

Deep Joint Rain Detection and Removal from a Single Image

Wenhan Yang¹, Robby T. Tan^{2,3}, Jiashi Feng², Jiaying Liu¹, Zongming Guo^{1*}, and Shuicheng Yan^{4,2}

¹Institute of Computer Science and Technology, Peking University, Beijing, P.R. China

²National University of Singapore, ³Yale-NUS College, ⁴360 AI Institute

Abstract

In this paper, we address a rain removal problem from a single image, even in the presence of heavy rain and rain streak accumulation. Our core ideas lie in our new rain image model and new deep learning architecture. We add a binary map that provides rain streak locations to an existing model, which comprises a rain streak layer and a background layer. We create a model consisting of a component representing rain streak accumulation (where individual streaks cannot be seen, and thus visually similar to mist or fog), and another component representing various shapes and directions of overlapping rain streaks, which usually happen in heavy rain. Based on the model, we develop a multi-task deep learning architecture that learns the binary rain streak map, the appearance of rain streaks, and the clean background, which is our ultimate output. The additional binary map is critically beneficial, since its loss function can provide additional strong information to the network. To handle rain streak accumulation (again, a phenomenon visually similar to mist or fog) and various shapes and directions of overlapping rain streaks, we propose a recurrent rain detection and removal network that removes rain streaks and clears up the rain accumulation iteratively and progressively. In each recurrence of our method, a new contextualized dilated network is developed to exploit regional contextual information and to produce better representations for rain detection. The evaluation on real images, particularly on heavy rain, shows the effectiveness of our models and architecture.

1. Introduction

Restoring rain images is important for many computer vision applications in outdoor scenes. Rain degrades visibility significantly and causes many computer vision sys-

*Corresponding author. This work was supported by National Natural Science Foundation of China under contract No. 61472011 and National Key Technology R&D Program of China under Grant 2015AA011605, with additional support by the State Scholarship Fund from the China Scholarship Council.



Figure 1. An example result of our proposed rain removal method that removes heavy rain streaks and enhances the visibility significantly. Top: The raw image with rain streaks (left) and the output image of our method (right). Bottom: closer looks at specific regions (blue and red) for more details.

tems to likely fail. Generally, rain introduces a few types of visibility degradation. Raindrops obstruct, deform and/or blur the background scenes. Distant rain streaks accumulate and generate atmospheric veiling effects similar to mist or fog, which severely reduce the visibility by scattering light out and into the line of sight. Nearby rain streaks exhibit strong specular highlights that occlude background scenes. These rain streaks can have various shapes and directions, particularly in heavy rain, causing severe visibility degradation.

In the past decades, many researchers have devoted their attention to solving the problem of restoring rain images. Some focus on rain image recovery from video sequences [3, 4, 5, 9, 13, 14, 15, 16, 44]. Others focus on rain removal from the single image, by regarding the rain streak removal problem as a signal separation problem [23, 18, 33, 9, 28], or by relying on nonlocal mean smoothing [24]. While there are varying degrees of success, majority of existing methods suffer from several limitations:

- Due to the intrinsic overlapping between rain streaks and background texture patterns, most methods tend to remove texture details in non-rain regions, leading to over-smoothing the regions.
- The degradation of rain is complex, and the existing rain model widely used in previous methods [23, 9]

is insufficient to cover some important factors in real rain images, such as the atmospheric veils due to rain streak accumulation, and different shapes or directions of streaks.

- The basic operation of many existing algorithms is on a local image patch or a limited receptive field (a limited spatial range). Thus, spatial contextual information in larger regions, which has been proven to be useful for rain removal [19], is rarely used.

Considering these limitations, our goal is to develop a novel rain model that is more capable of describing various rain conditions in real scenes, including rain streak accumulation and heavy rain, and then, use them to design an effective deep learning architecture. Here, we focus on a single input image.

To achieve the goal, we explore the possible rain models and deep learning architectures that can effectively restore rain images even in the presence of heavy rain. Our ideas are as follows. First, we introduce novel region-dependent rain models. In the models, we use a rain-streak binary map, where '1' indicates the presence of individually visible rain streaks in the pixels, and '0' otherwise. We also model the appearance of rain streak accumulation, and the various shapes and directions of overlapping streaks, to simulate heavy rain.

Second, based on the introduced model, we construct a deep network that jointly detects and removes rain. Rain streak regions are detected automatically and are used to constrain the rain removal. With this, our network is capable of performing an adaptive operation on rain and non-rain regions, preserving richer details.

Third, to retrieve more contextual information, we propose a contextualized dilated network to enlarge the receptive field. In this network, the features are extracted and refined progressively by aggregating information from several parallel convolutions with different dilated factors.

Finally, to restore images captured in the environment with both rain accumulation and various rain streak directions, we propose a recurrent rain detection and removal network that progressively removes rain streaks. Extensive experiments and evaluations demonstrate that our method outperforms state-of-the-art methods on both synthesized data and real data. Particularly for some heavy rain images, our method achieves considerably good results.

Hence, our contributions are:

1. The first method to model the rain-streak binary mask, and also to model the atmospheric veils due to rain streak accumulation as well as various shapes and directions of overlapping rain streaks. This enables us to synthesize more similar data to real rain images for the network training.
2. The first method to jointly detect and remove rains from single images. With the additional information of detected rain regions, our rain removal achieves better performance.
3. The first rain removal method that uses a contextualized dilated network to obtain more context while preserving rich local details.
4. The first method that addresses heavy rain by introducing a recurrent rain detection and removal network, where it removes rain progressively, enabling us to obtain good results even in significantly complex cases.

Our training and testing data, as well as our codes, will be publicly available ¹.

2. Related Work

Compared with the video based deraining problem, the single image based problem is more ill-posed, due to the lack of temporal information. Some single-image based rain removal methods regard the problem as a layer separation problem. Huang *et al.* [23] attempt to separate the rain streaks from the high frequency layer by sparse coding, with a learned dictionary from the HOG features. However, the capacity of the morphological component analysis, the layer separation, and learned dictionary are limited. Thus, it usually causes over-smoothness of the background. In [9], a generalized low rank model is proposed, where the rain streak layer is assumed to be low rank. Kim *et al.* [25] first detect rain streaks and then remove them with the nonlocal mean filter. Luo *et al.* [28] propose a discriminative sparse coding method to separate rain streaks from background images. A recent work [26] exploits the Gaussian mixture models to separate the rain streaks, achieving the state-of-the-art performance, however, still with slightly smooth background.

In recent years, deep learning-based image processing applications emerged with promising performance. These applications include denoising [35, 6, 7, 21, 1], completion [38], super-resolution [11, 12, 10, 31], deblurring [32], deconvolution [39] and style transfer [17, 40], *etc.* There are also some recent works on bad weather restoration or image enhancement, such as dehazing [8, 37], raindrop and dirt removal [13] and light enhancement [27]. Besides, with the superior modeling capacity than shallow models, DL-based methods begin to solve harder problems, such as blind image denoising [43]. In this paper, we use deep learning to jointly detect and remove rain.

3. Region-Dependent Rain Image Model

We briefly discuss the commonly used rain model, and then generalize it to explicitly include a rain-streak binary

¹http://www.icst.pku.edu.cn/struct/Projects/joint_rain_removal.html

map. Subsequently, we introduce a novel rain model that captures rain streak accumulation (atmospheric veils) and rain streaks that have various shapes and directions, which are absent in the existing rain models.

3.1. Rain Image Formation

The widely used rain model [26, 28, 19] is expressed as:

$$\mathbf{O} = \mathbf{B} + \tilde{\mathbf{S}}, \quad (1)$$

where \mathbf{B} is the background layer, and $\tilde{\mathbf{S}}$ is the rain streak layer. \mathbf{O} is the input image with rain streaks. Based on this model, rain removal is regarded as a two-signal separation problem, namely: Given the observation \mathbf{O} , how to separate the background \mathbf{B} from the rain streak $\tilde{\mathbf{S}}$ by relying on their unique characteristics. Existing rain removal methods using Eq. (1) suffer from the following two deficiencies. First, $\tilde{\mathbf{S}}$ in an image can have heterogeneous density, meaning some regions can have more rain streaks than other regions. For this case, it is hard to model $\tilde{\mathbf{S}}$ with a uniform sparsity assumption, which is needed for most of existing sparsity-based methods. Second, solving the signal separation problem in Eq. (1) without distinguishing the rain and non-rain regions will cause over-smoothness on the non-rain regions. The main cause for these two problems lies in the intrinsic complexity of modelling $\tilde{\mathbf{S}}$. In Eq. (1), $\tilde{\mathbf{S}}$ needs to constitute both streak locations and intensity contributions to the pixel values, which are not explored in existing methods.

To overcome these drawbacks, we first propose a generalized rain model as follows:

$$\mathbf{O} = \mathbf{B} + \mathbf{S}\mathbf{R}, \quad (2)$$

which includes a new region-dependent variable \mathbf{R} to indicate the locations of individually visible rain streaks. Here, elements in \mathbf{R} are binary values, where '1' indicates rain regions and '0' indicates non-rain regions. Note that, while \mathbf{R} can be easily estimated from \mathbf{S} via naive thresholding, modelling \mathbf{R} separately from \mathbf{S} provides two desirable benefits for learning based rain removal methods: (1) it gives additional information for the network to learn about rain streak regions, (2) it allows a new rain removal pipeline to detect rain regions first, and then to operate differently on rain-streak and non-rain-streak regions, preserving background details.

3.2. Rain Accumulation and Heavy Rain

The rain image model introduced in Eq. (2) captures region-dependent rain streaks. In the real world, rain appearance is not only formed by individual rain streaks, but also by accumulation of rain streaks. When rain accumulation is dense, the individual streaks cannot be observed clearly. This rain streak accumulation, whose visual effect is similar to mist or fog, causes the atmospheric veiling effect as well as blur, especially for distance scenes, as



(a) Heavy rain. (b) Rain accumulation.

Figure 2. (a) In heavy rain cases, the rain streaks have various shapes and directions (shown in blue). (b) Rain accumulation reduces the visibility for distant scenes (shown in red).

shown in Fig. 2.a. Aside from rain accumulation, in many occasions, particularly in heavy rain, rain streaks can have various shapes and directions that overlap to each other, as shown in Fig. 2.a and 2.b.

To accommodate these two phenomena (*i.e.*, rain streak accumulation and overlapping rain streaks with different directions), we create a new model. The model comprises of multiple layers of rain streaks, representing the diversity of rain streaks. It also includes the appearance of rain accumulation, by relying on the Koschmieder model that is approximately applicable to many turbid media, including mist, fog (*e.g.* [30]) and underwater (*e.g.* [34, 2]). Our new rain model is expressed as:

$$\mathbf{O} = \alpha \left(\mathbf{B} + \sum_{t=1}^s \tilde{\mathbf{S}}_t \mathbf{R} \right) + (1 - \alpha) \mathbf{A}, \quad (3)$$

where each $\tilde{\mathbf{S}}_t$ is a layer of rain streaks that have the same direction. t is the index of the rain-streak layers, and s is the maximum number of rain-streak layers. \mathbf{A} is the global atmospheric light, α is the atmospheric transmission. Based on Eq. (3), we can generate synthetic images that are better representative of natural images than those generated by Eq. (1). Thus, we can use these images to train our network. Fig. 3 shows some generated images. Note that, the rain accumulation appearance is enforced on the rain-contaminated image $\left(\mathbf{B} + \sum_{t=1}^s \tilde{\mathbf{S}}_t \mathbf{R} \right)$, hence Eq. (3) implies that, we can handle rain accumulation and rain streak removal separately, which provides convenience for our training.

In the following section, we first develop a deep convolutional network to detect and remove rain streaks from the rain images. Then, we consider rain accumulation and heavy rain situations, where we generalize our CNN to a recurrent model.

4. Joint Rain Streak Detection and Removal

We construct a multi-task network to perform **JO**int **R**ain **D**etection and **R**emoval (JORDER) that solves the inverse problem in Eq. (2) through end-to-end learning. Rain regions are first detected by JORDER to further constrain the rain removal. To leverage more context without losing

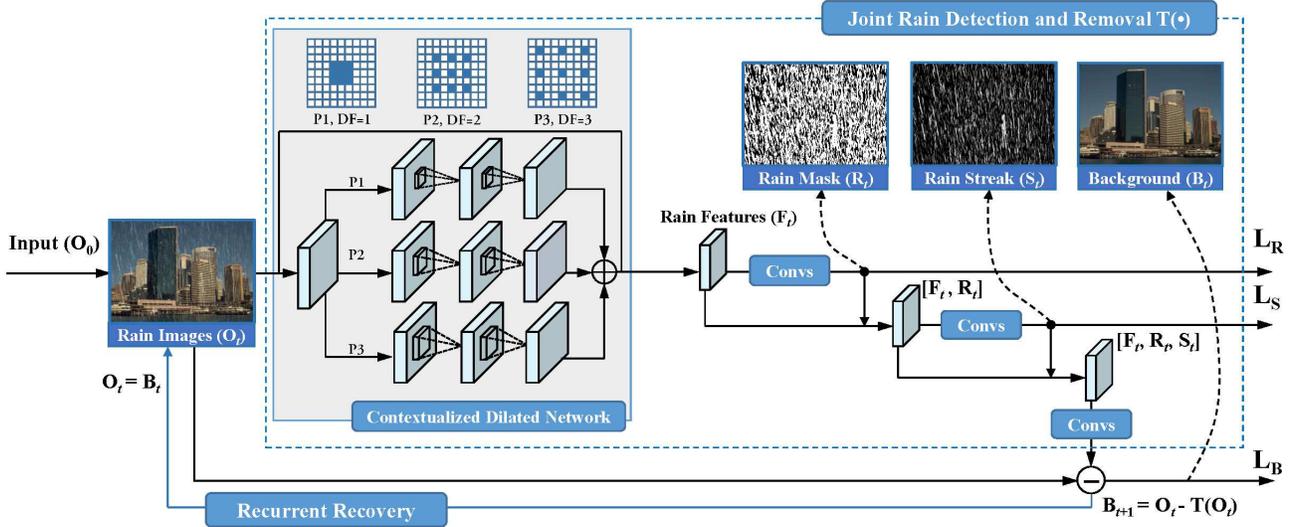


Figure 4. The architecture of our proposed recurrent rain detection and removal. Each recurrence is a multi-task network to perform a joint rain detection and removal (in the blue dash box). In such a network, a contextualized dilated network (in the gray region) extracts rain features F_t from the input rain image O_t . Then, R_t , S_t and B_t are predicted to perform joint rain detection, estimation and removal.

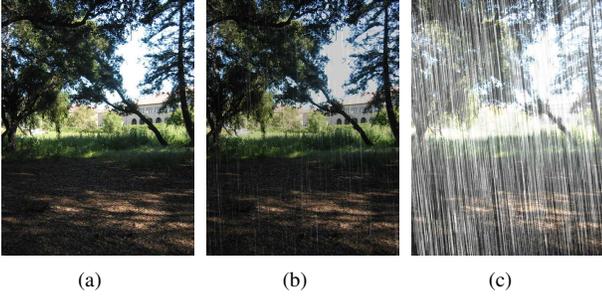


Figure 3. Synthesized rain images (b) and (c) from (a) following the process of Eq. (2) and Eq. (3), respectively.

local details, we propose a novel network structure – the contextualized dilated network – for extracting the rain discriminative features and facilitating the following rain detection and removal.

4.1. Multi-Task Networks for Joint Rain Detection and Removal

Relying on Eq. (2), given the observed rain image O , our goal is to estimate B , S and R . Due to the ill-posedness nature of the problem, we employ a maximum-a-posteriori (MAP) estimation:

$$\arg \min_{B, S, R} \|O - B - SR\|_2^2 + P_b(B) + P_s(S) + P_r(R), \quad (4)$$

where $P_b(B)$, $P_s(S)$ and $P_r(R)$ are the enforced priors on B , S and R , respectively. Previous priors on B and S include hand-crafted features, *e.g.* cartoon texture decomposition [23], and some data-driven models, such as sparse dictionary [28] and Gaussian mixture models [26]. For deep

learning methods, the priors of B , S and R are learned from the training data and are embedded into the network implicitly.

The estimation of B , S and R is intrinsically correlated. Thus, the estimation of B benefits from the predicted \hat{S} and \hat{R} . To convey this, the natural choice is to employ a multi-task architecture, which can be trained using multiple loss functions based on the ground truths of R , S and B (see the blue dash box in Fig. 4).

As shown in the figure, we first exploit a contextualized dilated network to extract the rain feature representation F . Subsequently, R , S and B are predicted in a sequential order, implying a continuous process of rain streak detection, estimation and removal. Each of them is predicted based on F :

1. R is estimated from the convolutional process of F ,
2. S is predicted from the convolutional process of the concatenation of $[F, \hat{R}]$,
3. B is computed from the convolutional process of the concatenation of $[F, \hat{R}, \hat{S}, O - \hat{R}\hat{S}]$.

There are several potential choices for the network structures, such as estimating the three variables in the order of S , R , B , or in parallel (instead of sequential). We compare some alternative architectures and demonstrate empirically the superiority of ours in the supplementary material.

4.2. Contextualized Dilated Networks

For rain removal task, contextual information from an input image is demonstrated to be useful for automatically identifying and removing the rain streaks [19]. Thus, we propose a contextualized dilated network to aggregate

context information at multiple scales for learning the rain features. The network gains contextual information in two ways: 1) through a recurrent structure, which is similar to the recurrent ResNet [41], and provides an increasingly larger receptive field for the subsequent layers; 2) in each recurrence, the output features aggregate the representations of the three convolution paths with different dilated factors and receptive fields.

Specifically, as shown in the gray region of Fig. 4, the network first transforms the input rain image into feature space via the first convolution. Then, the network refines the features progressively. In each recurrence, the results from the three convolution paths with different dilated factors are aggregated with the input features from the last recurrence via the identity forwarding. The dilated convolution [42] weights pixels with a step size of the dilated factor, and thus increases its receptive field without losing resolution. Our three dilated paths consist of two convolutions with the same kernel size 3×3 . However, with different dilated factors, different paths have their own receptive field. As shown in the top part of the gray region in Fig. 4, path P_2 consists of two convolutions with the dilated factor 2. The convolution kernel is shown as the case of DF= 2. Thus, cascading two convolutions, the three paths have their receptive fields of 5×5 , 9×9 and 13×13 .

4.3. Network Training

Let $\mathbf{F}_{rr}(\cdot)$, $\mathbf{F}_{rs}(\cdot)$ and $\mathbf{F}_{bg}(\cdot)$ denote the inverse recovery functions modelled by the learned network to generate the estimated rain streak binary map $\hat{\mathbf{R}}$, rain streak map $\hat{\mathbf{S}}$ and background image $\hat{\mathbf{B}}$ based on the input rain image \mathbf{O} . We use Θ to collectively represent all the parameters of the network.

We use n sets of corresponding rain images, background images, rain region maps and rain streak maps $\{(\mathbf{o}_i, \mathbf{g}_i, \mathbf{r}_i, \mathbf{s}_i)\}_{i=1}^n$ for training. We adopt the following joint loss function to train the network parametrized by Θ such that it is capable to jointly estimate \mathbf{r}_i , \mathbf{s}_i and \mathbf{g}_i based on rain image \mathbf{o}_i :

$$L(\Theta) = \frac{1}{n} \sum_{i=1}^n (\|\mathbf{F}_{rs}(\mathbf{o}_i; \Theta) - \mathbf{s}_i\|^2 + \lambda_1 \|\mathbf{F}_{bg}(\mathbf{o}_i; \Theta) - \mathbf{g}_i\|^2 - \lambda_2 (\log \hat{\mathbf{r}}_{i,1} \mathbf{r}_{i,1} + \log(1 - \hat{\mathbf{r}}_{i,2})(1 - \mathbf{r}_{i,2}))), \quad (5)$$

where

$$\hat{\mathbf{r}}_{i,j} = \frac{\exp \{ \mathbf{F}_{rs,j}(\mathbf{o}_i; \Theta) \}}{\sum_{k=1}^2 \exp \{ \mathbf{F}_{rs,k}(\mathbf{o}_i; \Theta) \}}, j \in \{1, 2\}.$$

Parameters λ_1 and λ_2 are the weighting factors. The network is trained to minimize the above loss, using the back-propagation.

5. Rain Removal in Real Image

In the previous section, we construct a multi-task network to jointly detect and remove rain streaks. In this section, we further enhance our network to handle both multiple rain-streak layers (where each layer has its own streak direction) and rain accumulation.

5.1. Recurrent Rain Detection and Removal

Our recurrent JORDAR network can be understood as a cascade of the convolutional joint rain detection and removal networks to perform progressive rain detection and removal and recover the background layer with increasingly better visibility.

Architecture. We define the process of the network in the blue dash box of Fig. 4, which generates the residual image $\mathbf{T}(\cdot)$ by computing the differences between \mathbf{O} and \mathbf{B} . Then, our recurrent rain detection and removal works as follows,

$$\begin{aligned} [\epsilon_t, \mathbf{R}_t, \mathbf{S}_t] &= \mathbf{T}(\mathbf{O}_t), \\ \mathbf{B}_t &= \mathbf{O}_t - \epsilon_t, \\ \mathbf{O}_{t+1} &= \mathbf{B}_t. \end{aligned} \quad (6)$$

In each iteration t , the predicted residue ϵ_t is accumulated and propagated to the final estimation via updating \mathbf{O}_t and \mathbf{B}_t . Note that, while the estimated rain mask \mathbf{R}_t and streak layer \mathbf{S}_t are not casted into the next recurrence directly, the losses to regularize them in fact provide strong side information to learn $\mathbf{T}(\cdot)$. The final estimation can be expressed as:

$$\mathbf{B}_\tau = \mathbf{O}_0 + \sum_{t=1}^{\tau} \epsilon_t, \quad (7)$$

where τ is the total iteration number. The method removes rain streaks progressively, part by part, based on the intermediate results from the previous step. The complexity of rain removal in each iteration is consequently reduced, enabling better estimation, especially in the case of heavy rain.

Network Training. Our recurrent JORDAR network introduces an extra time variable t to the loss function $L(\Theta)$ in Eq. (5) and gives $L(\Theta_t, t)$, where $L(\Theta_0, 0) = L(\Theta_0)$. When $t > 1$, $L(\Theta_t, t)$ is equivalent to $L(\Theta)$ that replaces \mathbf{o}_i and Θ by $\mathbf{o}_{i,t}$ and Θ_t , respectively, where $\mathbf{o}_{i,t}$ is generated from the t -th iterations of the process Eq. (6) on the initial \mathbf{o}_i . Then, the total loss function L_{Iter} for training \mathbf{T} is

$$L_{\text{Iter}}(\{\Theta_0, \dots, \Theta_\tau\}) = \sum_{t=0}^{\tau} L(\Theta_t, t). \quad (8)$$

5.2. Rain-Accumulation Removal

Distant rain streaks accumulate and form rain atmospheric veil, which is visually similar to fog. It causes visibility degradation, and thus needs to be removed. We call this process rain-accumulation removal. Since the degradation effect and the model (Eq. (3)) are similar to that of fog, our rain-accumulation removal is essentially similar to defogging (e.g., [8]). Like in defogging, the output of our rain removal clears up the veil effect and boosts the contrast.

Eq. (3) suggests that the rain-accumulation removal should be the first step in the whole process of deraining. However, placing it as a preprocessing degrades the quality of rain streaks. Since, all rain streaks, including those that are already sharp and clearly visible, are further boosted, causing the streaks to look different from those in the training images. Hence, in our proposed pipeline, we apply the streak removal first, followed by the rain-accumulation removal, and then the streak removal again. This, as it turns out, is beneficial, since the rain-accumulation removal will make the appearance of less obvious rain streaks (which are likely unnoticed by the first round of the streak removal) become more obvious.

For our rain-accumulation removal, we create another network based on the structure of contextualized dilated network, with only one recurrence, trained with the synthesized data generated with the random background reliance and transmission value (similar to a defogging technique [8]). Overall, we find that our sequential process (streak removal, rain-accumulation removal, and streak removal) is generally effective (see the experimental results and supplementary material for the evaluation).

6. Experimental Results

Datasets. We compare our method with state-of-the-art methods on a few benchmark datasets: (1) *Rain12*² [26], which includes 12 synthesized rain images with only one type of rain streaks; *Rain100L*, which is the synthesized data set with only one type of rain streaks (Fig. 5.c); (2) *Rain20L*, which is a subset of *Rain100L* used for testing the competing network architectures in the supplementary material; (3) *Rain100H*, which is our synthesized data set with five streak directions (Fig. 5.d). Note, while it is rare for a real rain image to contain rain streaks in many different directions, synthesizing this kind of images for training can boost the capacity of the network.

The images for synthesizing *Rain100L*, *Rain20L* and *Rain100H* are selected from BSD200 [29]. The dataset for training our network and another deep learning baseline – SRCNN for deraining – is BSD300, excluding the ones appeared in *Rain12*. The rain streaks are synthesized in two ways: (1) the photorealistic rendering techniques proposed

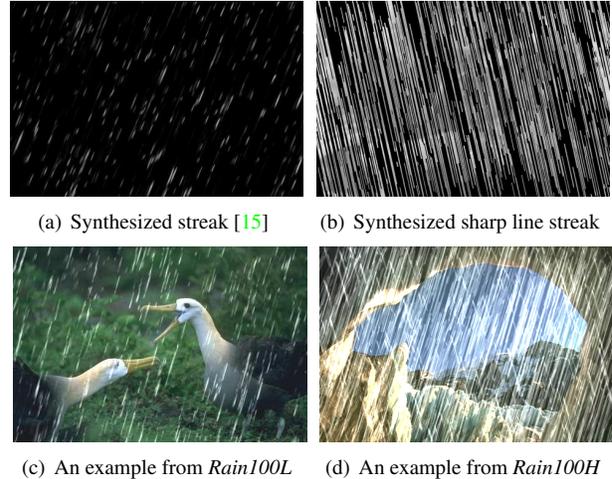


Figure 5. Examples of synthesized rain streaks and rain images.

by [15] as shown in Fig. 5.a; (2) the simulated sharp line streaks along a certain direction with a small variation within an image as shown in Fig. 5.b. We release our training and testing sets, as well as their synthesis codes to public.

Baseline Methods. We compare the four versions of our approaches, JORDER- (one version of our methods that has only one convolution path in each recurrence without using dilated convolutions), JORDER (Section 4), JORDER-R (Section 5.1), JORDER-R-DEVEIL (Section 5.2) with five state-of-the-art methods: image decomposition (ID) [23], CNN-based rain drop removal (CNN) [13], discriminative sparse coding (DSC) [28], layer priors (LP) [26] and a common CNN baseline for image processing – SRCNN [22], trained for deraining. SRCNN is implemented and trained by ourselves, while other methods are kindly provided by the authors.

For the experiments on synthesized data, two metrics – Peak Signal-to-Noise Ratio (PSNR) [20] and Structure Similarity Index (SSIM) [36] are used as comparison criteria. We evaluate the results only in the luminance channel, which has a significant impact on the human visual system to perceive the image quality. Our results and codes are online available.

Quantitative Evaluation. Table 1 shows the results of different methods on *Rain12*. As observed, our method considerably outperforms other methods in terms of both PSNR and SSIM. Table 2 presents the results of JORDER and JORDER-R on *Rain100H*. Note that, our JORDER-R is designed to handle such hard cases, thus achieves considerably better results than the other methods. The PSNR of JORDER-R gains over JORDER more than 1dB. Such a large gain demonstrates that the recurrent rain detection and removal significantly boosts the performance on synthesized heavy rain images.

²<http://yu-li.github.io/>

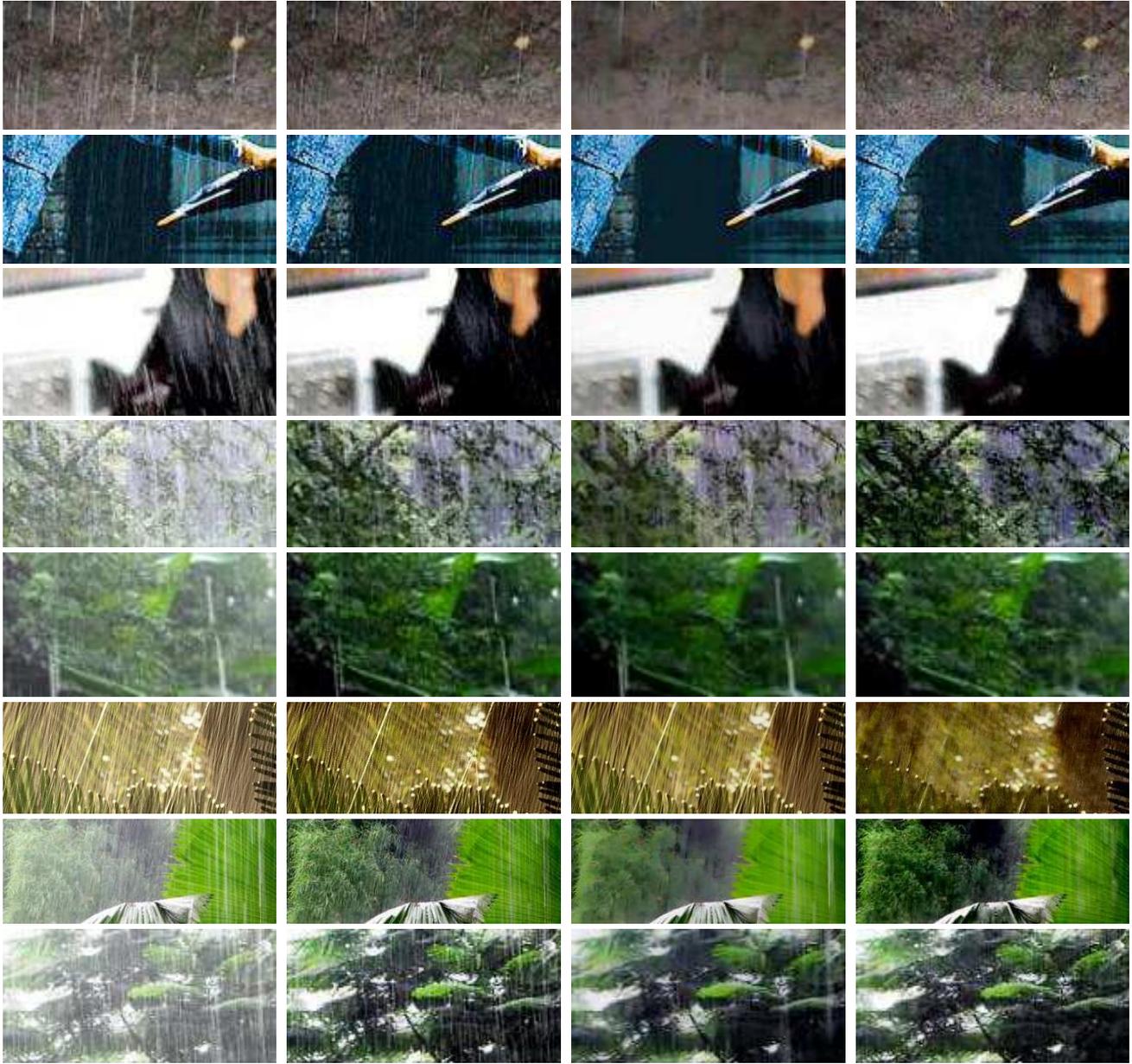


Figure 6. Results of different methods on real images. From top to down: rain image, DSC [28], LP [26] and JORDER-R. Their corresponding full images are presented in the supplementary material.

Qualitative Evaluation. Fig. 6 shows the results of real images. For fair comparisons, we use JORDER-R to process these rain images and do not handle rain accumulation on these results, to be consistent with other methods. As observed, our method significantly outperforms them and is successful in removing the majority of rain streaks.

We also compare all the methods in two extreme cases: dense rain accumulation, and heavy rain as shown in Fig. 7. Our method achieves promising results in removing the majority of rain streaks, enhancing the visibility and preserving details.

Table 3 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. CNN rain drop and some versions of our methods are implemented on GPU, while others are based on CPU. Our GPU versions is computationally efficient. The CPU version of JORDER, a lightest version of our method, takes up the shortest running time among all CPU-based approaches. In general, our methods in GPU are capable of dealing with a 500×500 rain image less than 10s, which is considerably faster than the existing methods.



Figure 7. Examples of JORDER-R-DEVEIL on heavy rain (left two images) and mist images (right two images).

Table 1. PSNR and SSIM results among different rain streak removal methods on *Rain12* and *Rain100L*.

Baseline	<i>Rain12</i>		<i>Rain100L</i>	
Metric	PSNR	SSIM	PSNR	SSIM
ID [23]	27.21	0.75	23.13	0.70
DSC [28]	30.02	0.87	24.16	0.87
LP [26]	32.02	0.91	29.11	0.88
CNN [13]	26.65	0.78	23.70	0.81
SRCNN [22]	34.41	0.94	32.63	0.94
JORDER-	35.86	0.95	35.41	0.96
JORDER	36.02	0.96	36.11	0.97

Table 2. PSNR and SSIM results among different rain streak removal methods on *Rain100H*.

Metric	ID [23]	LP [26]	DSC [28]
PSNR	14.02	14.26	15.66
SSIM	0.5239	0.4225	0.5444
Metric	JORDER-	JORDER	JORDER-R
PSNR	20.79	22.15	23.45
SSIM	0.5978	0.6736	0.7490

Table 3. The time complexity (in seconds) of JORDER compared with state-of-the-art methods. JR and JRD denote JORDER-R and JORDER-R-DEVEIL, respectively. (G) and (C) denote the implementation on GPU and CPU, respectively.

Scale	CNN [13] (G)	ID [23]	DSC [28]	LP [26]
80×80	0.85	449.94	14.32	35.97
500×500	6.39	1529.85	611.91	2708.20
Scale	JORDER (C)	JORDER (G)	JR (G)	JRD (G)
80×80	2.97	0.11	0.32	0.72
500×500	69.79	1.46	3.08	7.16

Evaluation on Streak and Rain-Accumulation Removal.

Fig. 8 shows the significant superiority of our method (f), namely, [streak removal, rain-accumulation removal, streak

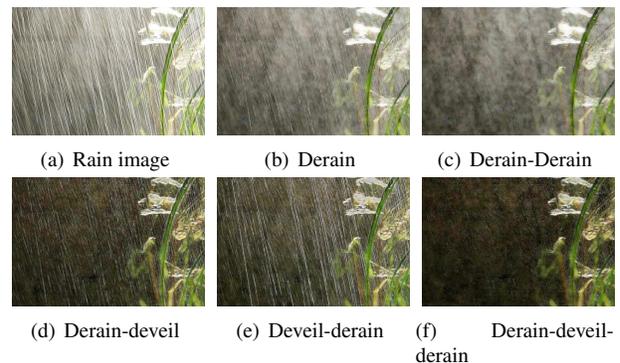


Figure 8. The results of JORDER-R-DEVEIL in different orders.

removal], than other potential combinations ((b)-(e)).

7. Conclusion

In this paper, we have introduced a new deep learning based method to remove rain from a single image, even in the presence of rain streak accumulation and heavy rain. A new region-dependent rain image model is proposed for additional rain detection and is further extended to simulate rain accumulation and heavy rains. Based on this model, we developed a fully convolutional network that jointly detect and remove rain. Rain regions are first detected by the network which naturally provides additional information for rain removal. To restore images captured in the environment with both rain accumulation and heavy rain, we introduced a recurrent rain detection and removal network that progressively removes rain streaks, embedded with the rain-accumulation removal network. Evaluations on real images demonstrated that our method outperforms state-of-the-art methods.

References

- [1] F. Agostinelli, M. R. Anderson, and H. Lee. Adaptive multi-column deep neural networks with application to robust image denoising. In *Proc. Annual Conf. Neural Information Processing Systems*. 2013. 2
- [2] C. Ancuti, C. O. Ancuti, T. Haber, and P. Bekaert. Enhancing underwater images and videos by fusion. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 81–88, June 2012. 3
- [3] P. C. Barnum, S. Narasimhan, and T. Kanade. Analysis of rain and snow in frequency space. *Int'l Journal of Computer Vision*, 86(2-3):256–274, 2010. 1
- [4] J. Bossu, N. Hautière, and J.-P. Tarel. Rain or snow detection in image sequences through use of a histogram of orientation of streaks. *International journal of computer vision*, 93(3):348–367, 2011. 1
- [5] N. Brewer and N. Liu. Using the shape characteristics of rain to identify and remove rain from video. In *Joint IAPR International Workshops on SPR and SSPR*, pages 451–458, 2008. 1
- [6] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising with multi-layer perceptrons, part 1: comparison with existing algorithms and with bounds. *arXiv:1211.1544*. 2
- [7] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising with multi-layer perceptrons, part 2: training trade-offs and analysis of their mechanisms. *arXiv:1211.1552*. 2
- [8] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE Trans. on Image Processing*, PP(99):1–1, 2016. 2, 6
- [9] Y.-L. Chen and C.-T. Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1968–1975, 2013. 1, 2
- [10] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen. Deep network cascade for image super-resolution. In *Proc. IEEE European Conf. Computer Vision*. 2014. 2
- [11] C. Dong, C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 2015. 2
- [12] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. In *EC-CV*. 2014. 2
- [13] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *Proc. IEEE Int'l Conf. Computer Vision*, December 2013. 1, 2, 6, 8
- [14] K. Garg and S. K. Nayar. Detection and removal of rain from videos. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, volume 1, pages I–528, 2004. 1
- [15] K. Garg and S. K. Nayar. Photorealistic rendering of rain streaks. In *ACM Trans. Graphics*, volume 25, pages 996–1002, 2006. 1, 6
- [16] K. Garg and S. K. Nayar. Vision and rain. *Int'l Journal of Computer Vision*, 75(1):3–27, 2007. 1
- [17] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv:1508.06576*, 2015. 2
- [18] 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 Transactions on multimedia*, 16(1):83–93, 2014. 1
- [19] 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'l Conf. Multimedia and Expo*, pages 164–169, 2012. 2, 3, 4
- [20] Q. Huynh-Thu and M. Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008. 6
- [21] V. Jain and S. Seung. Natural image denoising with convolutional networks. In *Proc. Annual Conf. Neural Information Processing Systems*. 2009. 2
- [22] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM Trans. Multimedia*, pages 675–678, 2014. 6, 8
- [23] L. W. Kang, C. W. Lin, and Y. H. Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Trans. on Image Processing*, 21(4):1742–1755, April 2012. 1, 2, 4, 6, 8
- [24] J.-H. Kim, C. Lee, J.-Y. Sim, and C.-S. Kim. Single-image deraining using an adaptive nonlocal means filter. In *IEEE Trans. on Image Processing*, pages 914–917, 2013. 1
- [25] 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'l Conf. Image Processing*, pages 914–917, Sept 2013. 2

- [26] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown. Rain streak removal using layer priors. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 2736–2744, 2016. [2](#), [3](#), [4](#), [6](#), [7](#), [8](#)
- [27] K. G. Lore, A. Akintayo, and S. Sarkar. Llnet: A deep autoencoder approach to natural low-light image enhancement. *arXiv preprint arXiv:1511.03995*, 2015. [2](#)
- [28] Y. Luo, Y. Xu, and H. Ji. Removing rain from a single image via discriminative sparse coding. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 3397–3405, 2015. [1](#), [2](#), [3](#), [4](#), [6](#), [7](#), [8](#)
- [29] 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. IEEE Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001. [6](#)
- [30] S. G. Narasimhan and S. K. Nayar. Vision and the atmosphere. *Int'l Journal of Computer Vision*, 48(3):233–254, 2002. [3](#)
- [31] C. Osendorfer, H. Soyer, and P. van der Smagt. Image super-resolution with fast approximate convolutional sparse coding. In *Neural Information Processing*. 2014. [2](#)
- [32] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf. Learning to deblur. *arXiv:1406.7444*, 2014. [2](#)
- [33] S.-H. Sun, S.-P. Fan, and Y.-C. F. Wang. Exploiting image structural similarity for single image rain removal. In *Proc. IEEE Int'l Conf. Image Processing*, pages 4482–4486, 2014. [1](#)
- [34] Y. Tian and S. G. Narasimhan. Seeing through water: Image restoration using model-based tracking. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 2303–2310, Sept 2009. [3](#)
- [35] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 2010. [2](#)
- [36] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. on Image Processing*, 13(4):600–612, 2004. [6](#)
- [37] H. Z. J. P. X. C. Wenqi Ren, Si Liu and M.-H. Yang. Single image dehazing via multi-scale convolutional neural networks. In *Proc. IEEE European Conf. Computer Vision*, pages 914–917, October 2016. [2](#)
- [38] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *Proc. Annual Conf. Neural Information Processing Systems*. 2012. [2](#)
- [39] L. Xu, J. S. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *Proc. Annual Conf. Neural Information Processing Systems*. 2014. [2](#)
- [40] Z. Yan, H. Zhang, B. Wang, S. Paris, and Y. Yu. Automatic photo adjustment using deep neural networks. *ACM Trans. Graphics*, 2015. [2](#)
- [41] W. Yang, J. Feng, J. Yang, F. Zhao, J. Liu, Z. Guo, and S. Yan. Deep Edge Guided Recurrent Residual Learning for Image Super-Resolution. *ArXiv*, April 2016. [5](#)
- [42] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representation*, 2016. [5](#)
- [43] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *ArXiv e-prints*, August 2016. [2](#)
- [44] 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'l Conf. Multimedia and Expo*, pages 461–464, 2006. [1](#)