Spatial Attentive Single-Image Deraining with a High Quality Real Rain Dataset

Tianyu Wang[1,2*] Xin Yang[1,2*] Ke Xu[1,2] Shaozhe Chen[1] Qiang Zhang[1] Rynson W.H. Lau[2†]

[1]Dalian University of Technology [2]City University of Hong Kong

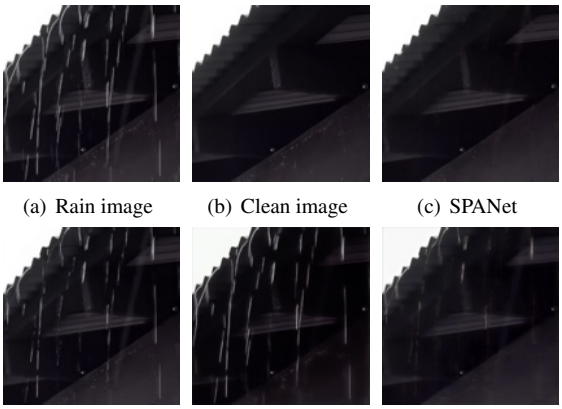
Abstract

Removing rain streaks from a single image has been drawing considerable attention as rain streaks can severely degrade the image quality and affect the performance of existing outdoor vision tasks. While recent CNN-based derainers have reported promising performances, deraining remains an open problem for two reasons. First, existing synthesized rain datasets have only limited realism, in terms of modeling real rain characteristics such as rain shape, direction and intensity. Second, there are no public benchmarks for quantitative comparisons on real rain images, which makes the current evaluation less objective. The core challenge is that real world rain/clean image pairs cannot be captured at the same time. In this paper, we address the single image rain removal problem in two ways. First, we propose a semi-automatic method that incorporates temporal priors and human supervision to generate a high-quality clean image from each input sequence of real rain images. Using this method, we construct a large-scale dataset of $\sim 29.5K$ rain/rain-free image pairs that covers a wide range of natural rain scenes. Second, to better cover the stochastic distribution of real rain streaks, we propose a novel SPatial Attentive Network (SPANet) to remove rain streaks in a local-to-global manner. Extensive experiments demonstrate that our network performs favorably against the state-of-the-art deraining methods.

1. Introduction

Images taken under various rain conditions often show low visibility, which can significantly affect the performance of some outdoor vision tasks, e.g., pedestrian detection [30], visual tracking [37], or road sign recognition [48]. Hence, removing rain streaks from input rain images is an important research problem. In this paper, we focus on the single-image rain removal problem.

In the last decade, we have witnessed a continuous progress on rain removal research with many methods proposed [20, 29, 26, 5, 47, 9], through carefully modeling the physical characteristics of rain streaks. Benefited from large-scale training data, recent deep-learning-based derainers [10, 11, 40, 42, 25, 45, 15] achieve further promising performances. Nonetheless, the single-image rain removal problem remains open in two ways, as discussed below.



(a) Rain image (b) Clean image (c) SPANet

(d) DDN [11] (e) DID-MDN [42] (f) RESCAN [25]

Figure 1. We address the single-image rain removal problem in two ways. First, we generate a high-quality rain/clean image pair ((a) and (b)) from each sequence of real rain images, to form a dataset. Second, we propose a novel SPANet to take full advantage of the proposed dataset. (c) to (f) compare the visual results from SPANet and from state-of-the-art derainers.

Lack of real training data. As real rain/clean image pairs are unavailable, existing derainers typically rely on synthesized datasets to train their models. They usually start with a clean image and add synthetic rain on it to form a rain/clean image pair. Although some works have been done to study the physical characteristics of rain, *e.g.*, rain direction [40] and rain density [42], their datasets still lack the ability to model a large range of real world rain streaks. For example, it is often very difficult to classify the rain density into one of the three levels (*i.e.*, light, medium and heavy) as in [42], and any misclassification would certainly affect the deraining performance. To simulate global rain effects, some methods adopt the nonlinear "screen blend mode" from Adobe Photoshop, or additionally superimpose haze on the synthesized rain images. However, these global settings can only be used in certain types of rain, or the background may be darkened, with the details lost.

* Joint first authors. † Rynson Lau is the corresponding author, and he led this project.

Lack of a real benchmark. Currently, researchers mainly rely on qualitatively evaluating the deraining performance on real rain images through visual comparisons. Fan *et al*. [45] also use an object detection task to help evaluate the deraining performance. Nevertheless, a high-quality real deraining benchmark is still much needed for quantitative evaluation of deraining methods.

In this paper, we address the single-image rain removal problem in two ways, as summarized in Figure 1. First, we address the lack of real training/evaluation datasets based on two observations: (1) as random rain drops fall in high velocities, they unlikely cover the same pixel all the time [13, 44], and (2) the intensity of a pixel covered by rain fluctuates above the true background radiance across a sequence of images. These two observations imply that we can generate one clean image from a sequence of rain images, where individual pixels of the clean image may be coming from different images of the sequence. Hence, we propose a semi-automatic method that incorporates rain temporal properties as well as human supervision to construct a large-scale real rain dataset. We show that it can significantly improve the performance of state-of-the-art derainers on real world rain images.

Second, we observe that real rain streaks can exhibit highly diverse appearance properties (*e.g*., rain shape and direction) within a single image, which challenges existing derainers as they lack the ability to identify real rain streaks accurately. To address this limitation, we exploit a spatial attentive network (SPANet), which first leverages horizontal/vertical neighborhood information to model the physical properties of rain streaks, and then remove them by further considering the non-local contextual information. In this way, the discriminative features for rain streak removal can be learned in a two-stage local-to-global manner. Extensive evaluations show that the proposed network performs favorably against the state-of-the-art derainers.

To summarize, this work has the following contributions:

1. We present a semi-automatic method that incorporates temporal properties of rain streaks and human supervision to generate a high quality clean image from a sequence of real rain images.

2. We construct a large-scale dataset of $\sim 29.5K$ high-resolution rain/clean image pairs, which covers a wide range of natural rain scenes. We show that it can significantly improve the performance of state-of-the-art derainers on real rain images.

3. We design a novel SPANet to effectively learn discriminative deraining features in a local-to-global attentive manner. SPANet achieves superior performance over state-of-the-art derainers.

2. Related works

Single-image rain removal. This problem is extremely challenging due to the ill-posed deraining formulation as:

$$B = O - R, \tag{1}$$

where O, R and B are the input rain image, the rain streak image, and the output derained image, respectively.

Kang *et al*. [20] propose to first decompose the rain image into high-/low-frequency layers and remove rain streaks in the high frequency layer via dictionary learning. Kim *et al*. [21] propose to use non-local mean filters to filter out rain streaks. Luo *et al*. [29] propose a sparse coding based method to separate rain streaks from the background. Li *et al*. [26] propose to use Gaussian mixture models to model rain streaks and background separately for rain removal. Chang *et al*. [5] propose to first affine transform the rain image into a space where rain streaks have vertical appearances and then utilize the low-rank property to remove rain streaks. Zhu *et al*. [47] exploit rain streak directions to first determine the rain-dominant regions, which are used to guide the process of separating rain streaks from background details based on rain-dominant patch statistics.

In [11, 10], deep learning is applied to single image deraining and achieves a significant performance boost. They model rain streaks as "residuals" between the input/output of the networks in an end-to-end manner. Yang *et al*. [40] propose to decompose the rain layer into a series of sub-layers representing rain streaks of different directions and shapes, and jointly detect and remove rain streaks using a recurrent network. In [43], Zhang *et al*. propose to remove rain streaks and recover the background via the Conditional GAN. Recently, Zhang and Patel [42] propose to classify rain density to guide the rain removal step. Li *et al*. [25] propose a recurrent network with a squeeze-and-excitation block [17] to remove rain streaks in multiple stages. However, the performances of CNN-based derainers on real rain images are largely limited by being trained only on synthetic datasets. These derainers also lack the ability to attend to rain spatial distributions. In this paper, we propose to leverage real training data as well as a spatial attentive mechanism to address the single image deraining problem.

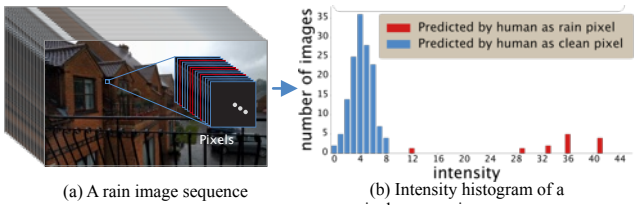
Multi-image rain removal. Unlike single-image deraining, rich temporal information can be derived from a sequence of images to provide additional constraints for rain removal. Pioneering works [12, 13] propose to apply photometric properties to detect rain streaks and estimate the corresponding background intensities by averaging the irradiance of temporal or spatial neighboring pixels. Subsequently, more intrinsic properties of rain streaks, such as chromatic property, are explored by [44, 28, 36]. Recent works [4, 8, 6, 21, 19, 35, 39, 24, 27] focus on removing the rain streaks from the background with moving objects.

Chen *et al*. [7] further propose a spatial-temporal content alignment algorithm to handle fast camera motion and dynamic scene contents, and a CNN to reconstruct high frequency background details.

However, these methods cannot be applied for our purpose of generating high-quality rain-free images. This is because if their assumptions (*e.g*., low-rank [8, 39, 24]) are violated, over-/under-deraining can happen to the entire sequence and further bury the true background radiance, *i.e*., the clean background pixels may not exist in this sequence. Hence, in this paper, we propose to use the original sequence of rain images to generate a clean image, and rely on human judgements on the qualities of generated rain-free images.

Generating the ground truth from real noisy images. One typical strategy [2, 33] to obtain a noise/noise-free image pair is to photograph the scene with a high ISO value and a short exposure time for the noise image, and a low ISO value and a long exposure time for the noise-free image. However, this strategy cannot be used here to capture rain-free images. As rain drops fall at a high speed, increasing the exposure time will enlarge the rain streaks, not removing them. Another approach to obtain a ground truth noise-free image is multi-frame fusion [46, 32, 1], which performs weighted averaging of a pre-aligned sequence of images taken from a static scene with a fixed camera setting. However, as rain streaks have brighter appearances and larger shapes than random noise, this approach is not able to accurately remove rain from the rain pixels. In contrast, we propose to refine the rain pixels based on the observation that the intensity values of the pixels covered by rain fluctuate above their true background intensities.

3. Real Rain Image Dataset



(a) A rain image sequence

(b) Intensity histogram of a pixel cross an image seqeuence

Figure 2. We trace the intensity of one pixel across an image sequence in (a). We ask a user to identify if this pixel in each frame is covered by rain (in red) or not (in blue). The intensity distribution of this pixel over all frames is show in (b). It shows that the intensity of the pixel tends to fluctuate in a smaller range if it is not covered by rain, as compared with that covered by rain.

We first conduct an experiment on how to select a suitable background value o_b from a collection of pixel values $O_l = \{o_{1l}, ..., o_{Nl}\}$ at spatial position l from a sequence of N rain images. We capture a video of a rain scene over a static background, as shown in Figure 2, and then ask a person to indicate (or predict) when a particular pixel is covered by rain and when it is not, across the N frames. We have observed two phenomena. First, rain streaks do not always cover the same pixel (the temporal property of video deraining [44]). Second, humans typically predict if a pixel is covered by rain or not based on the pixel intensity. If the intensity of the pixel is lower at a certain frame compared with the other frames, humans would predict that it is not covered by rain. This is because rain streaks tend to brighten the background. These two observations imply that, given a sequence of N consecutive rain images, we can approximate the true background radiance B_l at pixel l based on these human predicted rain-free pixel values (i.e., the blue region of the histogram in Figure 2(b)). If we assume that the ambient light is constant during this time span, we can then use the value that appears most frequently (i.e., mode in statistics) to approximate the background radiance.

Background approximation. Referring to Figure 3, given a set of pixel values O_l at position l from a sequence of N rain images, we first compute the mode of O_l as:

$$\phi_l = \Phi(O_l), \tag{2}$$

where Φ is the mode operation. However, since Eq. 2 does not consider the neighborhood information when computing ϕ_l, the resulting images tend to be noisy in dense rain streaks. So, we identify the percentile range (R_l^{min}, R_l^{max}) of the computed ϕ_l in O_l based on their intensity values as:

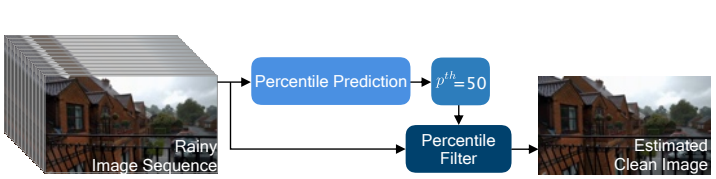
$$R_l^{min} = \frac{100\%}{N}\sum_{i=1}^{N}\{1|o_{il} < \phi_l\},$$
$$R_l^{max} = \frac{100\%}{N}\sum_{i=1}^{N}\{1|o_{il} > \phi_l\}. \tag{3}$$

Figure 3(c) shows an example. Instead of using polygonal lines to connect the mode values ϕ_l at all spatial positions, we can determine a suitable percentile $\hat{p}$ so that it crosses the highest number of percentile ranges (the red dash line in Figure 3(c)). In this way, the estimated background image is globally smoothed by computing $\hat{p}$ as:

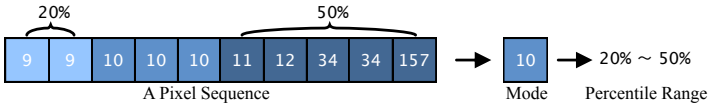
$$\hat{p} = \arg\max_p(\{\sum_{l=0}^{M-1}\{1|R_l^{min} < p < R_l^{max}\}\}_{p=0}^{100}), \tag{4}$$

where M is the number of pixels in a frame. Figure 4(e) shows an example that using the mode leads to noisy result, while our method in Figure 4(f) produces a cleaner image.

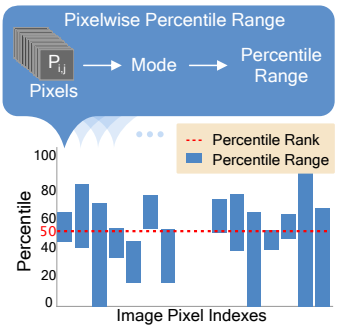
Selection of N for different rain scenes. Recall that we aim to generate one clean image from a sequence of N rain images. Our method assumes that for each pixel of the output clean image, we are able to find some input frames where the pixel is not covered by rain. To satisfy this assumption, we need to adjust N according



(a) Pipeline of Background Approximations



(b) Computing Percentile Range for Mode Value



(c) Percentile Prediction

Figure 3. Overview of our clean image generation pipeline (a). Given a sequence of rain images, we compute the mode for each pixel based on its intensity changes over time, and the percentile range of its mode (b). We then consider the global spatial smoothness by finding a percentile rank that can cross most of the percentile ranges (c).

(a) Input (b) Jiang [19] (c) Wei [39] (d) Li [24] (e) Mode filter (f) **Ours** (g) Ground Truth

Figure 4. A deraining example using a synthetic rain video of 100 frames. We show the best result of each method here. Refer to the supplementary for more results.

to the amount of rain as follows. First, we empirically set N to be $\{20, 100, 200\}$ depending on whether the rain is $\{sparse, normal, dense\}$, respectively, and generate an output image using our method. Second, we ask users to evaluate the image as humans are sensitive to rain streaks as well as other artifacts such as noise. If the image fails in the user evaluation, we adjust N by adding $\{10, 20, 50\}$ frames for $\{sparse, normal, dense\}$ rain streaks and then ask the users to evaluate the new output image again. We find that while 20 and 100 frames are usually large enough to obtain a clean image for $sparse$ and $normal$ rain streaks, N may go from 200 to 300 frames for $dense$ rain streaks. We deliberately start with smaller numbers of frames because we find that the more frames that we use, the higher chance that the video may contain noise, blur and shaking.

Discussion. An intuitive alternative to obtaining a rain-free image is to use a state-of-the-art video deraining method to first generate a sequence of derained results from the input rain sequence, and then average them or select the best result from them to produce a single final rain-free image. Unfortunately, there is no guarantee that rain streaks can be completely removed by the video deraining method, as shown in Figure 4(b)-(d). On the contrary, we rely on human judgements to generate high-quality rain-free images. We show a comparison between our method and three state-of-the-art video deraining methods [19, 39, 24] in Table 1 on 10 synthesized rain videos (10 black-background rain videos bought from [31] are imposed on 10 different background images), which clearly demonstrates the effectiveness of our method.

Dataset description. We construct a large-scale dataset using 170 real rain videos, of which 84 scenes are captured by us using an iPhone X or iPhone 6SP and 86 scenes are collected from StoryBlocks or YouTube. These videos cover common urban scenes (*e.g.*, buildings, avenues), suburb scenes (*e.g.*, streets, parks), and some outdoor fields (*e.g.*, forests). When capturing rain scenes, we also control the exposure durations as well as the ISO parameter to cover different lengths of rain streaks and illumination conditions. Using the aforementioned method, we generate $29,500$ high-quality rain/clean image pairs, which are split into $28,500$ for training and $1,000$ for testing. Our experiments show that this dataset helps improve the performance of state-of-the-art derainers.

Methods	Input	Jiang *et al.* [19]	Wei *et al.* [39]	Li *et al.* [24]	Ours
PSNR	25.40	32.79 (29.82)	27.30 (25.71)	32.59 (30.59)	**51.40**
SSIM	0.7228	0.8827 (0.8566)	0.9043 (0.8911)	0.9458 (0.9387)	**0.9907**

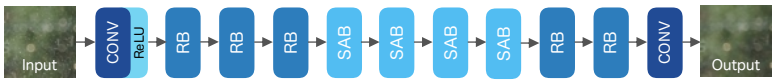
Table 1. Comparison with the state-of-the-art video deraining methods. In each method, we select the frame of highest PSNR for comparison. The average PSNR/SSIM are in brackets.

4. Proposed Model

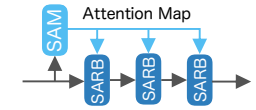
As real rain streaks may have highly diverse appearances across the image, we propose the SPANet to detect and remove rain streaks in a local-to-global manner, as shown in Figure 5(a). It is a fully convolutional network that takes one rain image as input and outputs a derained image.

4.1. Spatial Attentive Block

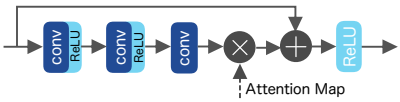
Review on IRNN architecture. Recurrent neural networks with ReLU and identity matrix initialization (IRNN) for natural language processing [23] have been shown to be easy to train, good at modeling long-range dependen-



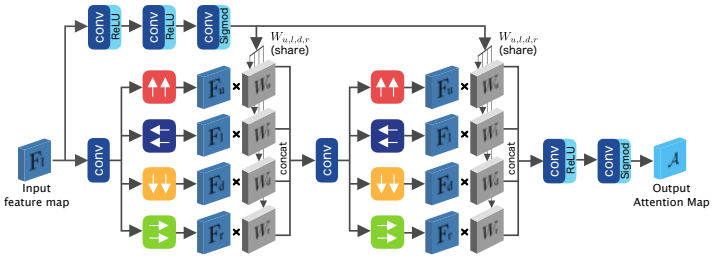
(a) Spatial Attentive Network **(SPANet)**



(b) Spatial Attentive Block **(SAB)**



(c) Spatial Attentive Residual Block **(SARB)**



(d) Spatial Attentive Module **(SAM)**

Figure 5. The architecture of the proposed **SPANet** (a). It adopts three standard residual blocks (**RB**s) [16] to extract features, four spatial attentive blocks (**SAB**s) to identify rain streaks progressively in four stages, and two residual blocks to reconstruct a clean background. A **SAB** (b) contains three spatial attentive residual blocks (**SARB**s) (c) and one spatial attentive module (**SAM**) (d). Dilation convolutions [41] are used in **RB**s and **SARB**s.

cies as well as efficient. When applied to computer vision problems, their key advantage is that information can be efficiently propagated across the entire image to accumulate long range varying contextual information, by stacking at least two RNN layers. In [3], a two-round four-directional IRNN architecture is used to exploit contextual information to improve small object detection. While the first round IRNN aims to produce the feature maps that summarize the neighboring contexts for each position of the input image, the second round IRNN further gathers non-local contextual information for producing global aware feature maps. Recently, Hu *et al*. [18] also exploit this two-round four-directional IRNN architecture to detect shadow regions based on the observation that directions play an important role in finding strong cues between shadow/non-shadow regions. They design a direction-aware attention mechanism to generate more discriminative contextual features.

We summarize the four-directional IRNN operation for computing feature $h_{i,j}$ at location (i, j) as:

$$h_{i,j} \leftarrow \max\left(\alpha_{dir}\, h_{i,j-1} + h_{i,j}, 0\right), \qquad (5)$$

where α_{dir} denotes the weight parameter in the recurrent convolution layer for each direction. Figure 6 illustrates how a two-round four-directional IRNN architecture accumulates global contextual information. Here, we extend the two-round four-directional IRNN model to the single-image rain removal problem, for the purpose of handling the significant appearance variations of real rain streaks.

Spatial attentive module (SAM). We build SAM based on the aforementioned two-round four-directional IRNN architecture. We use the IRNN model to project the rain streaks to the four main directions. Another branch is added to capture the spatial contextual information in order to selectively highlight the projected rain features, as shown in Figure 5(d). Unlike [18] that implicitly learns

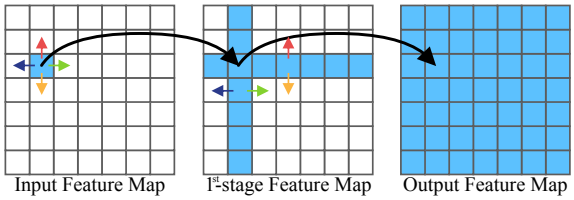
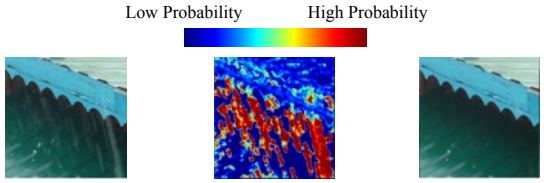


Figure 6. Illustration of how the two-round four-directional IRNN architecture accumulates global contextual information in two stages. In the first stage, for each position at the input feature map, four-directional (up, left, down, right) recurrent convolutional operations are performed to collect horizontal and vertical neighborhood information. In the second stage, by repeating the previous operations, the contextual information from the entire input feature map are obtained.

direction-aware features in the embedding space, we further use additional convolutions and sigmoid activations to explicitly generate the attention map through explicit supervision. The attention map indicates rain spatial distributions and is used to guide the following deraining process. Figure 7 shows the input rain images in (a) and our SPANet derained results in (c). We also visualize the attention maps produced by SAM in (b). We can see that SAM can effectively identify the regions affected by rain streaks, even though the rain streaks exhibit significant appearance variations (*i.e*., smooth and blurry in the first scene and sharp in the second scene).

Removal-via-detection. As shown in Figure 5(a), given an input rain image, three standard residual blocks (RBs) [16] are first used to extract features. We feed these features into a spatial attentive block (SAB) (Figure 5(b)), which uses a SAM to generate an attention map to guide three subsequent spatial attentive residual blocks (SARBs) (Figure 5(c)) to remove rain streaks via the learned negative residuals. The SAB is repeated four times. (Note that the



(a) Rain Image (b) Attention Map (c) SPANet Result

Figure 7. Visualization of the attention map. (a) shows one real rain image. (b) shows the corresponding attention map produced by SAM. Red color indicates pixels that are highly likely covered by rain. (c) shows the corresponding derained result by the proposed SPANet. This demonstrates the effectiveness of SAM in handling significant appearance variations of rain streaks. Refer to the supplementary for more results.

weights of the SAM in the four SABs are shared.) Finally, the resulting feature maps are fed to two standard residual blocks to reconstruct the final clean background image.

4.2. Training Details

Loss function. We adopt the following loss function to train SPANet:

$$\mathcal{L}_{total} = \mathcal{L}_1 + \mathcal{L}_{SSIM} + \mathcal{L}_{Att}. \quad (6)$$

We use the standard $\mathcal{L}_1$ loss to measure the per-pixel reconstruction accuracy. $\mathcal{L}_{ssim}$ [38] is used to constrain the structural similarities, and is defined as: $1 - SSIM(\mathcal{P}, \mathcal{C})$, where $\mathcal{P}$ is the predicted result and $\mathcal{C}$ is the clean image. We further apply the attention loss $\mathcal{L}_{att}$ as:

$$\mathcal{L}_{att} = \|\mathcal{A} - \mathcal{M}\|_2^2, \quad (7)$$

where $\mathcal{A}$ is the attention map from the first SAM in the network and $\mathcal{M}$ is the binary map of the rain streaks, which is computed by thresholding the difference between the rain image and clean image. In this binary map, a 1 indicates that the pixel is covered by rain and 0 otherwise.

Implementation details. SPANet is implemented using the PyTorch [34] framework on a PC with a E5-2640 v4 2.4GHz CPU and 8 NVIDIA Titan V GPUs. For loss optimization, we adopt the Adam optimizer [22] with a batch size of 32. We adopt scaling and cropping to augment the diversity of rain streaks. The learning rate is initialized at 0.005 and divided by 10 after 5K, 15K, 30K and 50K iterations. We train the network for 60K iterations.

5. Experiments

In this section, We first evaluate the effectiveness of the proposed dataset on existing CNN-based single-image derainers, and then compare the proposed SPANet to the state-of-the-art single-image deraining methods. Finally, we provide internal analysis to study the contributions of individual components of SPANet. Refer to the supplementary for more results.

Evaluation on the proposed dataset. The performances of existing CNN-based derainers [11, 40, 42, 25] trained on our dataset are shown in Table 2. It demonstrates that our real dataset can significantly improve the performance of CNN-based methods on real images. This is mainly due to the fact that existing synthesized datasets lack the ability to represent highly varying rain streaks. One visual example is given in Figure 9, from which we can see that the retrained derainers can produce cleaner images with more details compared to those trained on synthetic datasets. Note that we use their original codes for evaluation and retraining.

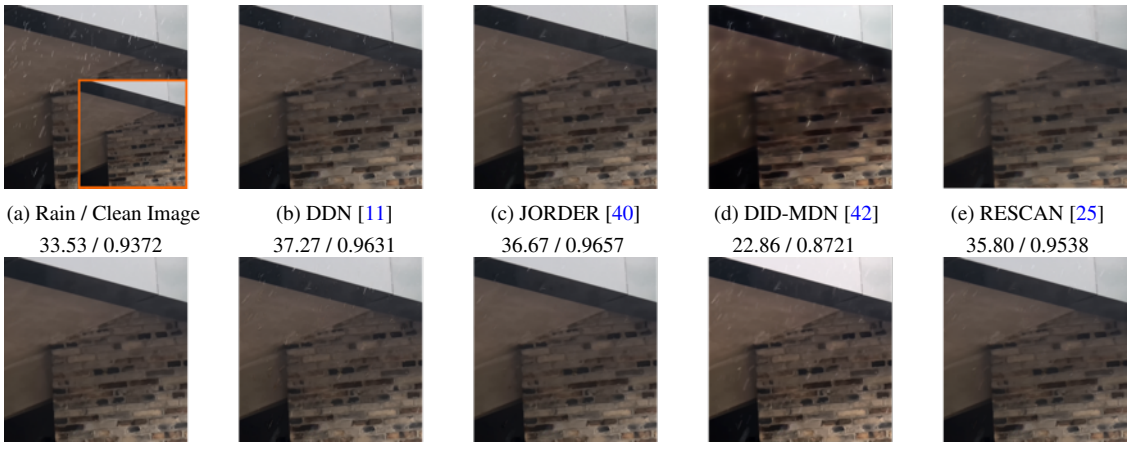
We also show the performance of non-CNN-based state-of-the-art methods in Table 2. We have an interesting observation here that the input rain images have similar or even higher average PSNR and SSIM scores compared with those of the derained results by the state-of-the-art derainers. As demonstrated in Figure 8, it is mainly caused by over deraining. Even though [29] is less dependent on training data (but still depends on a learned dictionary) as the deep learning methods ([40, 42, 25]), it fails when the rain exhibits unseen appearances and mistakenly removes the structures that are similar to rain streaks.

Figure 8. The difference maps (red boxes shown at the top-right) between the input rain image and results by deraining methods that suffer a PSNR drop. (Brighter indicates a higher difference.) We can see that [29, 40, 42, 25] tend to over-derain the image.

Evaluation on the proposed SPANet. Table 2 reports the performance of our SPANet, trained on the proposed dataset. It achieves a superior deraining performance com-

Methods	Rain Images	DSC [29] (ICCV'15)	LP [26] (CVPR'16)	SILS [14] (ICCV'17)	Clear [10] (TIP'17)	DDN [11] (CVPR'17)	JORDER [40] (CVPR'17)	DID-MDN [42] (CVPR'18)	RESCAN [25] (ECCV18)	**Our SPANet**
PSNR	32.64	32.33	32.99	33.40	31.31	33.28 (34.88)	32.16 (35.72)	24.91 (28.96)	30.36 (35.19)	**38.06**
SSIM	0.9315	0.9335	0.9475	0.9528	0.9304	0.9414 (0.9727)	0.9327 (0.9776)	0.8895(0.9457)	0.9553 (0.9784)	**0.9867**

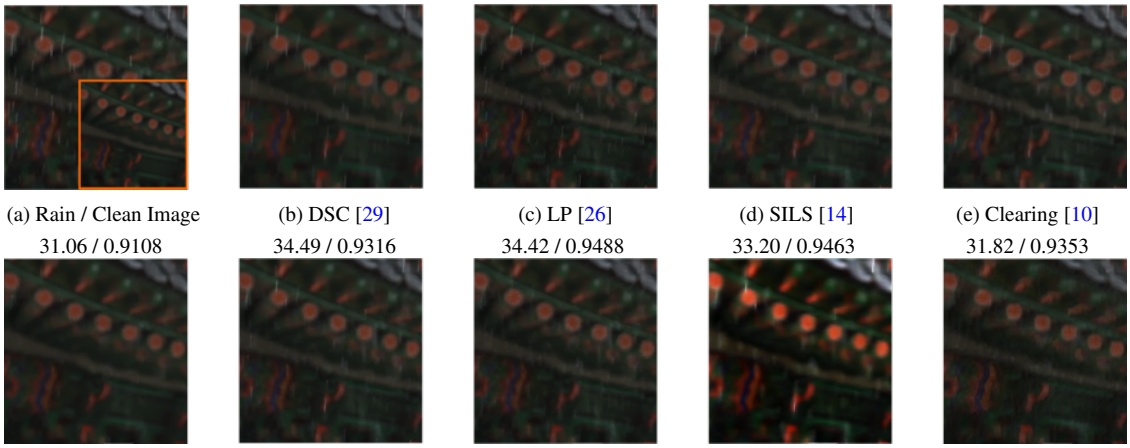
Table 2. Quantitative results for benchmarking the proposed SPANet and the state-of-the-art derainers on the proposed test set. The original codes of all these derainers are used for evaluation. We have also trained CNN-based state-of-the-art methods [11, 40, 42, 25] on our dataset, and results are marked in red. The best performance is marked in **bold**. Note that due to the lack of density labels for the rain images in our dataset, we only fine-tune the pre-trained model of DID-MDN [42] without the re-training label classification network.



(a) Rain / Clean Image 33.53 / 0.9372 | (b) DDN [11] 37.27 / 0.9631 | (c) JORDER [40] 36.67 / 0.9657 | (d) DID-MDN [42] 22.86 / 0.8721 | (e) RESCAN [25] 35.80 / 0.9538

(f) Our SPANet 43.49 / 0.9938 | (g) DDN [11] 38.36 / 0.9668 | (h) JORDER [40] 40.49 / 0.9834 | (i) DID-MDN [42] 26.54 / 0.9625 | (j) RESCAN [25] 39.29 / 0.9771

Figure 9. Visual comparison of the state-of-the-art CNN-based derainers trained on the original/proposed datasets. Methods in red mean that they are retrained on the proposed dataset. PSNR/SSIM results are included for reference.



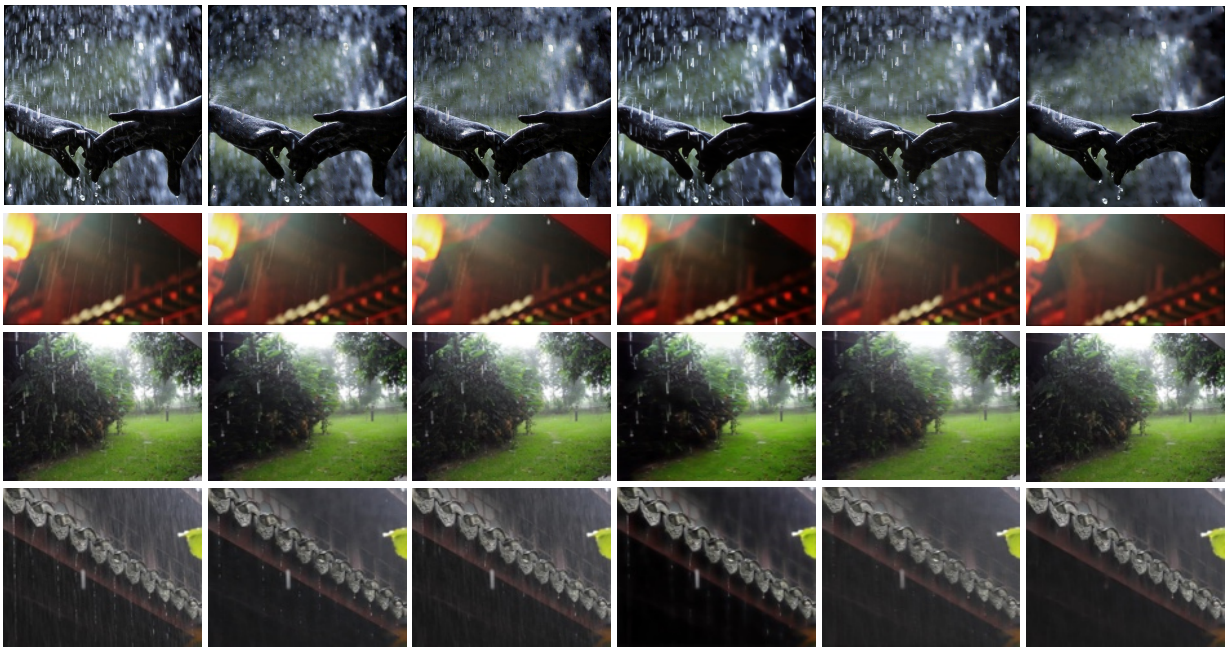
(a) Rain / Clean Image 31.06 / 0.9108 | (b) DSC [29] 34.49 / 0.9316 | (c) LP [26] 34.42 / 0.9488 | (d) SILS [14] 33.20 / 0.9463 | (e) Clearing [10] 31.82 / 0.9353

(f) Our SPANet 38.22 / 0.9764 | (g) DDN [11] 33.94 / 0.9460 | (h) JORDER [40] 35.09 / 0.9495 | (i) DID-MDN [42] 21.69 / 0.8018 | (j) RESCAN [25] 34.35 / 0.9265

Figure 10. Visual comparison of SPANet with the state-of-the-art derainers. PSNR/SSIM results are included for reference.

pared to the state-of-the-art derainers. This is because SPANet can identify the rain streak regions and remove them accurately. Figure 10 shows a visual example from our test set. We can see that while methods (b)−(e) tend to leave rain streaks unremoved and methods (g)−(j) tend to corrupt the background, the proposed SPANet (f) can produce much cleaner result. We also show some deraining examples on rain images collected from previous derain papers and the Internet in Figure 11. While existing derainers fail to remove the rain streaks and some of them tend to darken or blur the background, our SPANet can handle different kinds of rain streaks and preserve more details. Table 3 compares the performances of SPANet with the state-of-the-art derainers on the synthetic test set from [42], demonstrating the effectiveness of SPANet.

Internal analysis. We verify the importance of the spatial attentive module (SAM) and different ways of using it in Table 4. B_a is a basic Resnet-like network that does not use SAM. B_b, B_c, and B_f represent three variants of using only one SAM for four times (recall that we have four SAB blocks), four SAMs, and four SAMs that share the same weights for all operations, respectively. While we can see that all variants of incorporating the SAM improve the performance, B_f performs the best, as sharing the weights makes the deraining process inter-dependent on the four SAB blocks, which allows more attention to be put to the challenging real rain streak distributions. B_d is the SPANet but without the above attention branch in SAM. The comparison between B_d and B_f shows that attention branch is effective in leveraging the local contextual information aggregated from different directions. B_e is a variant that removes the attention loss supervision. It demonstrates the importance of providing explicit supervision on the attention map generation process.



(a) Rain (b) DDN [11] (c) JORDER [40] (d) DID-MDN [42] (e) RESCAN [25] (f) **Our SPANet**

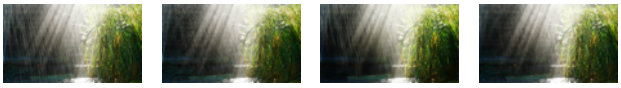
Figure 11. Visual comparison of SPANet with the state-of-the-art CNN-based derainers on some real rain images collected from previous derain papers and from the Internet.

Methods	Input	DSC [29]	LP [26]	Clear[10]	JORDER [40]	DDN [11]	JBO[47]	DID-MDN[42]	**Our SPANet**
DID-MDN Test Set	0.7781/21.15	0.7896/21.44	0.8352/22.75	0.8422/22.07	0.8622/24.32	0.8978/ 27.33	0.8522/23.05	0.9087/ 27.95	**0.9342/30.05**

Table 3. Comparison on the test set from [42]. SPANet is trained on the synthetic dataset from [42].

Methods	B_a	B_b	B_c	B_d	B_e	B_f
Resnet	✓	✓	✓	✓	✓	✓
Single SAM		✓				
4 SAMs w/o shared weights			✓			
4 SAMs w/ shared weights				✓	✓	✓
Self-Attention branch		✓	✓		✓	✓
Attention Loss		✓	✓	✓		✓
PSNR	37.43	37.43	37.47	37.70	37.39	**38.06**
SSIM	0.9856	0.9854	0.9854	0.9858	0.9856	**0.9867**

Table 4. Internal analysis of the proposed SPANet. The best performance is marked in **bold**.



(a) Input (b) JORDER (c) DID-MDN (d) Our SPANet

Figure 12. Failure case. Our method fails to remove extremely dense rain streaks.

6. Conclusion and Future Work

In this paper, we have presented a method to produce a high-quality clean image from a sequence of real rain images, by considering temporal priors together with human supervision. Based on this method, we have constructed a large-scale dataset of $\sim 29.5K$ rain/clean image pairs that cover a wide range of natural rain scenes. Experiments show that the performances of state-of-the-art CNN-based derainers can be significantly improved by training on the proposed dataset. We have also benchmarked state-of-the-art derainers on the proposed test set. We find that the stochastic distributions of real rain streaks, especially the varying appearances of rain streaks, often fail these methods. To this end, we present a novel spatial attentive network (SPANet) that can learn to identify and remove rain streaks in a local-to-global spatial attentive manner. Extensive evaluations demonstrate the superiority of the proposed method over the state-of-the-art derainers.

Our method does have limitations. One example is given in Figure 12, which shows that our method fails when processing haze-like heavy rain. It is because the proposed dataset generation method fails to select clean pixels from the misty video frames. As a result, the proposed network produces a haze-like result.

Currently, our dataset generation method relies on human judgements. This is partly due to the fact that there are no existing metrics that can assess the generated rain-free images, without clean images for reference. It would be interesting to develop an unsupervised mechanism for this purpose in the future.

Acknowledgement. This work was supported by NSFC (#91748104, #U1811463, #61632006, #61425002, #61751203), Key Research and Development Program of China (#2018YFC0910506), and the Open Project Program of the State Key Lab of CAD&CG (#A1901).

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018. 3

[2] Josue Anaya and Adrian Barbu. Renoir - a benchmark dataset for real noise reduction evaluation. *arXiv:1409.8230*, 2014. 3

[3] Sean Bell, C. Lawrence Zitnick, Kavita Bala, and Ross Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *CVPR*, 2016. 5

[4] Jérémie Bossu, Nicolas Hautière, and Jean-Philippe Tarel. Rain or snow detection in image sequences through use of a histogram of orientation of streaks. *IJCV*, 2011. 2

[5] Yi Chang, Luxin Yan, and Sheng Zhong. Transformed low-rank model for line pattern noise removal. In *ICCV*, 2017. 1, 2

[6] Jie Chen and Lap-Pui Chau. A rain pixel recovery algorithm for videos with highly dynamic scenes. *IEEE TIP*, 2014. 2

[7] Jie Chen, Cheen-Hau Tan, Junhui Hou, Lap-Pui Chau, and He Li. Robust video content alignment and compensation for rain removal in a cnn framework. In *CVPR*, 2018. 3

[8] Yi Lei Chen and Chiou Ting Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *ICCV*, 2013. 2, 3

[9] Shuangli Du, Yiguang Liu, Mao Ye, Zhenyu Xu, Jie Li, and Jianguo Liu. Single image deraining via decorrelating the rain streaks and background scene in gradient domain. *PR*, 2018. 1

[10] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. Clearing the skies: A deep network architecture for single-image rain streaks removal. *IEEE TIP*, 2017. 1, 2, 6, 7, 8

[11] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *CVPR*, 2017. 1, 2, 6, 7, 8

[12] Kshitiz Garg and Shree K. Nayar. Detection and removal of rain from videos. In *CVPR*, 2004. 2

[13] Kshitiz Garg and Shree K Nayar. Vision and rain. *IJCV*, 2007. 2

[14] Shuhang Gu, Deyu Meng, Wangmeng Zuo, and Lei Zhang. Joint convolutional analysis and synthesis sparse representation for single image layer separation. In *ICCV*, 2017. 6, 7

[15] Wei Zhang Huiyou Chang Le Dong Liang Lin Guanbin Li, Xiang He. Non-locally enhanced encoder-decoder network for single image de-raining. In *ACM MM*, 2018. 1

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[17] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 2

[18] Xiaowei Hu, Lei Zhu, Chi-Wing Fu, Jing Qin, and Pheng-Ann Heng. Direction-aware spatial context features for shadow detection. In *CVPR*, 2018. 5

[19] Tai-Xiang Jiang, Ting-Zhu Huang, Xi-Le Zhao, Liang-Jian Deng, and Yao Wang. A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors. In *CVPR*, 2017. 2, 4

[20] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE TIP*, 2012. 1, 2

[21] Jin-Hwan Kim, Jae-Young Sim, and Chang-Su Kim. Video deraining and desnowing using temporal correlation and low-rank matrix completion. *IEEE TIP*, 2015. 2

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[23] Quoc Le, Navdeep Jaitly, and Geoffrey Hinton. A simple way to initialize recurrent networks of rectified linear units. *arXiv:1504.00941*, 2015. 4

[24] Minghan Li, Qi Xie, Qian Zhao, Wei Wei, Shuhang Gu, Jing Tao, and Deyu Meng. Video rain streak removal by multiscale convolutional sparse coding. In *CVPR*, 2018. 2, 3, 4

[25] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *ECCV*, 2018. 1, 2, 6, 7, 8

[26] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael Brown. Rain streak removal using layer priors. In *CVPR*, 2016. 1, 2, 6, 7, 8

[27] Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming Guo. Erase or fill? deep joint recurrent rain removal and reconstruction in videos. In *CVPR*, 2018. 2

[28] Peng Liu, Jing Xu, Jiafeng Liu, and Xianglong Tang. Pixel Based Temporal Analysis Using Chromatic Property for Removing Rain from Videos. *CIS*, 2009. 2

[29] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. In *ICCV*, 2015. 1, 2, 6, 7, 8

[30] Jiayuan Mao, Tete Xiao, Yuning Jiang, and Zhimin Cao. What can help pedestrian detection? In *CVPR*, 2017. 1

[31] motionvfx. https://www.motionvfx.com/mplugs-48.html, 2014. 4

[32] Seonghyeon Nam, Youngbae Hwang, Yasuyuki Matsushita, and Seon Joo Kim. A holistic approach to cross-channel image noise modeling and its application to image denoising. In *CVPR*, 2016. 3

[33] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017. 3

[34] PyTorch. http://pytorch.org. 6

[35] Weihong Ren, Jiandong Tian, Zhi Han, Antoni Chan, and Yandong Tang. Video desnowing and deraining based on matrix decomposition. In *CVPR*, 2017. 2

[36] Varun Santhaseelan and Vijayan K Asari. Utilizing local phase information to remove rain from video. *IJCV*, 2015. 2

[37] Yibing Song, Chao Ma, Xiaohe Wu, Lijun Gong, Linchao Bao, Wangmeng Zuo, Chunhua Shen, Rynson W.H. Lau, and Ming-Hsuan Yang. Vital: Visual tracking via adversarial learning. In *CVPR*, 2018. 1

[38] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment - from error visibility to structural similarity. *IEEE TIP*, 2004. 6

[39] Wei Wei, Lixuan Yi, Qi Xie, Qian Zhao, Deyu Meng, and Zongben Xu. Should we encode rain streaks in video as deterministic or stochastic? In *ICCV*, 2017. 2, 3, 4

[40] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep joint rain detection and removal from a single image. In *CVPR*, 2017. 1, 2, 6, 7, 8

[41] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 5

[42] He Zhang and Vishal M. Patel. Density-aware single image de-raining using a multi-stream dense network. In *CVPR*, 2018. 1, 2, 6, 7, 8

[43] He Zhang, Vishwanath Sindagi, and Vishal Patel. Image de-raining using a conditional generative adversarial network. *arXiv:1701.05957*, 2017. 2

[44] Xiaopeng Zhang, Hao Li, Yingyi Qi, Wee Kheng Leow, and Teck Khim Ng. Rain removal in video by combining temporal and chromatic properties. In *ICME*, 2006. 2, 3

[45] Xueyang Fu Yue Huang Xinghao Ding Zhiwen Fan, Huafeng Wu. Residual-guide feature fusion network for single image deraining. In *ACM MM*, 2018. 1, 2

[46] Fengyuan Zhu, Guangyong Chen, and Pheng-Ann Heng. From noise modeling to blind image denoising. In *CVPR*, 2016. 3

[47] Lei Zhu, Chi-Wing Fu, Dani Lischinski, and Pheng-Ann Heng. Joint bi-layer optimization for single-image rain streak removal. In *ICCV*, 2017. 1, 2, 8

[48] Zhe Zhu, Dun Liang, Songhai Zhang, Xiaolei Huang, Baoli Li, and Shimin Hu. Traffic-sign detection and classification in the wild. In *CVPR*, 2016. 1