, $c \in (R,G,B)$ represents the color channel, $\mathbf{I}_c$ is c-th

channel of the input rainy image, $\mu_{\mathbf{I}}(i,j)$ denotes the mean pixel intensity within a local square window centered at the pixel (i,j), $\sigma_{\mathbf{I}}(i,j)$ denotes the standard deviation of the pixel intensities within the local window, ϵ is added for numerical stability, and $max(\cdot,\cdot)$ is the rectification function outputting the maximum between the two elements.

In Eq. (2), the subtracted value, $\mathbf{I}(i,j) - \mu_{\mathbf{I}}(i,j)$, takes a high absolute value if the intensity of the pixel (i,j) is significantly higher or lower than its neighbor pixels within the local window. As we focus on the rain, which usually shows a higher pixel intensity, we apply the rectification function to filter out negative values. The positive value after the rectification is further normalized by the standard deviation $\sigma_{\mathbf{I}}(i,j)$, which corresponds to local contrast, to enable the extraction of the pixels for weak rain regions as well as strong rain regions.

As show in Fig. 1(b), the derived RLCN image can cover most of the rain regions and thus can be used as a guide to learn the deraining network. However, since the RLCN image still contains non-rain edges and cannot be directly used as a rain mask, we learn an attention map by our proposed RLCN-GAM as detailed below.

2.3 RLCN-GAM

We propose RLCN-GAM to adapt the RLCN image as a soft attention map that assigns high attention scores to the pixels around rain regions while suppressing the scores for the high-value non-rain pixels in the RLCN image. RLCN-GAM is based on a self-additive attention module [12]. Following the mathematical symbol notations in Fig. 2(a) and Fig. 2(c), RLCN-GAM is expressed as

$$\mathbf{M}_s = \phi_2(\mathbf{W}_s^A * (\phi_1(\mathbf{W}_s * \mathbf{F}_s + \mathbf{W}_s^D * \mathbf{F}_s^D + \mathbf{W}_s^L * \mathbf{L}_s))), \tag{3}$$

where $\mathbf{M}_s$ is the attention map for scale s of the U-Net, $\mathbf{F}_s$ and $\mathbf{F}_s^D$ represent the features from the encoder and the decoder, respectively, and $\mathbf{L}_s$ is the RLCN image. $\mathbf{W}_s$ and $\mathbf{W}_s^D$ denote the weights for 1×1 convolutional layers, while $\mathbf{W}_s^L$ denotes the weight for the 3×3 convolutional layer to extract the features from the RLCN image. The convolution operation is expressed as $*$, and the resultant features are combined by the element-wise summation. Then, the rectified linear unit (ReLU) [4] function ϕ_1 is applied to the combined features, which followed by a one-channel 1×1 convolutional layer with the weight $\mathbf{W}_s^A$ to sum up the n-channel feature maps. The sigmoid activation function σ_2 is then applied to derive the attention map within the range of $[0, 1]$. Finally, using the obtained attention map, the re-weighted features $\mathbf{F}'_s$ is derived as $\mathbf{F}'_s = \mathbf{M}_s \circ \mathbf{F}_s$, where $\circ$ denotes the channel-wise and pixel-wise multiplication.

As shown in Fig. 1(c), our proposed RLCN-GAM can effectively generate the attention map, where rain streak regions show high attention scores while other edge regions appeared in the RLCN image of Fig. 1(c) have low attention scores, enabling substantially meaningful distinction between the attentive regions around the rain and non-rain backgrounds.

2.4 Recurrent Framework with RRB

We here introduce our recurrent framework incorporating the RLCN input and the RLCN-GAM. First, as shown in Fig. 2(d), we introduce Convolutional LSTM [19] to each RBs [5] of the encoder to share the information across successive recurrent stages. As the information flow of LSTM, the hidden state output of stage $t-1$, denoted as $\mathbf{H}^{t-1}$, and the features from the first convolutional layer are sent to the LSTM layer, where the gating signal inside LSTM decides which part of the information is preserved or thrown away. Then, the LSTM layer outputs the hidden state of current stage t, denoted as $\mathbf{H}^t$, which is used as the input of the second convolutional layer and also sent to the RRB of the next stage $t+1$. By applying LSTM, the information is effectively shared across successive recurrent stages.

3 Experimental Results

3.1 Datasets

Synthetic datasets: Five synthetic benchmark datasets, Rain100H [21], Rain200H [21], Rain100L [21], Rain200L [21], Rain800 [23] with pairs of rainy and groundtruth rain-free images are used to train and evaluate our RRANet. Rain100H/200H synthesize the heavy rain conditions, while Rain100L/200L synthesize the light rain conditions. Rain800 is another synthesized dataset containing both heavy and light rainy images.

Real-world datasets: SPA-Data [17] is used to evaluate the generalization of our RRANet to real-world scenarios, which contains high-quality 638492 training and 1000 test image pairs generated using video redundancy of real-world rainy videos and human supervision.

3.2 Comparison on Synthetic Datasets

Compared methods: We evaluate our RRANet on five synthetic datasets, Rain100H, Rain200H, Rain100L, Rain200L, and Rain800, by comparing it with state-of-the-art methods including high-frequency-component-based DDN [3], semi-supervised SIRR [18], rain-mask-based JORDER-E [20], model-driven RCDNet [16], and RNN-based RESCAN [8], PReNet [14], and BRN [13]. RCDNet and BRN are very recent state-of-the-art methods. All the methods are retrained with their original settings unless the pretrained models are provided.

Results: Table 2 reports the PSNR and the SSIM results. We can see that our RRANet provides good results with the other compared methods on each dataset, especially on Rain100H, Rain200L and Rain800, showing the strong adaptability of RRANet under different rain conditions. Compared to existing rain-mask-based JORDER-E and RNN-based methods (RESCAN, PReNet, and BRN), our RRANet shows significant performance improvements. From Fig. 4, we can see that only our RRANet can recover the details of the bridge, which are highly occluded

Table 1: Quantitative comparison on synthetic datasets (Red: rank 1st; Blue: rank 2nd)

Dataset	Rain100H		Rain200H		Rain100L		Rain200L		Rain800	
Metrics	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DDN [3]	26.79	0.814	26.10	0.807	34.61	0.959	34.39	0.960	25.47	0.836
RESCAN [8]	28.82	0.867	27.95	0.862	38.09	0.980	38.43	0.982	28.36	0.872
SIRR [18]	22.03	0.714	22.17	0.726	32.31	0.926	32.21	0.931	22.73	0.762
PReNet [14]	30.31	0.910	29.47	0.907	37.21	0.978	37.93	0.983	26.82	0.888
JORDER-E [20]	30.22	0.898	29.23	0.894	39.36	0.985	39.13	0.985	27.92	0.883
RCDNet [16]	31.26	0.912	**30.18**	0.909	**39.76**	**0.986**	**39.49**	**0.986**	**28.66**	0.893
BRN [13]	**31.32**	**0.924**	**30.27**	**0.919**	38.16	0.982	38.86	0.985	28.31	**0.896**
RRANet (ours)	**31.60**	**0.924**	**30.18**	**0.916**	**39.60**	**0.986**	**39.72**	**0.987**	**29.46**	**0.908**

Table 2: Quantitative comparison on real-world SPA-Data (Red: rank 1st; Blue: rank 2nd).

Methods	SPANet [17]		RCDNet [16]		RRANet(ours)	
Metrics	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Results	40.04	0.984	**41.05**	**0.985**	**43.53**	**0.991**

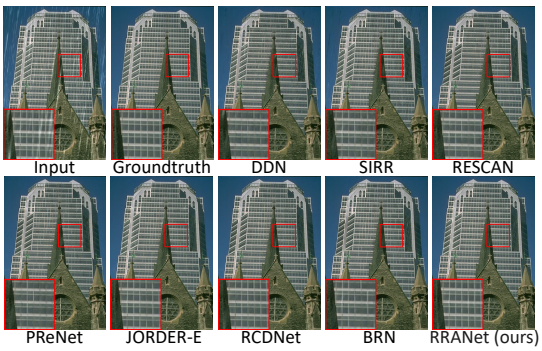

Figure 3: Qualitative comparison on Rain100L

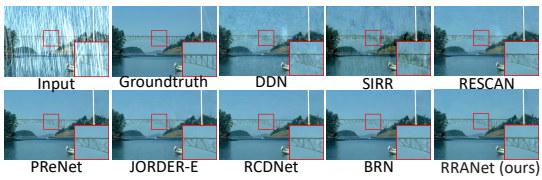

Figure 4: Qualitative comparison on Rain100H

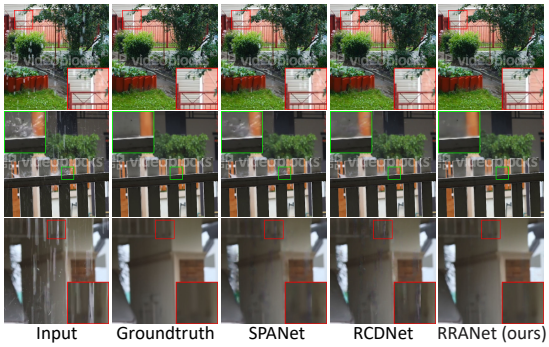

Figure 5: Qualitative comparison on real SPA-Data

by rain streaks in the input rainy image. From Fig. 3, we can see that our RRANet can better recover the window frames of the building that have similar appearance to the rain streaks, showing the high capability of our RRANet using the RLCN image.

3.3 Comparison on Real-world Dataset

Compared methods: We evaluate the generalization ability of our proposed RRANet for real-world images of SPA-Data [17]. SPANet [17] and RCDNet [16] are included into the comparison as they release their pretrained model on full SPA-Data's training set.

Results: Table 2 shows the PSNR and the SSIM results. We can see that our RRANet provides remarkably better results for real-world rainy images, compared with SPANet [17], which is designed and presented with SPA-Data, and a very recent state-of-the-art RCDNet [16], showing the strong real-world generalizaion capability of the proposed RRANet. As shown in Fig. 5, our RRANet can remove the rain more completely compared to the other methods, with least rain streaks remained, which proves that our RRANet is able to better deal with weak rain streaks and unpredictably complex rain patterns that generally appear in real-world scenarios.

4 Conclusion

In this paper, we have proposed a novel single image deraining network called RRANet that exploits an RLCN image, which can be calculated from the input rainy image by a simple computation, as an additional network input and also a guide to generate an attention map for deriving the attentive regions for training the network in an RNN framework. Because of the high capability of the RLCN image to suggest candidate rain regions, the derived attention map effectively focuses on the important regions around rain streaks. Experimental comparisons with existing methods have demonstrated that our proposed RRANet outperforms the existing state-of-the-art methods and produces higher-quality rain-removed images under synthetic and especially real-world scenarios.

References

[1] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proc. of IEEE Int. Conf. on Computer Vision (ICCV)*, pages 2722–2730, 2015.

[2] Liang-Jian Deng, Ting-Zhu Huang, Xi-Le Zhao, and Tai-Xiang Jiang. A directional global sparse model for single image rain removal. *Applied Mathematical Modelling*, 59:662–679, 2018.

[3] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3855–3863, 2017.

[4] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proc. of Int. Conf. on Artificial Intelligence and Statistics*, pages 315–323, 2011.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[6] Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In *Proc. of IEEE Int. Conf. on Computer Vision (ICCV)*, pages 2146–2153, 2009.

[7] Siyuan Li, Iago Breno Araujo, Wenqi Ren, Zhangyang Wang, Eric K Tokuda, Roberto Hirata Junior, Roberto Cesar-Junior, Jiawan Zhang, Xiaojie Guo, and Xiaochun Cao. Single image deraining: A comprehensive benchmark analysis. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3838–3847, 2019.

[8] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 254–269, 2018.

[9] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown. Rain streak removal using layer priors. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2736–2744, 2016.

[10] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017.

[11] Khan Muhammad, Jamil Ahmad, Zhihan Lv, Paolo Bellavista, Po Yang, and Sung Wook Baik. Efficient deep cnn-based fire detection and localization in video surveillance applications. *IEEE Trans. on Systems, Man, and Cybernetics*, 49(7):1419–1434, 2018.

[12] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention U-net: Learning where to look for the pancreas. In *Proc. of Conf. on Medical Imaging with Deep Learning (MIDL)*, pages 1–10, 2018.

[13] Dongwei Ren, Wei Shang, Pengfei Zhu, Qinghua Hu, Deyu Meng, and Wangmeng Zuo. Single image deraining using bilateral recurrent network. *IEEE Trans. on Image Processing*, 29:6852–6863, 2020.

[14] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3937–3946, 2019.

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc. of Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241, 2015.

[16] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A model-driven deep neural network for single image rain removal. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3103–3112, 2020.

[17] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 12270–12279, 2019.

[18] Wei Wei, Deyu Meng, Qian Zhao, Zongben Xu, and Ying Wu. Semi-supervised transfer learning for image rain removal. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3877–3886, 2019.

[19] SHI Xingjian, 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 *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 802–810, 2015.

[20] Wenhan Yang, Robby T Tan, Jiashi Feng, Zongming Guo, Shuicheng Yan, and Jiaying Liu. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 42(6):1377–1393, 2019.

[21] 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 *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1357–1366, 2017.

[22] Wenhan Yang, Robby T Tan, Shiqi Wang, Yuming Fang, and Jiaying Liu. Single image deraining: From model-based to data-driven and beyond. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2020.

[23] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *IEEE Trans. on Circuits and Systems for Video Technology*, 30(11):3943–3956, 2019.

[24] Lei Zhu, Chi-Wing Fu, Dani Lischinski, and Pheng-Ann Heng. Joint bi-layer optimization for single-image rain streak removal. In *Proc. of IEEE Int. Conf. on Computer Vision (ICCV)*, pages 2526–2534, 2017.