

Rain Streak Removal via Dual Graph Convolutional Network

Xueyang Fu^{1†}, Qi Qi^{2†}, Zheng-Jun Zha^{1*}, Yurui Zhu¹, Xinghao Ding²

¹University of Science and Technology of China, China

²Xiamen University, China

xyfu@ustc.edu.cn, qqxmu@stu.xmu.edu.cn, zhazj@ustc.edu.cn, zyr@mail.ustc.edu.cn, dxh@xmu.edu.cn

Abstract

Deep convolutional neural networks (CNNs) have become dominant in the single image de-raining area. However, most deep CNNs-based de-raining methods are designed by stacking vanilla convolutional layers, which can only be used to model local relations. Therefore, long-range contextual information is rarely considered for this specific task. To address the above problem, we propose a simple yet effective dual graph convolutional network (GCN) for single image rain removal. Specifically, we design two graphs to perform global relational modeling and reasoning. The first GCN is used to explore global spatial relations among pixels in feature maps, while the second GCN models the global relations across the channels. Compared to standard convolutional operations, the proposed two graphs enable the network to extract representations from new dimensions. To achieve the image rain removal, we further embed these two graphs and multi-scale dilated convolution into a symmetrically skip-connected network architecture. Therefore, our dual graph convolutional network is able to well handle complex and spatially long rain streaks by exploring multiple representations, e.g., multi-scale local feature, global spatial coherence and cross-channel correlation. Meanwhile, our model is easy to implement, end-to-end trainable and computationally efficient. Extensive experiments on synthetic and real data demonstrate that our method achieves significant improvements over the recent state-of-the-art methods.

Introduction

Rain can severely impair the performance of many computer vision systems, e.g., road surveillance, autonomous driving and consumer camera. Effectively removing rain streaks from images is an important task in the computer vision community. To address the de-raining problem, many algorithms have been designed to remove rain streaks from single rainy images. Unlike video based methods (Garg and Nayar 2007; Barnum, Narasimhan, and Kanade 2010; Ren

et al. 2017; Wei et al. 2017; Li et al. 2018b; Yang et al. 2020b), which have useful temporal information, single image de-raining is a significantly harder problem (Li et al. 2019; Wang et al. 2020a; Yang et al. 2020a). Furthermore, since success in single images can be extended to video, single image de-raining has received much research attention.

Related Work

In general, single image de-raining methods can be categorized into two classes: model-driven and data-driven. Model-driven methods are designed by using handcrafted image features to describe physical characteristics of rain streaks, or exploring prior knowledge to constrain the ill-posed problem. In (Kim et al. 2013), the de-rained image is obtained by filtering a rainy image with a nonlocal mean smoothing filter. Several model-driven methods adopt various priors to separate rain streaks from rainy images. For example, in (Kang, Lin, and Fu 2012), morphological component analysis based dictionary learning is used to remove rain streaks in high frequency regions. To recognize rain streaks, a self-learning based image decomposition method is introduced in (Huang et al. 2014). In (Luo, Xu, and Ji 2015), based on image patches, a discriminative sparse coding is proposed to distinguish rain streaks from non-rain content. In (Chen and Hsu 2013; Chang, Yan, and Zhong 2017), low-rank assumptions are used to model and separate rain streaks. A GMM based patch prior (Li et al. 2016) is introduced to accommodate multiple orientations and scales of rain streaks. In (Wang et al. 2017), the authors use a hierarchical scheme combined with dictionary learning to progressively remove rain and snow. In (Gu et al. 2017), the authors utilize convolutional analysis and synthesis sparse representation to extract rain streaks. In (Zhu et al. 2017), three priors are explored and combined into a joint optimization process for single image rain removal.

Recently, data-driven methods using deep learning have dominated high-level vision tasks (He et al. 2016; Huang et al. 2017; Cao et al. 2018) and low-level image processing (Eigen, Krishnan, and Fergus 2013; Ren et al. 2016; Zhang et al. 2018a,b; Zhang and Patel 2018a; Qin et al. 2020; Guo et al. 2020; Ren et al. 2020). In (Fu et al. 2017a), the authors use domain knowledge and train a 3 layers network on high-frequency parts to simplify the learning processing. This method was improved in (Fu et al. 2017b) by combin-

[†]Co-first authors contributed equally. *Corresponding author.

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 61901433, U19B2038 and 61620106009, the USTC Research Funds of the Double First-Class Initiative under Grant YD2100002003.

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

ing ResNet (He et al. 2016) and a global skip connection. Other methods focus on designing advanced network structure to improve de-raining performance. In (Yang et al. 2017, 2019), a recurrent dilated network with multi-task learning is proposed for joint rain streaks detection and removal. In (Li et al. 2018c), the recurrent neural network architecture is adopted and combined with squeeze-and-excitation (SE) blocks (Hu, Shen, and Sun 2018) for rain removal. To incorporate discriminative performance into the loss function, the generative adversarial networks (Goodfellow et al. 2014) is utilized in (Zhang, Sindagi, and Patel 2019) to improve visual quality. In (Zhang and Patel 2018b), a density aware multi-stream dense CNN is introduced to automatically determine the rain-density information. To capture global spatial dependencies for accurate rain streaks estimation, non-local operations are utilized to design end-to-end de-raining networks (Li et al. 2018a; Yu et al. 2019). To handle heavy rainy images with hazy effect, the authors of (Li, Cheong, and Tan 2019) design a deep network based on the physical model and insert auxiliary losses to train it. Similar to (Li, Cheong, and Tan 2019), the authors of (Hu et al. 2019) formulate the rain imaging process based on scene depth, and then introduce a depth-guided attention mechanism to handle heavy rain streaks. To take the location information of rain drops into consideration, an uncertainty guided multi-scale residual learning network is proposed in (Yasarla and Patel 2019) to learn the rain content at different scales. In (Wei et al. 2019), a semi-supervised learning paradigm is proposed to improve the generalization ability for unseen rain types. By unfolding a shallow ResNet repeatedly, a simple yet effective network architecture is proposed by (Ren et al. 2019) with progressive recurrent operations. In (Wang et al. 2019), a spatial attentive network is introduced to remove rain streaks in a local-to-global fashion. In (Zhu et al. 2019), a training strategy using unpaired data is proposed to achieve rain removal. Recently, in (Yang et al. 2020c), a fractal band learning network with cross-scale self-supervision is proposed to extract scale-robust rain features. In (Deng et al. 2020), a two-stage context aggregation network architecture is designed to well restore details. In (Yasarla, Sindagi, and Patel 2020), based on the Gaussian process, a semi-supervised learning for image de-raining is introduced. In (Wang et al. 2020b), a novel interpretable network is introduced to fully integrate conventional convolutional dictionary-based methods and deep learning.

Our Contributions

Despite the noticeable progress has been made in single image rain removal, most existing deep models adopt convolutional neural networks (CNNs) as backbones. While vanilla CNNs are only capable to model local spatial information, long-range dependencies for contextual modeling are ignored. To obtain larger receptive fields, several methods (Li et al. 2018c; Yang et al. 2019) adopt the dilated convolution (Yu and Koltun 2016). However, since the convolution operation is essentially a process of local weighted summation, what we get from dilated convolution is still local spatial information, i.e., obtaining one pixel value from one finite spatial area. Another direction is to combine C-

NNs and recurrent neural networks (RNNs) (Li et al. 2018c; Ren et al. 2019). Due to the repeated features utilization, the de-raining performance can be boosted. However, these methods focus on propagating spatial information without considering the correlation between channels. Therefore, for single image de-raining, the aspect of fully reasoning global spatial coherence and channel correlation has not been noticed. On the other hand, GCN (Kipf and Welling 2017) was proposed to effectively model long-range contextual information and has been used for various high-level vision tasks (Johnson, Gupta, and Fei-Fei 2018; Li et al. 2018d; Zha et al. 2020; Zhu et al. 2020). For image and video, the most widely used form of GCNs is the non-local network (Wang et al. 2018). However, directly implementing non-local operation requires a huge memory cost (Li et al. 2018a; Yu et al. 2019), which limits its practical values. In addition, there are still few methods to explore the potential value of GCNs for the image de-raining.

To address the above limitations, we propose a dual graph convolutional network to explore multi-dimension information for the tough image de-raining. Specifically, we first design two lightweight graphs to extract relation-aware features. The spatial GCN is derived from the non-local operation (Wang et al. 2018) and used to explore global spatial relationships between pixels. The channel GCN is designed to explore global interdependencies across channels of the feature map. Then, these two modules are combined with the dialed convolution to form the basic unit, which is further embedded in a symmetrically skip-connected architecture as the final de-raining network. In this way, our proposed network is able to extract rich representations from different dimensions. For example, utilizing dialed convolutional networks and spatial GCN can explore multi-scale local features and global spatial information. While the channel GCN can capture correlations among feature maps. Moreover, since the proposed spatial-wise graph and channel-wise graph are orthogonal, these two modules can provide complementary information. This makes our networks generate better task-specific feature representations to help boost the de-raining performance.

Our contributions are three-fold:

- We propose a dual graph convolutional network for single image rain removal. Our method integrates both local and global modeling into a single network, which allows it to explore both local spatial patterns and global contextual information.
- We propose two orthogonal graphs that can efficiently compute spatial coherence and channel correlation. By combining the two graphs and multi-scale dilated convolutions, we present a basic unit to form the de-raining network. This basic unit is able to extract relation-aware information for learning better task-specific representations.
- Our network has advantages of easy implementation, end-to-end training and efficient calculation. Experiments demonstrate that our model favorably performs against the state-of-the-art methods on both synthetic and real-world data sets.

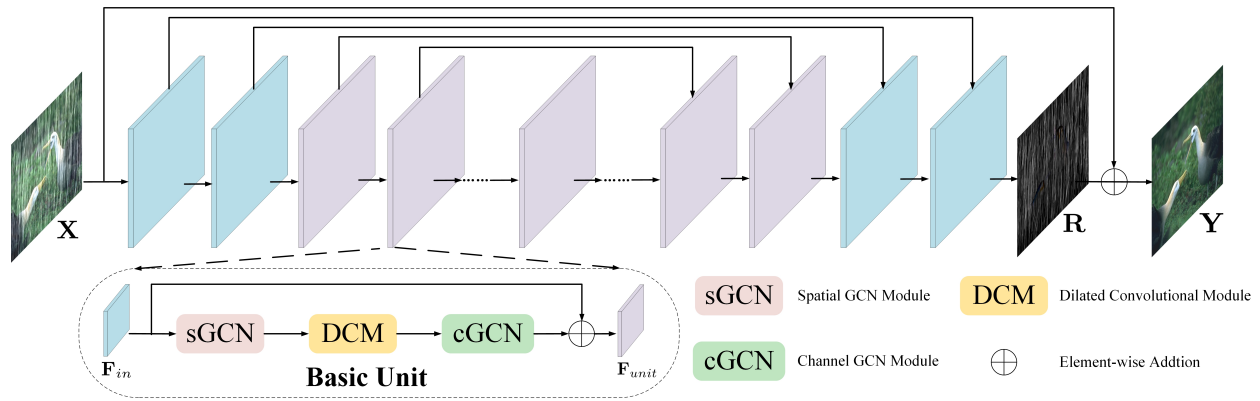


Figure 1: The overall architecture of our dual graph convolutional network for single image rain removal. The network consists of several basic units, in which multi-scale dilated convolutional modules and global GCN modules are adopted to capture both local and global information. Symmetrical skip-layer connections are deployed to pass shallow features to deep layers.

Methodology

Figure 1 illustrates the overall architecture of our dual graph convolutional network. To summarize at a high level, we design a basic unit contain multi-scale dilated convolutions with two graph modules. This unit is further embedded into a symmetrically skip-connected network architecture, which is able to pass image details from shallow layers to deep layers to benefit image restoration tasks (Mao, Shen, and Yang 2016). Below we detail our proposed method.

Overall Network Architecture

We design our dual graph convolutional network based on symmetrically skip-connected network architecture, and it takes a single rainy image \mathbf{X} as input and predicts its de-rained version \mathbf{Y} . Our network contains two feature extraction layers, several basic units and two reconstruction layers. The feature extraction layers are designed to extract shallow features from the rainy input using standard 3×3 convolutional operations. As shown in Figure 1, these two shallow features are further propagated to deeper layers using skip-connections to preserve raw information. The reconstruction layers are used to generate the de-rained image \mathbf{Y} by

$$\mathbf{Y} = \mathbf{X} + \mathbf{R} = \mathbf{X} + f(\mathbf{X}). \quad (1)$$

Note that the direct output of our network $f(\cdot)$ is the residual \mathbf{R} , which is a common technique used in existing methods (Wei et al. 2019; Zhang et al. 2020) to ease learning. The basic units, which are constructed by our proposed modules, will be detailed in subsequent sections.

Proposed Modules

For the specific image rain removal task, we propose one dilated convolutional module and two GCN modules to learn representations from multiple dimensions.

Dilated Convolutional Module. In general, stacking vanilla convolutional layers with a 3×3 kernel size can gradually increase receptive fields. However, rain streaks and object structures are spatially long, which requires larger receptive fields. Therefore, we utilize the dilated convolution

(Yu and Koltun 2016) to rapidly increase receptive fields. Dilated convolutions is able to increase the contextual area while preserving resolution and reducing parameters burden. The features obtained by dilated convolutions are defined as

$$\mathbf{F}_{DF} = \mathbf{K}_{DF} * \mathbf{F}_{in} + \mathbf{b}, \quad (2)$$

where DF is the dilation factor, \mathbf{K} is the convolutional kernel, \mathbf{F}_{in} is the input feature, \mathbf{F}_{DF} is the output feature of convolution with DF .

To obtain multi-scale spatial features, we further design a parallel structure as shown in Figure 2. Specifically, we design two paths in each dilated module. One path consists of two standard convolutional layers to capture small-scale spatial patterns, while the other one contains two dilated convolutional layers to rapidly increase the receptive field. The output of this module is generated by fusing the five features using one 1×1 convolutional operation. In this way, the fused features $\hat{\mathbf{F}}_{DCM}$ contains information of different receptive fields, i.e., 3×3 , 5×5 , 7×7 and 13×13 . This enables the dilated module to effectively extract multi-scale local spatial features. Although this module can obtain multi-scale local spatial representations, the information contained in the fused features is still from a local spatial region. Therefore, we further propose two global GCN modules to learn representations of global spatial coherence and channel correlation, respectively.

Spatial GCN Module. We first design a spatial GCN module to model the global spatial coherence. The motivation of this module is to explore the relation between one pixel and all pixels in the feature map. Let a feature map be $\mathbf{F} \in \mathbb{R}^{HW \times N}$, where N is the number of channel, H and W are the height and width of \mathbf{F} , respectively. The graph convolution is defined as (Kipf and Welling 2017)

$$\mathbf{F}_{GCN} = \mathbf{A}\mathbf{F}\mathbf{W}, \quad (3)$$

where \mathbf{A} is the adjacency matrix and \mathbf{W} is the weight matrix. Similar with the non-local network (Wang et al. 2018), we use three 1×1 convolution layers, $\theta(\cdot)$, $\nu(\cdot)$ and $\xi(\cdot)$, on the input feature map to reduce the channel number from N

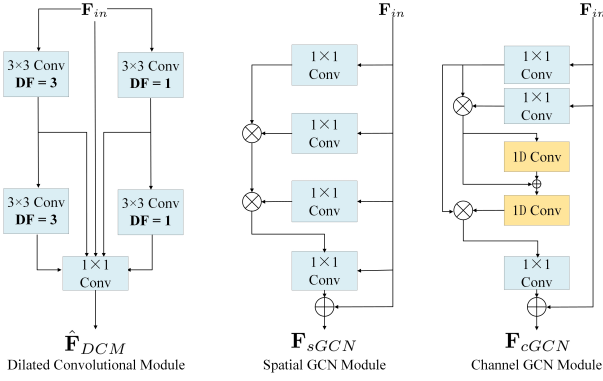


Figure 2: Architectures of our proposed modules which are used to form the basic unit. The dilated convolutional module aims to capture multi-scale local feature representations. The spatial and channel GCN modules focus on exploring global contextual information along two orthogonal dimensions. \otimes denotes matrix multiplication.

to $\frac{N}{2}$. As shown in Figure 2, the new feature is defined in the form of residual learning

$$\begin{aligned} \mathbf{F}_{sGCN} &= \mathbf{F}_{in} + \mathbf{A}_{sGCN} \mathbf{F}_s \mathbf{W}_{sGCN} \\ &= \mathbf{F}_{in} + \theta(\mathbf{F}_{in}) \nu(\mathbf{F}_{in})^\top \xi(\mathbf{F}_{in}) \mathbf{W}_{sGCN}, \end{aligned} \quad (4)$$

where \mathbf{F}_{sGCN} is output of spatial GCN module, \top is the transpose operation. $\theta(\cdot) \nu(\cdot)^\top$ is performed by matrix multiplication and can be seen as the adjacency matrix \mathbf{A}_{sGCN} . Note that the term $(\theta(\cdot) \nu(\cdot)^\top) \xi(\cdot)$ can be re-ordered to $\theta(\cdot) (\nu(\cdot)^\top \xi(\cdot))$ according to the associative rule. Compared with the generic non-local module (Wang et al. 2018) which calculates a large measurement matrix with size of $HW \times HW$, using the re-ordering can significantly reduce computation complexity from $O((HW)^2)$ to $O(HW)$. As shown in Figure 3, our spatial GCN module is nearly four times faster on both CPU and GPU than the non-local module of NLEDN (Li et al. 2018a). We use *softmax* operations to avoid numerical instabilities (Chen et al. 2018). The weighting process of \mathbf{W}_{GSM} is conducted by using one 1×1 convolution layer to perform a hidden-to-output operation. This spatial GCN module allows the network to produce coherent predictions that consider all pixels, which benefits for extracting information about spatially long rain streaks.

Channel GCN Module. In addition to exploring global spatial coherence, we also design a channel GCN module to reason channel correlations of the feature map. We model our channel GCN module as

$$\begin{aligned} \mathbf{F}_{cGCN} &= \mathbf{F}_{in} + \phi(\hat{\mathbf{F}}_c) \\ &= \mathbf{F}_{in} + \phi(\mathbf{A}_{cGCN} \mathbf{F}_c \mathbf{W}_{cGCN}), \end{aligned} \quad (5)$$

where \mathbf{A}_{cGCN} is the adjacency matrix measuring the relations of the graph, and \mathbf{W}_{cGCN} is the weight matrix.

In practice, to aggregate information from different channels, we adopt two 1×1 convolutions $\kappa(\cdot)$ and $\zeta(\cdot)$ on the input feature \mathbf{F}_{in} , where $\kappa(\mathbf{F}_{in}) \in \mathbb{R}^{HW \times \frac{N}{4}}$ and $\zeta(\mathbf{F}_{in}) \in \mathbb{R}^{HW \times \frac{N}{2}}$. Then a new feature \mathbf{F}_c that represents the channel

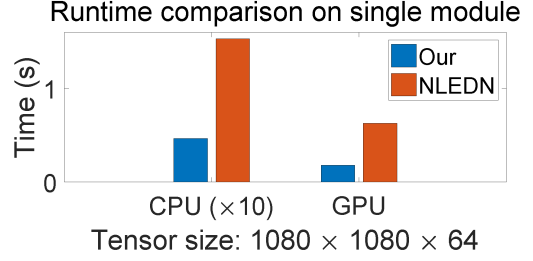


Figure 3: Runtime comparison between NLEDN non-local module (Li et al. 2018a) and our spatial GCN module. Note that the measurement unit of CPU is 10 times that of GPU.

correlation is given by

$$\mathbf{F}_c = \text{softmax}(\kappa(\mathbf{F}_{in})^\top \zeta(\mathbf{F}_{in})). \quad (6)$$

From the view of graph, the size of feature \mathbf{F}_c is $\frac{N}{4} \times \frac{N}{2}$, which means it contains $\frac{N}{4}$ nodes, and the dimension of each node is $\frac{N}{2}$. For this module, we construct a fully-connected graph with the adjacency matrix $\mathbf{A}_{cGCN} \in \mathbb{R}^{\frac{N}{4} \times \frac{N}{4}}$ and the weights $\mathbf{W}_{cGCN} \in \mathbb{R}^{\frac{N}{2} \times \frac{N}{2}}$ on the new feature \mathbf{F}_c to reason the channel correlation. As shown in Figure 2, the calculation of this module is defined by

$$\begin{aligned} \hat{\mathbf{F}}_c &= (\mathbf{I} + \mathbf{A}_{cGCN}) \mathbf{F}_c \mathbf{W}_{cGCN} \\ &= (\mathbf{I} + \mathbf{A}_{cGCN}) (\kappa(\mathbf{F}_{in})^\top \zeta(\mathbf{F}_{in})) \mathbf{W}_{cGCN}, \end{aligned} \quad (7)$$

where the adjacency matrix \mathbf{A}_{cGCN} and the weight \mathbf{W}_{cGCN} are implemented by 1D convolutions and learned from data. We follow (Chen et al. 2019) and utilize identity matrix \mathbf{I} to propagate the nodes, which is similar with the Laplacian regularization (Zhang et al. 2014).

Since the size of generated graph $\hat{\mathbf{F}}_c$ is $\frac{N}{4} \times \frac{N}{2}$, we add one function $\phi(\cdot)$ to perform the hidden-to-output operation. Specifically, we first multiply $\zeta(\mathbf{F}_{in})$ by the generated graph $\hat{\mathbf{F}}_c$ and then utilize one 1×1 convolutional layer to increase the number of channel to N . In this way, the size of the output \mathbf{F}_{cGCN} is $HW \times N$, so that it can participate in subsequent operations. By deploying the proposed channel GCN module into the basic unit, our model is allowed to capture correlations among channels of the feature map.

Dual Graph Convolutional Network

Based on the proposed modules, we construct the basic unit to form our dual graph convolutional network for single image rain removal as shown in Figure 1. Specifically, in each unit, we first deploy the spatial GCN module to capture global spatial information from previous unit. Then, these global spatial information is sent into the dilated convolutional module to assist it to extract multi-scale local spatial features. Finally, to obtain content information that is complementary to spatial information, we employ the channel GCN module to explore the correlation among the features that contain rich global and local spatial representations. Based on the above description, the process of our basic unit is

$$\mathbf{F}_{unit} = \mathbf{F}_{in} + cGCN(DCM(sGCN(\mathbf{F}_{in}))), \quad (8)$$

where $sGCN(\cdot)$, $DCM(\cdot)$ and $cGCN(\cdot)$ denote spatial GCN module, dilated convolutional module and channel GCN module, respectively. We utilize 11 basic units to construct the de-raining network. In addition, we adopt symmetric skip-connections to link shallow and deep layers. This can not only avoid the gradient vanishing, but also propagate image detail to improve the de-raining performance.

Loss Function

The most widely used loss function for training a network is mean squared error (MSE). However, MSE usually generates over-smoothed results because of its ℓ_2 penalty. To address this drawback, we adopt the mean absolute error (MAE) to balance rain removal and detail preservation

$$\mathcal{L} = \frac{1}{M} \sum_{i=1}^M \|\mathbf{Y}_i - \mathbf{Y}_{gt,i}\|_1, \quad (9)$$

where M is the number of training data, \mathbf{Y} and \mathbf{Y}_{gt} denote the output de-rained image and ground truth, respectively.

Implementation Details

We set the sizes of the kernels in fusion operations and GCN modules to 1×1 and the rest to 3×3 . The number of feature maps is 72 for all convolutions. The non-linear activation is ReLU (Krizhevsky, Sutskever, and Hinton 2012) and used in the dilated convolutional module. We use TensorFlow (Abadi et al. 2016) and Adam (Kingma and Ba 2014) with a mini-batch size of 10 to train our network. The training images are cropped into 100×100 patch pairs with horizontal flipping for data augmentation. We fix the learning rate to 0.0001 and terminate training after 300 epochs.

Experiments

Baseline Methods. We compare our network with three model-based methods: DSC (Luo, Xu, and Ji 2015), GMM (Li et al. 2016), JCAS (Gu et al. 2017), and eleven deep learning-based methods: DDN (Fu et al. 2017b), DID-MDN (Zhang and Patel 2018b), RESCAN (Li et al. 2018c), N-LEDN (Li et al. 2018a), JORDER-E (Yang et al. 2019), IDC-GAN (Zhang, Sindagi, and Patel 2019), SIRR (Wei et al. 2019), PReNet (Ren et al. 2019), SPANet (Wang et al. 2019), FBL (Yang et al. 2020c) and RCDNet (Wang et al. 2020b).

Synthetic Data

We use five representative synthetic data sets provided by GMM (Li et al. 2016), JORDER-E (Yang et al. 2019), DDN (Fu et al. 2017b) and DID-MDN (Zhang and Patel 2018b), respectively. These five data sets were generated using different synthetic strategies. The GMM data set contains 12 light rainy images. The JORDER-E dataset contains two subsets, each with 200 testing images, which have light rain streaks and challenging heavy rain streaks, respectively. The rest two data sets contain 1400 and 1200 testing images, respectively. We denote them as *Rain12*, *Rain200L*, *Rain200H*, *DDN-Data* and *DID-Data*.

We calculate PSNR and SSIM (Wang et al. 2004) for the quantitative evaluation. 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 (Y) channel (Yang et al. 2017). As shown in columns of Table 1, our method has the best overall results in terms of PSNR and SSIM on the five synthetic data sets.

We also show three visual results with different rain appearances in Figure 4. For the challenging *Rain200H* data set, it is clear that JCAS fails to process heavy rain streaks due to modeling limitations. Other deep learning-based methods except NLEDN and RCDNet are able to remove the rain streaks while tending to over-smooth the results. In addition, our method has a better detail recovery than NLEDN and RCDNet. For the relatively light rainy data sets *DDN-Data* and *DID-Data*, all methods generate de-rained results with close visual qualities. While our method is able to generate clearer object content, e.g., the cloth in the bottom left corner and the rope in the upper right corner. This is because our method also considers the channel relations to extract informative content. These visual comparisons demonstrate that our dual GCN achieves a good trade-off between rain removal and content preservation.

Real-world Data

To test the performance of our method in real scenarios, we conduct experiments on the recent public real-world rainy data set *SPA-Data* (Wang et al. 2019), which contains nearly 0.64 million rainy/clean image pairs for training and 1000 pairs for testing. Figure 5 shows one visual result and the rectangles indicate that our network can remove rain streaks while retaining more detailed information. Since this real-world data set contains ground truth, we can quantitatively evaluate the de-rained performance. As shown in the last row of Table 1, our method consistently achieves the best results.

To demonstrate the generalization ability of our method, we also conduct experiments on the real-world data set (Wang et al. 2019), which is different from synthetic data and collected from Internet without ground truth. As shown in Figure 6, when dealing with light rain streaks in real-world scenarios, other comparison methods can remove apparent rain streaks while blur some image content and textures. On the contrary, our network can simultaneously achieve rain streaks removal and details preservation. This experiment shows that our learned network can well generalize to unseen real-world data types.

Runtime and Parameter Numbers

Our network is able to achieve a good trade-off between calculation and storage. We use 100 images with a size of 1000×1000 for evaluation. Comparisons on average runtime and parameter numbers are shown in Table 2. It is clear that our method has a comparable GPU runtime compared with other deep learning methods, and is significantly faster than several deep models on a CPU. This is because our network can extract more effective representations, which leads to promising results with tolerable resource consumption.

Ablation Study

Since the two GCN modules are the core of our method, to demonstrate their necessity and effectiveness, we provide

Methods	Data sets		<i>Rain12</i>		<i>Rain200L</i>		<i>Rain200H</i>		<i>DID-Data</i>		<i>DDN-Data</i>		<i>SPA-Data</i>	
DSC (Luo, Xu, and Ji 2015)	30.07	0.8664	27.16	0.8663	14.73	0.3815	24.24	0.8279	27.31	0.8373	34.95	0.9416		
GMM (Li et al. 2016)	32.14	0.9145	28.66	0.8652	14.50	0.4164	25.81	0.8344	27.55	0.8479	34.30	0.9428		
JCAS (Gu et al. 2017)	33.10	0.9302	31.42	0.9173	14.69	0.4999	25.16	0.8509	26.81	0.8632	34.95	0.9453		
DDN (Fu et al. 2017b)	35.74	0.9514	34.68	0.9671	26.05	0.8056	30.97	0.9116	30.00	0.9041	36.16	0.9457		
DID-MDN (Zhang and Patel 2018b)	36.25	0.9562	35.40	0.9618	26.61	0.8242	31.30	0.9207	31.49	0.9146	38.16	0.9763		
RESCAN (Li et al. 2018c)	36.54	0.9568	36.09	0.9697	26.75	0.8353	33.38	0.9417	31.94	0.9345	38.11	0.9707		
NLEDN (Li et al. 2018a)	37.13	0.9614	39.13	0.9821	29.79	0.9005	34.68	0.9583	32.15	0.9398	42.97	0.9835		
JORDER-E (Yang et al. 2019)	36.73	0.9634	37.25	0.9752	29.35	0.8905	33.98	0.9502	32.01	0.9321	40.78	0.9801		
ID-CGAN (Zhang, Sindagi, and Patel 2019)	35.97	0.9543	35.19	0.9694	25.02	0.8430	30.25	0.9217	29.06	0.9162	38.47	0.9624		
SIRR (Wei et al. 2019)	35.71	0.9501	34.75	0.9690	26.55	0.8190	30.57	0.9104	30.01	0.9078	35.31	0.9411		
PreNet (Ren et al. 2019)	36.61	0.9604	37.80	0.9814	29.04	0.8991	33.17	0.9481	32.60	0.9459	40.16	0.9816		
SPANet (Wang et al. 2019)	35.92	0.9582	35.79	0.9653	26.27	0.8666	33.04	0.9489	29.85	0.9117	40.24	0.9811		
FBL (Yang et al. 2020c)	<u>37.86</u>	0.9612	39.02	0.9827	30.07	0.9021	34.26	0.9320	33.05	0.9334	42.80	0.9824		
RCDNet (Wang et al. 2020b)	37.71	0.9649	<u>39.17</u>	<u>0.9885</u>	<u>30.24</u>	<u>0.9048</u>	34.08	0.9532	<u>33.04</u>	<u>0.9472</u>	<u>43.36</u>	0.9831		
Ours	38.99	0.9703	40.73	0.9886	31.15	0.9125	<u>34.37</u>	0.9620	33.01	0.9489	44.18	0.9902		
ideal value		+∞	1.0000	+∞	1.0000	+∞	1.0000	+∞	1.0000	+∞	1.0000	+∞	1.0000	

Table 1: Comparison of average PSNR | SSIM values on six benchmark data sets. The best and the second best results are **boldfaced** and underlined.

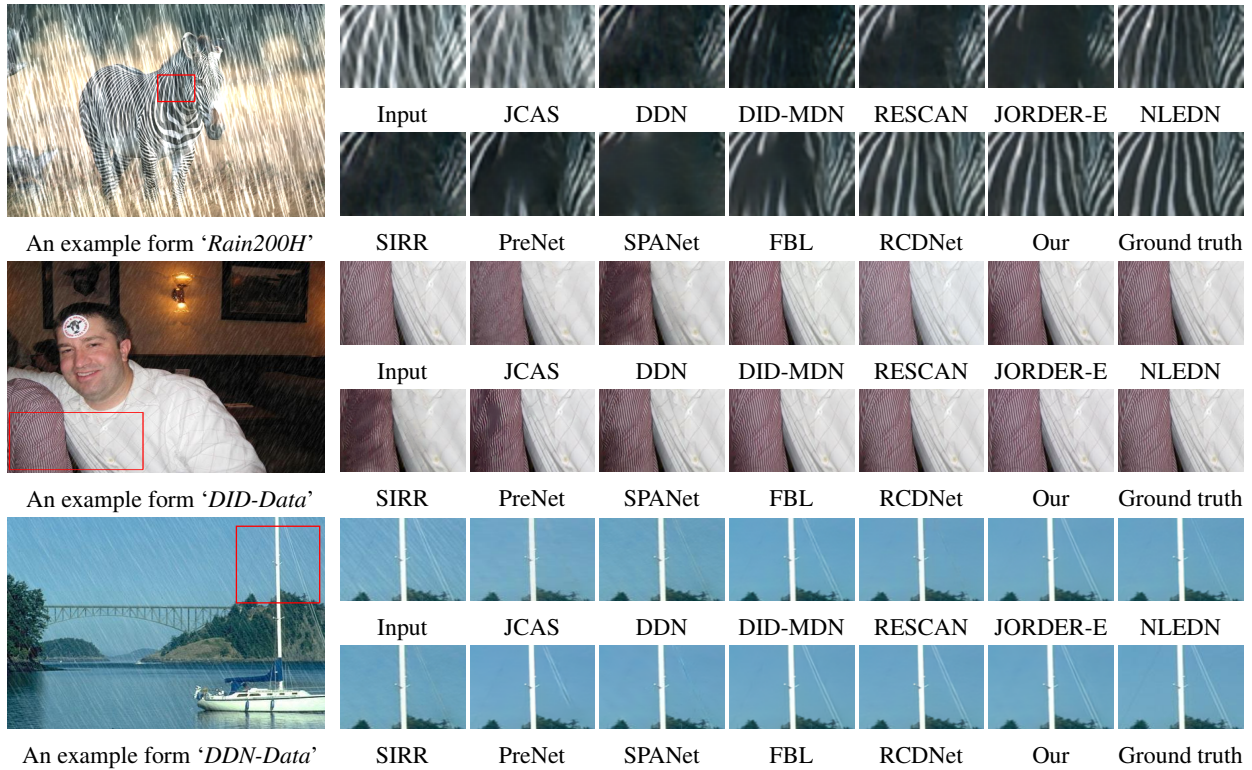


Figure 4: Visual comparisons on different synthetic data sets. Please zoom in for better visualization.

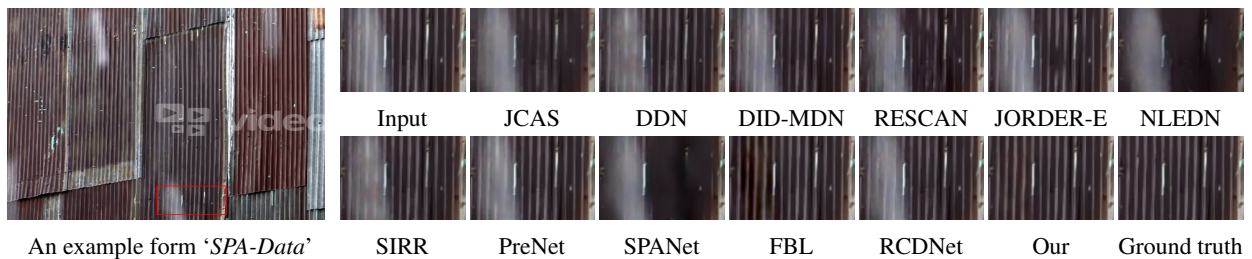


Figure 5: One visual comparison on the real-world data set 'SPA-Data'.



Figure 6: Visual comparisons on the Internet data set (Wang et al. 2019).

	DSC	GMM	JCAS	DDN	DID-MDN	RESCAN	NLEDN	JORDER-E	ID-CGAN	SSIR	PreNet	SPANet	FBL	RCDNet	Ours
CPU	198.32	681.81	587.46	3.21	77.24	70.43	48.52	207.03	15.47	3.51	95.66	—	243.73	34.55	18.11
GPU	—	—	—	0.34	0.77	1.53	0.84	1.74	0.37	0.47	0.69	0.43	0.41	0.57	0.31
# Params	—	—	—	58.2K	0.37M	54.7K	1.01M	4.17M	8.47M	0.18M	0.17M	0.28M	3.70M	3.17M	2.73M

Table 2: Comparisons on runtime (seconds) and parameter numbers. SPANet uses CuPy library and can only be run on GPU.

Settings	Baseline	Baseline + cGCN	Baseline + sGCN	Final Network
PSNR	29.13	30.53	30.61	31.15
SSIM	0.8892	0.9086	0.9103	0.9125

Table 3: Ablation study on the effect of our GCN modules.

Kernel #	Unit #	$L = 7$	$L = 11$	$L = 15$
	$K = 36$		29.17	30.47
$K = 72$		30.81	31.15	31.23
$K = 108$		30.95	31.24	31.37

Table 4: PSNR values on different parameter settings.

the ablation study and compare our network with its three variants over the challenging *Rain200H* data set. As shown in Table 3, compared to the baseline model, which only contains dilated convolutional modules and skip-connections, adding the channel GCN module achieves an improvement of 4.80% and 2.18% on average PSNR and SSIM, respectively. Similarly, the spatial GCN module improves the baseline by 5.08% and 2.37%. This is because the spatial GCN module is beneficial to remove spatially long rain streaks and preserve object structures, which helps improve SSIM performance. The best results can be obtained by combining the two GCN modules, which significantly increases the average PSNR and SSIM by 6.93% and 2.55 %, respectively.

We also test the impact of kernel number and unit number. Specifically, we test the kernel numbers $K \in \{36, 72, 108\}$ and basic unit numbers $L \in \{7, 11, 15\}$, and the PSNR results are shown in Table 4. It is clear that increasing kernels and units can generate higher performance. Adding basic unit results in larger modeling capacity, which has a greater advantage over increasing the number of kernels. However, increasing K and L eventually brings only limited improvement at the cost of storage and computation. Thus, to balance the trade-o between performance and speed, we choose $K = 72$ and $L = 11$ as our default setting.

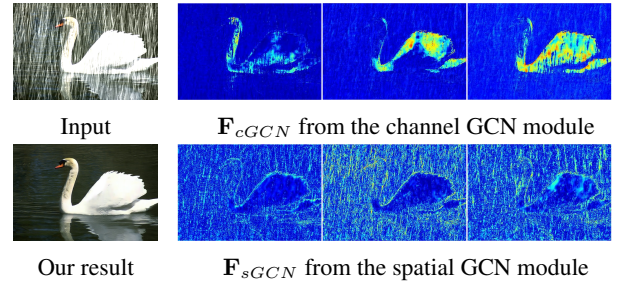


Figure 7: Visualizations of the two GCN modules.

Visualization

We now visualize some feature maps to see what types of representations the GCN modules have learned. Due to space limitation, for each GCN module, we only show three representative features in the last basic unit. As shown in the top row of Figure 7, since the channel GCN module is designed to explore correlations across channels, the generated features are able to highlight the animal’s body. This can be seen as response to global content-related information. On the other hand, as shown in the bottom row of Figure 7, the features generated by the spatial GCN have large response to dense and spatially coherent content, i.e., rain streaks and animal’s contour. This visualization demonstrates that the two proposed GCN modules is able to extract task-specific representations, which brings a significant improvement in both subjective and objective de-raining performance.

Conclusion

We introduced a new network to explore the contextual relationships for rain streak removal. We proposed two GCN modules to capture context along the spatial and channel dimensions, respectively. These two modules are further combined with a dilated convolutional module to form the basic unit, which are used to extract rich feature representations. The final de-raining network formed by the basic unit can well preserve object structures while removing complex and spatially long rain streaks.

References

- Abadi, M.; Agarwal, A.; Barham, P.; et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv: 1603.04467*.
- Barnum, P. C.; Narasimhan, S.; and Kanade, T. 2010. Analysis of rain and snow in frequency space. *Int'l. J. Computer Vision* 86(2): 256–274.
- Cao, X.; Zhou, F.; Xu, L.; Meng, D.; Xu, Z.; and Paisley, J. 2018. Hyperspectral image classification with Markov random fields and a convolutional neural network. *IEEE Trans. Image Process.* 27(5): 2354–2367.
- Chang, Y.; Yan, L.; and Zhong, S. 2017. Transformed Low-Rank Model for Line Pattern Noise Removal. In *ICCV*, 1726–1734.
- Chen, Y.; Kalantidis, Y.; Li, J.; Yan, S.; and Feng, J. 2018. A²-nets: Double attention networks. In *NeurIPS*, 350–359.
- Chen, Y.; Rohrbach, M.; Yan, Z.; Shuicheng, Y.; Feng, J.; and Kalantidis, Y. 2019. Graph-based global reasoning networks. In *CVPR*, 433–442.
- Chen, Y.-L.; and Hsu, C.-T. 2013. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *ICCV*, 1968–1975.
- Deng, S.; Wei, M.; Wang, J.; Feng, Y.; Liang, L.; Xie, H.; Wang, F. L.; and Wang, M. 2020. Detail-recovery Image Deraining via Context Aggregation Networks. In *CVPR*, 14560–14569.
- Eigen, D.; Krishnan, D.; and Fergus, R. 2013. Restoring an image taken through a window covered with dirt or rain. In *ICCV*, 633–640.
- Fu, X.; Huang, J.; Ding, X.; Liao, Y.; and Paisley, J. 2017a. Clearing the Skies: A deep network architecture for single-image rain removal. *IEEE Trans. Image Process.* 26(6): 2944–2956.
- Fu, X.; Huang, J.; Zeng, D.; Huang, Y.; Ding, X.; and Paisley, J. 2017b. Removing rain from single images via a deep detail network. In *CVPR*, 3855–3863.
- Garg, K.; and Nayar, S. K. 2007. Vision and rain. *Int'l. J. Computer Vision* 75(1): 3–27.
- Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NeurIPS*, 2672–2680.
- Gu, S.; Meng, D.; Zuo, W.; and Zhang, L. 2017. Joint convolutional analysis and synthesis sparse representation for single image layer separation. In *ICCV*, 1708–1716.
- Guo, C.; Li, C.; Guo, J.; Loy, C. C.; Hou, J.; Kwong, S.; and Cong, R. 2020. Zero-Reference Deep Curve Estimation for Low-Light Image Enhancement. In *CVPR*, 1780–1789.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.
- Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *CVPR*, 7132–7141.
- Hu, X.; Fu, C.-W.; Zhu, L.; and Heng, P.-A. 2019. Depth-attentional Features for Single-image Rain Removal. In *CVPR*, 8022–8031.
- Huang, D.-A.; Kang, L.-W.; Wang, Y.-C. F.; and Lin, C.-W. 2014. Self-learning based image decomposition with applications to single image denoising. *IEEE Trans. Multimedia* 16(1): 83–93.
- Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely Connected Convolutional Networks. In *CVPR*, 4700–4708.
- Johnson, J.; Gupta, A.; and Fei-Fei, L. 2018. Image generation from scene graphs. In *CVPR*, 1219–1228.
- Kang, L.-W.; Lin, C.-W.; and Fu, Y.-H. 2012. Automatic single image-based rain streaks removal via image decomposition. *IEEE Trans. Image Process.* 21(4): 1742–1755.
- Kim, J.-H.; Lee, C.; Sim, J.-Y.; and Kim, C.-S. 2013. Single-image deraining using an adaptive nonlocal means filter. In *IEEE ICIP*, 914–917.
- Kingma, D. P.; and Ba, J. 2014. Adam: A Method for Stochastic Optimization. In *ICLR*.
- Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.
- Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet classification with deep convolutional neural networks. In *NeurIPS*, 1097–1105.
- Li, G.; He, X.; Zhang, W.; Chang, H.; Dong, L.; and Lin, L. 2018a. Non-locally Enhanced Encoder-Decoder Network for Single Image De-raining. In *ACM MM*, 1056–1064.
- Li, M.; Xie, Q.; Zhao, Q.; Wei, W.; Gu, S.; Tao, J.; and Meng, D. 2018b. Video rain streak removal by multiscale convolutional sparse coding. In *CVPR*, 6644–6653.
- Li, R.; Cheong, L.-F.; and Tan, R. T. 2019. Heavy Rain Image Restoration: Integrating Physics Model and Conditional Adversarial Learning. In *CVPR*, 1633–1642.
- Li, S.; Araujo, I. B.; Ren, W.; Wang, Z.; Tokuda, E. K.; Junior, R. H.; Cesar-Junior, R.; Zhang, J.; Guo, X.; and Cao, X. 2019. Single image deraining: A comprehensive benchmark analysis. In *CVPR*, 3838–3847.
- Li, X.; Wu, J.; Lin, Z.; Liu, H.; and Zha, H. 2018c. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *ECCV*, 254–269.
- Li, Y.; Ouyang, W.; Zhou, B.; Shi, J.; Zhang, C.; and Wang, X. 2018d. Factorizable net: an efficient subgraph-based framework for scene graph generation. In *ECCV*, 335–351.
- Li, Y.; Tan, R. T.; Guo, X.; Lu, J.; and Brown, M. S. 2016. Rain Streak Removal Using Layer Priors. In *CVPR*, 2736–2744.
- Luo, Y.; Xu, Y.; and Ji, H. 2015. Removing rain from a single image via discriminative sparse coding. In *ICCV*, 3397–3405.
- Mao, X.; Shen, C.; and Yang, Y.-B. 2016. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *NeurIPS*, 2810–2818.

- Qin, X.; Wang, Z.; Bai, Y.; Xie, X.; and Jia, H. 2020. FFA-Net: Feature Fusion Attention Network for Single Image Dehazing. In *AAAI*, 11908–11915.
- Ren, D.; Zuo, W.; Hu, Q.; Zhu, P.; and Meng, D. 2019. Progressive image deraining networks: a better and simpler baseline. In *CVPR*, 3937–3946.
- Ren, W.; Liu, S.; Zhang, H.; Pan, J.; Cao, X.; and Yang, M.-H. 2016. Single image dehazing via multi-scale convolutional neural networks. In *ECCV*, 154–169.
- Ren, W.; Pan, J.; Zhang, H.; Cao, X.; and Yang, M.-H. 2020. Single image dehazing via multi-scale convolutional neural networks with holistic edges. *Int'l. J. Computer Vision* 128(1): 240–259.
- Ren, W.; Tian, J.; Han, Z.; Chan, A.; and Tang, Y. 2017. Video Desnowing and Deraining Based on Matrix Decomposition. In *ICCV*, 4210–4219.
- Wang, H.; Xie, Q.; Wu, Y.; Zhao, Q.; and Meng, D. 2020a. Single image rain streaks removal: a review and an exploration. *International Journal of Machine Learning and Cybernetics* 11(4): 853–872.
- Wang, H.; Xie, Q.; Zhao, Q.; and Meng, D. 2020b. A Model-driven Deep Neural Network for Single Image Rain Removal. In *CVPR*, 3103–3112.
- Wang, T.; Yang, X.; Xu, K.; Chen, S.; Zhang, Q.; and Lau, R. W. 2019. Spatial Attentive Single-Image Deraining with a High Quality Real Rain Dataset. In *CVPR*, 12270–12279.
- Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *CVPR*, 7794–7803.
- Wang, Y.; Liu, S.; Chen, C.; and Zeng, B. 2017. A Hierarchical Approach for Rain or Snow Removing in A Single Color Image. *IEEE Trans. Image Process.* 26(8): 3936–3950.
- Wang, Z.; Bovik, A. C.; Sheikh, H. R.; Simoncelli, E. P.; et al. 2004. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* 13(4): 600–612.
- Wei, W.; Meng, D.; Zhao, Q.; Wu, C.; and Xu, Z. 2019. Semi-supervised Transfer Learning for Image Rain Removal. In *CVPR*, 3877–3886.
- Wei, W.; Yi, L.; Xie, Q.; Zhao, Q.; Meng, D.; and Xu, Z. 2017. Should We Encode Rain Streaks in Video as Deterministic or Stochastic? In *ICCV*, 2516–2525.
- Yang, W.; Tan, R. T.; Feng, J.; Liu, J.; Guo, Z.; and Yan, S. 2017. Deep Joint Rain Detection and Removal From a Single Image. In *CVPR*, 1357–1366.
- Yang, W.; Tan, R. T.; Feng, J.; Liu, J.; Yan, S.; and Guo, Z. 2019. Joint Rain Detection and Removal from a Single Image with Contextualized Deep Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 42(6): 1377–1393.
- Yang, W.; Tan, R. T.; Wang, S.; Fang, Y.; and Liu, J. 2020a. Single image deraining: From model-based to data-driven and beyond. *IEEE Trans. Pattern Anal. Mach. Intell.* .
- Yang, W.; Tan, R. T.; Wang, S.; and Liu, J. 2020b. Self-Learning Video Rain Streak Removal: When Cyclic Consistency Meets Temporal Correspondence. In *CVPR*, 1720–1729.
- Yang, W.; Wang, S.; Xu, D.; Wang, X.; and Liu, J. 2020c. Towards Scale-Free Rain Streak Removal via Self-Supervised Fractal Band Learning. In *AAAI*, 12629–12636.
- Yasarla, R.; and Patel, V. M. 2019. Uncertainty guided multi-scale residual learning using a cycle spinning CNN for single image de-raining. In *CVPR*, 8405–8414.
- Yasarla, R.; Sindagi, V. A.; and Patel, V. M. 2020. Syn2Real Transfer Learning for Image Deraining Using Gaussian Processes. In *CVPR*, 2726–2736.
- Yu, F.; and Koltun, V. 2016. Multi-Scale Context Aggregation by Dilated Convolutions. In *ICLR*.
- Yu, W.; Huang, Z.; Zhang, W.; Feng, L.; and Xiao, N. 2019. Gradual Network for Single Image De-raining. In *ACM MM*, 1795–1804.
- Zha, Z.-J.; Liu, J.; Chen, D.; and Wu, F. 2020. Adversarial attribute-text embedding for person search with natural language query. *IEEE Transactions on Multimedia* 22(7): 1836–1846.
- Zhang, H.; and Patel, V. M. 2018a. Densely connected pyramid dehazing network. In *CVPR*, 3194–3203.
- Zhang, H.; and Patel, V. M. 2018b. Density-aware Single Image De-raining using a Multi-stream Dense Network. In *CVPR*, 695–704.
- Zhang, H.; Sindagi, V.; and Patel, V. M. 2019. Image deraining using a conditional generative adversarial network. *IEEE Trans. Circuits Syst. Video Technol.* 30(11): 3943–3956.
- Zhang, H.; Zha, Z.-J.; Yang, Y.; Yan, S.; and Chua, T.-S. 2014. Robust (semi) nonnegative graph embedding. *IEEE Transactions on Image Processing* 23(7): 2996–3012.
- Zhang, Y.; Li, K.; Li, K.; Wang, L.; Zhong, B.; and Fu, Y. 2018a. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 286–301.
- Zhang, Y.; Tian, Y.; Kong, Y.; Zhong, B.; and Fu, Y. 2018b. Residual dense network for image super-resolution. In *CVPR*, 2472–2481.
- Zhang, Y.; Tian, Y.; Kong, Y.; Zhong, B.; and Fu, Y. 2020. Residual dense network for image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.* .
- Zhu, H.; Peng, X.; Zhou, J. T.; Yang, S.; Chandrasekh, V.; Li, L.; and Lim, J.-H. 2019. Single image rain removal with unpaired information: A differentiable programming perspective. In *AAAI*, 9332–9339.
- Zhu, L.; Fu, C.-W.; Lischinski, D.; and Heng, P.-A. 2017. Joint Bi-Layer Optimization for Single-Image Rain Streak Removal. In *ICCV*, 2526–2534.
- Zhu, Y.; Zha, Z.-J.; Zhang, T.; Liu, J.; and Luo, J. 2020. A Structured Graph Attention Network for Vehicle Re-Identification. In *ACM MM*, 646–654.