Joint Self-Attention and Scale-Aggregation for Self-Calibrated Deraining Network

Cong Wang*
Dalian University of Technology
supercong94@gmail.com

Yutong Wu*
Dalian University of Technology
ytongwu@mail.dlut.edu.cn

Zhixun Su†
Dalian University of Technology
Key Laboratory for Computational Mathematics and Data Intelligence of Liaoning Province
zxsu@dlut.edu.cn

Junyang Chen
University of Macau
yb77403@umac.mo

ABSTRACT

In the field of multimedia, single image deraining is a basic pre-processing work, which can greatly improve the visual effect of subsequent high-level tasks in rainy conditions. In this paper, we propose an effective algorithm, called JDNet, to solve the single image deraining problem and conduct the segmentation and detection task for applications. Specifically, considering the important information on multi-scale features, we propose a Scale-Aggregation module to learn the features with different scales. Simultaneously, Self-Attention module is introduced to match or outperform their convolutional counterparts, which allows the feature aggregation to adapt to each channel. Furthermore, to improve the basic convolutional feature transformation process of Convolutional Neural Networks (CNNs), Self-Calibrated convolution is applied to build long-range spatial and inter-channel dependencies around each spatial location that explicitly expand fields-of-view of each convolutional layer through internal communications and hence enriches the output features. By designing the Scale-Aggregation and Self-Attention modules with Self-Calibrated convolution skillfully, the proposed model has better deraining results both on real-world and synthetic datasets. Extensive experiments are conducted to demonstrate the superiority of our method compared with state-of-the-art methods. The source code will be available at https://supercong94.wixsite.com/supercong94.

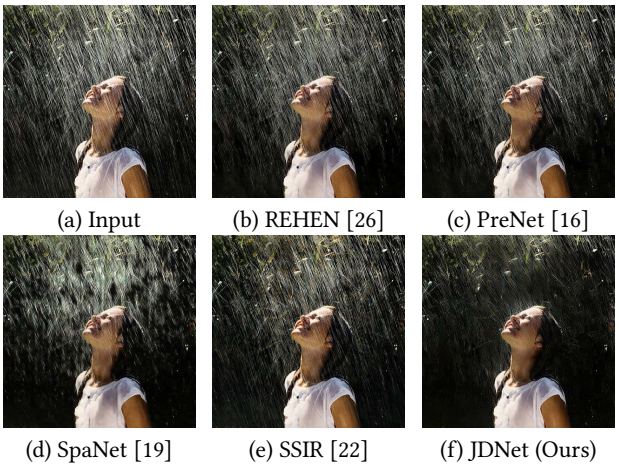
CCS CONCEPTS

• **Computing methodologies** → **Image processing**.

*Both authors contributed equally to this research.
†Corresponding author.

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413559



(a) Input (b) REHEN [26] (c) PreNet [16]
(d) SpaNet [19] (e) SSIR [22] (f) JDNet (Ours)

Figure 1: An example from real-world datasets.

KEYWORDS

Deraining; Self-Attention; Scale-Aggregation; Self-Calibrated Convolution

ACM Reference Format:
Cong Wang, Yutong Wu, Zhixun Su, and Junyang Chen. 2020. Joint Self-Attention and Scale-Aggregation for Self-Calibrated Deraining Network. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3413559

1 INTRODUCTION

As we all know, plenty of influence factors will greatly reduce the image quality, while affects the effect of subsequent image processing, such as Object Detection, Semantic Segmentation and Optical Character Recognition, etc. Rain is one of the common influence factors. How to restore a rain-free image from a given rainy image is a challenging problem. In this paper, we strive to solve the task by exploring a series of inner properties in the convolution layer.

We formulate the process of image deraining:

$$B = O - R, \tag{1}$$

where O represents the input rainy image, R is the rain streaks layer, and B is the deraining image, which is generally called the background layer. The restoration from a single rainy image has always been regarded as an ill-posed problem that there is numerous solution for a given rainy image. How to regulate the solution space to get stable and sole rain-free image becomes a core problem for the image deraining task.

In the early period, the deraining task was generally regarded as an optimization problem based on the prior information of images. [25] classifies mainly into two categories in prior-based methods, 1) employing Sparse Coding and 2) establishing Gaussian mixture models. The Sparse Coding-based methods convert the processing of the image into the processing of the signal and decompose the image into a low-frequency component and a high-frequency component. Based on this frequency decomposition process, Kang et al. [9] remove the rain component in rainy images using dictionary learning, Luo et al. [15] learn the dictionary of rain streak and background layers via Discriminative Sparse Coding, while others [13] apply Gaussian mixture models to model rain and background layers. These prior-based methods achieve better deraining performance to some extent under certain conditions by the given assumptions, but they either smooth out the edge details as rain streaks, or cannot handle large and dense rainy images. Moreover, the prior-based methods can generally be regarded as optimizing the cost function, which has a high time consumption.

With the rapid development of deep learning in recent years, many Convolutional Neural Networks (CNNs)-based methods for single image deraining have been proposed, gradually replaced by prior-based methods, such as [4, 7, 12, 24, 26, 27]. These methods employ deep networks to automatically extract features of layers, enabling them to model more complex mappings from rainy images to clean images, such as learning the binary rain streaks map [24]. Many subsequent methods enhance the deraining effects from the aspects of network structure complexity and image priors. For example, [12] and [7] combine multi-stage recurrent network and depth information with single image deraining, respectively. However, most of them rely on the traditional convolution pattern which achieves the network step by step by stacking several convolution layers that attach stationary weights to specific locations so that these layers do not learn features from different locations, leading to information drop-out. Furthermore, these models only consider the network design to apply to the deraining task while ignoring the inner structure of CNNs that lead to these methods is not robust to real-world rainy conditions. As shown in Fig. 1, other state-of-the-art methods fail to restore rain-free images, while our approach is able to remove most rain streaks and gets a clearer rain-free image.

To solve the above problems, we propose an effective deraining algorithm, called JDNet, from three aspects: 1) Pairwise Self-Attention module, 2) Scale-Aggregation module and 3) Self-Calibrated convolution. First of all, considering that the aggregation of information from a neighborhood cannot adapt to its content which will lose much important information in the traditional convolutional layer, we introduce a Pairwise Self-Attention module by paying attention to learn different locations at convolution. The introduced Pairwise Self-Attention module does not attach stationary weights to specific locations and is invariant to permutation and cardinality. In particular, weight computations of the Self-Attention module do not collapse the channel dimension and allows the feature aggregation to adapt to each channel. Secondly, we design a Scale-Aggregation module to learn the features from different scales. The proposed Scale-Aggregation module not only converts the convolution layer into deeper features, but also can maintain the original features from shallower ones. By designing the inner fusion between shallower and deeper layers, the proposed module can learn adaptively the features which part is more effective for rain removal. Thirdly, to improve the basic convolutional feature transformation process of CNNs, we bring in Self-Calibrated convolution to build long-range spatial and inter-channel dependencies around each spatial location that explicitly expand fields-of-view of each convolutional layer through internal communications and hence enriches the output features, which can help CNNs generate more discriminative representations by explicitly incorporating richer information. By devising jointly the there parts in the convolution layer, the proposed method has better deraining performance that can be able to remove heavy rain streaks and preserve better details.

2 RELATED WORK

2.1 Single Image Deraining

In order to generate accurate deraining results, researchers have made many trials, which can be simply divided into image prior-based methods and CNNs-based methods.

These image prior-based methods need to represent the characteristics of the rain-free image while maintaining consistency with the input image content. Kang et al. [9] employ the bilateral filter to decompose high-frequency information and low-frequency information from a rainy image. Sparse coding and dictionary learning are used to remove rain components in a rainy image. This work successfully eliminates the sparse light rain streaks. However, due to the extreme dependence on the preprocessing of the bilateral filter, it will produce blurred background details. In order not to confuse the rain line layer and the background layer, Luo et al. [15] introduce the mutual exclusivity property into discriminative sparse coding, and finally obtain deraining results which retain the clean texture details. In addition, [13] proposes Gaussian mixture models to model the rain layers and the background layers. These priors have a good effect on the removal of rain streaks with multiple directions and scales. Moreover, Zhu et al. [32] construct an iterative process to remove more rain. Although many prior-based methods have tried and improved for single image deraining, there are two common limits in these methods. On the one hand, the removal of large and dense rain streaks is not enough. On the other hand, the test time is too long.

Deep learning has an epoch-making significance in the field of image processing, which can simultaneously improve the speed of operations and the quality of completions. The CNNs-based methods generally use artificial means to generate a large number of paired datasets for training. Yang et al. [24] jointly perform a recurrent network of detection and rain removal, where the binary map is used in the detection process. In recent years, researchers have improved the network from different perspectives. In terms of models, Fu et al. [4] and Li et al. [12] introduce ResNet and SE-block

in their deraining networks, respectively. In terms of processes, Ren et al. [16] choose to implement simple networks in multiple stages instead of designing complex models. In particular, Wang et al. [19] design a deraining network based on the spatial attention module to pay special attention to the area where the rain is located.

2.2 Attention Mechanisms

In recent years, some methods have tried to add non-local modules [1, 20] or attention mechanisms [5, 6, 23, 33] to complex networks, which will establish the dependency of spatial location or channels or both. Since [17], Self-Attention module has been widely paid attention while it becomes a research hotspot. They propose Google Neural Machine Translation, which ignores the distance between words and directly calculates the dependency relationship. Later, Self-Attention is also used in computer vision tasks as a complement to convolution. In general channel-wise attention methods [5, 6, 18], attention weights reweight the activation in different channels. In particular, other methods [2, 3, 23] perform weighting operations in different contents and channels. Moreover, [14] discusses the Self-Calibrated convolution that considers more efficiently exploiting the convolutional filters in convolutional layers and designing powerful feature transformations to generate more expressive feature representations.

3 PROPOSED METHODS (JDNET)

3.1 Overview

In this section, we detail the proposed joint network for single image deraining, called JDNet. To take full advantage of features with different levels, dense connections are utilized to connect several joint units that are composed of Self-Attention module, Scale-Aggregation module and Self-Calibrated convolution.

Specifically, an input rainy image first passes through a convolution layer following an activation function to transform the channel dimension from image to feature. And then the transformed features are input into several joint units further extract the rain streaks information. At last, we obtain the rain streaks by a convolution layer following an activation function to transform the features to image. The detailed structure is shown in Fig. 2.

3.2 Self-Attention Module

In CNNs for image processing, each layer simultaneously realizes two functions, i.e., feature aggregation and feature transformation. The former integrates the features of all positions extracted by the kernel, and the latter performs transformation through linear mapping and nonlinear scalar functions. Since these two functions can be decoupled, if the feature transformation is simply set as an element-level operation composed of linear mapping and nonlinear scalar functions, then only the design of feature aggregation will be considered next. In this paper, we introduce the Pairwise Self-Attention module [29] to establish feature aggregation. Consistent with general Self-Attention modules, the final result is expressed as a weighted sum of adaptive weights and features:

$$y_i = \sum_{j \in \mathcal{R}(i)} \alpha(x_i, x_j) \odot \beta(x_j), \tag{2}$$

where x_i and x_j are feature maps with indexes i and j, respectively. $\odot$ is the Hadamard product called aggregation with the local footprint $\mathcal{R}(i)$, which is a set of indexes that can be aggregated with x_i. Please note that the number of parameters in the Pairwise Self-Attention module will not be affected by the size of footprint. In order to utilize more surrounding pixels, we set the size of footprint to 7×7. After this aggregation, the result y_i can be obtained.

The vector $\beta(x_j)$ generated by the function $\beta(\cdot)$ will be aggregated with the adaptive vector $\alpha(x_i, x_j)$ introduced later. Compared with ordinary weights, adaptive vector $\alpha(x_i, x_j)$ has strong content adaptability. It can be decomposed as follows:

$$\alpha(x_i, x_j) = \gamma(\delta(x_i, x_j)), \tag{3}$$

where $\delta(\cdot)$ and $\gamma(\cdot)$ respectively represent a relation function and a hybrid map composed of linear and nonlinear functions. Here, we use LeakyReLU as the previously mentioned nonlinear function. Based on the relation $\delta(\cdot)$, the function $\gamma(\cdot)$ is used to obtain a vector result, which can be combined with $\beta(x_j)$ in Eq. 2. In general, matching the output dimension of $\gamma(\cdot)$ with the dimension of $\beta(\cdot)$ is not a necessary thing because attention weights can be shared among a group of channels.

We choose the subtraction as the relation function, which can be formulated:

$$\delta(x_i, x_j) = \varphi(x_i) - \psi(x_j), \tag{4}$$

where $\varphi(\cdot)$ and $\psi(\cdot)$ are convolution operations while have matching output dimensions. $\delta(\cdot)$ calculates spatial attention for each channel instead of sharing between channels.

This Pairwise Self-Attention module we introduced is shown in Fig 3. In order to perform a more efficient process, these two branches through which the input feature passes reduce the dimensionality of channels appropriately.

The first branch is called the attention branch, which further extracts features through two convolutional layers of $\varphi(\cdot)$ and $\psi(\cdot)$, and then the relation $\delta(\cdot)$ and the map $\gamma(\cdot)$ are used successively to obtain the final attention weight α. The second branch employs the convolution operation $\beta(\cdot)$ to get features of reducing the channel dimension. After that, we use the Hadamard product to aggregate the results of the two branches and input them into Batch Normalization, LeakyReLU function and convolution operation. The last convolution operation expands the channel dimension of feature result back to the input channel and adds it to the original input feature.

Conventional convolution uses fixed kernels for feature aggregation. The kernel weights do not change with contents of input images, but change across the channels. The above Self-Attention module uses novel vector attention, which can generate content adaptation ability while maintaining the channel adaptation ability. This makes our deraining model have strong adaptability, which can effectively remove rain streaks when the distribution of rain is different from training datasets.

3.3 Scale-Aggregation Module

Extracting features of different scales is a means to improve the performance of vision tasks. In this paper, we design a Scale-Aggregation module to learn the features from different scales. The proposed Scale-Aggregation module not only converts the convolution into

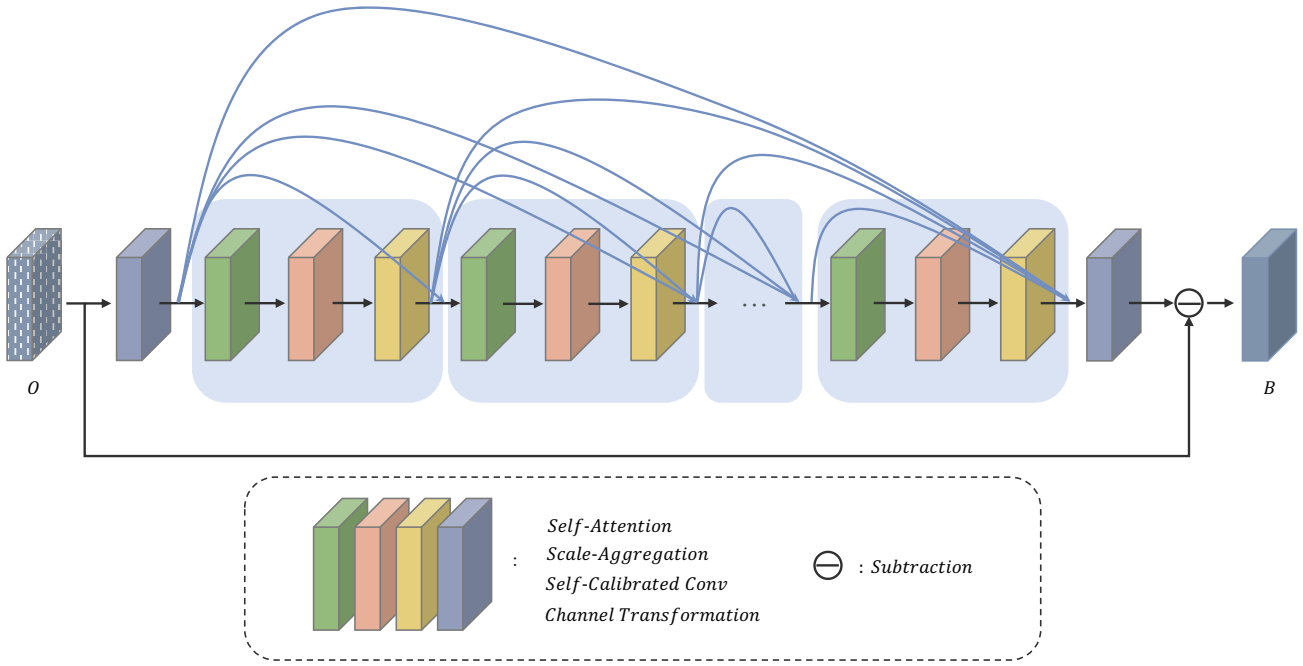


Figure 2: The architecture of Joint Network for deraining (JDNet). Each joint unit is composed of a Self-Attention module, Scale-Aggregation module and Self-Calibrated convolution. We use dense connections to inner-connect several joint units. And at the first layer and last layer, we use 3×3 convolution following a LeakyReLU to change the channel dimensions.

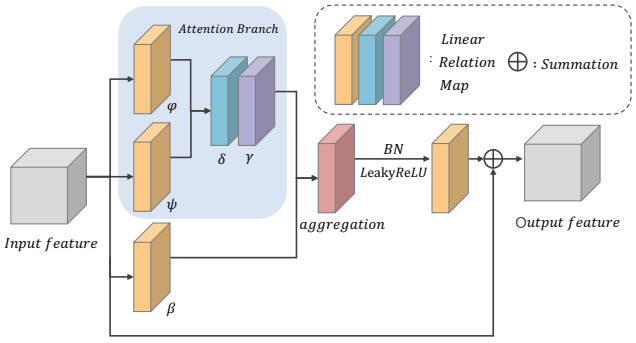


Figure 3: The architecture of Pairwise Self-Attention Module.

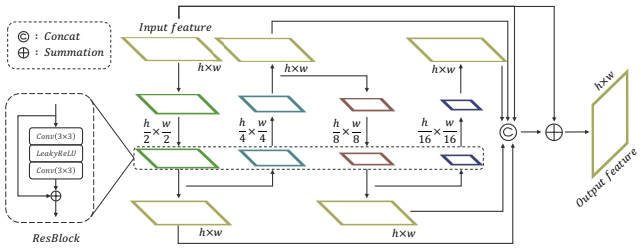


Figure 4: Scale-Aggregation Module.

deeper features, but also can maintain the original features from shallower ones. By designing the inner fusion between shallower and deeper layers, the proposed module can learn adaptively the features which part is more effective for rain removal. The structure of the Scale-Aggregation module which we propose is shown in Fig 4. At the beginning of this module, for a given input X, the feature map with a downsampling rate of 2 can be obtained by the following two formulas:

$$X^1 = LeakyReLU(Conv_3^2(X)), \tag{5}$$

$$X^1 = ResBlock_1(X^1), \tag{6}$$

where $Conv_i^j(\cdot)$ represents i×i convolution operation with stride j, $LeakyReLU(\cdot)$ represents an activation function with the parameter of 0.2, $ResBlock(\cdot)$ consists of an activation function between two 3×3 convolution layers. Let $X^0 = X$, the general formulas of the above process are:

$$X^i = LeakyReLU(Conv_3^2(X^{i-1})), \tag{7}$$

$$X^i = ResBlock_i(X^i), \tag{8}$$

where $i = 1, 2, \ldots, n$. Eventually, we upsample each X^i to the scale of original input and concatenate these results with the input features following a 1×1 convolution:

$$X^{out} = Conv_1^1\{Concat[X^0, Up(X^1), Up(X^2), \ldots, Up(X^n)]\}, \tag{9}$$

where $Up(\cdot)$ is an interpolation operation, $Concat(\cdot)$ represents the concatenation operation. As a result, we get the feature map restored to the original shape containing feature information of different scales.

3.4 Self-Calibrated Convolution

The structure of deep CNNs is becoming more and more complicated, which can enhance the learning ability of the network. The novel convolution called Self-Calibrated convolution [14] that we introduce considers improving the feature transformation process in convolution, as Fig 5.

A given group of filter sets K with the shape (C, C, k_h, k_w) is divided into four parts, i.e., $[K_1, K_2, K_3, K_4]$, where k_h and k_w are the spatial height and width, respectively. Each part, whose shape is $(C/2, C/2, k_h, k_w)$, is responsible for performing different functions. After splitting filters, the input X with the channel C is split into X_1 and X_2 through 1×1 convolution with the channel $C/2$.

In Self-Calibrated convolution, we perform feature transform at two scales: the original scale and the smaller scale after downsampling. For a given X, we adopt average pooling to reduce the scale:

$$T_1 = AvgPool_r(X_1), \tag{10}$$

where r is the downsampling rate and stride of the pooling process. Benefiting from the downsampling operation, the receptive field at each spatial location can be effectively expanded. Next, T_1 can be used as an input to the filter K_2 following the upsample operation which restores the feature back to the original scale, resulting in:

$$X_1^{'} = Up(\mathcal{F}_2(T_1)) = Up(T_1 * K_2), \tag{11}$$

where $\mathcal{F}_2(T_1) = T_1 * K_2$ is a simplified form of convolution. Then, the calibrated operation can be formulated as:

$$Y_1^{'} = \mathcal{F}_3(X_1) \odot Sigmoid(X_1 + X_1^{'}), \tag{12}$$

where $\mathcal{F}_3(X_1) = X_1 * K_3$, $Sigmoid(\cdot)$ is an activation function. The final result of the calibrated branch is calculated:

$$Y_1 = \mathcal{F}_4(Y_1^{'}), \tag{13}$$

where $\mathcal{F}_4(Y_1^{'}) = Y_1^{'} * K_4$.

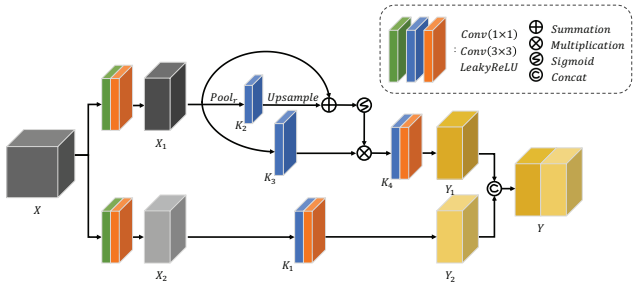


Figure 5: Self-Calibrated Convolution.

The other half of result can be obtained from another branch that does not require scale transformation. The formula is as follows:

$$Y_2 = \mathcal{F}_1(X_2) = X_2 * K_1. \tag{14}$$

Finally, we concatenate Y_1 and Y_2 in order to get the final result Y. Reviewing the entire Self-Calibrated convolution, it enables each spatial position to adaptively encode the context from a long-range region, which is also a huge difference between it and traditional convolution.

3.5 Loss Fuction

SSIM(Structural Similarity) [21] is an image quality evaluation metric in the range of $[0, 1]$. It measures the similarity of two images from brightness, contrast, and structure. When the two images are identical, SSIM is 1. The effectiveness of negative SSIM loss for image deraining has been confirmed in [16].

Hence, we use negative $SSIM$ loss as the loss function:

$$\mathcal{L} = -SSIM(\tilde{B}, B), \tag{15}$$

where $\tilde{B}$ and B are the deraining result and corresponding ground-truth, respectively.

3.6 Training Details

We use the PyTorch framework to train and test the proposed method. We trained the network for 1000 epochs with 32 joint units. Each pair of training samples will be randomly cropped to 64×64 pixels. Adam optimizer [10] is used with a learning rate of 5×10^{-4} which is divided by 10 after the $600th$ epoch and the $800th$ epoch. Both n in the Scale-Aggregation module and r in the Self-Calibrated convolution are set to 4, while the channel dimensionality of the entire network is 32. We train these networks on a PC with two NVIDIA GTX 1080Ti GPUs.

4 EXPERIMENTAL RESULTS

In this section, we will conduct training and corresponding tests on the synthetic datasets Rain100H[24], Rain100L[24], respectively. Rain100H and Rain100L contain 1800 pairs of training images and 200 pairs of testing images. Rain100H represents heavy rain and Rain100L represents light rain. The model trained under the Rain100H dataset is used to test the real-world datasets[11, 24, 28] and Rain12 [13].

In order to clearly show the superiority of our proposed method in terms of quantity and quality, we compare it with the state-of-the-art single image deraining methods published in ACM'MM[26], CVPR[4, 16, 19, 22], and ECCV[12] in the last three years.

4.1 Results and Analysis

Quantitative Comparison. We compare our proposed method with DDN [4], RESCAN [12], REHEN [26], PreNet [16], SpaNet [19], and SSIR [22] under the two metrics of SSIM [21] and PSNR [8].

We train our models on the synthetic datasets Rain100H and Rain100L, and compare the quantitative results obtained with the training methods under the corresponding dataset, respectively. The metric results of our and compared deraining methods are shown in Tab. 1. It can be clearly seen that the proposed method has achieved the highest SSIM and PSNR in all datasets. Compared with other state-of-the-art methods, our approach has a big improvement.

Qualitative Comparison. In Fig. 6, we show some synthetic examples on the Rain100H dataset. We can see that the proposed model can restore cleaner and clearer results, while other approaches

Table 1: Quantitative experiments evaluated on three synthetic datasets. The best results are highlighted in boldface.

	DDN		RESCAN		REHEN		PreNet		SpaNet		SSIR		Ours	
Dataset	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Rain100H	22.26	0.69	25.92	0.84	27.52	0.86	27.89	0.89	26.54	0.90	22.47	0.71	**30.02**	**0.92**
Rain100L	34.85	0.95	36.12	0.97	37.91	0.98	36.69	0.98	36.20	0.98	32.37	0.92	**38.65**	**0.99**
Rain12	28.66	0.91	33.75	0.95	35.84	0.96	34.77	0.96	33.59	0.96	24.14	0.78	**37.02**	**0.97**

(a) Input (b) DDN (c) RESCAN (d) REHEN (e) PreNet (f) SpaNet (g) SSIR (h) JDNet (i) GT

Figure 6: Examples about the comparison of our method with other methods on Rain100H dataset.

also hand down some artifacts or remaining rain streaks. Especially, SSIR [22] fails to work on synthetic datasets.

Furthermore, we provide more examples of real-world datasets to illustrate the superiority of the proposed method. Fig. 7 shows the results and we can observe that the proposed method is able to remove all of rain streaks and can preserve better details. However, other state-of-the-art methods hand down lots of rain streaks so that they fail to restore clear and clean rain-free images. Especially, SSIR [22] always fails to work and even leaves behind traces of rain streaks.

To sum up, our proposed is a more robust deraining method that not only can adapt various rainy conditions that can remove most of rain streaks, but also can better preserve image details and texture information, which benefits from our designed three learned parts: Self-Attention module, Scale-Aggregation module and Self-Calibrated convolution. This also illustrates our method has better deraining power.

4.2 Ablation Study

In this section, we analyze the importance of different modules. Especially, we utilize the Scale-Aggregation as the baseline module. And for different modules, their abbreviations are as follows:

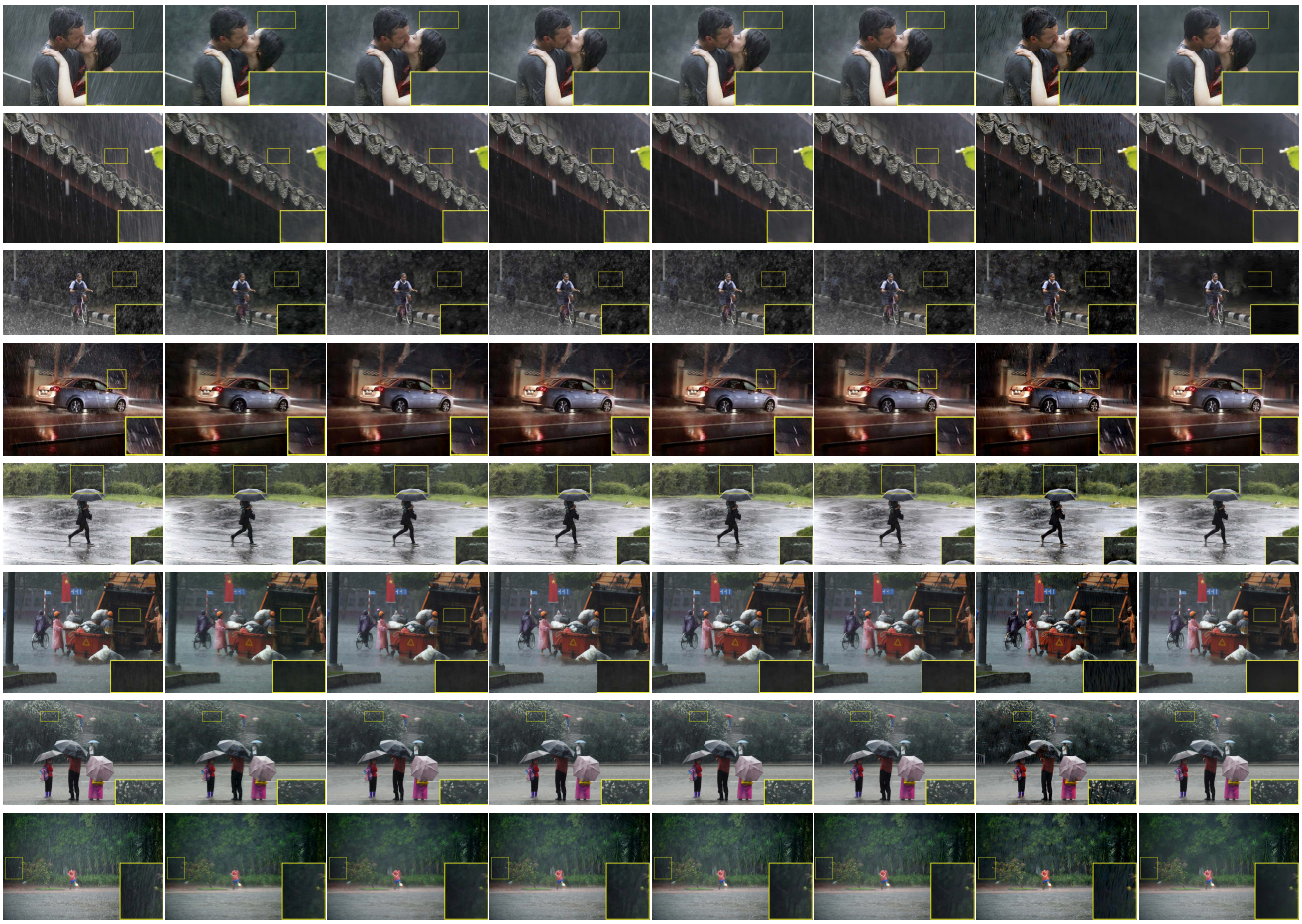
- R_1 : The Scale-Aggregation module as the baseline without Self-Attention module and Self-Calibrated convolution.
- R_2 : Baseline with Self-Calibrated convolution.
- R_3 : Baseline with Self-Attention module and Self-Calibrated convolution, i.e., our proposed final joint unit.

The results of different modules are shown in Tab. 2. We can see that the Self-Calibrated convolution can improve the deraining results compared with the baseline module, and the Self-Attention module further boosts the deraining performance compared with the baseline module with Self-Calibrated convolution. Compared with the baseline module, the introduced Self-Calibrated convolution and Self-Attention module improve the SSIM about 0.01.

Table 2: The results of different modules on Rain100H. The best results are highlighted in boldface.

Modules	SC conv	Self-Attention	SSIM	PSNR
R_1			0.9130	29.3357
R_2	√		0.9219	**30.0307**
R_3	√	√	**0.9221**	30.0160

In Fig. 8 we provide two deraining examples with the three modules on real-world datasets. We can observe that the self-attention, shown in Fig. 8 (d), plays an important role in the real-world deraining results. This phenomenon further demonstrates the the



(a) Input (b) DDN (c) RESCAN (d) REHEN (e) PreNet (f) SpaNet (g) SSIR (h) JDNet

Figure 7: Examples about the comparison of our method with other methods on real-world datasets.

importance of self-attention that has content adaptation ability to better help the network remove the rain streaks, especially on real-world datasets.

4.3 Discussion on Losses

In order to show the effectiveness of negative SSIM loss, we conduct experiments with MAE loss and MSE loss under the same conditions. As shown in Tab. 3, we can see that the negative SSIM loss has the best performance among the three losses. This illustrates that the negative SSIM is the most effective loss for the deraining task.

Table 3: The results of different losses on Rain100H.

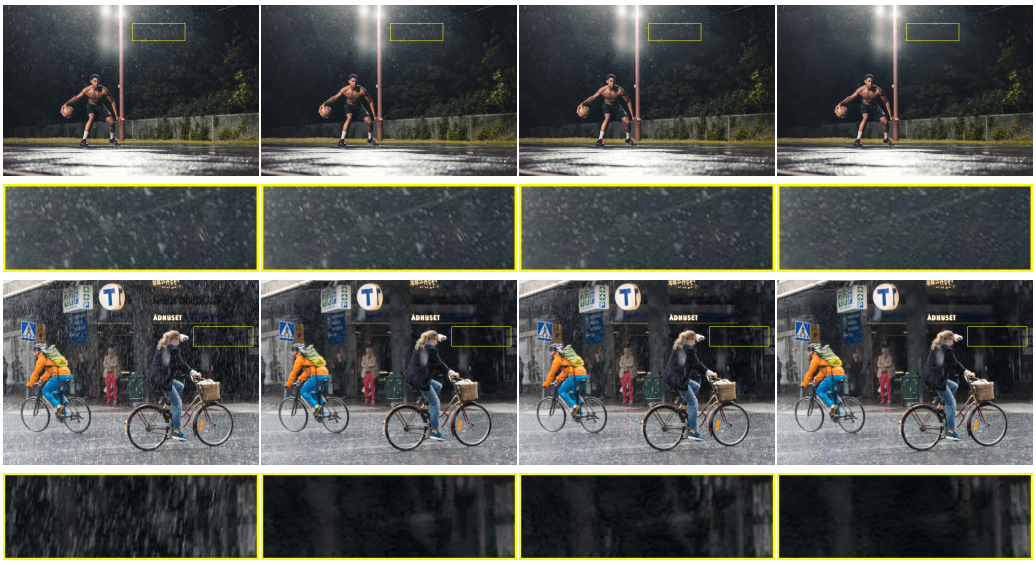
Loss	SSIM	PSNR
MAE	0.9002	29.2365
MSE	0.8896	28.7797
Negative SSIM	**0.9221**	**30.0160**

4.4 Application for High-level Tasks

In most conditions, the deraining serves as the preprocessing for some high-level tasks, e.g., segmentation and detection. In this section, we provide some visual examples for these applications. Fig. 9 shows the application of different deraining methods for semantic segmentation on a real-world image, which is directly tested using PSPNet [30] under the ADE20K [31] dataset. We can see that the proposed method has a better deraining performance and so makes PSPNet have a better segmentation result, while other methods can not generate satisfactory label prediction.

Fig. 10 shows the results of different methods in object detection on real-world datasets, where we use open source called Google Object Detection API on the Tensorflow framework. We can observe that the proposed method generates more accurate predictions, while other methods can not restore clearer rain-free images that lead to they fail to detect other existing objects.

The above two examples illustrate that our method can provide better preprocessing for other high-level tasks as a better derainer.



(a) input (b) R_1 (c) R_2 (d) R_3

Figure 8: The results of different modules from real-world datasets.

(a) Input (b) DDN (c) RESCAN (d) REHEN (e) PreNet (f) SpaNet (g) SSIR (h) JDNet

Figure 9: Semantic segmentation from real-world datasets.

(a) Input (b) DDN (c) RESCAN (d) REHEN (e) PreNet (f) SpaNet (g) SSIR (h) JDNet

Figure 10: Object detection from real-world datasets.

5 CONCLUSION

In this paper, we propose an effective deraining approach for single image deraining. The proposed model consists of three parts, including the Self-Attention module, Scale-Aggregation module and Self-Calibrated convolution. Each part of these modules can boost to generate clearer and cleaner rain-free images so that the overall network has better deraining performance. By exploring the inner correlation between different positions of convolution layers, the model is more robust to real-world rainy conditions. Extensive analysis and discussion illustrate the superiority of the proposed method compared with state-of-the-art approaches both on synthetic and real-world datasets.

ACKNOWLEDGEMENT

This work was supported by the Natural Science Foundation of China [grant numbers 61976041].

REFERENCES

[1] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. 2019. GCNet: Non-Local Networks Meet Squeeze-Excitation Networks and Beyond. In *IEEE International Conference on Computer Vision*. 1971–1980.

[2] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. 2017. SCA-CNN: Spatial and Channel-Wise Attention in Convolutional Networks for Image Captioning. In *IEEE Conference on Computer Vision and Pattern Recognition*. 6298–6306.

[3] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. 2019. Dual Attention Network for Scene Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 3146–3154.

[4] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. 2017. Removing Rain from Single Images via a Deep Detail Network. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1715–1723.

[5] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. 2018. Gather-Excite: Exploiting Feature Context in Convolutional Neural Networks. In *Conference on Neural Information Processing Systems*. 9423–9433.

[6] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-Excitation Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*. 7132–7141.

[7] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. 2019. Depth-Attentional Features for Single-Image Rain Removal. In *IEEE Conference on Computer Vision and Pattern Recognition*. 8022–8031.

[8] Q. Huynh-Thu and M. Ghanbari. 2008. Scope of validity of PSNR in image/video quality assessment. *Electronics Letters*. 44, 13 (2008), 800–801.

[9] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu. 2012. Automatic Single-Image-Based Rain Streaks Removal via Image Decomposition. *IEEE Trans. Image Process.* 21, 4 (2012), 1742–1755.

[10] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

[11] 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. 2019. Single Image Deraining: A Comprehensive Benchmark Analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*. 3838–3847.

[12] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. 2018. Recurrent Squeeze-and-Excitation Context Aggregation Net for Single Image Deraining. In *European Conference on Computer Vision*. 262–277.

[13] Yu Li, Robby T. Tan, Xiaojie Guo, Jiangbo Lu, and Michael S. Brown. 2016. Rain Streak Removal Using Layer Priors. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2736–2744.

[14] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Changhu Wang, and Jiashi Feng. 2020. Improving Convolutional Networks with Self-Calibrated Convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[15] Yu Luo, Yong Xu, and Hui Ji. 2015. Removing Rain from a Single Image via Discriminative Sparse Coding. In *IEEE International Conference on Computer Vision*. 3397–3405.

[16] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. 2019. Progressive Image Deraining Networks: A Better and Simpler Baseline. In *IEEE Conference on Computer Vision and Pattern Recognition*. 3937–3946.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Conference on Neural Information Processing Systems*. 5998–6008.

[18] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. 2017. Residual Attention Network for Image Classification. In *IEEE Conference on Computer Vision and Pattern Recognition*. 6450–6458.

[19] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson W. H. Lau. 2019. Spatial Attentive Single-Image Deraining With a High Quality Real Rain Dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*. 12270–12279.

[20] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-Local Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*. 7794–7803.

[21] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* 13, 4 (2004), 600–612.

[22] Wei Wei, Deyu Meng, Qian Zhao, Zongben Xu, and Ying Wu. 2019. Semi-Supervised Transfer Learning for Image Rain Removal. In *IEEE Conference on Computer Vision and Pattern Recognition*. 3877–3886.

[23] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. 2018. CBAM: Convolutional Block Attention Module. In *European Conference on Computer Vision*. 3–19.

[24] Wenhan Yang, Robby T. Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. 2017. Deep Joint Rain Detection and Removal from a Single Image. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1685–1694.

[25] Wenhan Yang, Robby T. Tan, Shiqi Wang, Yuming Fang, and Jiaying Liu. 2020. Single Image Deraining: From Model-Based to Data-Driven and Beyond. *IEEE Trans. Pattern Anal. Mach. Intell.* (2020).

[26] Youzhao Yang and Hong Lu. 2019. Single Image Deraining via Recurrent Hierarchy Enhancement Network. In *ACM International Conference on Multimedia*. 1814–1822.

[27] He Zhang and Vishal M. Patel. 2018. Density-Aware Single Image De-Raining Using a Multi-Stream Dense Network. In *IEEE Conference on Computer Vision and Pattern Recognition*. 695–704.

[28] He Zhang, Vishwanath Sindagi, and Vishal M. Patel. 2019. Image De-raining Using a Conditional Generative Adversarial Network. *IEEE Transactions on Circuits and Systems for Video Technology*. (2019).

[29] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. 2020. Exploring Self-Attention for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[30] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. Pyramid Scene Parsing Network. In *IEEE Conference on Computer Vision and Pattern Recognition*. 6230–6239.

[31] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2019. Semantic Understanding of Scenes Through the ADE20K Dataset. *Int. J. Comput. Vis.* 127, 3 (2019), 302–321.

[32] Lei Zhu, Chi-Wing Fu, Dani Lischinski, and Pheng-Ann Heng. 2017. Joint Bi-layer Optimization for Single-Image Rain Streak Removal. In *IEEE International Conference on Computer Vision*. 2545–2553.

[33] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. 2018. Learning Transferable Architectures for Scalable Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 8697–8710.