

RCDNet: An Interpretable Rain Convolutional Dictionary Network for Single Image Deraining

Hong Wang, Qi Xie, Qian Zhao, Yong Liang, and Deyu Meng, *Member, IEEE*.

Abstract—As a common weather, rain streaks adversely degrade the image quality and tend to negatively affect the performance of outdoor computer vision systems. Hence, removing rains from an image has become an important issue in the field. To handle such an ill-posed single image deraining task, in this paper, we specifically build a novel deep architecture, called rain convolutional dictionary network (RCDNet), which embeds the intrinsic priors of rain streaks and has clear interpretability. In specific, we first establish a RCD model for representing rain streaks and utilize the proximal gradient descent technique to design an iterative algorithm only containing simple operators for solving the model. By unfolding it, we then build the RCDNet in which every network module has clear physical meanings and corresponds to each operation involved in the algorithm. This good interpretability greatly facilitates an easy visualization and analysis on what happens inside the network and why it works well in inference process. Moreover, taking into account the domain gap issue in real scenarios, we further design a novel dynamic RCDNet, where the rain kernels can be dynamically inferred corresponding to input rainy images and then help shrink the space for rain layer estimation with few rain maps so as to ensure a fine generalization performance in the inconsistent scenarios of rain types between training and testing data. By end-to-end training such an interpretable network, all involved rain kernels and proximal operators can be automatically extracted, faithfully characterizing the features of both rain and clean background layers, and thus naturally lead to better deraining performance. Comprehensive experiments implemented on a series of representative synthetic and real datasets substantiate the superiority of our method, especially on its well generality to diverse testing scenarios and good interpretability for all its modules, as compared with state-of-the-art single image derainers both visually and quantitatively. Code is available in <https://github.com/hongwang01/RCDNet>.

Index Terms—Single image rain removal, dictionary learning, interpretable deep learning, generalization performance.

I. INTRODUCTION

IMAGES and videos captured in rainy scenes always suffer from noticeable visual degradations, tending to adversely affect outdoor computer vision tasks, such as automatic driving and video surveillance [1]. As a hot research topic, rain removal from images and videos has brought considerable attention to the research community [2]–[4]. In this work, we focus on the single image deraining task.

H. Wang, Q. Xie, Q. Zhao, and D. Meng are with School of Mathematics and Statistics and Ministry of Education Key Lab of Intelligent Networks and Network Security, Xi'an Jiaotong University, Shaanxi, P.R.China. E-mail: hongwang01@stu.xjtu.edu.cn, {xie.qi, timmy.zhaopian, dymeng}@mail.xjtu.edu.cn

Y. Liang is with the Faculty of Information Technology, Macau University of Science and Technology, Macau, China. E-mail:yliang@must.edu.mo

Q. Xie and D. Meng are corresponding authors.

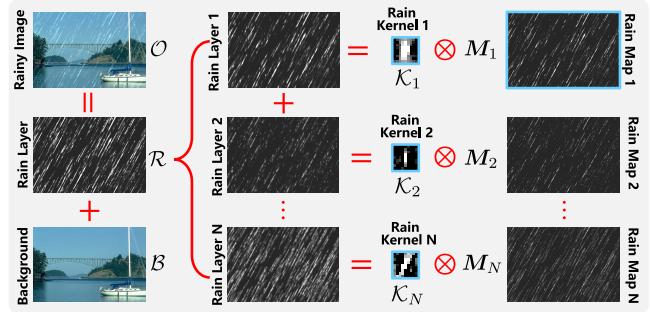


Fig. 1. Convolutional dictionary learning mechanism for rain layer.

Recent years have witnessed a significant progress in single image deraining, which can be mainly categorized into two research lines. One is traditional unsupervised (i.e., prior-based) method, which focuses on exploring the prior structures of background and rain layers to constrain the solution space of a carefully designed optimization model. Typically presented priors include frequency information [5], sparse representation [6]–[9], and local patch low-rankness [9]. Very recently, researchers explore that rain streaks repeatedly appear at different locations over a rainy image with similar local patterns, like shape, thickness, and direction [9], [10]. They formulate such an understanding (i.e., non-local self-similarity) as a convolutional dictionary learning model, where rain kernels are imposed on sparse rain maps, as intuitively depicted in Fig. 1. This idea has achieved state-of-the-art (SOTA) performance in video deraining when background frames are well extracted based on the temporal information and low-rankness prior in surveillance video sequences [11]. Albeit effective in certain specific scenarios, the rationality of these conventional deraining approaches largely depend on the reliability of such manually designed prior assumptions on the unknown background and rain streaks. However, with subjective and relatively simple forms, these hand-crafted priors cannot always comprehensively and adaptively reflect the complex and variant structures underlying real rainy images collected from different resources.

The other popular approach on this task is based on deep learning (DL). The main idea of current deep derainers is to utilize the pre-collected training samples to learn the mapping function from a rainy image to its corresponding rain-removed background layer with diverse network architectures, including simple CNN [12], [13], adversarial learning [14]–[16], recurrent and multi-stage networks [17]–[19], multi-scale fusion architectures [20]–[23], spatial attentive unit (SPANet) [24], encoder-decoder network [25]–[27], and complementary sub-

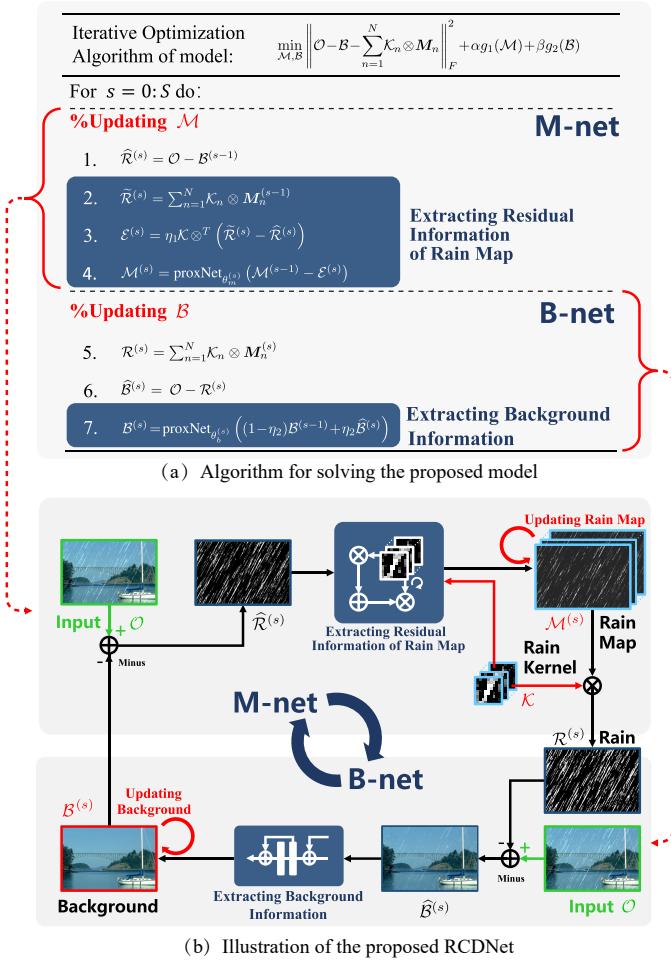


Fig. 2. (a) Rain convolutional dictionary (RCD) model and the proposed iterative solution algorithm. (b) Visual illustration of the proposed RCDNet corresponding to the algorithm in (a).

networks [28], [29]. Due to the powerful non-linear fitting capability of deep networks, these DL based techniques can generally achieve better deraining performance than conventional prior-based methods.

Albeit attaining a huge boost in deraining performance, these DL strategies still possess evident drawbacks. As seen, although the network architectures are becoming more diverse and complicated, there is still room for embedding the intrinsic prior knowledge of rain streaks so that one can design more interpretable network architecture and make the network output finely comply with expected prior properties. For example, the rain layers extracted by current DL based methods often contain some unexpected background details, which causes the over-smoothness of the derained results to a certain extent (as shown in Fig. 10). In fact, rational explicit constraints (e.g., sparsity and non-local similarity) on the rain layer should be helpful for alleviating this problem, which is however neglected by most of current deep single image derainers.

Another important issue lies in the generalization capability. Since the existing paired training sets are pre-collected and synthesized manually, it is inevitable that there is a bias about rain types between synthetic training data and real testing data. In this case, most of current deep deraining methods are prone to suffer from the over-fitting issue, since they generally adopt complicated and diverse network modules and put less

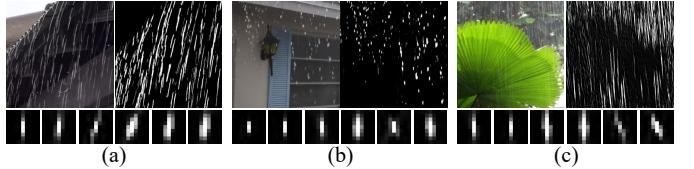


Fig. 3. Illustration of the rain kernels and rain layers estimated by DRCDNet for (a)-(c) three different testing samples from the real dataset, i.e., SPA-Data. Here DRCDNet is trained on the synthetic dataset, i.e., Rain100L.

emphasis on embedding the intrinsic prior constraint about rain layer. Thus, it is meaningful and also necessary to design a DL regime capable of finely fitting testing samples even when their rain types are different from training samples.

To address the aforementioned issues, a rational way is to embed the prior knowledge of rain layers into deep networks. This is because that prior structures can rationally regularize and constrain the solution space, which not only help avoid unexpected image details to be estimated as rain streaks, but also help alleviate the over-fitting problem of the network. In this paper, we explore the way to embed a well-studied prior model of rain layer (as shown in Fig. 1) into deep networks, and propose a novel network architecture with fine interpretability and generalization ability.¹ Specifically, our main contributions are summarized as follows:

Firstly, we utilize the intrinsic convolutional dictionary learning mechanism to encode rain shapes, and propose a concise rain convolutional dictionary (RCD) model for single rainy image. To solve it, we adopt the proximal gradient technique [31] to develop an optimization algorithm. Different from conventional solvers made up of complicated operators (e.g., Fourier transformation), the proposed algorithm only consists of simple computations (see Fig. 2 (a)) easy to be implemented by general network modules. This novel manner not only explicitly incorporates the intrinsic prior structures of rain streaks, but also facilitates us to easily unfold this algorithm into a deep network architecture.

Secondly, by unfolding every step of the algorithm, we construct an interpretable network for single image deraining, named as RCDNet. Every module in this network corresponds to the implementation operator of the proposed algorithm, and thus all network modules have clear physical interpretability as demonstrated in Fig. 2. Specifically, the RCDNet successively consists of M-net and B-net, updating the rain map \mathcal{M} and the background layer \mathcal{B} , respectively. All the operators in these two sub-networks are easy to understand and suitable for extracting rain layers, since they are consistent with a deraining algorithm.² Moreover, the rain layer extracted by RCDNet naturally complies with the prior constraints and can better exclude the background details as shown in Fig. 10.

¹As compared with our conference paper [30], the work has made substantial extensions. Specifically, a novel network with fine interpretability and generalization ability is designed. More model analysis, methodology expansions, visualization verifications, and experimental evaluations are provided. Especially, a core extension is that rain kernels (see Fig. 3) are adaptively inferred according to input rainy image. This dynamic prediction mechanism makes it possible to achieve better generalization performance even when the rain patterns are different between training and testing samples.

²This interpretable network design greatly facilitates us to analyze what happens during the network training, and understand the implementation mechanisms (see model visualization in Sec. VIII-B).

Thirdly, we further construct a dynamic rain convolutional dictionary network, called DRCDNet, for better generalization capability. Unlike RCDNet which estimates a large rain dictionary \mathcal{K} for the entire dataset, DRCDNet dynamically infers the rain kernel \mathcal{K} for each rainy sample. In this way, the number of the to-be-estimated rain map \mathcal{M} can be greatly reduced, and the hidden solution space for estimating rain layer is also greatly shrunk, which naturally improves the generalization ability. As presented in Fig. 3, in the cross-domain testing scenario, rain kernels are adaptively inferred according to the rain types of testing rainy images. To the best of our knowledge, we are the first to fully incorporate the intrinsic generative mechanism of rain layer into network design and also the first to design such a dynamic and flexible rain kernel prediction scheme.

Fourthly, under two kinds of testing settings (i.e., training/test domain match/mismatch settings), comprehensive experimental results substantiate the superiority of RCDNet and DRCDNet beyond conventional methods. Especially, attributed to the fine interpretability, not only the underlying rationality of such an interpretable network can be intuitively understood by general users through visualizing the amelioration process (e.g., the gradually rectified rain maps and background layers) over the network layers at all stages, but also the network can yield diverse rain kernels for describing rain shapes and proximal operators for delivering the prior knowledge of background and rain maps for a rainy image, facilitating their general availability to more real rainy images.

The paper is organized as follows. Sections II and III review the necessary notations and related works, respectively. Section IV presents the RCD model for rain removal as well as its optimization algorithm. Section V constructs the interpretable RCDNet, where the rain kernels are shared among the entire training samples, mainly usable under similar training-testing rain types. Section VI further builds the DRCDNet to adaptively infer rain kernels for diverse input rainy images, applicable to the mismatch of training-testing rain patterns. Section VII states the training details. Section VIII demonstrates the experimental evaluations to validate the superiority of the proposed network. The paper is finally concluded.

II. NOTATIONS AND PRELIMINARIES

For ease of understanding, we introduce some necessary notations and preliminaries as follows.

Denote $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ as a tensor of order N . The unfolding matrix $U_n^f(\mathcal{A}) \in \mathbb{R}^{I_n \times (I_1 \cdots I_{n-1} I_{n+1} \cdots I_N)}$ is composed by taking the mode- n vector of \mathcal{A} as its columns. This matrix can also be seen as the mode- n flattening of \mathcal{A} . The vectorization of \mathcal{A} is $\text{vec}(\mathcal{A}) \in \mathbb{R}^{I_1 I_2 \cdots I_N}$. All these can be easily achieved by the “reshape” function in PyTorch [32].

The symbol \otimes represents the 2-dimensional (2D) convolutional operation. When extending to the convolution in the form of tensor existed in deep networks,

$$\mathcal{Y} = \mathcal{C} \otimes \mathcal{X}, \quad (1)$$

where $\mathcal{C} \in \mathbb{R}^{k \times k \times N_o \times N_i}$, $\mathcal{X} \in \mathbb{R}^{H \times W \times N_i}$, and $\mathcal{Y} \in \mathbb{R}^{H \times W \times N_o}$. At the mode-3 of \mathcal{Y} , $\mathcal{Y}[:, :, j] = \sum_{n=1}^{N_i} \mathcal{C}[:, :, j, n] \otimes \mathcal{X}[:, :, n]$, $j = 1, 2, \dots, N_o$. The notation \otimes between

$\mathcal{C}[:, :, i, n]$ and $\mathcal{X}[:, :, n]$ is a 2D convolutional computation. The convolutional operation in Eq. (1) can be easily achieved by the off-the-shelf function “torch.nn.Conv2d” in PyTorch.

The symbol \otimes^d represents the depthwise convolutional operation. Specifically,

$$\mathcal{Z} = \mathcal{C} \otimes^d \mathcal{M}, \quad (2)$$

where $\mathcal{M} \in \mathbb{R}^{H \times W \times N}$ and $\mathcal{Z} \in \mathbb{R}^{H \times W \times N_o \times N_i \times N}$. Specifically, $\mathcal{Z}[:, :, j, k, n] = \mathcal{C}[:, :, j, k] \otimes \mathcal{M}[:, :, n]$, $j = 1, 2, \dots, N_o$, $k = 1, 2, \dots, N_i$, $n = 1, 2, \dots, N$. The depthwise convolutional operation in Eq. (2) can be easily performed through the group convolution by setting the parameter “group” in the function “torch.nn.Conv2d”.

III. RELATED WORK

In this section, we review the most related work, including video deraining and single image deraining.

A. Video Deraining Methods

Traditional Prior-based Methods. Garg *et al.* [33] made early attempt to study the visual effect of rain streaks on imaging systems, and proposed to adopt motion blur model and space-time model to describe the photometry and dynamics of rain streaks, respectively. Later, many physical properties of rain were investigated, including chromatic, temporal, spatial, and frequency domain characteristics [34]–[37]. In the past few years, researchers formulated intrinsic priors of rainy videos into model design, and adopted some iterative optimization algorithms for rain detection and removal. For example, the frequently-adopted prior knowledge include low-rankness among multi-frames [11], [38], [39], smoothness of background frame in the rain-perpendicular direction and that of rain streaks in the direction of raindrops [40], [41], sparsity and repeatability of rain [11], [42], [43]. Recently, Wei *et al.* [39] proposed to stochastically encode the rain layer patch as a mixture of Gaussian mixture model (P-MoG) for adapting a wide range of rain streaks. Li *et al.* [11] further investigated the characteristics of rain streaks, i.e., non-local self-similarity and multi-scale, and formulated them as a multi-scale convolutional sparse coding (MSCSC) model, achieving good performance on this video deraining task.

Deep Learning Based Methods. Recently, deep learning has attained tremendous success in various low-level vision tasks, such as image super-resolution [44], image denoising [45], [46], and image dehazing [47]. For video deraining, the early work [48] presented a convolutional neural network (CNN) architecture where superpixels were utilized as the basic element for content alignment. To improve the rain removal performance, Liu *et al.* [49] explored the wealth of temporal redundancy of videos and proposed a J4R-Net, which integrates rain degradation classification, spatial texture knowledge based rain removal, and temporal relevance based background reconstruction. To handle dynamic video contexts, the authors further designed a dynamic routing residue recurrent network [50]. Very recently, they embed a dual-level flow regularization into a two-stage recurrent network [51]. These deep supervised video derainers generally need to pre-collect paired training samples with high cost. To alleviate

this issue, Yang *et al.* [52] provides a self-learning deep framework. Although these approaches perform well, they generally cannot be finely applied to single image deraining task which has no temporal information.

B. Single Image Deraining Methods

Traditional Unsupervised Methods. To reconstruct background from a rainy image, the early attempts mainly focused on extracting rain-removed high frequency part (HFP) with various filtering strategies, such as guided filters [53], bilateral filtering [5], multiple guided filtering [54], guided L_0 smoothing filtering [55], and nonlocal means filtering [56]. During the past decades, researchers devoted themselves to designing prior terms for regularizing the to-the-estimated background/rain layer. E.g., Luo *et al.* [6] proposed an image patch-based discriminative sparse coding scheme. Li *et al.* [57] adopted GMM models to separate the background from rain streaks. Wang *et al.* [58] developed a 3-layer hierarchical scheme to categorize the HFP into rain/snow and rain/snow-free parts. Gu *et al.* [7] encoded the rain-free and rain parts as analysis and synthesis sparse representation models, respectively. Similarly, Zhang *et al.* [9] proposed to illustrate background and rain layers by learning a series of sparsity-based and low-rankness-based filters. Zhu *et al.* [8] modeled three regularizers from perspective of sparsity, gradient statistics, and similarity, respectively. The main drawback of these traditional model-based methods is that the hand-crafted prior assumptions are always subjective and limited, which might be possibly not able to faithfully and comprehensively reflect the complicated and diverse rain types collected from practice.

Deep Learning Based Methods. Recently, DL has achieved promising performance in this single image deraining task, showing evident superiority to conventional methods. In the early period, Fu *et al.* [12] proposed a CNN to extract discriminative features of rain in the HFP of a single rainy image and further developed a deep detail network which introduced the residual learning to speed up the training process [13]. Later, Zhang *et al.* [14] designed a rain density classifier aided multi-stream dense network. Further, the authors proposed a conditional generative adversarial network for better visual quality [15]. Recently, recurrent architectures have been intensively studied for rain removal in a stage-wise manner [17]–[19], [59], [60]. There are also some works incorporating the multi-scale learning by analyzing the self-similarity both in the same scale or across different scales [20]–[23], [61]. In [27] and [62], physical formulations are merged into the entire network design. The work [26] formulated an entangled representation learning model made up of a two-branched encoder. In [29], a detail-recovery image deraining network is proposed where rain removal and detail reconstruction are viewed as two separate tasks. A few researchers explored the rain imaging process and created some more realistic rainy images [2], [24], [27], [63]. To reduce the cost of pre-collecting abundant paired training samples and bridge the domain gap between synthetic and real data, semi-/un-supervised learning is also attracting much attention recently [16], [64]–[66]. At present, there is a new type of single image derainers that try to combine prior and DL methodologies, e.g., [67] and [68].

Albeit attaining significant success, most of these deep networks are assembled with some off-the-shelf modules in current DL toolkits and have less specific interpretability to this practical deraining task. Especially, they have not explicitly embedded sufficient prior knowledge underlying rain streaks into the network design. Hence, there is still large room for further performance improvement for this task. Besides, most of these deep derainers tend to suffer from the overfitting issue due to the training-testing bias about rain distributions.

IV. RCD MODEL FOR SINGLE IMAGE DERAINING

A. Model Formulation

Let $\mathcal{O} \in \mathbb{R}^{H \times W \times 3}$ denote an observed color rainy image, and we can rationally separate it as:

$$\mathcal{O} = \mathcal{B} + \mathcal{R}, \quad (3)$$

where \mathcal{B} and \mathcal{R} are the clear background and rain layers, respectively. H and W are the height and width of the image, respectively.³ To recover the background, most of current deep derainers focus on establishing complex network architectures to learn the mapping function between \mathcal{O} and \mathcal{B} (or \mathcal{R}).

Instead of designing complex networks heuristically, we first consider the traditional rain generation model which reflects the intrinsic prior structures of rain streaks [7], [9], [11]. In specific, with the rain convolutional dictionary (RCD) physical mechanism as visually illustrated in Fig. 1, the rain layer can be rationally expressed as:

$$\mathcal{R}^c = \sum_{n=1}^N \mathbf{K}_n^c \otimes \mathbf{M}_n, \quad c = 1, 2, 3, \quad (4)$$

where \mathcal{R}^c denotes the c^{th} color channel of \mathcal{R} . $\{\mathbf{K}_n^c\}_{n,c} \subset \mathbb{R}^{k \times k}$ is a set of rain kernels with the size $k \times k$ representing the repetitive local patterns of rain streaks, and $\{\mathbf{M}_n\}_n \subset \mathbb{R}^{H \times W}$ is the rain maps representing the locations where local patterns repeatedly appear. N is the number of rain kernels and \otimes is the 2D convolutional operation. For simplicity, throughout the paper, we rewrite Eq. (4) as:

$$\mathcal{R} = \sum_{n=1}^N \mathcal{K}_n \otimes \mathcal{M}_n = \mathcal{K} \otimes \mathcal{M}, \quad (5)$$

where $\mathcal{K}_n \in \mathbb{R}^{k \times k \times 3}$, $\mathcal{K} \in \mathbb{R}^{k \times k \times 3 \times N}$, and $\mathcal{M} \in \mathbb{R}^{H \times W \times N}$ are stacked by \mathbf{K}_n^c s, \mathcal{K}_n s, and \mathcal{M}_n s, respectively. The 2D convolutional operation \otimes between \mathcal{K}_n and \mathcal{M}_n is executed in the channel-wise manner, and the computation $\mathcal{K} \otimes \mathcal{M}$ is the extension of \otimes from 2D to tensor form.

We can rewrite the single rainy image model in Eq. (3) as:

$$\mathcal{O} = \mathcal{B} + \mathcal{K} \otimes \mathcal{M}. \quad (6)$$

Clearly, our goal is to estimate the \mathcal{K} , \mathcal{M} , and \mathcal{B} from \mathcal{O} . With sparse constraints on \mathcal{M} , it is easy to see that Eq. (5) can well model the sparsity and non-local similarity of rains.

³Note that Eq. (3) is an approximate model, which provides a rough direction for network learning. During the network implementation in Sec. V, we add an adjustment module to flexibly deal with complicated rainy images. Sec. VIII validates the effectiveness of our method in diverse rain scenarios.

The rain kernel \mathcal{K} can be viewed as a set of convolutional dictionary [10] for representing the repetitive and similar local patterns underlying rain streaks. In the training-testing domain match scenario where the rain patterns between training data and testing data are similar, a small number of rain kernels can finely represent a wide range of rain shapes [11]. Thus, they are the common knowledge for representing different rain types across all rainy images, and can be learned from abundant training samples by virtue of the strong learning ability of CNN with an end-to-end training manner (see more details in Sec. V). Thus, for predicting the clean background from an input rainy image, the key issue is to output \mathcal{M} and \mathcal{B} from \mathcal{O} with \mathcal{K} fixed. Correspondingly, the optimization problem is:

$$\min_{\mathcal{M}, \mathcal{B}} \|\mathcal{O} - \mathcal{B} - \mathcal{K} \otimes \mathcal{M}\|_F^2 + \lambda_1 p_1(\mathcal{M}) + \lambda_2 p_2(\mathcal{B}), \quad (7)$$

where λ_1 and λ_2 are trade-off parameters. $p_1(\cdot)$ and $p_2(\cdot)$ mean the penalty functions (i.e., regularizers) to deliver the prior structures of \mathcal{M} and \mathcal{B} , respectively.

The first term of the problem (7) is a rational approximate model for rain streak generation which well encodes the sparsity and non-local similarity of rain layer. Motivated by this, we believe that based on the solver of the problem (7), the constructed deep network modules are able to embed the prior of rain streak and constrain the space for rain layer estimation.

B. Optimization Algorithm

Deep unfolding technique is an intuitive way to combine the solver of optimization models with deep learning methods. This technique releases us from manually designing penalty terms, but also brings new challenges. The first one is to develop an optimization algorithm which only contains simple computations easy to be transformed to network modules.

Confronted with the convolutional dictionary representation model (7), the traditional solvers usually contain complex computations, e.g., the Fourier transform and inverse Fourier transform [10], [11], [69], tending to make this unfolding task difficult. We thus prefer to build a new algorithm where the to-the-estimated variables \mathcal{M} and \mathcal{B} are alternately updated by the proximal gradient technique [31]. In this way, the solution process only consists of simple computations, making it possible to easily achieve the transformation from the algorithm to network architectures. The details are as follows:

Updating \mathcal{M} : At the s^{th} iteration, the rain map \mathcal{M} can be updated by solving the quadratic approximation [31] of the problem (7) with regard to \mathcal{M} as:

$$\min_{\mathcal{M}} \frac{1}{2} \left\| \mathcal{M} - \left(\mathcal{M}^{(s-1)} - \eta_1 \nabla f \left(\mathcal{M}^{(s-1)} \right) \right) \right\|_F^2 + \lambda_1 \eta_1 p_1(\mathcal{M}), \quad (8)$$

where $\mathcal{M}^{(s-1)}$ is the updating result of the last iteration, η_1 is the stepsize parameter, and $f(\mathcal{M}^{(s-1)}) = \|\mathcal{O} - \mathcal{B}^{(s-1)} - \mathcal{K} \otimes \mathcal{M}^{(s-1)}\|_F^2$. Under general regularization terms [70], the solution of Eq. (8) is expressed as:

$$\mathcal{M}^{(s)} = \text{prox}_{\lambda_1 \eta_1} \left(\mathcal{M}^{(s-1)} - \eta_1 \nabla f \left(\mathcal{M}^{(s-1)} \right) \right). \quad (9)$$

Moreover, by substituting

$$\nabla f \left(\mathcal{M}^{(s-1)} \right) = \mathcal{K} \otimes^T \left(\mathcal{K} \otimes \mathcal{M}^{(s-1)} + \mathcal{B}^{(s-1)} - \mathcal{O} \right), \quad (10)$$

where \otimes^T denotes the transposed convolution,⁴ we can obtain the updating formula for \mathcal{M} as:⁵

$$\mathcal{M}^{(s)} = \text{prox}_{\lambda_1 \eta_1} \left(\mathcal{M}^{(s-1)} - \eta_1 \mathcal{K} \otimes^T \left(\mathcal{K} \otimes \mathcal{M}^{(s-1)} + \mathcal{B}^{(s-1)} - \mathcal{O} \right) \right), \quad (11)$$

where $\text{prox}_{\lambda_1 \eta_1}(\cdot)$ is the proximal operator dependent on the regularization term $p_1(\cdot)$ with respect to \mathcal{M} . Instead of being derived from manually-designed regularizer as in traditional prior-based methods, the form of the implicit proximal operator $\text{prox}_{\lambda_1 \eta_1}(\cdot)$ can be expressed through a convolutional network module and automatically adapted from training data in an end-to-end manner, which is described in Sec. V below.

Updating \mathcal{B} : Similarly, the quadratic approximation of the problem (7) with respect to \mathcal{B} is:

$$\min_{\mathcal{B}} \frac{1}{2} \left\| \mathcal{B} - \left(\mathcal{B}^{(s-1)} - \eta_2 \nabla g \left(\mathcal{B}^{(s-1)} \right) \right) \right\|_F^2 + \lambda_2 \eta_2 p_2(\mathcal{B}), \quad (12)$$

where $g(\mathcal{B}^{(s-1)}) = \|\mathcal{O} - \mathcal{B}^{(s-1)} - \mathcal{K} \otimes \mathcal{M}^{(s)}\|_F^2$. By substituting $\nabla g(\mathcal{B}^{(s-1)}) = \mathcal{K} \otimes \mathcal{M}^{(s)} + \mathcal{B}^{(s-1)} - \mathcal{O}$, it is easy to deduce that the final updating rule for \mathcal{B} is:⁵

$$\mathcal{B}^{(s)} = \text{prox}_{\lambda_2 \eta_2} \left((1 - \eta_2) \mathcal{B}^{(s-1)} + \eta_2 \left(\mathcal{O} - \mathcal{K} \otimes \mathcal{M}^{(s)} \right) \right), \quad (13)$$

where $\text{prox}_{\lambda_2 \eta_2}(\cdot)$ is the proximal operator correlated to the regularization term $p_2(\cdot)$ with respect to \mathcal{B} .

Based on this iterative algorithm, we can then construct our deep unfolding network as follows.

V. RAIN CONVOLUTIONAL DICTIONARY NETWORK

Inspired by the recent deep unfolding techniques in various tasks, e.g., deconvolution [71], compressed sensing [72], and image super-resolution [73], we build a novel network structure for this single image deraining task by separating and transforming each iterative step of the aforementioned algorithm as a specific form of network connection. Its specificity is that all network modules correspond to the algorithm operators, and thus the entire network has clear interpretability.

As shown in Fig. 4 (a), the proposed network consists of S stages, representing S iterations of the algorithm for solving (7). Each stage achieves sequential updates of \mathcal{M} and \mathcal{B} by the M-net and the B-net, respectively. Specifically, as displayed in Fig. 4 (b), in each stage of the network, the M-net takes the rainy image \mathcal{O} and the previous outputs $\mathcal{B}^{(s-1)}$ and $\mathcal{M}^{(s-1)}$ as inputs, and outputs an updated $\mathcal{M}^{(s)}$, and then the B-net takes \mathcal{O} and $\mathcal{M}^{(s)}$ as inputs, and outputs an updated $\mathcal{B}^{(s)}$.

From the updating rules (11) and (13), it is easily understood that the involved concise iterative computations can be naturally performed with commonly-used operators in normal networks [32]. The key issue of unrolling the algorithm is how to represent the two proximal operators $\text{prox}_{\lambda_1 \eta_1}(\cdot)$ and $\text{prox}_{\lambda_2 \eta_2}(\cdot)$. In this work, we adopt the deep residual network (ResNet) [74] to construct the operator as many other works [30], [75], [76] did.⁶ Then, we can separately decompose the updating rules for \mathcal{M} and \mathcal{B} into sub-steps

⁴The operation \otimes^T can be directly executed by the function as “torch.nn.ConvTranspose2d” in PyTorch [32].

⁵It can be proved that, with small enough η_1 and η_2 , Eq. (11) and Eq. (13) can both lead to the decrease of the objective function in (7) [31].

⁶Please refer to the supplementary materials for more analysis.

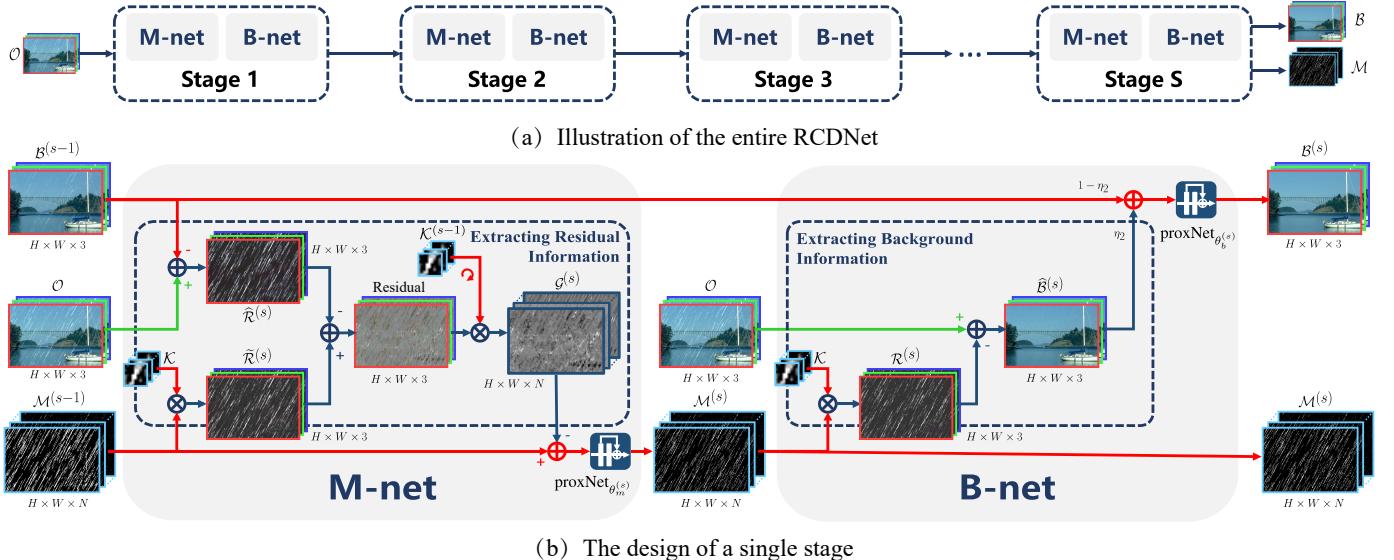


Fig. 4. (a) The proposed RCDNet with S stages. The network takes a rainy image \mathcal{O} as input and outputs the learned rain map \mathcal{M} and background image \mathcal{B} . (b) Illustration of the network architecture at the s^{th} stage. Each stage consists of M-net and B-net to accomplish the update of rain map \mathcal{M} and background layer \mathcal{B} , respectively. The images are better to be observed by zooming in on screen.

and achieve the following procedures for the s^{th} stage of the proposed rain convolutional dictionary network (RCDNet):

$$\text{M-net : } \begin{cases} \widehat{\mathcal{R}}^{(s)} = \mathcal{O} - \mathcal{B}^{(s-1)}, \\ \widehat{\mathcal{R}}^{(s)} = \mathcal{K} \otimes \mathcal{M}^{(s-1)}, \\ \mathcal{G}^{(s)} = \eta_1 \mathcal{K} \otimes^T (\widetilde{\mathcal{R}}^{(s)} - \widehat{\mathcal{R}}^{(s)}), \\ \mathcal{M}^{(s)} = \text{proxNet}_{\theta_m^{(s)}} (\mathcal{M}^{(s-1)} - \mathcal{G}^{(s)}), \end{cases} \quad (14)$$

$$\text{B-net : } \begin{cases} \mathcal{R}^{(s)} = \mathcal{K} \otimes \mathcal{M}^{(s)}, \\ \widehat{\mathcal{B}}^{(s)} = \mathcal{O} - \mathcal{R}^{(s)}, \\ \mathcal{B}^{(s)} = \text{proxNet}_{\theta_b^{(s)}} ((1 - \eta_2) \mathcal{B}^{(s-1)} + \eta_2 \widehat{\mathcal{B}}^{(s)}), \end{cases} \quad (15)$$

where $\text{proxNet}_{\theta_m^{(s)}}(\cdot)$ and $\text{proxNet}_{\theta_b^{(s)}}(\cdot)$ are two ResNets consisting of several Resblocks with the parameters $\theta_m^{(s)}$ and $\theta_b^{(s)}$ at the s^{th} stage, respectively.

We can then design the network architecture, as shown in Fig. 4, by transforming the operators in (14) and (15) step-by-step. All the parameters involved can be automatically fit from training data, including $\{\theta_m^{(s)}, \theta_b^{(s)}\}_{s=1}^S$, rain kernels \mathcal{K} , η_1 , and η_2 . Considering that in some scenarios, the composition of rainy images is complicated. Thus we further refine the reconstructed result $\mathcal{B}^{(s)}$ by feeding it into an extra ResNet which has the same structure as $\text{proxNet}_{\theta_b^{(s)}}(\cdot)$.

It should be indicated that every module has its specific physical meanings. As shown in Fig. 4 (b), at every stage, the M-net accomplishes the learning of the ameliorative gradient direction $\mathcal{G}^{(s)}$ of rain maps and further helps rectify the \mathcal{M} . Specifically, $\widehat{\mathcal{R}}^{(s)}$ is the rain layer estimated with the previous background $\mathcal{B}^{(s-1)}$, and $\widetilde{\mathcal{R}}^{(s)}$ is the rain layer achieved by the generative model (5) with the estimated $\mathcal{M}^{(s-1)}$. Then the M-net calculates the residual information between the two rain layers obtained in this way, and extracts the gradient updating direction $\mathcal{G}^{(s)}$ of rain maps with the transposed convolution of rain kernels to update the rain map. Next, the B-net recovers the background $\widehat{\mathcal{B}}^{(s)}$ estimated with current rain kernel and rain maps $\mathcal{M}^{(s)}$, and fuses such estimated $\widehat{\mathcal{B}}^{(s)}$ with the previously estimated $\mathcal{B}^{(s-1)}$ by weighted parameters η_2 and

$(1 - \eta_2)$ to get the updated background $\mathcal{B}^{(s)}$. Clearly, such an interpretable network design makes it easy to intuitively observe what happens inside the network flow and understand the intrinsic implementation mechanisms.⁷

Remark 1: As analyzed in Sec. I, for most of current deep single image derainers, the reconstructed background images are often blurred. However, RCDNet can alleviate this problem, since the involved RCD model (5) can regularize the extracted rain layer and help distinguish rain streaks from background details. This is finely validated by Fig. 10 below.

VI. DYNAMIC RCDNET

As seen, the large rain dictionary \mathcal{K} in RCDNet is sharing among the entire dataset. Such settings would be more applicable for the consistent case that training and testing datasets are with similar rain patterns. To further enhance the generalization capability, we construct a dynamic rain convolutional dictionary network, called DRCDNet. Specifically, in DRCDNet, the rain kernel \mathcal{K} is dynamically inferred for each rainy image. In this way, the number of the to-be-estimated rain map \mathcal{M} can be greatly reduced, and the hidden solution space for estimating rain layer is also greatly shrunk, which naturally improves the generalization ability. For clarity, RCDNet in Sec. V is named as consistent RCDNet (CRCDNet). The details of DRCDNet are as follows.

Model Formulation. For DRCDNet, we reformulate the rain kernel \mathcal{K}_n in Eq. (5) as:

$$\mathcal{K}_n = \mathcal{D} \alpha_n, \quad (16)$$

where $\mathcal{D} \in \mathbb{R}^{k \times k \times 3 \times d}$ is rain kernel dictionary representing common knowledge for conveying variant rain types across the entire training set, and d is the number of rain kernels in this dictionary. $\alpha_n \in \mathbb{R}^d$ denotes the weighting coefficient.⁸

⁷More details about network design are described in supplementary file.

⁸ $\mathcal{D} \alpha_n$ is computed between the tensor $\mathcal{D} \in \mathbb{R}^{k \times k \times 3 \times d}$ and the vector $\alpha_n \in \mathbb{R}^d$, expressed as $\mathcal{K}_n = \sum_{i=1}^d \mathcal{D}[:, :, :, i] \odot \alpha_n[i]$, which can be easily achieved by combining the “reshape” operation and the function as “torch.matmul”. \odot is point-wise multiplication.

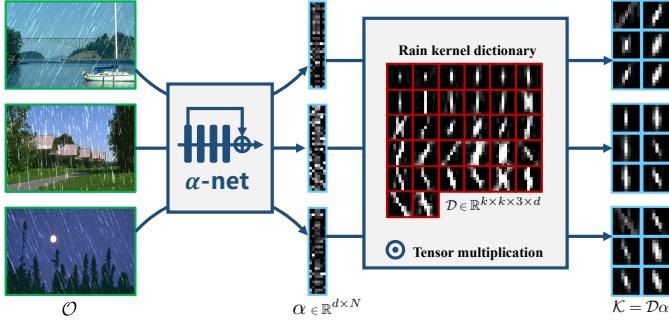


Fig. 5. Dynamic rain kernel inference in DRCDNet. The dictionary \mathcal{D} for DRCDNet is pre-trained like the kernel \mathcal{K} for RCDNet. But the involvement of α helps the DRCDNet possess adaptive learning capability, with dynamically predicted rain kernels \mathcal{K} for different testing rainy images \mathcal{O} .

Instead of pre-training and then fixing rain kernels \mathcal{K}_n s for any testing rainy image as CRCDNet does, DRCDNet can flexibly infer the rain kernels \mathcal{K}_n s for every rainy sample by dynamically updating α_n s. One can refer to Fig. 5 for easy understanding. This motivation is finely verified in Sec. VIII-B.

Then the rainy image in Eq. (6) is rewritten as:

$$\mathcal{O} = \mathcal{B} + \mathcal{D}\alpha \otimes \mathcal{M}, \quad (17)$$

where $\alpha \in \mathbb{R}^{d \times N}$ is stacked by α_n s and $\mathcal{D}\alpha \in \mathbb{R}^{k \times k \times 3 \times N}$.

Similar to the dictionary \mathcal{K} in CRCDNet, the common dictionary \mathcal{D} in this dynamic case can be automatically learned from training samples in an end-to-end manner by virtue of the strong non-linear fitting ability of deep network. Our goal is to estimate the unknown \mathcal{M} , \mathcal{B} , and α from \mathcal{O} . Thus the corresponding optimization problem is formulated as:

$$\begin{aligned} & \min_{\mathcal{M}, \mathcal{B}, \alpha} \|\mathcal{O} - \mathcal{B} - \mathcal{D}\alpha \otimes \mathcal{M}\|_F^2 + \lambda_1 p_1(\mathcal{M}) + \lambda_2 p_2(\mathcal{B}) + \lambda_3 p_3(\alpha), \\ & \text{s.t. } \|\alpha_n\|_2 = 1, n = 1, 2, \dots, N, \end{aligned} \quad (18)$$

where the explicit constraint, i.e., $\|\alpha_n\|_2 = 1$, is used to control the energy of weighting coefficient α_n so as to avoid affecting the learning of rain kernels. Similar to $p_1(\cdot)$ and $p_2(\cdot)$, we also prefer to automatically fit the regularizer $p_3(\cdot)$ for α via deep unrolling network modules.

Optimization Algorithm. With the similar algorithm for the problem (7) given in Sec. IV-B, we can easily derive the updating rules of \mathcal{M} and \mathcal{B} for the problem (18) as:

$$\begin{aligned} & \mathcal{M}^{(s)} = \\ & \text{prox}_{\lambda_1 \eta_1} \left(\mathcal{M}^{(s-1)} - \eta_1 (\mathcal{D}\alpha^{(s-1)}) \otimes^T (\mathcal{D}\alpha^{(s-1)} \otimes \mathcal{M}^{(s-1)} + \mathcal{B}^{(s-1)} - \mathcal{O}) \right), \end{aligned} \quad (19)$$

$$\mathcal{B}^{(s)} = \text{prox}_{\lambda_2 \eta_2} \left((1 - \eta_2) \mathcal{B}^{(s-1)} + \eta_2 \left(\mathcal{O} - \mathcal{D}\alpha^{(s-1)} \otimes \mathcal{M}^{(s)} \right) \right). \quad (20)$$

As for α , the quadratic approximation of the problem (18) with respect to α is derived as:

$$\min_{\alpha \in \Omega} \frac{1}{2} \left\| \alpha - \left(\alpha^{(s-1)} - \eta_3 \nabla h \left(\alpha^{(s-1)} \right) \right) \right\|_F^2 + \lambda_3 \eta_3 p_3(\alpha), \quad (21)$$

where $\Omega = \{\alpha | \|\alpha_n\|_2 = 1, n = 1, 2, \dots, N\}$. $h(\alpha^{(s-1)}) = \|\mathcal{O} - \mathcal{B}^{(s-1)} - \mathcal{D}\alpha^{(s-1)} \otimes \mathcal{M}^{(s)}\|_F^2$. Then, we can derive that

$$\frac{\partial h(\alpha^{(s-1)})}{\partial \alpha_n} = \left(U_4^f \left(\mathcal{D} \otimes^d \mathcal{M}_n^{(s)} \right) \right) \text{vec} \left(\mathcal{D}\alpha^{(s-1)} \otimes \mathcal{M}^{(s)} + \mathcal{B}^{(s)} - \mathcal{O} \right), \quad (22)$$

where the computed result of $\mathcal{D} \otimes^d \mathcal{M}_n^{(s)}$ has the size of $H \times W \times 3 \times d$. $U_4^f(\cdot)$ represents unfolding the result at the 4th mode and the resulted shape is $d \times 3HW$.

Clearly, the updating rule for α is finally derived as:

$$\alpha^{(s)} = \text{prox}_{\lambda_3 \eta_3} \left(\alpha^{(s-1)} - \eta_3 \nabla h \left(\alpha^{(s-1)} \right) \right), \quad (23)$$

$$\text{where } \nabla h \left(\alpha^{(s-1)} \right) = \left[\frac{\partial h(\alpha^{(s-1)})}{\partial \alpha_1}, \frac{\partial h(\alpha^{(s-1)})}{\partial \alpha_2}, \dots, \frac{\partial h(\alpha^{(s-1)})}{\partial \alpha_N} \right].$$

Such concise iterative rules (19)(20)(23) easily facilitate us to unfold this iterative algorithm into a deep interpretable network as follows. Note that the constraint space Ω can be easily achieved by embedding a normalization operation into the implicit proximal operator $\text{prox}_{\lambda_3 \eta_3}(\cdot)$.

Network Design. Similar to Sec. V, we subsequently decompose these updating rules (19)(20)(23) into sub-steps and achieve the following procedures for the s^{th} stage of the proposed DRCDNet:

$$\text{M-net : } \begin{cases} \widehat{\mathcal{R}}^{(s)} = \mathcal{O} - \mathcal{B}^{(s-1)}, \\ \widetilde{\mathcal{R}}^{(s)} = \mathcal{D}\alpha^{(s-1)} \otimes \mathcal{M}^{(s-1)}, \\ \mathcal{G}^{(s)} = \eta_1 \mathcal{D}\alpha^{(s-1)} \otimes^T \left(\widetilde{\mathcal{R}}^{(s)} - \widehat{\mathcal{R}}^{(s)} \right), \\ \mathcal{M}^{(s)} = \text{proxNet}_{\theta_m^{(s)}} \left(\mathcal{M}^{(s-1)} - \mathcal{G}^{(s)} \right), \end{cases} \quad (24)$$

$$\text{B-net : } \begin{cases} \mathcal{R}^{(s)} = \mathcal{D}\alpha^{(s-1)} \otimes \mathcal{M}^{(s)}, \\ \widehat{\mathcal{B}}^{(s)} = \mathcal{O} - \mathcal{R}^{(s)}, \\ \mathcal{B}^{(s)} = \text{proxNet}_{\theta_b^{(s)}} \left((1 - \eta_2) \mathcal{B}^{(s-1)} + \eta_2 \widehat{\mathcal{B}}^{(s)} \right), \end{cases} \quad (25)$$

$$\text{alpha-net : } \begin{cases} \widehat{\mathcal{R}}^{(s)} = \mathcal{O} - \mathcal{B}^{(s)}, \\ \widetilde{\mathcal{R}}^{(s)} = \mathcal{D}\alpha^{(s-1)} \otimes \mathcal{M}^{(s)}, \\ \mathcal{G}_{\alpha}^{(s)} = \eta_3 \left(U_4^f \left(\mathcal{D} \otimes^d \mathcal{M}_n^{(s)} \right) \right) \text{vec} \left(\widetilde{\mathcal{R}}^{(s)} - \widehat{\mathcal{R}}^{(s)} \right), \\ \alpha^{(s)} = \text{proxNet}_{\theta_a^{(s)}} \left(\alpha^{(s-1)} - \mathcal{G}_{\alpha}^{(s)} \right), \end{cases} \quad (26)$$

where $\mathcal{G}_{\alpha}^{(s)} = [\mathcal{G}_{\alpha_1}^{(s)}, \mathcal{G}_{\alpha_2}^{(s)}, \dots, \mathcal{G}_{\alpha_N}^{(s)}]$. The parameters in Eqs. (24) and (25) have been explained in Eqs. (14) and (15). For $\text{proxNet}_{\theta_a^{(s)}}(\cdot)$, it is a ResNet only consisting of one Resblock with the parameters $\theta_a^{(s)}$ at the s^{th} stage. Specifically, the Resblock simply contains two linear layers followed by a normalization operation at the second dimension of α .⁹

Then, by transforming the operators in (24)(25)(26) step-by-step, we can easily construct the DRCDNet. Clearly, at each stage, the DRCDNet is composed of three sub-networks, i.e., M-net, B-net, and alpha-net. Specifically, by comparing (14)(15) with (24)(25), respectively, we can directly construct the M-net and B-net by replacing the rain kernel \mathcal{K} in Fig. 4(b) with $\mathcal{D}\alpha^{(s-1)}$. For alpha-net, its structure is built as shown in Fig. 6. In this DRCDNet, all the involved parameters, including $\{\theta_m^{(s)}, \theta_b^{(s)}, \theta_a^{(s)}\}_{s=1}^S$, rain kernel dictionary \mathcal{D} , η_1 , η_2 , and η_3 , can be automatically learned from training data.

Remark 2: Similar to the CRCDNet, all the network modules in DRCDNet are correspondent to the iterative computations (19)(20)(23) and thus the DRCDNet also has clear interpretability. Compared with CRCDNet, DRCDNet has specific merits. First, at the testing phase, although the common rain kernel dictionary \mathcal{D} is pre-trained and fixed, the dynamic inference of α makes it possible to achieve the flexible prediction

⁹Please refer to the supplementary files for more details about DRCDNet.

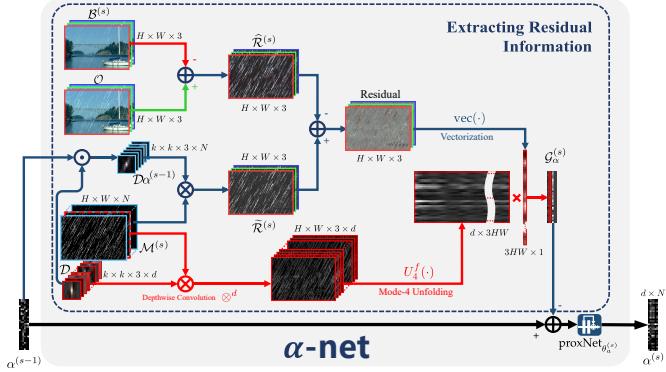


Fig. 6. The α -net of the proposed DRCDNet at the s^{th} stage. Every network module is correspondingly constructed based on Eq. (26).

of rain kernel $\mathcal{D}\alpha$ according to the rain types of variant testing rainy images. Besides, it is easily understood that as compared with the entire dataset, depicting a rainy image generally needs few rain kernels. Hence, we can choose smaller N for DRCDNet. Equivalently, the channel number of rain map \mathcal{M} is also smaller than that in CRCDNet. Under this setting, the hidden space for estimating rain layer is greatly shrunk, which naturally helps improve the generalization ability. This is comprehensively substantiated in Sec. VIII-D.

Remark 3: Compared with the general channel attention mechanism, the α -net has specific characteristics. First, instead of weighting feature maps on the channel dimension, we focus on weighting the rain kernel dictionary \mathcal{D} , which would save the computational cost. Second, the α -net is built based on an optimization algorithm and thus it has clear physical interpretability. Third, as shown in Fig. 5, the obtained rain kernel \mathcal{K} has obvious physical meanings, which validates the effectiveness of such weighting operators.

VII. NETWORK TRAINING

Training Loss. For simplicity, we adopt the mean square error (MSE) [53] for the learned background and the rain layer at every stage as the training objective function:

$$L = \sum_{s=0}^S \rho_s \left\| \mathcal{B}^{(s)} - \mathcal{B} \right\|_F^2 + \sum_{s=1}^S \gamma_s \left\| \mathcal{O} - \mathcal{B} - \mathcal{R}^{(s)} \right\|_F^2, \quad (27)$$

where $\mathcal{B}^{(s)}$ and $\mathcal{R}^{(s)}$ separately denote the derained result and extracted rain layer at the s^{th} stage ($s = 0, 1, \dots, S$), as expressed in Eq. (15) for CRCDNet and Eq. (25) for DRCDNet. $\mathcal{B}^{(0)}$ is initialized by a convolutional operator on \mathcal{O} . ρ_s and γ_s are tradeoff parameters, and simply set as $\rho_s = \gamma_s = 1$ and others as 0.1 for all experiments to make the outputs at the final stage play a dominant role. More parameter settings are discussed in supplementary file.

Implement Details. We use PyTorch [32] to implement our network training, based on a NVIDIA GeForce GTX 1080Ti GPU. For both CRCDNet and DRCDNet, we adopt the Adam optimizer [77] with the batch size of 10 and the patch size of 64×64 . The initial learning rate is 1×10^{-3} and divided by 5 every 25 epochs. The total epoch is 100. It is worth mentioning that in all experiments, these parameter settings always keep the same. This would show the favorable robustness and generality of our method.

TABLE I
BENCHAMRK DATASETS DESCRIPTIONS. THE CORRESPONDING RAIN TYPES ARE OBTAINED FROM DIFFERENT SOURCES.

Dataset	Training pairs#	Testing pairs#	Scenario
Rain100L [19]	200	100	Synthetic
Rain100H [19]	1800	100	Synthetic
Rain1400 [13]	12600	1400	Synthetic
Dense10 [64]	0	10	Synthetic
Sparse10 [64]	0	10	Synthetic
SPA-Data [24]	638492	1000	Real
Internet-Data [24]	0	146 (no label)	Real
MPID_Rain+Mist(R) [2]	0	30 (no label)	Real

TABLE II
EFFECT OF STAGE NUMBER S ON THE PERFORMANCE OF CRCDNET.

Stage #	$S=0$	$S=2$	$S=8$	$S=11$	$S=17$	$S=20$
PSNR	35.93	38.46	39.60	39.81	40.00	39.91
SSIM	0.9689	0.9813	0.9850	0.9855	0.9860	0.9858

VIII. EXPERIMENTAL RESULTS

We first conduct model verification to verify the working mechanisms of the proposed network. Then we evaluate the superiority of CRCDNet by comparing it with other SOTA single image derainers based on synthetic datasets. Finally, the performance of DRCDNet is verified by generalization experiments where rain patterns are obviously different between training samples and testing ones.

A. Details Explanations

Benchmark Datasets. Eight datasets are adopted as listed in Table I, including 5 synthesized ones and 3 real ones.¹⁰

Comparison Methods. We compare our network with current SOTA single image derainers, including:¹¹

- 1) Prior-based methods: DSC [6] and JCAS [7];
- 2) Deep learning methods: Clear [12], DDN [13], RESCAN [17], PReNet [18], SPANet [24], and JORDER_E [19];
- 3) Semi-supervised method: SIRR [64].

Performance Metrics. For paired data, the classical metrics are PSNR [78] and SSIM [79]. Since the human visual system is sensitive to the Y channel of a color image in YCbCr space, similar to [15], [18], [57], we also compute PSNR and SSIM based on the luminance channel. While for unlabeled data, we adopt the non-reference indicators, i.e., naturalness image quality evaluator (NIQE) [80] and blind/referenceless image spatial quality evaluator (BRISQUE) [81]. The higher PSNR and SSIM are, and the lower NIQE and BRISQUE are, the better the rain-removed result is.

B. Model Verification

Here we utilize Rain100L to execute the model verification.

Stage Number S . Table II reports the effect of stage number S on deraining performance of the proposed CRCDNet. Here, $S = 0$ represents the fact that without adopting the RCD mechanism, the initialization $\mathcal{B}^{(0)}$ is directly regarded as the final rain-removed result. Taking $S = 0$ as a baseline, it is easily seen that with only 2 stages, our method achieves significant rain removal performance improvement, substantiating the essential role of the constructed M-net and B-net.

¹⁰Detailed explanations are included in supplementary material.

¹¹The code/project links can be found from <https://github.com/hongwang01/Video-and-Single-Image-Deraining>.

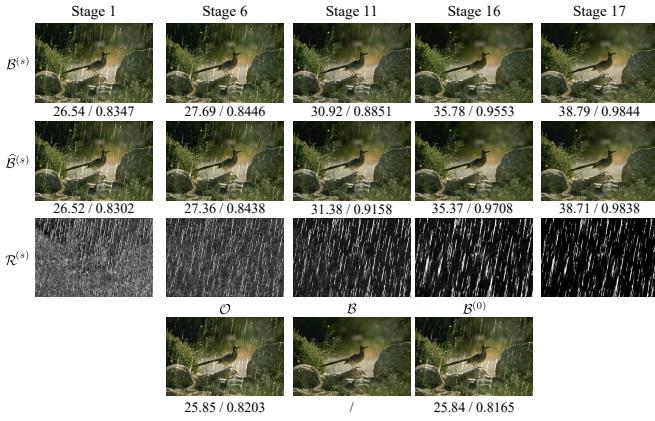


Fig. 7. Visualization of the recovery background $B^{(s)}$, $\hat{B}^{(s)}$ of CRCDNet and the rain layer $R^{(s)}$ at different stages. The stage number S is 17. PSNR/SSIM are listed below corresponding results for easy reference.

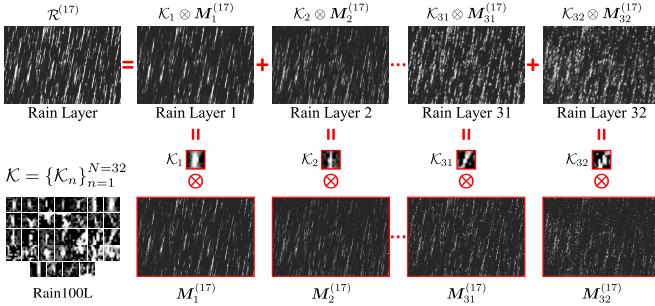


Fig. 8. At the final stage $s = 17$, the extracted rain layer, rain kernels \mathcal{K}_n , and rain maps M_n for the input \mathcal{O} in Fig. 7. The lower left is the rain kernels \mathcal{K} learned by CRCDNet based on Rain100L training pairs. In CRCDNet, $N = 32$.

We also find that when $S = 20$, the performance gain is not obvious, since larger S would make the gradient propagation more difficult. Based on this observation, we easily set S as 17 for the CRCDNet throughout all our experiments. More results and discussions are listed in supplemental file.

Network Visualization. We then visually show how the interpretability of CRCDNet facilitates an easy analysis on the implementation mechanism inside the network modules. Fig. 7 presents the extracted background $B^{(s)}$ (1st row), $\hat{B}^{(s)}$ (2nd row) that represents the role of M-net in helping restore clean background, and rain layer $R^{(s)}$ (3rd row) at different stages of CRCDNet. We can find that with the increase of s , $R^{(s)}$ covers more rain streaks and fewer image details, and $\hat{B}^{(s)}$ and $B^{(s)}$ are also gradually ameliorated. These should be attributed to the proper guidance of the RCD prior for rain streaks and the mutual promotion of M-net and B-net that enables the CRCDNet to be evolved to a right direction.

RCD Model Visualization. For the input \mathcal{O} in Fig. 7, the rain kernels and the rain maps learned by CRCDNet are presented in Fig. 8. Clearly, the CRCDNet finely extracts proper rain layers explicitly complying with the RCD model (5). This not only verifies the reasonability of our method but also manifests the peculiarity of our proposal. On one hand, we utilize a M-net to learn sparse rain maps instead of directly learning rain streaks that makes learning process easier. On the other hand, we exploit training data to automatically learn rain kernels representing general repetitive local patterns of rain with diverse shapes. This facilitates their general availability

to more real-world rainy images.

Rain Kernel Visualization. By training DRCDNet on Rain100L, the learned rain kernel dictionary \mathcal{D} is shown at the right of Fig. 9. As compared with the dictionary \mathcal{K} learned by CRCDNet in Fig. 8, we can easily find that the rain kernels in \mathcal{D} are fairly diverse. With the trained model on Rain100L, we test typical rainy samples from different sources, including training/test domain match cases (a)-(c) and mismatch cases (d)-(f). As shown in each column of (a)-(f), the extracted rain layers (2nd row) contain less background details, and the inferred rain kernels (3rd row) are finely in accordant with the rain patterns (e.g., directions, scales, thickness) in input rainy images (1st row). Besides, we can also observe that the rain kernels (3rd row) for every testing sample are not simply selected from \mathcal{D} , and they are adaptively inferred by DRCDNet, even with new rain patterns not in \mathcal{D} . This not only validates the effectiveness of the dynamic RCD modelling manner (17) for rain layer, but also reflects the advantages of the DRCDNet over adaptive inference. Compared with CRCDNet where rain kernels are always fixed as the dictionary \mathcal{K} for any testing sample, such adaptive strategy makes DRCDNet have potential to obtain better generalization ability. This will be comprehensively validated in the following experiments.

C. Training-test Domain Match Experiments

In this section, we evaluate the proposed CRCDNet and DRCDNet in the case that the rain types of testing data are consistent with that of training data, based on the benchmark datasets including Rain100L, Rain100H, and Rain1400.

Performance Comparison on Rain100L. Fig. 10 illustrates the deraining performance of all competing methods on a rainy image from Rain100L. As shown, for background recovery, traditional model-based DSC and JCAS leave obvious rain streaks, and deep derainers lose certain useful image textures. However, the proposed CRCDNet and DRCDNet perform better in sufficiently removing the rain streaks and finely preserving image details. *Moreover, we can easily observe that the rain layers extracted by CRCDNet and DRCDNet both contain fewer unexpected background details, which validates the reliability of embedding the RCD prior constraints.*

Performance Comparison on Rain100H. Fig. 11 presents the derained results of all competing methods on one typical rainy image from Rain100H. It is easy to see that the rain removal performance of most comparison methods is adversely degraded by heavy rains. Comparatively, the proposed CRCDNet, as well as DRCDNet, still achieve better visual quality and quantitative measures.

Performance Comparison on Rain1400. Fig. 12 displays the rain-removed results on Rain1400 with 14 rain types. We can find that our CRCDNet and DRCDNet both have better capability in effectively dealing with background restoration and rain removal, which is benefitted by the regularization of the RCD model for rain layer.

Table III reports the average PSNR and SSIM computed on the entire testing data of each synthesized dataset. It is seen that in this training-testing domain match case, our CRCDNet attains significant deraining performance on each evaluation dataset and DRCDNet performs comparable to CRCDNet.

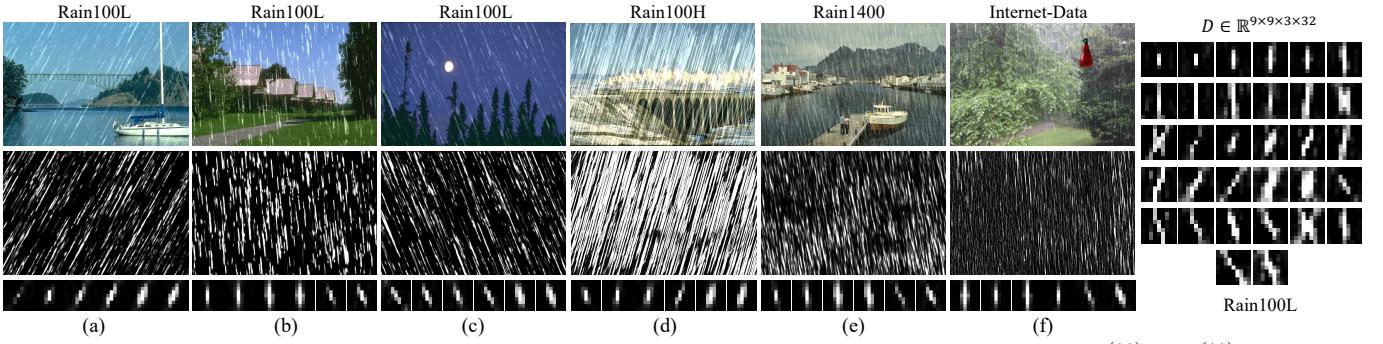


Fig. 9. The left: for each column in (a)-(f), (1st row) rainy images, (2nd row) the extracted rain layers $\mathcal{R}^{(11)} = \sum_{n=1}^{N=6} \mathcal{D}\alpha_n^{(11)} \otimes M_n^{(11)}$, and (3rd row) the corresponding rain kernels $\mathcal{D}\alpha^{(11)} = \{\mathcal{D}\alpha_n^{(11)}\}_{n=1}^N$ dynamically predicted by DRCDNet. The right: the rain kernel dictionary \mathcal{D} learned by DRCDNet based on Rain100L training set. Especially, the rainy images in (a)-(c) are from Rain100L testing set and the ones in (d)-(f) are from other testing set. In DRCDNet, $S = 11$, $d = 32$, and $N = 6$. More explanations are included in the supplemental file.

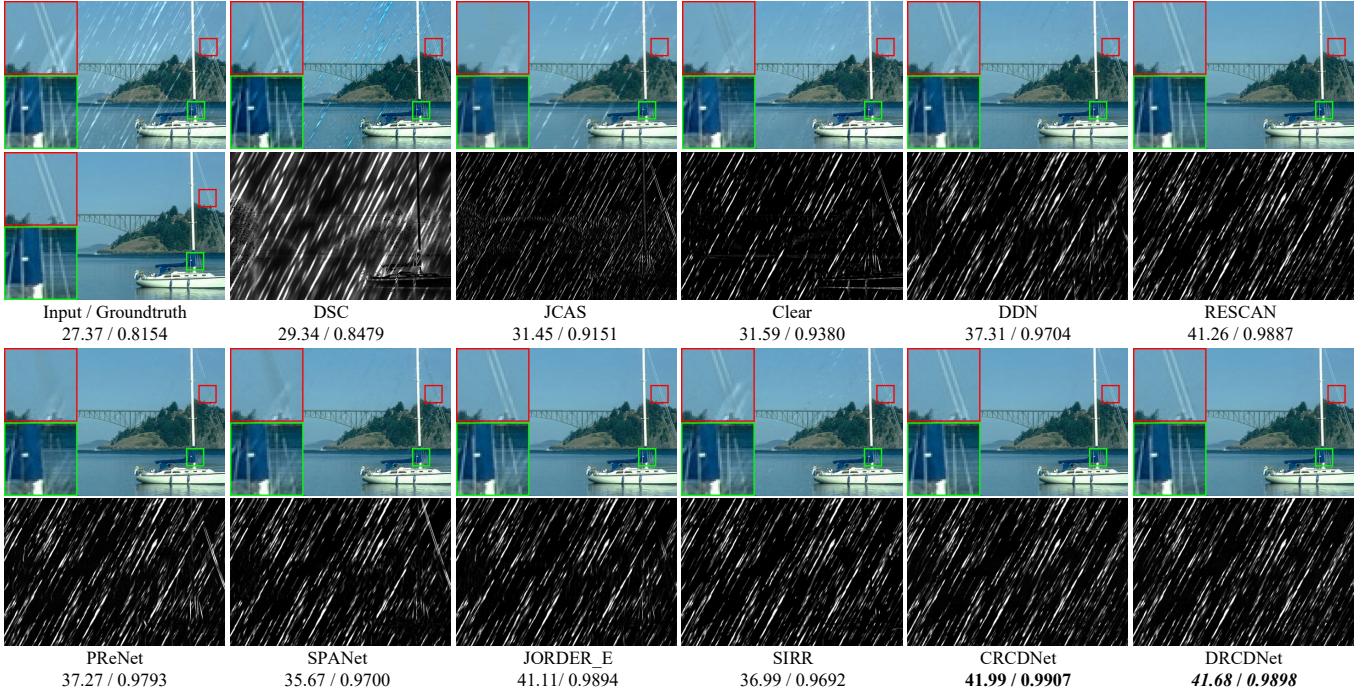


Fig. 10. (Training-test domain match) 1st column: input rainy image from Rain100L (upper) and groundtruth (lower). 2nd-12th columns: derained results (upper) with two demarcated areas zoomed in 4 times for easy observation, and extracted rain layers (lower) by 11 competing methods. PSNR/SSIM is listed below every derained result is for easy comparison. Bold and bold italic indicate top 1st and 2nd best results, respectively.

TABLE III

TRAINING-TEST DOMAIN MATCH CASE: AVERAGE PSNR AND SSIM COMPARISONS ON THREE DATASETS. BOLD AND BOLD ITALIC INDICATE TOP 1ST AND 2ND BEST RESULTS, RESPECTIVELY.

Datasets	Rain100L		Rain100H		Rain1400	
Metrics	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Input	26.90	0.8384	13.56	0.3709	25.24	0.8097
DSC [6]	27.34	0.8494	13.77	0.3199	27.88	0.8394
JCAS [7]	28.54	0.8524	14.62	0.4510	26.20	0.8471
Clear [12]	30.24	0.9344	15.33	0.7421	26.21	0.8951
DDN [13]	32.38	0.9258	22.85	0.7250	28.45	0.8888
RESCAN [17]	38.52	0.9812	29.62	0.8720	32.03	0.9314
PReNet [18]	37.54	0.9795	30.08	0.9050	32.09	0.9418
SPANet [24]	35.33	0.9694	25.11	0.8332	29.85	0.9148
JORDER_E [19]	37.89	0.9803	30.21	0.8957	32.00	0.9347
SIRR [64]	32.37	0.9258	22.47	0.7164	28.44	0.8893
CRCDNet	40.00	0.9860	31.28	0.9093	33.04	0.9472
DRCDNet	39.66	0.9852	30.50	0.8974	33.03	0.9466

D. Training-test Domain Mismatch Experiments

Here we evaluate the CRCDNet and DRCDNet in the case that rain types are inconsistent between training and testing.

Performance Comparison on Dense10 and Sparse10. We first adopt Dense10 and Sparse10 to evaluate the generalization capability of all DL competing methods trained on Rain100H. Fig. 13 shows the derained results on the input rainy image from Dense10 which has quite different rain types from Rain100H. In such an obvious domain mismatch testing case, almost all the approaches fail to remove mass rain streaks. However, with the flexible rain kernel prediction mechanism, our DRCDNet gets better visual effect than CRCDNet.

From the quantitative results in Table IV, we can find that CRCDNet is competing and DRCDNet obtains higher PSNR and SSIM. This tells us that the proper embedding of prior constraints are helpful to alleviate the over-fitting issue and the dynamic inference model would make the space for rain layer estimation tighter and then help further imporve the generalization performance.

Performance Comparison on SPA-Data. This real dataset is composed of complicated rain patterns, diverse shooting

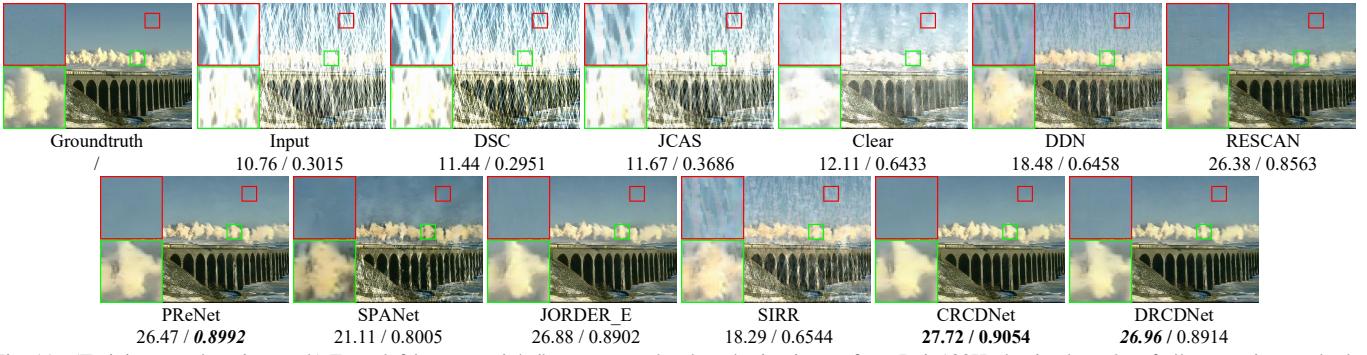


Fig. 11. (Training-test domain match) From left/upper to right/lower: groundtruth and rainy image from Rain100H, derained results of all competing methods.

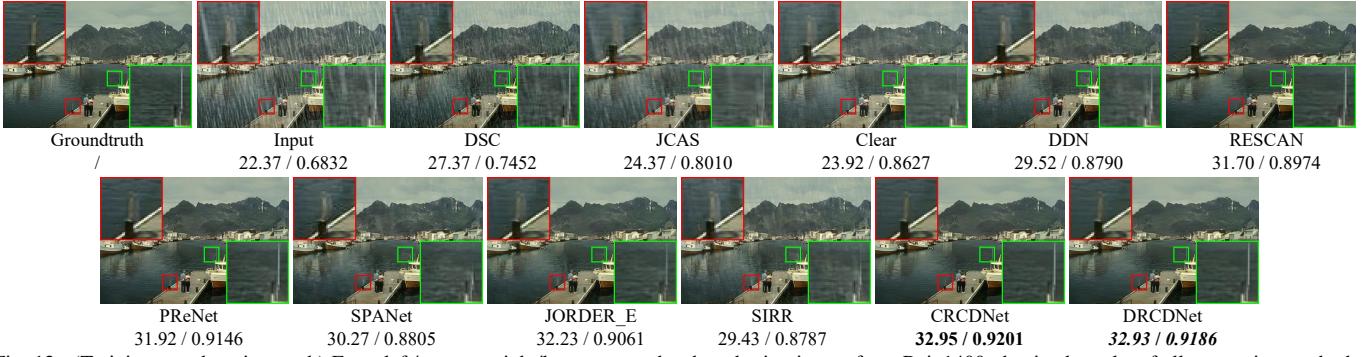


Fig. 12. (Training-test domain match) From left/upper to right/lower: groundtruth and rainy image from Rain1400, derained results of all competing methods.

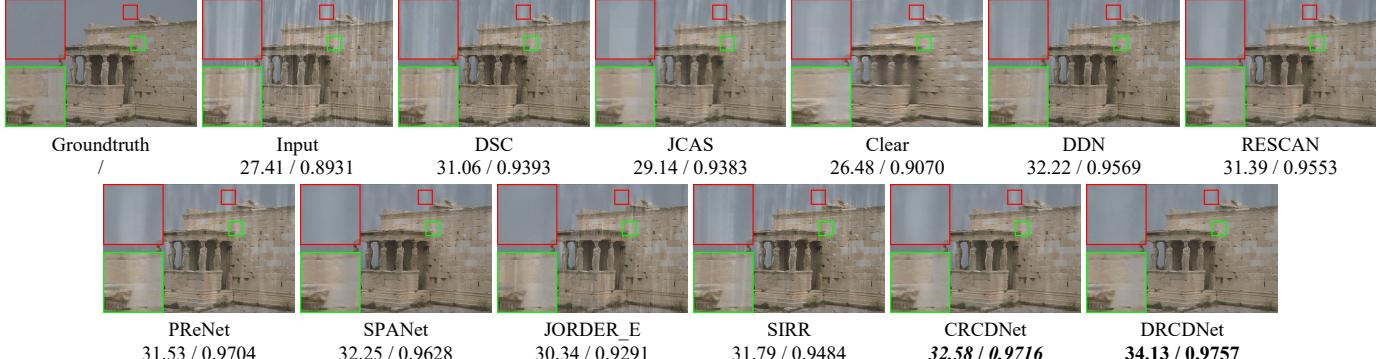


Fig. 13. (Training-test domain mismatch case) From left/upper to right/lower: groundtruth and input rainy image from Dense10, derained results of traditional DSC and JCAS, generalized results of all deep competing methods trained on Rain100H.

TABLE IV

TRAINING-TEST DOMAIN MISMATCH CASE: AVERAGE PSNR AND SSIM COMPARISONS ON DENSE10 AND SPARSE10. FOR DL BASED METHODS, THE MODELS ARE TRAINED ON RAIN100H. BOLD AND BOLD ITALIC INDICATE TOP 1ST AND 2ND BEST RESULTS, RESPECTIVELY.

Datasets	Metrics	Input	DSC	JCAS	Clear	DDN	RESCAN	PReNet	SPANet	JORDER_E	SIRR	CRCDNet	DRCDNet
Dense10	PSNR↑	19.17	20.85	19.93	19.10	21.38	21.81	21.91	22.11	21.66	21.23	22.07	22.47
	SSIM↑	0.8495	0.8811	0.8694	0.8600	0.8965	0.9073	0.9236	0.9170	0.9093	0.8925	0.9241	0.9255
Sparse10	PSNR↑	25.42	26.37	26.38	24.45	27.83	28.73	29.02	27.55	27.66	27.48	29.07	29.28
	SSIM↑	0.8956	0.8989	0.9043	0.8785	0.9249	0.9337	0.9412	0.9301	0.9257	0.9181	0.9422	0.9431

scenes, and rich background details. All these factors bring great challenges to accurate rain layer extraction and then the improvement of generalization performance on the dataset. Fig. 14 displays the reconstructed images where deep methods are trained on Rain100L. Clearly, the proposed DRCDNet performs better on both rain removal and detail preservation.

Table V provides the quantitative comparisons under different testing scenarios. As for the generalization case from Rain100L to SPA-Data, although DRCDNet achieves the higher PSNR and SSIM than CRCDNet, due to the simplicity of rain types in Rain100L and the complexity of rainy samples in SPA-Data, the generalization performance of DR-

CDNet is not prominent. However, by utilizing Rain100L and Rain1400 with 14 rain types as training data, the generalization performance of DRCDNet is largely improved.

Performance Comparison on Internet-Data. Fig. 15 shows the derained results on a typical rainy sample from the real Internet-Data where deep derainers are trained on both Rain100L and Rain100H. As seen, traditional DSC and JCAS, and some DL ones leave obvious rains in the restored backgrounds, and most of algorithms seriously blur the image details. Comparatively, our DRCDNet achieves the lowest BRISQUE and NIQE. We also provide quantitative compar-

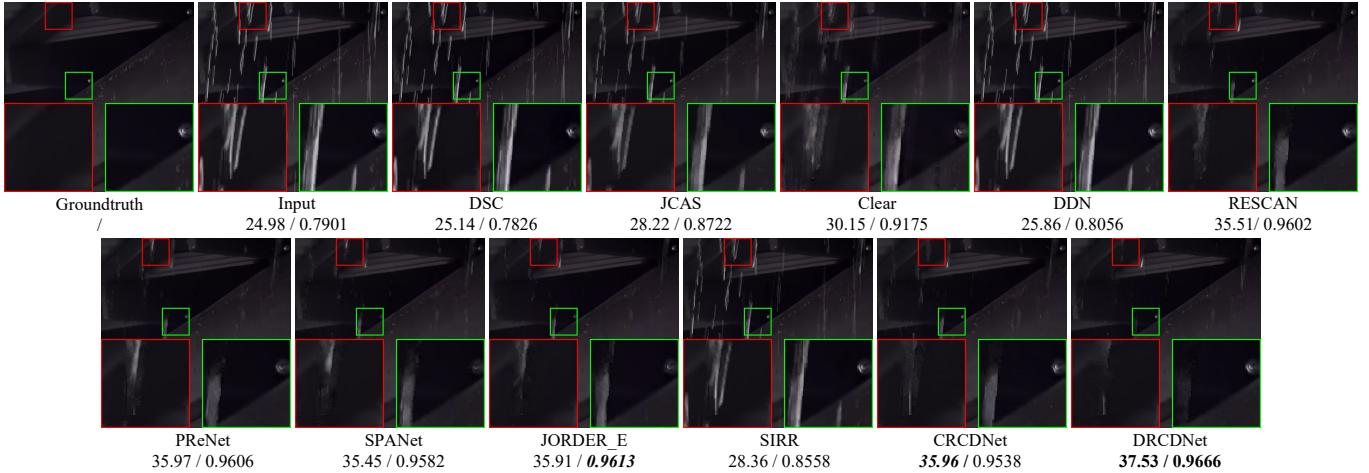


Fig. 14. (Training-test domain mismatch case) From left/upper to right/lower: groundtruth and input rainy image from the testing samples of SPA-Data, derained results of traditional DSC and JCAS, generalized results of all DL based competing methods trained on Rain100L.



Fig. 15. Training-test domain mismatch case: From left/upper to right/lower: input rainy image from Internet-Data, derained results of traditional DSC and JCAS, generalized results of all DL based competing methods trained on the training data of both Rain100L and Rain100H, each with two demarcated areas zoomed in 4 times for easy observation. NIQE/BRISQUE results are included for easy reference.

TABLE V
TRAINING-TEST DOMAIN MISMATCH CASE: AVERAGE PSNR AND SSIM COMPARISONS ON THE TESTING DATA OF SPA-DATA.

	Input	DSC	JCAS	Clear	DDN	RESCAN	PReNet	SPANet	JORDER_E	SIRR	CRCDNet	DRCDNet
Data (training/testing): Rain100L/SPA-Data; Difficulty: high.												
PSNR↑	34.15	34.83	34.95	32.66	34.66	34.70	34.91	35.13	35.04	34.66	34.88	35.23
SSIM↑	0.9269	0.9410	0.9451	0.9420	0.9346	0.9376	0.9407	0.9443	0.9405	0.9350	0.9377	0.9407
Data (training/testing): Rain100L+Rain1400/SPA-Data; Difficulty: high.												
PSNR↑	34.15	34.83	34.95	31.48	34.67	33.98	34.65	32.25	33.98	34.51	35.26	35.53
SSIM↑	0.9269	0.9410	0.9451	0.9357	0.9410	0.9432	0.9411	0.9393	0.9413	0.9336	0.9455	0.9512

TABLE VI
TRAINING-TEST DOMAIN MISMATCH CASE: AVERAGE BRISQUE AND NIQE COMPARISONS ON INTERNET-DATA.

	Input	DSC	JCAS	Clear	DDN	RESCAN	PReNet	SPANet	JORDER_E	SIRR	CRCDNet	DRCDNet
Data (training/testing): Rain100H/Internet-Data; Difficulty: high.												
BRISQUE↓	28.52	25.52	38.03	31.76	27.23	28.30	26.81	27.42	27.64	27.78	28.32	26.05
NIQE↓	5.1039	5.0539	5.3634	5.2781	4.8501	4.6730	4.6765	4.9052	4.6869	4.8276	4.7677	4.6606
Data (training/testing): Rain100L+Rain100H/Internet-Data; Difficulty: high.												
BRISQUE↓	28.52	25.52	38.03	32.39	27.05	25.80	25.01	26.74	25.48	26.44	26.00	23.35
NIQE↓	5.1039	5.0539	5.3634	5.3539	4.8806	4.4377	4.4630	4.8273	4.5247	4.8204	4.6082	4.4207

TABLE VII

COMPARISON OF NETWORK PARAMETERS AND TEST RUNNING TIME FOR THE INPUT IMAGE WITH SIZE 515 × 512 ON GPU.

Methods	Clear	DDN	RESCAN	PReNet	SPANet
Parameter #	754,691	57,369	149,823	168,963	283,716
Time (Seconds)	0.50	0.61	0.61	0.21	0.42
Methods	JORDER_E	SIRR	CRCDNet	DRCDNet	/
Parameter #	4,169,024	58,578	2,858,546	2,251,406	/
Time (Seconds)	3.12	3.65	0.84	0.71	/

son under different testing settings, as listed in Table VI.¹²

¹²More experiments are provided in supplemental file.

E. Network Parameters and Inference Time

Table VII presents the comparisons including network parameters and average inference time on a NVIDIA GeForce GTX1080Ti GPU. This shows that the proposed CRCDNet and DRCDNet are comparable to other competing methods.

IX. CONCLUSION

In this paper, we have proposed a novel interpretable network architecture, called RCDNet, specifically for the single image rain removal task. As compared with most of current deep derainers, the peculiarity is that we explicitly

embed the intrinsic rain convolutional dictionary (RCD) prior model of rain streaks into deep networks. Besides, each module in RCDNet has its own specific physical meanings, and is correspondent to the implementation operators of the algorithm designed for solving the RCD model. This makes the network have easily visualized interpretation for all its module elements and thus facilitates its easy analysis for what happens in the network. Furthermore, considering that the rain patterns of training data are inconsistent with testing data in most real scenarios, we have carefully designed a dynamic rain kernel inference mechanism and correspondingly built an interpretable DRCDNet, which can dynamically infer the corresponding rain kernels complying with diverse rain types of testing rainy images. This helps shrink the space for estimating rain layer and makes the network capable of being finely generalized to testing data even with the rain kernels different from training data. All these superiorities have been comprehensively substantiated by a series of experiments, including model verification, network visualization, rain kernel visualization, training/test domain match/mismatch evaluations. Besides, the extracted elements through the end-to-end learning by the network, like the diverse rain kernels, are also potentially useful for the related tasks on rainy images.

REFERENCES

- [1] M. S. Shehata, J. Cai, W. M. Badawy, T. W. Burr, M. S. Pervez, R. J. Johannesson, and A. Radmanesh, “Video-based automatic incident detection for smart roads: The outdoor environmental challenges regarding false alarms,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 2, pp. 349–360, 2008.
- [2] S. Li, I. B. Araujo, W. Ren, Z. Wang, E. K. Tokuda, R. H. Junior, R. Cesario-Junior, J. Zhang, X. Guo, and X. Cao, “Single image deraining: A comprehensive benchmark analysis,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3838–3847.
- [3] H. Wang, Y. Wu, M. Li, Q. Zhao, and D. Meng, “A survey on rain removal from video and single image,” *arXiv:1909.08326*, 2019.
- [4] W. Yang, R. T. Tan, S. Wang, Y. Fang, and J. Liu, “Single image deraining: From model-based to data-driven and beyond,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- [5] L. W. Kang, C. W. Lin, and Y. H. Fu, “Automatic single-image-based rain streaks removal via image decomposition,” *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1742–1755, 2012.
- [6] L. Yu, X. Yong, and J. Hui, “Removing rain from a single image via discriminative sparse coding,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3397–3405.
- [7] S. Gu, D. Meng, W. Zuo, and Z. Lei, “Joint convolutional analysis and synthesis sparse representation for single image layer separation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1708–1716.
- [8] L. Zhu, C. W. Fu, D. Lischinski, and P. A. Heng, “Joint bi-layer optimization for single-image rain streak removal,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2526–2534.
- [9] Z. He and V. M. Patel, “Convolutional sparse and low-rank coding-based rain streak removal,” in *IEEE Winter Conference on Applications of Computer Vision*, 2017, pp. 1259–1267.
- [10] F. Huang and A. Anandkumar, “Convolutional dictionary learning through tensor factorization,” *Computer Science*, pp. 1–30, 2015.
- [11] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6644–6653.
- [12] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, “Clearing the skies: A deep network architecture for single-image rain removal,” *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2944–2956, 2017.
- [13] X. Fu, J. Huang, D. Zeng, H. Yue, X. Ding, and J. Paisley, “Removing rain from single images via a deep detail network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3855–3863.
- [14] H. Zhang and V. M. Patel, “Density-aware single image de-raining using a multi-stream dense network,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 695–704.
- [15] H. Zhang, V. Sindagi, and V. M. Patel, “Image de-raining using a conditional generative adversarial network,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.
- [16] Y. Wei, Z. Zhang, J. Fan, Y. Wang, S. Yan, and M. Wang, “Derainy-clegan: An attention-guided unsupervised benchmark for single image deraining and rainmaking,” *arXiv preprint arXiv:1912.07015*, 2019.
- [17] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha, “Recurrent squeeze-and-excitation context aggregation net for single image deraining,” in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 254–269.
- [18] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, “Progressive image deraining networks: a better and simpler baseline,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3937–3946.
- [19] W. Yang, R. T. Tan, J. Feng, J. Liu, S. Yan, and Z. Guo, “Joint rain detection and removal from a single image with contextualized deep networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2019.
- [20] X. Fu, B. Liang, Y. Huang, X. Ding, and J. Paisley, “Lightweight pyramid networks for image deraining,” *IEEE transactions on neural networks and learning systems*, 2019.
- [21] Y. Zheng, X. Yu, M. Liu, and S. Zhang, “Residual multiscale based single image deraining,” in *Conference on BMVC*, 2019.
- [22] R. Yasara and V. M. Patel, “Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image de-raining,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8405–8414.
- [23] K. Jiang, Z. Wang, P. Yi, C. Chen, B. Huang, Y. Luo, J. Ma, and J. Jiang, “Multi-scale progressive fusion network for single image deraining,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8346–8355.
- [24] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, and R. W. Lau, “Spatial attentive single-image deraining with a high quality real rain dataset,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12270–12279.
- [25] G. Li, H. Xiang, Z. Wei, H. Chang, and L. Liang, “Non-locally enhanced encoder-decoder network for single image de-raining,” in *2018 ACM Multimedia Conference*, 2018.
- [26] G. Wang, C. Sun, and A. Sowmya, “Erl-net: Entangled representation learning for single image de-raining,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5644–5652.
- [27] X. Hu, C.-W. Fu, L. Zhu, and P.-A. Heng, “Depth-attentional features for single-image rain removal,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8022–8031.
- [28] J. Pan, S. Liu, D. Sun, J. Zhang, and M.-H. Yang, “Learning dual convolutional neural networks for low-level vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3070–3079.
- [29] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14560–14569.
- [30] H. Wang, Q. Xie, Q. Zhao, and D. Meng, “A model-driven deep neural network for single image rain removal,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3103–3112.
- [31] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.
- [32] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” 2017.
- [33] K. Garg and S. K. Nayar, “Detection and removal of rain from videos,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2004, pp. I–I.
- [34] X. Zhang, H. Li, Y. Qi, W. K. Leow, and T. K. Ng, “Rain removal in video by combining temporal and chromatic properties,” in *IEEE International Conference on Multimedia and Expo*, 2006, pp. 461–464.

- [35] W.-J. Park and K.-H. Lee, "Rain removal using kalman filter in video," in *International Conference on Smart Manufacturing Application*, 2008, pp. 494–497.
- [36] J. Bossu, N. Hautière, and J.-P. Tarel, "Rain or snow detection in image sequences through use of a histogram of orientation of streaks," *International journal of computer vision*, vol. 93, no. 3, pp. 348–367, 2011.
- [37] P. C. Barnum, S. Narasimhan, and T. Kanade, "Analysis of rain and snow in frequency space," *International journal of computer vision*, vol. 86, no. 2-3, p. 256, 2010.
- [38] K. Jin-Hwan, S. Jae-Young, and K. Chang-Su, "Video deraining and desnowing using temporal correlation and low-rank matrix completion," *IEEE Transactions on Image Processing*, vol. 24, no. 9, pp. 2658–2670, 2015.
- [39] W. Wei, L. Yi, Q. Xie, Q. Zhao, D. Meng, and Z. Xu, "Should we encode rain streaks in video as deterministic or stochastic?" in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2516–2525.
- [40] 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 *Proceedings of the ieee conference on computer vision and pattern recognition*, 2017, pp. 4057–4066.
- [41] ——, "Fastderain: A novel video rain streak removal method using directional gradient priors," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1–1, 2018.
- [42] Y. L. Chen and C. T. Hsu, "A generalized low-rank appearance model for spatio-temporally correlated rain streaks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1968–1975.
- [43] W. Ren, J. Tian, H. Zhi, A. Chan, and Y. Tang, "Video desnowing and deraining based on matrix decomposition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4210–4219.
- [44] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015.
- [45] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [46] Z. Yue, H. Yong, Q. Zhao, D. Meng, and L. Zhang, "Variational denoising network: Toward blind noise modeling and removal," in *Advances in Neural Information Processing Systems*, 2019, pp. 1688–1699.
- [47] H. Wu, J. Liu, Y. Xie, Y. Qu, and L. Ma, "Knowledge transfer dehazing network for nonhomogeneous dehazing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 478–479.
- [48] C. Jie, C. H. Tan, J. Hou, L. P. Chau, and L. He, "Robust video content alignment and compensation for rain removal in a cnn framework," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6286–6295.
- [49] J. Liu, W. Yang, S. Yang, and Z. Guo, "Erase or fill? deep joint recurrent rain removal and reconstruction in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3233–3242.
- [50] ——, "D3r-net: Dynamic routing residue recurrent network for video rain removal," *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 699–712, 2018.
- [51] W. Yang, J. Liu, and J. Feng, "Frame-consistent recurrent video deraining with dual-level flow," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [52] W. Yang, R. T. Tan, S. Wang, and J. Liu, "Self-learning video rain streak removal: When cyclic consistency meets temporal correspondence," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1720–1729.
- [53] X. Jing, Z. Wei, L. Peng, and X. Tang, "Removing rain and snow in a single image using guided filter," in *IEEE International Conference on Computer Science and Automation Engineering*, vol. 2, 2012, pp. 304–307.
- [54] X. Zheng, Y. Liao, W. Guo, X. Fu, and X. Ding, "Single-image-based rain and snow removal using multi-guided filter," in *International Conference on Neural Information Processing*, 2013, pp. 258–265.
- [55] X. Ding, L. Chen, X. Zheng, H. Yue, and D. Zeng, "Single image rain and snow removal via guided I0 smoothing filter," *Multimedia Tools and Applications*, vol. 75, no. 5, pp. 2697–2712, 2016.
- [56] J. H. Kim, C. Lee, J. Y. Sim, and C. S. Kim, "Single-image deraining using an adaptive nonlocal means filter," in *IEEE International Conference on Image Processing*, 2014, pp. 914–917.
- [57] Y. Li, "Rain streak removal using layer priors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2736–2744.
- [58] Y. Wang, S. Liu, C. Chen, and B. Zeng, "A hierarchical approach for rain or snow removing in a single color image," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3936–3950, 2017.
- [59] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Deep joint rain detection and removal from a single image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1357–1366.
- [60] Y. Yang and H. Lu, "Single image deraining via recurrent hierarchy enhancement network," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1814–1822.
- [61] H. Wang, Y. Wu, Q. Xie, Q. Zhao, Y. Liang, and D. Meng, "Structural residual learning for single image rain removal," *arXiv preprint arXiv:2005.09228*, 2020.
- [62] R. Li, L.-F. Cheong, and R. T. Tan, "Heavy rain image restoration: Integrating physics model and conditional adversarial learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1633–1642.
- [63] S. S. Halder, J.-F. Lalonde, and R. d. Charette, "Physics-based rendering for improving robustness to rain," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 10203–10212.
- [64] W. Wei, D. Meng, Q. Zhao, Z. Xu, and Y. Wu, "Semi-supervised transfer learning for image rain removal," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3877–3886.
- [65] R. Yasarla, V. A. Sindagi, and V. M. Patel, "Syn2real transfer learning for image deraining using gaussian processes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2726–2736.
- [66] X. Jin, Z. Chen, J. Lin, Z. Chen, and W. Zhou, "Unsupervised single image deraining with self-supervised constraints," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 2761–2765.
- [67] P. Mu, J. Chen, R. Liu, X. Fan, and Z. Luo, "Learning bilevel layer priors for single image rain streaks removal," *IEEE Signal Processing Letters*, vol. 26, no. 2, pp. 307–311, 2019.
- [68] R. Liu, Z. Jiang, X. Fan, and Z. Luo, "Knowledge-driven deep unrolling for robust image layer separation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 5, pp. 1653–1666, 2019.
- [69] B. Wohlberg, "Efficient convolutional sparse coding," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014.
- [70] D. L. Donoho, "De-noising by soft-thresholding," *IEEE transactions on information theory*, vol. 41, no. 3, pp. 613–627, 1995.
- [71] J. Zhang, J. Pan, W.-S. Lai, R. W. Lau, and M.-H. Yang, "Learning fully convolutional networks for iterative non-blind deconvolution," 2017.
- [72] Y. Yang, J. Sun, H. Li, and Z. Xu, "Admm-net: A deep learning approach for compressive sensing mri," *arXiv preprint arXiv:1705.06869*, 2017.
- [73] K. Zhang, L. V. Gool, and R. Timofte, "Deep unfolding network for image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3217–3226.
- [74] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [75] D. Yang and J. Sun, "Proximal dehaze-net: A prior learning-based deep network for single image dehazing," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 702–717.
- [76] Q. Xie, M. Zhou, Q. Zhao, D. Meng, W. Zuo, and Z. Xu, "Multispectral and hyperspectral image fusion by ms/hs fusion net," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1585–1594.
- [77] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2014.
- [78] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of psnr in image/video quality assessment," *Electronics Letters*, vol. 44, no. 13, pp. 800–801, 2008.
- [79] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [80] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a \pm completely blind \pm image quality analyzer," *IEEE Signal processing letters*, vol. 20, no. 3, pp. 209–212, 2012.
- [81] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on image processing*, vol. 21, no. 12, pp. 4695–4708, 2012.