# A Model-driven Deep Neural Network for Single Image Rain Removal

Hong Wang[1,*] Qi Xie[1,*] Qian Zhao[1], Deyu Meng[2,1,†]
[1]Xi'an Jiaotong University; [2]Macau University of Science and Technology
{hongwang01,xq.liwu}@stu.xjtu.edu.cn timmy.zhaoqian@gmail.com dymeng@mail.xjtu.edu.cn

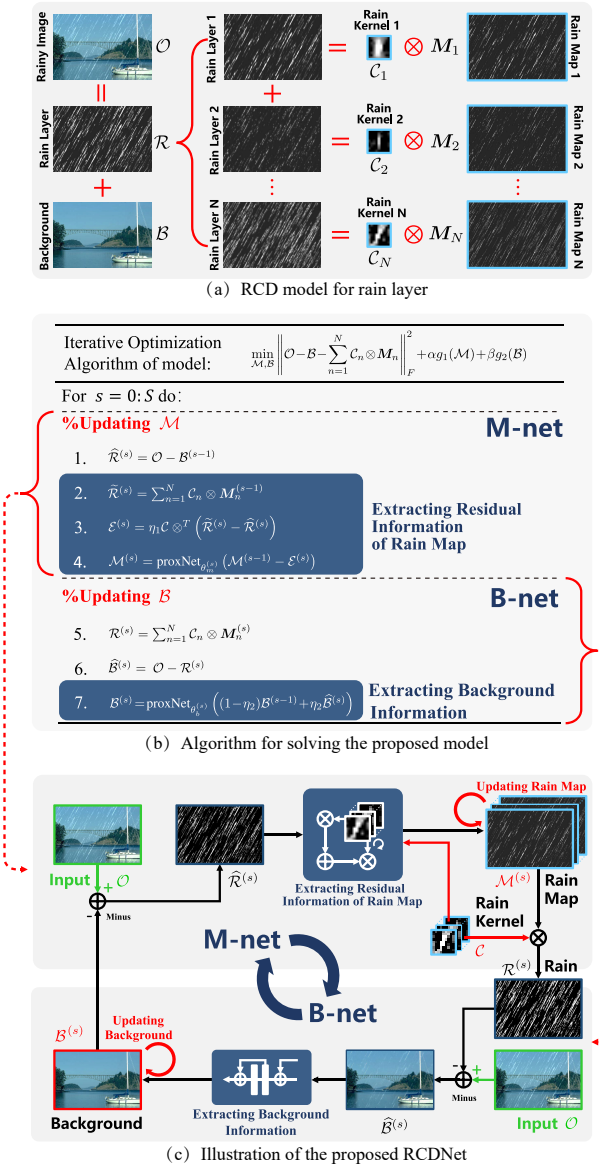
## Abstract

*Deep learning (DL) methods have achieved state-of-the-art performance in the task of single image rain removal. Most of current DL architectures, however, are still lack of sufficient interpretability and not fully integrated with physical structures inside general rain streaks. To this issue, in this paper, we propose a model-driven deep neural network for the task, with fully interpretable network structures. Specifically, based on the convolutional dictionary learning mechanism for representing rain, we propose a novel single image deraining model and utilize the proximal gradient descent technique to design an iterative algorithm only containing simple operators for solving the model. Such a simple implementation scheme facilitates us to unfold it into a new deep network architecture, called rain convolutional dictionary network (RCDNet), with almost every network module one-to-one corresponding to each operation involved in the algorithm. By end-to-end training the proposed RCDNet, all the 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 its better deraining performance, especially in real scenarios. Comprehensive experiments substantiate the superiority of the proposed network, especially its well generality to diverse testing scenarios and good interpretability for all its modules, as compared with state-of-the-arts both visually and quantitatively.*

## 1. Introduction

Images taken under various rain conditions often suffer from unfavorable visibility, and always severely affect the performance of outdoor computer vision tasks, such as objection tracking [5], video surveillance [37], and pedestrian detection [31]. Hence, removing rain streaks from rainy images is an important pre-processing task and has drawn much research attention in the recent years [39, 26].

In the past years, various methods have been proposed for single image rain removal task. Many researchers made focus on exploring physical properties of rain layer and




(a) RCD model for rain layer

(b) Algorithm for solving the proposed model

(c) Illustration of the proposed RCDNet

Figure 1. (a) Rain convolutional dictionary (RCD) model for rain layer. (b) The formulated optimization model and the corresponding iterative solution algorithm. (c) Visual illustration of the proposed RCDNet one-to-one corresponding to the algorithm (b).

†Corresponding author
*Equal contribution

background layer, and introduced various prior structures to regularize and separate them. Along this research line, the representative methods include layer priors with Gaussian mixture model (GMM) [28], discriminative sparse coding (DSC) [51], and joint convolutional analysis and synthesis sparse representation (JCAS) [13]. Especially, inspired by the fact that rain streaks repeatedly appear at different locations over a rainy image with similar local patterns like shape, thickness, and direction, very recently researchers represented this configuration of rain layer by the convolutional dictionary learning model [15, 16]. Such a representation finely delivers this prior knowledge by imposing rain kernels (conveying repetitive local patterns) on sparse rain maps, as intuitively depicted in Fig. 1 (a). These methods thus achieved state-of-the-art (SOTA) performance when the background can also be well represented, e.g., by low-rank prior in surveillance video sequences [25].

Albeit effective in certain applications, the rationality of these techniques depends on the subjective prior assumptions imposed on the unknown background and rain layers to be recovered. In real scenarios, however, such learning regimes could not always adapt to different rainy images with complex, diverse, and variant structures collected from different resources. Besides, these methods generally need time-consuming iterative computations, often with efficiency issue in real applications.

Driven by the significant success of deep learning (DL) in low level vision, recent years have also witnessed the rapid progress of deep convolutional neural networks (CNN) for single image rain removal [8, 52, 53, 40]. The current DL-based derainers mainly focus on designing network modules, and then train network parameters based on abundant rainy/clean image pairs to extract the background layer. Typical deraining network structures include deep detail network (DDN) [9], recurrent squeeze-and-excitation context aggregation module (RESCAN) [27], progressive image deraining network (PReNet) [35], spatial attentive unit (SPANet) [41], and many others.

These DL strategies, however, also possess evident deficiencies. The most significant one is their weak interpretability. Network structures are always complicated and diverse, making it difficult to analyze the role of different modules and understand the underlying insights of their mechanism. Besides, most of them treat CNN as an encapsulated end-to-end mapping module without deepening into the rationality, and neglect the intrinsic prior knowledge of rain streaks such as sparsity and nonlocal similarity. This makes this methodology easily trapped into the overfitting-to-training-sample issue.

To alleviate the aforementioned issues, this paper designs an interpretable deep network, which sufficiently considers the characteristics of rain streaks and attempts to combine the advantages of the conventional model-driven prior-based and current data-driven DL-based methodologies. Specifically, our contributions are mainly three-fold:

Firstly, we propose a concise rain convolutional dictionary (RCD) model for single image by exploiting the intrinsic convolutional dictionary learning mechanism to encode rain shapes, and specifically adopt the proximal gradient technique [2] to design an optimization algorithm for solving it. Different from traditional solvers for the RCD model containing complex operations (e.g., Fourier transformation), the algorithm only contains simple computations (see Fig. 1 (b)) easy to be implemented by general network modules. This facilitates our algorithm capable of being easily unfolded into a deep network architecture.

Secondly, by unfolding the algorithm, we design a new deep network architecture for image deraining, called RCDNet. The specificity of this network lies on its exact step-by-step corresponding relationship between its modules and the algorithm operators, and thus successively possesses the interpretability of all its modules as that of all steps in the algorithm. Specifically, as shown in Fig. 1 (b) and (c), each iteration of the algorithm contains two sub-steps, respectively updating the rain map (convoluted by the learned rain kernels) and background layer, and each stage of the RCDNet also contains two sub-networks (M-net and B-net). Each output of the intermediate layer in the network is thus with clear interpretation, which greatly facilitates a deeper analysis on what happens inside the network during training, and a comprehensive understanding why the network works or not (as the analysis presented in Sec. 5.2).

Thirdly, comprehensive experimental results substantiate the superiority of the RCDNet beyond SOTA conventional prior-based and current DL-based methods both quantitatively and visually. Especially, attributed to its well interpretability, not only the underlying rationality and insights of the network can be intuitively understood through visualizing the amelioration process (like the gradually rectified background and rain maps) over all network layers by general users, but also the network can yield generally useful rain kernels for expressing 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-world rainy images.

The paper is organized as follows. Sec. 2 reviews the related rain removal work. Sec. 3 presents the RCD model for rain removal as well as the algorithm designed for solving it. Then Sec. 4 introduces the unfolding deep network for the algorithm. The experimental results are demonstrated in Section 5 and the paper is finally concluded.

## 2. Related work

In this section, we give a brief review on the most related work on rain removal for images. Depending on the input data, the existing algorithms can be categorized into two

groups: video based and single image based ones.

### 2.1. Video deraining methods

Garg and Nayar [10] first tried to analyze the visual effects of raindrops on imaging systems, and utilized a space-time correlation model to capture the dynamics of raindrops and a physics-based motion blur model to illustrate the photometry of rain. For better visual quality, they further proposed to increase the exposure time or reduce the depth of field of a camera [12, 11]. Later, both temporal and chromatic properties of rain were considered and then background layer was extracted from rainy video by utilizing different strategies such as K-means clustering [55], Kalman filter [33], and GMM [3]. Besides, a spatio-temporal frequency based raindrop detection method was provided in [1].

In recent years, researchers introduced more intrinsic characteristics of rainy video to the task, e.g., similarity and repeatability of rain streaks [4], low-rankness among multi-frames [20], and sparsity and smoothness of rain streaks [18]. To handle heavy rain and dynamic scenes, a matrix decomposition based video deraining algorithm was presented in [36]. Afterwards, rain streaks were encoded as a patch based GMM to adapt a wider range of rain variations [45]. More characteristics of rain streaks in a rainy video were explored including repetitive local patterns and multi-scale configurations and they were described as multi-scale convolutional sparse coding model [25]. More recently, there are some DL-based methods proposed for this task. Chen *et al.* [19] presented a CNN architecture and utilized superpixel to handle torrential rain fall with opaque streak occlusions. To further improve visual quality, Liu *et al.* [30] designed a joint recurrent rain removal and reconstruction network that integrated rain degradation classification, rain removal, and background details reconstruction. To handle dynamic video contexts, they further developed a dynamic routing residue recurrent network [29]. Though these methods work well for videos, they cannot directly perform in single image cases due to the lack of temporal knowledge.

### 2.2. Single image deraining methods

Compared with video deraining task under a sequence of images, rain removal from a single image is much more challenging. The early attempts utilized the model-driven strategies by decomposing a single rainy image into low frequency part (LFP) and high frequency part (HFP) and then specifically extracted rain layer from the HFP based on various processing such as guided filter [6, 21] and nonlocal means filtering [23]. Later, researchers made more focus on exploring the prior knowledge of rain and rain-free layers of a rainy image, and designing proper regularizer to extract and separate them [22, 38, 51, 28, 42, 56]. E.g., [13] considered the specific sparsity characteristics of rain-free and rain parts and expressed them as the joint analysis and synthesis sparse representation models, respectively. [15] used a similar manner to deliver local repetitive patterns of rain streaks across the image as the RCD model. Albeit achieving good performance on certain scenarios, these prior-based methods rely on the subjective prior assumptions, while could not always generally work well for practical complicated and highly diverse rain shapes in real rainy images collected from different resources.

Recently, a number of DL-based single image rain streak removal methods were proposed through constructing diverse network modules [8, 9, 27, 52, 53]. To handle heavy rain, Yang *et al.* [49] developed a multi-stage joint rain detection and estimation network for single image (JORDER_E). Very recently, Ren *et al.* [35] designed a PReNet that repeatedly unfolded several Resblocks and a LSTM layer. Wang *et al.* [41] presented an attention unit based SPANet for removing rain in a local-to-global manner. Through using abundant rainy/clean image pairs to train the deep model, these methods achieve favorable visual quality and SOTA quantitative measures of derained results. Most of these methods, however, just utilize network modules assembled with some off-the-shelf components in current DL toolkits to directly learn background layer in an end-to-end way, and largely ignore the intrinsic prior structures inside the rain streaks. This makes them lack of evident interpretability in their network architectures and still have room for further performance enhancement.

At present, there is a new type of single image derainers that try to combine prior and DL methodologies. For example, Mu *et al.* [32] utilized CNN to implicitly learn prior knowledge for background and rain streaks, and formulated them into traditional bi-layer optimization iterations. Wei *et al.* [44] provided a semi-supervised rain removal method (SIRR) that described rain layer prior as a general GMM and jointly trained the backbone–DDN. Albeit obtaining initial success, they still use CNN architectures as their main modules to construct the network, which is thus still lack of sufficient interpretability.

## 3. RCD model for single image deraining

### 3.1. Model formulation

For a observed color rainy image denoted as $\mathcal{O} \in \mathbb{R}^{H\times W\times 3}$, where $H$ and $W$ are the height and width of the image, respectively, it can be rationally separated as:

$$\mathcal{O} = \mathcal{B} + \mathcal{R}, \tag{1}$$

where $\mathcal{B}$ and $\mathcal{R}$ represent the background and rain layers of the image, respectively. Then, the aim of most of DL-based deraining methods is to estimate the mapping function (expressed by a deep network) from $\mathcal{O}$ to $\mathcal{B}$ (or $\mathcal{R}$).

Instead of heuristically constructing a complex deep network architecture, we first consider the problem under the conventional prior-based methodology through exploiting the prior knowledge for representing rain streaks [13, 15, 25]. Specifically, as shown in Fig. 1 (a), by adopting the RCD mechanism, the rain layer can be modeled as:

$$\mathcal{R}^c = \sum_{n=1}^{N} \boldsymbol{C}_n^c \otimes \boldsymbol{M}_n, c = 1, 2, 3, \tag{2}$$

where $\mathcal{R}_c$ denotes the $c^{\text{th}}$ color channel of $\mathcal{R}$, and $\{\boldsymbol{C}_n^c\}_{n,c} \subset \mathbb{R}^{k\times k}$ is a set of rain kernels which describes the repetitive local patterns of rain streaks, and $\{\boldsymbol{M}_n\}_n \subset \mathbb{R}^{H\times W}$ represents the corresponding rain maps representing the locations where local patterns repeatedly appear. $N$ is the number of kernels and $\otimes$ is the 2-dimensional (2D) convolutional operation. For conciseness, we rewrite (2) as $\mathcal{R} = \sum_{n=1}^{N} \mathcal{C}_n \otimes \boldsymbol{M}_n$ throughout the paper, where $\mathcal{C}_n \in \mathbb{R}^{k\times k\times 3}$ is the tensor form of $\boldsymbol{C}_n^c$s and the convolution is performed between $\mathcal{C}_n$ and the matrix $\boldsymbol{M}_n$ one channel by one channel. Then, we can rewrite the model (1) as:

$$\mathcal{O} = \mathcal{B} + \sum_{n=1}^{N} \mathcal{C}_n \otimes \boldsymbol{M}_n. \tag{3}$$

It should be noted that the rain kernels actually can be viewed a set of convolutional dictionary [16] for representing repetitive and similar local patterns underlying rain streaks, and a small number of rain kernels can finely represent wide range of rain shapes[1]. They are common knowledge for representing different rain types across all rainy images, and thus could be learned from abundant training data by virtue of the strong learning capability of end-to-end training manner of deep learning (see more details in Sec. 4). Unlike rain kernels, the rain maps must vary with the input rainy image as the locations of rain streaks are totally random. Therefore, for predicting the clean image from a testing input rainy one, the key issue is to output $\boldsymbol{M}_n$s and $\mathcal{B}$ from $\mathcal{O}$ with the rain kernels $\mathcal{C}_n$s fixed, and the corresponding optimization problem is:

$$\min_{\mathcal{M},\mathcal{B}} \left\| \mathcal{O} - \mathcal{B} - \sum_{n=1}^{N} \mathcal{C}_n \otimes \boldsymbol{M}_n \right\|_F^2 + \alpha g_1(\mathcal{M}) + \beta g_2(\mathcal{B}), \tag{4}$$

where $\mathcal{M} \in \mathbb{R}^{H\times W\times N}$ is the tensor form of $\boldsymbol{M}_n$s. $\alpha$ and $\beta$ are trade-off parameters. $g_1(\cdot)$ and $g_2(\cdot)$ mean the regularizers to deliver the prior structures of $\boldsymbol{M}_n$ and $\mathcal{B}$, respectively.

### 3.2. Optimization algorithm

Since we want to build a possibly perfect step-by-step corresponding deep unfolding network architecture for solving the problem (4), it is critical to build an algorithm which contains only simple computations easy to be transformed to network modules. The traditional solvers for RCD-based model usually contain certain complicated operations, e.g., the Fourier transform and inverse Fourier transform [16, 46, 25], which are hard to accomplish such exact transformation from algorithm to network structure. We thus prefer to build a new algorithm for solving the problem through alternately updating $\mathcal{M}$ and $\mathcal{B}$ by proximal gradient method [2]. In this manner, only simple computations can be involved. The details are as follows:

**Updating** $\mathcal{M}$: The rain maps $\mathcal{M}$ can be updated by solving the quadratic approximation [2] of the problem (4) as:

$$\min_{\mathcal{M}} \frac{1}{2} \left\| \mathcal{M} - \left( \mathcal{M}^{(s-1)} - \eta_1 \nabla f\left( \mathcal{M}^{(s-1)} \right) \right) \right\|_F^2 + \alpha\eta_1 g_1\left(\mathcal{M}\right), \tag{5}$$

where $\mathcal{M}^{(s-1)}$ is the updating result of the last iteration, $\eta_1$ is the stepsize parameter, and $f\left(\mathcal{M}^{(s-1)}\right) = \left\| \mathcal{O} - \mathcal{B}^{(s-1)} - \sum_{n=1}^{N} \mathcal{C}_n \otimes \boldsymbol{M}_n^{(s-1)} \right\|_F^2$. Corresponding to general regularization terms [7], the solution of Eq. (5) is:

$$\mathcal{M}^{(s)} = \text{prox}_{\alpha\eta_1}\left( \mathcal{M}^{(s-1)} - \eta_1 \nabla f\left( \mathcal{M}^{(s-1)} \right) \right). \tag{6}$$

Moreover, by substituting

$$\nabla f\left(\mathcal{M}^{(s-1)}\right) = \mathcal{C} \otimes^T \left( \sum_{n=1}^{N} \mathcal{C}_n \otimes \boldsymbol{M}_n^{(s-1)} + \mathcal{B}^{(s-1)} - \mathcal{O} \right), \tag{7}$$

where $\mathcal{C} \in \mathbb{R}^{k\times k\times N\times 3}$ is a 4-D tensor stacked by $\mathcal{C}_n$s, and $\otimes^T$ denotes the transposed convolution[2], we can obtain the updating formula for $\mathcal{M}$ as[3]:

$$\mathcal{M}^{(s)} = \text{prox}_{\alpha\eta_1}\left( \mathcal{M}^{(s-1)} - \eta_1 \mathcal{C} \otimes^T \left( \sum_{n=1}^{N} \mathcal{C}_n \otimes \boldsymbol{M}_n^{(s-1)} + \mathcal{B}^{(s-1)} - \mathcal{O} \right) \right), \tag{8}$$

where $\text{prox}_{\alpha\eta_1}(\cdot)$ is the proximal operator dependent on the regularization term $g_1(\cdot)$ with respect to $\mathcal{M}$. Instead of choosing a fixed regularizer in the model, the form of the proximal operator can be automatically learned from training data. More details will be presented in the next section.

**Updating** $\mathcal{B}$: Similarly, the quadratic approximation of the problem (4) with respect to $\mathcal{B}$ is:

$$\min_{\mathcal{B}} \frac{1}{2} \left\| \mathcal{B} - \left( \mathcal{B}^{(s-1)} - \eta_2 \nabla h\left( \mathcal{B}^{(s-1)} \right) \right) \right\|_F^2 + \beta\eta_2 g_2(\mathcal{B}). \tag{9}$$

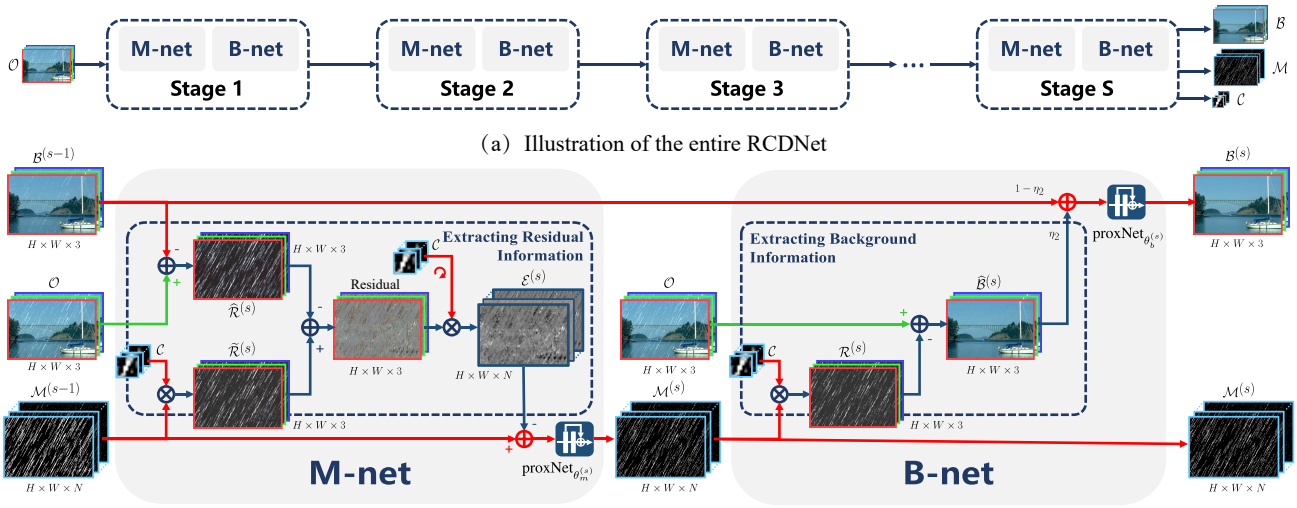
where $\nabla h\left(\mathcal{B}^{(s-1)}\right) = \sum_{n=1}^{N} \mathcal{C}_n \otimes \boldsymbol{M}_n^{(s)} + \mathcal{B}^{(s-1)} - \mathcal{O}$, and it is easy to deduce that the final updating rule for $\mathcal{B}$ is[3]:

$$\mathcal{B}^{(s)} = \text{prox}_{\beta\eta_2}\left( (1-\eta_2)\, \mathcal{B}^{(s-1)} + \eta_2 \left( \mathcal{O} - \sum_{n=1}^{N} \mathcal{C}_n \otimes \boldsymbol{M}_n^{(s)} \right) \right). \tag{10}$$

[1] We simply set $N = 32$ for all our experiments.

[2] For any tensor $\mathcal{A} \in \mathbb{R}^{H\times W\times 3}$, we can calculate the $n^{\text{th}}$ channel of $\mathcal{C} \otimes^T \mathcal{A}$ by $\sum_{c=1}^{3} \mathcal{C}_{\{:,:,n,c\}} \otimes^T \mathcal{A}_{\{:,:,c\}}$.

[3] It can be proved that, with small enough $\eta_1$ and $\eta_2$, Eq. (8) and Eq. (10) can both lead to the reduction of objective function (4) [2].




(a) Illustration of the entire RCDNet

(b) The design of a single stage

Figure 2. (a) The proposed network with $S$ stages. The network takes a rainy image $\mathcal{O}$ as input and outputs the learned rain kernel $\mathcal{C}$, rain map $\mathcal{M}$, and clean background image $\mathcal{B}$. (b) Illustration of the network architecture at the $s^{\text{th}}$ 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 zoomed in on screen.

where $\text{prox}_{\beta\eta_2}(\cdot)$ is the proximal operator correlated to the regularization term $g_2(\cdot)$ with respect to $\mathcal{B}$.

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

## 4. The rain convolutional dictionary network

Inspired by the recently raised deep unfolding techniques in various tasks such as deconvolution [54], compressed sensing [50], and dehazing [48], we build a network structure for single image rain removal task by unfolding each iterative steps of the aforementioned algorithm as the corresponding network module. We especially focus on making all network modules one-to-one corresponding to the algorithm implementation operators, for better interpretability.

As shown in Fig. 2 (a), the proposed network consists of $S$ stages, corresponding to $S$ iterations of the algorithm for solving (4). Each stage achieves the sequential updates of $\mathcal{M}$ and $\mathcal{B}$ by M-net and B-net. As displayed in Fig. 2 (b), exactly corresponding to each iteration of the algorithm, in each stage of the network, M-net takes the observed 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 B-net takes $\mathcal{O}$ and $\mathcal{M}^{(s)}$ as inputs, and outputs an updated $\mathcal{B}^{(s)}$.

### 4.1. Network design

The key issue of unrolling the algorithm here is how to represent the two proximal operators involved in (8) and (10) while other operations can be naturally performed with generally used operators in normal networks [34]. In this work, we simply choose a ResNet [14] to construct the two proximal operators as many other works did [47, 48]. Then, we can separately decompose the updating rules for $\mathcal{M}$ as (8) and $\mathcal{B}$ as (10) into sub-steps and achieve the following procedures for the $s^{\text{th}}$ stage of the RCDNet:

$$\text{M-net}: \begin{cases} \widehat{\mathcal{R}}^{(s)} = \mathcal{O} - \mathcal{B}^{(s-1)}, \\ \widetilde{\mathcal{R}}^{(s)} = \sum_{n=1}^{N} \mathcal{C}_n \otimes \boldsymbol{M}_n^{(s-1)}, \\ \mathcal{E}^{(s)} = \eta_1 \mathcal{C} \otimes^T \left(\widetilde{\mathcal{R}}^{(s)} - \widehat{\mathcal{R}}^{(s)}\right), \\ \mathcal{M}^{(s)} = \text{proxNet}_{\theta_m^{(s)}}\left(\mathcal{M}^{(s-1)} - \mathcal{E}^{(s)}\right), \end{cases} \tag{11}$$

$$\text{B-net}: \begin{cases} \mathcal{R}^{(s)} = \sum_{n=1}^{N} \mathcal{C}_n \otimes \boldsymbol{M}_n^{(s)}, \\ \widehat{\mathcal{B}}^{(s)} = \mathcal{O} - \mathcal{R}^{(s)}, \\ \mathcal{B}^{(s)} = \text{proxNet}_{\theta_b^{(s)}}\left((1-\eta_2)\mathcal{B}^{(s-1)} + \eta_2 \widehat{\mathcal{B}}^{(s)}\right), \end{cases} \tag{12}$$

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 $s^{\text{th}}$ stage, respectively.

We can then design the network architecture, as shown in Fig. 2, by transforming the operators in (11) and (12) step-by-step. All the parameters involved can be automatically learned from training data in an end-to-end manner, including $\{\theta_m^{(s)}, \theta_b^{(s)}\}_{s=1}^{S}$, rain kernels $\mathcal{C}$, $\eta_1$, and $\eta_2$.

It should be indicated that both of the two sub-networks are very interpretable. As shown in Fig. 2 (b), the M-net accomplishes the extraction of residual information $\mathcal{E}^{(s)}$ of rain maps. 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 (2) with the estimated $\mathcal{M}^{(s-1)}$. Then the M-net calculates the residual information between the two rain layers obtained in this two ways, and extracts the residual information $\mathcal{E}^{(s)}$ 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 this estimated $\widehat{\mathcal{B}}^{(s)}$ with the previously estimated $\mathcal{B}^{(s-1)}$ by

weighted parameters $\eta_2$ and $(1-\eta_2)$ to get the updated background $\mathcal{B}^{(s)}$. Here, we set $\mathcal{M}^{(0)}$ as 0 and initialize $\mathcal{B}^{(0)}$ by a convolutional operator on $\mathcal{O}$[4].

***Remark:*** From Fig. 2, the input tensor of $\text{proxNet}_{\theta_b^{(s)}}(\cdot)$ has the same size $H \times W \times 3$ as the to-be-estimated $\mathcal{B}$. Evidently, this is not beneficial for learning $\mathcal{B}$ since most of the previous updating information would be compressed due to few channels. To better keep and deliver image features, we simply expand the input tensor at the 3rd mode for more channels in experiments (see more in supplemental file).

### 4.2. Network training

**Training loss.** For simplicity, we adopt the mean square error (MSE) [21] for the learned background and rain layer at every stage as the training objective function:

$$L = \sum_{s=0}^{S} \lambda_s \left\| \mathcal{B}^{(s)} - \mathcal{B} \right\|_F^2 + \sum_{s=1}^{S} \gamma_s \left\| \mathcal{O} - \mathcal{B} - \mathcal{R}^{(s)} \right\|_F^2, \quad (13)$$

where $\mathcal{B}^{(s)}$ and $\mathcal{R}^{(s)}$ separately denote the derained result and extracted rain layer as expressed in (12) at the $s^{\text{th}}$ stage ($s = 0, 1, \cdots, S$). $\lambda_s$ and $\gamma_s$ are tradeoff parameters[5].

**Implement details.** We implement our network based on a NVIDIA GeForce GTX 1080Ti GPU. We adopt the Adam optimizer [24] with the batch size of 16 and the patch size of 64×64. The initial learning rate is $1\times10^{-3}$ and divided by 5 every 25 epochs. The total epoch is 100.

## 5. Experimental results

We first conduct ablation study and model visualization to verify the underlying mechanism of the proposed network, and then present experiments on synthesized benchmark datasets and real datasets for performance evaluation.

### 5.1. Ablation study

**Dataset and performance metrics.** In this section, we use Rain100L to perform all the ablation studies. The synthesized dataset consists of 200 rainy/clean image pairs for training and 100 pairs for testing [49]. Two performance metrics are employed, including peak-signal-to-noise ratio (PSNR) [17] and structure similarity (SSIM) [43]. Note that as the human visual system is sensitive to the Y channel of a color image in YCbCr space, we compute PSNR and SSIM based on this luminance channel.

Table 1 reports the effect of stage number $S$ on deraining performance of our network. Here, $S = 0$ means that the initialization $\mathcal{B}^{(0)}$ is directly regraded as the recovery result.

[4]More network design details are described in supplemental file.

[5]In all experiments, we simply set $\lambda_S = \gamma_S = 1$ to make the outputs at the final stage play a dominant role, and other parameters as 0.1 to help find the correct parameter in each stage. More parameter settings are discussed in supplementary material.

Table 1. Effect of stage number $S$ on the performance of RCDNet.

| Stage No. | $S$=0 | $S$=2 | $S$=5 | $S$=8 | $S$=11 | $S$=14 | $S$=17 | $S$=20 |
|---|---|---|---|---|---|---|---|---|
| PSNR | 35.93 | 38.46 | 39.35 | 39.60 | 39.81 | 39.90 | 40.00 | 39.91 |
| SSIM | 0.9689 | 0.9813 | 0.9842 | 0.9850 | 0.9855 | 0.9858 | 0.9860 | 0.9858 |

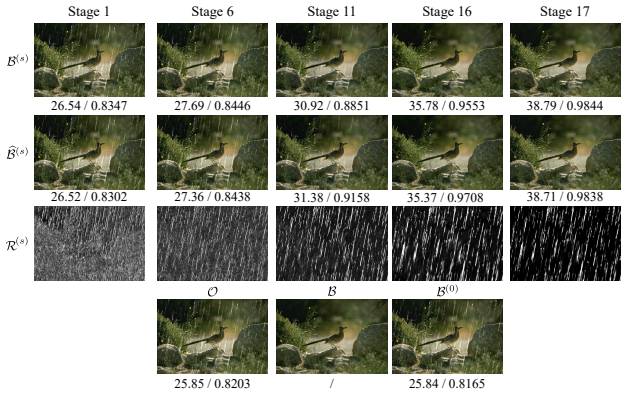



Figure 3. Visualization of the recovery background $\mathcal{B}^{(s)}$, $\widehat{\mathcal{B}}^{(s)}$ as expressed in Eq. (12), and the rain layer $\mathcal{R}^{(s)}$ at different stages. The stage number $S$ is 17. PSNR/SSIM for reference. The images are better to be zoomed in on screen.

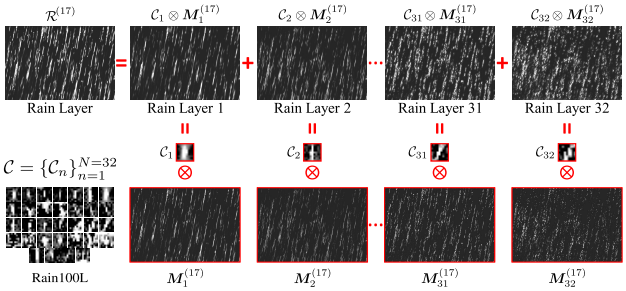



Figure 4. At the final stage $s = 17$, the extracted rain layer, rain kernels $\mathcal{C}_n$, and rain maps $\boldsymbol{M_n}$ for the input $\mathcal{O}$ in Fig. 3. The lower left is the rain kernels $\mathcal{C}$ learned from Rain100L. The images are better to be zoomed in on screen.

Taking $S = 0$ as a baseline, it is seen that only with 2 stages, our method achieves significant rain removal performance, which validates the essential role of the proposed M-net and B-net. We also observe that when $S = 20$, its deraining performance is slightly lower than that of $S = 17$ since larger $S$ would make gradient propagation more difficult. Based on such observation, we easily set $S$ as 17 throughout all our experiments. More ablation results and discussions are listed in supplementary material.

### 5.2. Model verification

We then show how the interpretability of this RCDNet facilitates an easy analysis for the working mechanism inside the network modules.

Fig. 3 presents the extracted background layer $\mathcal{B}^{(s)}$ (1st row), $\widehat{\mathcal{B}}^{(s)}$ (2nd row) that represents the role of M-net in helping restore clean background, and rain layer $\mathcal{R}^{(s)}$ (3rd row) at different stages. We can find that with the increase of $s$, $\mathcal{R}^{(s)}$ covers more rain streaks and fewer image details, and $\widehat{\mathcal{B}}^{(s)}$ and $\mathcal{B}^{(s)}$ are also gradually ameliorated. These should

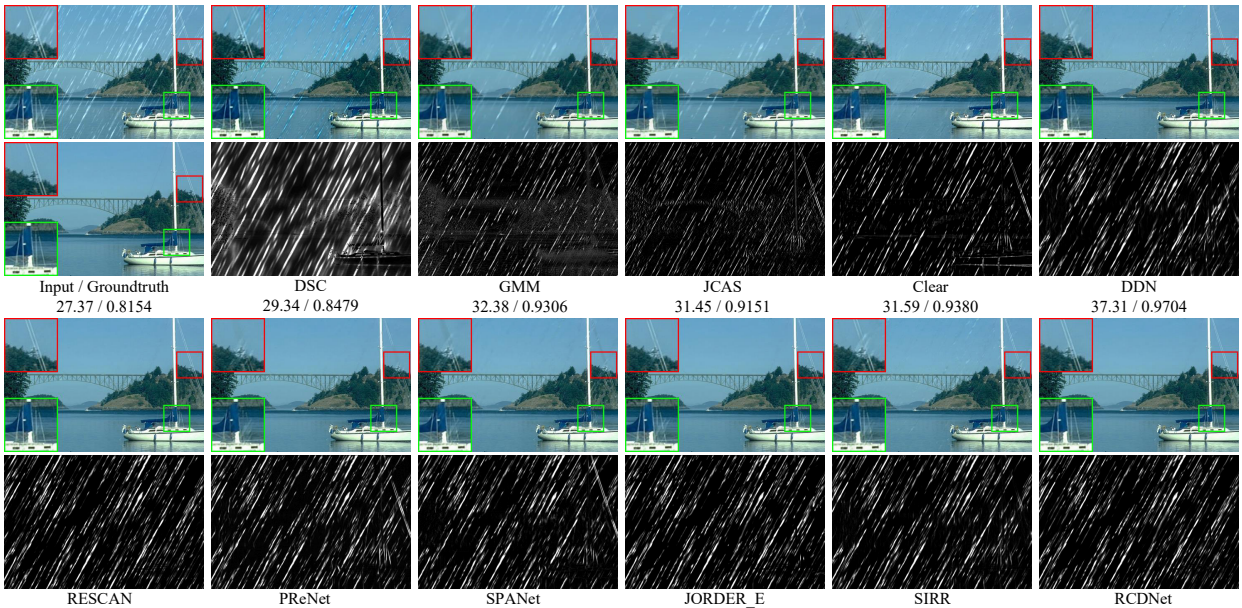



Figure 5. 1st column: input rainy image (upper) and groundtruth (lower). 2nd-12th column: derained results (upper) and extracted rain layers (lower) by 11 competing methods. PSNR/SSIM for reference. Bold indicates top 1st rank.

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 RCDNet to be evolved to a right direction.

Fig. 4 presents the learned rain kernels and the rain maps for the input $\mathcal{O}$ in Fig. 3. Clearly, the RCDNet finely extracts proper rain layers explicitly based on the RCD model. 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.

Table 2. PSNR and SSIM comparisons on four benchmark datasets. Bold and bold italic indicate top 1st and 2nd rank, respectively.

| Datasets | Rain100L | | Rain100H | | Rain1400 | | Rain12 | |
|---|---|---|---|---|---|---|---|---|
| Metrics | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| Input | 26.90 | 0.8384 | 13.56 | 0.3709 | 25.24 | 0.8097 | 30.14 | 0.8555 |
| DSC[51] | 27.34 | 0.8494 | 13.77 | 0.3199 | 27.88 | 0.8394 | 30.07 | 0.8664 |
| GMM[28] | 29.05 | 0.8717 | 15.23 | 0.4498 | 27.78 | 0.8585 | 32.14 | 0.9145 |
| JCAS[13] | 28.54 | 0.8524 | 14.62 | 0.4510 | 26.20 | 0.8471 | 33.10 | 0.9305 |
| Clear[8] | 30.24 | 0.9344 | 15.33 | 0.7421 | 26.21 | 0.8951 | 31.24 | 0.9353 |
| DDN[9] | 32.38 | 0.9258 | 22.85 | 0.7250 | 28.45 | 0.8888 | 34.04 | 0.9330 |
| RESCAN[27] | 38.52 | 0.9812 | 29.62 | 0.8720 | 32.03 | 0.9314 | 36.43 | 0.9519 |
| PReNet[35] | 37.45 | 0.9790 | 30.11 | ***0.9053*** | ***32.55*** | ***0.9459*** | 36.66 | 0.9610 |
| SPANet[41] | 35.33 | 0.9694 | 25.11 | 0.8332 | 29.85 | 0.9148 | 35.85 | 0.9572 |
| JORDER_E[49] | ***38.59*** | ***0.9834*** | ***30.50*** | 0.8967 | 32.00 | 0.9347 | ***36.69*** | ***0.9621*** |
| SIRR[44] | 32.37 | 0.9258 | 22.47 | 0.7164 | 28.44 | 0.8893 | 34.02 | 0.9347 |
| RCDNet | **40.00** | **0.9860** | **31.28** | **0.9093** | **33.04** | **0.9472** | **37.71** | **0.9649** |

## 5.3. Experiments on synthetic data

**Comparison methods and datasets.** We then compare our network with the current SOTA single image deraining, including model-based DSC [51], GMM [28], and JCAS [13]; DL-based Clear [8], DDN [9], RESCAN [27], PReNet [35], SPANet [41], JORDER_E [49], and SIRR [44][6], based on four benchmark datasets, including Rain100L, Rain100H [49], Rain1400 [9], and Rain12 [28].

Fig. 5 illustrates the deraining performance of all competing methods on a rainy image from Rain100L. As shown, the deraining result of RCDNet is better than that of other methods in sufficiently removing the rain streaks and finely recovering the image textures. Moreover, the rain layer extracted by RCDNet contains fewer unexpected background details as compared with other competing methods. Our RCNet thus achieves the best PSNR and SSIM.

Table 2 reports the quantitative results of all competing methods. It is seen that our RCDNet attains best deraining performance among all methods on each dataset. This substantiates the flexibility and generality of our method, in diverse rain types contained in these datasets.

## 5.4. Experiments on real data

We then analyze the performance of all methods on two real datasets from [41]: the first one (called SPA-Data) contains 638492 rainy/clean image pairs for training and 1000 testing ones, and the second one (called Internet-Data) includes 147 rainy images without groundtruth.

Table 3 and Fig. 6 compare the derained results on SPA-Data of all competing methods visually and quantitatively. It is easy to see that even for such complex rain patterns, the proposed RCDNet still achieves an evident superior perfor-

[6]The code/project links for these comparison methods are listed in supplementary material.

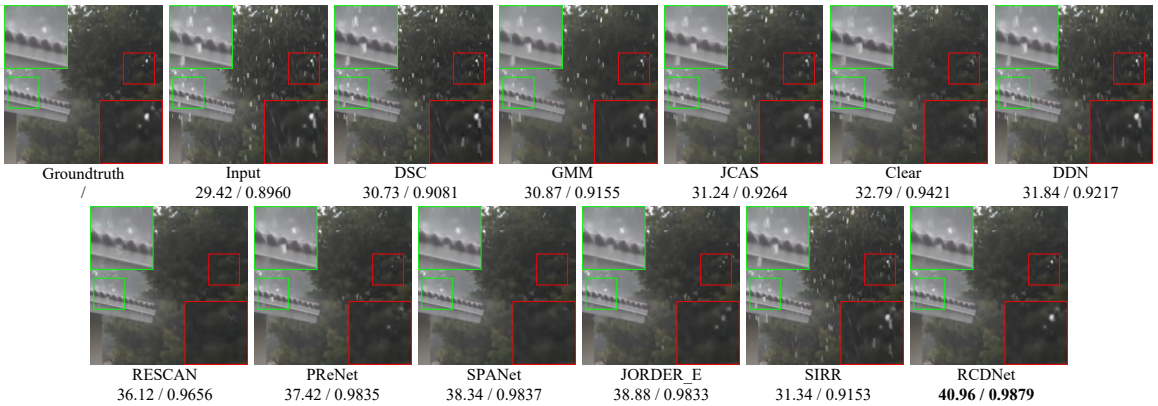



Figure 6. Rain removal performance comparisons on a rainy image from SPA-Data. The images are better to be zoomed in on screen.

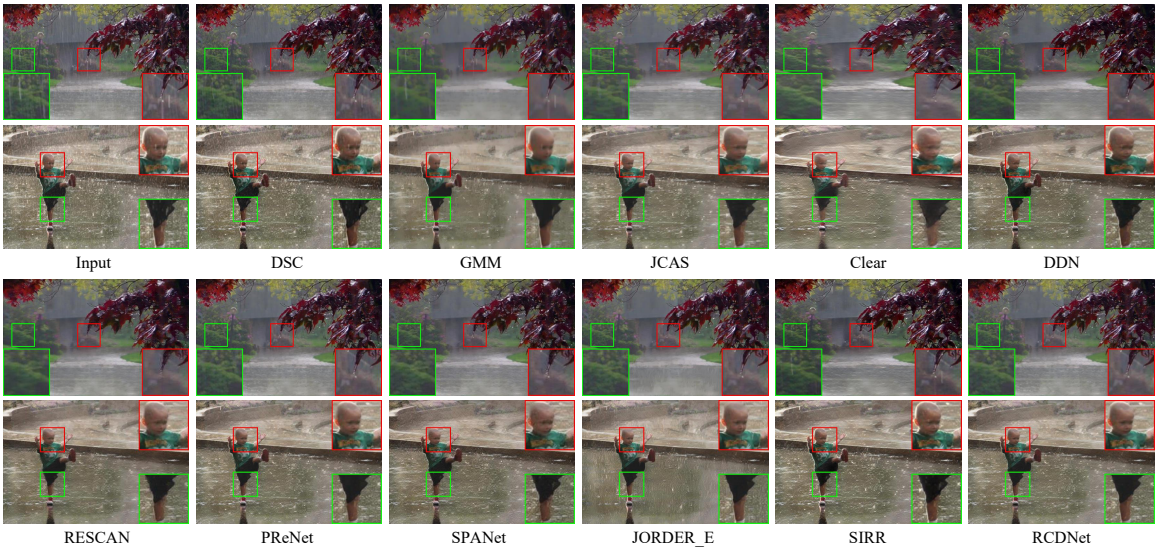



Figure 7. Derained results for two samples with various rain patterns from Internet-Data. The images are better to be zoomed in on screen.

Table 3. PSNR and SSIM comparisons on SPA-Data [41].

| Methods | Input | DSC | GMM | JCAS | Clear | DDN |
|---|---|---|---|---|---|---|
| PSNR | 34.15 | 34.95 | 34.30 | 34.95 | 34.39 | 36.16 |
| SSIM | 0.9269 | 0.9416 | 0.9428 | 0.9453 | 0.9509 | 0.9463 |
| Methods | RESCAN | PReNet | SPANet | JORDER_E | SIRR | RCDNet |
| PSNR | 38.11 | 40.16 | 40.24 | ***40.78*** | 35.31 | **41.47** |
| SSIM | 0.9707 | ***0.9816*** | 0.9811 | 0.9811 | 0.9411 | **0.9834** |

mance than other methods. Especially, similar to its superiority in synthetic experiments, it is also observed that our method better removes the rain streaks and recovers image details than other competing ones.

Further, we select two real hard samples with various rain densities to evaluate the generalization ability of all competing methods. From Fig. 7, we can find that traditional model-based methods tend to leave obvious rain streaks. Although DL-based comparison methods remove apparent rain streaks, they still leave distinct rain marks or blur some image textures. Comparatively, our RCDNet better preserves background details as well as removes more rain streaks. This shows its good generalization capability to unseen complex rain types.

## 6. Conclusion

In this paper, we have explored the intrinsic prior structure of rain streaks that can be explicitly expressed as convolutional dictionary learning model, and proposed a novel interpretable network architecture for single image deraining. Each module in the network can one-to-one correspond to the implementation operators of the algorithm designed for solving the model, and thus the network is almost "white-box" with easily visualized interpretation for all its module elements. Comprehensive experiments implemented on synthetic and real rainy images validate that such interpretability brings a good effect of the proposed network, and especially facilitates the analysis for how it happens in the network and why it works in testing prediction process. The extracted elements through the end-to-end learning by the network, like the rain kernels, are also potentially useful for the related tasks on rainy images.

**Acknowledgment.** This research was supported by the China NSFC projects under contract 11690011, 61721002, U1811461 and MoE-CMCC "Artifical Intelligence" Project with No. MCM20190701

## References


[1] Peter C Barnum, Srinivasa Narasimhan, and Takeo Kanade. Analysis of rain and snow in frequency space. *International journal of computer vision*, 86(2-3):256, 2010. 3

[2] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009. 2, 4

[3] Jérémie Bossu, Nicolas Hautière, and Jean-Philippe Tarel. Rain or snow detection in image sequences through use of a histogram of orientation of streaks. *International journal of computer vision*, 93(3):348–367, 2011. 3

[4] Yi Lei Chen and Chiou Ting Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1968–1975, 2013. 3

[5] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–575, 2003. 1

[6] Xinghao Ding, Liqin Chen, Xianhui Zheng, Huang Yue, and Delu Zeng. Single image rain and snow removal via guided l0 smoothing filter. *Multimedia Tools and Applications*, 75(5):2697–2712, 2016. 3

[7] David L Donoho. De-noising by soft-thresholding. *IEEE transactions on information theory*, 41(3):613–627, 1995. 4

[8] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing*, 26(6):2944–2956, 2017. 2, 3, 7

[9] Xueyang Fu, Jiabin Huang, Delu Zeng, Huang Yue, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3855–3863, 2017. 2, 3, 7

[10] Kshitiz 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*, volume 1, pages I–I, 2004. 3

[11] Kshitiz Garg and Shree K Nayar. When does a camera see rain? In *Tenth IEEE International Conference on Computer Vision*, volume 2, pages 1067–1074, 2005. 3

[12] Kshitiz Garg and Shree K Nayar. Vision and rain. *International Journal of Computer Vision*, 75(1):3–27, 2007. 3

[13] Shuhang Gu, Deyu Meng, Wangmeng Zuo, and Zhang Lei. Joint convolutional analysis and synthesis sparse representation for single image layer separation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1708–1716, 2017. 2, 3, 4, 7

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[15] Zhang He and Vishal M. Patel. Convolutional sparse and low-rank coding-based rain streak removal. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1259–1267, 2017. 2, 3, 4

[16] Furong Huang and Animashree Anandkumar. Convolutional dictionary learning through tensor factorization. *Computer Science*, pages 1–30, 2015. 2, 4

[17] Q. Huynh-Thu and M. Ghanbari. Scope of validity of p-snr in image/video quality assessment. *Electronics Letters*, 44(13):800–801, 2008. 6

[18] Tai Xiang Jiang, Ting Zhu Huang, Xi Le Zhao, Liang Jian Deng, and Yao Wang. A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 4057–4066, 2017. 3

[19] Chen Jie, Cheen Hau Tan, Junhui Hou, Lap Pui Chau, and Li 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*, pages 6286–6295, 2018. 3

[20] Kim Jin-Hwan, Sim Jae-Young, and Kim Chang-Su. Video deraining and desnowing using temporal correlation and low-rank matrix completion. *IEEE Transactions on Image Processing*, 24(9):2658–2670, 2015. 3

[21] Xu Jing, Zhao Wei, Liu Peng, and Xianglong Tang. Removing rain and snow in a single image using guided filter. In *IEEE International Conference on Computer Science and Automation Engineering*, volume 2, pages 304–307, 2012. 3, 6

[22] 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*, 21(4):1742–1755, 2012. 3

[23] Jin Hwan Kim, Chul Lee, Jae Young Sim, and Chang Su Kim. Single-image deraining using an adaptive nonlocal means filter. In *IEEE International Conference on Image Processing*, pages 914–917, 2014. 3

[24] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Computer Science*, 2014. 6

[25] Minghan Li, Qi Xie, Qian Zhao, Wei Wei, Shuhang Gu, Jing Tao, and Deyu Meng. Video rain streak removal by multiscale convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6644–6653, 2018. 2, 3, 4

[26] Siyuan Li, Iago Breno Araujo, Wenqi Ren, Zhangyang Wang, Eric K Tokuda, Roberto Hirata Junior, Roberto Cesar-Junior, Jiawan Zhang, Xiaojie Guo, and Xiaochun Cao. Single image deraining: A comprehensive benchmark analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3838–3847, 2019. 1

[27] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *Proceedings of the European Conference on Computer Vision*, pages 254–269, 2018. 2, 3, 7

[28] Yu Li. Rain streak removal using layer priors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2736–2744, 2016. 2, 3, 7

[29] Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming Guo. D3r-net: Dynamic routing residue recurrent network for

video rain removal. *IEEE Transactions on Image Processing*, 28(2):699–712, 2018. 3

[30] Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming Guo. Erase or fill? deep joint recurrent rain removal and reconstruction in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3233–3242, 2018. 3

[31] O. Ludwig, David Delgado, Valter Goncalves, and Urbano Nunes. Trainable classifier-fusion schemes: an application to pedestrian detection. In *International IEEE Conference on Intelligent Transportation Systems*, pages 1–6, 2009. 1

[32] Pan Mu, Jian Chen, Risheng Liu, Xin Fan, and Zhongxuan Luo. Learning bilevel layer priors for single image rain streaks removal. *IEEE Signal Processing Letters*, 26(2):307–311, 2019. 3

[33] Wan-Joo Park and Kwae-Hi Lee. Rain removal using kalman filter in video. In *International Conference on Smart Manufacturing Application*, pages 494–497, 2008. 3

[34] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[35] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: a better and simpler baseline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3937–3946, 2019. 2, 3, 7

[36] Weihong Ren, Jiandong Tian, Han Zhi, Antoni Chan, and Yandong Tang. Video desnowing and deraining based on matrix decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4210–4219, 2017. 3

[37] M. S. Shehata, Jun Cai, W. M. Badawy, T. W. Burr, M. S. Pervez, R. J. Johannesson, and Ahmad Radmanesh. Video-based automatic incident detection for smart roads: The outdoor environmental challenges regarding false alarms. *IEEE Transactions on Intelligent Transportation Systems*, 9(2):349–360, 2008. 1

[38] Shao-Hua Sun, Shang-Pu Fan, and Yu-Chiang Frank Wang. Exploiting image structural similarity for single image rain removal. In *IEEE International Conference on Image Processing (ICIP)*, pages 4482–4486, 2014. 3

[39] Hong Wang, Yichen Wu, Minghan Li, Qian Zhao, and Deyu Meng. A survey on rain removal from video and single image. *arXiv:1909.08326*, 2019. 1

[40] Hong Wang, Qi Xie, Yichen Wu, Qian Zhao, and Deyu Meng. Single image rain streaks removal: a review and an exploration. *International Journal of Machine Learning and Cybernetics*, pages 1–20, 2020. 2

[41] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12270–12279, 2019. 2, 3, 7, 8

[42] 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*, 26(8):3936–3950, 2017. 3

[43] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Processing*, 13(4):600–612, 2004. 6

[44] Wei Wei, Deyu Meng, Qian Zhao, Zongben Xu, and Ying Wu. Semi-supervised transfer learning for image rain removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3877–3886, 2019. 3, 7

[45] Wei Wei, Lixuan Yi, Qi Xie, Qian Zhao, Deyu Meng, and Zongben Xu. Should we encode rain streaks in video as deterministic or stochastic? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2516–2525, 2017. 3

[46] Brendt Wohlberg. Efficient convolutional sparse coding. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014. 4

[47] Qi Xie, Minghao Zhou, Qian Zhao, Deyu Meng, Wangmeng Zuo, and Zongben Xu. Multispectral and hyperspectral image fusion by ms/hs fusion net. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1585–1594, 2019. 5

[48] Dong Yang and Jian Sun. Proximal dehaze-net: A prior learning-based deep network for single image dehazing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 702–717, 2018. 5

[49] Wenhan Yang, Robby T. Tan, Jiashi Feng, Jiaying Liu, Shuicheng Yan, and Zongming Guo. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2019. 3, 6, 7

[50] Yan Yang, Jian Sun, Huibin Li, and Zongben Xu. Admm-net: A deep learning approach for compressive sensing mri. *arXiv preprint arXiv:1705.06869*, 2017. 5

[51] Luo Yu, Xu Yong, and Ji Hui. Removing rain from a single image via discriminative sparse coding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3397–3405, 2015. 2, 3, 7

[52] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 695–704, 2018. 2, 3

[53] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019. 2, 3

[54] Jiawei Zhang, Jinshan Pan, Wei-Sheng Lai, Rynson WH Lau, and Ming-Hsuan Yang. Learning fully convolutional networks for iterative non-blind deconvolution. 2017. 5

[55] Xiaopeng Zhang, Hao Li, Yingyi Qi, Wee Kheng Leow, and Teck Khim Ng. Rain removal in video by combining temporal and chromatic properties. In *IEEE International Conference on Multimedia and Expo*, pages 461–464, 2006. 3

[56] Lei Zhu, Chi Wing Fu, Dani Lischinski, and Pheng Ann Heng. Joint bi-layer optimization for single-image rain streak removal. In *Proceedings of the IEEE international conference on computer vision*, pages 2526–2534, 2017. 3