Single Image Deraining Using Context Aggregation Recurrent Network

Qunfang Tang[1,2], Jie Yang[1], Haibo Liu[2], Zhiqiang Guo[1], and Wenjing Jia[3]

[1]Hubei Key Laboratory of Broadband Wireless Communication and Sensor Networks, Wuhan University of Technology, Wuhan, China

[2]School of Electrical Information Engineering, Hunan Institute of Technology, Hengyang, China

[3]Global Big Data Technologies Centre (GBDTC), University of Technology Sydney, Ultimo, Australia

Single image deraining is a challenging problem due to the presence of non-uniform rain densities and the ill-posedness of the problem. Moreover, over-/under-deraining can directly impact the performance of vision systems. To address these issues, we propose an end-to-end Context Aggregation Recurrent Network, called CARNet, to remove rain streaks from single images. In this paper, we assume that a rainy image is the linear combination of a clean background image with rain streaks and propose to take advantage of the context information and feature reuse to learn the rain streaks. In our proposed network, we first use the dilation technique to effectively aggregate context information without sacrificing the spatial resolution, and then leverage a gated subnetwork to fuse the intermediate features from different levels. To better learn and reuse rain streaks, we integrate a LSTM module to connect different recurrences for passing the information learned from the previous stages about the rain streaks to the following stage. Finally, to further refine the coarsely derained image, we introduce a refinement module to better preserve image details. As for the loss function, the L1-norm perceptual loss and SSIM loss are adopted to reduce the gridding artifacts caused by the dilated convolution. Experiments conducted on synthetic and real rainy images show that our CARNet achieves superior deraining performance both qualitatively and quantitatively over the state-of-the-art approaches.

***Index Terms*—Image deraining, context awareness, dilated convolution, recurrent network, perceptual loss.**

I. Introduction

IMAGES captured outdoors in rainy weather engender the degeneracy phenomenon, such as low contrast or saturation loss, which seriously affects the performance of outdoor visual systems. In order to improve the performance of computer vision applications such as image classification, object detection, and video surveillance [1], [2], it is important to develop algorithms to remove rain streaks, especially the high density ones in heavy rain conditions. Therefore, image deraining has attracted much attention and also achieved some progress. However, due to the randomness in physical nature of rain streaks and the ill-posed problem, image deraining remains a challenging problem.

In the past, many rain removal techniques based on video systems (or sequential image data) have been proposed. These techniques used inter-frame information to detect and remove rain [3], [4], [5], [6], which has made it relatively easy to remove rain from the video. In contrast, single image deraining is more challenging due to less information available.

In recent years, deep learning-based single image deraining methods [7], [10], [11], [12], [13], [14], [15], [8], [9] were proposed and have achieved rapid progress. In [8], Chen *et al.* used the latest smoothed dilation technique and a gated subnetwork for image dehazing and deraining. In fact, the network was initially designed for dehazing. Given the similarity between a hazy image and a rainy image, Chen *et al.* applied the network for deraining and have achieved good performance. Besides, Ren *et al.* [9] proposed a simpler yet better baseline deraining network, in which they adopted both the stage-wise results and the original rainy image as the input for each residual network (ResNet) to improve the deraining performance. However, as shown in Fig. 1, the method of [7] tends to result in under-deraining (see Fig. 1(b)), and the methods of [8] and [9] are prone to produce over-deraining results (see Fig. 1(c) and (d)).

(a) Rainy image (b) GMM [7] (c) GCANet [8] (d) PReNet [9] (e) CARNet (without refinement) (f) CARNet

Fig. 1: Deraining results obtained with GMM [7], GCANet [8], PReNet [9] and our CARNet (see in a zoomed-in mode).

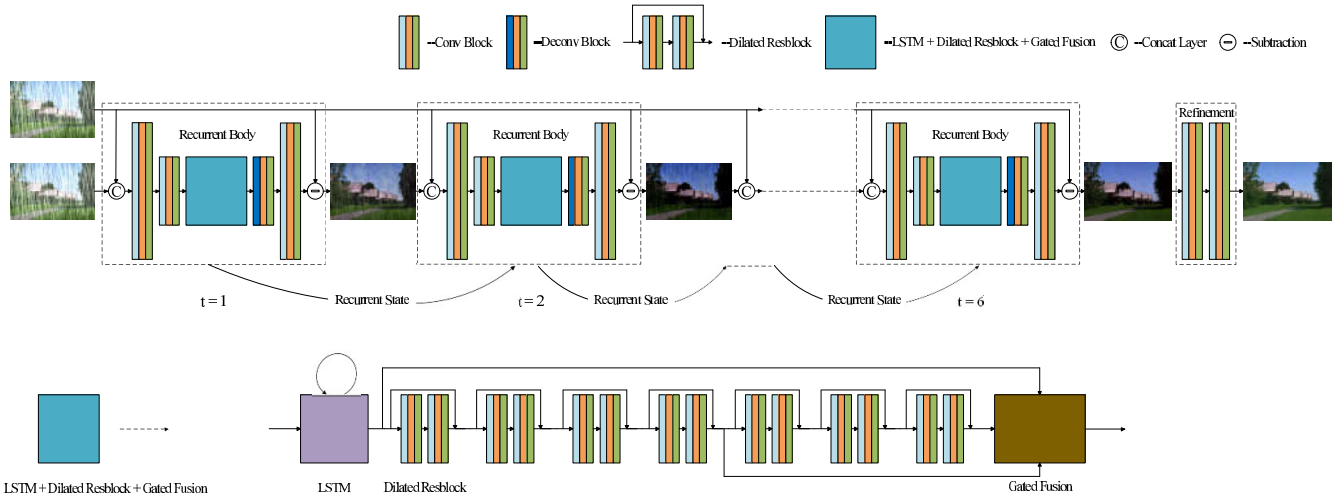


Fig. 2: The architecture of our CARNet.

Inspired by the ideas of [8] and [9], to address the problem of over- or under-deraining, in this paper we propose an end-to-end Context Aggregation Recurrent Network, called CARNet, to remove rain streaks from single images. Note that, many existing deraining methods have taken the context information into account but they have ignored the feature reuse. In our work, a rainy image can be seen as a linear combination of a clean background image with rain streaks. We propose to take advantage of the context information and feature reuse to learn the rain streaks from rainy images.

As shown in Fig. 2, our CARNet is composed of a Recurrent Body and a Refinement module. The Recurrent Body mainly consists of a Long Short Term Memory (LSTM) module, seven Dilated ResBlocks and a Gated Fusion module. We adopt the dilated convolutions to help with preserving the local and global contexts eradicating the problem of over-deraining. The LSTM module can pass the information learned during the previous stages about the rain streaks to the current Dilated Resblocks for better restoration effect. The Dilated ResBlock can effectively aggregate context information without sacrificing spatial resolution, and the Gated Fusion module is used to determine the importance of different levels and fuse them. Both of them are beneficial to a coarse-to-fine rain streaks estimation, thereby improving the quality of rain removal. Moreover, in the optimization process of our CARNet, we cascade a Refinement module to further refine the derained image to preserve more image details. Last but not the least, to obtain more effective processing, we use a recursive learning scheme similar to the idea in [9], allowing the network to gradually expand without increasing model parameters.

Compared to the method of [8], as shown in Fig. 1(c) and (f), one can see that the result of [8] has obvious rain traces after deraining, *i.e.*, the result of [8] suffers the under-deraining issue, whereas our CARNet's result shows little signs of rains.

Moreover, from Fig. 1(d) and (f) it can be seen that, our CARNet is capable of obtaining more image details after deraining with higher clarity thanks to the Dilated ResBlock and Gated Fusion module. Comparing Fig. 1(e) with (f) in a zoomed-in mode, one can see that our Refinement module can also help alleviate the gridding artifacts caused by the dilated convolution. Furthermore, we design a hybrid loss function combining the L1-norm perceptual loss with the Structural SIMilarity (SSIM) loss [16] to train our CARNet. As shown in Fig. 8(b) and (f) of the ablation studies, the SSIM loss can effectively reduce the gridding artifacts, which can also be regarded as a perceptual metric based on image properties like luminance, contrast and structure.

To validate the effectiveness of our proposed CARNet, comprehensive experiments are conducted on both synthetic and real rainy image datasets. Compared with the state-of-the-art methods on the three synthetic datasets, our CARNet achieves best quantitative and qualitative deraining results. Moreover, the visually pleasant deraining results obtained on real rainy images show the generalization ability of our proposed CARNet model.

In summary, the main contribution of this work is threefold:

- We propose an end-to-end CARNet for single image deraining which takes advantage of the context information and feature reuse to learn the rain streaks. The learned information about rain streaks is then propagated over multiple stages via LSTM to obtain better deraining results.
- We propose a hybrid loss function which considers both L1-norm perceptual loss and SSIM loss to improve the visual quality and effectively reduce the gridding artifacts.
- Extensive experiments conducted on both synthetic and real rainy images demonstrate the superiority of our CARNet over the state of the arts in both qualitative and quantitative measures.

The remainder of this paper is organized as follows. In Section II, the background and related work are introduced. In Section III, the proposed CARNet is illustrated in details. In Section IV, the experiment setup and results are presented and analyzed. Finally, the conclusion is drawn in Section V.

II. Background and Related Work

In this section, we first introduce the rain model to describe various components of a rainy image. Then, we review some recent single image deraining methods by grouping them into optimization-based methods and deep learning based methods.

(a) Rainy image O (b) Clean image B (c) Rain streaks S

Fig. 3: Rainy image decomposition. A rainy image (a) can be viewed as the superposition of a clean background image (b) and a rain streak image (c).

A. Rain Model

Mathematically, the widely used rain model [17], [7], [18] considers a rainy image as the linear combination of a clean background image with rain streaks, which is expressed as:

$$O = B + S, \tag{1}$$

where B is the clean background image, S represents the rain streaks, and O denotes the obtained image with rain streaks, as shown in Fig. 3.

Thus, the goal of single image deraining is to recover B from O, *i.e.*,

$$B = O - S. \tag{2}$$

Similar to image denosing and image separation [19], [20], single image deraining also has the ill-posed problem. In this paper, we aim to learn S from O to obtain B with our proposed CARNet.

B. Optimization-based Deraining Methods

Optimization-based deraining methods usually form an optimization problem based on some priors such as background image priors and rain streaks priors [21], [18], [7], [22], [23]. In [21], Chen *et al.* proposed a low-rank appearance model to remove rain streaks given that they always come in similar and repeated patterns. Built upon a nonlinear generative model of rainy image, Luo *et al.* [18] proposed a dictionary learning based algorithm, where rain streak layer and background image layer were forced to share the fewest dictionary items. Li *et al.* [7] assumed that Gaussian Mixture Model-based methods (GMMs) could accommodate multiple orientations and scales of rain streaks and proposed a patch-based GMM prior to rain streaks separation. However, these optimization-based deraining methods still suffer limited deraining quality due to the complexity of rainy images.

C. Deep Learning based Deraining Methods

Deep learning based methods can be regarded as data-driven methods. The success of deep learning in several computer vision tasks (*e.g.*, [24], [25], [26], [27], [28]) has driven researchers to develop deep learning based approaches for image deraining, which have become more and more popular in recent years.

A natural solution is to build a non-linear function to recover a clean background image from the corresponding rainy image. By combining the high-frequency detail-layer content of an image and regressing on the negative residual information, Fu *et al.* [10] proposed an end-to-end deep learning framework for single image deraining based on deep convolutional neural networks (CNNs).

In [13], Li *et al.* proposed a simple and effective deraining method based on CNN, which was trained in a recursive manner. Wang *et al.* [29] constructed a new real-world rain dataset, and proposed a spatial attentive network to learn discriminative deraining features in a local-to-global attentive manner. In [30], the rainy-to-clean image translation model was decomposed as a combination of an embedding learning net and a conditional generator. Then, an entangled representation learning mechanism was proposed to obtain better deraining results. In [31], Du *et al.* proposed a variational image deraining method by formulating image deraining in a conditional variational auto-encoder framework, which could learn a mapping from a single input image to many outputs. In [32], by uncovering the correlations of rain streaks in an image, a multi-scale progressive fusion network was proposed, which collaboratively represented rain streaks from multiple scales via the pyramid representation. Recently, the semi-supervised learning scheme has been adopted in image deraining due to the lack of paired rainy-clean images in real world such as [33], [34].

In this work, our CARNet utilizes the dilated CNN and residual learning to obtain rain streaks, and leverages on a recurrent scheme to further improve the deraining performance.

III. Context Aggregation Recurrent Network

In this section, our CARNet is elaborated from three perspectives, namely, network architecture, loss function and network algorithm. To this end, we first present the proposed CARNet architecture, then discuss the loss function used in our network, and finally illustrate the detailed algorithm.

A. CARNet Architecture

As shown in Fig. 2, the proposed CARNet consists of two parts, *i.e.*, the Recurrent Body and the Refinement module. The Recurrent Body is composed of three conv blocks, one deconv block, one LSTM module, seven Dilated ResBlocks and one Gated Fusion module. The Refinement module consists of two conv blocks.

Conv Block + Deconv Block. In the Recurrent Body, the conv block and deconv block appear symmetrically, and each block includes a convolutional layer, a normalization layer (InstanceNorm) and an activation function (ReLU). All the filters in conv blocks are with a size of 3×3 with 1×1 padding. Due to the concatenation of 3-channel RGB images, the first convolutional layer of the Recurrent Body has 6 and 64 channels for input and output, respectively. The second convolutional layer downsamples the feature maps by 1/2 to obtain the recursive input feature map I_{in}. In order to upsample the feature map to the original resolution, the filter in deconv block is with a size of 4×4, a stride of 2, and 1×1 padding. After passing the last conv block, we get the intermediate rain streaks S_{im}, and use Eq. 2 to obtain the intermediate background image B_{im}.

LSTM Module + Dilated ResBlock. LSTM modules usually work with loops. In our CARNet, a LSTM module

is used to pass the information learned from the previous steps about rain streaks to the next step, and the structure of LSTM can be found in [35], [36]. In order to increase the original receptive field without reducing resolution [8], the dilated convolution is adopted in our CARNet, and the dilation rates of these seven ResBlocks are set as (2, 2, 2, 4, 4, 4, 1), respectively. The filters in the LSTM module and Dilated ResBlocks are also with a size of 3×3. However, the padding size of the LSTM module is 1×1, and the padding sizes in the Dilated ResBlocks are the same as the dilation rates. The LSTM module and Dilated ResBlock in our CARNet benefit each other. In other words, the Dilated ResBlock has greater capacity to affect the accuracy of the extraction of rain streaks compared to the LSTM module. However, the advantage of the LSTM module is that it can make full use of the information in previous iterations [37]. Therefore, combining the benefits of the LSTM module and Dilated ResBlock can greatly improve the performance of single image deraining.

Gated Fusion Module. It is well known that fusing the features of different levels can effectively achieve overall performance improvement in deep learning [38], [39]. From Fig. 2, three feature maps from different levels F_l, F_m and F_h are fed into the gated subnetwork, and the recursive output feature map I_{out} is computed as:

$$I_{out} = W_l * F_l + W_m * F_m + W_h * F_h, \tag{3}$$

where W_l, W_m and W_h are the weights obtained from the gated fusion module and they correspond to each feature level, respectively. The Gated Fusion module is a convolutional layer, which has 3×64 and 3 channels for the input and output, respectively. The input is the concatenation of F_l, F_m and F_h, and the kernel size of the filter is 3×3.

Refinement Module. To further refine the derained images and preserve more image details, two convolutional layers with ReLU are adopted in the Refinement module. The size and padding of the two filters are $(7\times7,\ 3\times3)$ and $(3\times3,\ 1\times1)$, respectively. After six (6) recursive operations, B_{im} is regarded as the input of the Refinement module to obtain the deraining image B_{out}.

B. Loss Function

In recent studies, loss functions measuring the difference of high-level feature representations, such as the loss on certain layers in CNN [40], has demonstrated much better visual performance than the per-pixel loss used in traditional CNNs [41]. However, as shown in Fig. 8(b), despite the fact that the loss function measured on certain feature layers can obtain better visual effect, it usually fails to achieve good quantitative performance simultaneously. In order to further improve the deraining performance of our CARNet, we propose a hybrid loss function L, which is defined as:

$$L = L_p + L_{ssim}, \tag{4}$$

where L_p represents the perceptual loss and L_{ssim} represents the SSIM loss.

In Eq. 4, L_p is the feature loss from the layers relu1_2 and relu2_2 of the VGG-16 model [42]. In order to obtain more edge details from the deraining image, L_1 norm is adopted in the perceptual loss L_p to minimize the distance between adjacent feature layers. In this paper, SSIM loss, denoted as L_{ssim}, is used to reduce the gridding artifacts caused by the dilated convolution during the training process, and further improve the quantitative performance of deraining. Different from the negative SSIM loss in [9], our SSIM loss only needs to calculate the similarity between the derained image B_{out} and the corresponding ground-truth clean image B_{GT}, *i.e.*

$$L_{ssim} = 1 - SSIM(B_{out}, B_{GT}), \tag{5}$$

where $SSIM(\cdot)$ indicates the similarity function.

C. CARNet Algorithm

By combing the previous two subsections with the content of Fig. 2, Algorithm 1 summarizes the steps involved to train our CARNet, where Num is the number of training images.

Algorithm 1 CARNet algorithm flow.

1: **for** $i = 1;\ i < Num;\ i++$ **do**
2: **for** $t = 1;\ t < 7;\ t++$ **do**
Step 1. Using the conv block to obtain the recursive input feature map I_{in};
Step 2. Adopting LSTM module, Dilated ResBlock and Gated Fusion module to get the recursive output feature map I_{out} with Eq. 3;
Step 3. Passing the deconv block to get the intermediate rain streaks S_{im};
Step 4. Using Eq. 2 to obtain the intermediate background image B_{im} for the next stage;
3: **end for**
Step 5. Regarding B_{im} as the input of the Refinement module to obtain the deraining image B_{out};
Step 6. Training with the hybrid loss function L in Eq. 4 and Eq. 5 to carry on the next loop;
4: **end for**

IV. Experiments and Results

To demonstrate the performance of our proposed CARNet, a series of experiments are conducted and compared with the state-of-the-art methods [7], [10], [14], [11], [13], [8], [9] using both synthetic and real-world images.

The proposed CARNet is implemented using Pytorch [43], and is trained on a PC with Intel Core i7 CPU 3.6 GHz, 16GB RAM and NVIDIA TITAN Xp. In our experiments, the patch size is 100×100, and the batch size is 10. Adam is used as the optimization algorithm and the models are trained for up to 80 epochs. The learning rate starts from 0.001 and is decayed by 0.1 for every 30 epochs.

A. Datasets and Evaluation Metrics

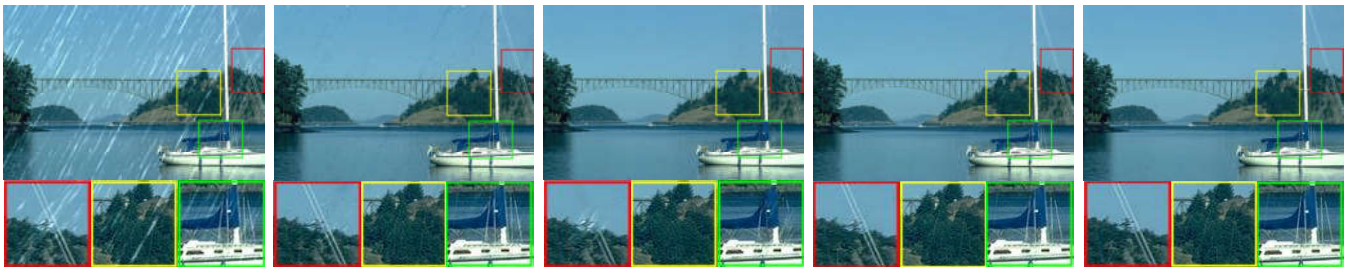
Datasets. The proposed CARNet is first evaluated on three synthetic datasets, namely, Rain100L [11], Rain100H [11] and Rain800 [41]. Rain100L is a synthesized dataset with only one type of rain streaks and is light rain. In Rain100H and Rain800, the rain pixels contain different intensities and orientations, and they are heavy rains. In addition, the real-world rainy images used in [41], [9], [1] are adopted to demonstrate the

TABLE I: Average PSNR and SSIM comparison (PSNR/SSIM) on the synthetic datasets Rain100L [11] and Rain100H [11]. The best and second best results are highlighted in red and green, respectively.

Method	GMM [7]	DDN [10]	RGN [14]	JORDER [11]	RESCAN [13]	GCANet [8]	PReNet [9]	CARNet
Rain100L	28.66/0.865	32.16/0.936	33.16/0.963	36.61/0.974	-	35.03/0.957	37.48/0.979	38.34/0.981
Rain100H	15.05/0.425	21.92/0.764	25.25/0.841	26.54/0.835	28.88/0.866	27.12/0.821	29.46/0.899	30.62/0.907

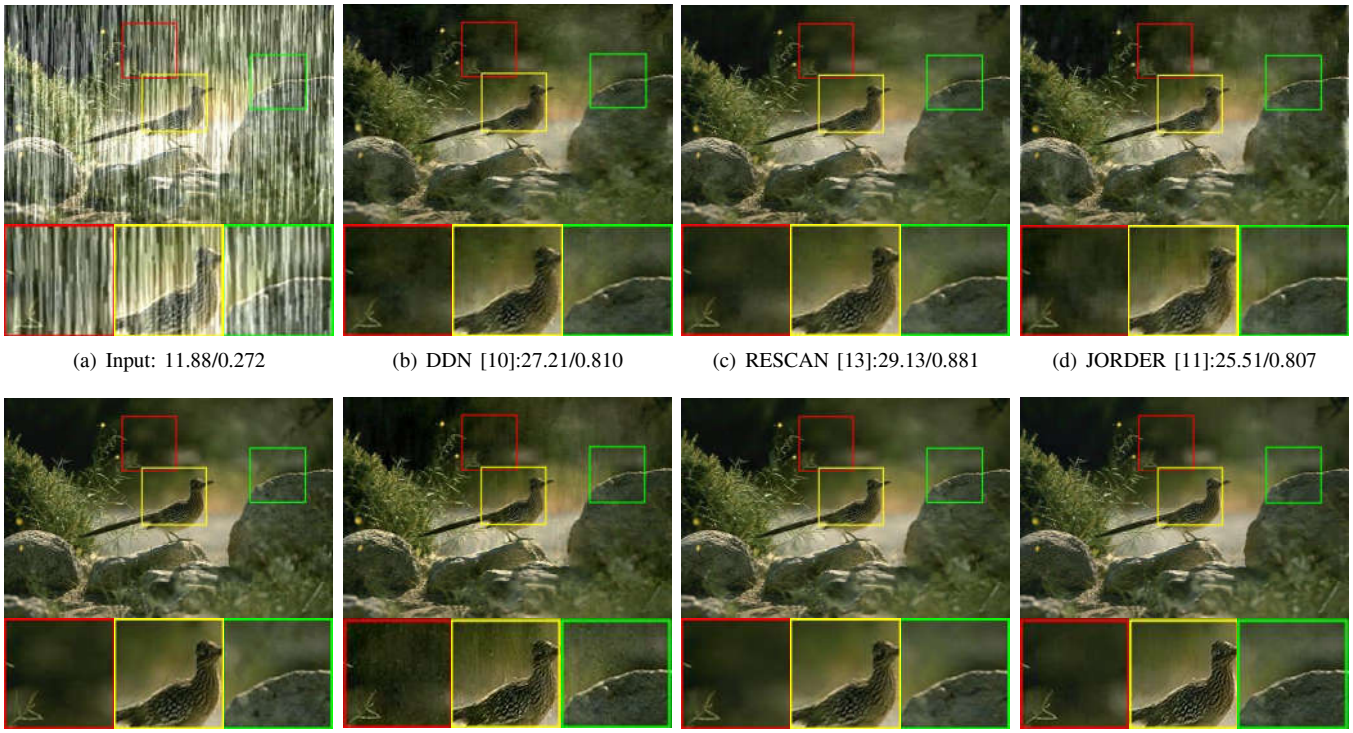
TABLE II: Quantitative comparison on Rain800 [41]. The best and second best results are highlighted in color. ▸ means the metrics are cited from [45].

Method	Rainy image	GMM [7]	DDN [10]	RGN [14]	JORDER [11]▸	RESCAN [13]	GCANet [8]	PReNet [9]	CARNet
PSNR	22.60	22.86	25.22	25.57	26.03	27.70	27.68	26.43	28.12
SSIM	0.707	0.742	0.841	0.850	0.850	0.867	0.879	0.888	0.896



(a) Input: 27.36/0.816 (b) GCANet: 37.73/0.967 (c) PReNet: 36.96/0.977 (d) CARNet: 41.13/0.989 (e) Ground truth: Inf/1

Fig. 4: Qualitative comparison and PSNR/SSIM of deraining results by GCANet [8], PReNet [9] and our CARNet on an image from Rain100L.



(a) Input: 11.88/0.272 (b) DDN [10]:27.21/0.810 (c) RESCAN [13]:29.13/0.881 (d) JORDER [11]:25.51/0.807

(e) Ground truth: Inf/1 (f) GCANet [8]:28.72/0.856 (g) PReNet [9]:30.28/0.909 (h) CARNet: 31.25/0.928

Fig. 5: Qualitative comparison and PSNR/SSIM of deraining results by DDN [10], RESCAN [13], JORDER [11], GCANet [8], PReNet [9] and our CARNet on an image from Rain100H.

effectiveness of our CARNet. These rainy images are used for evaluation purpose only.

Evaluation Metrics. In our experiments, two widely used metrics, namely, Peak Signal to Noise Ratio (PSNR) [44] and SSIM, are adopted as the quality metrics, and both are calculated using the same methods as in [9]. Generally speaking, the higher the values of PSNR and SSIM, the better the rain removal effect.

B. Evaluation on Synthetic Datasets

In this subsection, we compare the quantitative and qualitative performance of different methods on Rain100L, Rain100H

TABLE III: Comparison of our CARNet ($t = 6$) with four loss functions on Rain100H. The best results are highlighted in red.

Loss	Rainy image	L_p	$L_0(L_p + L_{pw-L2})$	$L_1(L_p + L_{pw-L1})$	$L(L_p + L_{ssim})$
PSNR	13.55	28.34	29.93	30.31	30.62
SSIM	0.379	0.835	0.893	0.898	0.907

TABLE IV: Comparison of our CARNet with different t on Rain100H. The best results are highlighted in red.

Stage	2	3	4	5	6	7
PSNR	27.27	28.74	30.38	30.41	30.62	30.18
SSIM	0.878	0.899	0.904	0.907	0.907	0.902

TABLE V: Comparison of our CARNet on Rain100H with/without the Refinement Module (RM) using two loss functions. The best results are highlighted in red.

RM + Loss	Rainy image	without RM + L_0	without RM + L	with RM + L
PSNR	13.55	22.85	29.65	30.62
SSIM	0.379	0.771	0.887	0.907

and Rain800 datasets.

Quantitative results of Li *et al.* [7], Fu *et al.* [10], Fan *et al.* [14], Yang *et al.* [11], Li *et al.* [13], Chen *et al.* [8], Ren *et al.* [9] and our CARNet on Rain100L and Rain100H are tabulated in Table I. Note that, except for Chen *et al.* [8] and our CARNet, the results of other methods in Table I are all cited from [9]. As for GCANet [8], we retrain it on Rain100L and Rain100H with the default settings. As shown in Table I, our CARNet achieves superior quantitative performance.

To make it easier for readers to understand qualitatively, two groups of comparative experiments are presented. As described in Section I, the proposed CARNet is inspired by the ideas of [9] and [8]. Therefore, we firstly compare the deraining results obtained by our CARNet and those obtained by GCANet [8] and PReNet [9] on Rain100L, as shown in Fig. 4.

From Fig. 4, GCANet [8] still retains a signification portion of rain traces after rain removal, and PReNet [9] mistakenly removes part of the sailing ropes as rain streaks. In contrast, our CARNet is able to retain the details of the deraining images while effectively removing the rain streaks.

Fig. 5 shows the deraining results of different methods on Rain100H. From qualitative perspective, both PReNet [9] and our CARNet have obtained comparable results, with the rain streaks completely removed. In comparison, visible dark noises along rain directions can still be observed from the results achieved by the other methods [9].

To further evaluate the performance of our CARNet, Rain800 is adopted to train DDN [10], RGN [14], RESCAN [13], GCANet [8], PReNet [9] and our CARNet. It should be noted that DDN [10], RGN [14], RESCAN [13], GCANet [8] and PReNet [9] are retrained with the default settings. From Table II, our CARNet obtains the highest scores in terms of PSNR and SSIM. As shown in Fig. 6, it can be seen that our CARNet is able to successfully remove the rain streaks and better preserve image details.

C. Evaluation on Real-World Rainy Images

The real-world rainy images contain rain streaks with different scales and complex degradation. In this subsection, two real-world rainy images, *i.e.*, 'Outdoor rain' and 'Courtyard rain', are used to compare the effectiveness of different methods, as shown in Fig. 7.

For the real rainy image 'Outdoor rain', one can see that the rain removal effect of GMM [7] (see Fig. 7(b)) is not obvious. DDN [10], JORDER [11] and GCANet [8] are able to remove majority of the rain streaks, but they still leave some rain residuals or traces in the derained images. Holistically speaking, RGN [14], RESCAN [13] and PReNet [9] achieve rain removal effect as good as our CARNet.

From the deraining results on the real rainy image 'Courtyard rain', there are obvious rain streaks in the results obtained by GMM [7], DDN [10], JORDER [11], RESCAN [13] and GCANet [8]. RGN [14] and PReNet [9] tend to add extra artifacts in the deraining images. In comparison, our CARNet obtains a clearer deraining result.

Therefore, in general, our CARNet outperforms the above methods in terms of rain streaks removal and image details preservation.

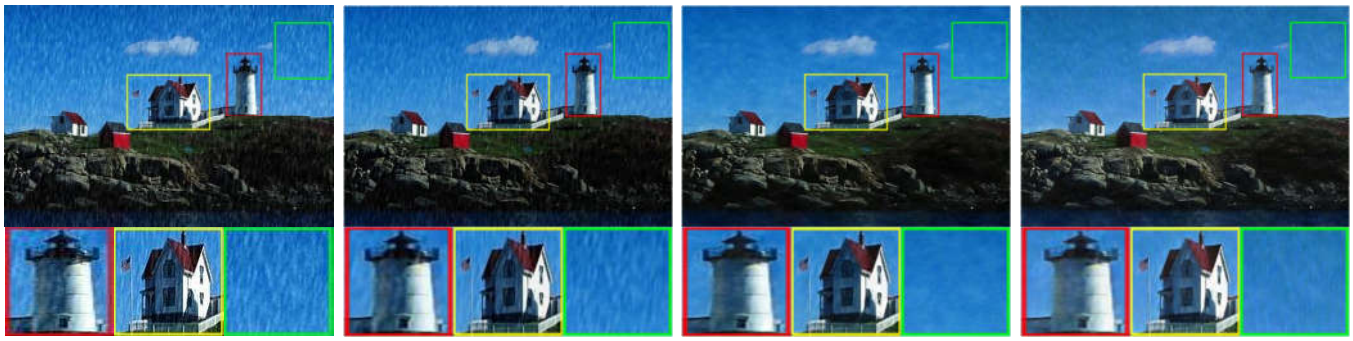
D. Ablation Studies

In our proposed CARNet, there are three important parts, *i.e.*, loss function, iteration number and Refinement module. In this subsection, we discuss their respective characteristics. All the ablation studies are conducted on Rain100H.

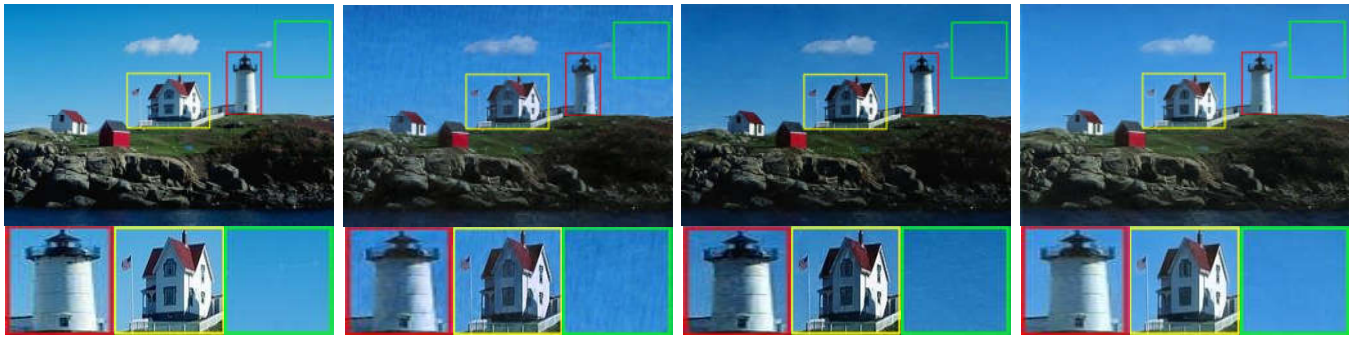
Impact of the Hybrid Loss Function. To demonstrate the effectiveness of our proposed hybrid loss function, we train four of our CARNet models by minimizing 1) the L1-norm perceptual loss, denoted as L_p, 2) the hybrid loss function $L_0 = L_p + L_{pw-L2}$, where L_{pw-L2} is the pixel-wise L2-norm loss, 3) the hybrid loss function $L_1 = L_p + L_{pw-L1}$, where L_{pw-L1} is the pixel-wise L1-norm loss, and 4) the default hybrid loss function $L = L_p + L_{ssim}$ in Eq. 4, respectively.

Table III lists their PSNR and SSIM values obtained on Rain100H. Fig. 8 illustrates the deraining results with the four loss functions with our CARNet model. In general, we can get a good rain removal effect by using any of L_p, L_0, L_1 or L. However, from Fig. 8(c), it leaves a little bit of rain traces and appears a little unnatural. That is because $L2$-norm correlates poorly with image quality as perceived by a human observer [46]. In addition, by looking at Fig. 8(b) with a zoomed-in mode, there are some gridding artifacts which contribute to the low quantitative score. That is because we adopt the dilated convolution to increase the original receptive field, which also produces gridding artifacts [47]. In comparison, as shown in Fig. 8(e) or (f), combining with L_{pw-L1} or L_{ssim}, the gridding artifacts have been greatly alleviated. Meanwhile, we have achieved better quantitative scores from Table III. Holistically speaking, the hybrid loss function L_1 produces rain removal effect as good as our default hybrid loss function L. However, it is inferior to ours in terms of structural restoration, such as the edge of the moon. Therefore, the proposed CARNet is trained with the hybrid loss function L.

Impact of the Iteration Number. As described in [13], the network model with iterations can improve deraining performance stage by stage. Table IV lists the PSNR and SSIM values of our CARNet with stages $t = 2, 3, 4, 5, 6, 7$. In general, the larger the iteration number, the higher the average values

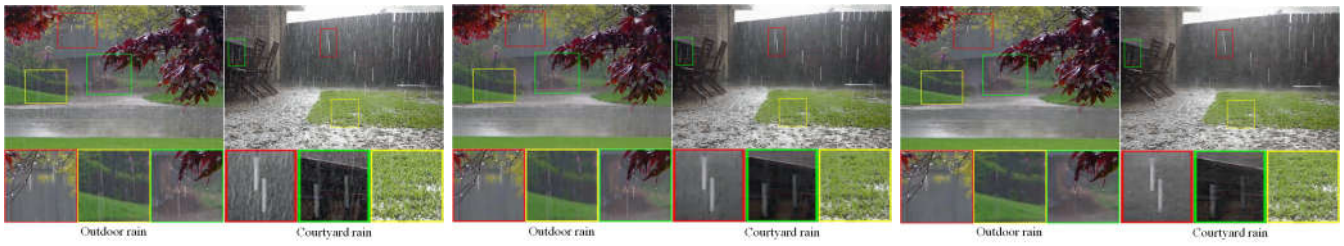


(a) Input: 21.38/0.655 (b) GMM [7]: 19.98/0.713 (c) DDN [10]: 20.46/0.809 (d) RESCAN [13]: 23.00/0.828

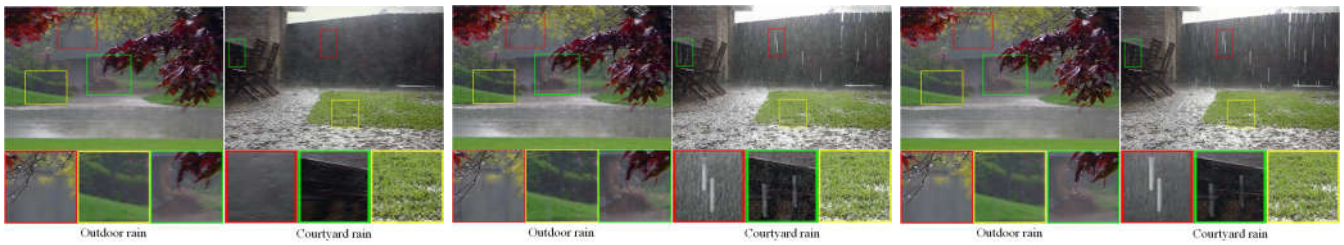


(e) Ground truth: Inf/1 (f) GCANet [8]: 23.18/0.829 (g) PReNet [9]: 20.48/0.853 (h) CARNet: 27.27/0.878

Fig. 6: Qualitative comparison and PSNR/SSIM of deraining results obtained with GMM [7], DDN [10], RESCAN [13], GCANet [8], PReNet [9] and our CARNet on an example image from Rain800.



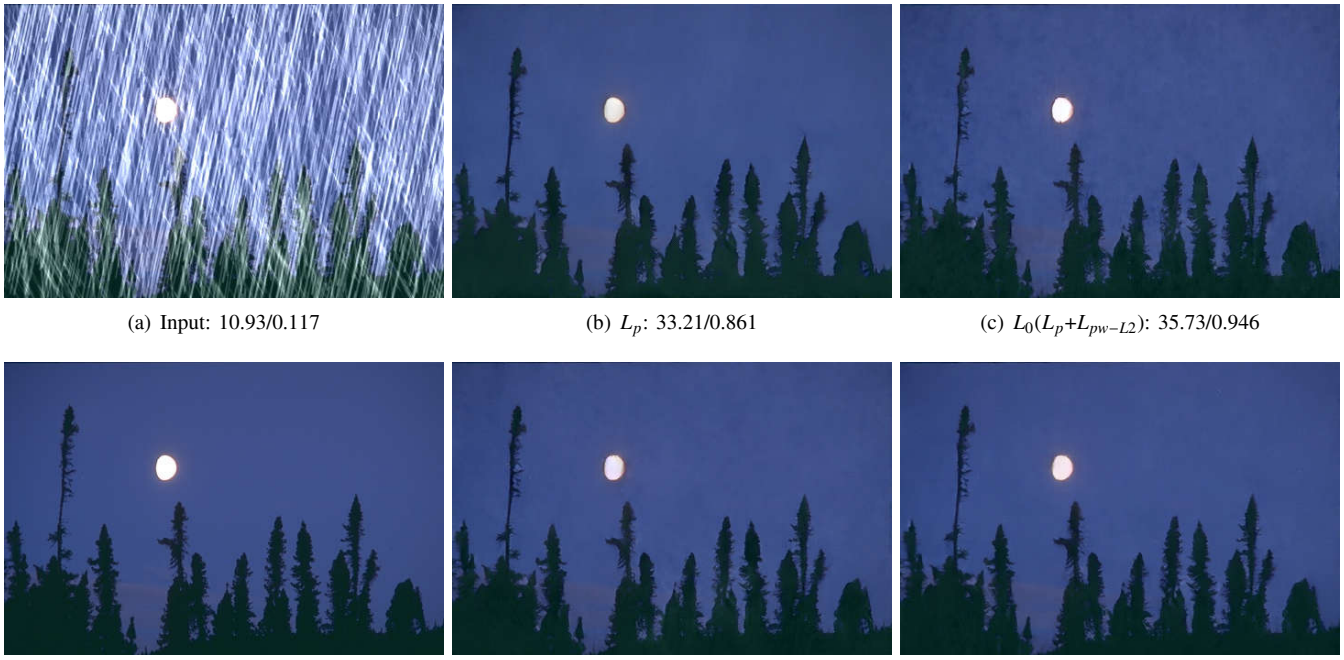
(a) Input (b) GMM [7] (c) DDN [10]



(d) RGN [14] (e) JORDER [11] (f) RESCAN [13]

(g) GCANet [8] (h) PReNet [9] (i) CARNet

Fig. 7: Visual quality comparison of deraining results by GMM [7], DDN [10], RGN [14], JORDER [11], RESCAN [13], GCANet [8], PReNet [9] and our CARNet on two real rainy images (see in a zoomed-in mode).



(a) Input: 10.93/0.117 (b) L_p: 33.21/0.861 (c) $L_0(L_p{+}L_{pw-L2})$: 35.73/0.946

(d) Ground truth: Inf/1 (e) $L_1(L_p{+}L_{pw-L1})$: 36.46/0.957 (f) $L(L_p{+}L_{ssim})$: 37.77/0.965

Fig. 8: Qualitative comparison and PSNR/SSIM scores of our CARNet models trained with four different loss functions (see in a zoomed-in mode).

(a) Input: 16.20/0.511 (b) t=2:28.84/0.910 (c) t=4:30.97/0.925

(d) Ground truth: Inf/1 (e) t=6:31.54/0.929 (f) t=7:30.22/0.917

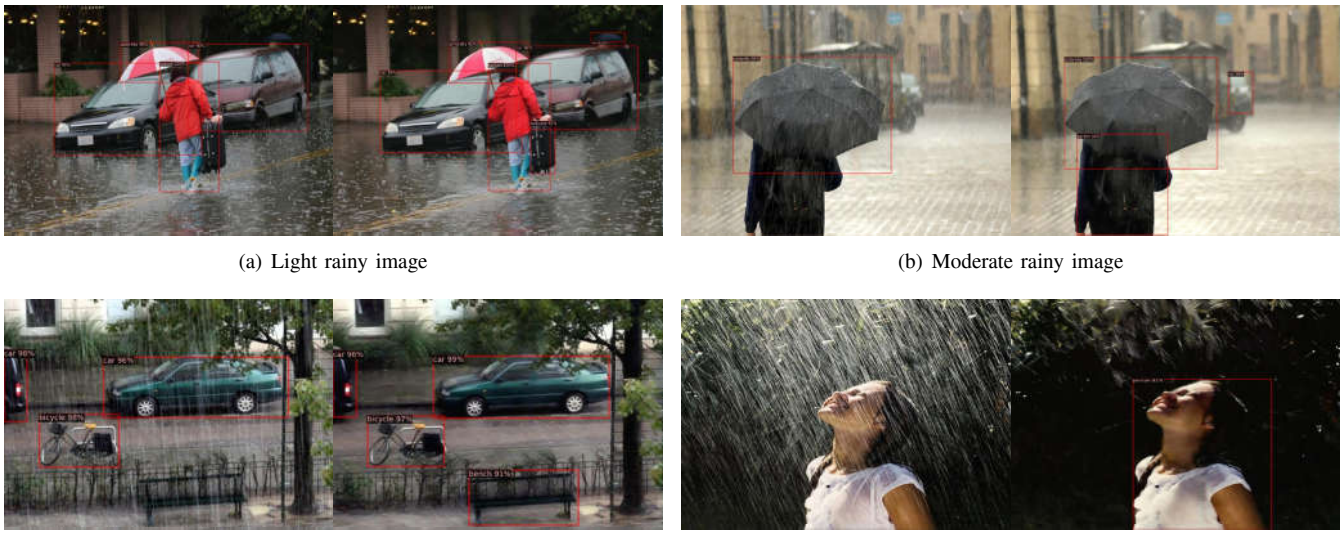
Fig. 9: Qualitative comparison and PSNR/SSIM of our CARNet trained by different iteration numbers.

of PSNR and SSIM. However, a larger t also means that the network model will become more difficult to train. Table IV and Fig. 9 show that when the value of t is larger than 6, the deraining performance of our CARNet drops significantly. Thus, we choose $t = 6$ in our CARNet for all of the above experiments.

Effectiveness of the Refinement Module. In this paper, the Refinement module is cascaded with the Recurrent Body for further improving the deraining output. To demonstrate its effectiveness, combining with the loss function, we retrain our CARNet using two loss functions without the Refinement module, separately. As shown in Table V, we obtain the best result with the default settings, *i.e.*, with RM + L, compared to the other two cases. That is to say, the Refinement module is able to effectively enhance the performance of rain removal. In addition, it can be seen from Fig. 10 (c) and (d) that the

(a) Input: 15.72/0.530 (b) w/o RM + L_0:24.93/0.848 (c) w/o RM + L:27.16/0.886 (d) with RM + L:27.69/0.902 (e) Ground truth: Inf/1

Fig. 10: Qualitative comparison and PSNR/SSIM of deraining results by our CARNet on an image from Rain100H with/without (without is abbreviated as "w/o") the Refinement Module (RM) using the two loss functions (see in a zoomed-in mode).



(a) Light rainy image (b) Moderate rainy image

(c) Heavy rainy image (d) Heavy rainy image

Fig. 11: Real-world examples of object detection. *Left*: Detection results on rainy images using Faster R-CNN (referred to as "direct detection"); *Right*: Detection results on deraining images obtained by our CARNet using Faster R-CNN (referred to as "deraining + detection").

Refinement module makes sure finer image details are well preserved. Therefore, our CARNet consists of the Refinement module.

E. Evaluation on Object Detection Results

Single image deraining can be regarded as a preprocessing step to improve the performance of other high level vision tasks such as face recognition and object detection [2]. To ascertain deraining performance improvement using our CARNet, four visual results of object detection by combing with the popular Faster R-CNN model [48] are shown in Fig. 11. Four real-world rainy images coming from [41], [9], [1] are adopted with one image corresponding to light rain, one for moderate rain and two for heavy rain. The results are shown in Fig. 11.

It is clear that, after being preprocessed by our CARNet, the detection performance has a noticeable improvement over the naive Faster R-CNN. It is worth mentioning that in Fig. 11(d), given the heavy rain conditions, we can not detect anything using Faster R-CNN without deraining. After the rain removal by our CARNet, it is labeled correctly as 'Person'. In addition, 300 real rainy images from [1] are used to collect statistics of the impact of rain removal on object detection. From our experimental results, when using the 'deraining + detection' method, the number of detected objects increases from 954 to 1061. That is to say, the derained images obtained by our CARNet can improve the object-detection effect.

V. Conclusion

In this paper, we have proposed an end-to-end context aggregation recurrent network, called CARNet, for single image deraining. Our CARNet consists of the Recurrent Body and the Refinement module, and aims to learn the rain streaks using the context information and feature reuse. During the training process, we have proposed a hybrid loss function of perceptual loss with L1 norm and SSIM loss to reduce the gridding artifacts caused by the dilated convolution and have obtained better visual quality. Detailed experiments and comparisons on synthetic and real rainy images have demonstrated that our CARNet outperforms the state-of-the-art methods.

For future work, the semi-supervised and un-supervised methods for single image deraining have attracted more and more attention, as shown in [33], [34]. We will extend our work to the semi-supervised learning for deraining in future.

Acknowledgment

The authors would like to thank the authors of compared papers, who provided related images and original codes, and

the anonymous reviewers for their insightful comments and valuable suggestions.

This work was supported by National Natural Science Foundation of China (No.51879211), Hunan Provincial Natural Science Foundation of China (No.2017JJ3053), and Hunan Provincial Science Research Project of China (Nos.17A051).

REFERENCES

[1] X. Fu, B. Liang, Y. Huang, X. Ding, and J. Paisley, "Lightweight pyramid networks for image deraining," *IEEE transactions on neural networks and learning systems*, 2019.

[2] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE transactions on image processing*, vol. 21, no. 4, pp. 1742–1755, 2011.

[3] P. C. Barnum, S. Narasimhan, and T. Kanade, "Analysis of rain and snow in frequency space," *International journal of computer vision*, vol. 86, no. 2-3, p. 256, 2010.

[4] J. Bossu, N. Hautière, and J.-P. Tarel, "Rain or snow detection in image sequences through use of a histogram of orientation of streaks," *International journal of computer vision*, vol. 93, no. 3, pp. 348–367, 2011.

[5] S. You, R. T. Tan, R. Kawakami, Y. Mukaigawa, and K. Ikeuchi, "Adherent raindrop modeling, detectionand removal in video," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 9, pp. 1721–1733, 2015.

[6] V. Santhaseelan and V. K. Asari, "Utilizing local phase information to remove rain from video," *International Journal of Computer Vision*, vol. 112, no. 1, pp. 71–89, 2015.

[7] 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*, 2016, pp. 2736–2744.

[8] D. Chen, M. He, Q. Fan, J. Liao, L. Zhang, D. Hou, L. Yuan, and G. Hua, "Gated context aggregation network for image dehazing and deraining," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 1375–1383.

[9] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, "Progressive image deraining networks: a better and simpler baseline," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3937–3946.

[10] 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*, 2017, pp. 3855–3863.

[11] 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*, 2017, pp. 1357–1366.

[12] H. Zhang and V. M. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 695–704.

[13] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha, "Recurrent squeeze-and-excitation context aggregation net for single image deraining," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 254–269.

[14] Z. Fan, H. Wu, X. Fu, Y. Hunag, and X. Ding, "Residual-guide feature fusion network for single image deraining," *arXiv preprint arXiv:1804.07493*, 2018.

[15] P. Xiang, L. Wang, F. Wu, J. Cheng, and M. Zhou, "Single-image de-raining with feature-supervised generative adversarial network," *IEEE Signal Processing Letters*, vol. 26, no. 5, pp. 650–654, 2019.

[16] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[17] D.-A. Huang, L.-W. Kang, M.-C. Yang, C.-W. Lin, and Y.-C. F. Wang, "Context-aware single image rain removal," in *2012 IEEE International Conference on Multimedia and Expo*. IEEE, 2012, pp. 164–169.

[18] Y. Luo, Y. Xu, and H. Ji, "Removing rain from a single image via discriminative sparse coding," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3397–3405.

[19] X.-J. Mao, C. Shen, and Y.-B. Yang, "Image denoising using very deep fully convolutional encoder-decoder networks with symmetric skip connections," *arXiv preprint arXiv:1603.09056*, vol. 2, 2016.

[20] H. Zhang and V. M. Patel, "Convolutional sparse coding-based image decomposition." in *BMVC*, 2016.

[21] Y.-L. Chen and C.-T. 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.

[22] L. Zhu, C.-W. Fu, D. Lischinski, and P.-A. Heng, "Joint bi-layer optimization for single-image rain streak removal," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2526–2534.

[23] H. Zhang and V. M. Patel, "Convolutional sparse and low-rank coding-based rain streak removal," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 1259–1267.

[24] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1395–1403.

[25] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[27] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. 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, pp. 4681–4690.

[28] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. 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.

[29] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, and R. W. Lau, "Spatial attentive single-image deraining with a high quality real rain dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 270–12 279.

[30] G. Wang, C. Sun, and A. Sowmya, "Erl-net: Entangled representation learning for single image de-raining," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5644–5652.

[31] Y. Du, J. Xu, Q. Qiu, X. Zhen, and L. Zhang, "Variational image deraining," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 2406–2415.

[32] K. Jiang, Z. Wang, P. Yi, C. Chen, B. Huang, Y. Luo, J. Ma, and J. Jiang, "Multi-scale progressive fusion network for single image deraining," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8346–8355.

[33] W. Wei, D. Meng, Q. Zhao, Z. Xu, and Y. Wu, "Semi-supervised transfer learning for image rain removal," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3877–3886.

[34] R. Yasarla, V. A. Sindagi, and V. M. Patel, "Syn2real transfer learning for image deraining using gaussian processes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2726–2736.

[35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[36] S. Xingjian, 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*, 2015, pp. 802–810.

[37] Y. Lan, H. Xia, H. Li, S. Song, and L. Wu, "Double recurrent dense network for single image deraining," *IEEE Access*, vol. 8, pp. 30 615–30 627, 2020.

[38] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[39] H. Zhang and V. M. Patel, "Densely connected pyramid dehazing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3194–3203.

[40] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*. Springer, 2016, pp. 694–711.

[41] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *IEEE transactions on circuits and systems for video technology*, 2019.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[43] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch. 2017," in *Long Beach, California, USA: Autodiff Workshop*, 2016.

[44] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of psnr in image/video quality assessment," *Electronics letters*, vol. 44, no. 13, pp. 800–801, 2008.

[45] W. Yang, R. T. Tan, J. Feng, Z. Guo, S. Yan, and J. Liu, "Joint rain detection and removal from a single image with contextualized deep networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 6, pp. 1377–1393, 2019.

[46] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on computational imaging*, vol. 3, no. 1, pp. 47–57, 2016.

[47] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2018, pp. 1451–1460.

[48] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.