Single Image Deraining via Recurrent Residual Multiscale Networks

Yupei Zheng, Xin Yu, Miaomiao Liu, and Shunli Zhang*

***Abstract*—Existing deraining approaches represent rain streaks with different rain layers and then separate the layers from the background image. However, because of the complexity of real-world rain, such as various densities, shapes and directions of rain streaks, it is very difficult to decompose a rain image into clean background and rain layers. In this paper, we develop a novel single image deraining method based on residual multiscale pyramid to mitigate the difficulty of rain image decomposition. To be specific, we progressively remove rain streaks in a coarse-to-fine fashion, where heavy rain is first removed in coarse-resolution levels and then light rain is eliminated in fine-resolution levels. Furthermore, based on the observation that residuals between a restored image and its corresponding rain image give critical clues of rain streaks, we regard the residuals as an attention map to remove rains in the consecutive finer-level image. To achieve a powerful yet compact deraining framework, we construct our network by recurrent layers and remove rain with the same network in different pyramid levels. In addition, we design a multiscale kernel selection network (MSKSN) to facilitate our single network to remove rain streaks at different levels. In this manner, we reduce 81% of the model parameters without decreasing deraining performance compared to our prior work [1]. Extensive experimental results on widely-used benchmarks show that our approach achieves superior deraining performance compared to the state-of-the-art.**

***Index Terms*—Single image deraining, multiscale, residual attention, kernel selection**

I. Introduction

RAIN streaks often result in image quality degradation, such as blurriness and heavy occlusion, as shown in Fig. 1. They also severely deteriorate the performance of many outdoor surveillance systems, such as object detection [2] and tracking [3]. Therefore, removal of rain streaks from a single image is important for various outdoor vision-based applications. Previous single-image based deraining methods [4–9] often remove rain via discriminative representation learning of rain streaks and background details. However, these methods may fail to deal with real-world rain due to its complexity such as overlappings between rain streaks and complex background texture.

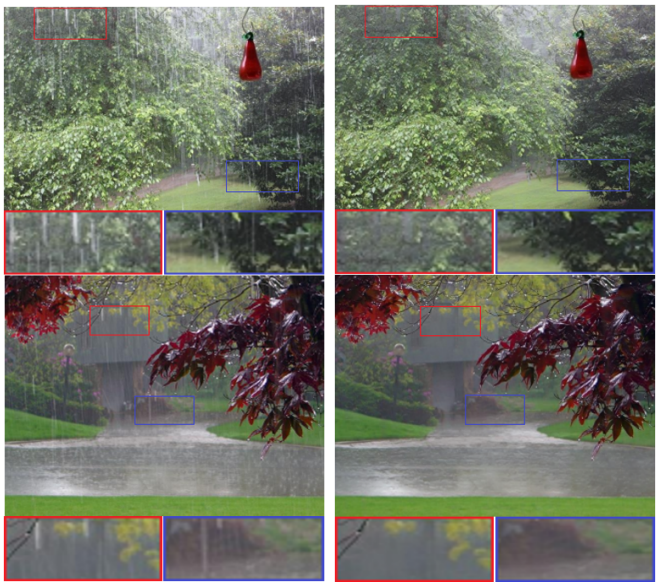
Yupei Zheng and Shunli Zhang are with the School of Software Engineering, Beijing Jiaotong University, Haidian, 100044, Beijing, China. e-mail: {ypzhengx, slzhang}@bjtu.edu.cn.

Xin Yu is with the Faculty of Engineering and IT, School of Computer Science, University of Technology Sydney, Ultimo 2007. e-mail:{xin.yu}@uts.edu.au.

Miaomiao Liu is with the College of Engineering and Computer Science, Australian National University, Canberra, ACT 2601. e-mail:{miaomiao.liu}@anu.edu.au.

* Shunli Zhang is the corresponding author.

* Our code is available at https://github.com/Zhengyupei/MSKSN



(a) (b)

Fig. 1: Left: Two real-world images degraded by rain streaks. Right: Our approach removes rain streaks and significantly improves visibility. Zoom in for a better look at the restoration quality.

Thanks to the powerful feature representation ability of convolutional neural networks (CNN), deep learning based deraining methods [10–16] have been proposed and achieved promising performance. Those methods usually aim to decompose rain into several layers which contain different types (e.g. directions and shapes) of rain streaks. Thus, those approaches mainly focus on learning representations of different rain layers and then remove them from a rain image. However, previous works have the following shortcomings:

(1) In real-world scenarios, it is often very challenging to ascertain the accurate number of the rain layers accounting for different types of rain streaks, (*e.g.*, different shapes, directions, and densities). Inaccurate estimation of the number of layers will result in removing rain streaks incompletely or excessively.

(2) Rain streaks with different densities might be in the same image and different rain layers always overlap with each other. It is suggested by [15] that different models should be used to handle different densities of rain (*i.e.*, heavy, medium, and light rain streaks). However, building a model which can accurately identify the rain densities is difficult. Thus, previous methods might inadequately

remove rain streaks or excessively smooth background details.

To tackle the above issues, this paper proposes a novel deraining method based on multiscale pyramid. In particular, rain streaks are removed while background details are constructed in a coarse-to-fine fashion. In this way, we decompose the image into different levels of pyramid, and remove the heavy streaks first in the coarse level. As the resolutions in the pyramid increase, background details are gradually recovered while lighter rain is removed. In other words, the structure in the coarse level is recovered first and then fine details are recovered in higher resolutions. Therefore, the density of the rain is unnecessary to be explicitly estimated, avoiding inadequate- or over-removal of rain. Since the residual between the recovered and original rain images in fine level mainly contains the missing details and removed rain streaks, the residual can be regarded as an input attention map to provide cues for recovering the missing background texture. Based on the weights of the attention map, our network is able to identify the regions with rain and the densities of the rain streaks, which guides us to remove rain in a coarse-to-fine fasion. The proposed method can effectively reduce the influence of excessively or insufficiently removing rain from a single image.

In our previous work [1], we used different sub-network structure and residual based attention maps to remove rain streaks in each level of a pyramid. While being effective in handling various scales of rain, the network has a large number of parameters (∼6,426K in total) and might not be deployed easily in some practical applications. Moreover, those sub-networks need to be trained sequentially. In other words, when the network in the coarse level converges, we start to train the network in the finer level. In this paper, we propose a more compact yet powerful network (∼1,218K parameters). To facilitate removing rain at different pyramid levels (stages) while sharing the same network weights, we further design a multi-scale kernel selection block, which extracts rain layers at different scales adaptively. In this fashion, we achieve a lightweight deraining network while improving the rain removal performance.

All in all, our main contributions are summarized as follows:

(1) A novel deraining framework based on residual multi-scale is proposed to remove rain streaks from background images. To the best of our knowledge, this is the first attempt to tackle the deraining problem from coarse-to-fine perspective.
(2) The residual between the reconstructed image and original rain image is fully exploited and regarded as an attention map, which provides rain density and intensity information and assists rain streak recognition and background recovery.
(3) Thanks to our residual embedded pyramid framework, our method leaves out the procedure of estimating rain densities and then avoids the artifacts of incomplete or excessive rain removal caused by inaccurate estimation.
(4) Recurrent layers are designed to construct a light-weight network, and employ the same network structure to remove rain at each level of the pyramid. Thus, we significantly reduce the model parameters. Moreover, a multiscale kernel selection block is proposed to further boost the performance of the rain layer extraction for our single deraining network.

This work is an extension of our prior work [1]. In that work, following the multiscale pyramid framework, the deraining is decomposed into three stages, whose network structures are different due to the increasing resolutions. In the first stage, the deraining model is constructed by the autoencoder which consists of convolutional layers. In the second stage, the network architecture is implemented by residual blocks based autoencoder to recover more background details. In the third stage, dense blocks are used to extract more features and recover the finest details. Moreover, different loss functions, *i.e.*, an Euclidean loss function, a perceptual loss function, and a similarity loss function, are employed in different deraining stages separately. The deraining models in the different levels are trained stage by stage. In this paper, we have made the following substantial improvements in comparison to our previous version: (i) Considering the scales of rain streaks are different across the pyramid levels, we propose a multiscale kernel selection block to tackle different scales of rain. Our presented blocks improve the effectiveness of rain streak removal. (ii) We re-design our proposed network architecture to allow sharing parameters within each pyramid level via a recurrent mechanism. By incorporating the recurrent mechanism, we reduce *81%* of the network parameters without reducing the qualitative and quantitative performance. (iii) Different from our preliminary work, where three distinct models are trained stage by stage, we further improve our training efficiency by training our single model in an end-to-end manner. This is also a direct benefit from our new network architecture. (iv) The experimental results demonstrate our method achieves superior performance compared to state-of-the-art methods on popular widely-used benchmarks.

II. Related Work

In the past decades, many methods have been developed to remove rain streaks. According to the type of inputs, we group deraining methods into two categories: video-based and single-image rain streak removal methods.

A. Video based Deraining Methods

The methods removing rain streaks based on video use the spatio-temporal relations among frames to detect and remove rain streaks in a video. Garg and Nayar [17] develop a correlation model to capture rain dynamics and build a motion blur model based on physics, with which the rain in videos are removed. Chen *et al*. [18] propose a low-rank appearance model to capture the spatio-temporally correlated rain streaks from natural image frames. Jiang *et al*. [19] observe that rain drops are sparse and smooth along rain directions while rain-free video contents are smooth in the directions perpendicular to rain streaks. Taking these intrinsic characteristics into consideration, they design a video based rain-removal model. Wei *et al*. [20] encode rain streaks in a

stochastic manner and remove rain streaks by modeling rain streak layers, moving objects and static background in a video.

Ren *et al*. [21] divide rain streaks in a video into two categories: sparse and dense ones. They formulate the detection of moving objects and sparse rain as a multi-label Markov Random Fields, and assume dense rain follows a Gaussian distribution. Then, the rain is removed from videos by matrix decomposition. [22] takes super-pixels as basic processing units to remove rain and design a convolutional network to address blur artifacts. Liu *et al*. [23] build a deraining model by integrating rain classification, rain removal with spatial texture and background image recovery with temporal coherence. Liu *et al*. [24] design a hybrid rain model to depict rain streaks and occlusions, and use a recurrent mechanism in temporal dimension for feature fusion and residual learning for reconstruction. Although we also employ the residual recurrent network, we stress that it is quite different from [24]. In our method, the recurrent mechanism is used in spatial and feature domains while the residual is regarded as the attention map in the different deraining stages. Morevoer, state-of-the-art video-based deraining methods [25–28] highly rely on the temporal consistency of video contents, which does not exist in a single image.

B. Single Image based Deraining Methods

For the single image based deraining task, Kang and Fu [5] decompose a raw rain image into high-frequency and low-frequency layers, and then employ sparse coding to remove rain streaks in the high-frequency layer. Luo *et al*. [7] propose a dictionary learning based algorithm to restore a clean image from a rain image. Li *et al*. [6] propose Gaussian mixture models (GMMs) to model rain streaks and background separately, and their presented models are also able to address multiple orientations and scales of the rain streaks. Regarding rain streaks have obvious line patterns, Chang *et al*. [4] present a low-rank image decomposition framework for rain streaks removal. Zhu *et al*. [8] alternatingly remove rain layers and background details from an input rain image. Deng *et al*. [9] propose an efficient method to remove rain streaks from an input rain image by taking the intrinsic structural and directional knowledge of rain streaks into account.

Very recently, Fu *et al*. [10, 11] decompose an input image into a low-frequency layer and a high-frequency detail layer and then remove rain from the detail layer via a CNN. However, those methods may fail to remove heavy rain when thick rain streaks cannot be well separated from image details by a guided filter. Inspired by the decomposition idea, Fu *et al*. [29] exploit a Gaussian-Laplacian pyramid for rain removal in a single image. Unlike their method, this paper also exploits the residual deduced from a prior stage as an attention map to facilitate rain removal. Li *et al*. [30] train Siamese sub-networks to handle rain streaks in different scales following a recurrent way. Yang *et al*. [14] present a recurrent contextualized dilated network for joint rain detection and removal. Li *et al*. [12] learn the weight of each rain layers by the operation of squeeze-and-excitation context aggregation [31], and then remove rain based on the weighted rain layers. Zhang *et al*. [15] first classify the rain level of an input rain image, and then remove rain in accordance with the classification results. Wang *et al*. [13] design a two-round and four-directional RNN [32] to extract rain and contextual features, and a direction-aware attention mechanism to remove rain streaks. Hu *et al*. [33] remove rain streaks in an input image by designing a depth-guided attention mechanism. Yang *et al*. [34] use parallel convolutional layers with multiscale kernel sizes and squeeze-and-excitation aggregation to construct a recurrent enhancement network, progressively removing rain streaks from images.

Although above methods achieve good progress, there still exist some limitations. As those methods represent rain by learning different rain layers and then separate them from the background image, they often rely on the accuracy of rain layer estimation. However, rain layers may not represent rain accurately due to the intrinsic overlapping among different rain streaks. Moreover, rain streaks may exhibit various densities in an image. Thus, using a single-scale model may not effectively remove different densities of rain. Therefore, the performance improvements of prior methods tend to saturate.

III. Proposed Method

Our proposed deraining network is realized under a multiscale pyramid framework, inspired by [35]. The processing of each level of the pyramid is taken as a *stage*. We intend to remove rain streaks with different densities at different stages (see Fig. 2). In this way, heavy rain in low-resolution images can be removed more easily than that in original high-resolution versions, easing the procedure of rain removal from a single image. Besides, our network fully explores the residual between our restored rain-free image and its corresponding rain image. Specifically, the residuals provide the information about missing background details as well as the regions with rain, and thus allows our network to pay more attention to those regions. To achieve light-weight deraining network, we exploit recurrent layers in our network. Different from our previous conference version [1], we employ the same network structure containing recurrent layers to remove rain streaks at all the stages. Thus, our method obtains a compact model. We further design a multiscale kernel selection block (MKSB) to choose kernel sizes adaptively. MKSB allows us to apply a single network in different stages without decreasing the deraining performance. The rain image formulation and our proposed multiscale recurrent residual deraining network are presented as follows.

A. Rain Image Formulation

A rain image I is generally defined as

$$I = B + \sum_{i=1}^{N} R_i, \tag{1}$$

where B represents a clean background image, R_i indicates the i-th rain layer, and N denotes the number of rain layers. However, there often exists overlap among rain streaks in real-world scenes. It is difficult to separate the rain streaks

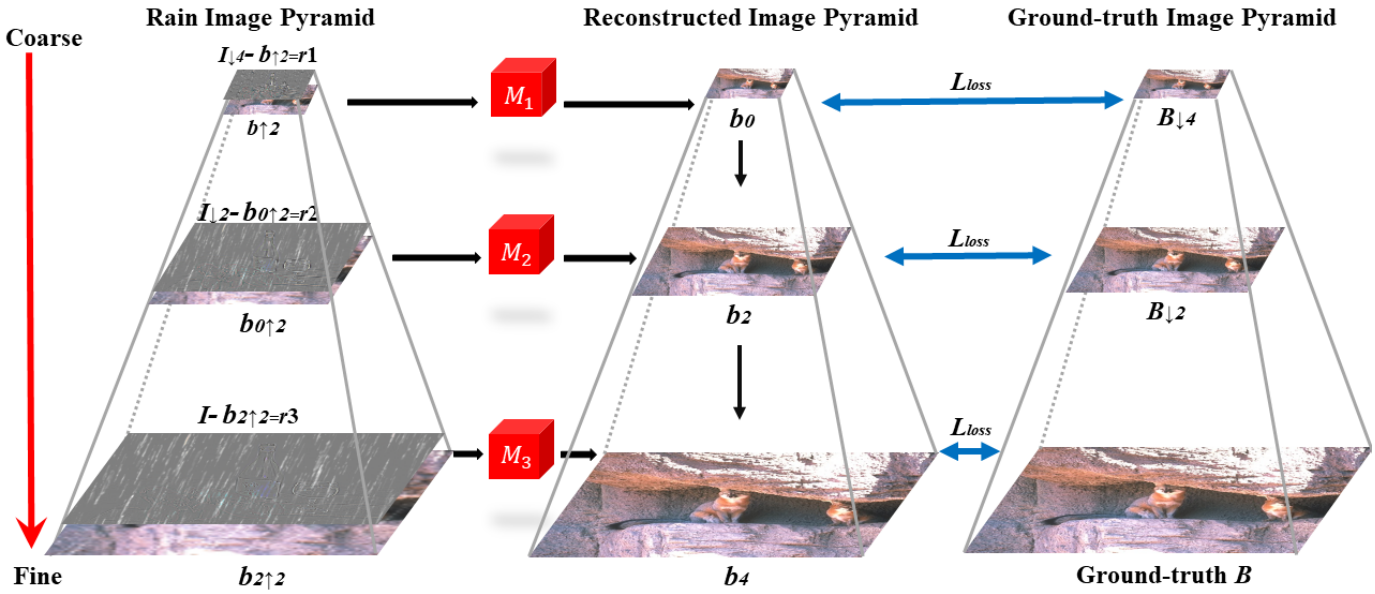


Fig. 2: Illustration of our proposed pipeline. Our network removes rain and recovers background details in a coarse-to-fine manner. I is an input rain image. $M_m, m = 1, 2, 3$ denotes the deraining metwork in the m-th stage. $\downarrow n$ and $\uparrow n$ indicate downsampling and upsampling an image by a factor of $n\times$. L_{loss} is the loss function in each stage.

into individual layers. In addtion, the inaccurate estimation of the rain layer number often affects the performance of removal. Instead of rain layer estimation and separation, we introduce a recurrent residual multiscale pyramid framework that gradually removes rain streaks by multiple stages.

B. Residual Multiscale Pyramid Framework

Since various rain steaks may exist in an image at the same time and might not be separated into isolated layers, we propose a residual multiscale deraining framework without estimating densities of rain explicitly. As illustrated in Fig. 2, a 3-level image pyramid is constructed, by which rain streaks with heavy to light densities are removed and background details are recovered gradually.

A downsampling operation based on the aggressive scaling factor removes severe rain streaks while reserving the image structure. As seen in Fig. 3a, for an input rain image, background details and light rain streaks are invisible in the low-resolution level. Stage 1 removes heavy rain and recovers the background structure in the coarse level. As illustrated in Fig. 2, we downsample a rain image I by factor $8\times$, namely b, and then upsample b with factor $2\times$, denoted as $b_{\uparrow 2}$. After that we downsample the same rain image I with factor $4\times$, denoted as $I_{\downarrow 4}$. The residual between $b_{\uparrow 2}$ and $I_{\downarrow 4}$, namely $r1$, is calculated as the attention map in the first stage. Fig. 3d shows an example of $r1$. As shown in Fig. 3d, $r1$ retains information of regions contaminated by heavy rain as well as background structure. The residual $r1$ is used as an attention map for heavy rain streaks removal and background structure recovery in stage 1.

Similarly, in stage 2, the deraining result of stage 1, namely $b0$, is processed by upsamping with factor $2\times$, denoted as $b0_{\uparrow 2}$. The residual between $b0_{\uparrow 2}$ and $I_{\downarrow 2}$ is denoted as $r2$. An example of the $r2$ is shown in Fig. 3e. We construct $b0_{\uparrow 2}$ and $r2$ as an image pair, and then feed this pair of images to the deraining model in stage 2, namely M_2, for medium rain removal and background details recovery.

In the last stage, the result from stage 2 is upsampled and denoted as $b2_{\uparrow 2}$, and residual between I and $b2_{\uparrow 2}$ is calculated as an attention map $r3$ in the original resolution level. As visible in Fig. 3f, light rain streaks which are not visible in the former two stages and fine image details appear in the attention map $r3$. $r3$ and $b2_{\uparrow 2}$ as an image pair are fed into the deraining model in stage 3, denoted as M_3, to obtain the final deraining result. Since the former two stages have removed the severe and medium rain streaks, we focus on removing light rain and recovering fine background details in this stage.

The procedure of our deraining method is expressed as follows:

$$y_n = f_n(y_{n-1_{\uparrow 2}} \oplus (I_{\downarrow 2^{3-n}} - y_{n-1_{\uparrow_2}})), \tag{2}$$

where n represents the n-th deraining stage and $n \in \{1, 2, 3\}$. y_n is the deraining result of the stage n, f_n is the deraining function, and y_{n-1} is the deraining result of the previous stage. $\oplus$ denotes concatenation operation, I is the rain image, $I_{\downarrow n}$ indicates downsampling I by the factor of $n\times$, and $y_{n-1\uparrow n}$ is the operation of upsampling the deraining result from the previous stage by the factor of $n\times$. Note that, y_0 is initiated as $I_{\downarrow 8}$.

Fig. 3 demonstrates an example of the input rain images and the residual attention maps in the three stages. As illustrated in Fig. 3c, finer background details and light rain streaks mainly appear in the high-resolution image. Fig. 3a and Fig. 3b show that as the resolutions of the images decrease,

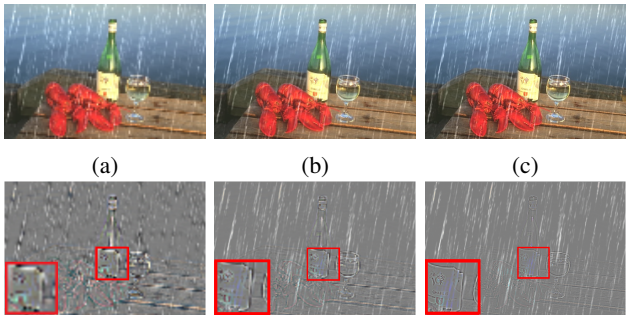


Fig. 3: Visualization of the input rain images and the residual attention maps in the three stages. (a), (b), and (c) are the rain images in the first, second, and third stages. (d), (e), and (f) are the residual attention maps in the first, second, and third stages, respectively.

heavy rain becomes mild and the structure of background is easily recognized. Therefore, it is much easier to remove heavy rains and restore the background structure in the low-resolution images. Benefiting from our mutliscale framework and residual attention guided mechanism, our method removes rain streaks while retaining fine background details.

C. Recurrent Mechanism

In our preliminary work [1], we design three different deraining networks for different stages, and as the resolution of the recovered image increases, the parameters of each network increase dramatically. Note that those three networks need to be trained in a stage-wise manner, which leads to the following two drawbacks: (i) The training of those three networks is not efficient. (ii) The network cannot be deployed easily in the practical application due to the large parameter size of the three networks.

To improve our training efficiency and achieve a compact model, we re-design and apply the rain removal networks with the same structure containing recurrent layers at different pyramid levels. In particular, we first employ a Conv-LeakyReLU layer to transform images into feature space. We then introduce a feature extraction block, named Multiscale Kernel Selection Block (MKSB), to further extract rain features. To recover an RGB image from the image features, we adopt a convolutional layer afterwards. The architecture of our deraining network is illustrated in Fig. 4a.

D. Multiscale Kernel Selection Networks

Considering the types (*e.g.*, densities, sizes and shapes) of rain streaks varies among the pyramid levels, we introduce **M**ultiscale **K**ernel **S**election **B**locks **(MKSB)** to perform rain feature extraction in different pyramid levels adaptively. In this way, our single network can be applied to the multiple stages without decreasing the deraining performance.

As shown in Fig. 4b, MKSB first extracts input features by using two Conv-LeakyReLU layers. Then, two parallel convolutional branches are exploited to further extract features with different receptive fields, *i.e.*, one with a kernel size of 3×3 pixels and the other with a kernel size of 5×5 pixels. We then propose an attention module to enable our MKSB to incorporate different receptive fields. Specifically, we firstly merge the features from those two branches via an element-wise summation. Then, we apply global average-pooling to generate a channel-wise global representation, denoted by s. Inspired by [12], a fully-connected layer f_z is used to generate a compact feature importance weight, namely z, along the feature channels for the adaptive selections. Furthermore, two soft attentions across channels, denoted as w_1 and w_2, respectively, are employed to select different spatial scales adaptively via two fully-connected layers f_1 and f_2. Note that, the summation of the n-th elements in w_1 and w_2 equals 1. At last, the features are re-weighted by the attention weights generated from different kernels. In this manner, rain features can be extracted adaptively in different stages.

Our MKSB facilitates our deraining network to capture rain streaks in different scales. Fig. 4b illustrates the architecture of our MKSB. Note that, as illustrated in Fig. 4a, we employ multiple MKSBs to capture rain layers and the stacked MKSBs may lead to a heavy-weight network. Therefore, we applied the recurrent mechanism to the stacked MKSBs. In this manner, the parameters of MKSBs are shared and thus we achieve a light-weight deraining network.

E. Training Loss Function

We construct a 3-level pyramid in our method. In each stage, we employ three loss functions, including a pixel-wise intensity similarity loss L_2, a perceptual loss [36] L_p, and a loss L_s (SSIM), to train the deraining model. The pixel-wise intensity similarity loss is expressed as:

$$L_2 = \frac{1}{CWH}\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{h=1}^{H}\|M(X^{c,w,h}) - B^{c,w,h}\|^2, \quad (3)$$

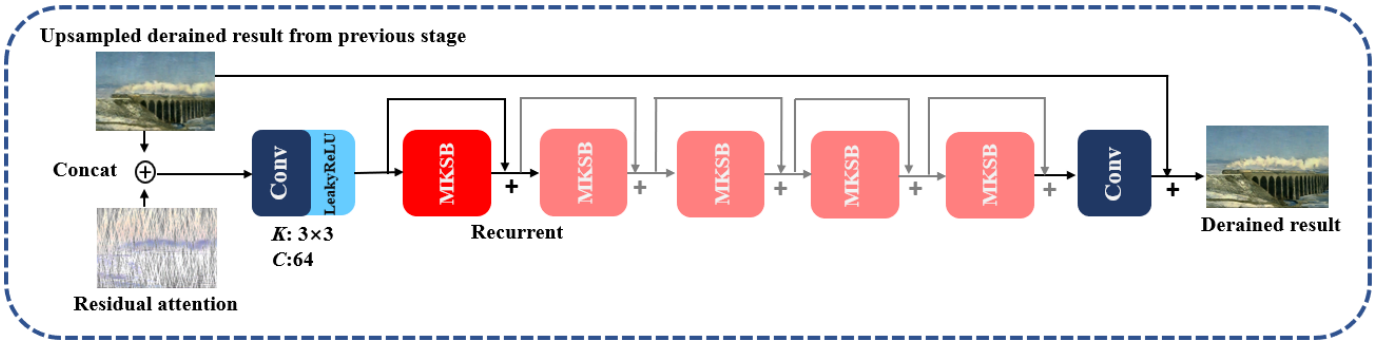
where X is the input image of each stage. At each stage, we concatenate the upsampled derained result from previous stage and the residual attention as our input X. C, W, and H correspond to the channel number, width, and height of the image respectively. $M(X)$ denotes the output rain-free image, and B indicates the corresponding ground-truth.

Aside from the pixel-wise intensity similarity loss, we apply a perceptual loss [36] to decrease global discrepancy between the features of generated rain-free images and those of their corresponding ground-truths. The perceptual loss is written as:

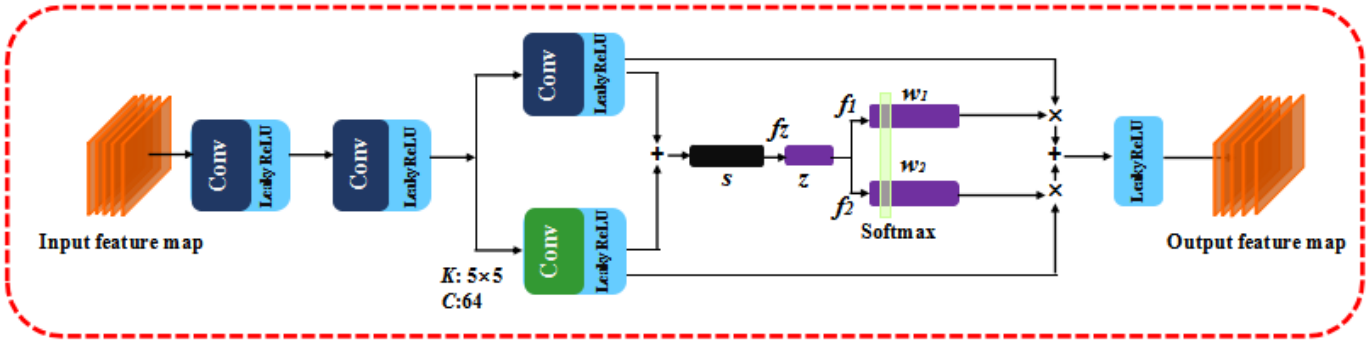
$$L_p = \frac{1}{CWH}\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{h=1}^{H}\|\Psi(M(X^{c,w,h})) - \Psi(B^{c,w,h})\|^2, \quad (4)$$

where Ψ represents a pre-trained feature extraction model [36, 37]. In our method, the feature maps from Relu2_2 in VGG16 is utilized.

Similar to [38], the loss L_s (SSIM) can be used to measure the structure similarity between a restored image and its



(a) MSKSN



(b) MKSB

Fig. 4: (a) The architecture of the proposed multiscale kernel selection networks **(MSKSN)**. (b) The structure of the multiscale kernel selection block **(MKSB)**. K denotes the convolutional kernel size, C represents the number of the feature map channels, f_z, f_1, and f_2 are fully-connected layers.

ground-truth image, expressed as:

$$L_s = 1 - \frac{1}{CWH}\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{h=1}^{H} SSIM(M(X^{c,w,h}), B^{c,w,h}), \tag{5}$$

In Eqn. (5), SSIM of two image patches v and t is defined

$$SSIM = \frac{(2u_v u_t + c_1)(2\sigma_{vt} + C_2)}{({u_v}^2 + {u_t}^2 + C_1)({\sigma_v}^2 + {\sigma_t}^2 + C_1)}, \tag{6}$$

where u_v and u_t represent the average intensities of patch v and t, respectively. σ_v^2 and σ_t^2 denote the variances of their corresponding patches, and σ_{vt} is the covariance between v and t. We set the patch size to 11×11 pixels.

The loss of our deraining model in each stage is written as:

$$L_{loss} = aL_2 + bL_p + cL_s, \tag{7}$$

where a, b, and c are weights for different loss functions.

In the pyramid, each stage has the corresponding clean and rain images. Overall, the total objective of our proposed method is written as:

$$L = \sum_{m\in\{1,2,3\}} L_{loss}^m, \tag{8}$$

where m represents the m-th stage. This multi-stage supervision guides the network to remove rain streaks and reconstruct clean images in a coarse-to-fine fashion. Note that, for our new network architecture, we do not need to train the network until converge in the coarse level and then switch to finer-level ones.

F. Implementation Details

In all our experiments, we establish a 3-level image pyramid. In the pyramid, a higher level corresponds to lower-resolution images. Each level is a downsampled version of its previous level by factor 2×. The residual maps are obtained by subtracting the upsampled derained images from the corresponding rain images. In the MKSB, the kernel size of the two parallel branches are 3×3 and 5×5 pixels respectively, and the channel number of the first fully-connected layer is 32. The downsampled rain-free images are employed as the supervision signals, and we employ the Adam optimizer [39] to train our network. In the loss function, we set a, b, and c as 1, 0.01, and 0.1, respectively. The decay rates for the first and second moment estimates are set to 0.9 and 0.99, respectively. The learning rate is initially set to 0.001, and the batch size is set as 4. We train our network with rain and clean image patches, and the patch size is 256×256 pixels. After 30 epochs, the learning rate is decayed by a factor of 0.7 every 10 epochs. We will release our source codes and training protocols for the purpose of research reproduction.

IV. Experiments

Due to the difficulty of obtaining the pairs of rain-free and rain images in real cases, our network is first trained on synthetic datasets, and then evaluated on both the synthetic and real-world rain images. Moreover, seven state-of-the-art methods are included for comparisons: discriminative sparse coding (DSC) [7], Gaussian mixture model (GMM) [6], DNN

[11], RESCAN [12], DID-MDN [15], DuRN-S [40], SPANET [13], and LPNet[29]. For fair comparisons, we use the public codes or models provided by the authors to produce the deraining results.

A. Datasets and Evaluation Metric

We compare our proposed MSKSN with state-of-the-art methods on five synthetic datasets:

- **Rain1400** [11] contains 14,000 rain and clean image pairs. They are synthesized from 1,000 clean images with 14 kinds of different rain-streaks directions and scales. Following the protocols in [11], we select 9,000 image pairs for training and the remaining 4,900 pairs for evaluation.
- **Rain12000** [15] contains 12,000 pairs of images for training, where the number of images with light, medium and heavy rain is all 4000. Moreover, 1,200 synthetic image pairs are contained for test as well.
- **Rain100H** [14] has 1,800 pairs of heavy rain images for training and 100 pairs for testing. As suggested by [14], although some synthesized examples in Rain100H are inconsistent with real rain images, using these data for training can further enhance the robustness of models.
- **Rain200L** [41] has 1,800 pairs of light rain images for training and 200 pairs of images for testing.
- **Rain800** [16] consists of 700 rain and clean image pairs for training and 100 pairs for testing.

Moreover, the work [14] also provides a real-world rain dataset, denoted as **Real15**, which contains 15 images downloaded from the Internet. The collected images in the Real15 have different contents, and rain densities and directions. Since those images do not have the corresponding ground-truth, we evaluate the performance of our network in real cases on this dataset.

In our experiments, Peak Signal to Noise Ratio (PSNR) [42] and Structural Similarity (SSIM) [38] are employed to evaluate the performance quantitatively on the synthetic datasets. We only compare the derained images with other methods qualitatively on the real-world dataset Real15.

B. Results on Synthetic Dataset

We compare our method with the seven state-of-the-art methods mentioned above. DSC and GMM are sparse coding based methods. DNN, RESCAN, DID-MDN, DuRN-S, SPANET, and LPNet are deep learning based methods.

Table I indicates the results of different methods on these five testing sets in terms of PSNR and SSIM. As shown in Table I, our method considerably outperforms the other methods in both PSNR and SSIM. In particular, our results achieve 0.62dB, 0.11dB, 0.51dB, 0.23dB and 0.53dB improvements in PSNR, and achieve the improvements of 0.003, 0.013, 0.031, 0.008 and 0.028 in SSIM on these five datasets respectively in comparison to the methods which obtain the second best performance, apart from our preliminary work [1].

In Fig. 5, Fig. 6, and Fig. 7, we randomly provide three qualitative comparisons with the state-of-the-art methods on the synthetic datasets.

As shown in Fig. 5, since GMM removes rain by estimating the rain layer and the clean background layer with patch-based image priors and represents rain streaks based on sparse coding, the performance is limited by the image patch as well as the representation ability of sparse coding. Thus in the case of heavy rain, the performance of the method degrades seriously. Deep learning based methods, such as DNN, LPNet, RESCAN, DID-MDN, DuRN-S, and SPANET outperform DSC and GMM. Although those methods better remove rain streaks, the edges of the mountain are blurred and some heavy rain streaks are still visible in the derained results of Fig. 5. On the contrary, our method removes heavy rain streaks while recovering the shape of the mountain edge.

Another example on the dataset Rain800 is shown in Fig. 6. In this case, rain streaks are relatively sparse and light than those in Fig. 5. As observed, the traditional method GMM leaves some rain streaks in the derained result. The deep learning based methods DNN, RESCAN, DID-MDN, DuRN-S, and SPANET remove most of the rain streaks. However, some artifacts remain in the derained results. Our MSKSN removes rain streaks as well as generates a derained result in better visual quality.

Fig. 7 shows an example on the Rain12000. In this case, many rain streaks of different scales appear in an image. As observed, compared to the other state-of-the-art methods, our MSKSN successfully removes rain streaks in different scales while reducing artifacts.

C. Results on Real-World Dataset

As real-world rain images often do not have rain-free ground-truth, we only visually evaluate the deraining performance on Real15. Fig. 8, Fig. 9, and Fig. 10 illustrate three examples of the derained results of the real-world rain images.

As seen in Fig. 8, rain streaks are long, and rain streaks of different densities tangle with each other. In Fig. 9, rain streaks are shorter than those in Fig. 8. As observed in Fig. 8 and Fig. 9, rain streaks still remain in the derained results of the state-of-the-art methods, while our MSKSN achieves better visual quality with much cleaner background images.

Fig. 10 shows another derained example of a real-world image. As illustrated in Fig. 10, some rain streaks are still visible in other state-of-the-art methods' derained results. In contrast, our MSKSN removes rain more thoroughly while recovering clearer background.

Fig. 8, Fig. 9, and Fig. 10 demonstrate that in real-world case it is common that rain streaks of different scales and densities appear simultaneously and often tangle with each other in an image. Removing rain of different scales and densities in a single scale may not be effective. In contrast, benefiting from our multiscale residual based scheme, our MSKSN is able to remove real-world rain streaks more effectively.

D. Ablation Study

In this section, all the ablation studies are conducted on the Rain100H, Rain200L, and Rain800.

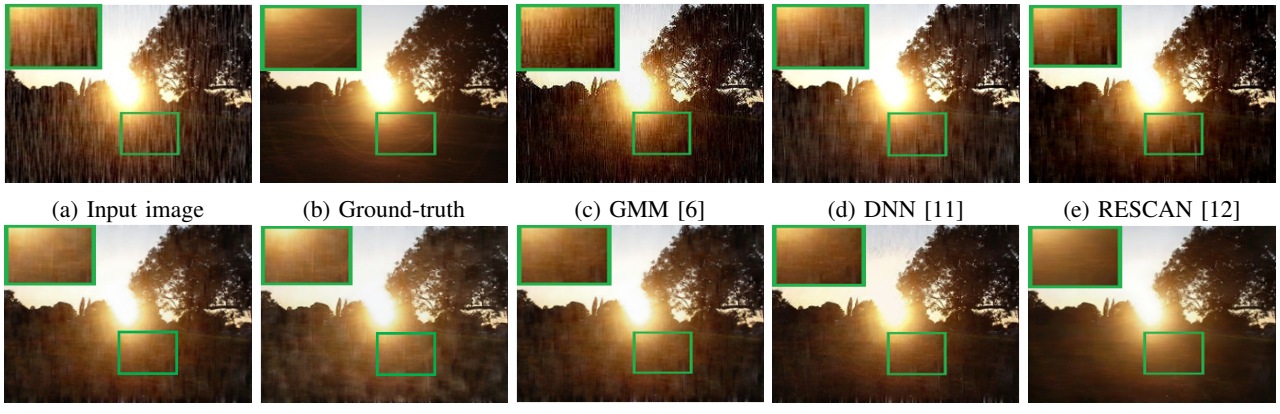
TABLE I: Quantitative comparisons on five synthetic benchmarks. The first and second rows denote results in terms of PSNR(dB) and SSIM, respectively.

Methods	Rain1400	Rain12000	Rain100H	Rain200L	Rain800
DSC [7]	22.03 0.799	21.44 0.789	15.66 0.544	23.39 0.870	18.56 0.599
GMM [6]	25.64 0.836	22.75 0.835	14.26 0.423	29.11 0.880	22.27 0.741
DNN [11]	28.24 0.901	27.33 0.898	24.95 0.781	32.04 0.938	21.16 0.732
RESCAN [12]	28.57 0.891	27.42 0.885	26.45 0.846	36.64 0.975	24.09 0.841
LPNet [29]	26.16 0.860	24.87 0.853	23.73 0.811	34.26 0.954	21.97 0.812
DID-MDN [15]	27.99 0.869	27.95 0.908	17.39 0.612	25.70 0.858	21.89 0.795
DuRN-S [40]	31.30 0.919	33.21 0.925	28.19 0.894	37.23 0.972	26.33 0.871
SPANET [13]	28.08 0.926	28.64 0.911	26.49 0.912	35.18 0.983	23.46 0.862
Our_preliminary [1]	30.21 0.925	29.41 0.923	28.26 0.909	37.18 0.971	26.46 0.894
Ours (MSKSN)	**31.92** **0.929**	**33.32** **0.938**	**28.70** **0.943**	**37.46** **0.991**	**26.86** **0.899**

(a) Input image (b) Ground-truth (c) GMM [6] (d) DNN [11] (e) RESCAN [12]

(f) LPNet [29] (g) DID-MDN [15] (h) DuRN-S [40] (i) SPANET [13] (j) Ours (MSKSN)

Fig. 5: Comparisons with the state-of-the-art on the Rain100H. The input image contains severe rain streaks. MSKSN removes rain streaks more thoroughly while the edges of the stone and the mountain are blurred by other methods.

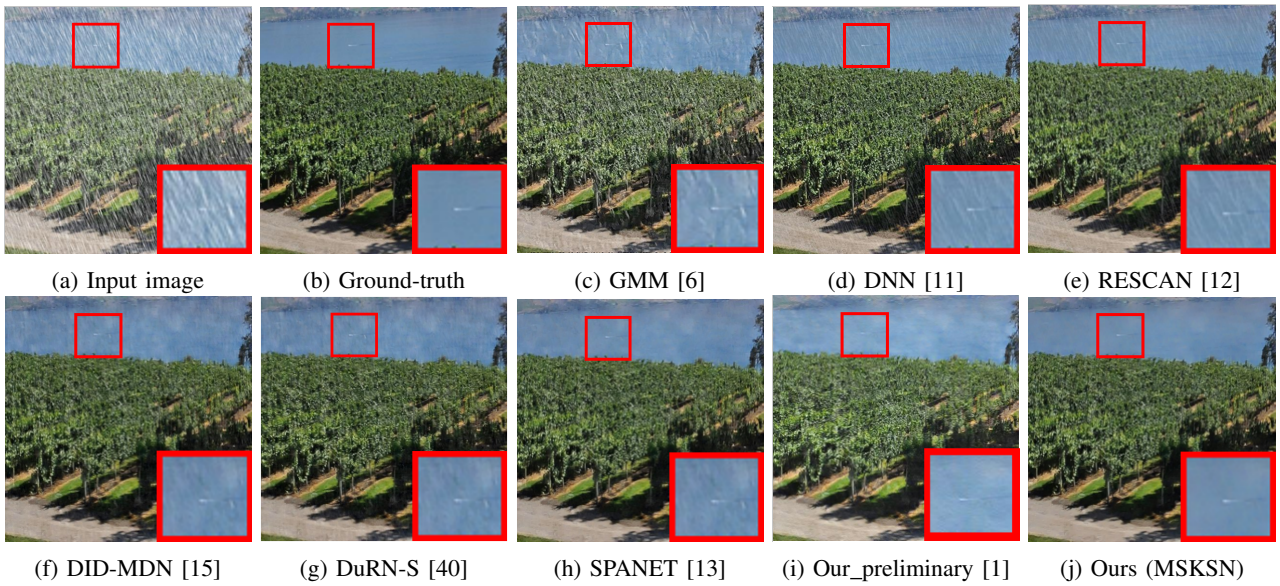


(a) Input image (b) Ground-truth (c) GMM [6] (d) DNN [11] (e) RESCAN [12]

(f) DID-MDN [15] (g) DuRN-S [40] (h) SPANET [13] (i) Our_preliminary [1] (j) Ours (MSKSN)

Fig. 6: Comparisons with the state-of-the-art on the Rain800. MSKSN removes rain streaks more thoroughly while reducing artifacts.

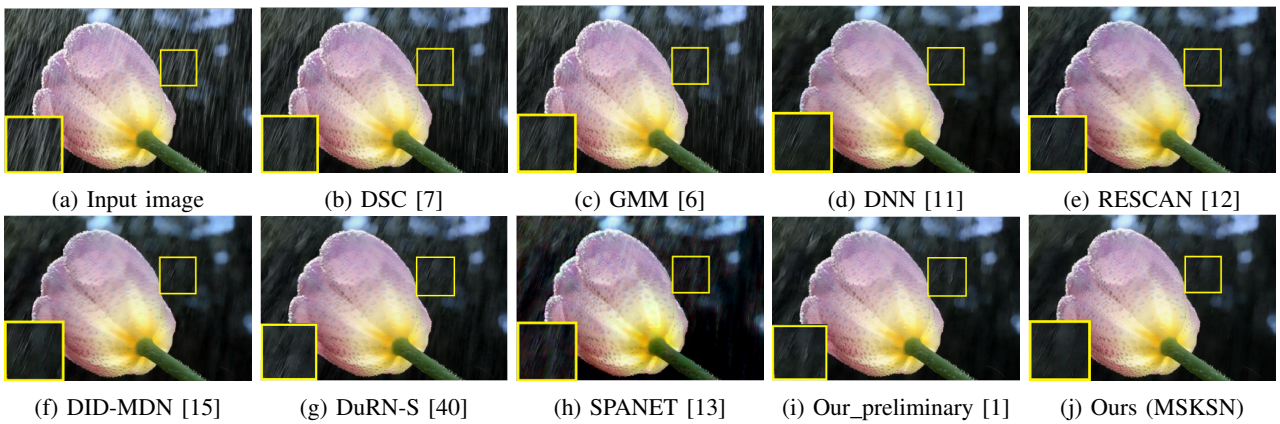
*1) **Multiscale pyramid network**:* To investigate the effects of our proposed deraining framework, we design the experiments shown in Table II. **(1)** Removing rain with only a 1-level pyramid network, denoted as **1-level**; **(2)** Removing rain with



(a) Input image (b) Ground-truth (c) GMM [6] (d) DNN [11] (e) RESCAN [12]

(f) DID-MDN [15] (g) DuRN-S [40] (h) SPANET [13] (i) Our_preliminary [1] (j) Ours (MSKSN)

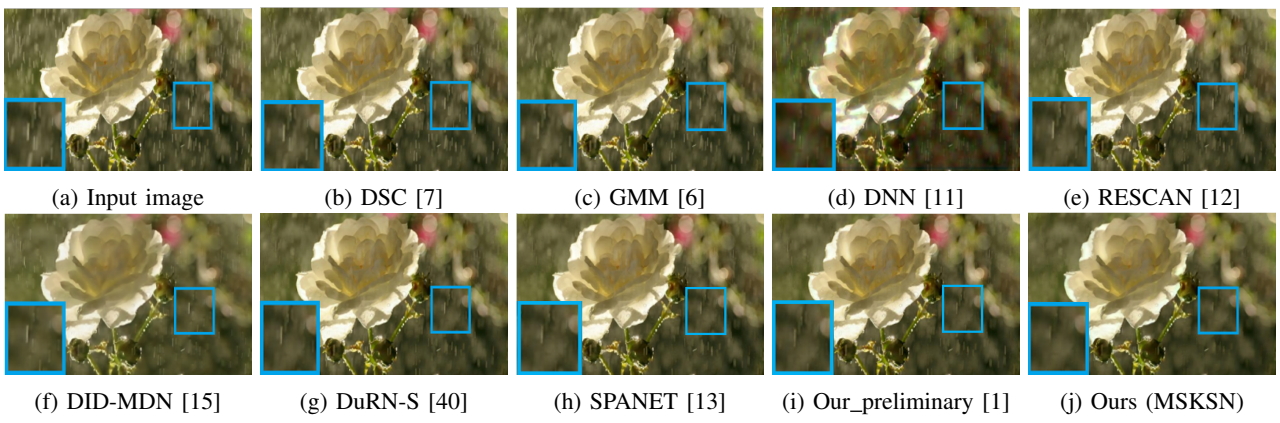
Fig. 7: Comparisons with the state-of-the-art on the Rain12000. MSKSN removes rain streaks more thoroughly while reducing artifacts. The derained image quality of MSKSN is better.



(a) Input image (b) DSC [7] (c) GMM [6] (d) DNN [11] (e) RESCAN [12]

(f) DID-MDN [15] (g) DuRN-S [40] (h) SPANET [13] (i) Our_preliminary [1] (j) Ours (MSKSN)

Fig. 8: Comparisons with the state-of-the-art on the Real15. In the input image, rain streaks of different scales tangle with each other. Our MSKSN removes more rain streaks at the overlapping areas.



(a) Input image (b) DSC [7] (c) GMM [6] (d) DNN [11] (e) RESCAN [12]

(f) DID-MDN [15] (g) DuRN-S [40] (h) SPANET [13] (i) Our_preliminary [1] (j) Ours (MSKSN)

Fig. 9: Comparisons with the state-of-the-art on the Real15. The visual quality of the derained result of MSKSN is better.

a 2-level pyramid network, marked as **2-level**. The cases **(1)** and **(2)** are used to demonstrate the contribution of different image pyramid levels to the final performance.

As shown in Table II, with the help of the multiscale pyramid framework, we attain better deraining performance in comparison to rain removal only in original resolution.

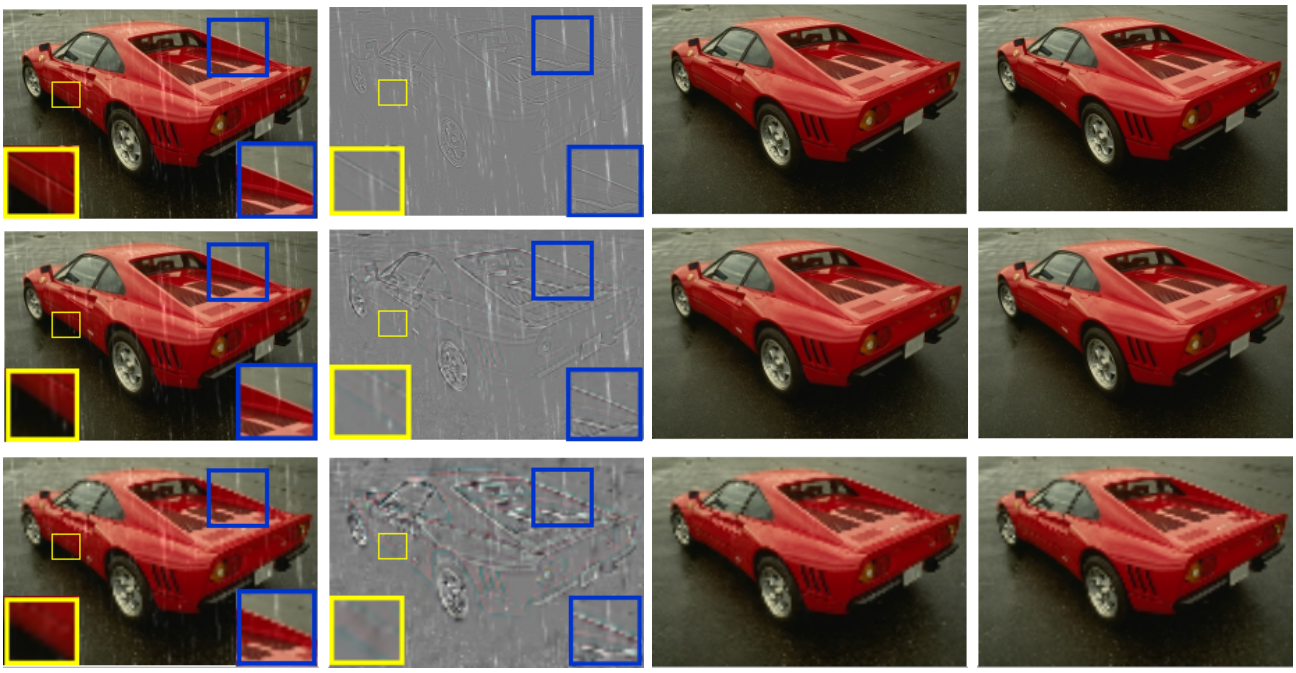
(a) Input image (b) DSC [7] (c) GMM [6] (d) DNN [11] (e) RESCAN [12]

(f) DID-MDN [15] (g) DuRN-S [40] (h) SPANET [13] (i) Our_preliminary [1] (j) Ours (MSKSN)

Fig. 10: Comparisons with the state-of-the-art on the Rain15. The visual quality of the derained result of MSKSN is better.

(a) Input image (b) Ground-truth Inf/1 (c) 1-level 27.31/0.926 (d) 2-level 27.98/0.931 (e) 3-level 28.47/0.940

Fig. 11: Illustration of the results with and without our multiscale strategy. PSNR/SSIM scores are provided for reference.



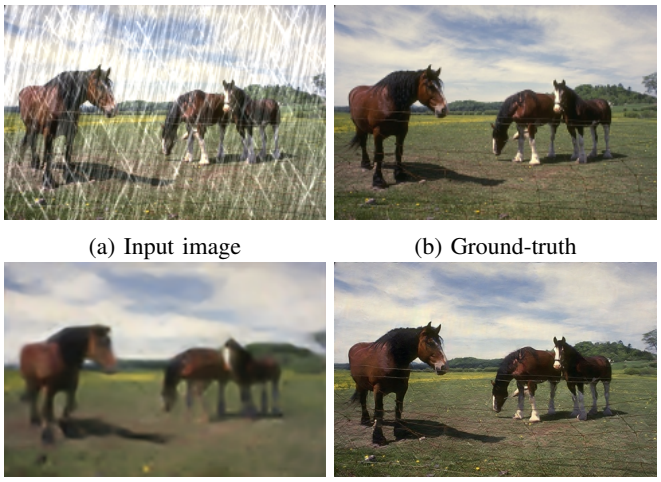
(a) Rain image (b) Residual attention (c) Derained result (d) Ground-truth

Fig. 12: Illustraions of derained results and the residual attention maps in different stages. The rain images, the residual attentions, the derained results, and the ground-truths corresponding to the first, second, and third stages are shown in row 3, row 2, and row 1, respectively.

As visible in Fig. 11, compared to using a single scale for detraining, our multiscale method significantly facilitates removal of heavy rain. As shown in Fig. 12a and Fig. 12b, since light rain streaks (*i.e.*, the yellow box) become invisible and heavy rain streaks (*i.e.*, the blue box) are removed in the low-resolution image, our method reduces the difficulty of addressing rain with various densities in an image. Fig. 12c illustrates intermediate derained results and the residual

TABLE II: Ablation study of the pyramid framework and the residual attention.

Methods	Metric	Rain100H	Rain200L	Rain800
(1) 1-level	PSNR(dB)	27.92	36.92	26.18
	SSIM	0.930	0.985	0.882
(2) 2-level	PSNR(dB)	28.44	37.38	26.49
	SSIM	0.940	0.990	0.890
(3) 3-level-ro	PSNR(dB)	20.79	22.43	19.16
	SSIM	0.741	0.763	0.721
MSKSN	PSNR(dB)	**28.70**	**37.46**	**26.86**
	SSIM	**0.943**	**0.991**	**0.899**



(a) Input image (b) Ground-truth

(c) 3-level-ro (d) MSKSN

Fig. 13: Illustration of the results with and without residual attention mechanism.

attention at different stages. Background image details are gradually restored as the image resolution becomes larger.

*2) **Residual attention mechanism**:* To explore the contribution of the residual attention, we conduct the experiments: Removing rain with a 3-level multiscale pyramid without the residual attention, denoted as **3-level-ro**, on the three datasets.

As seen in Fig. 13, the image recovered by **3-level-ro** is blurred while our method restores rich background details. As manifested in Table II, the rain removal performance decreases dramatically without our residual attention mechanism. This indicates that our residual attention mechanism aids in removing rain streaks significantly.

*3) **Kernel selection mechanism**:* In order to investigate the effects of the kernel selection mechanism in our MSKSN for the deraining task, we conduct the following experiments: **(1)** Removing the branch in which the kernel size of the convolutional layer is 3×3 in MKSB, marked as **MKSB-3**; **(2)** Removing the branch in which the receptive field of the convolutional layer is 5×5 in MKSB, denoted as **MKSB-5**; **(3)** Removing both of the two branches in MKSB, denoted as **MKSB-roks**. **(4)** In MKSB, besides the two branches, adding another branch in which the receptive field of the convolutional layer is 7×7, denoted as **MKSB+7**.

As shown in Table IV, without the two branches, the performance of MSKSN degrades 1.09dB, 1.01dB, and 1.09dB in PSNR, and 0.009, 0.009, and 0.003 in SSIM on the RainH, RainL, and Rain800 respectively. The experiments MKSB-3 and MKSB-5 are the same as only assigning different weights for the different channels of feature maps without the kernel selection. However, since the same network structure is employed to remove rain in the three stages, and the sizes of rain streaks vary as the image resolution increases, adjusting receptive field adaptively in the different stages helps with capturing rain streaks of different scales. Thus, the kernel selection mechanism promotes the performance of our MSKSN significantly. With more branches, the performance of MKSB+7 improves 0.14dB, 0.03dB, and 0.04dB in PSNR on the RainH, RainL, and Rain800 respectively. However, incorporating more branches will increase parameters as well as require more GPU memory to train the model. Thus, considering the balance between the performance and efficiency, we set the number of the branches to 2 in our MKSB.

To dissect the impacts of the kernel selective mechanism in our MSKSN, we calculate the ratios of the weights of the two attention vectors in MKSB. In particular, we divide the attention weights of the branch in which the kernel size is 5×5, by the attention weights of the branch in which the receptive field is 3×3, and then we count the numbers that the ratios are greater than 1 in each MKSB. Since we unfold a MKSB 5 times in a deraining network, we denote the MKSB at the ith recurrent time as $MKSB_i$.

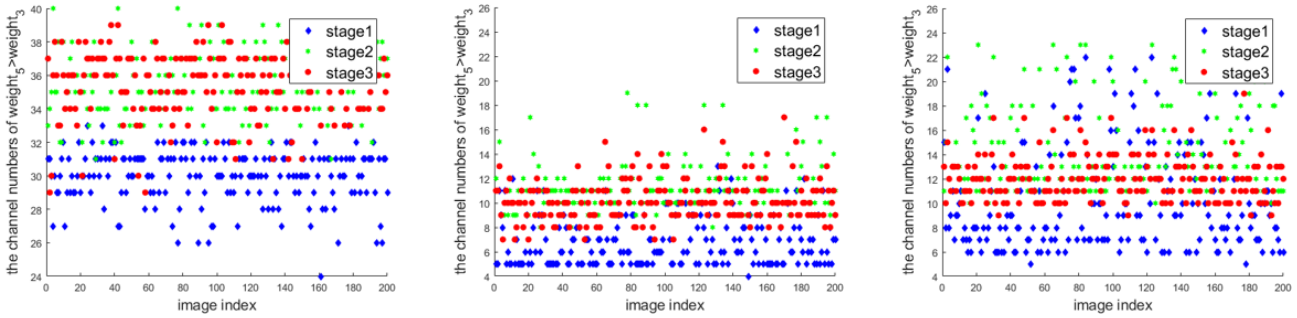
Fig. 14 shows the visualizations of all MKSBs in the three stages on the Rain200L. As illustrated in Fig. 14, in each stage, the neurons enlarge their receptive fields as the network becomes deeper. In particular, the neurons in the stage 1 enlarge the receptive fields faster than the neurons in the

TABLE III: Ablation study of the recurrent mechanism.

Methods	Metric	Rain100H	Rain200L	Rain800
(1) MSKSN.1	PSNR(dB)	26.85	35.69	24.65
	SSIM	0.929	0.987	0.881
(2) MSKSN.4	PSNR(dB)	28.51	37.32	26.75
	SSIM	0.940	0.989	0.897
(3) MSKSN.S	PSNR(dB)	29.28	37.63	26.90
	SSIM	0.945	0.991	0.899
(4) MSKSN-AC	PSNR(dB)	27.43	36.25	26.64
	SSIM	0.933	0.980	0.884
(5) MSKSN.S-AC	PSNR(dB)	28.58	37.35	26.77
	SSIM	0.939	0.990	0.897
MSKSN	PSNR(dB)	28.70	37.46	26.86
	SSIM	0.943	0.991	0.899

TABLE IV: Ablation study of the kernel selection mechanism.

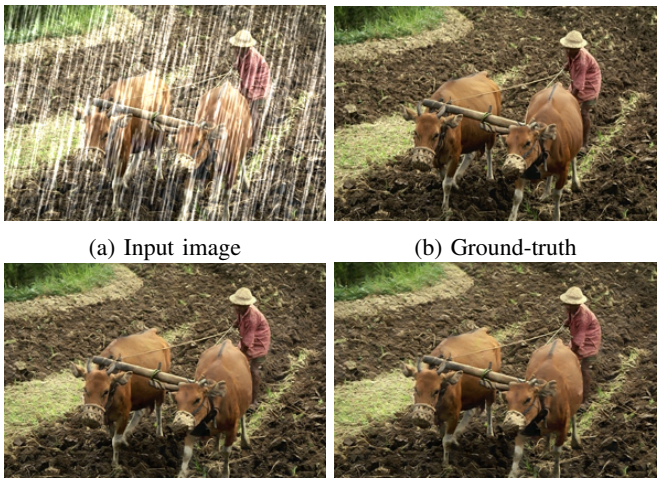
Methods	Metric	Rain100H	Rain200L	Rain800
(1) MKSB-3	PSNR(dB)	28.23	37.04	26.37
	SSIM	0.936	0.986	0.897
(2) MKSB-5	PSNR(dB)	27.96	36.80	26.10
	SSIM	0.936	0.984	0.896
(7) MKSB-roks	PSNR(dB)	27.61	36.45	25.77
	SSIM	0.934	0.982	0.896
(3) MKSB+7	PSNR(dB)	28.84	37.49	26.90
	SSIM	0.944	0.991	0.899



(a) initialization of MKSB_1 (b) MKSB_1 (c) MKSB_2

(d) MKSB_3 (e) MKSB_4 (f) MKSB_5

Fig. 14: Illustration of the kernel selective mechanism on the Rain200L. (a) The visualization of the number when the ratios are greater than 1 in $MKSB_1$ before training. (b), (c), (d), (e), and (f) are the visualizations of the number when the ratios are greater than 1 of the five MKSBs in the three stages.



(a) Input image (b) Ground-truth

(c) MSKSN (d) MSKSN.S

Fig. 15: Illustration of the derained results of MSKSN and MSKSN.S.

stage 2 and stage 3. The reason is that as the first stage in our deraining network, to achieve removing heavy rain and recovering background structure, the neurons enlarge receptive fields to gather more information and better capture features. In the stage 2, since the deraining network in this stage mainly removes medium rain streaks and restores some details of background, the neurons in the stage 2 enlarge their receptive fields to capture the features of the medium rain, and the pace of the increase is relatively lower than that in the stage 1. Since the former two stages have removed the heavy and the medium rain streaks, the goal of the last stage is to remove light rain and recover fine details of the image. The neurons in this stage also enlarge their receptive fields as the network becomes deeper, and the rate of the increase in this stage is lower than that in the previous two stages.

Overall, the kernel selective mechanism helps with our coarse-to-fine manner and improves the deraining performance significantly.

*4) **Recurrent mechanism**:* In our MSKSN, we recurrently unfold a MKSB 5 times instead of piling up 5 MKSBs one after another. To evaluate the effects of the recurrent mechanism, we conduct the following experiments: **(1)** Recurrently unfolding a MKSB just one time. denoted as **MSKSN.1**; **(2)** Recurrently unfolding a MKSB 4 times, marked as **MSKSN.4**; **(3)** Stacking 5 MKSB instead of unfolding a MKSB 5 times in MSKSN, denoted as **MSKSN.S**; **(4)** Applying the recurrent mechanism only across pyramid levels, marked as **MSKSN.S-AC**; **(5)** Applying the recurrent mechanism within as well as across pyramid levels, denoted as **MSKSN-AC**. In particular, cases **(1)** and **(2)** are designed to demonstrate how the recurrent mechanism contributes to the deraining performance. **(3)**, **(4)**, and **(5)** are designed to reduce the parameters of our network.

As indicated in Table III, the average PSNR and SSIM values of method MSKSN is slightly lower than MSKSN.S on the three datasets. However, MSKSN decreases 78% of the model parameters compared to MSKSN.S. Fig. 15 indicates that the derained result of MSKSN and the result of MSKSN.S are visually comparable.

TABLE V: Impacts of different loss functions. The first and second rows denote results in PSNR(dB) and SSIM, respectively.

Methods	Rain100H	Rain200L	Rain800
(1) $L_2 + L_2 + L_2$	28.47 0.936	37.35 0.989	26.57 0.892
(2) $L_2L_p + L_2L_p + L_2L_p$	28.52 0.940	37.38 0.990	26.76 0.894
(3) $L_2L_s + L_2L_s + L_2L_s$	28.57 0.941	37.42 0.990	26.80 0.896
(4) **MSKSN**	**28.70** **0.943**	**37.46** **0.991**	**26.86** **0.899**

Table III lists the quantitative evaluation values of MSKSN.1 and MSKSN.4. As observed, more recurrent numbers lead to higher average PSNR and SSIM. However, a larger recurrent number requires more GPU memory to train the network. Thus, we set the recurrent number as 5 in our MSKSN.

In the experiment MSKSN.S-AC, we apply the recurrent mechanism only across differnet pyramid levels. The result is shown in Table III. The performance of MSKSN.S-AC is inferior than our MSKSN. MSKSN.S-AC has 47,963 parameters while MSKSN has 30,225 parameters. In MSKSN-AC, we apply the recurrent mechanism within as well as across pyramid levels. In this case, the model has the least parameters of 10,075. However, the performance of MSKSC-AC is inferior than our MSKSN.

Overall, considering the trade-off between the model size and the performance, we choose MSKSN in which the recurrent number of MKSB is 5 as our proposed model.

*5) **Different losses**:* To investigate the impact of different losses on the final deraining performance, we design experiments by using different losses to train our network in Table V: (1) only using the pixel-wise intensity similarity loss L_2, (2) using the pixel-wise intensity and perceptual losses $L_2 + L_p$, (3) using the pixel-wise intensity similarity loss and SSIM loss $L_2 + L_s$, and (4) our final losses $L_2 + L_p + L_s$. As indicated in Table V, the perceptual loss and SSIM loss play important roles in improving deraining performance. Thus we employ both of the two loss functions to train our MSKSN.

V. Conclusion

In this paper, we proposed a recurrent residual attention based multiscale deraining network to realize rain streaks removal in a coarse-to-fine fashion. By leveraging the pyramid framework, we ease the deraining procedure and remove different degrees of rain in different scales while recovering background image from coarse structure to fine details. As the residual between the upsampled rain-free image and its corresponding rain image provides important clues to localize regions with rain, we fully exploit it as an attention map to let our deraining network better focus on rain regions recovery under our pyramid framework. We also apply a recurrent mechanism to our network architecture in each pyramid level to achieve a compact yet powerful deraining network. Extensive qualitative and quantitative results manifest that our proposed method achieves the state-of-the-art performance.

Acknowledgment

We would like to thank the associate editor and anonymous reviewers for their very valuable and constructive suggestions. This work was supported by the National Natural Science Foundation of China (No.61976017 and No.61601021), the Beijing Natural Science Foundation (No.4202056 and No.L172022), China Scholarship Council (No.201807095044) and the Fundamental Research Funds for the Central Universities (2020JBM078). The research in this paper was also supported by the Australian Research Council (DE180100628), and Australian Research Council Centre of Excellence for Robotic Vision (CE140100016).

References

[1] Yupei Zheng, Xin Yu, Miaomiao Liu, and Shunli Zhang. Residual multiscale based single image deraining. 2019.

[2] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018.

[3] Shunli Zhang, Xin Yu, Yao Sui, Sicong Zhao, and Li Zhang. Object tracking with multi-view support vector machines. *IEEE Transactions on Multimedia*, 17(3):265–278, 2015.

[4] Yi Chang, Luxin Yan, and Sheng Zhong. Transformed low-rank model for line pattern noise removal. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1726–1734, 2017.

[5] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Transactions on Image Processing*, 21(4):1742–1755, 2011.

[6] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown. Rain streak removal using layer priors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2736–2744, 2016.

[7] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3397–3405, 2015.

[8] Lei Zhu, Chi-Wing Fu, Dani Lischinski, and Pheng-Ann Heng. Joint bi-layer optimization for single-image rain streak removal. In *Proceedings of the IEEE international conference on computer vision*, pages 2526–2534, 2017.

[9] 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.

[10] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing*, 26(6):2944–2956, 2017.

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

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

[13] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12270–12279, 2019.

[14] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1357–1366, 2017.

[15] He Zhang and Vishal 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*, pages 695–704, 2018.

[16] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[17] Kshitiz Garg and Shree K Nayar. Detection and removal of rain from videos. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. IEEE, 2004.

[18] Yi-Lei Chen and Chiou-Ting Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1968–1975, 2013.

[19] Tai-Xiang Jiang, Ting-Zhu Huang, Xi-Le Zhao, Liang-Jian Deng, and Yao Wang. A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 4057–4066, 2017.

[20] Wei Wei, Lixuan Yi, Qi Xie, Qian Zhao, Deyu Meng, and Zongben Xu. Should we encode rain streaks in video as deterministic or stochastic? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2516–2525, 2017.

[21] Weihong Ren, Jiandong Tian, Zhi Han, Antoni Chan, and Yandong Tang. Video desnowing and deraining based on matrix decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4210–4219, 2017.

[22] Jie Chen, Cheen-Hau Tan, Junhui Hou, Lap-Pui Chau, and He Li. Robust video content alignment and compensation for rain removal in a cnn framework. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6286–6295, 2018.

[23] Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming Guo. Erase or fill? deep joint recurrent rain removal and reconstruction in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3233–3242, 2018.

[24] Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming Guo. D3r-net: Dynamic routing residue recurrent network for video rain removal. *IEEE Transactions on Image Processing*, 28(2):699–712, 2018.

[25] Minghan Li, Qi Xie, Qian Zhao, Wei Wei, Shuhang Gu, Jing Tao, and Deyu Meng. Video rain streak removal by multiscale convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6644–6653, 2018.

[26] Tai-Xiang Jiang, Ting-Zhu Huang, Xi-Le Zhao, Liang-Jian Deng, and Yao Wang. Fastderain: A novel video rain streak removal method using directional gradient priors. *IEEE Transactions on Image Processing*, 28(4):2089–2102, 2018.

[27] Yazan Hamzeh and Samir A Rawashdeh. Framework for simulating and rem,/oving rain in stereo-image videos. In *2018 IEEE International Conference on Electro/Information Technology (EIT)*, pages 0796–0802. IEEE, 2018.

[28] Jingfeng Zang, Guibin Ren, Jianning Dong, Yan Piao, and Seio Jim. Removal of rain video based on temporal intensity and chromatic constraint of raindrops. *Evolutionary Intelligence*, 12(3):349–355, 2019.

[29] 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*, 31(6):1794–1807, 2020.

[30] Ruoteng Li, Loong-Fah Cheong, and Robby T Tan. Single image deraining using scale-aware multi-stage recurrent network. *arXiv preprint arXiv:1712.06830*, 2017.

[31] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[32] Quoc V Le, Navdeep Jaitly, and Geoffrey E Hinton. A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint arXiv:1504.00941*, 2015.

[33] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. Depth-attentional features for single-image rain removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8022–8031, 2019.

[34] Youzhao Yang and Hong Lu. Single image deraining using a recurrent multi-scale aggregation and enhancement network. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1378–1383. IEEE, 2019.

[35] X. Yu, F. Xu, S. Zhang, and L. Zhang. Efficient patch-wise non-uniform deblurring for a single image. *IEEE Transactions on Multimedia*, 16(6):1510–1524, 2014.

[36] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-

resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[38] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

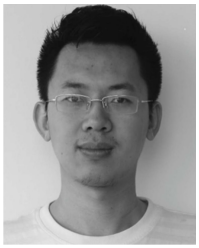
[39] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[40] Xing Liu, Masanori Suganuma, Zhun Sun, and Takayuki Okatani. Dual residual networks leveraging the potential of paired operations for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7007–7016, 2019.

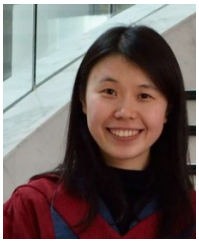
[41] Wenhan Yang, Robby T Tan, Jiashi Feng, Zongming Guo, Shuicheng Yan, and Jiaying Liu. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE transactions on pattern analysis and machine intelligence*, 42(6):1377–1393, 2019.

[42] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008.

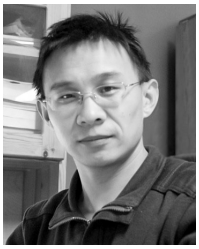
Yupei Zheng received her B.S. degree in School of Software Engineering from Beijing Jiaotong University, Beijing, China, in 2017. She was a visiting student with the Australian National University, Canberra, from 2018 to 2019. Her current research interests include low level computer vision, image processing, and deep learning.



Xin Yu received his B.S. degree in Electronic Engineering from University of Electronic Science and Technology of China, Chengdu, China, in 2009, and received his Ph.D. degree in the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2015. He also received a Ph.D. degree in the College of Engineering and Computer Science, Australian National University, Canberra, Australia, in 2018. He is currently a Lecturer in University of Technology Sydney. His interests include computer vision and image processing.



Miaomiao Liu is a Lecturer and an ARC DECRA Fellow in the Research School of Engineering, the Australian National University. She was a Research Scientist at Data61/CSIRO from 2016-2018. Prior to that she was a researcher in NICTA. She received the BEng, MEng, and PhD degrees from Yantai Normal University, Yantai, China, Nanjing University of Aeronautics and Astronautics, Nanjing, China, and the University of Hong Kong, Hong Kong SAR, China, in 2004, 2007, and 2012, respectively. Her research interests include 3D vision, 3D reconstruction and 3D scene modeling and Understanding. She is a member of the IEEE.



Shunli Zhang received the B.S. and M.S. degrees in electronics and information engineering from Shandong University, Jinan, China, in 2008 and 2011, respectively, and the Ph.D. degree in signal and information processing from Tsinghua University in 2016. He was a visiting scholar in Carnegie Mellon University, Pittsburgh, from 2018 to 2019. He is currently a faculty member in School of Software Engineering, BeijingJiaotong University. His research interests include pattern recognition, computer vision, and image processing.