IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 18, 2019, accepted January 15, 2020, date of publication January 20, 2020, date of current version January 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2967891

Coupled Rain Streak and Background Estimation via Separable Element-wise Attention

YINJIE TAN[1], QIANG WEN[1], JING QIN[2], JIANBO JIAO[3], GUOQIANG HAN[1], AND SHENGFENG HE[1], (Member, IEEE)

[1]Department of Computer Science and Engineering, South China University of Technology, Guangzhou 510641, China
[2]Department of Nursing, The Hong Kong Polytechnic University, Hong Kong
[3]Department of Engineering Science, University of Oxford, Oxford OX1 2JD, U.K.

Corresponding author: Shengfeng He (hesfe@scut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61472145, Grant 61972162, and Grant 61702194, in part by the Hong Kong Research Grants Council under Project PolyU 152035/17E, in part by the Special Fund of Science and Technology Research and Development on Application from Guangdong Province (SF-STRDA-GD) under Grant 2016B010127003, in part by the Guangzhou Key Industrial Technology Research Fund under Grant 201802010036, in part by the Guangdong Natural Science Foundation under Grant 2017A030312008, and in part by the CCF-Tencent Open Research Fund (CCF-Tencent) under Grant RAGR20190112.

ABSTRACT Single image de-raining is challenging especially in the scenarios with dense rain streaks. Existing methods resolve this problem by predicting the rain streaks of the image, which constrains the network to focus on local rain streaks features. However, dense rain streaks are visually similar to mist or fog (with large intensities), in this case, the training objective should be shifted to image recovery instead of extracting rain streaks. In this paper, we propose a coupled rain streak and background estimation network that explores the intrinsic relations between two tasks. In particular, our network produces task-dependent feature maps, each part of the features correspond to the estimation of rain streak and background. Furthermore, to inject element-wise attention to all the convolutional blocks for better understanding the rain streaks distribution, we propose a Separable Element-wise Attention mechanism. In this way, dense element-wise attention can be obtained by a sequence of channel and spatial attention modules, with negligible computation. Extensive experiments demonstrate that the proposed method outperforms state-of-the-arts on 5 existing synthesized rain datasets and the real-world scenarios, without extra multi-scale or recurrent structure.

INDEX TERMS Background estimation, de-raining, element-wise attention.

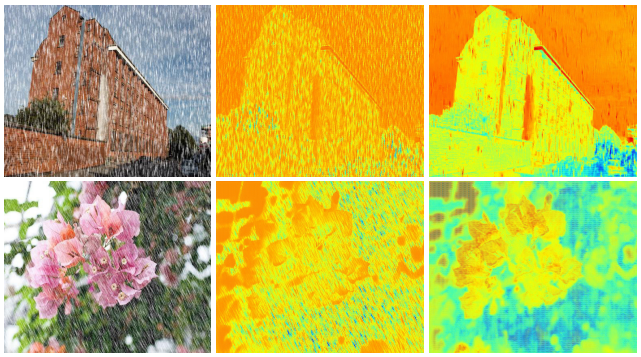
I. INTRODUCTION

Most existing computer vision systems are designed for disturbance-free scenarios. Therefore, rain streaks in an image degrade visibility and prevent many computer vision algorithms from working properly. Addressing this visibility problem is challenging due to the random rain streaks distribution. Early researches [2], [16], [17] treat it as a signal separation problem using low rank decomposition or Gaussian mixture models (GMM), or resolve it in a denoising manner with a nonlocal mean smoothing algorithm [13]. Recently, deep learning based models [4], [25], [27] learn from synthesized data and achieve preferable performance due to the powerful ability of feature representation.

Notwithstanding the demonstrated success, these deep models suffer from two main issues. First, most state-of-the-art models [4], [15], [25], [27] focus on predicting rain streaks only. While this is reasonable as the rain streaks are sparse and contain simple texture information, it enforces the network focus only on local feature representations. As can be seen in Fig. 1b, the feature responses learned from a residual prediction network highlights rain streaks other than background regions. On the other hand, the dense rain streak scenario is visually similar to mist or fog, which makes the prediction of rain streaks easy but difficult in recovering original image content. A network that predicts a rain-free background shows a different learning focus (see Fig. 1c), and these two different objectives may complement each other.

Second, the attention on rain streak distribution is not fully explored in de-raining models. Although a spatial visual attention map is incorporated as one of the network inputs in raindrop removal [19], the attention module should be injected into feature levels of the entire network. Attention not

The associate editor coordinating the review of this manuscript and approving it for publication was Kathiravan Srinivasan.



(a) Input (b) Residual features (c) Background features

FIGURE 1. Deep networks that estimate the rain streaks residual (b) or rain-free background (c) show substantially different feature responses (the activation value of the neurons). Relying on predicting the residual only cannot handle dense rain streaks scenarios.

only filters out the redundant information but also improve the representation of the features. In this sense, traditional spatial attention is not enough as it shares the same weights to all the channels of the feature maps. However, learning an element-wise attention module with the same size of the feature maps hugely increases the computational overhead.

In this paper, we address the above two problems by proposing a coupled rain streak and background estimation network with Separable Element-wise Attention. The proposed network produces task-dependent features so that the intrinsic relationship between two tasks can be explored during training. Furthermore, we implement element-wise attention using a sequence of channel and spatial attention modules. The combination of the channel and spatial attention modules is able to achieve the element-wise attention with negligible computation, in this way it can be applied to all the convolutional blocks. Extensive experiments show that the proposed method outperforms state-of-the-art de-raining methods on 5 benchmarks and real-world scenarios. More importantly, our superior performance is obtained without additional multi-scale or recurrent structures.

To summarize, our contributions are three-fold:

- We propose to jointly estimate rain streaks and background in the same network with task-dependent features. This simple approach shows significant improvement over individual prediction of two tasks.
- We present a Separable Element-wise Attention module. This method allows focusing on important feature elements while suppressing redundant ones. Additionally, our separable implementation enables involving element-wise attention with negligible computation efforts. It is a general component and can be applied to other deep models.
- Extensive experiments conducted on 5 challenging benchmarks and real-world data demonstrate the effectiveness of the proposed approaches over state-of-the-art methods.

II. RELATED WORK

Rain streak removal is challenging, and therefore early works leverage the additional temporal information from multiple frames. Garg and Nayar [5] propose to detect and remove rain streaks based on the dynamics and photometry of rain. Besides temporal information, other information such as chromatic properties and shape characteristics of rains, are also utilized in [29] and [1] respectively. Recently, video rain removal are addressed using low-rank matrix [14], optical flow in local phase information [21], and matrix decomposition [20].

Different from video-based de-raining with temporal information, single image rain removal is an ill-posed problem and therefore much more challenging. Many traditional methods solve this problem with additional prior information and regard it as a signal separation problem. Kang *et al.* [12] and Sun *et al.* [22] separate images into high and low frequency parts by analyzing the morphological and structural information of rain images. Luo *et al.* [17] separate rain streaks and background scene by discriminative sparse coding method. In addition, Gaussian mixture models (GMM) [9], [16] are used to decompose the rainy image into background and rain streaks layers. Low rank models are also used to separate the input image into the different layers in [2], [3], [26]. Zhang and Patel [13] propose a novel idea and try to recover the rain-free image by nonlocal means filter. Although these methods can detect and remove rain streaks, their main limitation is over-smoothing the image details since a lot of texture and fine structure information belongs to the high frequency part.

Recent approaches adopt deep learning and achieve notable success in single image de-raining. Fu *et al.* [4] introduce a model to predict the residual rain streaks using the decomposed high frequency part as input. Yang *et al.* [25] present a deep recurrent model with a dilated network to detect and remove rain streaks iteratively. Zhang *et al.* [27] propose a density classifier and combine the predicted label with the features of a multi-stream network for de-raining. Li *et al.* [15] integrate deep convolutional and recurrent neural networks to remove rain streaks in a multi-stage manner.

As we mentioned above, all of these methods predict the residual rain streaks and neglect semantic background information. Additionally, they do not involve attention in the network.

III. APPROACH

A rain image O is commonly formulated as the linear combination of the rain-free background B and rain streaks R layers as follows:

$$O = B + R. \tag{1}$$

We aim to estimate both two layers simultaneously in the same network. Below we discuss the detail.

A. NETWORK DESIGN

The pipeline of the proposed method is shown in Fig. 2. Given an input rainy image O_{in}, our network predicts the

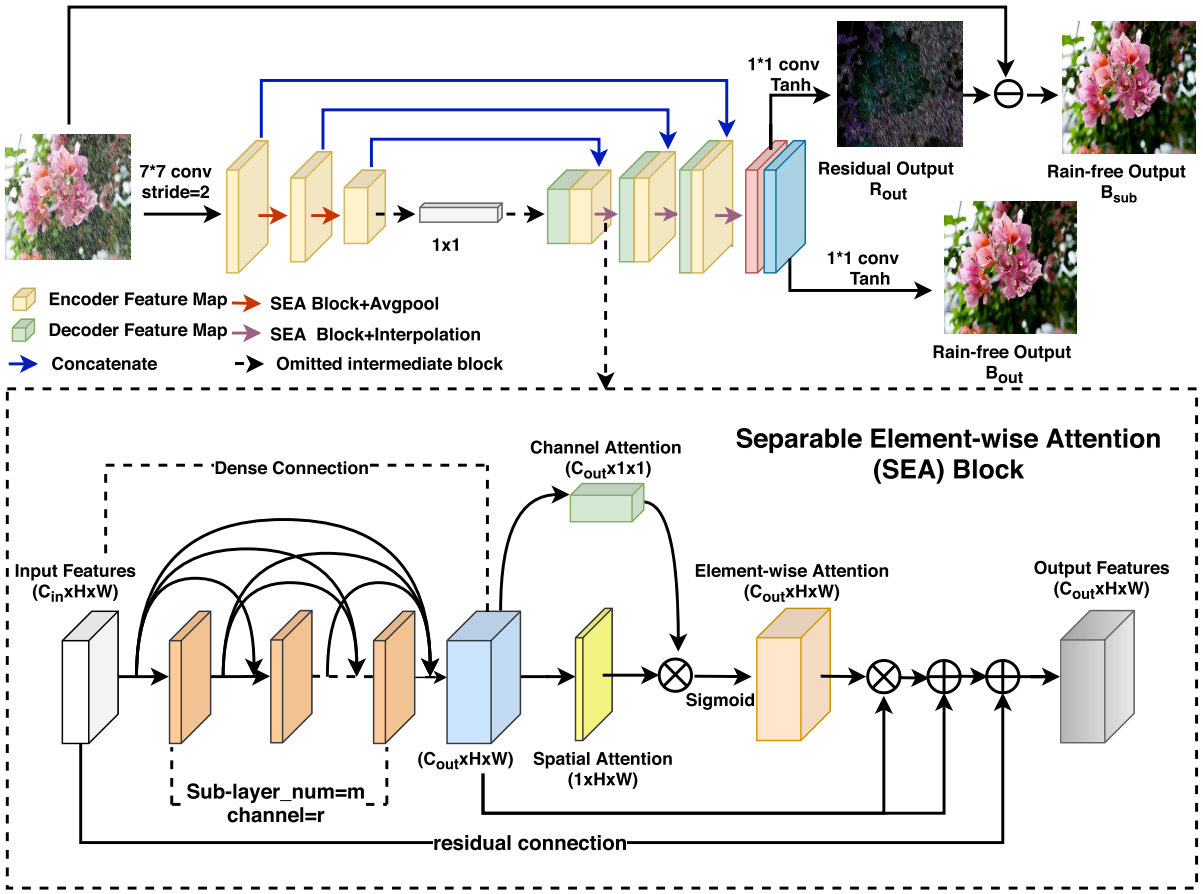


FIGURE 2. The pipeline of the proposed method and detailed structure of the Separable Element-wise Attention Block.

residual rain streak image R_{out} and the rain-free image B_{out}. By subtracting R_{out} from O_{in}, we can obtain another indirect rain-free image B_{sub}.

Our network has a plain encoder-decoder architecture. In each resolution level of the encoder and decoder except the outermost layer, we replace the single convolution with the proposed Separable Element-wise Attention (SEA) block to enrich feature representations. Average pooling is used as the downsampling operation and bilinear interpolation is used for upsampling. Skip layers concatenate the feature maps of the encoder to the feature maps of the same resolution in the decoder before feeding it to the next block.

To cope with the joint estimation of rain streaks and background, we output task-dependent features in the last layer. In particular, the last feature maps are separated into two parts. The first part corresponds to the rain streaks residual, while the other part generates a rain-free background image. Unlike traditional multi-task learning that shares all the features and uses them to output the final results at the same time, we explicitly coordinate the corresponding features of two tasks. This is able to avoid the imbalance of feature maps for two outputs and enforce the responsibility of each part that reduces the information interference at the final prediction. Although we share all the features except the last layer that generates two outputs, the entire network is governed to produce two independent features. This one-to-many supervision encourages interactions between two substantially different tasks within the network, leading to diverse and rich features representations.

B. SEPARABLE ELEMENT-WISE ATTENTION

Rain streaks distribution is of great importance to either removing rain streaks or estimating background. Intuitively, this information is modeled as the spatial attention to govern network training. However, each map of the high-dimensional features is substantially different from each other, and they may correspond to different objectives that cannot be unified using a single spatial attention map. Directly computing the element-wise attention for all the convolutional blocks leads to high computational costs. Inspired by the separable bilateral filter [18] in the signal processing area, we propose the Separable Element-wise Attention to the network.

As shown in the bottom part of Fig. 2, the proposed Separable Element-wise Attention block is mainly composed of two parts. The first part is a dense connection module [10], which propagates the output of each convolutional layer to subsequent convolutional layers within the block, promoting the information and gradient flow.

The second part of the SEA block is the proposed element-wise attention module. This module calculates the channel attention $A_c(x_m) \in \mathbb{R}^C$ and spatial attention $A_s(x_m) \in \mathbb{R}^{H \times W}$

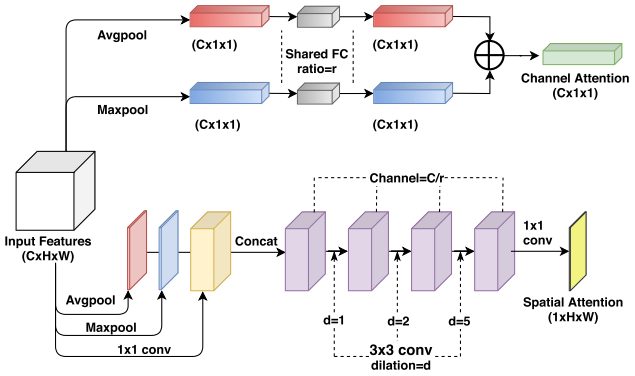


FIGURE 3. The architecture of our spatial and channel attention sub-modules.

of the input feature maps $x_m \in \mathbb{R}^{C\times H\times W}$ in different branches separately. Then these two attentions are expanded to the same size as x_m and then multiplied together to generate a 3D attention volume $A(x_m) \in \mathbb{R}^{C\times H\times W}$. In this way, the element-wise attention can be obtained by the spatial and channel attention modules, and all the feature elements can be focused or suppressed during the training of the network. The detailed architecture of our spatial and channel attention modules are shown in Fig. 3. The ins and outs are specifically made in the following passage.

1) CHANNEL ATTENTION

Our channel attention focuses on the relation between different channels, aiming to assign higher weights to those important feature maps. To reduce the computational complexity, we aggregate the spatial information by global average pooling and global max pooling, encoding the input feature maps softly into two vectors $\{V_c^{avg}, V_c^{max}\} \in \mathbb{R}^{C\times 1\times 1}$. Then both vectors are fed to a shared fully connected (FC) layer and the outputs are added together to obtain the final attention. The objective of this strategy is to consider both the global and local information of feature maps. To reduce the number of parameters, there is only one hidden layer and the number of neurons in the hidden layer is set to C/r, where r is a reduction ratio.

2) SPATIAL ATTENTION

Different from the channel attention module, the spatial attention focuses on the relation between different locations on the feature maps, aiming to emphasize the spatially discriminative information. Similar to the channel attention, we first apply average pooling and max pooling on the input feature maps along the channel axis, which obtains two maps $\{M_s^{avg}, M_s^{max}\} \in \mathbb{R}^{1\times H\times W}$. In addition, we apply an 1×1 convolution on the input feature maps and obtain another map $M_s^{1\times 1} \in \mathbb{R}^{(C/r-2)\times H\times W}$, where r is the same reduction ratio as in the channel attention. We concatenate these maps and feed it to three 3×3 dilated convolutions and then a 1×1 convolutions to get the final spatial attention map $A_s(x_m) \in \mathbb{R}^{H\times W}$. We follow [23] to set the dilated rates of those three dilated convolution layers as 1, 2, 5, respectively. It can avoid sampling in the checkerboard pattern that skips pixels within the convolutional regions. At the same time, as its well-known properties, dilated convolution can compute attention values with a large receptive field.

At the end of the SEA block, we utilize residual connection directly from input to output. If the number of channels is different, we use a 1×1 convolution on the input feature maps to fit the channel number. Residual connections can avoid the notorious problem of gradient vanishing or exploding [7]. At the final output layer of the network, we use half of the feature maps for rain-free image prediction and another half for rain streak residual prediction. The final output is obtained by averaging two rain-free images B_{sub} and B_{out}.

C. TRAINING OBJECTIVES AND DETAILS

We use four loss functions to optimize the proposed network.

1) PIXEL LOSS

Given the ground truth rain-free image B_{gt}, the pixel loss is defined as follows:

$$\begin{aligned}\mathfrak{L}_p &= \frac{\|B_{sub}-B_{gt}\|_1}{N_{gt}} + \frac{\|B_{out}-B_{gt}\|_1}{N_{gt}} \\ &= \frac{\|O_{in}-R_{out}-B_{gt}\|_1}{N_{gt}} + \frac{\|B_{out}-B_{gt}\|_1}{N_{gt}},\end{aligned} \tag{2}$$

where $N_{gt} = C\times H\times W$ denotes the number of pixels in the ground truth. Pixel loss measures the accuracy of each pixel between the network outputs and their corresponding ground truth by L_1 distance.

2) PERCEPTUAL AND STYLE LOSSES

We introduce perceptual and style losses [6] into the network, which are used to measure the content and style differences between two images. The reconstructed image should be close to the ground truth image not only in pixel-level, but also in high- and semantic-level. We first define the perceptual loss:

$$\mathfrak{L}_{perc} = \sum_p \frac{\left\|\Phi^p_{B_{sub}} - \Phi^p_{B_{gt}}\right\|_1}{N_{\Phi^p_{B_{gt}}}} + \sum_p \frac{\left\|\Phi^p_{B_{out}} - \Phi^p_{B_{gt}}\right\|_1}{N_{\Phi^p_{B_{gt}}}}, \tag{3}$$

where $\Phi^p_{B_s}$ represents the feature maps at p-th layer of the ImageNet-pretrained VGG-16 model. Pool1, pool2, and pool3 layers are selected in our method. We use L_1 distance to compute the corresponding feature maps between both B_{out} and B_{sub} and the ground truth B_{gt}.

Style loss is also calculated based on the projected VGG feature maps, but it is actually calculating the L1 distance of the Gram matrix of each VGG feature maps:

$$\mathfrak{L}_{style_{B'}} = \sum_p \frac{\left\|K_p((\Phi^p_{B_{sub}})^T(\Phi^p_{B_{sub}}) - (\Phi^p_{B_{gt}})^T(\Phi^p_{B_{gt}}))\right\|_1}{C_pC_p}, \tag{4}$$

$$\mathfrak{L}_{style_B} = \sum_p \frac{\left\|K_p((\Phi^p_{B_{out}})^T(\Phi^p_{B_{out}}) - (\Phi^p_{B_{gt}})^T(\Phi^p_{B_{gt}}))\right\|_1}{C_pC_p}. \tag{5}$$

Here, feature maps Φ^p are expanded to the matrix of size $C_p \times (H_p W_p)$, which then generates a $C_p \times C_p$ Gram matrix. It represents the autocorrelation of each feature map. K_p is a normalization factor with value $1/C_p H_p W_p$.

3) EDGE LOSS

Due to the influence of rain streaks, the edges of the background are discontinuous or blurred. Using pixel loss only cannot guarantee edges correctness. To this end, we extract edges for the outputs and ground truth using Sobel operator, and then compute their L1 distances to enforce correct edges:

$$\mathfrak{L}_{edge} = \frac{\left\| f_s(B_{sub}) - f_s(B_{gt}) \right\|_1}{N_{gt}} + \frac{\left\| f_s(B_{out}) - f_s(B_{gt}) \right\|_1}{N_{gt}}, \quad (6)$$

where $f_(s)$ denotes the Sobel operator.

Then the total loss is the summation of the above losses.

$$\mathfrak{L}_{total} = \lambda_r \mathfrak{L}_p + \lambda_p \mathfrak{L}_{perc} + \lambda_s \mathfrak{L}_{style} + \lambda_e \mathfrak{L}_{edge}, \quad (7)$$

where the λ_* denotes the weight of the corresponding loss term.

IV. EXPERIMENTS

In this section, we evaluate our proposed method on both synthetic and real collected rainy data. We also make a comparison with other state-of-the-art methods on these datasets.

A. EXPERIMENT SETTINGS

1) TRAINING SETTINGS

We first describe the hyper-parameters used in our model. For each SEA block, the growth-rate, which is the feature number of sub-convolutional layer [10] in dense connection part, is set to 32, and the number of sub-convolutional layers in dense part is 8 for the innermost 9 blocks and 4 for the remaining. This setting is based on the resolution and feature numbers in each level. Furthermore, the reduction ratio r in the attention module of each SEA block is set to 16 according to the analysis in [8]. The weight of each loss item are as follows: $\lambda_r = 500$, $\lambda_p = 1.5$, $\lambda_s = 250$, $\lambda_{edge} = 1$. Within these weights, the pixel loss contributes the most and reconstructs the image structure at the beginning of training, while the edge loss, perceptual loss and style loss are used to further refine the image. The input image is resized to 512×512 and the batch size is 5. Our method is implemented with the Pytorch framework on an NVIDIA 1080 Ti GPU. We use the SGD optimizer with momentum equals 0.9 and the initial learning rate is set to 2×10^{-4}. The learning rate decreases linearly from the 50th epoch to the 300th epoch.

2) DATASETS

In order to evaluate the de-raining ability of our method, we utilize three synthesis datasets in the experiments. The first one is the Rain800 dataset [28], which includes 700 images as the training set and 100 images as the testing set. The second one is the Rain200 dataset [25] (extended from Rain100), including two subsets representing: 1) heavy rain set (Rain200H) that is synthesized with five types of streaks, and 2) light rain set (Rain200L) that is synthesized with only one streak type. Each set contains 1,800 images for training and 200 images for testing. In the experiment, we train a model based on the training set of Rain200H and evaluate it with both testing set of Rain200H and Rain200L. We exclude Rain200L from the training set since the rain streak patterns of Rain200L are included in Rain200H, and in this way we can evaluate the generalization ability of the methods. The third dataset is the DIDMDN dataset, including one training set and two testing sets. The training set consists of 12,000 images, synthesized by adding three different densities (light, medium, heavy) of rain streaks to 4,000 rain-free images. The first testing set, denoted as *DID-Test1*, is constructed in a similar way to the training set and contains 1,200 images in total. The second one is obtained by randomly sampling 1,000 images from the synthetic dataset provided by Fu *et al*. [4], which is also utilized to test the generalization capability, denoted as *DID-Test2*. Since the proposed model predicts the rain-free image as one of the output, in order to avoid overfitting caused by predicting the same rain-free image multiple times, we choose the same number of training images with different backgrounds from three density levels, to build a new training dataset, with 4,000 images in total for our experiment.

For real-world dataset, we use the real-world rainy images provided by Yang *et al*. [25] and Zhang *et al*. [28]. We also collect some photos from the web, most of which are captured in street and city scenes, which are more consistent with the application scenario of the de-raining task.

3) MEASUREMENT AND COMPARISON

We evaluate the de-raining methods by the commonly used peak signal to noise ratio (PSNR) [11] and Structure Similarity Index (SSIM) [24] metrics. For real images, we mainly present the qualitative comparison and user study (see our supplementary materials), due to the absence of corresponding ground-truth. We compare our proposed method with several state-of-the-art CNN-based methods, including DDN [4], JORDER [25], DID-MDN [27], SCAN and its recurrent version (RESCAN) [15].

B. EVALUATION ON SYNTHETIC DATASET

For a comprehensive evaluation, we train one model on each of the training sets mentioned above and test the model with the corresponding testing sets. For a fair comparison, we fine-tune their models on the corresponding training sets with the same number of epochs as ours, except the JORDER method that only provides a model trained on Rain200H and no training details.

The quantitative results of PSNR and SSIM are shown in Table 1. We can see that our method performs better than all the other deep learning based methods. Although the latest RESCAN method achieves the best result among previous methods on almost all synthesized datasets, it performs worse than the DID-MDN method on restoring structure

TABLE 1. Quantitative results of each method in (PSNR/SSIM).

Trainset	Rain800	Rain200H		DID-MDN	
Testset	Rain800	Rain200H	Rain200L	DID-Test1	DID-Test2
DDN [4]	23.16 / 0.8451	23.38 / 0.8143	24.22 / 0.8716	27.86 / 0.8990	24.17 / 0.8241
JORDER [25]	21.13 / 0.8103	24.37 / 0.8658	29.16 / 0.9526	25.50 / 0.8717	24.08 / 0.8408
DID-MDN [27]	23.57 / 0.8714	23.43 / 0.8590	26.44 / 0.9218	27.91 / 0.9150	25.09 / 0.8578
SCAN [15]	25.10 / 0.8644	24.48 / 0.846	28.01 / 0.9398	31.14 / 0.9318	24.83 / 0.8306
RESCAN [15]	25.54 / 0.8807	25.97 / 0.8950	30.27 / 0.9551	32.01 / 0.9448	25.12 / 0.8340
Proposed	**25.62 / 0.8880**	**26.67 / 0.9007**	**31.15 / 0.9602**	**32.52 / 0.9512**	**26.40 / 0.8680**

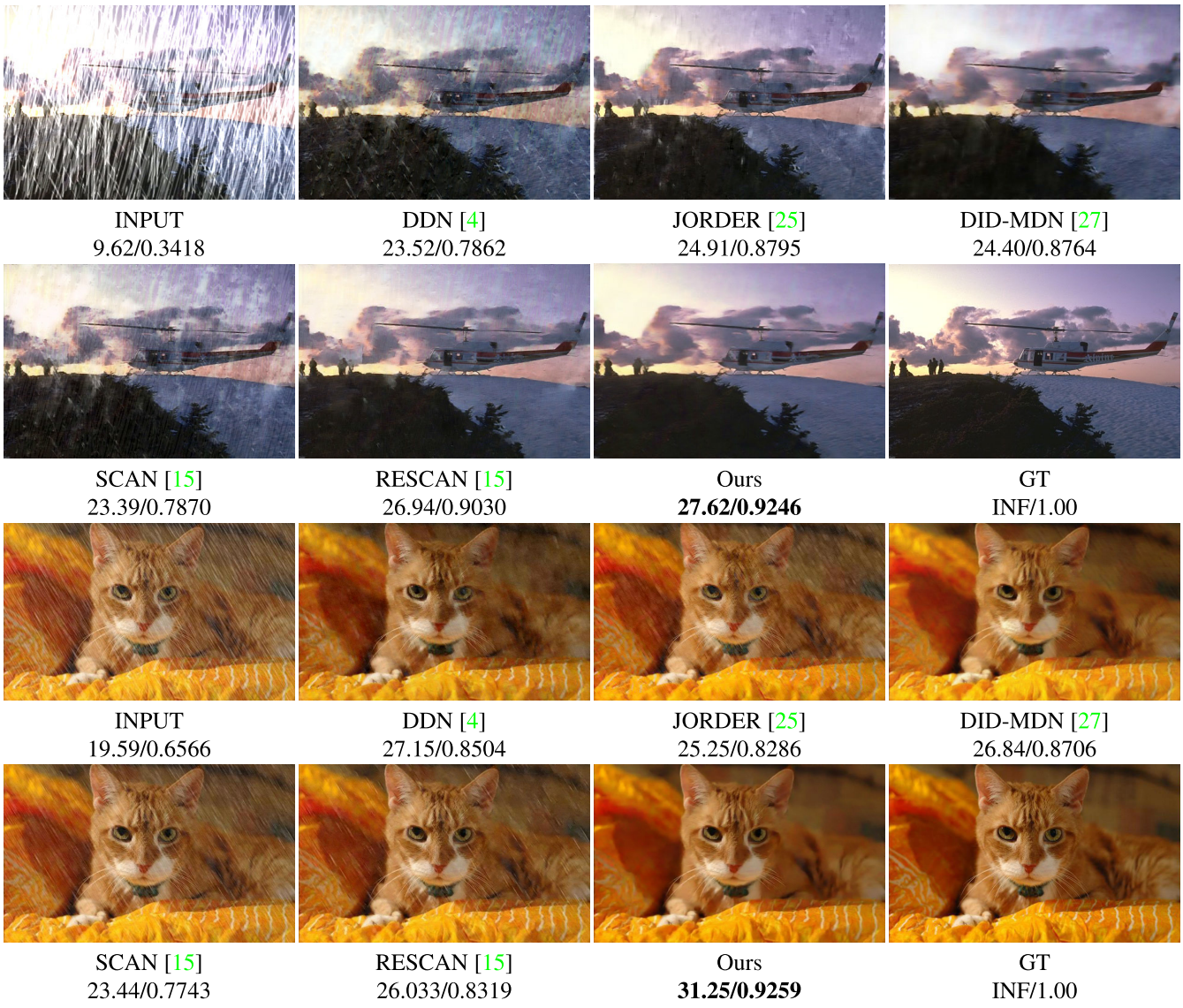


FIGURE 4. Results and evaluations of each method on synthetic images. The first image is chosen from Rain200H testing set. The second image is chosen from *DID-Test2*.

information (SSIM) on *DID-Test2*, implying that it cannot well generalize to unseen rain streaks. In contrast, our method performs better than previous methods on both *DID-Test1* and *DID-Test2*. In addition, the rainy image is only processed once without combining RNN (as used in RESCAN) or other extra refinement networks (as used in DID-MDN).

Fig. 4 shows the visualization results of all methods. The first image is chosen from the testing set of Rain200H, which is the most difficult dataset since the original images are mostly destroyed. We can see that both DID-MDN and RESCAN methods are able to well remove the rain streaks and restore the color of the original image. However, their results contain distortions and unsmooth regions on the background and details of objects (better zoom-in on the digital version). In contrast, our method performs well for both detail recovery and background smoothing. The second image is

TABLE 2. Validation of proposed strategy (predicting residual rain-streaks and rain-free results at the same time) in (PSNR/SSIM).

Methods	DID-Test1	DID-Test2
B_{sub} (Rain-Streak Only)	32.14/0.9445	25.84/0.8545
B_{out} (Rain-free Only)	31.86/0.9412	25.62/0.8497
B_{sub} (w/o Task-dependent)	31.97/0.9432	25.73/0.8510
B_{out} (w/o Task-dependent)	31.81/0.9402	25.66/0.8493
B_{sub} (Ours)	**32.46/0.9507**	**26.32/0.8677**
B_{out} (Ours)	**32.34/0.9498**	**26.12/0.8678**

chosen from the *DID-Test2* to show the generalization of each model. We can see that on this image, most methods including the recent RESCAN are not able to completely remove the rain streaks. Although there are no obvious white rain streaks on the result of DID-MDN method, there exist many misplacements and distortions in rain streaks shape, which results in low PSNR and SSIM. Compared to other methods, our proposed model is able to remove the unseen rain streaks and well restore the structure and intensity of the original image.

C. EVALUATION ON REAL-WORLD DATASET

The final goal of the de-raining task is to apply in real-world scenes. As a result, we perform another evaluation on rainy images captured in the real-world. For a fair comparison, we select the model trained by Rain200H for each method since Rain200H can further enhance the robustness of the network as mentioned in [25]. Example results on real-world de-raining are shown in Fig. 5. It can be observed that our proposed method can well remove the rain streaks and does not break the original structure. The result of the second image shows that our proposed method even performs well on removing rain-drop form and watermark form rain streaks, while other methods fail to handle such types of rain streaks. To further evaluate the proposed method on real-world data, we conduct a user study in the supplementary materials.

D. ABLATION STUDY

In this section, we study the effectiveness of each term/module in our model. To better test the fitting and generalization ability of each module, we train and test on the DID-MDN dataset.

Firstly, we validate the effectiveness of our main strategy, which simultaneously estimates the rain-free image and the residual rain streak image. In this ablation study, we train three additional models as shown in Table 2. The ''Rain-Streak Only'' refers to the model only predicting the residual rain streak image (and subtracting by rainy image to get the rain-free background). ''Rain-free Only'' refers to the model only predicting the rain-free background. ''w/o Task-dependent'' refers to the model predicting two outputs using the last feature maps without separating them into task-dependent features. In addition, we use the notation B_{sub} and B_{out} in Sec. III to indicate the different rain-free outputs.

From the result in Table 2, we can see that when jointly predicting two outputs without task-dependent features, their performances decrease compared with predicting only one output. This implies that simply adding an extra prediction task in the network cannot benefit removal performance. However, when predicting results with task-dependent feature maps, our results obtain a significant improvement compared with the single output, even though the number of feature maps for each output is reduced by half. It reveals that the motivation of our method which uses the rain-free background as one of the outputs provides more information and enables better interaction between different kinds of features.

TABLE 3. Validation of each module in (PSNR/SSIM).

Methods	DID-Test1	DID-Test2
Unet	30.28/0.9285	25.59/0.8505
SEA	32.42/0.9493	26.14/0.8613
SEA+edge loss	32.48/0.9498	26.24/0.8652
SEA+edge+perceptual&style	**32.52/0.9512**	**26.40/0.8680**

TABLE 4. Voting results of DDN [4], DIDMDN [27], JORDER [25], RESCAN [15] and our method on real images. 'Selected' represents the number of images obtain the most votes.

Guideline	Best Derain and Detail	
Measure	Voted	Selected
Not Sure	9	0
DDN	92	2
DID	171	5
JORDER	69	0
RESCAN	67	0
Our	**492**	**23**

Next, we perform experiments to compare the effectiveness of the element-wise attention, perceptual and style losses, and edge loss. The results are shown in Table 3. It can be observed that each module and loss has a positive effect on the removal performance and generalization ability of the model. Note the proposed Separable Element-wise Attention (SEA) block significantly boosts the performance.

E. USER STUDY ON REAL RAINY IMAGES

To further evaluate the effectiveness of our proposed method, we conduct a user study on 30 real rainy images. These images are collected to simulate the actual usage of a deraining system, they are captured in close-up shots, pedestrians, buildings in heavy rains, or images with a black background and strong white light source to simulate the rainy scene at night. We compare our method with DDN [4], JORDER [25], DID-MDN [27], and RESCAN [15]. We invite 30 people to participate in the survey to choose the one that is the best rain-free and most natural image after the deraining process. Results are shown in Table 4, where ''Voted'' represents the total number of votes for the corresponding method, and ''Selected'' represents the number of images obtaining the most votes. We can see that our proposed method obtains the most votes, and DID-MDN ranks the second. It reveals that although RESCAN method shows good performance on the training set, DID-MDN generalizes better in real-world scenes. On the contrary, the proposed method performs the best on both the synthetic scenes and real-world scenes.

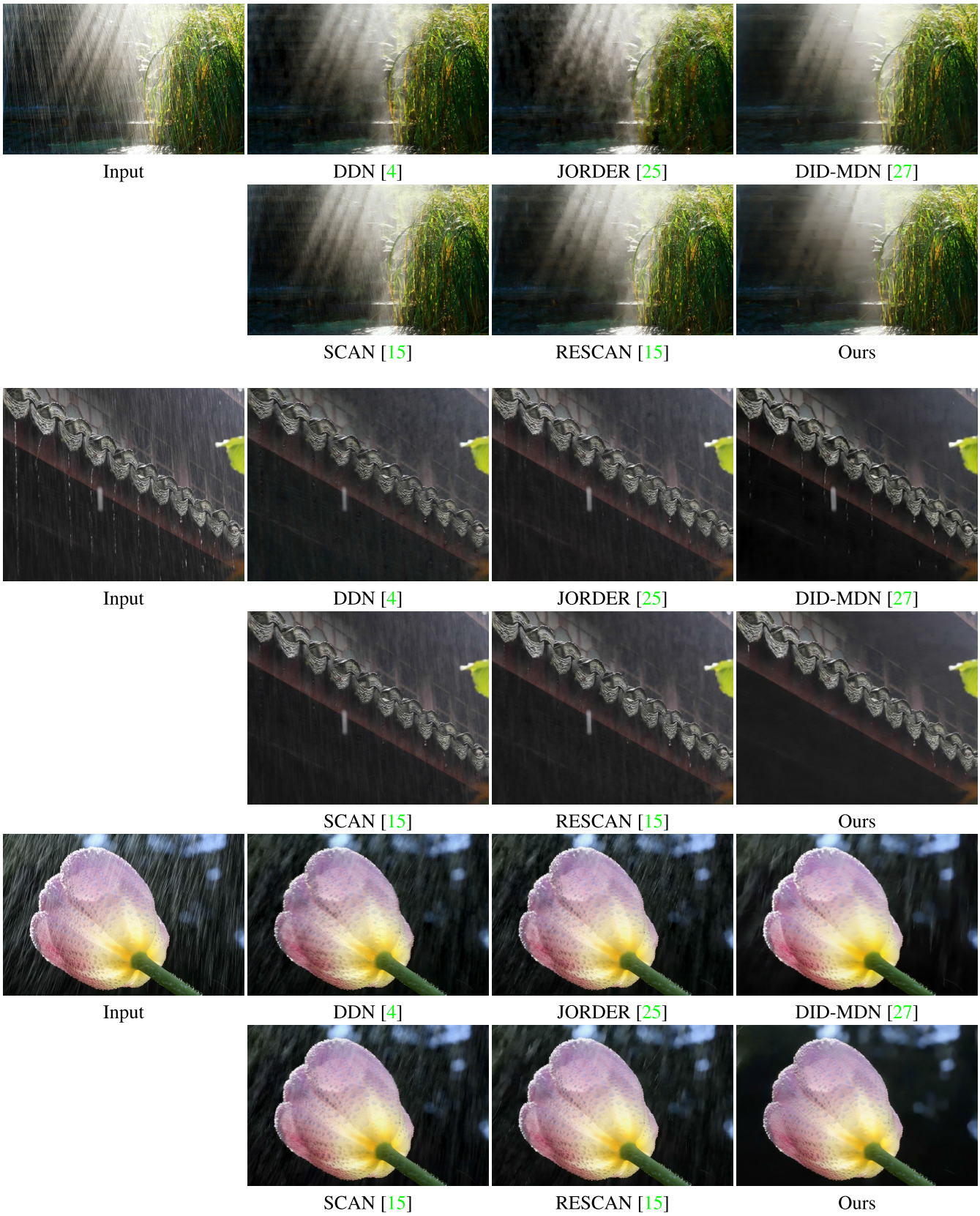


FIGURE 5. Qualitative results of each method on real-world images.

V. CONCLUSION

We propose a coupled rain streak and background estimation network with Separable Element-wise Attention modules. It addresses the problem of rain streaks removal from two aspects. First, we delve into the problem of the estimation for rain streak and rain-free background, and these

two tasks are bridged by task-dependent features. Second, we present a Separable Element-wise Attention module to explore the rain streaks distribution in all the layers of the network. It is achieved by two attention modules: the spatial and channel attention modules. All existing convolutional blocks can inject such element-wise attention on the fly. Extensive experiments demonstrate that the proposed method achieves superior performance against state-of-the-art methods, both quantitatively and qualitatively. The proposed Separable Element-wise Attention is a general framework, which we believe to be effective in other vision tasks.

REFERENCES

[1] N. Brewer and N. Liu, "Using the shape characteristics of rain to identify and remove rain from video," in *Proc. Joint IAPR Int. Workshops Stat. Techn. Pattern Recognit. (SPR) Struct. Syntactic Pattern Recognit. (SSPR)*, 2008, pp. 451–458.

[2] Y. Chang, L. Yan, and S. Zhong, "Transformed low-rank model for line pattern noise removal," in *Proc. ICCV*, 2017, pp. 1726–1734.

[3] Y.-L. Chen and C.-T. Hsu, "A generalized low-rank appearance model for spatio-temporally correlated rain streaks," in *Proc. ICCV*, 2013, pp. 1968–1975.

[4] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *Proc. CVPR*, 2017, pp. 1715–1723.

[5] K. Garg and S. K. Nayar, "Detection and removal of rain from videos," in *Proc. CVPR*, 2004, p. I.

[6] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. CVPR*, Jun. 2016, pp. 2414–2423.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[8] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," in *Proc. CVPR*, 2018, pp. 7132–7141.

[9] D.-A. Huang, L.-W. Kang, Y.-C.-F. Wang, and C.-W. Lin, "Self-learning based image decomposition with applications to single image denoising," *IEEE Trans. Multimedia*, vol. 16, no. 1, pp. 83–93, Jan. 2014.

[10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. CVPR*, 2017, pp. 4700–4708.

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

[12] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1742–1755, Apr. 2012.

[13] J.-H. Kim, C. Lee, J.-Y. Sim, and C.-S. Kim, "Single-image deraining using an adaptive nonlocal means filter," in *Proc. ICIP*, 2013, pp. 914–917.

[14] J.-H. Kim, J.-Y. Sim, and C.-S. Kim, "Video deraining and desnowing using temporal correlation and low-rank matrix completion," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2658–2670, Sep. 2015.

[15] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha, "Recurrent squeeze-and-excitation context aggregation net for single image deraining," in *Proc. ECCV*, 2018, pp. 254–269.

[16] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain streak removal using layer priors," in *Proc. CVPR*, 2016, pp. 2736–2744.

[17] Y. Luo, Y. Xu, and H. Ji, "Removing rain from a single image via discriminative sparse coding," in *Proc. ICCV*, 2015, pp. 3397–3405.

[18] S. Paris and F. Durand, "A fast approximation of the bilateral filter using a signal processing approach," in *Proc. ECCV*, 2006, pp. 568–580.

[19] R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu, "Attentive generative adversarial network for raindrop removal from a single image," in *Proc. CVPR*, 2018, pp. 2482–2491.

[20] W. Ren, J. Tian, Z. Han, A. Chan, and Y. Tang, "Video desnowing and deraining based on matrix decomposition," in *Proc. CVPR*, 2017, pp. 4210–4219.

[21] V. Santhaseelan and V. K. Asari, "Utilizing local phase information to remove rain from video," *Int. J. Comput. Vis.*, vol. 112, no. 1, pp. 71–89, Mar. 2015.

[22] S.-H. Sun, S.-P. Fan, and Y.-C. F. Wang, "Exploiting image structural similarity for single image rain removal," in *Proc. ICIP*, 2014, pp. 4482–4486.

[23] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *Proc. WACV*, 2018, pp. 1451–1460.

[24] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[25] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Deep joint rain detection and removal from a single image," in *Proc. CVPR*, 2017, pp. 1357–1366.

[26] H. Zhang and V. M. Patel, "Convolutional sparse and low-rank coding-based rain streak removal," in *Proc. WACV*, 2017, pp. 1259–1267.

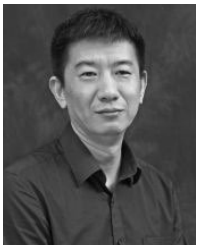
[27] H. Zhang and V. M. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *Proc. CVPR*, 2018, pp. 695–704.

[28] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," 2017, *arXiv:1701.05957*. [Online]. Available: https://arxiv.org/abs/1701.05957

[29] X. Zhang, H. Li, Y. Qi, W. K. Leow, and T. K. Ng, "Rain removal in video by combining temporal and chromatic properties," in *Proc. ICME*, 2006, pp. 461–464.

YINJIE TAN received the B.Sc. degree from the School of Computer Science and Engineering, South China University of Technology, in 2018, where he is currently pursuing the master's degree. His research interests include computer vision, image processing, and deep learning.

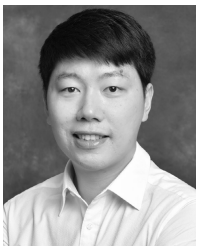
QIANG WEN received the B.Eng. degree from the School of Information Science and Engineering, Central South University, in 2018. He is currently pursuing the master's degree with the School of Computer Science and Engineering, South China University of Technology. His research interests include computer vision, image processing, and deep learning.



JING QIN received the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong, in 2009. He has been an Assistant Professor with The Hong Kong Polytechnic University, since 2016. His research interests include visualization, human–computer interaction, and deep learning.

JIANBO JIAO received the Ph.D. in computer science from the City University of Hong Kong. He is currently a Postdoctoral Researcher with the Department of Engineering Science, University of Oxford, and a member of the Biomedical Image Analysis (BioMedIA) Group and the Visual Geometry Group (VGG). His research interests include computer vision and machine learning.

GUOQIANG HAN received the B.Sc. degree from Zhejiang University, Hangzhou, China, in 1982, and the master's and Ph.D. degrees from Sun Yat-sen University, Guangzhou, China, in 1985 and 1988, respectively. He was the Dean of the School of Computer Science and Engineering. He is currently a Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou. He has published over 100 research articles. His current research interests include multimedia, computational intelligence, machine learning, and computer graphics.



SHENGFENG HE (Member, IEEE) received the B.Sc. and M.Sc. degrees from the Macau University of Science and Technology, and the Ph.D. degree from the City University of Hong Kong. He was a Research Fellow with the City University of Hong Kong. He is currently an Associate Professor with the School of Computer Science and Engineering, South China University of Technology. His research interests include computer vision, image processing, computer graphics, and deep learning.