Semi-supervised Image Deraining Using Knowledge Distillation

Xin Cui, Cong Wang, Dongwei Ren, Yunjin Chen, Pengfei Zhu

***Abstract*—Image deraining has achieved considerable progress based on supervised learning with synthetic training pairs, but is usually limited in handling real-world rainy images. Although semi-supervised methods are suggested to exploit real-world rainy images when training deep deraining models, their performances are still notably inferior. To address this crucial issue, this work proposes a semi-supervised image deraining network with knowledge distillation (SSID-KD) for better exploiting real-world rainy images. In particular, the consistency of feature distribution of rain streaks extracted from synthetic and real-world rainy images is enforced by adopting knowledge distillation. Moreover, as for the backbone in SSID-KD, we propose the multi-scale feature fusion module and the pyramid fusion module to better extract deep features of rainy images. SSID-KD can relieve the problem of over-deraining or under-deraining for real-world rainy images, while it can keep comparable performance with supervised deraining methods on several benchmark datasets. Extensive experiments on both synthetic and real-world rainy images have validated that our SSID-KD not only can achieve better deraining results than existing semi-supervised deraining methods but also are quantitatively comparable with state-of-the-art supervised deraining methods. Benefiting from the well exploration of real-world rainy images, our SSID-KD can obtain more visually plausible deraining results. The source code and trained models are publicly available at https://github.com/cuiyixin555/SSID-KD.**

***Index Terms*—Single Image Deraining, Knowledge Distillation, Semi-supervised Learning**

I. Introduction

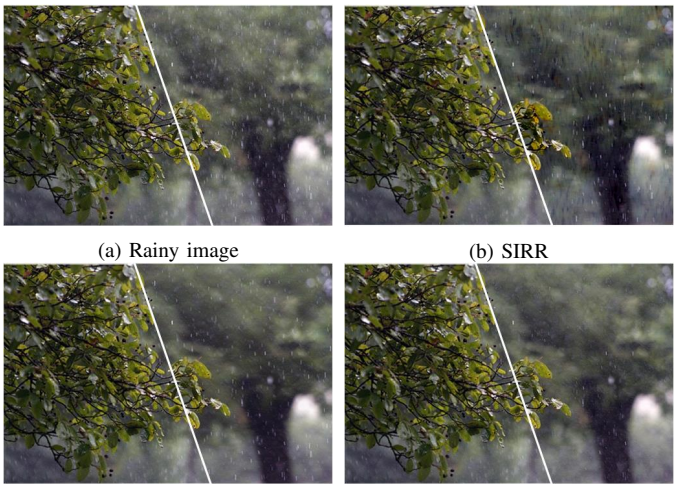
Images captured in the rainy outdoor environments usually suffer from serious degradations, *e.g.*, rain streaks can obscure the people and objects in the captured scene. With long-term rainwater accumulation, the rain curtain effect is formed, which not only degrades human perception but also yields significant performance drops in the following-up tasks *e.g.*, object detection, semantic segmentation, *etc*. Therefore, single image deraining lays the foundation of middle and high-level visual tasks and has wide applications in object detection, outdoor recognition, and automatic driving [3], [4], when facing rainy weather conditions. Generally, a rainy image O can be expressed as the linear sum between a background image B and a rain streak layer R,

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

Image deraining aims to clearly remove rain streaks, while recovering texture details of background images.

In pioneering works, the image deraining task is generally regarded as an optimization problem by exploiting the priors of rain streaks and clean images [5]–[14]. These methods usually assume rain streaks as a high-frequency signal and the background layer as a low-frequency signal, but their deraining results are not satisfactory due to the complicated composition patterns in real-world rainy images. These conventional optimization methods are also very time-consuming. With the rapid development of deep learning in recent years, learning-based deraining methods have achieved great progress [15]–[24], [24]–[41], showing significant performance gains over conventional optimization methods. In these methods, deep networks are employed to automatically extract rain streaks, where massive training pairs can be exploited to learn the mappings from rainy images to clean images [16]. Under a supervised learning framework, many subsequent methods enhance the deraining effects from the aspects of designing network structures.

Albeit significant progress on benchmark datasets, supervised deraning methods heavily rely on paired training images, which are usually synthesized by assuming the additive liner



(a) Rainy image (b) SIRR (c) Syn2Real (d) SSID-KD

Fig. 1. Comparison of semi-supervised deraining methods, *i.e.*, SIRR [1], Syn2Real [2] and our SSID-KD, on real-world rainy image.

This work was supported by National Natural Science Foundation of China under Grant No. 62172127 and Alibaba Group through Alibaba Innovative Research Program. (Corresponding author: Dongwei Ren)

X. Cui and C. Wang contributed equally to this work.

X. Cui and P. Zhu are with the College of Intelligence and Computing, Tianjin University, Tianjin 300350, China (e-mail: 2019216101@tju.edu.cn, zhupengfei@tju.edu.cn).

C. Wang is with Department of Computing, The Hong Kong Polytechnic University (e-mail: supercong94@gmail.com).

D. Ren is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: rendongwei-hit@gmail.com).

Y. Chen is with Alibaba Cloud, Hangzhou, China (email: chenyunjin_-nudt@hotmail.com).

composition pattern. However, rain streaks and their composition patterns in real-world rainy images are much more complicated, making supervised deraining methods suffer from poor generalization ability. For example, deraining models trained with Rain200H [16], a synthetic dataset with heavy rain streaks, usually cannot obtain visually satisfactory deraining results when handling real-world rainy images. This is because there is a big distribution gap between synthetic and real-world rainy images, yielding incomplete rain removal or over smoothed background images. Then, unsupervised learning and semi-supervised learning are suggested to exploit real-world rainy images, aiming to improve generalization ability when handling real-world rainy images. In [42], Zhu *et al*. proposed to adopt CycleGAN [43] to exploit unpaired real rainy images. In [23], Xu *et al*. designed a new Quality-Task-Perception loss to deal with various restoration tasks including image deraining. But unsupervised deraining methods are quantitatively inferior to existing supervised deraining methods. Semi-supervised learning is another way to benefit from real rainy images. In [1], [2], transfer learning was introduced to transfer deraining models trained on synthetic images to real rainy images. These semi-supervised deraining methods may also be inferior to supervised methods in terms of quantitative metrics, and there is leeway to improve deraining visual quality.

In this paper, to address the crucial issue of existing image deraining methods, we propose a semi-supervised image deraining method using knowledge distillation (SSID-KD), as well as a novel deraining backbone network. Given the real-world rainy images without ground-truth clean images, the training of deraining model cannot be tackled by a pixel-wise loss function, and we suggest enforcing the consistency of high-dimensional rain streaks feature between synthetic and real-world rainy images by adopting knowledge distillation. Our SSID-KD model is trained with two stages. In the first stage, a teacher model is trained on paired synthetic training samples, which is actually supervised image deraining (SID). In the second stage, unpaired synthetic rainy images and real-world rainy images are taken to train the teacher network and student network, by enforcing the consistency of feature distribution of rain streaks by minimizing Kullback–Leibler Divergence loss (KL-Loss). Since real-world rainy images have no ground truths, we further introduce a Dark Channel loss (DC-Loss) as a semi-supervised loss for training student model, inspired by [44]–[46], resulting in our final SSID-KD model. As for the deraining backbone network, we propose a multi-scale feature fusion block (MFFB) and pyramid feature fusion block (PFFB), which are organized in a multi-scale framework, enabling our SSID-KD can better extract deep features from synthetic and real rainy images.

We note that different from conventional knowledge distillation, the teacher network and the student network need to share weights during semi-supervised training. Our deraining model is also updated on synthetic training data because the features of rain streaks in high dimensions between synthetic and real rainy images are very similar and can benefit each other during training. To better illustrate the point, the comparison in Fig. 2 confirms the performance of our model on three real-world datasets, *i.e*., SPA [47], Real275, IVIPC [48], [49]. We select three synthetic datasets Rain200H [16], Rain1200 [50] and Rain1400 [15] for first stage training. Given that real datasets have no ground truths, we adopt NIQE [51] as an evaluation metric. Through two stages of training, SSID-KD is better than SID, where knowledge distillation between synthetic and real rain streak features is activated on rain removal actually.

Extensive experiments have been conducted on both synthetic and real-world rainy benchmark datasets. Our SSID-KD model quantitatively outperforms not only semi-supervised deraining methods but also state-of-the-art supervised deraining methods. In particular, our SSID-KD is compared with two semi-supervised methods SIRR [1] and Syn2Real [2], as shown in Fig. 1, where the deraining performance of SSID-KD is the best in terms of visual perception.

Our contributions can be summarized from three aspects:

- We propose a simple yet effective semi-supervised deraining approach using knowledge distillation, where real-world rainy images can be better exploited to benefit the generalization ability of the deraining model.
- Two novel modules MFFB and PFFB are proposed to better extract features of rain streaks. These two components are organized in multi-scale framework to serve as backbone network for image deraining.
- Extensive experiments on synthetic and real-world rainy images have been conducted to validate that our SSID-KD is superior to both supervised and semi-supervised deraining methods.

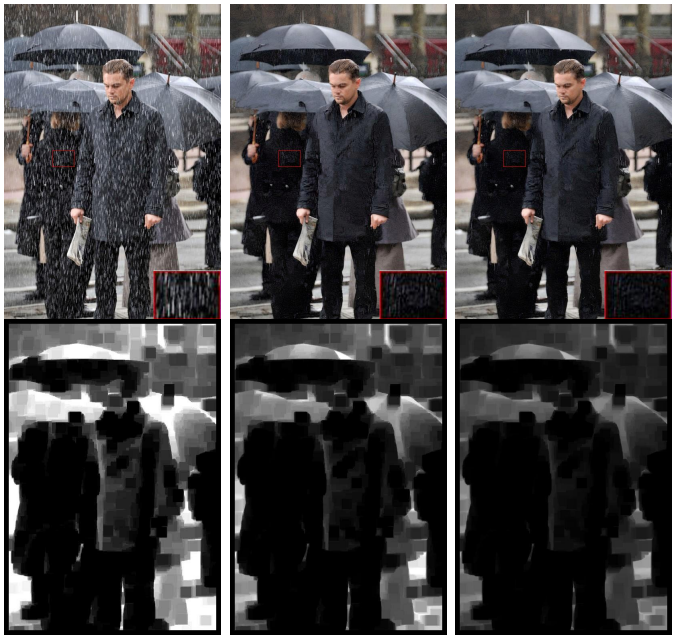
II. Related Work

In this section, we review some image deraining methods in Sec. II-A and the unpaired training strategy for Semi-supervised learning in Sec. II-B.

A. Single Image Deraining

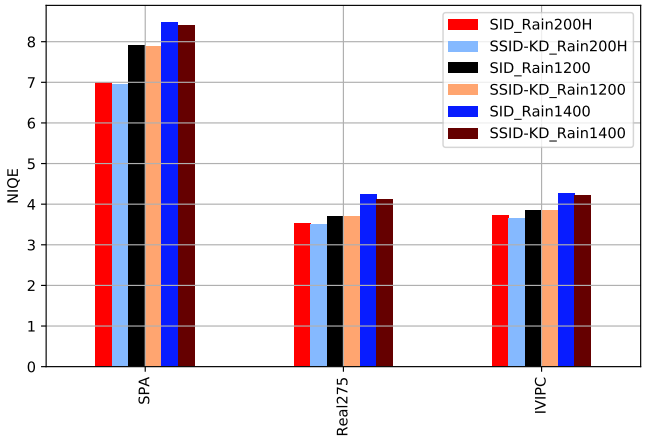
Different from video-based methods [6], [52]–[56] for rain removal, single image deraining could be more difficult due to the lack of correlated information from serial frames. In this section, we only review the single image deraining problem.

1) Prior-Based Methods: Many early works attempt to solve the problem with image priors [5], [7]–[11], [14], [33], [44]–[46], [57]–[59]. In generally, rain streaks are usually regarded as a high-frequency component; Thus, single-image rain removal can be regarded as a signals decomposition task. Kang *et al*. [12] firstly applied signal decomposition to the image deraining, which removed the high-frequency rain streak by dictionary learning. Besides, sparse coding is also widely used in image rain removal tasks. Luo *et al*. [5] proposed a sparse coding structure combined with discriminative learning strategies to separate the rain streak layer from a background image layer. The pixels of the rain streak layer is mostly composed of 0 and 1, which is viewed as a sparse matrix; Thus, the background image layer can be regarded as a low-rank matrix because it usually contains a lot of repetitive textures. In [9], Chen *et al*. proposed a low-rank representation-based method that promoted the deraining performance by taking advantage of the low-rank model. In



(a) (b) (c)

NIQE Value on 3 Real-World Datasets with SID and SSID-KD



	SPA	Real275	IVIPC
SID_Rain200H	7.0006	3.5290	3.7406
SSID-KD_Rain200H	6.9636	3.5105	3.6488
SID_Rain1200	7.9148	3.7057	3.8643
SSID-KD_Rain1200	7.8821	3.6968	3.8632
SID_Rain1400	8.4704	4.2530	4.2688
SSID-KD_Rain1400	8.4132	4.1173	4.2191

Fig. 2. *Top row*: (a) Real rainy image, (b) deraining result of supervised image deraining network (SID) only using synthetic data, (c) deraining result of our SSID-KD, corresponding to which dark channel images are presented (The darker is better). *Bottom row*: The NIQE values on real-world rainy datasets SPAData [47], Real275, IVIPC [48], [49] by SID and SSID-KD.

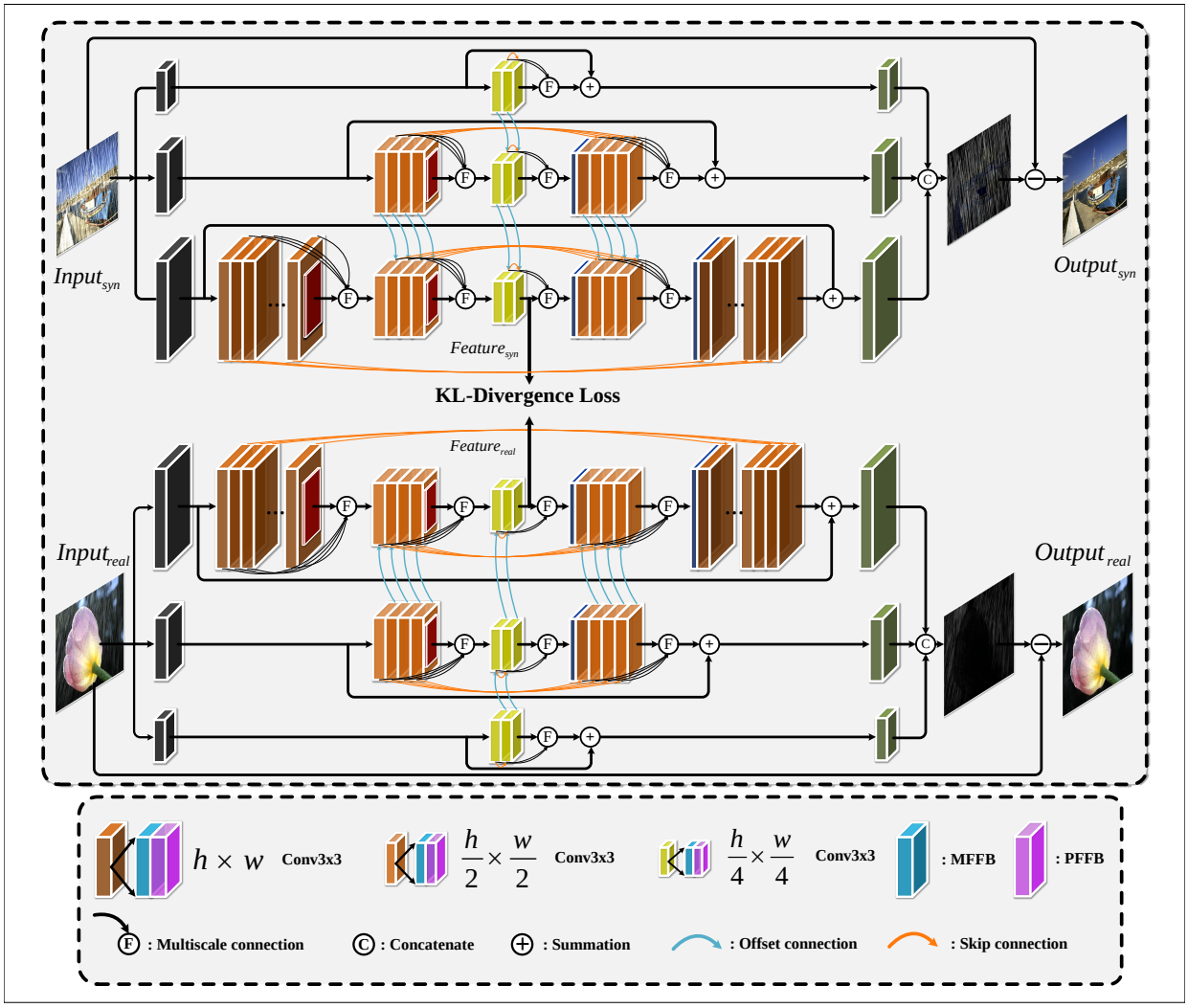
[58], they applied kernel regression to the deraining framework by using a non-local mean filter.

Considering that an image can be viewed as the result of the superposition of various signals, the problem of image deraining can be converted into signal separation. Based on the high and low-frequency decomposition process, Jiang *et al*. [7] suggested dictionary learning that can remove the main component in rainy images using dictionary learning, while sparse-code based model [5], [9] can learn the dictionary of rain streaks and background layers via discriminative sparse coding. In [10], Ren *et al*. regarded image deraining as image deconvolution, and adopted optimized methods to solve rain streak kernels and the corresponding clear background images, while others [6], [8], [11] applied Gaussian mixture models to model rain and background layers. The above prior-based models were established under certain conditions by some given assumptions; In other words, these prior-based methods can achieve better deraining performance on some specific datasets. While on other test datasets, such as large and dense rainy images, the deraining images still existed with blurry edges and rain-streaks residues. In addition, these prior-based methods can generally be regarded as solving convex optimization problems, which is usually very time-consuming.

2) Deep Learning-based Method: Deep learning-based image deraining [4], [15]–[20], [22], [26], [27], [32], [35], [37]–[41], [47], [50], [60]–[65], [65]–[67] has been widely studied with the supervised learning manner, where various network architectures are designed to learn the mapping from rainy images to clean background images. Subsequently, more complicated network architectures are proposed to better extract deep features from rainy images. An Encoder-decoder framework is mainly adopted to generate target image, which is also applied to image processing, such as [20], [68], [69]. Wang *et al*. [18], [20] combined the structure of encoder-decoder with multi-scale learning to better extract streak feature patterns during the processing of downsampling and upsample. Zamir *et al*. [41], Dong *et al*. [70] and Jiang *et al*. [66] made good use of dense connection to learn multi-scale features between different network layers, thereby reducing the risk of delivery drop. Ren *et al*. [19] and [26] presented a better baseline model by investigating the input, output, and loss function focused on network architecture, whose framework was adopted a recurrent structure and can handle heavy rain streak accumulation better than previous works. In [16], multi-scale dilation convolutional network was designed to jointly detect and remove rain streaks. In [50], densely connected CNN was adopted for jointly estimating rain density and removing rain streaks. Besides, there are several works to incorporate lightweight networks in a cascaded scheme [28] or in a Laplacian pyramid framework [4]. Moreover, in [34], the authors proposed to take advantage of adversarial learning to enhance the texture details in derained images. Most recently, the pre-trained transformer [24] is introduced to significantly improve the quantitative metrics for image deraining. To sum up, supervised learning-based deraining methods have achieved excellent performance on paired synthetic datasets, but the trained deraining model are likely to poorly generalize to real-world rainy images.

B. Unpaired Training Strategy for Semi-supervised Learning

However, the deraining model trained with a deep-based supervised methods also has shortcomings because they don't work for most real scenes. Therefore, much semi-supervised rain removal algorithms [1], [2], [42], [43] have emerged in recent years. Semi-supervised learning is suggested to exploit



Layer	enLayer_1	enLayer_2	enLayer_4	deLayer_2	deLayer_1
In_channles	3	24	32	48	32
Out_channels	24	32	48	32	24
Kernel size	3	3	3	3	3
Stride	1	1	1	1	1
Pad	1	1	1	1	1
Unit nums	10	4	2	4	10

Fig. 3. The architecture of semi-supervised deraining with knowledge distillation (SSID-KD). In the first training stage, the teacher deraining model (top row) is trained on paired synthetic training samples and is used to initialize the student deraining model. In the second training stage, the student deraining model (middle row) is finetuned only using real-world rainy images, where KL loss is adopted to enforce the consistency of feature distribution of rain streaks of synthetic and real rainy images. Our SSID-KD can achieve better results than existing semi-supervised and supervised deraining methods. Besides, the configuration of the network backbone is presented in the bottom table, where "enLayer" denotes the encoder layer and "deLayer" denotes the decoder layer. The details of MFFB and PFFB can be found in Figs. 5 and 4, respectively.

real-world rainy images, leading to better generalization in practical applications. In [42], Zhu *et al*. proposed to adopt CycleGAN [43] to exploit unpaired real rainy images, which can improve the generalization ability to real rainy images. Unpair training strategy is also another way to benefit from real rainy images. In [1], SIRR was proposed to transfer the deraining model trained on synthetic images to real rainy images. In [2], Syn2Real was proposed by adopting Gaussian processes to exploit both synthetic and real rainy images. But these semi-supervised deraining methods may also be inferior to supervised methods in terms of quantitative metrics, and there is leeway to improve deraining visual quality.

III. The Proposed Method

In this section, we first present the details of network architecture in Sec. III-A, and then give the proposed semi-supervised deraining framework using knowledge distillation in Sec. III-B.

A. Network Architecture

The overall structure of our SSID-KD and the configuration of main layer is shown in Fig. 3, where the teacher model and

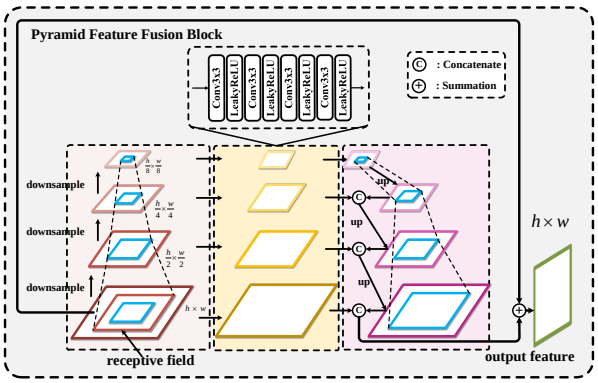


Fig. 4. Pyramid Feature Fusion Block.

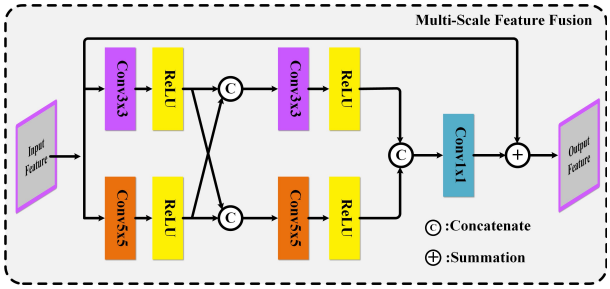


Fig. 5. Multi-scale Feature Fusion Block.

student model share the same architecture, *i.e.*, a three-layer pyramid structure. The input of each layer has been downsampled with rate 2, which is implemented by max-pooling. As for each layer, multi-scale encoder-decoder architecture is adopted for better extracting rain streaks. A variety of feature scales correspond to different numbers of feature extraction modules. The scale of the input rainy images directly affects the deraining ability. Therefore, we reserve 20 unit blocks for full scale to guarantee enough rain streak information, as well as 8 unit blocks for half scale and 2 unit blocks for quarter scale, with dense connections between unit blocks. In addition, features between different layers interact with each other to compensate for the loss of information due to scaling down, which be noted as an offset connection and is denoted as a blue arrow. We also adopt skip connections denoted as an orange arrows in the same layer to prevent gradient vanishes in backpropagation.

We note that different from conventional convolution operation in the original encoder-decoder, the backbone in our SSID-KD is composed of two key modules, i.e., Pyramid Feature Fusion Block (PFFB) and Multi-scale Feature Fusion Block (MFFB), which are detailed in the following.

1) Pyramid Feature Fusion Block: To enhance the robustness of deraining ability, multi-scale learning for feature maps is the key step. Inspired by JDNet [18], DCSFN [20], MSBDN [70] and MSPFN [66], those algorithms all confirm that extracting features at different scales is beneficial to improve the performance. Normally, multi-scale learning strategies are divided into two categories. One is bottom-up, while the other is top-down. In this paper, different from the above models, we design a Pyramid Feature Fusion Block (PFFB), which combines the processing of up-sampling and down-sampling to learn the features from different scales in one unit. PFFB not only converts the convolution into deeper features but also can maintain the original features from shallower ones. The structure of PFFB is shown in Fig. 4, which contains three stages. At the first stage, a given rain streak feature R is downsampled at the rate of 2.

$$\begin{aligned} \mathcal{R}^o &= LeakyReLU(Conv_3^1(\mathcal{R})), \\ \mathcal{R}^r &= ResBlock_3(\mathcal{R}^o), \\ \mathcal{R}^k &= MaxPool_k(\mathcal{R}^r), \end{aligned} \tag{2}$$

where $Conv_i^j(\cdot)$ represents $i \times i$ convolution operation with stride j, $LeakyReLU(\cdot)$ represents an activation function with the parameter of 0.2, $ResBlock(\cdot)$ consists of an activation function between two 3×3 convolution layers. At last, we adopted $MaxPool_k$ to downsample $\mathcal{R}$, where scales $k = 1, 2, 4, 8$. At the second stage, the feature $\mathcal{R}$ at different scales $\mathcal{R}^1, \mathcal{R}^2, \mathcal{R}^4, \mathcal{R}^8$ would be convolved with 4 convolution layers with LeakyReLU activation. This procedure is formulated as

$$\mathcal{R}^i = LeakyReLU(Conv_3^1(\mathcal{R}^{i-1})), \tag{3}$$

where $i = 1, 2, 4, 8$. At the final stage, we upsample each $\mathcal{R}^i$ to the scale of the original input and concatenate these results with the input features following a 1×1 convolution, which can be described as

$$\begin{aligned} \mathcal{R}^{\Delta} &= Concat[\mathcal{R}^1, Up(\mathcal{R}^2), Up(\mathcal{R}^4), Up(\mathcal{R}^8)], \\ \mathcal{R}^{out} &= Conv_1^1\{\mathcal{R}^{\Delta}\}, \end{aligned} \tag{4}$$

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

Different from JDNet [18] model, as for the two adjacent scales, the small-scale feature is upsampled first and its output is concatenated with the latter one. Thus, four different scale features are gradually concatenated until it is restored to the original size, while the Scale-Aggregation block proposed by JDNet uses only one convolution layer, which directly aggregates four scale features, leading to the lack of small-scale feature in their whole network.

2) Multi-scale Feature Fusion Block: Dilated convolution network proposed firstly by Yang *et al.* [16], following which many similar structures, *e.g.*, SRNet [64], MSRB [25], benefit from multi-scale feature extraction. However, these structures would cause feature distortion by large kernel dilation convolution, especially when the size of the dilation convolution kernel has been set up to 5 or larger.

In order to solve the above problems, we propose MFFB as shown in Fig. 5. We construct a two-branch network, where different branches adopt different convolutional kernel rates. One branch with 3×3 convolution is applied to extract fine-scale features, while the other branch with 5×5 convolution is used to extract the coarse-scale features. Therefore, the output from different scale kernels can be fused with each other to

prevent feature map distortion. For a given streak feature input $\mathcal{R}$, MFFB can be defined as

$$\begin{aligned} \mathcal{R}^i &= ReLU(Conv_3(\mathcal{R}^{i-1})), \\ \mathcal{R}^j &= ReLU(Conv_5(\mathcal{R}^{j-1})), \\ \mathcal{R}^{out} &= Conv_1(Concat[\mathcal{R}^i, \mathcal{R}^j]) + \mathcal{R}, \end{aligned} \tag{5}$$

where i denotes the output from 3×3 convolution and j denotes the output from 5×5 convolution, and finally we use the residual strategy to output the features.

B. Semi-supervised Image Deraining using Knowledge Distillation

Currently, deep learning-based single image deraining methods are usually developed within the supervised learning framework, which is limited to the specific synthetic training data. As shown in Fig. 3, we propose to exploit real-world rainy images without corresponding ground truth when training deraining networks. Different from [1], [2], we propose a simple yet effective knowledge distillation strategy to enforce the feature consistency of rain streaks from synthetic and real-world rainy images. Overall, our SSID-KD consists of two steps: (i) Training a teacher deraining model on paired synthetic rainy and ground-truth clean images, and (ii) Training a student deraining model by simultaneously exploiting synthetic and real-world rainy images. Inspired by [62], [63], [71], [72], it is confirmed that image embedding in high-dimension has similarities. Therefore, we convert both streak features from a low-dimensional space to a latent code through the teacher network and student network respectively. The latent code from the teacher network is regarded as a pseudo label when training the student deraining model by minimizing KL divergence between these two distributions. What needs to be emphasized is that the parameters in the teacher model are frozen when training the student model in conventional knowledge distillation for classification. But for the image deraining task, features of synthetic and real rain streaks can benefit each other. And thus, the teacher model in our SSID-KD is also finetuned.

In the first training stage, training the teacher network is a supervised learning procedure. We simply adopt negative SSIM [73] as the training loss of teacher deraining model,

$$\mathcal{L}_{\mathcal{T}} = -SSIM(\tilde{B}_{\mathcal{T}}, B^{gt}), \tag{6}$$

where $\tilde{B}_{\mathcal{T}}$ and B^{gt} are the deraining result and corresponding ground-truth, respectively.

In the second training stage, the student network is first initialized with the parameters of the teacher model. And then the student network is finetuned by enforcing the intermediate teacher features and intermediate student features with some semi-supervised losses that are key to utilizing the feature of real rain streaks, including KL Loss, DC-Loss, and Total Variation Loss (TV-Loss). Finally, the overall loss function for the training student model is formulated as

$$\mathcal{L}_{\mathcal{S}} = \lambda_1 \mathcal{L}_{KL} + \lambda_2 \mathcal{L}_{DC}(\tilde{R}_{\mathcal{S}}) + \lambda_3 \mathcal{L}_{TV}(\tilde{B}_{\mathcal{S}}), \tag{7}$$

where the trade-off parameters are empirically set as $\lambda_1 = \lambda_2 = \lambda_3 = 1 \times 10^{-6}$. Next, we will elaborate on these three losses.

1) KL Loss: Our SSID-KD adopts the Encoder-Decoder framework. The teacher embedding $\mathcal{V}_{syn}$ and the student embedding $\mathcal{V}_{real}$ would be output from the Encoder layer. Thus, $\mathcal{V}_{syn}$ is the pseudo-label of $\mathcal{V}_{real}$, whose rain streak features are converted and aggregated in a higher dimensional vector to calculate the KL loss. The process can be described as,

$$\begin{aligned} \mathcal{P}_{real} &= Softmax(\mathcal{V}_{real}), \\ \mathcal{P}_{syn} &= Softmax(\mathcal{V}_{syn}), \\ \mathcal{L}_{KL} &= KL(\mathcal{P}_{real}, \mathcal{P}_{syn}), \end{aligned} \tag{8}$$

by which the consistency of real and synthetic rain streaks can be enforced. Actually, we adopt the API provided by Pytorch [74] directly in the practical implementation.

2) Dark Channel Loss: Moreover, considering the possible haze phenomenon in heavy rainy images, we propose to further add the dark channel [44], [45] constraint on the deraining images of the student model. For a given image $\mathcal{I}$, the value of the dark channel is defined by

$$\mathcal{D}(\mathcal{I}) = \min_{y \in \mathcal{N}(x)} [\min_{c \in r,g,b} \mathcal{I}^c(y)], \tag{9}$$

where x and y are pixel coordinates, $\mathcal{I}^c$ denotes c-th color channel, and $\mathcal{N}(x)$ is an image patch centered at x. As for rain streaks, the maximum a pixel is one. So, we apply max-pooling for acquiring the minimum of patches and calculate the minimum distance from the zero vector. For given synthetic streak feature $\mathcal{R}^s$ and real-world streak feature $\mathcal{R}^r$, the processing procedure can be described as

$$\begin{aligned} \mathcal{N}^s_{max} &= Maxpool3D(\mathcal{R}^s), \\ \mathcal{N}^r_{max} &= Maxpool3D(\mathcal{R}^r), \\ \mathcal{N}^s_{dc} &= 1 - \mathcal{N}^s_{max}, \\ \mathcal{N}^r_{dc} &= 1 - \mathcal{N}^r_{max}, \\ \mathcal{L}_{DC} &= ||\mathcal{N}^s_{dc} - 0||_1 + ||\mathcal{N}^r_{dc} - 0||_1, \end{aligned} \tag{10}$$

where $\mathcal{N}^s_{max}$ and $\mathcal{N}^r_{max}$ is max value of pixel in $\mathcal{R}^s$ and $\mathcal{R}^r$, respectively. The dark channel pixel can be obtained by subtracting from 1. Finally, we compute the distance between the dark channel value and 0. Fig. 2 has confirmed our conclusion, where the deraining image from SSID-KD is better than SID notably.

3) Total Variation Loss: To achieve better deraining quality, we adopt the total variation loss [75] to enforce the second training stage to generate images that have the same statistical properties as clean images. The specific implementation of the TV loss formulation is as follows

$$\mathcal{L}_{TV} = ||\nabla_h \mathcal{I}||_1 + ||\nabla_v \mathcal{I}||_1, \tag{11}$$

where ∇_h and ∇_v represent the horizontal and vertical differential operators respectively.

C. Discussion

In our work, we aim to bridge the notable performance gaps between existing semi-supervised deraining methods [1], [2] and supervised deraining methods. Image deraining, as a regression problem, is significantly different from the classification task. In [76], cross-entropy loss with distilling temperature

TABLE I
COMPARISON OF DIFFERENT BENCHMARK DATASETS. SIRR, SYN2REAL AND OUR SSID-KD ARE SEMI-SUPERVISED DERAINING METHODS, WHILE THE OTHER METHODS ARE IN SUPERVISED LEARNING FAMILY. RED, BLUE AND CYAN COLORS ARE USED TO INDICATE TOP 1ST, 2ND AND 3RD RANK, RESPECTIVELY.

Methods	Rain200H		Rain1200		Rain1400		Rain12		SPA-Data	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
NLEDN [61] (ACM MM'18)	27.3149	0.8904	30.7989	0.9127	31.0138	0.9206	33.0278	0.9615	30.5964	0.9363
RESCAN [26] (ECCV'18)	26.6189	0.8398	32.0793	0.9038	30.8814	0.9114	32.6857	0.9508	30.5855	0.9380
ReHEN [60] (ACM MM'19)	27.5254	0.8663	30.4558	0.8702	30.9840	0.9156	35.0950	0.9400	32.6524	0.9297
PReNet [19] (CVPR'19)	27.6400	0.8872	27.3073	0.8712	30.6087	0.9181	34.7912	0.9644	32.7195	0.9317
DCSFN [20] (ACM MM'20)	28.5874	0.9037	32.5967	0.9271	30.1857	0.9001	35.2163	0.9671	32.7628	0.9365
MSPFN [66] (CVPR'20)	25.5536	0.8039	30.3903	0.8202	24.2786	0.9170	34.2532	0.9469	29.5378	0.9193
RCDNet [38] (CVPR'20)	28.6983	0.8928	32.2731	0.9111	31.0163	0.9164	31.0385	0.9069	30.1744	0.9201
VRGNet [40] (CVPR'21)	27.4943	0.8619	32.1339	0.9065	30.5778	0.9089	34.5975	0.9605	33.4424	0.9409
SIRR [1] (CVPR'19)	14.4201	0.4501	24.4270	0.7713	25.7725	0.8224	24.1383	0.7768	22.6659	0.7474
Syn2Real [2] (CVPR'20)	14.4950	0.4021	28.8120	0.8400	28.5821	0.8586	28.4346	0.9038	31.8243	0.9307
SID	28.9250	0.9079	32.2880	0.9293	32.2576	0.9203	36.2915	0.9609	33.0260	0.9364
SSID-KD	28.7066	0.9005	32.4237	0.9202	30.5398	0.9136	35.4726	0.9682	31.9741	0.9371

is adopted on classification outputs, where the student model is distilled by the classification logits of the teacher model. In our SSID-KD, KL-divergence loss is adopted to distill latent code of student network, by which the deep features of rain streaks in high dimension between synthetic and real rainy images can be exploited to benefit generalization ability of student network when handling real-world rainy images. In terms of knowledge distillation, our contribution is how to utilize KL-divergence for distilling deep features of rain streaks, instead of designing a new knowledge distillation formulation for a regression problems. Although the basic distilling formulation in SSID-KD based on KL-divergence is similar to cross-entropy in [76], this paper also has contributions to the overall semi-supervised learning strategy and network backbone. As for semi-supervised training strategy, to further relieve the domain gap between synthetic and real rain streaks, two regularization terms, *i.e.*, dark channel loss and total variation loss, are introduced to cooperate with KL-divergence loss. As for the network backbone, two modules MFFB and PFFB are proposed, which not only improve the performance of the semi-supervised deraining model (SSID-KD) but also can act as basic modules in a supervised deraining model (SID). Our SID is superior to other state-of-the-art supervised deraining models. Our SSID-KD is among the first semi-supervised deraining method that obtains satisfactory deraining performance while existing semi-supervised methods Syn2Real [2] and SIRR [1] are much inferior.

IV. EXPERIMENTAL RESULTS

In this section, we conduct extensive experiments to demonstrate the effectiveness of the proposed method on four synthetic datasets and three real-world datasets. Besides the direct quantitative and qualitative evaluation of deraining images, we also evaluate competing deraining methods in terms of the performance of object detection and semantic segmentation on rainy images. In the ablation study, our SSID-KD is analyzed.

TABLE II
THE VALUES OF NIQE ON REAL DATASETS. RED, BLUE AND CYAN COLORS ARE USED TO INDICATE TOP 1ST, 2ND AND 3RD RANK, RESPECTIVELY.

Methods	SPA-Data NIQE	Real275 NIQE	IVIPC NIQE
NLEDN [61] (ACM MM'18)	7.1806	3.5324	3.7363
RESCAN [26] (ECCV'18)	7.0772	3.8965	3.7887
ReHEN [60] (ACM MM'19)	7.1281	3.7355	3.7634
PReNet [19] (CVPR'19)	7.1949	3.7213	3.7714
DCSFN [20] (ACM MM'20)	7.2649	3.6375	3.7695
MSPFN [66] (CVPR'20)	7.9280	3.8616	3.9771
RCDNet [38] (CVPR'20)	8.4286	3.7597	3.7627
VRGNet [40] (CVPR'21)	7.1281	3.7903	3.7680
SIRR [1] (CVPR'19)	7.5571	3.5492	3.7419
Syn2Real [2] (CVPR'20)	7.1190	4.0372	4.2141
SID	7.0006	3.5290	3.7406
SSID-KD	6.9636	3.5105	3.6488

A. Datasets

1) Synthetic Datasets: We evaluate our method on synthetic datasets including Rain200H [16], Rain1200 [50], Rain1400 [15] and Rain12 [8]. Rain200H [16] dataset has heavy rain with different shapes, directions, and sizes, which is the most challenging dataset, including 1800 images for training and 200 images for testing. Rain1200 [50] dataset has three different levels of rain images, including heavy rain, medium rain, and light rain, as well as containing 12000 training images and 1200 testing images. Rain1400 [15] dataset has medium-level rainy images, which includes 12600 images for training and 1400 images for testing. Rain12 [8] dataset only contains 12 test rain images, whose deraining images are predicted by SID and SSID-KD trained on Rain200H.

2) Real-world Datasets: As is well known, the ground-truths corresponding to real-world rainy images is difficult to obtain. Thanks to SPANet [47], it provides 1000 real-world rainy images with ground-truth clean images for testing. Besides, in order to better prove the robustness of our model's performance, we also select two real-world test datasets, *i.e.*, IVIPC [48], [49] with 128 images and Real275 with 275

Fig. 6. Comparisons on synthetic rainy images from the Rain200H dataset.

images that were captured in real scenes from Internet, on which deraining results are measured using NIQE.

B. Implementation Details

Our SSID-KD is trained on the synthetic datasets Rain200H, Rain1200, and Rain1400 as well as on real-world images from Real275. Our teacher and student networks are implemented using the Pytorch [74] framework with the ADAM [77] as the optimizer. We train the networks on a PC equipped with two NVIDIA GTX 2080Ti GPUs. In our experiments, the training of SSID is divided into two stages: the teacher network and the student network. We train the teacher network with 520 epochs, and the learning rate is set as 0.005, which is divided by 10 at 312 epochs and 416 epochs. The patch size is 120×120, and the batch size is 12. We train the student network with 10 epochs, and the learning rate is set as 0.0001. The training patch size is 128×128 to make the student network learn more image detail and texture information, and the batch size is 2.

C. Comparisons with State-of-the-Arts

1) Quantitative Comparison: We compare our proposed model SSID-KD with supervised deraining models, including NLEDN [61], RESCAN [26], ReHEN [60], PReNet [19], DCSFN [20], MSPFN [66], RCDNet [38] as well as VRGNet [40], and semi-supervised deraining models, including SIRR [1] and Syn2Real [2]. We adopt PSNR, SSIM [73] and NIQE [51] to evaluate the quality of each restored image. Table I and Table II show the quantitative evaluation results. Our model SSID-KD can generate the results with the highest PSNR and SSIM values compared to the semi-supervised deraining methods, while is comparable with supervised deraining methods.

2) Qualitative Comparison: Fig. 6 exhibits some synthetic examples from Rain200H dataset. Syn2Real [2] does not remove the rain streaks. Although other evaluated models generate better results than Syn2Real [2], there still exist some artifacts or rain streak residuals in the restored images. In contrast, our SSID-KD generates better deraining results, which are visually close to the ground-truth images. In addition, we also provide some examples from the real-world datasets in Fig. 7 to demonstrate the effectiveness of the proposed algorithm. We note that the other methods do not effectively remove the rain streaks, while our models generate better results.

3) Computational Efficiency Comparison: We compare the inference time of our method and the other state-of-the-art deraining methods, including supervised deraining methods, *i.e.*, NLEDN [61], RESCAN [26], ReHEN [60], PReNet [19], DCSFN [20], MSPFN [66], RCDNet [38] and VRGNet [40], and two semi-supervised deraining methods, *i.e.*, Syn2Real [2] and SIRR [1]. In particular, we adopt these methods to

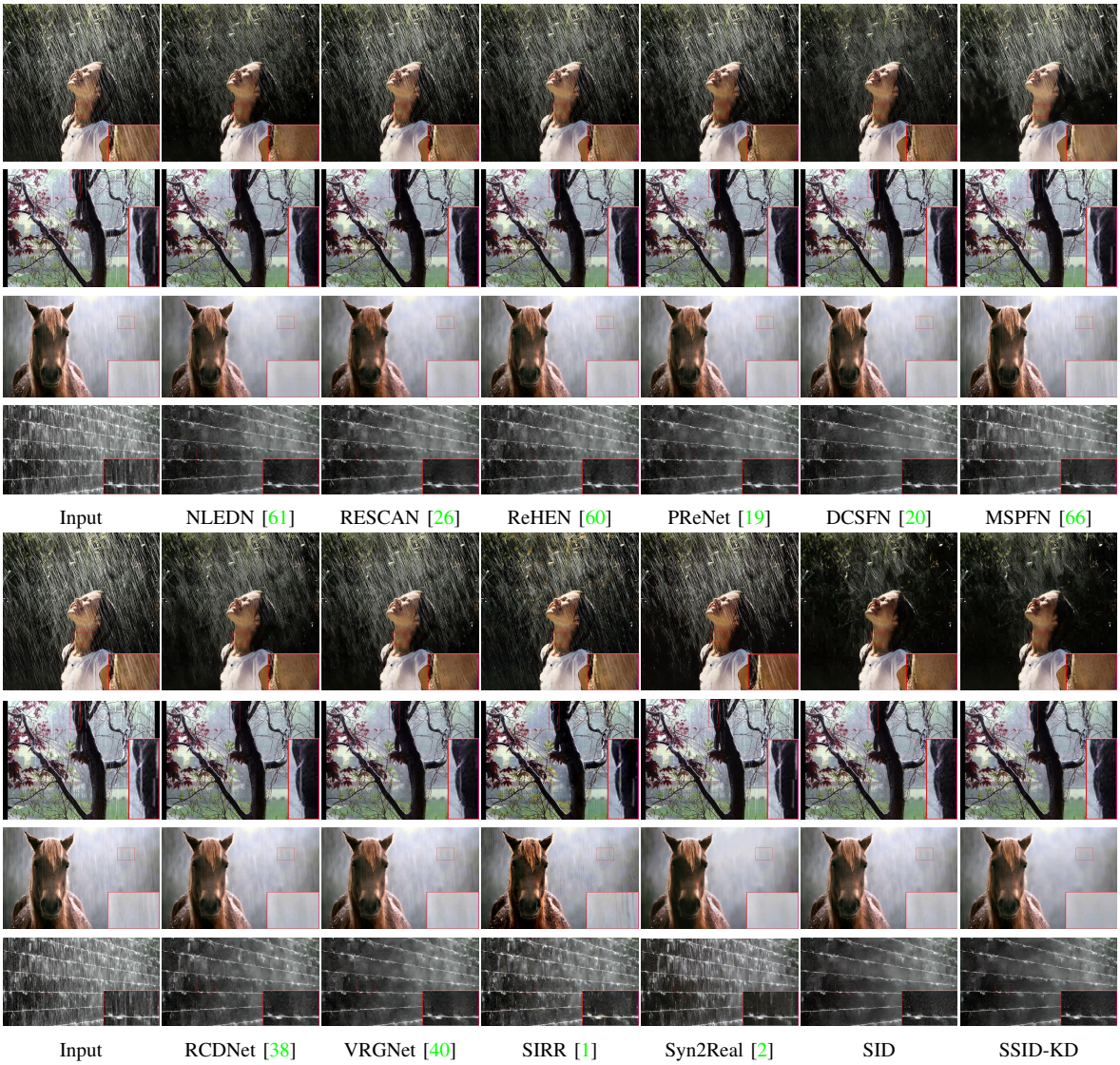


Fig. 7. Comparison of real rainy images from Real275 datasets and IVIPC datasets. The first two rows are from Real275, and the last two rows are from IVIPC.

TABLE III
RUNNING TIME (*seconds*) COMPARISON OF COMPETING FOR DERAINING METHODS. SIRR, SYN2REAL, AND OUR SSID-KD ARE SEMI-SUPERVISED DERAINING METHODS, WHILE THE OTHER METHODS ARE IN A SUPERVISED LEARNING FAMILY.

Image size	NLEDN [61]	RESCAN [26]	ReHEN [60]	PReNet [19]	SIRR [1]	SID
512×512	0.094	0.026	0.016	0.104	0.442	0.299
1024×1024	0.259	0.070	0.064	0.400	0.855	0.572
Image size	DCSFN [20]	MSPFN [66]	RCDNet [38]	VRGNet [40]	Syn2Real [2]	SSID-KD
512×512	0.362	0.831	0.817	0.102	0.028	0.301
1024×1024	0.971	2.310	1.748	0.394	0.032	0.575

process rainy images with sizes 512×512 and 1024×1024, and the running time is recorded on an NVIDIA Geforce 2080Ti GPU. As shown in Table III, our SID ranks at the median of supervised deraining methods in terms of running time. As for the inference efficiency of SSID-KD, its computational cost is the same as SID, since they share the same network backbone. In comparison to semi-supervised deraining methods, SSID-KD is faster than SIRR and slower than Syn2Real. Syn2Real is a lightweight model with low computational cost, but its deraining performance is much inferior to other methods as in Table I and Table II. In summary, our SID and SSID-KD can provide a better trade-off between deraining performance and computational efficiency. Moreover, Table IV shows the number of MACs and parameters of competing methods.

TABLE IV
THE COMPARISON OF COMPETING FOR DERAINING METHODS IN THE THEORETICAL AMOUNT OF MULTIPLY-ADD OPERATIONS IN CONVOLUTIONAL NEURAL NETWORKS (MACS) AND THE NUMBER OF PARAMETERS IN THE DERAINING MODEL (PARAMS). SIRR, SYN2REAL, AND OUR SSID-KD ARE SEMI-SUPERVISED DERAINING METHODS, WHILE THE OTHER METHODS ARE IN A SUPERVISED LEARNING FAMILY.

Image size	NLEDN [61]	RESCAN [26]	ReHEN [60]	PReNet [19]	SIRR [1]	SID
MACs(G)	22.18	32.32	22.49	66.58	32.37	64.66
Params(M)	1.01	0.15	1.07	0.17	2.58	4.43
Image size	DCSFN [20]	MSPFN [66]	RCDNet [38]	VRGNet [40]	Syn2Real [2]	SSID-KD
MACs(G)	34.89	59.71	79.71	91.51	21.10	64.66
Params(M)	2.11	3.47	3.17	0.17	2.62	4.43

Although our SID and SSID-KD have the highest network parameters, their MAC operations are less than PReNet [19], RCDNet [38] and VRGNet [40]. As for the other methods, *e.g.*, PReNet [19] and RESCAN [26], a recursive strategy is adopted to reuse parameters, resulting in fewer parameters but more MACs. The parameters and MACs of our SID and SSID-KD can also be further reduced by adopting recursive strategy and model compression when facing practical applications.

4) Evaluation of Object Detection and Semantic Segmentation: By taking object detection and semantic segmentation as examples, we further evaluate the competing training derainig methods, when applying them to process rainy images in high-level computation vision tasks. In MSPFN [66], Jiang *et al*. provides two new rainy datasets for evaluating deraining performance on object detection and semantic segmentation. For object detection, 350 images are randomly selected from the MS COCO dataset [78], while for semantic segmentation, 150 images are randomly selected from BDD dataset [79]. Then, diverse rain streaks are randomly added to each image, resulting in rainy COCO* and BDD* datasets, based on which these deraining algorithms can be evaluated in terms of detection and segmentation performance on derained images. For a rainy image, different deraining methods are first adopted to produce derained images, based on which pre-trained YOLO v3 [80] model RefineNet [81] model are applied for object detection and semantic segmentation, respectively. Besides PSNR and SSIM, these training methods can be evaluated in terms of performance of detection and segmentation, as reported in Table V and Table VI.

From Table V and Table VI, one can see that all the deraining methods can improve the performance of detection and segmentation than directly taking rainy images as input. As for taking ground-truth clean images as input, the performance of detection and segmentation is much higher than any deraining method, leaving significant room for developing better deraining methods. Nevertheless, our SID can lead to the best quantitative results on both detection and segmentation tasks. As for our SSID-KD, its performance is a litter inferior to SID and some supervised deraining methods. Actually, it is reasonable, because the rain streaks in COCO* and BDD* are synthetic, thus having a domain gap with real-world rain streaks. Benefiting from semi-supervised learning with knowledge distillation, our SSID-KD is more beneficial to real-world images. As shown in Fig. 8 and Fig. 9, derained images by our SSID-KD are cleaner, and the effects of detection and segmentation are much better than the other methods. Moreover, in comparison to two semi-supervised deraining methods, *i.e.*, SIRR and Syn2Real, our SSID-KD significantly outperforms them quantitatively (Table V and Table VI) and qualitatively (Fig. 8 and Fig. 9), indicating the superiority of our semi-supervised learning strategy.

D. Ablation Study

In this section, we provide further analysis of the proposed model. All the experimental results are conducted on Rain200H [16] and Real275 datasets.

1) Analysis of Connection Form of Network: The whole network architecture includes many multi-scale connections and skips connections. Therefore, in terms of different connection forms, we will conduct experiments on the no-skip connection (no-SC), skip connection (SC), multi-scale connection with addition (MC-Add), and multi-scale connection with concatenation (MC-Concat). To demonstrate the effect of each component on image deraining, we disable the component in the proposed model and train it using the same settings as the proposed model for fair comparisons. The results are illustrated in Table VII. We can observe that both skip connection and multi-scale connection have improvements for the deraining results. Inspired by MSBDN [70], we adopt the strategy of multi-scale connection by addition. The experimental results proved that its effect is not as good as the multi-scale connection by concatenation that we proposed. As for MC-Add, the output features with different scales need to be restored to the original size by interpolation, and the addition operation usually breaks the pixel value threshold, and it doesn't work as well as MC-Concate.

2) Analysis of the Number of Layers: Because the number of network layers is also a key factor, we further evaluate its effect on image deraining. We note that both the teacher network and the student network adopt three layers structure, which is shown in Fig. 3. The first layer and second layers as an auxiliary layer can compensate for the loss of information caused by down-sampling in the main layer (the third layer). In order to verify the role of the auxiliary layers, we conduct three experiments that carry out no small-scale layer for offsetting ($N = 1$) adopt a 2 times down-sampling layer for compensation ($N = 2$), and both use a 4 times down-sampling layer and 2 times down-sampling layer for compensation ($N = 3$). We can see the best results are obtained when

TABLE V
COMPARISON OF OBJECT DETECTION PERFORMANCE ON RAINY IMAGES FROM COCO* DATASET. FOR RAINY IMAGES, A DERAINING METHOD IS FIRST ADOPTED TO PRODUCE DERAINED IMAGES, ON WHICH A PRE-TRAINED YOLO V3 MODEL IS THEN USED FOR DETECTION. SIRR, SYN2REAL, AND OUR SSID-KD ARE SEMI-SUPERVISED DERAINING METHODS, WHILE THE OTHER METHODS ARE IN A SUPERVISED LEARNING FAMILY.

	Rainy images	NLEDN [61]	RESCAN [26]	ReHEN [60]	PReNet [19]	SIRR [1]	SID
Precision(%)	23.01	30.19	28.83	30.85	31.28	23.85	33.16
Recall(%)	28.84	36.94	35.57	37.41	38.09	30.01	40.87
IoU(%)	54.92	58.39	60.01	59.94	59.47	55.70	62.53
	Clean images	DCSFN [20]	MSPFN [66]	RCDNet [38]	VRGNet [40]	Syn2Real [2]	SSID-KD
Precision(%)	46.57	33.42	32.49	32.98	32.97	24.07	32.17
Recall(%)	53.61	39.88	39.28	40.47	40.55	29.68	39.31
IoU(%)	75.13	61.85	61.73	62.17	62.39	55.93	61.69

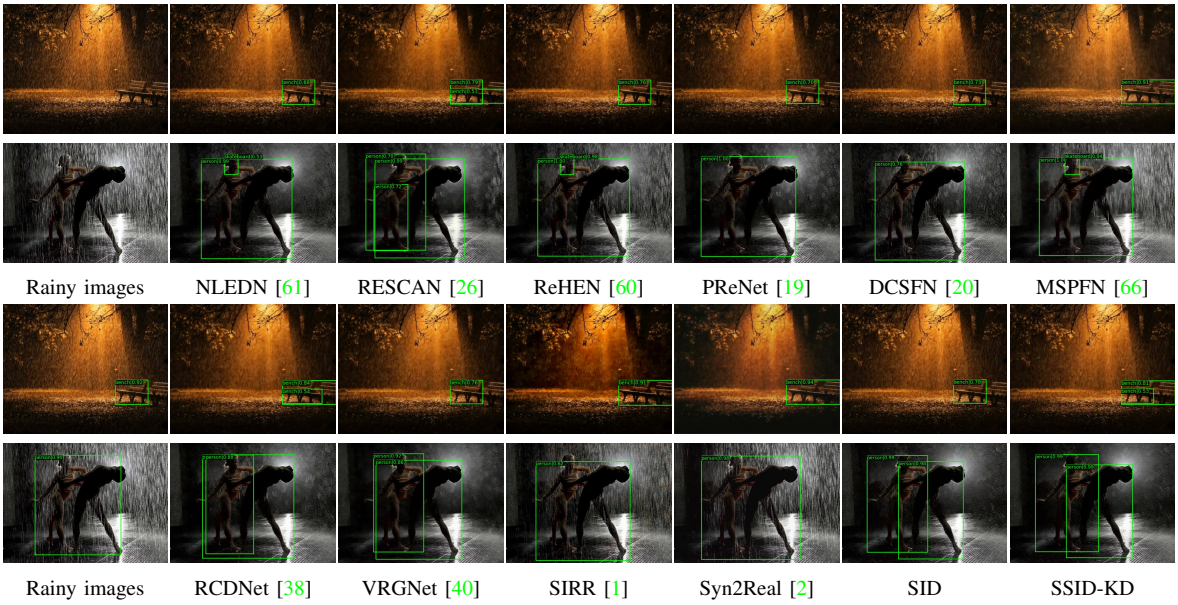


Fig. 8. Comparison of objection detection task on real-world rainy images.

TABLE VI
COMPARISON OF SEMANTIC SEGMENTATION PERFORMANCE ON RAINY IMAGES FROM BDD* DATASET. FOR RAINY IMAGES, A DERAINING METHOD IS FIRST ADOPTED TO PRODUCE DERAINED IMAGES, ON WHICH A PRE-TRAINED REFINENET MODEL IS ADOPTED FOR SEGMENTATION. SIRR, SYN2REAL, AND OUR SSID-KD ARE SEMI-SUPERVISED DERAINING METHODS, WHILE THE OTHER METHODS ARE IN A SUPERVISED LEARNING FAMILY.

	Rainy images	NLEDN [61]	RESCAN [26]	ReHEN [60]	PReNet [19]	SIRR [1]	SID
mPA(%)	32.98	44.73	45.49	45.07	50.13	33.78	53.61
mIoU(%)	20.77	30.52	31.97	30.92	32.99	20.99	36.88
	Clean images	DCSFN [20]	MSPFN [66]	RCDNet [38]	VRGNet [40]	Syn2Real [2]	SSID-KD
mPA(%)	71.60	53.04	51.81	52.96	53.32	33.97	52.79
mIoU(%)	64.37	36.41	36.04	35.90	36.84	21.09	36.03

$N = 3$, which also demonstrates our proposed small-scale compensation strategy is beneficial.

3) Analysis of Unit Blocks: In the whole network structure, we adopt the PFFB block combined with the MFFB block as the convolution unit. For verifying the effectiveness of a single block, we conduct two experiments on the unit block, as shown in Table IX. The PFFB can help our model learn multi-scale features, and thus achieves higher PSNR and SSIM when processing separately, while the MFFB adopts convolution kernels with different scales to enhance the fitting ability of the model, benefiting the processing quality of real-world images. Our experimental results validate that PFFB and MFFB significantly contribute to deraining performance improvement.

4) Analysis on Loss Functions: The proposed semi-supervised method consists of total variation (TV), dark channel (DC), and KL losses. The visual example and NIQE values are presented in Fig. 10. We can see that the DC loss is able to improve the deraining results, while the combination of TV & DC & KL loss achieves the best performance on real-world

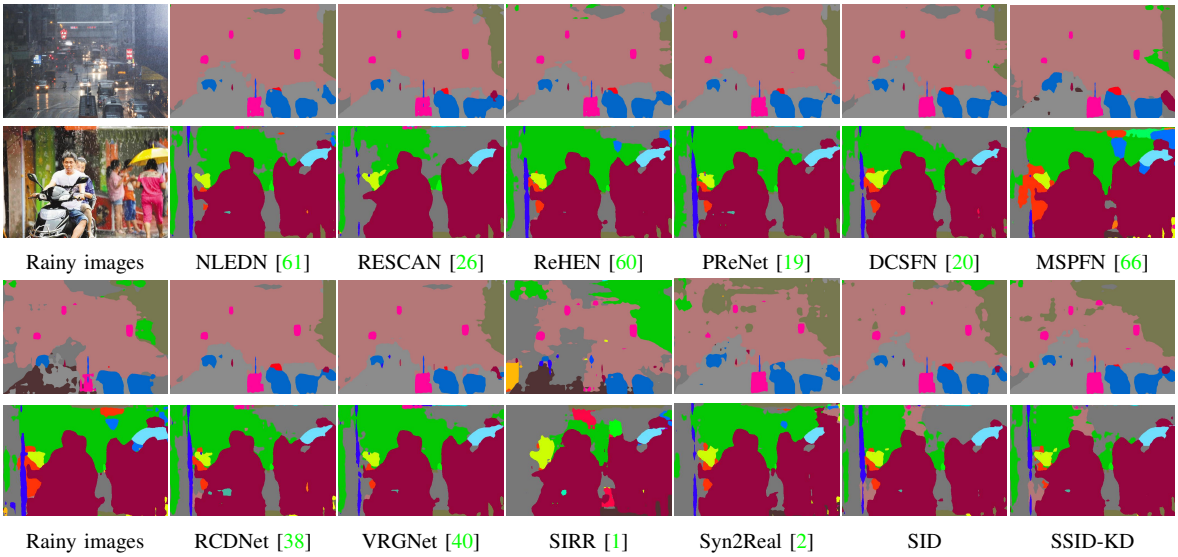


Fig. 9. Comparison of semantic segmentation task on real-world rainy images.

TABLE VII
COMPARISON ON THE CONNECTION FORM OF NETWORK.

	No-SC			SC		
Datasets	Rain200H		Real275	Rain200H		Real275
Metrics	PSNR	SSIM	NIQE	PSNR	SSIM	NIQE
SID	26.305	0.8560	3.8247	27.568	0.8861	3.7110
SSID-KD	25.963	0.8543	3.6914	27.192	0.8526	3.6727
	MC-Add			MC-Concat		
Datasets	Rain200H		Real275	Rain200H		Real275
Metrics	PSNR	SSIM	NIQE	PSNR	SSIM	NIQE
SID	28.135	0.8964	3.6913	**28.925**	**0.9079**	3.5290
SSID-KD	27.618	0.8715	3.6647	28.707	0.901	**3.5105**

TABLE VIII
COMPARISON OF THE NUMBER OF LAYERS.

	$N=1$		$N=2$		$N=3$	
Datasets	Rain200H		Rain200H		Rain200H	
Metrics	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SID	28.549	0.9048	28.674	0.9061	**28.925**	**0.9079**
SSID-KD	28.437	0.9025	28.495	0.9045	28.707	0.9005
Datasets	Real275		Real275		Real275	
Metrics	NIQE		NIQE		NIQE	
SID	3.9475		3.6930		3.5290	
SSID-KD	3.7485		3.6504		**3.5105**	

datasets. We can observe that it is able to generate a clearer deraining result on the real-world images, while partial TV loss and TV & DC loss are ineffective in clearly removing some rain streaks in the cropped region. DC loss and KL loss can make good use of the high-dimensional embedding $\mathcal{V}_{syn}$ and $\mathcal{V}_{real}$ to interact with each other to correct the randomness of the synthetic rain streaks distribution, while TV loss can ensure that the texture of deraining images would not be distorted in the horizontal and vertical gradients of the pixels.

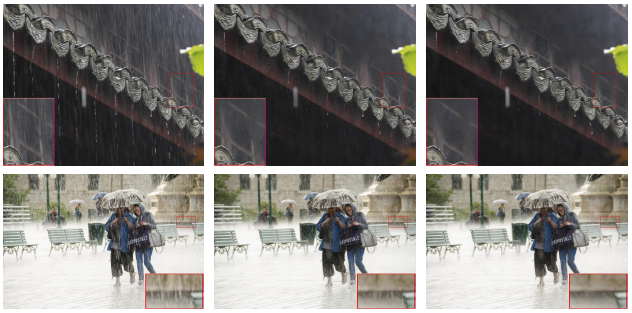
Metric	TV	TV&DC	TV&DC&KL (Ours)
NIQE	3.9790	3.6765	**3.5105**

Fig. 10. Effect of loss function on real-world dataset Real275.

TABLE IX
COMPARISON OF THE NUMBER OF LAYERS.

	MFFB		PFFB		MFFB+PFFB	
Datasets	Rain200H		Rain200H		Rain200H	
Metrics	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SID	27.571	0.8934	28.041	0.8964	**28.925**	**0.9079**
SSID-KD	27.490	0.8925	28.017	0.8925	28.707	0.9005
Datasets	Real275		Real275		Real275	
Metrics	NIQE		NIQE		NIQE	
SID	3.6019		3.7578		3.5290	
SSID-KD	3.5384		3.6397		**3.5105**	

5) Analysis of Knowledge Distillation: We analyze the effect of knowledge distillation, and the results are illustrated in Fig. 11. As can be seen that the proposed knowledge



(a) (b) (c)

Fig. 11. Analysis of knowledge distillation on real-world dataset Real275. (a) Input. (b) Results of teacher network. (c) Results of student network.

(a) Input (b) Ours

Fig. 12. A failure example is where rain streaks are too long and possibly do not appear in real-world training images.

distillation (i.e., training with student network) is able to obtain clearer deraining results, while the teacher network which is only trained on the synthetic dataset cannot completely remove rain streaks. This demonstrates that our proposed knowledge distillation manner contributes to improving deraining performance.

6) Limitation: Our method does not obtain good results when dealing with some cases with too heavy and long rain streaks. As shown in Fig. 12, the rainy image contains a lot of slender and large streaks, which do not appear in the three real-world training datasets.

V. Conclusion

In this work, we proposed a semi-supervised approach based on knowledge distillation to tackle image deraining, *i.e.*, SSID-KD. The teacher deraining model is first trained on synthetic training pairs, and then KL loss is adopted to enforce the consistency of feature distribution of rain streaks from the teacher and student deraining models. The student deraining model exploits real-world rainy images with TV loss, DC loss, and KL loss, significantly improving its generalization ability. Moreover, we propose a novel network backbone for SSID-KD, where two modules PFFB and MFFB are designed to better extract features from rainy images. Extensive experiments on synthetic and real-world benchmark datasets have validated the effectiveness of our SSID-KD, which quantitatively and qualitatively outperforms existing semi-supervised deraining methods and is comparable with state-of-the-art supervised deraining methods. In future work, the proposed semi-supervised framework is the potential to be extended to other relevant low-level vision tasks, *e.g.*, blind image denoising.

References

[1] W. Wei, D. Meng, Q. Zhao, Z. Xu, and Y. Wu, "Semi-supervised transfer learning for image rain removal," in *IEEE CVPR*, 2019.
[2] R. Yasarla, V. A. Sindagi, and V. M. Patel, "Syn2real transfer learning for image deraining using gaussian processes," in *IEEE CVPR*, 2020.
[3] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE TIP*, 2012.
[4] X. Fu, B. Liang, Y. Huang, X. Ding, and J. Paisley, "Lightweight pyramid networks for image deraining," *IEEE TNNLS*, 2020.
[5] Y. Luo, Y. Xu, and H. Ji, "Removing rain from a single image via discriminative sparse coding," in *IEEE ICCV*, 2015.
[6] J.-H. Kim, J.-Y. Sim, and C.-S. Kim, "Video deraining and desnowing using temporal correlation and low-rank matrix completion," *IEEE TIP*, 2015.
[7] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and Y. Wang, "Fastderain: A novel video rain streak removal method using directional gradient priors," *IEEE TIP*, 2018.
[8] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain streak removal using layer priors," in *IEEE CVPR*, 2016.
[9] Y.-L. Chen and C.-T. Hsu, "A generalized low-rank appearance model for spatio-temporally correlated rain streaks," in *IEEE ICCV*, 2013.
[10] D. Ren, W. Zuo, D. Zhang, L. Zhang, and M.-H. Yang, "Simultaneous fidelity and regularization learning for image restoration," *IEEE TPAMI*, 2021.
[11] J. Xu, L. Zhang, and D. Zhang, "External prior guided internal prior learning for real-world noisy image denoising," *IEEE TIP*, 2018.
[12] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE TIP*, 2012.
[13] S. Gu, D. Meng, W. Zuo, and L. Zhang, "Joint convolutional analysis and synthesis sparse representation for single image layer separation," in *IEEE CVPR*, 2017.
[14] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and Y. Wang, "A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors," in *IEEE CVPR*, 2017.
[15] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *IEEE CVPR*, 2017.
[16] 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 *IEEE CVPR*, 2017.
[17] D. Ren, W. Shang, P. Zhu, Q. Hu, D. Meng, and W. Zuo, "Single image deraining using bilateral recurrent network," *IEEE TIP*, 2020.
[18] C. Wang, Y. Wu, Z. Su, and J. Chen, "Joint self-attention and scale-aggregation for self-calibrated deraining network," in *ACM MM*, 2020.
[19] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, "Progressive image deraining networks: A better and simpler baseline," in *IEEE CVPR*, 2019.
[20] C. Wang, X. Xing, Y. Wu, Z. Su, and J. Chen, "Dcsfn: Deep cross-scale fusion network for single image rain removal," in *ACM MM*, 2020.
[21] K. Jiang, Z. Wang, P. Yi, C. Chen, Z. Han, T. Lu, B. Huang, and J. Jiang, "Decomposition makes better rain removal: An improved attention-guided deraining network," *IEEE TCSVT*, 2021.
[22] Y. Ding, M. Li, T. Yan, F. Zhang, Y. Liu, and R. W. Lau, "Rain streak removal from light field images," *IEEE TCSVT*, 2021.
[23] W. Xu, X. Chen, H. Guo, X. Huang, and W. Liu, "Unsupervised image restoration with quality-task-perception loss," *IEEE TCSVT*, 2022.
[24] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-trained image processing transformer," in *IEEE CVPR*, 2021.
[25] H. Zhu, C. Wang, Y. Zhang, Z. Su, and G. Zhao, "Physical model guided deep image deraining," in *IEEE ICME*, 2020.
[26] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha, "Recurrent squeeze-and-excitation context aggregation net for single image deraining," in *ECCV*, 2018.
[27] L. Zhu, Z. Deng, X. Hu, H. Xie, X. Xu, J. Qin, and P.-A. Heng, "Learning gated non-local residual for single-image rain streak removal," *IEEE TCSVT*, 2021.
[28] Z. Fan, H. Wu, X. Fu, Y. Hunag, and X. Ding, "Residual-guide feature fusion network for single image deraining," in *ACM MM*, 2018.
[29] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, "Scale-recurrent network for deep image deblurring," in *IEEE CVPR*, 2018.
[30] C. Tian, Y. Xu, and W. Zuo, "Image denoising using deep cnn with batch renormalization," *NNs*, 2020.
[31] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang, "Non-local recurrent network for image restoration," in *NeurIPS*, 2018.
[32] X. Cui, W. Shang, D. Ren, P. Zhu, and Y. Gao, "Semi-supervised single image deraining with discrete wavelet transform," in *PRICAI*, 2021.

[33] W. Dong, P. Wang, W. Yin, G. Shi, F. Wu, and X. Lu, "Denoising prior driven deep neural network for image restoration," *IEEE TPAMI*, 2018.

[34] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *IEEE TCSVT*, 2019.

[35] L. Cai, S.-Y. Li, D. Ren, and P. Wang, "Dual recursive network for fast image deraining," in *IEEE ICIP*, 2019.

[36] R. Yasarla and V. M. Patel, "Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image de-raining," in *IEEE CVPR*, 2019.

[37] W. Shang, P. Zhu, D. Ren, and H. Shi, "Bilateral recurrent network for single image deraining," in *IEEE ICASSP*, 2020.

[38] H. Wang, Q. Xie, Q. Zhao, and D. Meng, "A model-driven deep neural network for single image rain removal," in *IEEE CVPR*, 2020.

[39] B. Pang, D. Zhai, J. Jiang, and X. Liu, "Single image deraining via scale-space invariant attention neural network," in *ACM MM*, 2020.

[40] H. Wang, Z. Yue, Q. Xie, Q. Zhao, Y. Zheng, and D. Meng, "From rain generation to rain removal," in *IEEE CVPR*, 2021.

[41] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Multi-stage progressive image restoration," in *IEEE CVPR*, 2021.

[42] H. Zhu, X. Peng, J. T. Zhou, S. Yang, V. Chanderasekh, L. Li, and J.-H. Lim, "Singe image rain removal with unpaired information: A differentiable programming perspective," in *AAAI*, 2019.

[43] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *IEEE ICCV*, 2017.

[44] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE TPAMI*, 2011.

[45] J. Pan, D. Sun, H. Pfister, and M.-H. Yang, "Deblurring images via dark channel prior," *IEEE TPAMI*, 2018.

[46] J. Pan, Z. Hu, Z. Su, and M.-H. Yang, "l_0 -regularized intensity and gradient prior for deblurring text images and beyond," *IEEE TPAMI*, 2017.

[47] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, and R. W. Lau, "Spatial attentive single-image deraining with a high quality real rain dataset," in *IEEE CVPR*, 2019.

[48] Q. Wu, L. Wang, K. N. Ngan, H. Li, and F. Meng, "Beyond synthetic data: A blind deraining quality assessment metric towards authentic rain image," in *IEEE ICIP*, 2019.

[49] Q. Wu, L. Wang, K. N. Ngan, H. Li, F. Meng, and L. Xu, "Subjective and objective de-raining quality assessment towards authentic rain image," *IEEE TCSVT*, 2020.

[50] H. Zhang and V. M. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *IEEE CVPR*, 2018.

[51] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "completely blind" image quality analyzer," *IEEE SPL*, 2013.

[52] J. Chen, C.-H. Tan, J. Hou, L.-P. Chau, and H. Li, "Robust video content alignment and compensation for rain removal in a cnn framework," in *IEEE CVPR*, 2018.

[53] M. Li, Q. Xie, Q. Zhao, W. Wei, S. Gu, J. Tao, and D. Meng, "Video rain streak removal by multiscale convolutional sparse coding," in *IEEE CVPR*, 2018.

[54] J. Liu, W. Yang, S. Yang, and Z. Guo, "Erase or fill? deep joint recurrent rain removal and reconstruction in videos," in *IEEE CVPR*, 2018.

[55] W. Yang, J. Liu, and J. Feng, "Frame-consistent recurrent video deraining with dual-level flow," in *IEEE CVPR*, 2019.

[56] K. Garg and S. K. Nayar, "Detection and removal of rain from videos," in *IEEE CVPR*, 2004.

[57] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *IEEE TIP*, 2017.

[58] J.-H. Kim, C. Lee, J.-Y. Sim, and C.-S. Kim, "Single-image deraining using an adaptive nonlocal means filter," in *IEEE ICIP*, 2013.

[59] L. Zhu, C.-W. Fu, D. Lischinski, and P.-A. Heng, "Joint bilayer optimization for single-image rain streak removal," in *IEEE ICIP*, 2017.

[60] Y. Yang and H. Lu, "Single image deraining via recurrent hierarchy enhancement network," in *ACM MM*, 2019.

[61] G. Li, X. He, W. Zhang, H. Chang, L. Dong, and L. Lin, "Non-locally enhanced encoder-decoder network for single image de-raining," in *ACM MM*, 2018.

[62] Y. Du, J. Xu, X. Zhen, M.-M. Cheng, and L. Shao, "Conditional variational image deraining," *IEEE TIP*, 2020.

[63] Y. Du, J. Xu, Q. Qiu, X. Zhen, and L. Zhang, "Variational image deraining," in *IEEE WACV*, 2020.

[64] H. Wang, Y. Wu, Q. Xie, Q. Zhao, Y. Liang, S. Zhang, and D. Meng, "Structural residual learning for single image rain removal," *Knowledge-Based Systems*, 2021.

[65] S. Deng, M. Wei, J. Wang, Y. Feng, L. Liang, H. Xie, F. L. Wang, and M. Wang, "Detail-recovery image deraining via context aggregation networks," in *IEEE CVPR*, 2020.

[66] K. Jiang, Z. Wang, P. Yi, C. Chen, B. Huang, Y. Luo, J. Ma, and J. Jiang, "Multi-scale progressive fusion network for single image deraining," in *IEEE CVPR*, 2020.

[67] H. Wang, M. Li, Y. Wu, Q. Zhao, and D. Meng, "A survey on rain removal from video and single image," *arXiv preprint arXiv:1909.08326*, 2019.

[68] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *EMNLP*, 2014.

[69] D. Ren, K. Zhang, Q. Wang, Q. Hu, and W. Zuo, "Neural blind deconvolution using deep priors," in *IEEE CVPR*, 2020.

[70] H. Dong, J. Pan, L. Xiang, Z. Hu, X. Zhang, F. Wang, and M.-H. Yang, "Multi-scale boosted dehazing network with dense feature fusion," in *IEEE CVPR*, 2020.

[71] L. Li, Y. Dong, W. Ren, J. Pan, C. Gao, N. Sang, and M.-H. Yang, "Semi-supervised image dehazing," *IEEE TIP*, 2020.

[72] M. Hong, Y. Xie, C. Li, and Y. Qu, "Distilling image dehazing with heterogeneous task imitation," in *IEEE CVPR*, 2020.

[73] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE TIP*, 2004.

[74] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS Autodiff Workshop:The Future of Gradient-based Machine Learning Softwareand Techniques*, 2017.

[75] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, 1992.

[76] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," in *NeurIPS*, 2020.

[77] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[78] J. U. Holger Caesar and V. Ferrari, "Cocostuff: Thing and stuff classes in context," in *IEEE CVPR*, 2018.

[79] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving video database with scalable annotation tooling," in *IEEE CVPR*, 2020.

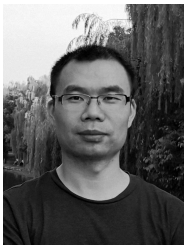
[80] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[81] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *IEEE CVPR*, 2017.

Xin Cui received the Master's Degree in Computer Technology from Tianjin University in 2022 and the Bachelor's Degree in Computer Science and Technology from China University of Mining and Technology in 2019. Now, she works as a Computer Vision Engineer at Intel China Research Institute. Her research interests include low-level computer vision with deep learning.

Cong Wang is currently a Ph.D. student at the Department of Computing of The Hong Kong Polytechnic University. He received the Master's Degree in Computational Mathematics from Dalian University of Technology in 2020 and the Bachelor's Degree in Mathematics and Applied Mathematics from Inner Mongolia University in 2017. His research interests include computer vision and deep learning.

Dongwei Ren received the Ph.D. degree in computer application technology from the Harbin Institute of Technology, Harbin, China, in 2017. From 2018 to 2021, he was an Assistant Professor with the College of Intelligence and Computing, Tianjin University. He is currently an Associate Professor with the School of Computer Science and Technology, Harbin Institute of Technology. His research interests include computer vision and deep learning.



Yunjin Chen received the BSc degree in applied physics from the Nanjing University of Aeronautics and Astronautics, China, and the MSc degree in optical engineering from the National University of Defense Technology, China, and the PhD degree in computer science from Graz University of Technology, Austria, in 2007, 2010, and 2015, respectively. Currently, he is a senior algorithm expert at Alibaba Cloud. His current research interests are image/video restoration, and its application at cloud video transcoding system.

Pengfei Zhu received the Ph.D. degree from The Hong Kong Polytechnic University, Hong Kong, China, in 2015. He is currently an Associate Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. His research interests are focused on machine learning and computer vision.