D3R-Net: Dynamic Routing Residue Recurrent Network for Video Rain Removal

Jiaying Liu, *Senior Member, IEEE*, Wenhan Yang, *Student Member, IEEE*, Shuai Yang, and Zongming Guo, *Member, IEEE*

***Abstract*— In this paper, we address the problem of video rain removal by considering rain occlusion regions, i.e., very low light transmittance for rain streaks. Different from additive rain streaks, in such occlusion regions, the details of backgrounds are completely lost. Therefore, we propose a hybrid rain model to depict both rain streaks and occlusions. Integrating the hybrid model and useful motion segmentation context information, we present a Dynamic Routing Residue Recurrent Network (D3R-Net). D3R-Net first extracts the spatial features by a residual network. Then, the spatial features are aggregated by recurrent units along the temporal axis. In the temporal fusion, the context information is embedded into the network in a "*dynamic routing*" way. A heap of recurrent units takes responsibility for handling the temporal fusion in given contexts, e.g., rain or non-rain regions. In the certain forward and backward processes, one of these recurrent units is mainly activated. Then, a context selection gate is employed to detect the context and select one of these temporally fused features generated by these recurrent units as the final fused feature. Finally, this last feature plays a role of "*residual feature*." It is combined with the spatial feature and then used to reconstruct the negative rain streaks. In such a D3R-Net, we incorporate motion segmentation, which denotes whether a pixel belongs to fast moving edges or not, and rain type indicator, indicating whether a pixel belongs to rain streaks, rain occlusions, and non-rain regions, as the context variables. Extensive experiments on a series of synthetic and real videos with rain streaks verify not only the superiority of the proposed method over state of the art but also the effectiveness of our network design and its each component.**

***Index Terms*— Video rain removal, dynamic routing, spatial temporal residue, recurrent neural network.**

I. Introduction

BAD weather conditions cause a series of visibility degradations and alter the content and color of images. Such signal distortion and detail loss lead to the failure of many outdoor computer vision applications, which generally rely on clean video frames as their input. Rain streaks, as one of the most common degradations in rain frames, make severe intensity fluctuations in small regions, and thus obstruct and blur the scene.

In the past decades, many researchers have been dedicated to rain image/video restoration. The rain removal from a single image [27], [32], [41], [47] solves this problem by signal separation between rain streaks and background images (non-rain images), based on their texture appearances. Frequency domain representation [32], sparse representation [41], Gaussian mixture model [37] and deep networks [18], [65] are adopted as basic models to differentiate rain streaks and background images. Furthermore, video-based methods [1]–[3], [10], [16], [19], [21], [22], [70] solve the problem based on both spatial and temporal redundancies. Some works [19], [21], [22] built on physical models, such as directional and chromatic properties of rains. Others [7], [10], [31], [35] further utilized temporal dynamics, including continuity of background motions, random appearing of streaks among frames, and explicit motion modeling, to facilitate video rain removal.

These methods achieve good performance in some cases. However, they still neglect some important issues:

- In real-world scenarios, degradations caused by rain streaks are more complex. The additive rain model widely used in previous methods [10], [32] is insufficient to cover visual effects of some important degradations in practice. When the light transmittance of rain streaks is low, their corresponding background regions are totally occluded, and the whole occlusion regions only present the rain reliance.
- The spatial and temporal redundancies are considered separately. These two kinds of information are intrinsically correlated and complementary. The potential of jointly exploiting the information is not fully explored. Low rank based methods [35], [58] have made some attempts. However, they usually rely on the assumption of a static background. Therefore, their results may be degraded when large and violent motions are included.
- Although some previous works [6], [28], [62] try to include context information, *e.g.* categories [28] or motion segmentations [6], [62], a general and easily equipped framework for that purpose is lacked. These previous works need deliberate expert efforts to embed the context information to facilitate rain streak removal. Once the commonly seen contexts or rain streak statistics change, the pipeline needs to be rebuilt.
- For learning-based video rain streak removal, training for recovery purposes remains challenging. The training

Manuscript received February 15, 2018; revised July 16, 2018; accepted September 3, 2018. Date of publication September 13, 2018; date of current version October 11, 2018. This work was supported in part by the National Natural Science Foundation of China under Contract 61772043, in part by the CCF-Tencent Open Research Fund, and in part by NVIDIA Corporation with the GPU. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Dacheng Tao. *(Corresponding author: Wenhan Yang.)*

The authors are with the Institute of Computer Science and Technology, Peking University, Beijing 100080, China (e-mail: liujiaying@pku.edu.cn; yangwenhan@pku.edu.cn; pkuwilliamyang@pku.edu.cn; guozongming@pku.edu.cn).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author. The material includes more analysis of D3RNet. The total size of the video is 32.9 MB. Contact liujiaying@pku.edu.cn for further questions about this work.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2018.2869722

relies on the video pairs synthesized from a large-scale high-quality video dataset with various scenes and objects. It is cost-heavy to collect such a dataset to synthesize rain frames.

Considering these limitations of existing works, our goal is to build a novel video rain model that can describe various types of rain in practice, including both rain streaks and rain occlusions. Then, based on this model, we further develop a deep learning architecture to solve the corresponding inverse problem. We aim to develop a systematic approach to train the network with a rain video dataset synthesized from a medium-sized high-quality video set.

To achieve these goals, we explore possible rain models and deep learning architectures that can effectively restore clean frames even when rain occlusion regions appear and are flexible to embed context information. We first develop a hybrid rain model to depict both rain streaks and occlusions. Then, a **D**ynamic **R**outing **R**esidue **R**ecurrent **Net**work (**D3R-Net**) is built to seamlessly integrate context variable estimations, and a rain removal based on both spatial appearance feature and temporal coherence. The rain type indicator and motion segmentation are embedded into D3R-Net in a dynamic routing way, flexible to be extended to incorporate other context information. This paper is an extension of our previous conference paper [38]. Based on the rain degradation model in the preliminary work, we choose a parallel technical route to address the problem of the video rain removal with dynamically detected video contexts. Novel deep recurrent networks as well as a more effective basic component – spatial temporal residue learning – for video modeling are developed. At the same time, a flexible framework to detect and incorporate video contexts is built. We add extensive experimental analysis to evaluate the effectiveness of the proposed framework on several datasets. Our contributions are as follows,

- We propose a novel hybrid video rain model that visits various rain cases including rain occlusions. In rain occlusion regions, the pixels are replaced by rain reliance. This regional information is then embedded into the proposed method for video deraining.
- We are the first to solve the problem of video rain removal with deep recurrent networks. Specifically, a D3R-Net is proposed. The rain streaks appear randomly among frames, whereas the motions of backgrounds are tractable. Considering that, recurrent neural networks (RNN) are employed to encode the information of adjacent background frames from their degraded observations, obtaining representative features for deraining. Furthermore, our D3R-Net utilizes a spatial temporal residue learning, where the temporally fused feature plays a role of "residue feature".
- Based on the proposed refined hybrid rain model, and further considerations of the commonly seen context variables that appeared in previous works, D3R-Net is seamlessly integrated with motion segmentation and rain type indicator in a "*dynamic routing*" framework. Its core idea is that, the network components have several copies. Each copy is good at handling the rain removal in a given context. Then, in each training or testing iteration, the network is constructed dynamically based on the detected context. This "*dynamic routing*" framework and the added contexts lead to a performance gain.

The remainder of this paper is organized as follows: Section II gives a brief overview of the related work. In Section III, we present our hybrid video rain model and the related rain removal context. In Section IV, the proposed dynamic routing residue recurrent neural network is built step by step and then the context information is embedded into the network in the "*dynamic routing*" way. Experimental results are illustrated in Section V. Finally, concluding remarks are given in Section VI.

II. Related Work

A. Single Image Rain Removal

Single image deraining is a highly ill-posed problem and is addressed by a signal separation or texture classification route. Kang *et al.* [32] attempted to separate rain streaks from the high frequency layer by sparse coding. Then, a generalized low rank model [10] was proposed, where the rain streak layer is assumed to be low rank. Kim *et al.* [34] first detected rain streaks and then removed them with the nonlocal mean filter. Luo *et al.* [41] proposed a discriminative sparse coding method to separate rain streaks from background images. In [37], Gaussian mixture modsels are exploited to separate the rain streaks.

The presence of deep learning promoted the development of image processing. The related topics include super-resolution [15], [24], [33], [36], [59], [61], [63], [64], [66], compression artifacts removal [4], [13], [71], denoising [8], [9], [68], low light enhancement [40], [53], [57], image and video compression [25], [26], [49], [60], *et al*. As for the single image rain removal, deep learning-based methods also led to a fast development and offered new state-of-the-art performance. In [17] and [18], deep networks that take the image detail layer as their inputs and predict the negative residues are constructed. They have good capacities to keep texture details. But they cannot handle heavy rain cases where rain streaks are dense. Yang *et al.* [65] proposed a deep joint rain detection and removal method was proposed to recurrently remove rain streaks and accumulations, obtaining impressive results in heavy rain cases. Zhu *et al.* [72] proposed a rain removal method by decomposing the rain image into a rain-free background layer R and a rain-streak layer B. The method then removes rain-streak details from B and removes non-streak details from R alternately. In [67], a novel density-aware multi-stream densely connected convolutional neural network is proposed for joint rain density estimation and rain streak removal. Chang *et al.* [5] aimed to address line pattern noise removal, and used an image decomposition model to map the input image to a domain where the line pattern appearance has an extremely distinct low-rank structure. Wang *et al.* [52] regarded rain removal as an image-to-image translation problem, and developed a perceptual generative adversarial network to address it. In this network, the generative adversarial loss and the perceptual adversarial loss are integrated, and the sub-modules of the network are

trained alternately. Compared with these works, which utilize deep networks to address the problem of single image rain removal, our work explores to remove rains from videos by jointly modeling intra-frame dependencies and inter-frame motion dynamics with recurrent neural networks.

B. Video Rain Removal

Garg and Nayar were the first to focus on modeling rains, *i.e.* the photometric appearance of rain drops [21] and addressing rain detection and removal based on dynamic motion of rain drops and irradiance constraint [19], [22]. In their subsequent work [20], camera settings are explored to control the visibility of rain drops. These early attempts heavily rely on the linear space-time correlation of rain drops, and thus fail when rain streaks are diversified in scales and densities. Later works formulate rain streaks with more flexible and intrinsic models. In [70], the temporal and chromatic properties of rain are visited to differentiate rain, background and moving objects. In [39], a theory of chromatic property of rain is developed. Barnum *et al.* [1] utilized the features in Fourier domain for rain removal. Santhaseelan and Asari [44] developed phase congruency features to detect and remove rain streaks. Successive works make their efforts in distinguishing fast moving edges and rain streaks. In [2] and [3], the size, shape and orientation of rain streaks are used as discriminative features. In [10], the spatio-temporal correlation of local patches are encoded by a low-rank model to separate rain streaks and natural frame signals. Jiang *et al.* [31] further considered the overall directional tendency of rain streaks, and used two unidirectional total variation regularizers to constrain the separation of rain streaks and background. The presence of learning-based method, with improved modeling capacity, brings in new opportunities. Chen and Chau [7] proposed to embed motion segmentation by Gaussian mixture model into rain detection and removal. Tripathi and Mukhopadhyay [50], [51] trained Bayes rain detector based on spatial and temporal features. Kim *et al.* [35] trained an SVM to refine the roughly detected rain maps. Wei *et al.* [58] encoded rain streaks as patch-based mixtures of Gaussian, which is capable of finely adapting a wider range of rain variations. In [43], a matrix decomposition model is presented to divide rain streaks or snowflakes into two categories: sparse and dense ones, for video desnowing and deraining. Compared with previous methods, our work is the first one to employ deep networks to handle video rain removal. Beyond that, instead of hand-crafting pipelines to model rain context, we provide a flexible and convenient framework – "*dynamic routing*" for that purpose to facilitate video rain removal.

III. Hybrid Video Rain Model and Rain Removal Context

In this section, we first focus on building a single rain model that can describe non-rain, rain streak and rain occlusion regions. Then, we discuss the context of rain removal, *i.e.*, the degradation type in this hybrid video rain model, which can be regarded as side information to benefit rain removal.

Fig. 1. Left and middle panels: two adjacent rain frames. Right panel: the rain streaks in these rain frames, denoted in blue and red colors, respectively. The presented streaks have similar shapes and directions, and however, their distributions in spatial locations are uncorrelated.

Fig. 2. Examples of rain occlusions in video frames. Compared with additive rain streaks, the rain occlusions (denoted in red color) contain little structural details of the background image.

A. Additive Rain Model

The widely used rain model [28], [37], [41] is expressed as:

$$\mathbf{O} = \mathbf{B} + \mathbf{S}, \tag{1}$$

where $\mathbf{B}$ is the background frame without rain streaks, and $\mathbf{S}$ is the rain streak frame. $\mathbf{O}$ is the captured image with rain streaks. Based on Eq. (1), rain removal is regarded as a signal separation problem [37], [41], [65]. Namely, given the observation $\mathbf{O}$, removing rain streaks is to estimate the background $\mathbf{B}$ and rain streak $\mathbf{S}$, based on the different characteristics of the rain-free images and rain streaks.

This single-frame rain synthesis model in Eq. (1) can be extended to a multi-frame one by adding a time dimension as follows,

$$\mathbf{O}_t = \mathbf{B}_t + \mathbf{S}_t, \quad t = 1, 2, ..., N, \tag{2}$$

where t and N signify the current time-step and total number of the frames, respectively. Rain streaks $\mathbf{S}_t$ are assumed to be independent identically distributed random samples [46]. Their locations across the frames are uncorrelated, as shown in Fig. 1.

However, in practice, degradations generated by rain streaks are very complex. For example, when the rain level is moderate or even heavy, the light transmittance of rain drop becomes low and the rain region of $\mathbf{O}_t$ presents identical intensities, as shown in Fig. 2. In this case, the signal superposition of rain frames includes rain streaks and rain occlusions. Based on Eq. (1), the deduced $\mathbf{S}_t = \mathbf{O}_t - \mathbf{B}_t$ deviates from its original distribution and contains more structure details. Rain removal in rain occlusion regions needs to remove the rain reliance

and fill in the missing details. Thus, it is harder to learn a mixture mapping that restores signals in all regions without distinction. It is meaningful to build a unified hybrid model that describes both two kinds of degradation to guide solving the task of rain removal.

B. Occlusion-Aware Rain Model

To address this issue, we propose a hybrid rain model that is adaptive to model rain occlusions. In such a model, all pixels in rain frames are classified into two groups: 1) the ones following the additive rain model in Eq. (1); 2) the others whose pixel values are just equal to the rain reliance. The formulation of such a hybrid rain model is given as follows,

$$\mathbf{O}_t = (1 - \alpha_t)(\mathbf{B}_t + \mathbf{S}_t) + \alpha_t \mathbf{A}_t, \tag{3}$$

where $\mathbf{A}_t$ is the rain reliance map and α_t is an alpha matting map defined as follows,

$$\alpha_t(i, j) = \begin{cases} 1, & \text{if } (i, j) \in \Omega_{\mathbf{S}}, \\ 0, & \text{if } (i, j) \notin \Omega_{\mathbf{S}}, \end{cases} \tag{4}$$

where $\Omega_{\mathbf{S}}$ is the region where the light transmittance of rain drop is low, which is defined as *rain occlusion region*.

C. Rain Removal Context

Based on Eqs. (3) and (4), the inverse mapping of the rain streaks and rain occlusions is quite different. Thus, estimating α_t is important for rain removal. Besides, as summarized in previous works [2], [3], [7], one of the most difficult issues for video rain removal is the overlapping of fast moving edges and rain streaks in the feature space. Thus, a preferred method should first detect these context variables, *e.g.* rain type and motion segmentation, and then perform rain removal accordingly. In our work, the difference of adjacent frames are used as a standard to classify motion regions. For ground truth background frames, if the square of the difference of two adjacent frames is greater than 0.01, the region is denoted as motion regions. Till now, we regard rain type and motion segmentation as the context of rain removal. In the next section, we build a deep network architecture to predict the context and utilize the information to facilitate rain removal.

IV. Dynamic Routing Residue Recurrent Neural Network for Rain Removal

In this section, we first construct a spatial-temporal residue recurrent neural network step by step for rain removal as shown in Fig. 3. Then, we extend the network to a dynamic routing RNN, as shown in Fig. 5. In each recurrence of the network, there are multiple recurrent unit paths, but only one path is mainly activated based on the detected context, as shown in Fig. 4.

A. Spatial-Temporal Residue Recurrent Network

Single frame rain streak removal aims to recover the rain-free background (target frame) based on a rain image (input frame). Several popular image processing networks [14], [56], [64] use a convolutional neural network (CNN) model to extract features from the input frame and then map it to the target one. A typical CNN architecture consists of three

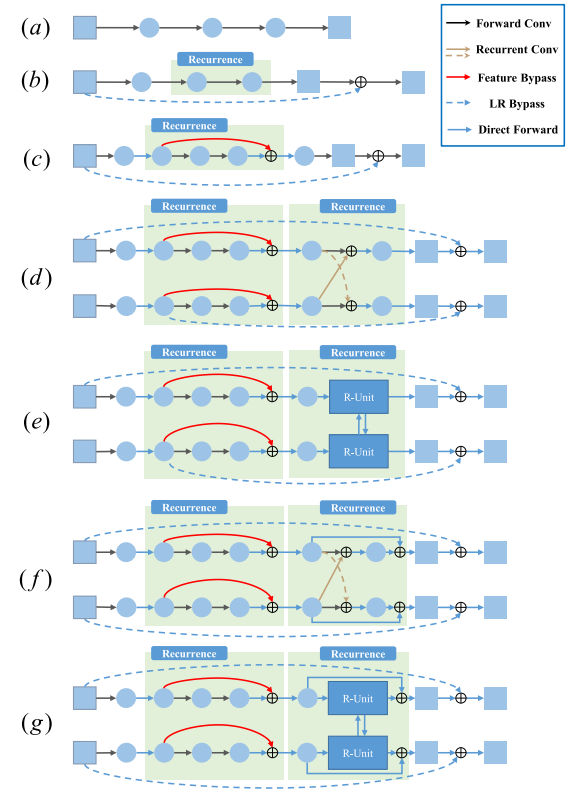


Fig. 3. Network architectures from a vanilla convolutional neural network (CNN) to our proposed spatial-temporal residue recurrent network. (a) vanilla CNN. (b) CNN with LR bypass connections. (c) CNN has both LR and feature bypass connections. (ResNet) (d) Multiple ResNets are connected by convolutional recurrent units to model inter-frame dependencies. (e) Gated recurrent units (R-Unit) are used to connect different ResNets to model inter-frame redundancies. (f) Temporal fused features by convolutional recurrent units are added with the spatial ones and play a role of "residual features" that are complements to spatial features. (g) Temporal fused features by gated recurrent unit (R-Unit) are added with the spatial ones and play a role of "residual features" that are complements to spatial features. (Best viewed in color.)

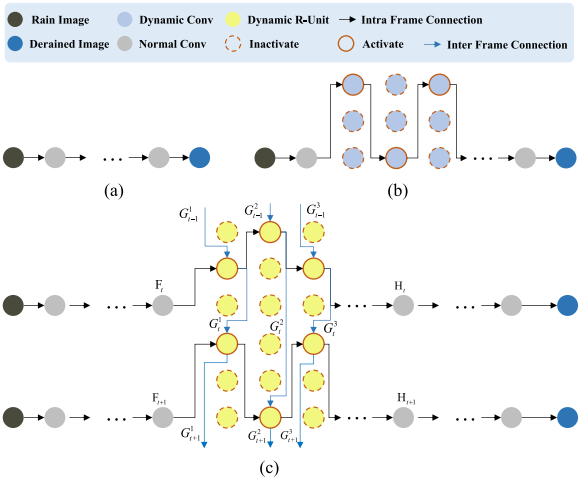


Fig. 4. Network architecture of dynamic routing CNN and RNN. (a) vanilla CNN. (b) CNN with dynamic routing mechanism. (**Dynamic Routing CNN**) The convolutional path is constructed based on the detected rain removal context. (c) RNN with dynamic routing mechanism. (**Dynamic Routing RNN**) The recurrent unit path is built based on the detected rain removal context. (Best viewed in color.)

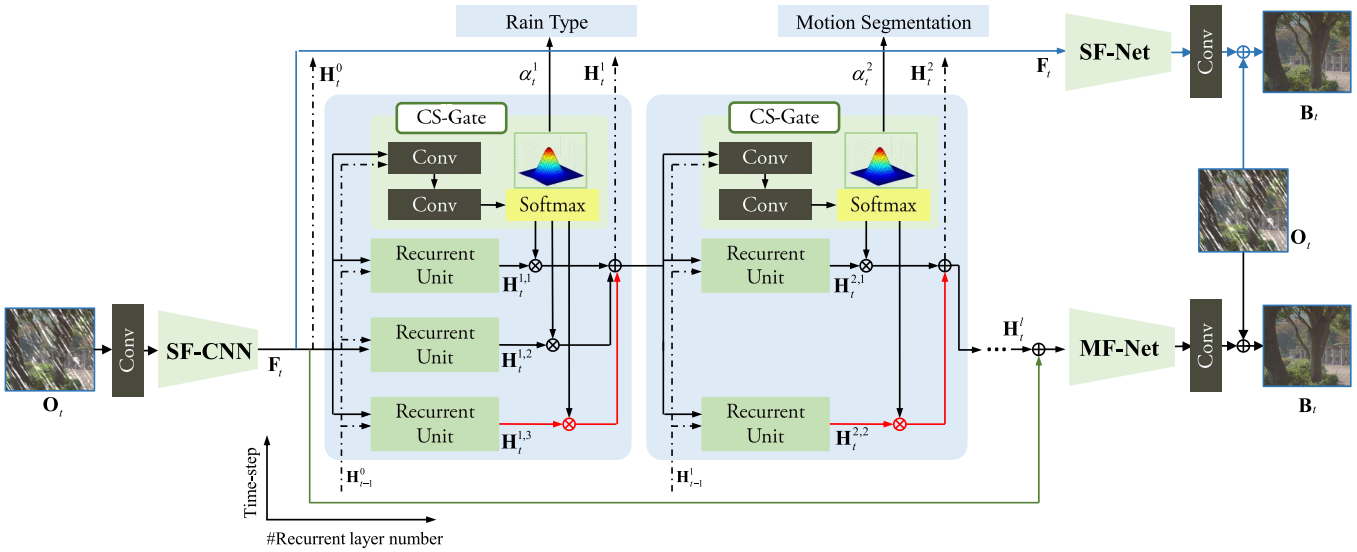


Fig. 5. The framework of **D**ynamic **R**outing **R**ecurrent **R**edidue **Net**work (**D3R-Net**). We first employ a single frame CNN (SF-CNN) to extract features $\mathbf{F}_t$ of the t-th frame $\mathbf{O}_t$. Then, the subsequent network components predict the clean background frames by two paths: 1) single-frame path (denoted by blue lines); 2) multi-frame path (denoted by black lines and red lines). The multi-frame path works in a dynamic routing way. (Best viewed in color).

convolutional layers as proposed in [14] which jointly performs sparse coding and reconstruction over the input frames as shown in Fig. 3(a). However, striving for directly recovering the complete target frames may make the CNN models fail to recover some important high frequency details. In contrast, using deep networks to model the difference signals [33], [68] as shown in Fig. 3(b), equivalently residue signals or negative rain streaks, could recover high frequency details better. The added bypass connection in Fig. 3(c) leads the network training to converge faster and to a better state.

To utilize temporal redundancies and model motion context among frames, the recurrent units are used to fuse spatial features along the temporal axis. The recurrent units can be convolutional recurrent connections [29] as shown in Fig. 3(d) or gated ones, *i.e.* long short-term memory units [48] and gated recurrent units [11] as shown in Fig. 3(e). They are proven effective in capturing inter-frame dependencies and inferring the missing high-frequency details in a series of video restoration tasks, *e.g.* video super-resolution [29], [48]. However, this architecture has its drawbacks, especially when its training usually relies on the pretraining of spatial CNN. First, all the information that input into the next stage of the network comes from the temporal fusion step only. The training of such a temporal fusion in the finetuning step may first goes through a dropped performance. Second, the temporal fusion units, *e.g.* convolutional recurrent units or GRUs, are good at modeling inter-frame dependencies. However, in this fusion step, some spatial appearance details extracted from single frames may be lost.

To address these issues, we propose to use residual RNN architecture to replace the normal RNN, as shown in Figs. 3(f) and (g). In each recurrence, we do not directly input the temporally fused features into the next stage of the network. Instead, we first combine the temporally fused features and single frame spatial features by summation, where the temporally fused features play a role of residue features. Then, the aggregated features are forwarded to the next stage of the network and transformed into the predicted target frame. This combination is significant, because these combined two paths can provide temporal dynamics while preserving the spatial appearance details, and thus offer better modeling capacities.

B. Dynamic Routing RNN

The generic CNN handles a task with the same components and parameters for all contexts. The formulation of a convolutional layer as shown in Fig. 4(a) is represented as follows,

$$\mathbf{H} = f\left(U\mathbf{F} + b\right), \tag{5}$$

where $\mathbf{F}$ is the layer input, and $\mathbf{H}$ is the layer output. f is usually a nonlinear function, such as ReLU or tanh. U and b are weight and bias of the convolution. This layer maps the input feature $\mathbf{F}$ to output feature $\mathbf{H}$ given any context.

Intuitively, this "one for all" architecture may have limitations when we expect the network can focus on different mappings in various contexts. For example, in video rain removal, we expect that foreground textures are preserved in non-rain regions and the background regions can be smoothed to remove sparkle noises. Thus, to improve the adaptability of the generic CNN model, we set a series of network compositions, and to select some of them to construct a deep network based on the given context online. As shown in Fig. 4(b), for some layers, called dynamic convolutions, there are three convolutions for one convolution layer position. In each forward or backward process, only one of the three convolutions is selected and activated. Naturally, this paradigm can be extended to apply for RNN, as shown in Fig. 4(c). For dynamic recurrent units (Dynamic R-Unit), there are also multiple units for each layer position. In each forward or

backward process, a sub-network is constructed with one activated recurrent unit for each layer position.

However, these hard designs are difficult to be optimized in an end-to-end manner. Thus, in the following, we propose an equivalent soft dynamic routing RNN/CNN. We change the normal convolution operation to a dynamic routing one as shown in Fig. 4(b) as follows,

$$\mathbf{H} = \int_{\sigma} f\left(U\mathbf{F} + b|\alpha\right) g(\alpha), \tag{6}$$

where α is a context variable, *e.g.* an indicator that illustrates whether a pixel belongs to non-rain, rain streak or rain occlusion regions. $f\left(U\mathbf{F} + b|\alpha\right)$ is the conditional convolution, given the context variable α. $g(\alpha)$ is a probability density function of α having

$$\int_{\alpha} g(\alpha) = 1. \tag{7}$$

Eq. (6) equals to conducting convolution filters with various α. Then, these filtered results are weighted together based on appearance probability of α. When α is discrete-valued, Eq. (6) is derived as

$$\begin{aligned} \mathbf{H} &= \sum_{\alpha_i} f^i\left(U\mathbf{F} + b\right) g(\alpha^i), \\ \sum_i g(\alpha^i) &= 1, \\ f^i(\cdot) &= f(\cdot|\alpha = \alpha^i). \end{aligned} \tag{8}$$

Similarly, the recurrent neural network can be extended to a dynamic routing one. The vanilla recurrent unit works in the following way,

$$\mathbf{H}_t = f\left(U\mathbf{F}_t + W\mathbf{H}_{t-1}\right), \tag{9}$$

where $\mathbf{F}_t$ is the input at the time step t, and $\mathbf{H}_t$ is the hidden state at the time step t. f is usually a nonlinear function, such as ReLU or tanh. The hidden state $\mathbf{H}_t$ can be regarded as the memory of the network. $\mathbf{H}_t$ captures information about what happened in all previous time steps. Similar to the change from (5) to (6), given the context information α_t at time-step t, Eq. (9) is updated as follows,

$$s_t = \int_{\alpha_t} f\left(U\mathbf{F}_t + W\mathbf{H}_{t-1}|\alpha_t\right) g(\alpha_t), \tag{10}$$

$$\int_{\alpha_t} g(\alpha_t) = 1. \tag{11}$$

When α_t is discrete-valued, Eq. (10) can be derived as

$$\mathbf{H}_t = \sum_i f^i\left(U\mathbf{F}_t + W\mathbf{H}_{t-1}\right) g(\alpha_t^i), \tag{12}$$

where

$$\begin{aligned} \sum_i g(\alpha_t^i) &= 1, \\ f^i(\cdot) &= f(\cdot|\alpha_t = \alpha_t^i). \end{aligned}$$

Similarly, the implications of Eqs. (12)-(13) are quite simple. To get a meaningful output $\mathbf{H}_t$, we first estimate a multi-channel map $\left\{g(\alpha_t^i)\right\}$ showing the appearance probability of each context. It shows whether a location belongs to a category, *e.g.* smooth regions or fast moving edge regions, rain regions or non-rain regions *et al*. Then, based on the probability map $\left\{g(\alpha_t^i)\right\}$, $\mathbf{H}_t$ is inferred by weighting the results obtained from the corresponding mappings $\left\{f^i(\cdot)\right\}$.

C. Dynamic Routing Residue Recurrent (D3R) Neural Network for Rain Removal

Based on the above-mentioned dynamic routing mechanism, we build a **D**ynamic **R**outing **R**ecurrent **R**esidual **Net**work (**D3RNet**). The whole network architecture is illustrated as Fig. 5. Briefly, we first extract the features $\mathbf{F}_t$ of each frame by a residual CNN. Then, the subsequent components of D3R-Net predict the negative rains by two paths:

- Single-frame path (denoted by blue lines). This path directly transforms single frame spatial feature $\mathbf{F}_t$ into the negative rains to estimate the clean background frame. This path forces the extracted $\mathbf{F}_t$ meaningful.
- Multi-frame path (denoted by black and red lines). This path first fuses the spatial features along the temporal axis in a dynamic routing way. Several recurrent units are expected to take responsibility for handling the temporal fusion in given contexts, *e.g.* rain or non-rain regions, to generate a series of temporally fusion results $\left\{\mathbf{H}_t^{i,j}\right\}$. In the certain forward and backward processes, one of these recurrent units is mainly activated in each time-step. A **C**ontext **S**election **Gate** (**CS-Gate**) is used to detect the context and select one of these fused features (*e.g.* denoted by red lines) as the final fused feature in the given context, *e.g.* $\mathbf{H}_t^{1,3}$ and $\mathbf{H}_t^{2,2}$ in Fig. 5. Then, the temporally fused feature is combined with the spatial feature from a skip connection (denoted by green line) by a summation operation. At last, the combined feature is projected into the predicted negative rain streaks by a CNN.

The details and formalized descriptions of D3R-Net are illustrated in the following.

1) Single Frame CNN Extractor (SF-CNN): The residual learning architecture [23], [65] is used for single frame CNN feature extraction. As shown in Fig. 6, residual blocks are stacked to build a CNN network. In formulation, let $\mathbf{f}_{t,k,\text{in}}^{\text{c}}$ denote the input feature map of the k-th residual block. The output feature map of the k-th residual block, $\mathbf{f}_{t,k,\text{out}}^{\text{c}}$, is progressively updated as follows:

$$\begin{aligned} \mathbf{f}_{t,k,\text{out}}^{\text{c}} &= \max\left(0, \mathbf{W}_{t,k,\text{mid}}^{\text{c}} * \mathbf{f}_{t,k,\text{mid}}^{\text{c}} + \mathbf{b}_{t,k,\text{mid}}^{\text{c}} + \mathbf{f}_{t,k,\text{in}}^{\text{c}}\right), \\ \mathbf{f}_{t,k,\text{mid}}^{\text{c}} &= \max\left(0, \mathbf{W}_{t,k,\text{in}}^{\text{c}} * \mathbf{f}_{t,k,\text{in}}^{\text{c}} + \mathbf{b}_{t,k,\text{in}}^{\text{c}}\right), \end{aligned} \tag{13}$$

where $*$ signifies the convolution operation. $\mathbf{W}$ and $\mathbf{b}$ with subscripts and superscripts denote the weight and bias of the corresponding convolution layers, respectively. $\mathbf{f}_{t,k,\text{in}}^{\text{c}} = \mathbf{f}_{t,k-1,\text{out}}^{\text{c}}$ is the output features of the $(k-1)$-th residual block. There is a by-pass connection here between $\mathbf{f}_{t,k,\text{in}}^{\text{c}}$ and $\mathbf{f}_{t,k,\text{out}}^{\text{c}}$. This architecture is proven effective in increasing the network depth and improving network training. The output feature map is denoted as $\mathbf{F}_t$, where t is the time-step of the frame. $\mathbf{F}_t$ encodes the spatial information of $\mathbf{O}_t$.

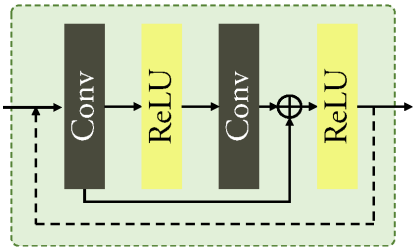


Fig. 6. The CNN architecture for single frame CNN feature extraction. R-Net, C-Net and JRC-Net adopt this network architecture as well.

2) Recurrent Units: Compared to the single frame rain removal, video rain removal can make use of temporally sequential information. To make use of temporal redundancies, we use recurrent units to connect different frames and fuse their features along the temporal axis. After obtaining the aggregated feature in the last time-step of the given recurrent layer $\mathbf{H}_{t-1}^{j}$ and that in the last time-step of the previous recurrent layer $\mathbf{H}_{t-1}^{j-1}$, the recurrent units are used to fuse them to generate the aggregated feature of the current time-step in the given recurrent layer $\mathbf{H}_{t}^{j}$, where j indexes layer number and t indexes the time-step. $\mathbf{H}_{t}^{0}$ is initialized as $\mathbf{F}_t$. In this fusion process, Gated recurrent units (GRU) [12] are used. With gate functions, the neuron chooses to read and reset at a time-step. This architecture updates and aggregates internal memory progressively, which facilitates its modeling of long-term temporal dynamics of sequential data. The formulations are presented as follows,

$$\begin{aligned}
\mathbf{H}_{t}^{j} &= \left(1-\mathbf{z}_{t}^{j}\right)\mathbf{H}_{t-1}^{j-1}+\mathbf{z}_{t}^{j}\widetilde{\mathbf{H}}_{t}^{j},\\
\widetilde{\mathbf{H}}_{t}^{j} &= \tanh\left(\mathbf{W}_{h}\mathbf{H}_{t}^{j-1}+\mathbf{U}_{h}\left(\mathbf{r}_{t}^{j}\odot\mathbf{H}_{t-1}^{j-1}\right)\right),\\
\mathbf{z}_{t}^{j} &= \sigma\left(\mathbf{W}_{z}\mathbf{H}_{t}^{j-1}+\mathbf{U}_{z}\mathbf{H}_{t-1}^{j-1}\right),\\
\mathbf{r}_{t}^{j} &= \mathrm{ReLU}\left(\mathbf{W}_{r}\mathbf{H}_{t}^{j-1}+\mathbf{U}_{r}\mathbf{H}_{t-1}^{j-1}\right),
\end{aligned} \tag{14}$$

where $\mathbf{H}_{t}^{j}$ is interpreted as the aggregated memory, representing the accumulated information at the t-th time-step from adjacent frames. $\mathbf{H}_{t}^{j}$ is also the output of the unit. $\mathbf{r}_{t}^{j}$ is the read gate, controlling the input information from adjacent frames to the current one. $\mathbf{z}_{t}^{j}$ is the update gate, deciding how much information of the current frame should be updated to the hidden state. $\widetilde{\mathbf{H}}_{t}^{j}$ is the new memory information generated at the t-th time-step.

*3) Context Selection **Gate** (**CS-Gate**):* To percept the context information in modeling temporal dynamics to benefit the joint spatial and temporal learning, we use a component to detect the context of rain frames explicitly, which further guides the successive spatial and temporal feature fusion. CS-Gate takes $\mathbf{H}_{t-1}^{j-1}$ and $\mathbf{H}_{t}^{j-1}$ as its input, and predicts $\hat{\alpha}_t$ as follows,

$$\begin{aligned}
\mathbf{f}_{t,0}^{j,d} &= \left[\mathbf{H}_{t}^{j-1},\mathbf{H}_{t-1}^{j-1}\right],\\
\mathbf{f}_{t,1}^{j,d} &= \mathbf{W}_{t,1}^{\mathrm{d}}*\mathbf{f}_{t,0}^{j,d}+\mathbf{b}_{t,1}^{j,d},\\
\mathbf{f}_{t,2}^{j,d} &= \mathbf{W}_{t,2}^{j,d}*\mathbf{f}_{t,1}^{j,d}+\mathbf{b}_{t,2}^{d},\\
\hat{\alpha}_t\left(k\right) &= \frac{\exp\left(\mathbf{f}_{t,2}^{d}\left(k\right)\right)}{\sum_{s=1,2,\ldots,S_t}\exp\left(\mathbf{f}_{t,2}^{d}\left(s\right)\right)},
\end{aligned} \tag{15}$$

where k indexes the feature map channel, which corresponds to the context variable, and S_t is the total number of that. In our implementation, $\hat{\alpha}_t$ aims to predict rain type indicator and motion segmentation as shown in Fig. 5.

4) Contextualized Fusion: To benefit the joint spatial temporal feature learning in different contexts, we enable to use several recurrent units at a given time-step of a recurrent layer. Thus, the aggregated feature $\mathbf{H}_{t}^{j}$ is extended to $\mathbf{H}_{t}^{j,s}$, where s indexes the context variable.

Given these features, the output of CS-Gate and the predicted probability of a context variable $\hat{\alpha}_t$, the final fused feature is calculated as follows,

$$\mathbf{H}_{t}^{j}=\sum_{s=1}^{S}\hat{\alpha}_t\left(s\right)\mathbf{H}_{t}^{j,s}. \tag{16}$$

5) Spatial Temporal Residue Fusion: After the last l-th recurrent layer, we then combine both temporally fused feature $\mathbf{H}_{t}^{l}$ and spatial feature $\mathbf{F}_t$ as follows,

$$\mathbf{M}_t=\mathbf{H}_{t}^{l}+\mathbf{F}_t. \tag{17}$$

6) Single-Frame Reconstruction (SF-Rect): SF-Rect aims to separate rain streaks based on only spatial features, which makes $\mathbf{F}_t$ good at distinguishing rain streaks and normal textures. The estimated negative rain streak layer and clean background frame are represented as follows,

$$\mathbf{r}_{t}^{s} = f_{\mathrm{sf}}\left(\mathbf{F}_t\right), \tag{18}$$

$$\hat{\mathbf{B}}_{t}^{s} = \hat{\mathbf{O}}_t+\mathbf{r}_{t}^{s}. \tag{19}$$

7) Multi-Frame Reconstruction (MF-Rect): MF-Rect aims to separate rain streaks or fill in missing rain occlusion regions based on temporal dynamics, which makes the network capable of modeling motions and temporal dynamics of background among frames. The estimated negative rain streak layer and clean background frame are represented as follows,

$$\mathbf{r}_{t}^{m} = f_{\mathrm{mf}}\left(\mathbf{M}_t\right), \tag{20}$$

$$\hat{\mathbf{B}}_{t}^{m} = \hat{\mathbf{O}}_t+\mathbf{r}_{t}^{m}. \tag{21}$$

8) Loss Function: As above-mentioned, let $\hat{\mathbf{B}}_{t}^{s}$, $\hat{\mathbf{B}}_{t}^{m}$ and $\hat{\alpha}_t$ denote the estimated background frame with only spatial features, the estimated background frame with both spatial and temporal features, and context type mask. Let $\mathbf{B}_t$ and α_t denote the ground-truth background frame and the degradation type mask. The loss function of the network includes three terms: context detection error, background estimation error based on only spatial features, and that based on both spatial and temporal features,

$$\begin{aligned}
l_{\mathrm{all}} &= \lambda_d l_{\mathrm{detect}}+\lambda_s l_{\mathrm{s\text{-}rect}}+\lambda_m l_{\mathrm{m\text{-}rect}},\\
l_{\mathrm{detect}} &= \sum_{t\in T}\left[\log\left(\sum_{k=1,2,\ldots,S_t}\exp\left(\mathbf{f}_{t,2}^{d}\left(k\right)\right)\right)-\alpha_t\right],\\
l_{\mathrm{s\text{-}rect}} &= \left\|\hat{\mathbf{B}}_{t}^{s}-\mathbf{B}_t\right\|_2^2,\\
l_{\mathrm{m\text{-}rect}} &= \left\|\hat{\mathbf{B}}_{t}^{m}-\mathbf{B}_t\right\|_2^2,
\end{aligned} \tag{22}$$

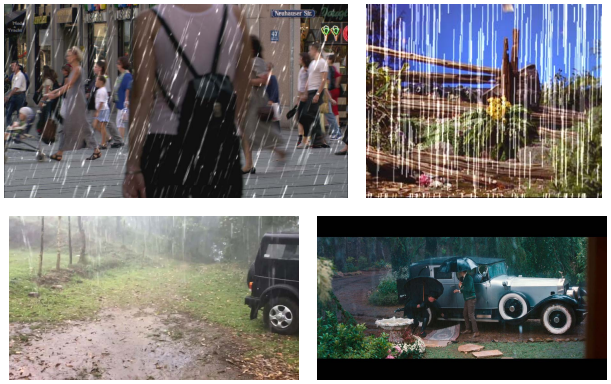


Fig. 7. Top left panel: one example of *RainSynLight25*. Top right panel: one example of *RainSynComplex25*. Bottom panel: two examples of *RainPractical10*.

where T is the full set of the time-step that is incorporated with rain removal context by dynamic routing. λ_d, λ_s, and λ_m are set to 0.001, 1, and 1, respectively.

V. Experimental Results

We perform extensive experiments to demonstrate the superiority of D3R-Net, as well as effectiveness of its each component. Due to the limited space, some results are presented in the supplementary material.

A. Datasets

We compare D3R-Net with state-of-the-art methods on a few benchmark datasets:

- *RainSynLight25*, which is synthesized by non-rain sequences with the rain streaks generated by the probabilistic model [21]. Compared with the original procedure in [21], we use a simplified approach. For a sampled location, we randomly select one streak from the streak database [21], transform it with a sampled direction (from $-50°$ to $50°$) and zoom it with a random scale (from 0.2 to 3). The parameters of directions and scales are consistent but with small-scale variations within a streak map. The used streaks vary from tiny drizzling to heavy rain storm and vertical rain to slash line.
- *RainSynComplex25*, which is synthesized by non-rain sequences with the rain streak generated by the probabilistic model [21], sharp line streaks [65] and sparkle noises.
- *RainPractical10*, ten rain video sequences we collected from practical scenes from Youtube website,[1] GIPHY[2] and movie clips.

Some examples of *RainSynLight25*, *RainSynComplex25*, and *RainPractical10* are provided in Fig. 7. Our synthesized training and testing data is from CIF testing sequences, HDTV sequences[3] and HEVC standard testing sequences.[4] The augmented video clips are synthesized based on BSD500 [42], with the artificially simulated motions, including rescaling and displacement. More information about training data and training details are provided in the supplementary material.

B. Comparison Methods

We compare D3R-Net with six state-of-the-art methods: discriminative sparse coding (DSC) [41], layer priors (LP) [37], joint rain detection and removal (JORDER) [65], deep detail network (DetailNet) [18], tensor-based video rain streaks removal (FastDeRain) [31], temporal correlation and low-rank matrix completion (TCLRM) [35]. DSC, LP, JORDER and DetailNet are single frame deraining methods. SE and TCLRM are video derainig methods. JORDER and DetailNet are deep-learning based methods.

For the experiments on synthesized data, five metrics Peak Signal-to-Noise Ratio (PSNR) [30], Structure Similarity Index (SSIM) [55], Visual Information Fidelity (VIF) [45], feature-similarity (FSIM) [69], and Universal image Quality Index (UQI) [54] are used as comparison criteria. Following previous works, we evaluate the results only in the luminance channel, since human visual system is more sensitive to luminance than chrominance information.

C. Quantitative Evaluation

Table I shows the results of different methods on *RainSynLight25* and *RainSynComplex25*. As observed, our method considerably outperforms other methods in terms of both PSNR and SSIM. The PSNR of D3R-Net is higher than that of JORDER, the state-of-the-art sinlge image rain removal method, with margins at more than 2.5dB and 6.5dB on *RainSynLight25* and *RainSynComplex25*, respectively. D3R-Net also obtains higher SSIM values than JORDER, with margins at about 0.0199 and 0.1968 on *RainSynLight25* and *RainSynComplex25*, respectively. Compared with SE and TCLRM, D3R-Net also achieves higher PSNR and SSIM. The gains of PSNR are more than 5dB and 8dB on *RainSynLight25* and *RainSynComplex25*, respectively. The gains of SSIM are more than 0.08 and 0.25 on *RainSynLight25* and *RainSynComplex25*, respectively.

D. Qualitative Evaluation

Figs. 8-9 show the results of synthetic images. It is clearly observed that, our D3R-Net produces the cleanest result with the least texture detail loss (least structure details remaining in estimated rain streak layers). Figs. 10-13 show the results of practical images. We here only present the zooming-in local results. Their corresponding full results are provided in the supplementary material.[5] TCLRM and D3R-Net remove the majority of rain streaks successfully. However, the result of TCLRM may contain artifacts in the area with large motions, as denoted by the red arrows. D3R-Net achieves superior performance in both removing rain streaks and avoiding artifacts.

E. Ablation Analysis on Network Architecture

We compare the results with different compositions of the proposed method. The results with two baseline RNNs are provided: bidirectional recurrent convolutional network (BRCN)

[1] https://www.youtube.com/
[2] https://giphy.com/
[3] https://media.xiph.org/video/derf/
[4] http://ftp.kw.bbc.co.uk/hevc/hm-10.0-anchors/bitstreams/
[5] http://www.icst.pku.edu.cn/struct/Projects/VideoRainRemoval/Supple.mp4

TABLE I

OBJECTIVE RESULTS AMONG DIFFERENT RAIN STREAK REMOVAL METHODS ON *RainSynLight25* (DENOTED BY *Light*) AND *RainSynComplex25* (DENOTED BY *Complex*)

Methods	Rain Images		DetailNet		TCLRM		JORDER	
Dataset	*Light*	*Complex*	*Light*	*Complex*	*Light*	*Complex*	*Light*	*Complex*
PSNR	23.69	14.67	25.72	16.50	28.77	17.31	30.37	20.20
SSIM	0.8058	0.4563	0.8572	0.5441	0.8693	0.4956	0.9235	0.6335
VIF	0.4184	0.2001	0.4225	0.2180	0.4714	0.1807	0.5124	0.2460
FSIM	0.8440	0.6450	0.8848	0.7012	0.9216	0.6916	0.9171	0.7419
UQI	0.9845	0.8467	0.9882	0.8695	0.9960	0.8862	0.9932	0.9560
Methods	LP		DSC		FastDeRain		D3R-Net	
Dataset	*Light*	*Complex*	*Light*	*Complex*	*Light*	*Complex*	*Light*	*Complex*
PSNR	27.09	17.65	25.63	17.33	29.42	19.25	**32.96**	**27.03**
SSIM	0.8566	0.5364	0.8328	0.5036	0.8683	0.5385	**0.9434**	**0.8303**
VIF	0.5135	0.2478	0.4293	0.2109	0.4995	0.2479	**0.6555**	**0.3822**
FSIM	0.8908	0.7030	0.8736	0.6765	0.9129	0.7351	**0.9660**	**0.8891**
UQI	0.9922	0.8878	0.9889	0.9058	0.9964	0.9051	**0.9985**	**0.9875**

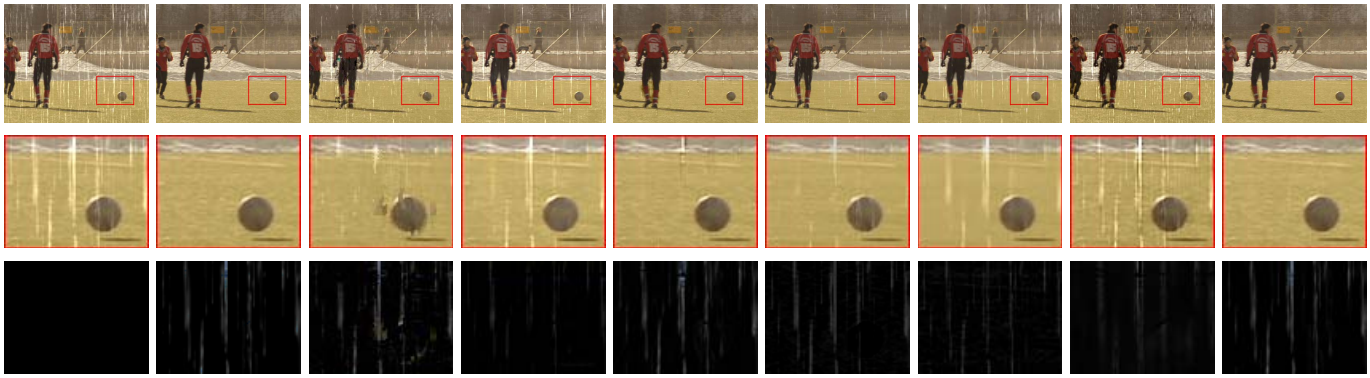


Fig. 8. Results of different methods on an example of *RainSynLight25*. From top to bottom: whole image, local regions of the estimated background layer, and local regions of the estimated rain streak layer. (a) Rain image. (b) Ground truth. (c) TCLRM. (d) DetailNet. (e) JORDER. (f) FastDeRain. (g) LP. (h) DSC. (i) D3R-Net.

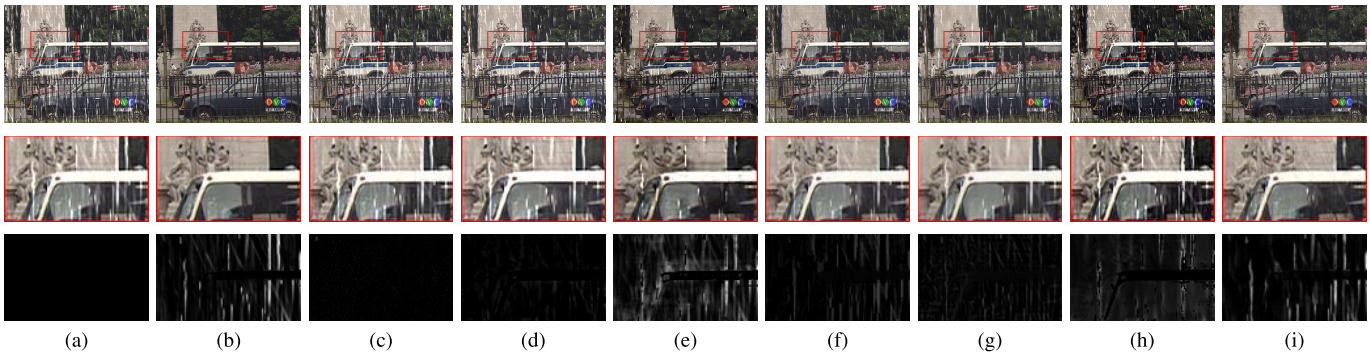


Fig. 9. Results of different methods on an example of *RainSynComplex25*. From top to bottom: whole image, local regions of the estimated background layer, and local regions of the estimated rain streak layer. (a) Rain image. (b) Ground truth. (c) TCLRM. (d) DetailNet. (e) JORDER. (f) FastDeRain. (g) LP. (h) DSC. (i) D3R-Net.

and GRU. JORDER is the single frame version. B-R denotes the raw BRCN version without temporal residue learning. B denotes the BRCN with temporal residue learning. B+R is the BRCN embedded with rain type in a dynamic routing way. B+M is a BRCN embedded with motion segmentation in a dynamic routing way. B+R+M is incorporated with both rain type and motion segmentation. G-R denotes the raw GRU without temporal residue learning. G denotes the GRU network with temporal residue learning. G+R is the GRU embedded with rain type in a dynamic routing way. G+M is a GRU embedded with motion segmentation in a dynamic routing way. G+R+M is incorporated with both rain type and motion segmentation.

The comparison results are presented in Table II and Table III. The comparison between JORDER and B-R, and that between JORDER and G-R show the importance of joint modeling spatial and temporal redundancy. From JORDER to B-R and G-R, the performance is largely improved with

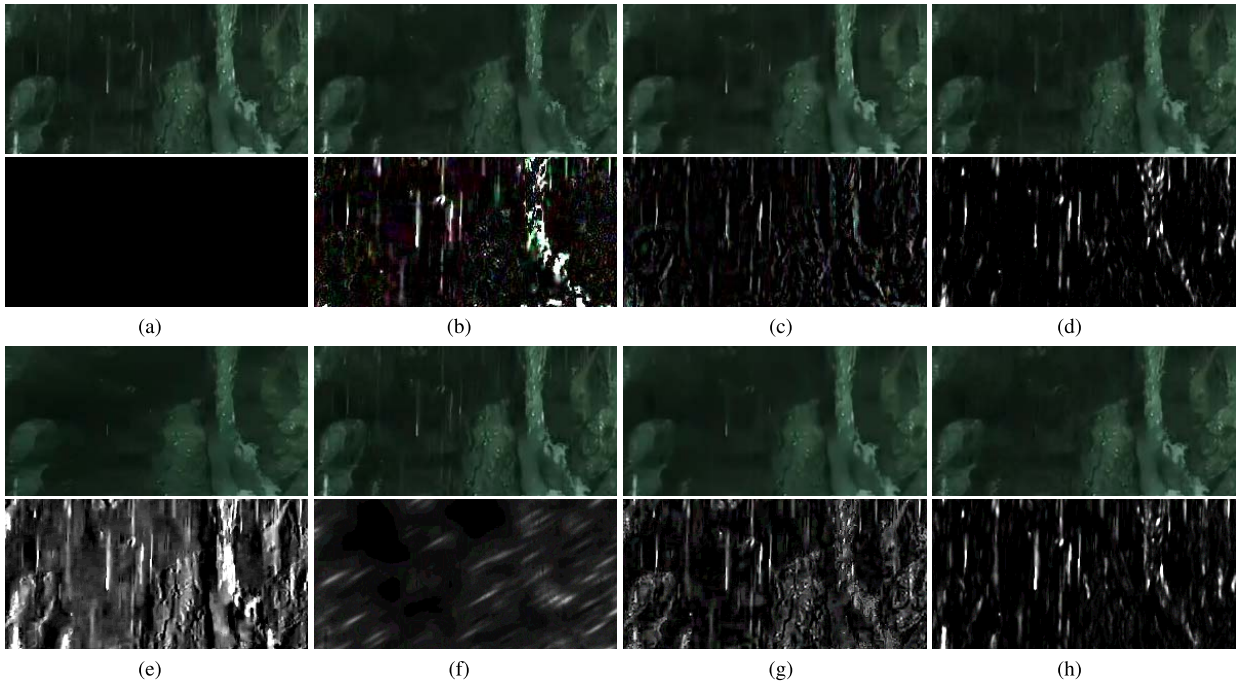


Fig. 10. Results of different methods on practical images. Their full resolution results are provided in the supplementary material. (a) Rain image. (b) TCLRM. (c) DetailNet. (d) JORDER. (e) FastDeRain. (f) DSC. (g) LP. (h) D3R-Net.

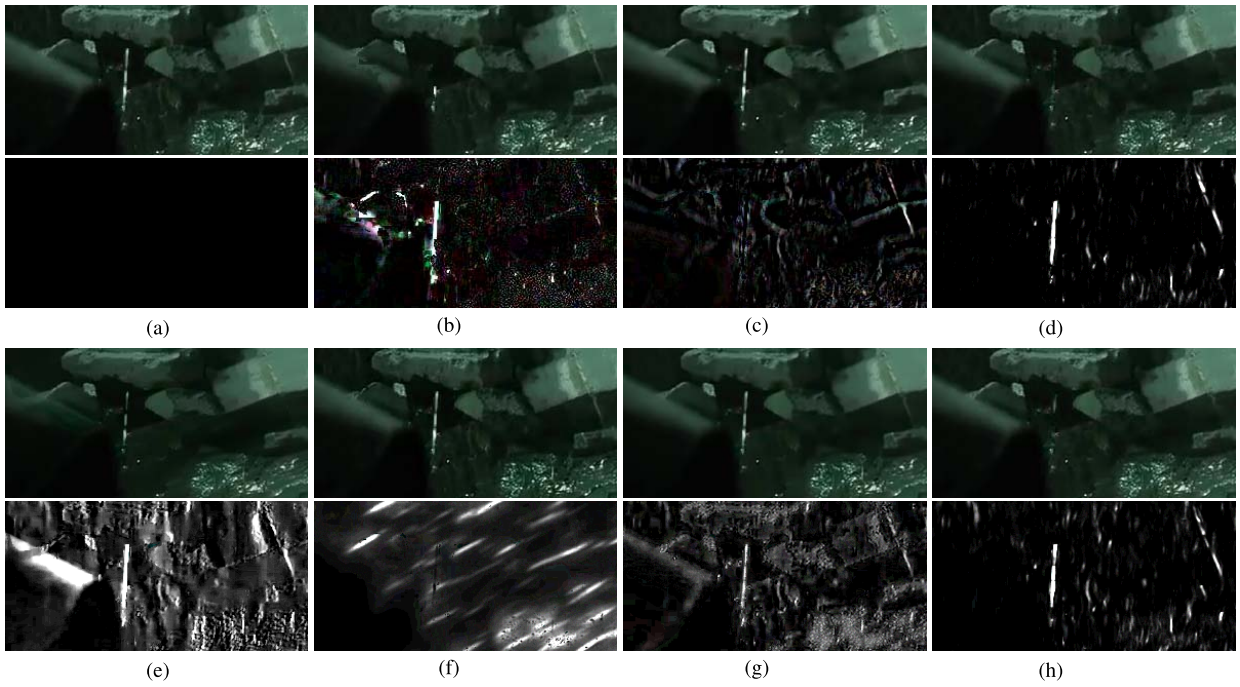


Fig. 11. Results of different methods on practical images. Their full resolution results are provided in the supplementary material. (a) Rain image. (b) TCLRM. (c) DetailNet. (d) JORDER. (e) FastDeRain. (f) DSC. (g) LP. (h) D3R-Net.

gains of 5.48dB in PSNR, 0.1434 in SSIM and 6.18 dB in PSNR, 0.1762 in SSIM, respectively. The usage of spatial temporal residue learning (B and G) leads to higher metric scores, with gains of 0.20dB in PSNR, 0.0167 in SSIM and 0.32 dB in PSNR, 0.0135 in SSIM, compared with B-R and G-R respectively. It can be also observed that, embedding motion segmentation and rain type in the dynamical routing way can boost the performance and the joint incorporation provides the best evaluation performance. Note that, for a fair comparison, we control that the parameter number of raw BRCN is greater than that of BRCN embedded with rain type and motion segmentation and that the parameter number of raw GRU is greater than that of GRU embedded with rain type and motion segmentation. The channel number of the recurrent

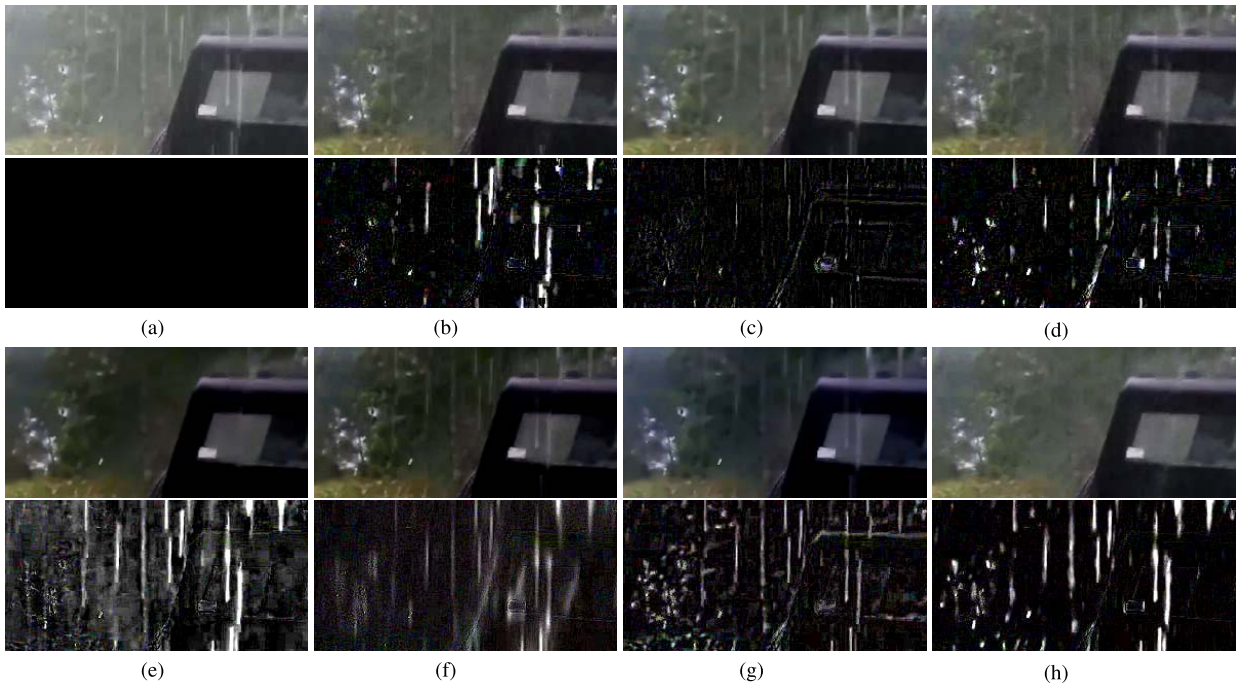


Fig. 12. Results of different methods on practical images. Their full resolution results are provided in the supplementary material. (a) Rain image. (b) TCLRM. (c) DetailNet. (d) JORDER. (e) FastDeRain. (f) DSC. (g) LP. (h) D3R-Net.

TABLE II
OBJECTIVE EVALUATION RESULTS AMONG DIFFERENT VERSIONS OF THE PROPOSED METHOD WITH BRCN ARCHITECTURE ON *RainSynComplex25*

Methods	JORDER	B-R	B	B+M	B+R	B+R+M
PSNR	20.20	25.68	25.88	26.48	26.44	**26.77**
SSIM	0.6335	0.7769	0.7936	0.8158	0.8140	**0.8270**
VIF	0.2460	0.3159	0.3312	0.3583	0.3574	**0.3780**
FSIM	0.7419	0.8589	0.8677	0.8780	0.8768	**0.8853**
UQI	0.9560	0.9817	0.9827	0.9843	0.9842	**0.9846**

TABLE III
OBJECTIVE EVALUATION RESULTS AMONG DIFFERENT VERSIONS OF THE PROPOSED METHOD WITH GRU ARCHITECTURE ON *RainSynComplex25*

Methods	JORDER	G-R	G	G+M	G+R	G+R+M
PSNR	20.20	26.38	26.70	26.81	26.85	**27.03**
SSIM	0.6335	0.8097	0.8232	0.8271	0.8282	**0.8303**
VIF	0.2460	0.3498	0.3683	0.3798	0.3791	**0.3822**
FSIM	0.7419	0.8758	0.8829	0.8872	0.8881	**0.8891**
UQI	0.9560	0.9850	0.9862	0.9865	0.9871	**0.9875**

layers of raw BRCN and GRU is 64 and that embedded with rain type and motion segmentation is 16. The comparison of B+R+M and G further demonstrates the effectiveness of the proposed dynamical routing context embedding method. B+R+M achieves superior performance with less parameters.

F. Computer Vision Applications

Our D3R-Net not only significantly improves the visibility but also enhances the performance of successive computer vision system. Fig. 14 presents the optical flow estimation of synthesized rain frames, non-rain frames and the derained results of our D3R-Net. It is demonstrated that, the existence of rain streaks contaminates the optical flow estimation. Comparatively, the optical flow estimation of the derained results by D3R-Net is more accurate, visually similar to that of ground truth non-rain frames.

G. Running Time Comparison

Table IV compares the running time of several state-of-the-art methods. All baseline methods are implemented in MATLAB. Our methods are implemented on the Caffe's Matlab wrapper. DetailNet, JORDER, FastDeRain and D3R-Net are implemented on GPU. LP, DSC and TCLRM are implemented on CPU. We evaluate the running time of all algorithms with the following machine configuration: Intel Core(TM) i7-6850K @ 3.60GHz, 64 GB memory and TITAN GeForce GTX 1080. Our D3R-Net obtains comparable running time to FastDeRain and JORDER, and runs much faster than TCLRM, LP and DSC. In general, our methods in GPU are capable of dealing with a 500×500 rain image less than 5s.

H. Performance and Parameter Analysis

We also provide the objective results and parameter numbers of deep learning-based methods in Table V. It is observed that, compared with the performance improvement (0.81 dB and 1.86 dB in PSNR as well as 0.0263 and 0.1023 in SSIM) from JORDER to DetailNet with a cost of more than 5 times additional parameters, the performance improvement (2.59 dB and 6.83 dB in PSNR as well as 0.0199 and 0.1968 in SSIM) from JORDER to D3R-Net is quite efficient and economical. It is showed that, our D3R-Net uses more

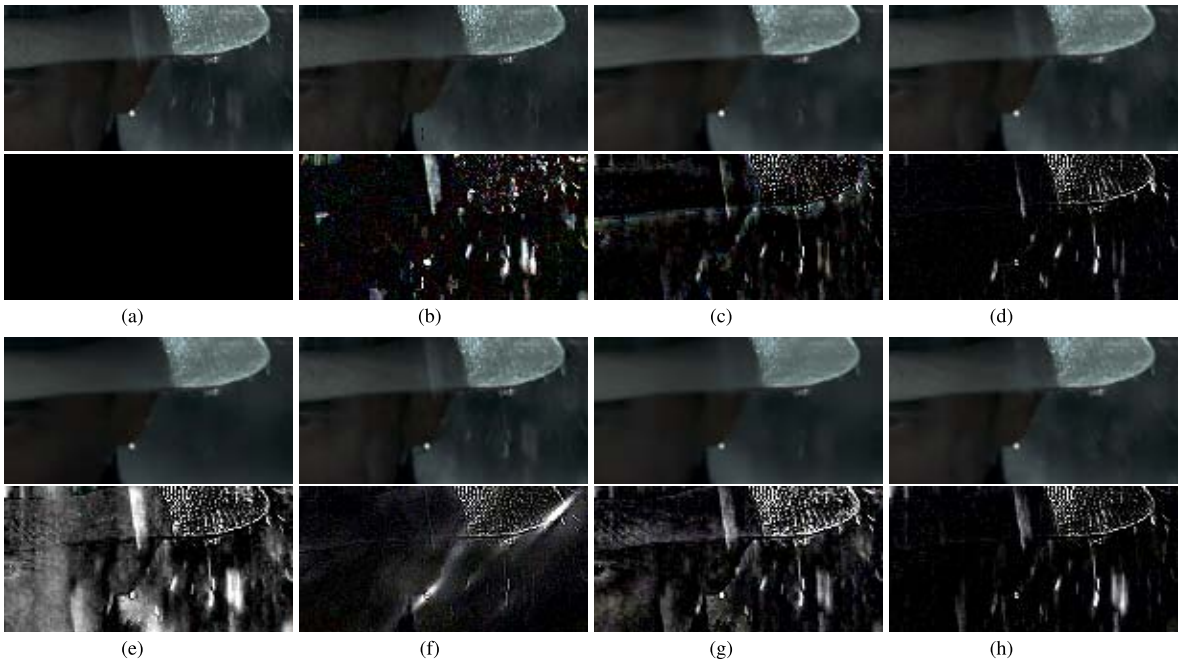


Fig. 13. Results of different methods on practical images. Their full resolution results are provided in the supplementary material. (a) Rain image. (b) TCLRM. (c) DetailNet. (d) JORDER. (e) FastDeRain. (f) DSC. (g) LP. (h) D3R-Net.

TABLE IV

OBJECTIVE RESULTS AND PARAMETER ANALYSIS AMONG DIFFERENT RAIN STREAK REMOVAL METHODS ON *RainSynLight25* (DENOTED BY *Light*) AND *RainSynComplex25* (DENOTED BY *Complex*)

Methods	Rain Images		DetailNet		JORDER		D3R-Net	
Dataset	*Light*	*Complex*	*Light*	*Complex*	*Light*	*Complex*	*Light*	*Complex*
PSNR	23.69	14.67	29.56	18.34	30.37	20.20	32.96	27.03
SSIM	0.8058	0.4563	0.8972	0.5312	0.9235	0.6335	0.9434	0.8303
VIF	0.4184	0.2001	0.4985	0.2185	0.5124	0.2460	0.6555	0.3822
FSIM	0.8440	0.6450	0.9082	0.7328	0.9171	0.7419	0.9660	0.8891
UQI	0.9845	0.8467	0.9912	0.9340	0.9932	0.9560	0.9985	0.9875
Parameter Number	-		57,369		369,792		543,280	

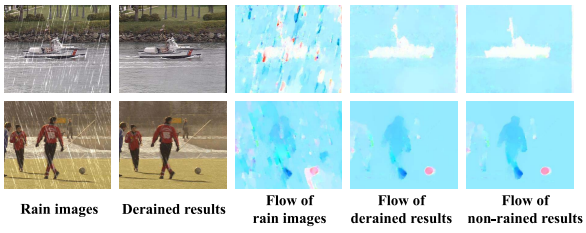


Fig. 14. Evaluation of optical flow estimation on synthetic rain images and derained results.

TABLE V

THE TIME COMPLEXITY (IN SECONDS) OF D3R-NET COMPARED WITH STATE-OF-THE-ART METHODS

Scale	–	DetailNet	TCLRM	JORDER
80×80	–	0.05	2.31	0.11
500×500	–	0.93	64.14	1.46
Scale	LP	DSC	FastDeRain	D3R-Net
80×80	35.97	14.32	0.09	0.13
500×500	2708.20	611.91	2.71	3.06

parameters, however, significant gains are indeed achieved. It is worthwhile to introduce more parameters to model the temporal dependencies between frames and incorporate the detected video context in D3R-Net.

VI. CONCLUSION

In this paper, we proposed a hybrid rain model to depict both rain streaks and occlusions. Then, a **D**ynamic **R**outing **R**esidue **R**ecurrent **Net**work (**D3R-Net**) was built to seamlessly integrate context variable estimations, and a rain removal based on both spatial appearance feature and temporal coherence. The rain type indicator and motion segmentation were embedded into D3R-Net in a dynamic routing way, flexible to be extended to incorporate other context information. Extensive experiments on a series of synthetic and practical videos with rain streaks verified the superiority of the proposed method over previous state-of-the-art methods.

REFERENCES

[1] P. C. Barnum, S. Narasimhan, and T. Kanade, "Analysis of rain and snow in frequency space," *Int. J. Comput. Vis.*, vol. 86, nos. 2–3, pp. 256–274, Jan. 2010.

[2] J. Bossu, N. Hautiére, and J.-P. Tarel, "Rain or snow detection in image sequences through use of a histogram of orientation of streaks," *Int. J. Comput. Vis.*, vol. 93, no. 3, pp. 348–367, 2011.

[3] N. Brewer and N. Liu, "Using the shape characteristics of rain to identify and remove rain from video," in *Proc. Joint IAPR Int. Workshops SPR SSPR*, 2008, pp. 451–458.

[4] L. Cavigelli, P. Hager, and L. Benini, "CAS-CNN: A deep convolutional neural network for image compression artifact suppression," in *Proc. Int. Conf. Neural Netw. (IJCNN)*, May 2017, pp. 752–759.

[5] Y. Chang, L. Yan, and S. Zhong, "Transformed low-rank model for line pattern noise removal," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1735–1743.

[6] J. Chen and L.-P. Chau, "Rain removal from dynamic scene based on motion segmentation," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2013, pp. 2139–2142.

[7] J. Chen and L.-P. Chau, "A rain pixel recovery algorithm for videos with highly dynamic scenes," *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1097–1104, Mar. 2014.

[8] J. Chen, J. Chen, H. Chao, and M. Yang, "Image blind denoising with generative adversarial network based noise modeling," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3155–3164.

[9] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, Jun. 2017.

[10] Y.-L. Chen and C.-T. Hsu, "A generalized low-rank appearance model for spatio-temporally correlated rain streaks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1968–1975.

[11] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. (Dec. 2014). "Empirical evaluation of gated recurrent neural networks on sequence modeling." [Online]. Available: https://arxiv.org/abs/1412.3555

[12] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS) Workshop Deep Learn.*, Montreal, QC, Canada, Dec. 2014.

[13] C. Dong, Y. Deng, C. Change Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 576–584.

[14] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 184–199.

[15] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[16] D. Eigen, D. Krishnan, and R. Fergus, "Restoring an image taken through a window covered with dirt or rain," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 633–640.

[17] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2944–2956, Jun. 2017.

[18] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3855–3863.

[19] K. Garg and S. K. Nayar, "Detection and removal of rain from videos," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, Jun./Jul. 2004, p. I-528.

[20] K. Garg and S. K. Nayar, "When does a camera see rain?" in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2, Oct. 2005, pp. 1067–1074.

[21] K. Garg and S. K. Nayar, "Photorealistic rendering of rain streaks," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 996–1002, Jul. 2006.

[22] K. Garg and S. K. Nayar, "Vision and rain," *Int. J. Comput. Vis.*, vol. 75, no. 1, pp. 3–27, 2007.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[24] Y. Hu, J. Liu, W. Yang, S. Deng, L. Zhang, and Z. Guo, "Real-time deep image super-resolution via global context aggregation and local queue jumping," in *Proc. IEEE Vis. Commun. Image Process.*, Dec. 2017, pp. 1–4.

[25] Y. Hu, W. Yang, S. Xia, W.-H. Cheng, and J. Liu, "Enhanced intra prediction with recurrent neural network in video coding," in *Proc. Data Compress. Conf.*, Mar. 2018, p. 413.

[26] Y. Hu, W. Yang, S. Xia, and J. Liu, "Optimized recurrent network for intra prediction in video coding," in *Proc. IEEE Vis. Commun. Image Process (VCIP)*, to be published.

[27] D.-A. Huang, L.-W. Kang, Y.-C. F. Wang, and C.-W. Lin, "Self-learning based image decomposition with applications to single image denoising," *IEEE Trans. Multimedia*, vol. 16, no. 1, pp. 83–93, Jan. 2014.

[28] D.-A. Huang, L.-W. Kang, M.-C. Yang, C.-W. Lin, and Y.-C. F. Wang, "Context-aware single image rain removal," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2012, pp. 164–169.

[29] Y. Huang, W. Wang, and L. Wang, "Bidirectional recurrent convolutional networks for multi-frame super-resolution," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2016, pp. 235–243.

[30] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electron. Lett.*, vol. 44, no. 13, pp. 800–801, Jun. 2008.

[31] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and Y. Wang, "A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4057–4066.

[32] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1742–1755, Apr. 2012.

[33] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1646–1654.

[34] J.-H. Kim, C. Lee, J.-Y. Sim, and C.-S. Kim, "Single-image deraining using an adaptive nonlocal means filter," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2013, pp. 914–917.

[35] J.-H. Kim, J.-Y. Sim, and C.-S. Kim, "Video deraining and desnowing using temporal correlation and low-rank matrix completion," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2658–2670, Sep. 2015.

[36] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Fast and accurate image super-resolution with deep laplacian pyramid networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[37] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain streak removal using layer priors," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2736–2744.

[38] J. Liu, W. Yang, S. Yang, and Z. Guo, "Erase or fill? Deep joint recurrent rain removal and reconstruction in videos," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3233–3242.

[39] P. Liu, J. Xu, J. Liu, and X. Tang, "Pixel based temporal analysis using chromatic property for removing rain from videos," *Comput. Inf. Sci.*, vol. 2, no. 1, pp. 53–60, 2009.

[40] K. G. Lore, A. Akintayo, and S. Sarkar, "LLNet: A deep autoencoder approach to natural low-light image enhancement," *Pattern Recognit.*, vol. 61, pp. 650–662, Jan. 2017.

[41] Y. Luo, Y. Xu, and H. Ji, "Removing rain from a single image via discriminative sparse coding," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3397–3405.

[42] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2. Jul. 2001, pp. 416–423.

[43] W. Ren, J. Tian, Z. Han, A. Chan, and Y. Tang, "Video desnowing and deraining based on matrix decomposition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2838–2847.

[44] V. Santhaseelan and V. K. Asari, "Utilizing local phase information to remove rain from video," *Int. J. Comput. Vis.*, vol. 112, no. 1, pp. 71–89, Mar. 2015.

[45] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 430–444, Feb. 2006.

[46] S. Starik and M. Werman, "Simulation of rain in videos," in *Proc. Texture Workshop ICCV*, Jun. 2003, pp. 406–409.

[47] S.-H. Sun, S.-P. Fan, and Y.-C. F. Wang, "Exploiting image structural similarity for single image rain removal," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2014, pp. 4482–4486.

[48] X. Tao, H. Gao, R. Liao, J. Wang, and J. Jia, "Detail-revealing deep video super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 4472–4480.

[49] G. Toderici *et al.*, "Full resolution image compression with recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5435–5443.

[50] A. K. Tripathi and S. Mukhopadhyay, "A probabilistic approach for detection and removal of rain from videos," *IETE J. Res.*, vol. 57, no. 1, pp. 82–91, 2011.

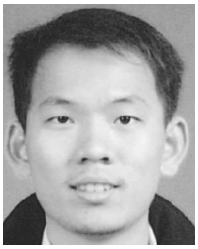
[51] A. K. Tripathi and S. Mukhopadhyay, "Video post processing: Low-latency spatiotemporal approach for detection and removal of rain," *IET Image Process.*, vol. 6, no. 2, pp. 181–196, Mar. 2012.

[52] C. Wang, C. Xu, C. Wang, and D. Tao, "Perceptual adversarial networks for image-to-image transformation," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 4066–4079, Aug. 2018.

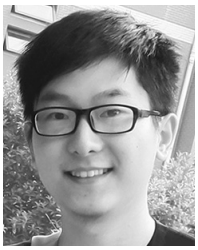
[53] W. Wang, C. Wei, W. Yang, and J. Liu, "GLADNet: Low-light enhancement network with global awareness," in *Proc. IEEE Conf. Autom. Face Gesture Recognit. (FG)*, May 2018, pp. 751–755.
[54] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Process. Lett.*, vol. 9, no. 3, pp. 81–84, Mar. 2002.
[55] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[56] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 370–378.
[57] C. Wei, W. Wang, W. Yang, and J. Liu, "Deep retinex decomposition for low-light enhancement," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Sep. 2018, p. 1.
[58] W. Wei, L. Yi, Q. Xie, Q. Zhao, D. Meng, and Z. Xu, "Should we encode rain streaks in video as deterministic or stochastic?" in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2516–2525.
[59] W. Yang, J. Feng, G. Xie, J. Liu, Z. Guo, and S. Yan, "Video super-resolution based on spatial-temporal recurrent residual networks," *Comput. Vis. Image Understand.*, vol. 168, pp. 79–92, Mar. 2018.
[60] S. Xia, W. Yang, Y. Hu, S. Ma, and J. Liu, "A group variational transformation neural network for fractional interpolation of video coding," in *Proc. Data Compress. Conf.*, Mar. 2018, pp. 127–136.
[61] S. Xia, W. Yang, J. Liu, and Z. Guo, "Dual recovery network with online compensation for image super-resolution," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2018, pp. 1–5.
[62] X. Xue, X. Jin, C. Zhang, and S. Goto, "Motion robust rain detection and removal from videos," in *Proc. IEEE 14th Int. Workshop Multimedia Signal Process. (MMSP)*, Sep. 2012, pp. 170–174.
[63] W. Yang, S. Deng, Y. Hu, J. Xing, and J. Liu, "Real-time deep video spatial resolution upconversion system (STRUCT++ Demo)," in *Proc. ACM Multimedia Conf.*, Oct. 2017, pp. 1255–1256.
[64] W. Yang *et al.*, "Deep edge guided recurrent residual learning for image super-resolution," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5895–5907, Dec. 2017.
[65] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Deep joint rain detection and removal from a single image," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1357–1366.
[66] W. Yang, S. Xia, J. Liu, and Z. Guo, "Reference guided deep super-resolution via manifold localized external compensation," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.
[67] H. Zhang and V. M. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 695–704.
[68] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.
[69] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, Aug. 2011.
[70] X. Zhang, H. Li, Y. Qi, W. K. Leow, and T. K. Ng, "Rain removal in video by combining temporal and chromatic properties," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2006, pp. 461–464.
[71] X. Zhang, W. Yang, Y. Hu, and J. Liu, "DMCNN: Dual-domain multi-scale convolutional neural network for compression artifacts removal," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2018, pp. 390–394.
[72] L. Zhu, C.-W. Fu, D. Lischinski, and P.-A. Heng, "Joint bi-layer optimization for single-image rain streak removal," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2545–2553.

Jiaying Liu (S'08–M'10–SM'17) received the B.E. degree in computer science from Northwestern Polytechnic University, Xi'an, China, and the Ph.D. degree (Hons.) in computer science from Peking University, Beijing, China, in 2005 and 2010, respectively. She is currently an Associate Professor with the Institute of Computer Science and Technology, Peking University. She has authored over 100 technical articles in refereed journals and proceedings, and she holds 28 granted patents. Her current research interests include image/video processing, compression, and computer vision.

She was a Visiting Scholar with the University of Southern California, Los Angeles, from 2007 to 2008. She was a Visiting Researcher at Microsoft Research Asia in 2015 supported by the Star Track for Young Faculties. She has also served as a TC Member in the IEEE CAS-MSA/EOT and APSIPA IVM, and an APSIPA Distinguished Lecturer from 2016 to 2017. She is a CCF Senior Member.



Wenhan Yang (S'17) received the B.S. and Ph.D. (Hons.) degrees in computer science from Peking University, Beijing, China, in 2012 and 2018, respectively. He was a Visiting Scholar with the National University of Singapore from 2015 to 2016. He is currently a Post-Doctoral Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore. His current research interests include deep-learning-based image processing, bad weather restoration, related applications, and theories.



Shuai Yang received the B.S. degree in computer science from Peking University, Beijing, China, in 2015, where he is currently pursuing the Ph.D. degree with the Institute of Computer Science and Technology.

His current research interests include image inpainting, depth map enhancement, and image stylization.

Zongming Guo (M'09) received the B.S. degree in mathematics and the M.S. and Ph.D. degrees in computer science from Peking University, Beijing, China, in 1987, 1990, and 1994, respectively.

He is currently a Professor with the Institute of Computer Science and Technology, Peking University. His current research interests include video coding, processing, and communication.

Dr. Guo is an Executive Member of the China Society of Motion Picture and Television Engineers. He was a recipient of the First Prize of the State Administration of Radio Film and Television Award in 2004, the First Prize of the Ministry of Education Science and Technology Progress Award in 2006, the Second Prize of the National Science and Technology Award in 2007, and the Wang Xuan News Technology Award and the Chia Tai Teaching Award in 2008. He received the Government Allowance granted by the State Council in 2009. He received the Distinguished Doctoral Dissertation Advisor Award from Peking University in 2012 and 2013.