Learning Dual Convolutional Dictionaries for Image De-raining

Chengjie Ge
cjge@mail.ustc.edu.cn
University of Science and Technology of China
Hefei, Anhui, China

Xueyang Fu*
xyfu@ustc.edu.cn
University of Science and Technology of China
Hefei, Anhui, China

Zheng-Jun Zha
zhazj@ustc.edu.cn
University of Science and Technology of China
Hefei, Anhui, China

ABSTRACT

Rain removal is a vital and highly ill-posed low-level vision task. While currently existing deep convolutional neural networks (CNNs) based image de-raining methods have achieved remarkable results, they still possess apparent shortcomings: First, most of the CNNs based models are lack of interpretability. Second, these models are not embedded with physical structures of rain streaks and background images. Third, they omit useful information in the background images. These deficiencies result in unsatisfied de-raining results in some sophisticated scenarios. To solve the above problems, we propose a Deep Dual Convolutional Dictionary Learning Network (DDCDNet) for these specific tasks. We firstly propose a new dual dictionary learning objective function, and then unfold it into the form of neural networks to learn prior knowledge from the data automatically. This network tries to learn the rain-streaks layer and the clean background using two dictionary learning networks instead of merely predicting the rain-streaks layer like most of the de-raining methods. To further increase the interpretability and generalization capability, we add sparsity and adaptive dictionary to our network to generate dynamic dictionary for each image based on content. Experimental results reveal that our model possesses outstanding de-raining ability on both synthetic and real-world data sets in terms of PSNR and SSIM as well as visual appearance.

CCS CONCEPTS

• **Computing methodologies** → **Reconstruction**.

KEYWORDS

Image De-raining; Convolutional Dictionary Learning; Optimization; Deep Unfolding; Image Restoration

*Corresponding author: Xueyang Fu (xyfu@ustc.edu.cn). This work was supported by the National Key R&D Program of China under Grant 2020AAA0105702, the National Natural Science Foundation of China (NSFC) under Grants U19B2038 and 61901433, the University Synergy Innovation Program of Anhui Province under Grants GXXT-2019-025, the Fundamental Research Funds for the Central Universities under Grant WK2100000024, and the USTC Research Funds of the Double First-Class Initiative under Grant YD2100002003.

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3548117

ACM Reference Format:
Chengjie Ge, Xueyang Fu, and Zheng-Jun Zha. 2022. Learning Dual Convolutional Dictionaries for Image De-raining. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, Lisboa, Portugal, 9 pages. https://doi.org/10.1145/3503161.3548117

1 INTRODUCTION

Rainy day is a common weather in our daily life, the presence of rain streaks would exert negative effects on many multimedia tasks like face-recognition, human–computer interaction, object detection, etc. To mitigate this issue, numerous de-raining methods have been raised. Those methods can be roughly summarized into two categories: model-driven and data-driven methods. Model-driven de-raining methods need complex handcrafted priors [12, 16, 26, 27], which have deficiencies like time-consuming and model-insufficiency. While data-driven deep learning based methods are able to extract the meaningful information from the rain images and remove the rain streaks easier [8, 9, 11, 22–25, 28, 30, 35, 38, 41, 44]. However, substantial numbers of networks are end-to-end learning, leading to the whole network merely looks like a black box. Moreover, many de-raining methods [25, 35, 47] use the prior knowledge that the original image **O** is equal to the background layer **B** plus the rain-stripe layer **R**:

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

Based on this equation, those methods directly predict **R** instead of **B** since the amount of information in the rain-streaks layer **R** is sparser than the background layer **B**, thus making the whole network easier to converge. In general, Eq. (1) is an essential priori estimation, and the methods above work barely satisfactory when the rain streaks are sparse. However, those methods omit the useful information in **B**. As the rain streaks become denser, it will cover large parts of contents in the background layer. Since the information in the **R** is monotonous, the increment of information in **R** can not compensate for the lost content in **B**. This makes the original ill-posed problem more difficult to handle, resulting in unsatisfactory de-raining results.

To alleviate the aforementioned problems, we argue to treat the information in **R** and **B** equally by adopting a dual dictionary unfolding network for the specific image de-raining. First, we design a simple network to initialize the background **B** and rain layer **R**. Second, our model uses two dictionary learning models to predict **R** and **B** respectively. Each dictionary model is composed of two deep unfolding models with sparse coding and adaptive dictionaries. Moreover, our unfolding model has closed-form solutions for both dictionaries and sparse encoding. In this way, our de-raining network has the capability to precisely generate dynamic dictionary and sparse coefficient individually. The detailed model structure

will be introduced in the methodology part. The main contributions of this paper are summarized as follows:

- We propose a model called deep dual convolutional dictionary network (DDCDNet) to learn the knowledge of rain-streaks layer and background respectively.
- DDCDNet learns the prior from the training data through unfolding process, overcoming the poor interpretability and short-comings of handcrafted priors.
- Extensive experiment results show that our DDCDNet has achieved the state-of-the-art methods on both synthetic and real-world data sets. Especially under heavy rain situations, our DDCDNet not only restore fine textures from damaged picture but also improve the PSNR more than 0.7dB compared with the previous best model [37].

2 RELATED WORK

2.1 Single image de-raining

In the past decades, numerous promising de-raining methods have been raised. These methods have promoted the de-raining task to a great extent, which can be roughly divided into two categories.

Traditional methods. Traditional rain removal methods usually need us to manually construct a large number of priors. These methods focus on the physical structure of the rain streaks as well as the background image. Kang *et al.* [16] propose a method to split one image into high-frequency and low-frequency parts by using a bilateral layer (MCA). Luo *et al.* [27] manage to split the background layer and rain-streaks layer by applying highly discriminative sparse coding (DSC). Li *et al.* [26] use a model based on Gaussian mixture model to fit the distribution and the scale of rain-streaks patterns (GMM). Similarly, using the complementary representation mechanism of analysis sparse representation (ASR) and synthesis sparse representation (SSR), Gu *et al.* [12] propose a joint convolution analysis and synthesis sparse representation model (JCAS). Albeit these traditional methods perform well in some given situations, the generalization of these methods leave much to be desired. Especially, when the rain-streaks layer does not meet the prior of these models, the removal of rain streaks will be insufficient or excessive.

Deep learning methods. Recently, the data-driven deep learning methods become dominant in high-level [5, 14] and low-level vision tasks [13, 29, 31, 42]. Thanks to the powerful feature extraction ability of deep convolutional neural networks (CNNs), we are able to witness the rapid development of this field. Fu *et al.* [10] is the first to introduce residual neural networks into de-raining task. Rescan [25] use the dilation convolution and recurrent neural networks (RNNs) to repeatedly consider the relationship between contextual information among different de-raining stages. Li *et al.* [22] introduce a non-local encoder-decoder block to understand the abstract feature representation in the rain-streaks layer. JORDER-E [41] propose a model that uses a rain pattern binary mask to mark the location area of rain spots, which can represent the superposition of rain patterns of different shapes and directions. They also raise two new synthesized data sets as a baseline for de-raining. Ren *et al.* [30] design a progressive de-rain network (PReNet) by unfolding several Resblocks and long short term networks (LSTMs). Wang *et al.* [38] propose a spatial attention-based image de-raining network using high-quality real data sets and a method to obtain rain-free images from a series of real rain images. DCSFN [35] propose a cross-scale network to fuse features by using inner-scale connection blocks. DRDnet [7] adopt two-sub parallel networks Rain Residual Network (RRN) and Detail Repair Network (DRN) for extracting rain streaks and image details separately. Zhang *et al.* [47] propose a model based on self-attention, which also merges adaptive pyramid structure to reduce the computational burden and obtains multi-attention features by using Haar wavelet transform. RCDnet [36] propose a model-driven deep learning de-raining method. They use a set of rain kernels to extract the rain layer feature map, then applying the proximal gradient descent to update the network. Wang *et al.* [37] propose a rain streaks generative model, which could generate numerous similar training pairs. It can be embedded in other de-raining methods to retain better performance.

2.2 Deep unfolding

Different from traditional end-to-end models, deep unfolding methods are devoted to hold the strong learning capability while retaining good interpretability. The deep unfolding methods date back to decades ago [32, 34]. Since then, certain numbers of algorithms(*e.g.*, iterative shrinkage-threshold [17], half-quardratic splitting [2, 46] and primal-dual [1]) have been unfolded for different vision tasks such as image de-blurring [20, 33], image de-noising [6, 21, 48], image super-resolution [46] and image de-mosaicking [19]. Although deep unfolding has added interpretability to deep learning model, they also introduce some drawbacks. First, the lack of CNNs would lower the performance when solving the sub-problems. Second, many deep unfolding methods do not have a closed-form solution, which may result in non-convergence of the whole network [46]. Third, most of the deep unfolding methods assume dictionaries as fixed patterns, which decrease the adaptability of image representation. Therefore, a model that surmounts the overall drawbacks is needed.

3 METHODOLOGY

In general, we often split one rainy image as the combination of rain-streaks layer **R** and background layer **B** as Eq.(1) shows. Most of the de-raining networks focus on predicting **R** rather than **B** because the information in rain layer **R** is sparser than **B**. However, as the rain streaks become denser, the streaks will cover quite a lot of background information. Reconstructing clear background from the rain image will become more difficult because the information volume in **R** is quite limited.

3.1 Overall Architecture

To avert the disappointments above, we need to make use of the information in **B** to help us restore more detailed textures. So, we design a Deep Dual Convolutional Dictionary Learning Network to extract features from both rain layer and background layer. As shown in Figure 1, the whole network is composed of **T** stages, each stage contains two dictionary learning blocks.

3.2 Dictionary Learning for R and D

Inspired by K-SVD [3], one picture **Y** can be decomposed as the multiple of dictionary matrix **D** and sparse encoding matrix **X**, where

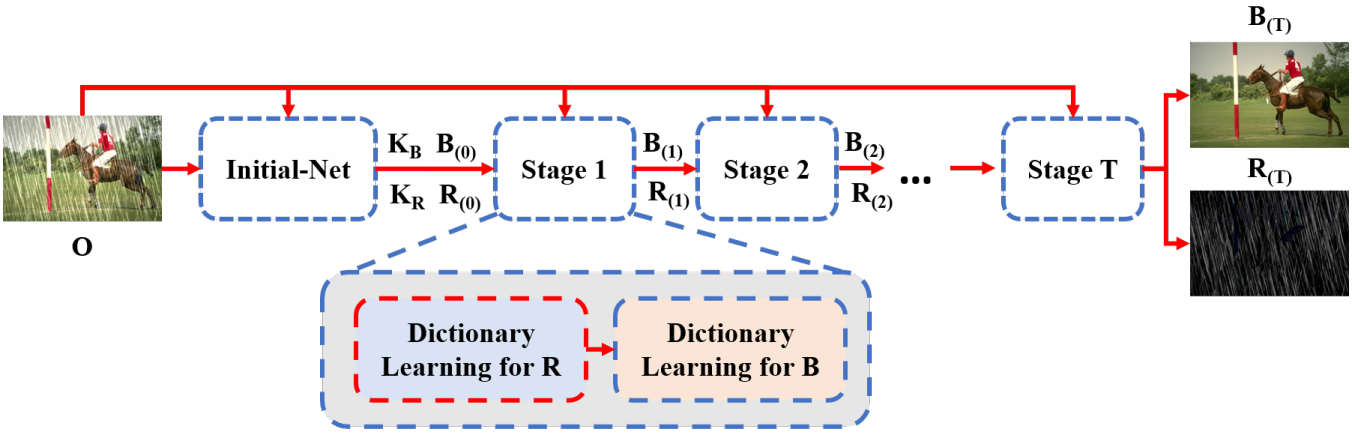

Figure 1: The proposed network. The network consists of T stages with two dictionary learning models. At each stage, our network firstly predict R then predict B successively.

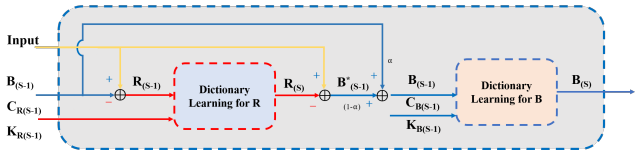

Figure 2: The architecture of our DDCDNet in one stage.

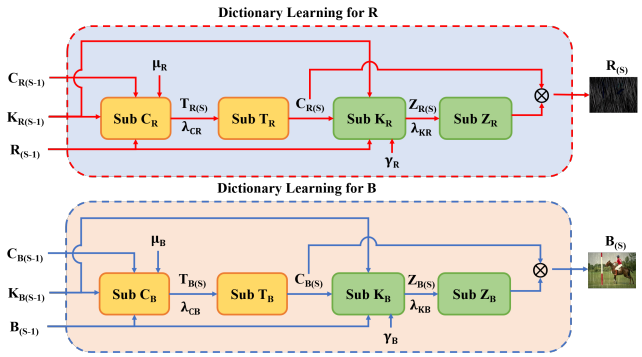

Figure 3: The detailed architecture of our learning blocks. Each dictionary learning block consists four sub blocks, which enabling the unfolding methods to solve the equations iteratively.

the **D** stores the feature in **Y** and **X** represents how these features combined. In this way, the rain-streaks layer and background layer can be modeled as:

$$\mathbf{R} = \sum_{i=1}^{T} \mathbf{K}_{\mathbf{R}}^{(i)} \otimes \mathbf{C}_{\mathbf{R}}^{(i)}, \tag{2}$$

$$\mathbf{B} = \sum_{i=1}^{T} \mathbf{K}_{\mathbf{B}}^{(i)} \otimes \mathbf{C}_{\mathbf{B}}^{(i)}. \tag{3}$$

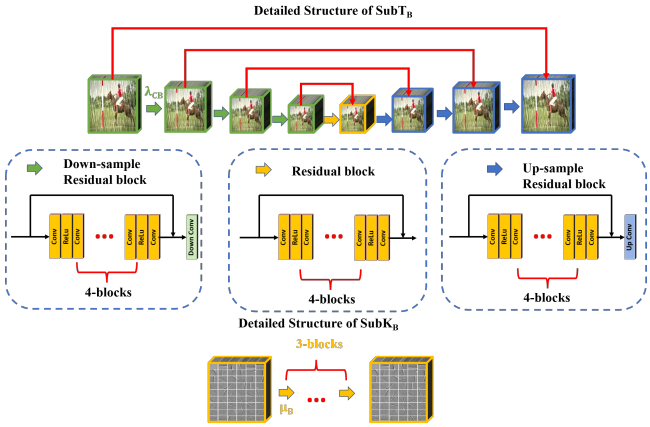

Figure 4: Architectures of sub blocks in dictionary learning.

In Eq.(1) and Eq.(2) the $\mathbf{R}, \mathbf{B} \in \mathbb{R}^{H \times W \times 3}$, and $\mathbf{K}_{\mathrm{R}}^{(i)}$, $\mathbf{K}_{\mathrm{B}}^{(i)}$ represent the dictionary of **R** and **B** in the i th stage; $\mathbf{C}_{\mathrm{R}}^{(i)}$, $\mathbf{C}_{\mathrm{B}}^{(i)}$ represent the sparse encoding of **R** and **B** in the i th stage. Now, our Eq.(1) can be rewritten as:

$$\mathbf{O} = \sum_{i=1}^{T} \mathbf{K}_{\mathbf{R}}^{(i)} \otimes \mathbf{C}_{\mathbf{R}}^{(i)} + \sum_{i=1}^{T} \mathbf{K}_{\mathbf{B}}^{(i)} \otimes \mathbf{C}_{\mathbf{B}}^{(i)}. \tag{4}$$

We notice that our model can be divided into two sub optimization problems. Previous dictionary learning for de-raining task usually assumes the rain dictionaries or background dictionaries as fixed patterns, and this will lead to the whole model lack of generalization ability and loss of image details. Therefore, we add adaptive dictionary item for each stage and use the powerful learning ability of deep learning to extract the deep prior for each image. After applying the adaptive dictionary **D**, our optimization functions can

be written as:

$$\min_{\mathbf{K_R},\mathbf{C_R}} \frac{1}{2\sigma^2}\|\mathbf{O}-\mathbf{B}-\sum_{i=1}^{T}\mathbf{K_R}^{(i)}\otimes\mathbf{C_R}^{(i)}\|_2^2+\lambda_{\mathbf{CR}}\psi(\mathbf{K_R})+\lambda_{\mathbf{KR}}\phi(\mathbf{C_R}). \tag{5}$$

$$\min_{\mathbf{K_B},\mathbf{C_B}} \frac{1}{2\sigma^2}\|\mathbf{B}-\sum_{i=1}^{T}\mathbf{K_B}^{(i)}\otimes\mathbf{C_B}^{(i)}\|_2^2+\lambda_{\mathbf{CB}}\psi(\mathbf{K_B})+\lambda_{\mathbf{KB}}\phi(\mathbf{C_B}). \tag{6}$$

For simplicity, we firstly focus on optimizing Eq.(6). In order to solve this bi-criteria optimization equation, we unfold $\mathbf{C_B}$. The optimization model of $\mathbf{C_B}$ is (we omit the superscript of i and the summation notation for simpler expression):

$$\min_{\mathbf{K_B},\mathbf{C_B}} \|\mathbf{B}-\mathbf{K_B}\otimes\mathbf{C_B}\|_2^2+\lambda_{\mathbf{CB}}\psi(\mathbf{C_B}), \tag{7}$$

Eq.(7) can be decomposed as fidelity item and penalty item. Then we follow [48] and utilize Half Quadratic Splitting (HQS) algorithm [15] to Eq.(7) due to its simplicity and fast convergence. By introducing an auxiliary variable $\mathbf{T_B}$ and penalty parameter μ_B, optimizing Eq.(7) can be addressed by solving the following sub-problems for $\mathbf{C_B}$ and $\mathbf{T_B}$ as shown in Eq.(8) and Eq.(9):

$$\mathbf{T_B}=arg\min_{\mathbf{T_B}}\|\mathbf{B}-\mathbf{K_B}\otimes\mathbf{T_B}\|_2^2+\mu_{\mathcal{B}}\sigma^2\|\mathbf{T_B}-\mathbf{C_B}\|_2^2, \tag{8}$$

$$\mathbf{C_B}=arg\min_{\mathbf{C_B}}\frac{\mu_{\mathcal{B}}}{2}\phi\|\mathbf{T_B}-\mathbf{C_B}\|_2^2+\lambda_{\mathbf{CB}}\psi(\mathbf{C_B}). \tag{9}$$

Then we perform the same unfolding operation on $\mathbf{K_B}$, we will get the similar iterative equations:

$$\mathbf{Z_B}=arg\min_{\mathbf{Z_B}}\|\mathbf{B}-\mathbf{Z_B}\otimes\mathbf{C_B}\|_2^2+\gamma_{\mathcal{B}}\sigma^2\|\mathbf{Z_B}-\mathbf{K_B}\|_2^2, \tag{10}$$

$$\mathbf{K_B}=arg\min_{\mathbf{K_B}}\frac{\mu_{\mathcal{B}}}{2}\phi\|\mathbf{Z_B}-\mathbf{K_B}\|_2^2+\lambda_{\mathbf{KB}}\psi(\mathbf{K_B}). \tag{11}$$

We design four sub-blocks to solve Eq.(8-11). According to [4], by taking the derivatives of the Eq.(8) and Eq.(10), these equations have closed form solutions:

$$\mathcal{T}_{\mathcal{B}}=\mathcal{F}^{-1}\left\{(\mathcal{K_B}^{\mathcal{H}}\mathcal{K_B}+\mu_{\mathcal{B}}\sigma^2\mathrm{I})^{-1}(\mathcal{K_B}^{\mathcal{H}}\mathcal{B}+\mu_{\mathcal{B}}\sigma^2\mathcal{C_B})\right\}, \tag{12}$$

$$\mathcal{Z}_{\mathcal{B}}=\mathcal{F}^{-1}\left\{(\mathcal{C_B}^{\mathcal{H}}\mathcal{C_B}+\gamma_{\mathcal{B}}\sigma^2\mathrm{I})^{-1}(\mathcal{C_B}^{\mathcal{H}}\mathcal{B}+\gamma_{\mathcal{B}}\sigma^2\mathcal{K_B})\right\}. \tag{13}$$

In these equations, $\mathcal{F}$ stands for the Fast Fourier Transform (FFT); $\mathcal{F}^{-1}$ is the inverse Fast Fourier Transform (IFFT); $\mathcal{K_B}=\mathcal{F}(\mathbf{K_B})$ and $\mathcal{K}_{\mathcal{B}}^{\mathcal{H}}$ denotes complex conjugate of $\mathcal{K_B}$. Eq.(9) and Eq.(11) are complex convex optimization problems, which are hard and time-consuming to find closed-form solutions. In this case we adopt two neural networks $Sub\mathrm{T_B}$ and $Sub\mathrm{Z_B}$ to accomplish this process. The structures of $Sub\mathrm{T_B}$ and $Sub\mathrm{Z_B}$ are shown in Figure 3. The $Sub\mathrm{T_B}$ is an encoder-decoder structure. The convolution layers in encoder structure have 32, 64, 96 and 128 channels and each block is connected with four residual layers. Since the background dictionaries contain much less spatial information compared with sparse encoding layers, we adopt 3 residual-layers to extract the features from background dictionaries. The unfolding methods for Eq.(5) closely resemble with the previous steps. It should be noticed that the information in $\mathbf{R}$ is less than $\mathbf{B}$, so we change the channel numbers of the convolution layers in $Sub\mathrm{T_R}$ to 32, 48, 64 and 96.

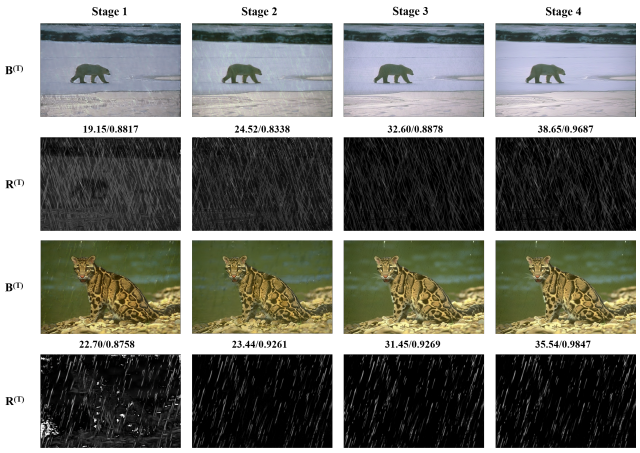

Figure 5: Visualization of the background image (with PSNR | SSIM) and rain-streaks image at different learning stages.

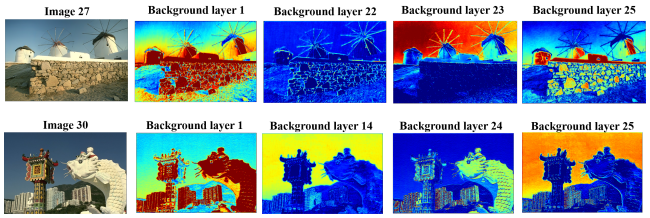

Figure 6: Visualization of learned feature maps in background layer. It is obvious to find that our network is able to highlight the elements in the background.

4 EXPERIMENTS

4.1 Implementation Details

All the experiments are implemented on a NVIDIA GeForce GTX 1080Ti GPU for 800 epochs. We use Pytorch1.7 and Adam gradient descent [18] with a batch size of 10 to train our network. All the training examples are cropped into 128×128 patch pairs with random horizontal flipping. We set the initial learning rate of 6e-4 and divided by 2 every 200 epochs.

4.2 Loss Function

Training loss. It has been proved that the SSIM loss is more suitable than L1 (MAE) and L2 (MSE) loss for de-raining task [30]. We adopt the progressive multi-stage SSIM loss for the rain-layer $\mathbf{R}$ and background $\mathbf{B}$:

$$\mathcal{L}=-\sum_{i=1}^{T}\alpha_i\cdot SSIM(\mathbf{B^{gt}},\mathbf{B^{(i)}})-\sum_{i=1}^{T}\beta_i\cdot SSIM(\mathbf{O}-\mathbf{B^{gt}},\mathbf{R^{(i)}}), \tag{14}$$

where α_i and β_i are equal to $\frac{i}{T}$. More details will be shown in the ablation study.

4.3 Model Visualization

Stage visualization. We visually show the background images and rain-streaks layers under different predicting stages. The first and

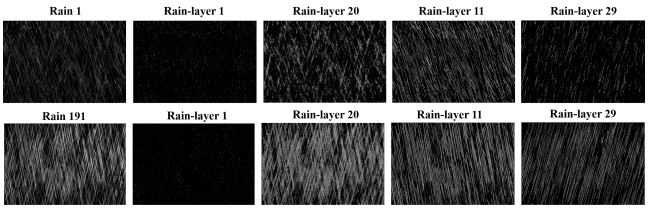

Figure 7: Visualization of learned feature maps in rain-streaks layer. Our network could capture the different components of the rain-streaks. The images are better observed by zooming in on screen.

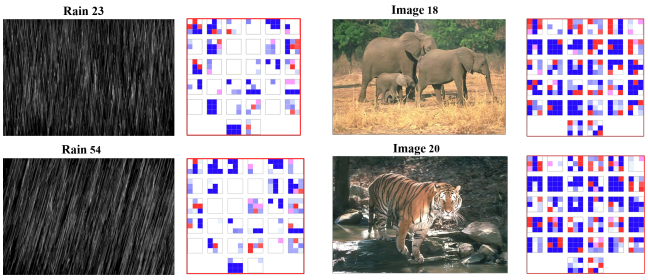

Figure 8: Visualization of adaptive dictionaries for rain-streaks and background images.

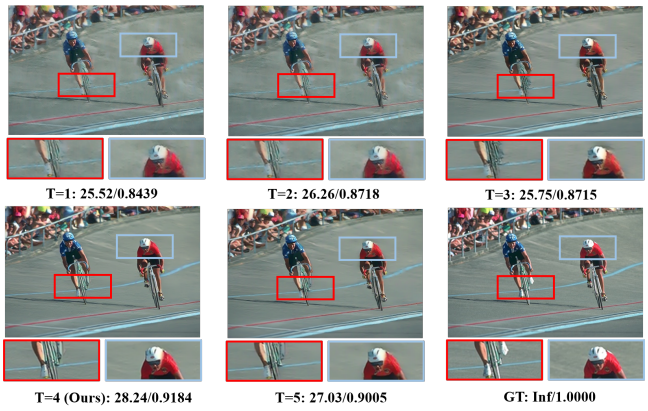

Figure 9: De-raining results on image 99 in Rain200H for ablation study.

Table 1: Ablation study on different dictionary learning methods. When applying single dictionary learning on R, we adopt a U-net structure for predicting B.

Version	Name	PSNR/SSIM
1	Single dictionary learning on **R**	29.86/90.29
2	Single dictionary learning on **B**	30.75/91.21
3	Dual dictionary learning	31.48/92.88

Table 2: Ablation study on adaptive dictionaries in different learning blocks

Version	Name	PSNR/SSIM
1	w/o adaptive **D** in **R**	30.74/92.12
2	w/o adaptive **D** in **B**	30.68/91.96
3	w/o adaptive **D** in two stage	30.15/91.22

Table 3: Analysis of different stage number based on PSNR and SSIM.

Stage No.	T=1	T=2	T=3	T=4	T=5
PSNR	28.81	28.9	30.17	31.48	30.62
SSIM	0.8818	0.8901	0.9077	0.9288	0.9154

third rows in Figure 5 are the prediction results of two images from data sets Rain200H and Rain200L. The second and fourth rows are the rain-streaks layers of the two images respectively. It is obvious to find that with the increase of the predicting stage **T**, the $\mathbf{B}^{(\mathbf{T})}$ contains less rain streaks and the $\mathbf{R}^{(\mathbf{T})}$ preserves more rain streaks and fewer image contents.

Feature map and adaptive dictionary visualization. We visualize the feature maps of the background images and rain-streaks layers learned by our dual-dictionary model. As shown in Figure 6, our model is able to highlight the components in the background, i.e., the buildings, walls and outline of the sky in row 1; sculpture, tower and the buildings in the distance in row 2. As to the feature maps for rain-layers in Figure 7, our model can also capture the different features in rain-streaks layers, i.e., it is obvious to find that the feature-map 11 records the rain streaks from left to right while feature-map 29 records the rain streaks in the opposite direction. At the same time, it should be noticed that the feature-map 1 records the intersection points of rain streaks from different directions and feature-map 20 denotes the overall architecture of the rain streaks. To validate the adaptive dictionary in our dual-dictionary model, we also visualize the dictionaries for **B** and **R** as shown in Figure 8. It can be seen that our DDCDNet can generate different dictionary patterns according to the contents of **B** and **R**, enabling our final results to retain more detailed textures.

Model generalization visualization. It has been verified that our DDCDNet is able to generate adaptive dictionary kernels for different images. So we testify the generalization ability of our model. The model is trained on Rain200H without any fine-tone process. As shown in Figure 10, our DDCDNet could capture the features of the diverse rain-layers and the extracted rain-layers contain fewer background information even the training/testing domain mismatch with each other.

4.4 Ablation Study

In this section, we demonstrate the ablation study in Table (1-3) to explore the effects of different elements in our DDCDNet. All the experiments are implemented on Rain200H. We perform our ablation study on three sections: dictionary learning strategies, stage number and loss function.

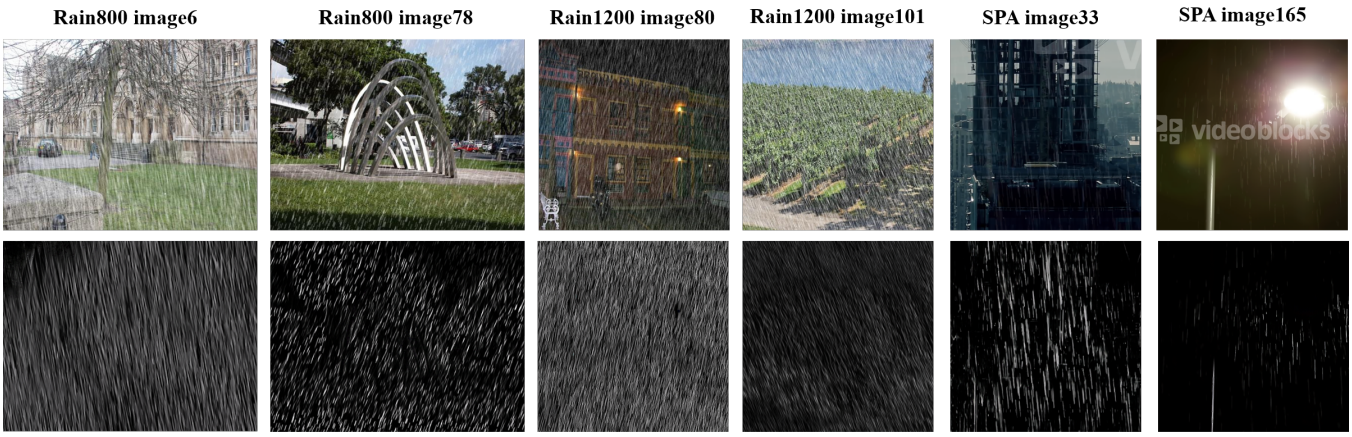

Figure 10: Visualization of generalization ability. All the samples are tested with the model trained on Rain200H data set.

Table 4: De-raining performance on Rain200H of the proposed DDCDNet with different loss functions. Noticed that when applying the progressive Multi-Stage loss, the α_i and β_i are set to $\frac{i}{T}$

Version	Name	Expression of Loss Function	PSNR/SSIM
1	L1	$L1(\mathbf{B^{gt}}, \mathbf{B^{(T)}})$	29.63/90.6
2	SSIM	$-SSIM(\mathbf{B^{gt}}, \mathbf{B^{(T)}})$	31.27/92.74
3	Multi-Stage L1	$\sum_{i=1}^{T} 0.1 \cdot L1(\mathbf{B^{gt}}, \mathbf{B^{(i)}}) + \sum_{i=1}^{T} 0.1 \cdot L1(\mathbf{O} - \mathbf{B^{gt}}, \mathbf{R^{(i)}})$	29.56/90.85
4	Multi-Stage SSIM	$-\sum_{i=1}^{T} 0.1 \cdot SSIM(\mathbf{B^{gt}}, \mathbf{B^{(i)}}) - \sum_{i=1}^{T} 0.1 \cdot SSIM(\mathbf{O} - \mathbf{B^{gt}}, \mathbf{R^{(i)}})$	31.3/92.81
5	Progressive Multi-Stage L1	$\sum_{i=1}^{T} \alpha_i \cdot L1(\mathbf{B^{gt}}, \mathbf{B^{(i)}}) + \sum_{i=1}^{T} \beta_i \cdot L1(\mathbf{O} - \mathbf{B^{gt}}, \mathbf{R^{(i)}})$	29.82/90.93
6	Progressive Multi-Stage SSIM	$-\sum_{i=1}^{T} \alpha_i \cdot SSIM(\mathbf{B^{gt}}, \mathbf{B^{(i)}}) - \sum_{i=1}^{T} \beta_i \cdot SSIM(\mathbf{O} - \mathbf{B^{gt}}, \mathbf{R^{(i)}})$	31.48/92.88

Table 5: Comparison of average PSNR and SSIM results on six common benchmark data sets. The 1^{st} and 2^{nd} are highlighted with boldface and underline.

Methods	Test12	Rain200H	Rain200L	Rain1200	Rain800	SPA-Data
DSC	30.07/0.8664	14.73/0.3815	27.16/0.8663	24.24/0.8279	22.61/0.7530	34.95/0.9416
GMM	32.14/0.9145	14.50/0.4164	28.66/0.8652	25.81/0.8344	25.71/0.8020	34.30/0.9428
JCAS	33.10/0.9302	14.69/0.4999	31.42/0.9173	25.16/0.8509	26.32/0.8375	34.95/0.9453
RESCAN	36.54/0.9568	26.75/0.8353	36.09/0.9697	33.38/0.9417	26.58/0.8726	38.11/0.9707
SPANet	35.92/0.9582	26.27/0.8666	35.79/0.9653	33.04/0.9489	27.81/0.8927	40.24/0.9811
DRDNet	36.27/0.9414	29.02/0.8849	35.19/0.9751	33.73/0.9232	28.21/0.901	38.72/0.9737
PReNet	36.61/0.9604	29.04/0.8991	37.80/0.9814	33.17/0.9481	27.06/0.9026	40.16/0.9816
JORDER-E	36.73/0.9634	29.35/0.8903	39.12/0.9840	34.05/0.9308	28.13/0.8996	40.78/0.9801
DCSFN	36.52/0.9599	29.46/0.8932	38.86/0.9840	34.19/0.9615	28.53/0.9034	40.98/0.9805
RCDNet	37.71/0.9649	30.24/0.9098	39.87/0.9875	34.08/0.9532	28.59/0.9137	41.47/0.9834
Multifocal	37.74/0.9712	30.17/0.9060	40.02/0.9870	34.89/0.9612	28.54/0.9082	-/-
VRGNet(JORDER+)	37.81/0.9716	30.73/0.9121	38.64/0.9843	34.24/0.9560	29.03/0.9134	42.54/0.9840
Ours	**37.86/0.9757**	**31.48/0.9288**	**40.08/0.9889**	**35.08/0.9645**	**29.57/0.9165**	**43.75/0.9872**

Different dictionary learning strategies. In order to emphasize the effectiveness of dual dictionary learning, we adopt single dictionary learning on **R** and **B**. As shown in Table 1, the dual dictionary will improve the PSNR index around 1.5dB. It should be noticed that when applying single dictionary learning on **R**, we perform an encoder-decoder structure with the same channel number we mentioned before to predict **B**. The single dictionary learning on **B** is demonstrated in the same way. We visualize an example in Figure 9 for demonstration. Meanwhile, to verify the efficacy of adpative dictionaries, we conduct two experiments to explore it. Results are shown in the Table 2.

Different stage number. We select the stage number from 1 to 5.

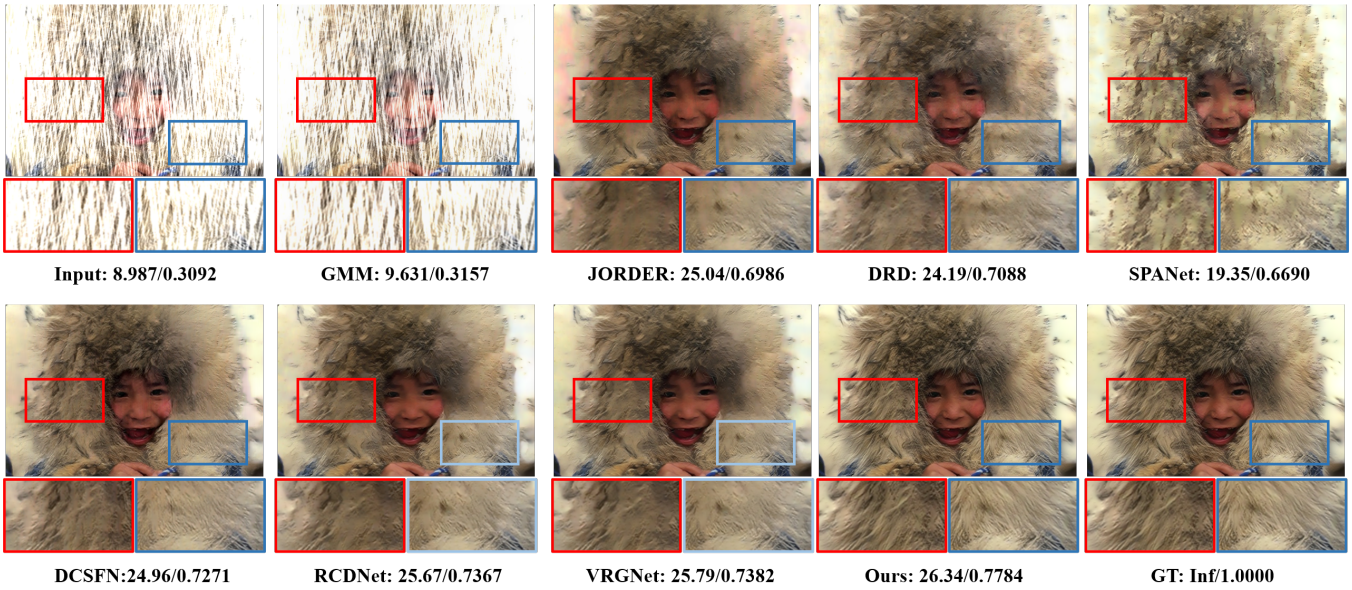

Figure 11: De-raining results on image 40 in Rain200H.

Figure 12: De-raining results on image 63 in Rain200H.

It can be seen in Table 3 that the PSNR and SSIM [39] indexes rise with the increase of stage number firstly. However, when the stage number is larger than 4, the performance decreases slightly. We adopt the stage number equal to 4 throughout the whole process since increasing the stage number would exert negative influence on gradient back-propagation.

Loss function. In this section, we demonstrate six different forms to further explore the proper loss function. As shown in Table 4, adopting the SSIM loss will increase the PSNR index for nearly 1.5dB compared with the L1 loss. After using the muti-stage and progressive multi-stage loss, the performance will increase 0.05dB and 0.2dB.

4.5 Experiments on Synthetic data sets

We evaluate our DDCDnet on five representative synthetic data sets: Test12, Rain200H, Rain200L, Rain800 and Rain1200. Those five data sets are gennerated by applying different strategies. The Test12 [26] only contains 12 light rain images. Rain200L and Rain200H have

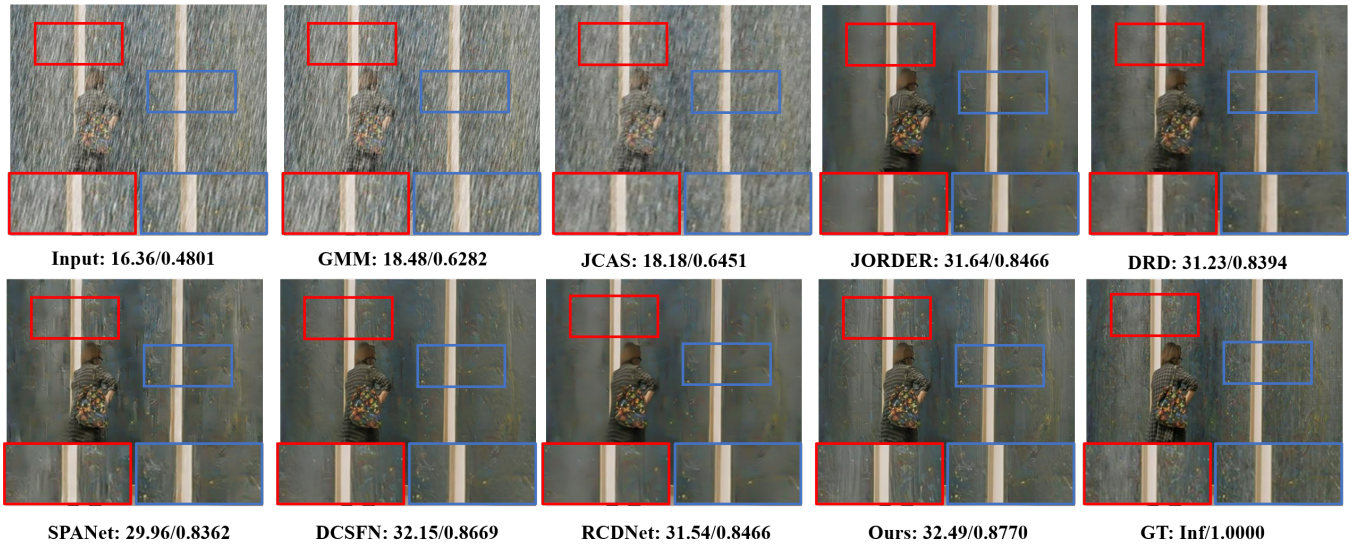

Figure 13: De-raining results on image 229 in Rain1200.

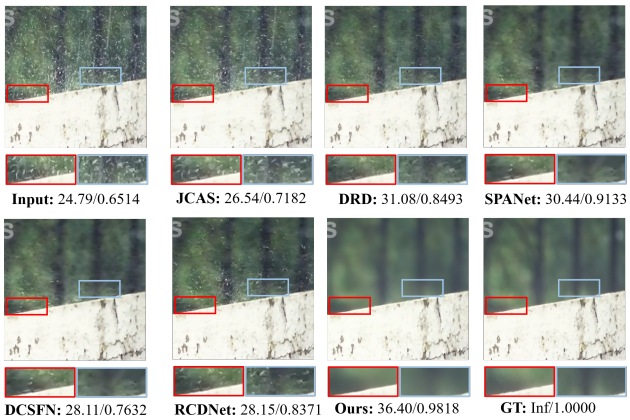

Figure 14: De-raining results on image 896 in SPA-Data. As shown in Figure 14, previous de-raining methods fail to remove rain streaks and rain-drops while our networks are able to generate rain-free image.

the same ground truth image. The Rain200L has light rain streaks while the Rain200H contains demanding extensive rain streaks [41]. Rain800 [45] contains 700 training pairs and 100 testing pairs. Rain1200 [43] includes 12000 training pairs and 1200 testing pairs. We use the indexes PSNR and SSIM [39] for the comparison with previous SOTA methods as evaluating indicator. We compute these two indicators on luminance (**Y**) channel [40]. As shown in Table 5, our model achieves the best overall results. For visualization, we use examples in Rain200H and Rain1200 for demonstration. For Rain200H, we select two changeling images, as shown in Figure 11 and Figure 12 , it is obvious that other de-raining methods generate over-smoothed image, while our DDCDNet could recover more detailed texture contents. For Rain1200, the de-raining results are shown in Figure 13, we could find that our model could retain more imprints on the blackboard compared with other methods. This is due to the addition of adaptive dynamic dictionary model.

4.6 Experiments on Real-World data set

To validate the peformance of DDCDNet in real-world situations, we also verify our model on SPA-Data [38]. The SPA-Data contains 638492 training pairs and 1000 testing pairs. As shown in Figure 14, our network can generate clearer background while retaining more detail textures. We also achieve best PSNR and SSIM [39] indexes as shown in the last column of Table 5. These results show that our DDCDNet achieves a balance between de-raining and texture preservation.

5 CONCLUSION

In this paper, we introduce a Deep Dual Convolutional Dictionary Learning Network for rain removal task. We propose two dictionary learning models to capture the rain-streaks layer and background layer separately. By applying the unfolding methods and HQS [15] algorithm, our network is able to generate rain-free image while retaining interpretability. These two dictionary learning models are further embedded with adaptive dictionaries, which enable our network to reconstruct more detailed texture information from rain images. Combined with these valid components, our DDCDNet achieves outstanding results on several common benchmarks compared with previous state-of-the-art de-raining methods.

REFERENCES

[1] Jonas Adler and Ozan Oktem. 2018. Learned Primal-Dual Reconstruction. *IEEE Transactions on Medical Imaging* 37, 6 (jun 2018), 1322–1332. https://doi.org/10.1109/tmi.2018.2799231

[2] Manya V. Afonso, José M. Bioucas-Dias, and Mário A. T. Figueiredo. 2010. Fast Image Recovery Using Variable Splitting and Constrained Optimization. *IEEE Transactions on Image Processing* 19, 9 (2010), 2345–2356. https://doi.org/10.1109/TIP.2010.2047910

[3] Michal Aharon, Michael Elad, and Alfred Bruckstein. 2006. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing* 54, 11 (2006), 4311–4322.

[4] Hilton Bristow, Anders Eriksson, and Simon Lucey. 2013. Fast convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 391–398.

[5] Xiangyong Cao, Feng Zhou, Lin Xu, Deyu Meng, Zongben Xu, and John Paisley. 2018. Hyperspectral Image Classification With Markov Random Fields and a Convolutional Neural Network. *IEEE Transactions on Image Processing* 27, 5 (2018), 2354–2367. https://doi.org/10.1109/TIP.2018.2799324

[6] Yunjin Chen and Thomas Pock. 2015. Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration. (08 2015). https://doi.org/10.13140/RG.2.1.2460.1446

[7] S. Deng, M. Wei, J. Wang, L. Liang, H. Xie, and M. Wang. 2019. DRD-Net: Detail-recovery Image Deraining via Context Aggregation Networks. (2019).

[8] Zhiwen Fan, Huafeng Wu, Xueyang Fu, Yue Huang, and Xinghao Ding. 2018. Residual-Guide Network for Single Image Deraining. 1751–1759. https://doi.org/10.1145/3240508.3240694

[9] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. 2017. Clearing the Skies: A Deep Network Architecture for Single-Image Rain Removal. *IEEE Transactions on Image Processing* 26, 6 (2017), 2944–2956. https://doi.org/10.1109/TIP.2017.2691802

[10] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. 2017. Removing Rain From Single Images via a Deep Detail Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[11] Xueyang Fu, Qi Qi, Yue Huang, Xinghao Ding, Feng Wu, and John Paisley. 2018. A Deep Tree-Structured Fusion Model for Single Image Deraining. (11 2018).

[12] Shuhang Gu, Deyu Meng, Wangmeng Zuo, and Lei Zhang. 2017. Joint Convolutional Analysis and Synthesis Sparse Representation for Single Image Layer Separation. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 1717–1725. https://doi.org/10.1109/ICCV.2017.189

[13] Chunle Guo, Chongyi Li, Jichang Guo, Chen Change Loy, Junhui Hou, Sam Kwong, and Runmin Cong. 2020. Zero-reference deep curve estimation for low-light image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1780–1789.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778. https://doi.org/10.1109/CVPR.2016.90

[15] Ran He, Wei-Shi Zheng, Tieniu Tan, and Zhenan Sun. 2014. Half-Quadratic-Based Iterative Minimization for Robust Sparse Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 2 (2014), 261–275. https://doi.org/10.1109/TPAMI.2013.102

[16] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu. 2012. Automatic Single-Image-Based Rain Streaks Removal via Image Decomposition. *IEEE Transactions on Image Processing* 21, 4 (2012), 1742–1755. https://doi.org/10.1109/TIP.2011.2179057

[17] Donghwan Kim and Jeffrey A. Fessler. 2018. Another Look at the Fast Iterative Shrinkage/Thresholding Algorithm (FISTA). *SIAM Journal on Optimization* 28, 1 (jan 2018), 223–250. https://doi.org/10.1137/16m108940x

[18] Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations* (12 2014).

[19] Filippos Kokkinos and Stamatios Lefkimmiatis. 2018. Deep image demosaicking using a cascade of convolutional residual denoising networks. In *Proceedings of the European conference on computer vision (ECCV)*. 303–319.

[20] Jakob Kruse, Carsten Rother, and Uwe Schmidt. 2017. Learning to Push the Limits of Efficient FFT-Based Image Deconvolution. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 4596–4604. https://doi.org/10.1109/ICCV.2017.491

[21] Stamatios Lefkimmiatis. 2017. Non-local color image denoising with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 3587–3596.

[22] Guanbin Li, Xiang He, Wei Zhang, Huiyou Chang, Le Dong, and Liang Lin. 2018. Non-locally Enhanced Encoder-Decoder Network for Single Image De-raining. In *ACM International Conference on Multimedia*.

[23] Guanbin Li, Xiang He, Wei Zhang, Huiyou Chang, Le Dong, and Liang Lin. 2018. Non-locally enhanced encoder-decoder network for single image de-raining. In *Proceedings of the 26th ACM international conference on Multimedia*. 1056–1064.

[24] Ruoteng Li, Loong-Fah Cheong, and Robby T Tan. 2019. Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1633–1642.

[25] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. 2018. Recurrent Squeeze-and-Excitation Context Aggregation Net for Single Image Deraining. In *European Conference on Computer Vision (ECCV)*. Springer, 262–277.

[26] Yu Li, Robby T. Tan, Xiaojie Guo, Jiangbo Lu, and Michael S. Brown. 2016. Rain Streak Removal Using Layer Priors. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2736–2744. https://doi.org/10.1109/CVPR.2016.299

[27] Yu Luo, Yong Xu, and Hui Ji. 2015. Removing Rain from a Single Image via Discriminative Sparse Coding. In *2015 IEEE International Conference on Computer Vision (ICCV)*. 3397–3405. https://doi.org/10.1109/ICCV.2015.388

[28] Jinshan Pan, Sifei Liu, Deqing Sun, Jiawei Zhang, Yang Liu, Jimmy Ren, Zechao Li, Jinhui Tang, Huchuan Lu, Yu-Wing Tai, et al. 2018. Learning dual convolutional neural networks for low-level vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3070–3079.

[29] Xu Qin, Zhilin Wang, Yuanchao Bai, Xiaodong Xie, and Huizhu Jia. 2020. FFA-Net: Feature fusion attention network for single image dehazing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 11908–11915.

[30] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. 2019. Progressive image deraining networks: A better and simpler baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3937–3946.

[31] Weihong Ren, Jiandong Tian, Zhi Han, Antoni Chan, and Yandong Tang. 2017. Video Desnowing and Deraining Based on Matrix Decomposition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2838–2847. https://doi.org/10.1109/CVPR.2017.303

[32] Kegan GG Samuel and Marshall F Tappen. 2009. Learning optimized MAP estimates in continuously-valued MRF models. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 477–484.

[33] Uwe Schmidt and Stefan Roth. 2014. Shrinkage Fields for Effective Image Restoration. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2774–2781. https://doi.org/10.1109/CVPR.2014.349

[34] Jian Sun and Marshall F Tappen. 2011. Learning non-local range Markov random field for image restoration. In *CVPR 2011*. IEEE, 2745–2752.

[35] Cong Wang, Xiaoying Xing, Yutong Wu, Zhixun Su, and Junyang Chen. 2020. DCSFN: Deep Cross-scale Fusion Network for Single Image Rain Removal. In *ACM International Conference on Multimedia*.

[36] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. 2020. A Model-Driven Deep Neural Network for Single Image Rain Removal. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[37] Hong Wang, Zongsheng Yue, Qi Xie, Qian Zhao, Yefeng Zheng, and Deyu Meng. 2021. From Rain Generation to Rain Removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 14791–14801.

[38] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson W.H. Lau. 2019. Spatial Attentive Single-Image Deraining with a High Quality Real Rain Dataset. In *The IEEE Conference on Computer Vision and Pattern Recognition*.

[39] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612. https://doi.org/10.1109/TIP.2003.819861

[40] Wenhan Yang, Robby Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. 2016. Joint Rain Detection and Removal via Iterative Region Dependent Multi-Task Learning. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[41] Wenhan Yang, Robby T. Tan, Jiashi Feng, Zongming Guo, Shuicheng Yan, and Jiaying Liu. 2020. Joint Rain Detection and Removal from a Single Image with Contextualized Deep Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 6 (2020), 1377–1393. https://doi.org/10.1109/TPAMI.2019.2895793

[42] He Zhang and Vishal M Patel. 2018. Densely Connected Pyramid Dehazing Network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

[44] He Zhang, Vishwanath Sindagi, and Vishal M Patel. 2019. Image de-raining using a conditional generative adversarial network. *IEEE transactions on circuits and systems for video technology* 30, 11 (2019), 3943–3956.

[45] He Zhang, Vishwanath Sindagi, and Vishal M Patel. 2019. Image de-raining using a conditional generative adversarial network. *IEEE transactions on circuits and systems for video technology* 30, 11 (2019), 3943–3956.

[46] Kai Zhang, Luc Van Gool, and Radu Timofte. 2020. Deep unfolding network for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3217–3226.

[47] Zheyu Zhang, Yurui Zhu, Xueyang Fu, Zhiwei Xiong, Zhengjun Zha, and Feng Wu. 2021. Multifocal Attention-Based Cross-Scale Network for Image De-raining. In *ACM International Conference on Multimedia*.

[48] Hongyi Zheng, Hongwei Yong, and Lei Zhang. 2021. Deep convolutional dictionary learning for image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 630–641.