

Semi-Supervised Video Deraining with Dynamic Rain Generator

Zongsheng Yue¹, Jianwen Xie², Qian Zhao¹, Deyu Meng^{1,3,*}

¹Xi'an Jiaotong University, Xi'an, China

²Baidu Research, Bellevue, USA

³The Macau University of Science and Technology, Macau, China

zsyam@gmail.com, jianwen@ucla.edu, timmy.zhaoqian@gmail.com, dymeng@mail.xjtu.edu.cn

<https://github.com/zsyOAOA/S2VD>

Abstract

While deep learning (DL)-based video deraining methods have achieved significant success recently, they still exist two major drawbacks. Firstly, most of them do not sufficiently model the characteristics of rain layers of rainy videos. In fact, the rain layers exhibit strong physical properties (e.g., direction, scale and thickness) in spatial dimension and natural continuities in temporal dimension, and thus can be generally modelled by the spatial-temporal process in statistics. Secondly, current DL-based methods seriously depend on the labeled synthetic training data, whose rain types are always deviated from those in unlabeled real data. Such gap between synthetic and real data sets leads to poor performance when applying them in real scenarios. Against these issues, this paper proposes a new semi-supervised video deraining method, in which a dynamic rain generator is employed to fit the rain layer, expecting to better depict its insightful characteristics. Specifically, such dynamic generator consists of one emission model and one transition model to simultaneously encode the spatially physical structure and temporally continuous changes of rain streaks, respectively, which both are parameterized as deep neural networks (DNNs). Further more, different prior formats are designed for the labeled synthetic and unlabeled real data, so as to fully exploit the common knowledge underlying them. Last but not least, we also design a Monte Carlo EM algorithm to solve this model. Extensive experiments are conducted to verify the superiorities of the proposed semi-supervised deraining model.

1. Introduction

Rain is a very common bad weather that exists in many video data. The appearance of rain not only negatively affects the visual quality of video, but also seriously deteri-

orates the performance of subsequent video processing algorithms, e.g., semantic segmentation [38], object detection [9], and autonomous driving [7]. Therefore, as an necessary video pre-processing step, video deraining has attracted increasing attentions in computer vision community.

As an ill-posed inverse problem raised by Garg and Nayar [15], various methods have been proposed to handle the video deraining task. Most of the traditional methods focus on exploiting rational prior knowledge for the background or rain layers so as to obtain a proper separation between them. For example, low-rankness [23, 24, 52] is widely used to encode the temporal correlations of background video. As for rain streaks, many physical characteristics, such as photometric appearance [16], geometrical features [41], chromatic consistency [36], local structure correlations [8] and multi-scale convolutional sparse coding [31], are explored in past years. Different from such deterministic assumptions for rain streaks, Wei *et al.* [52] firstly regard them as random variables, and model them using mixture of Gaussian (MoG) distribution. Albeit substantiated to be effective in some ideal scenarios, these traditional methods are mainly limited by the subjective manually-designed prior knowledges and huge computation burden.

Recently, owing to the powerful nonlinear fitting capability of DNNs, DL-based methods facilitate significant improvements for the video deraining task. The core idea of this methodology is to directly train a derainer parameterized by DNNs based on synthetic rainy/clean video pairs in an end-to-end manner. Most of these methods leverage different technologies, e.g., superpixel alignment [6], dual-level flow [54] and self-learning [56], to extract the clean background from rainy video. In addition, Liu *et al.* [34, 35] design a recurrent network to jointly perform both the rain degradation classification and rain removal tasks.

Even though these DL-based methods have achieved impressive deraining results on some synthetic benchmarks, there still exists large room to further increase their performance and generalization capability in real applications. On

*Corresponding author.

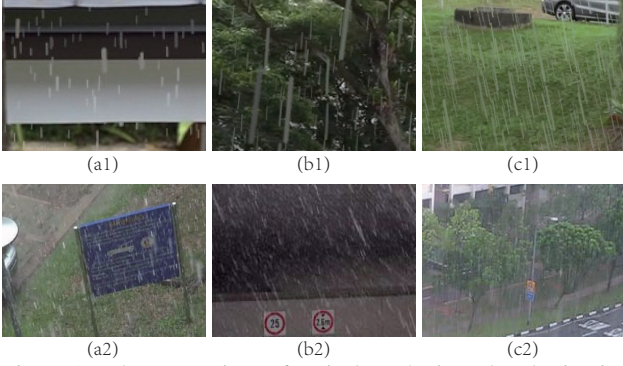


Figure 1. The comparison of typical synthetic and real rainy images in *NTURain* data set. (a1)-(c1): synthetic rainy images, (a2)-(c2): real rainy images.

one hand, most of these methods make efforts on depicting the background, but ignore to model the intrinsic characteristics of rain layer. In fact, the rain layers in video can be understood as a dynamic sequence both in spatial and temporal spaces. Specifically, along spatial dimension, the randomly scattered rain streaks in each frame are with evident physical structures (e.g. direction, scale and thickness), and the rain layers in different frames along temporal dimension correspond to a continuous time series. Therefore, elaborately exploiting and encoding such insightful knowledges underlying rain layers in video data is expected to facilitate the rain removal task.

On the other hand, it is well known that the performance of DL-based methods heavily relies on large amount of pre-collected training data, i.e., rainy/clean video pairs. In fact, due to the high labor cost to obtain such video pairs in real scenes, most of current methods have to use synthetic ones, which are manually simulated based on photo-realistic rendering technique [17] or professional photography and human supervision [49]. Fig. 1 lists several typical frames of synthetic and real rainy images in *NTURain* [6] data set, which is widely used as benchmark in current video deraining methods. It can be easily seen that the rain patterns in synthetic and real rainy images are with evident differences, and the real ones obviously contain more complex and diverse rain types. Because of such gap between synthetic and real data sets, these DL-based methods deteriorate seriously in real cases. To deal with general video deraining task, it is thus critical to build a rational semi-supervised learning manner to sufficiently exploit the common knowledge in labeled synthetic and unlabeled real data.

To address these issues, in this paper we propose a semi-supervised video deraining method, in which a dynamic rain generator is adopted to mimic the generation process of rain layers in video, hopefully better characterizing its intrinsic knowledge simultaneously from spatial and temporal dimensions. Besides, the real rainy videos are taken into consideration in our model as unlabeled data, in order

to achieve more robust deraining results. In summary, the contributions of this work are as follows:

Firstly, we propose a new probabilistic video deraining method, in which a dynamic rain generator, consisting of a transition model and a emission model, is employed to fit the rain layer in videos. Specifically, the transition model is used to encode the continuous changes of rains among adjacent frames, while the emission model maps the state space to the observed rain streaks. To increase the capacities of such generator, both the transition and emission models are parameterized as DNNs.

Secondly, a semi-supervised learning mechanism is designed by constructing different prior formats for labeled synthetic data and unlabeled real data. Specifically, for the labeled synthetic data, the corresponding ground truth rain-free videos are embedded into one elaborate prior distribution as a strong constraint. As for the unlabeled real data, we introduce the 3-D Markov Random Field (MRF) to encode the temporal consistencies and correlations of the underlying background.

Thirdly, a Monte Carlo EM algorithm is designed to solve our model. In the expectation step, the posterior of latent variables are intractable because of the DNNs employed in generator and derainer, thus the Langevin dynamic is adopted to approximate the expectation.

2. Related work

In this section, we give a short recap for the developments on the video/image deraining methods.

2.1. Video Deraining Methods

To the best of our knowledge, Garg and Nayar [15] firstly proposed the problem of video deraining, and developed a rain detector based on the photometric appearance of rain. Later, they further explored the relationships between rain effects and some camera parameters [16–18].

Inspired by these seminal works, various video deraining methods have been proposed in past years, focusing on seeking more rational prior knowledge for the rain or background. For example, both the chromatic property [36, 62] and shape characteristics [2, 3] of rain in time domain have been employed to identify and remove rain layers from the captured rainy videos, while the regular visual effects of rain in global frequency space were also exploited by [1]. Besides, Santhaseelan and Asari [43] employed local phase congruency to detect rain based on chromatic constraints. Notably, Wei *et al.* [52] firstly regarded rain streaks as random variables and model them by patch-based MoG distribution. In addition, matrix/tensor factorization technologies were also very popular in the field of video deraining, mainly used to encode the correlations of background video along time dimension, including [8, 23, 24, 27, 41].

In recent years, DL-based methods represent a new trend along this research line. In [31], Li *et al.* employed the multi-scale convolutional sparse coding to encode the repetitive local patterns under different scales of rain streaks. Chen *et al.* [6] proposed to decompose the scene into superpixels and then align the scene content at superpixel segmentation level, and finally a CNN is used to compensate the lost details and add normal textures to the deraining results. In [35], Liu *et al.* designed a recurrent neural network to jointly implement both of the rain degradation classification and rain removal tasks. And in [34], a hybrid rain model is proposed to model both rain streaks and occlusions. Besides, Yang *et al.* [54] also built a two-stage recurrent networks that utilize dual-level regularizations toward video deraining. Very recently, Yang *et al.* [56] proposed a self-learning manner for this task by taking both temporal correlations and consistencies into consideration.

While DL-based methods have achieved impressive performance on some synthetic benchmarks, they are still very hard to be applied in real applications due to the large gap between their used synthetic data and the real data. Therefore, in order to increase the generalization capacities of deraining model in real task, it is critical to design a semi-supervised learning framework to fully mine the informations both in the labeled synthetic data and unlabeled real data. This paper mainly focuses on this issue.

2.2. Single Image Deraining Methods

For literature comprehensiveness, we also briefly review the single image deraining methods. The single image deraining method can be roughly divided into two categories, i.e., model-based methods and DL-based methods. Most of the model-based methods formulated the deraining task as a decomposition problem between the rain and background layers, and various technologies have been employed to deal with it, such as morphological components analysis [25], non-local means filter [26], and sparse coding [5, 37]. Besides, some prior knowledges of rain and background are also explored in this field, mainly including sparsity and low-rankness [4, 19, 58], narrow directions of rain and the similarities of rain patches [64], and Gaussian mixture model (GMM) [33].

The earliest DL-based method was proposed by Fu *et al.* [12, 13], in which CNNs are adopted to remove rains from the high frequency part of rainy images. Led by these two works, DL-based methods began to dominate the research in this field. Many effective and advanced network architectures [14, 21, 30, 32, 40, 49] were put forward in recent years. And some works attempted to jointly handle the rain removal task with other related tasks, like rain detection [55], rain density estimation [59], so as to obtain better deraining performance. Besides, some useful priors, e.g., multi-scale [22, 57, 63], convolutional sparse cod-

ing [47] and bilevel layer prior [39], were also embedded into the DL-based methods to sufficiently mine the potentials of DNNs. Different from the above methods, Zhang *et al.* [60] and Wang *et al.* [46] both introduced adversarial learning manner to enhance the realistic of the derained images, and Wei *et al.* [51] proposed a semi-supervised deraining model that can be better generalized to real tasks.

Naturally, single image deraining method can be directly used in the video deraining task by taking each video as some independent single images. However, because of ignoring the abundant temporal informations contained in video, it is very hard to obtain satisfied performance using such manner. Thus it is necessary to design rational deraining model dedicated for video data.

3. Semi-Supervised Video Deraining Model

Given a labeled data set $\mathcal{D} = \{\mathcal{Y}^i, \mathcal{X}^i\}_{i=1}^{N_l}$ and an unlabeled data set $\mathcal{U} = \{\mathcal{Y}^i\}_{i=1}^{N_u}$, where \mathcal{Y}^i and \mathcal{X}^i denote the i -th rainy and clean videos, respectively, we aim to construct a semi-supervised probabilistic model based on them and then design an EM algorithm to solve it.

3.1. Model Formulation

Let $\mathcal{Y} = \{\mathcal{Y}_t\}_{t=1}^n$ denote any rainy video in \mathcal{D} or \mathcal{U} , where $\mathcal{Y}_t \in \mathcal{R}^{h \times w}$ is the t -th image frame. Similar to [31, 33], we decompose the rainy video \mathcal{Y} into three parts, i.e.,

$$\mathcal{Y} = f(\mathcal{Y}; W) + \mathcal{R} + \mathcal{E}, \quad \mathcal{E}_{ijt} \sim \mathcal{N}(0, \sigma^2), \quad (1)$$

where $f(\mathcal{Y}; W)$, \mathcal{R} and \mathcal{E} are the recovered rain-free background, rain layer and residual term, respectively, and \mathcal{E}_{ijt} is the element of \mathcal{E} at location (i, j, t) . The residual term is assumed as zero-mean Gaussian distribution with variance σ^2 . $f(\cdot; W)$, which is parameterized by DNNs, denotes a function that maps the observed rainy video to the underlying rain-free background, and is called as “derainer” in this paper. Next, we consider how to model the derainer parameter W and rain layer \mathcal{R} :

Modelling background layer: As is well known, one general prior knowledge for video data is that the rain-free background is with strong correlations and similarities along spatial and temporal dimensions. Therefore, for any rainy video $\mathcal{Y} \in \mathcal{U}$, we encode such knowledge through the following MRF prior distribution for W :

$$p(W) \propto \exp \left(-\rho \sum_{i,j,t} \mathbf{v}^T \boldsymbol{\gamma} \right), \quad (2)$$

where $\mathbf{v} = \begin{bmatrix} |f_{i+1,j,t} - f_{ijt}| \\ |f_{i,j+1,t} - f_{ijt}| \\ |f_{i,j,t+1} - f_{ijt}| \end{bmatrix}$, $\boldsymbol{\gamma} = \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix}$, f_{ijt} denotes the element of $f(\mathcal{Y}; W)$ at location (i, j, t) . ρ and $\boldsymbol{\gamma}$ are both manual hyper-parameters, and the latter represents the strength of smoothness constraint on the spatial and temporal dimensions. As for the rainy video $\mathcal{Y} \in \mathcal{D}$, the known rain-free

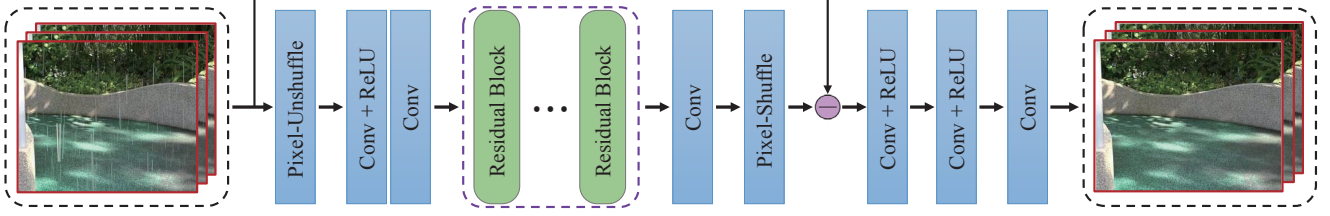


Figure 2. The network architecture for the derainer $f(\cdot; W)$. In this figure, all “Conv”s denote the 3-D convolution layer.

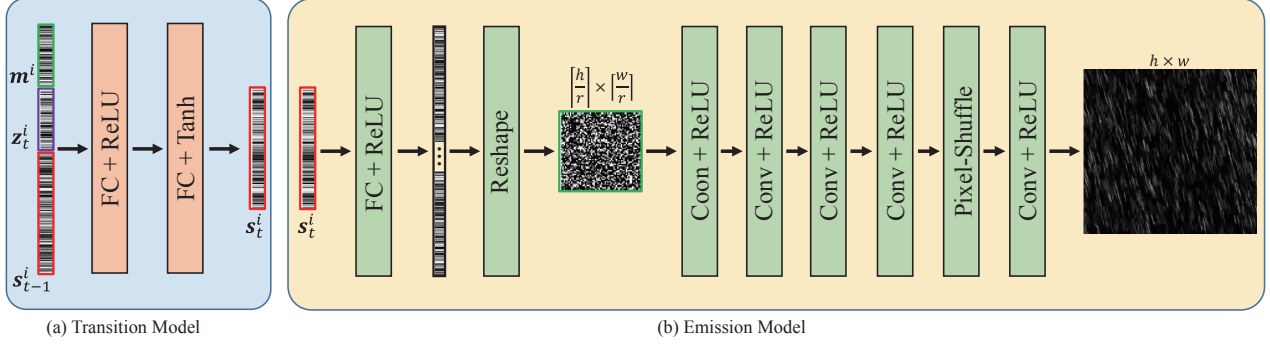


Figure 3. The network architecture of the transition model and emission model in the dynamic rain generator. In this figure, “FC”, “Conv” and “Tanh” denote fully connected, 2-D convolution and hyperbolic tangent layers, respectively. And “Pixel-Shuffle” is the sub-pixel layer [45] with scale factor r .

background \mathcal{X} can be further embedded into Eq. (2) as another strong prior, i.e.,

$$p(W) \propto \exp \left(-\frac{\|f(\mathcal{Y}; W) - \mathcal{X}\|_2}{\varepsilon_0^2} - \rho \sum_{i,j,t} \mathbf{v}^T \boldsymbol{\gamma} \right), \quad (3)$$

where ε_0 is a very small hyper-parameter close to zero.

As for the derainer $f(\cdot; W)$, we adopt a simple network architecture as shown in Fig. 2. Without any special designs, it only contains several 3-D convolution layers and residual blocks [20]. To accelerate the computation, the pixel-unshuffle [61] and pixel-shuffle [45] layers are added to the head and the tail of it, respectively.

Modelling rain layer: Intuitively, the rain layer is a dynamic sequence both in spatial and temporal process, thus we naturally employ the spatial-temporal process [11, 53] in statistics to characterize it. Let’s denote \mathcal{R}_t as the t -th frame of rain layer \mathcal{R} , and then our dynamic rain generator can be formulated as follows,

$$\mathbf{s}_t = F(\mathbf{s}_{t-1}, \mathbf{z}_t; \boldsymbol{\alpha}), \quad (4)$$

$$\mathcal{R}_t = H(\mathbf{s}_t; \boldsymbol{\beta}), \quad (5)$$

where

$$\mathbf{z}_t \sim \mathcal{N}(0, \mathbf{I}), \quad \mathbf{s}_0 \sim \mathcal{N}(0, \mathbf{I}), \quad (6)$$

\mathbf{s}_t represents the hidden state variable in t -th frame, and \mathbf{z}_t the noise vector. Specifically, Eq. (4) is the transition model with parameters $\boldsymbol{\alpha}$ expecting to depict the changes of rains between two adjacent frames, and Eq. (5) is the emission

model with parameters $\boldsymbol{\beta}$ that maps the hidden state space to the observed rain layer. Note that the noise vectors $\{\mathbf{z}_t\}_{t=1}^n$ are independent of each other, encoding the random factors that affect the rains (e.g., wind, camera motion) in the transition from \mathbf{s}_{t-1} to \mathbf{s}_t .

Further more, we extend such generator to an advanced version for multiple rain videos. Specifically, for the i -th rain video $\mathcal{R}^i = \{\mathcal{R}_t^i\}_{t=1}^n$, another vector $\mathbf{m}^i \sim \mathcal{N}(0, \mathbf{I})$ is introduced to account for the variations of rain patterns, and thus the transition model of Eq. (4) can be reformulated as:

$$\mathbf{s}_t^i = F(\mathbf{s}_{t-1}^i, \mathbf{z}_t^i, \mathbf{m}^i; \boldsymbol{\alpha}), \quad (7)$$

where \mathbf{m}^i is fixed for the i -th rain video. For notation convenience, we simply write Eqs. (7) and (5) together as follows:

$$\mathcal{R}^i = G(\mathbf{s}_0^i, \mathbf{z}^i, \mathbf{m}^i; \boldsymbol{\theta}), \quad (8)$$

where $\mathbf{z}^i = \{\mathbf{z}_t^i\}_{t=1}^n$, $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$. In practice, we use the extended version of Eq. (8) to simultaneously fit the rain layers in each mini-batch data.

To increase the capacities of such dynamic generator, both of the transition model and emission model are parameterized as DNNs. Following [53], we used a two-layers mutli-layer perceptron (MLP) in Fig. 3 (a) as the transition model. For the emission model, we elaborately design a CNN architecture that takes the state variable \mathbf{s}_t as input and outputs the rain image as shown in Fig. 3 (b), which is mainly inspired by a recent work [48] that uses CNN as a latent variable model to generate rain streaks.

Remark: The employment of such dynamic generator to fit the rain layers is one of the main contributions of this work, which directly affects the deraining performance of the entire model. Therefore, it is necessary to validate the capabilities of such dynamic generator on simulating the rain layers. To prove this point, we pre-collected some rain layer videos synthesized by commercial Adobe After Effects¹ software from YouTube as source videos, and trained such dynamic generator to recover them. Empirically, we found that such dynamic generator is able to sufficiently mimic the given rain layer videos. Due to page limitation, these experiments are put into the supplementary materials.

3.2. Maximum A Posteriori Estimation

Combining Eqs. (1)-(6), a full probabilistic model is obtained for video deraining. Then our goal turns to maximize the posteriors w.r.t the model parameters W and θ , i.e.,

$$\begin{aligned} \max_{W, \theta} \log p(W, \theta | \mathcal{Y}) &\propto \log p(\mathcal{Y} | W, \theta) + \log p(W) \\ &\triangleq \mathcal{L}(\mathcal{Y}; W, \theta), \end{aligned} \quad (9)$$

where $p(\mathcal{Y} | W, \theta)$ is the likelihood of observed rainy video \mathcal{Y} . According to Eqs. (1) and (8), it can be written as:

$$\begin{aligned} p(\mathcal{Y} | W, \theta) &= \int p(\mathcal{Y} | W, \theta, z) p(z) dz \\ &= \int \mathcal{N}(f(\mathcal{Y}; W) + G(s_0, z; \theta), \sigma^2 I) p(z) dz. \end{aligned}$$

Finally, we directly optimize the problem of Eq. (9) on the whole labeled and unlabeled data sets, i.e.,

$$\max_{W, \theta} \sum_{\mathcal{Y}^i \in \mathcal{D}} \mathcal{L}(\mathcal{Y}^i; W, \theta) + \sum_{\mathcal{Y}^j \in \mathcal{U}} \mathcal{L}(\mathcal{Y}^j; W, \theta). \quad (10)$$

The insight behind Eq. (10) is to learn a general mapping from rainy videos to clean ones, based on large amount of data samples in \mathcal{D} and \mathcal{U} , which is expected to obtain a more efficient and robust derainer than that in traditional inference paradigm implementing on single video.

Most notably, if only considering labeled data set, our method naturally degenerates into a supervised deraining model. However, the addition of unlabeled real data increases the generalization capacity in real deraining tasks as shown in the ablation studies in Sec. 4.2.2.

3.3. Inference and Learning Algorithm

For notation brevity, we only consider one data sample \mathcal{Y} in this part. Inspired by the technology of alternative back-propagation through time [53], a Monte Carlo EM [10] algorithm is designed to maximize $\mathcal{L}(\mathcal{Y}; W, \theta)$, in which one expectation step samples latent variable z from its posterior $p(z | \mathcal{Y})$, and the next maximization step updates the model parameters W and θ based on current sampled z .

¹<https://www.adobe.com/products/aftereffects.html>

Algorithm 1 Inference and learning procedure for S2VD

Input: training data $\mathcal{D} = \{\mathcal{Y}^{b_j}, \mathcal{X}^{b_j}\}_{j=1}^{B_l}$ and $\mathcal{U} = \{\mathcal{Y}^{b_j}\}_{j=B_l+1}^{B_l+B_u}$, where \mathcal{Y}^{b_j} denotes the j -th mini-batch data, number of Langevin steps l .

Output: the derainer parameters W .

- 1: Initialize W and $\theta^{b_j}, z^{b_j}, j = 1, 2, \dots, B_l + B_u$.
 - 2: **while** not converged **do**
 - 3: **for** $j = 1, 2, \dots, B_l + B_u$ **do**
 - 4: Sample the mini-batch data $\{\mathcal{Y}^{b_j}, \mathcal{X}^{b_j}\}$ or \mathcal{Y}^{b_j} .
 - 5: **E-Step:** For each data example \mathcal{Y}^i in current mini-batch \mathcal{Y}^{b_j} , run l steps of Langevin dynamics to sample z^i following Eq. (12).
 - 6: **M-Step:** Update W and θ^{b_j} by Eq. (15).
 - 7: **end for**
 - 8: **end while**
-

E-Step: Let $(W^{\text{old}}, \theta^{\text{old}})$ and $p_{\text{old}}(z | \mathcal{Y})$ denote current model parameters and the posterior under them, we can sample z from $p_{\text{old}}(z | \mathcal{Y})$ using the Langevin dynamic [29]:

$$\begin{aligned} z^{(\tau+1)} &= z^{(\tau)} + \frac{\delta^2}{2} \left[\frac{\partial}{\partial z} \log p_{\text{old}}(z | \mathcal{Y}) \right] \Big|_{z=z^{(\tau)}} + \delta \xi^{(\tau)} \\ &= z^{(\tau)} - \frac{\delta^2}{2} \left[\frac{\partial}{\partial z} g(z) \right] \Big|_{z=z^{(\tau)}} + \delta \xi^{(\tau)} \end{aligned} \quad (11)$$

where

$$g(z) = \frac{1}{2\sigma^2} \left\| \mathcal{Y} - f(\mathcal{Y}; W^{\text{old}}) - G(z, s^0; \theta^{\text{old}}) \right\|_2 + \frac{1}{2} \|z\|_2, \quad (12)$$

τ indexes the time step for Langevin dynamics, δ denotes the step size. And $\xi^{(\tau)}$ is the Gaussian white noise, which is added to prevent trapping into local modes. A key point in Eq. (11) is $\frac{\partial}{\partial z} \log p_{\text{old}}(z | \mathcal{Y}) = \frac{\partial}{\partial z} \log p_{\text{old}}(\mathcal{Y}, z)$, and the right term can be easily calculated.

In practice, for the purpose of avoiding the high computational cost of MCMC, Eq. (11) starts from the previous updated results of z . As for the initialized state vector s_0 and the rain variation vector m of Eq. (8), we also sample them together with z using the Langevin dynamics.

M-Step: Denote the sampled latent variable in E-Step as \tilde{z} , M-Step aims to maximize the approximate upper bound w.r.t. W and θ as follows:

$$\begin{aligned} \max_{W, \theta} \mathcal{Q}(W, \theta) &= \int p_{\text{old}}(z | \mathcal{Y}) \log p(\mathcal{Y}, z | W, \theta) dz + \log p(W) \\ &\approx \log p(\mathcal{Y}, \tilde{z} | W, \theta) + \log p(W). \end{aligned} \quad (13)$$

Equivalently, Eq. (13) can be further rewritten as the following minimization problem, i.e.,

$$\begin{aligned} \min_{W, \theta} \hat{\mathcal{L}}(W, \theta) &= \frac{1}{2\sigma^2} \left\| \mathcal{Y} - f(\mathcal{Y}; W) - G(\tilde{z}, s^0; \theta) \right\|_2 + \\ &\quad \rho \sum_{i,j,t} v^T \gamma + 1_{[\mathcal{Y} \in \mathcal{D}]} \cdot \frac{\|f(\mathcal{Y}; W) - \mathcal{X}\|_2}{\varepsilon_0^2}, \end{aligned} \quad (14)$$

Table 1. PSNR/SSIM results of different methods on the synthetic testing data set of *NTURain*. The best and second best results are highlighted in red and blue, respectively.

Clip No.	Rain		DSC [37]		FastDerain [24]		DDN [13]		PReNet [40]		SpacCNN [6]		SLDNet [56]		S2VD	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
a1	29.71	0.9149	27.15	0.9079	29.29	0.9159	31.79	0.9481	32.13	0.9511	30.57	0.9334	33.72	0.9508	36.39	0.9658
a2	29.30	0.9284	28.84	0.9224	30.21	0.9245	30.34	0.9360	30.41	0.9375	31.29	0.9356	33.82	0.9512	33.06	0.9519
a3	29.08	0.8964	26.73	0.8942	29.94	0.9039	30.70	0.9301	30.73	0.9316	30.63	0.9247	33.12	0.9404	35.75	0.9564
a4	32.62	0.9381	30.58	0.9381	34.69	0.9707	35.77	0.9689	35.77	0.9700	35.30	0.9620	37.35	0.9722	39.53	0.9779
b1	30.03	0.8956	30.06	0.9015	29.35	0.9139	32.53	0.9465	32.66	0.9491	32.26	0.9454	34.21	0.9482	37.34	0.9712
b2	30.69	0.8874	30.85	0.9017	31.90	0.9520	33.89	0.9559	33.74	0.9557	35.11	0.9677	35.80	0.9595	40.55	0.9821
b3	32.31	0.9299	31.30	0.9295	29.28	0.9287	35.38	0.9663	35.34	0.9681	34.69	0.9566	36.34	0.9614	38.82	0.9754
b4	29.41	0.8933	30.61	0.9089	27.70	0.9095	32.62	0.9462	33.17	0.9526	34.87	0.9536	33.85	0.9469	37.53	0.9657
avg.	30.41	0.9108	29.52	0.9130	30.54	0.9255	32.87	0.9497	32.99	0.9519	33.11	0.9475	34.89	0.9540	37.37	0.9683

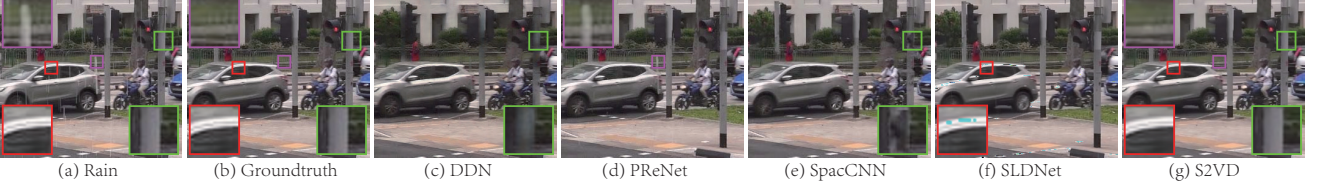


Figure 4. Visual results of different methods on one typical image in *NTURain* synthetic testing data set. From left to right: (a) rainy image, (b) ground truth image, (c)-(g) deraining results by DDN, PReNet, SpacCNN, SLDNet and our S2VD.

where $1_{[\mathcal{Y} \in \mathcal{D}]}$ equals to 1 when \mathcal{Y} comes from the labeled data set \mathcal{D} otherwise 0. Naturally, we can update W and θ by gradient descent based on the back-propagation (BP) algorithm [42] as follows,

$$\Lambda \leftarrow \Lambda - \eta \frac{\partial}{\partial \Lambda} \hat{\mathcal{L}}(W, \theta), \quad \Lambda \in \{W, \theta\}, \quad (15)$$

where η denotes the step size.

Due to the capacity limitation, we empirically find it is very difficult to fit the rain layers in all of the training videos using only one generator defined in Eq. (8). Therefore, we adopt one generator for each mini-batch data. With such strategy, our model performs stably well when setting the mini-batch size as 12 throughout all our experiments. The detailed steps of our algorithm are listed in Algorithm 1.

4. Experimental Results

In this section, we conducted some experiments to evaluate the effectiveness of the proposed semi-supervised video deraining model on synthetic and real data sets. Then we give some addition analysis about it. And we briefly denote our Semi-Supervised Video Deraining model as S2VD in the following presentation.

4.1. Evaluation on Rain Removal Task

Training Details: To train S2VD, we employ the synthesized training data of *NTURain* [6] as labeled data set, which contains 8 rain-free video clips of various scenes. For each rain-free video, 3 or 4 rain layers are synthesized by Adobe After Effects with different settings, and then added to them as rainy ones. As for unlabeled data, 7 real rainy videos without ground truth in the testing data of *NTURain* are employed. To relieve the burden of GPU memory, we used truncated back-propagation through time in training, meaning that the whole training sequence were divided into

different non-overlapped chunks for forward and backward propagation. And the length of chunk is set as 20.

The Adam [28] algorithm is used to optimize the model parameters in M-Step of Algorithm 1. All the network parameters are initialized by [44]. The initialized learning rates for the transition model, emission model and the derainer are set as $1e-3$, $1e-4$ and $2e-4$, respectively, and decayed by multiplying 0.5 after 30 epochs. The mini-batch size is set as 12, and each video is clipped into small blocks with spatial size 64×64 . Note that at the begining 5 epochs, we only update the parameter W to pretrain the derainer, which makes the training more stable. As for the hyperparameters, $\varepsilon_0^2 = 1e-6$, $\rho = 0.5$, $\gamma = [1, 1, 2]^T$, and more analysis on them is presented in Sec. 4.2.

4.1.1 Evaluation on Synthetic Data

We test our S2VD on the synthetic testing data set of *NTU-Rain* [6], which consists of two groups of data sets. The videos in the first group (with prefix “a” in Table 1) are captured by a panning and unstable camera, and those in the second group (with prefix “b” in Table 1) by a fast moving camera with speed range between 20 to 30 km/h. As for the compared methods, six SOTAs are considered, including one model-based image deraining method DSC [37], one model-based video deraining method FastDerain [24], two DL-based image deraining methods DDN [13] and PReNet [40], two DL-based video deraining methods SpacCNN [6] and SLDNet [56]. The average PSNR and SSIM [50] are used as quantitative metrics, which are evaluated only in the luminance channel due to the sensitiveness of us to the luminance information.

Table 1 lists the average PSNR/SSIM results on 8 testing video clips. Evidently, our S2VD method attains the best (7 out of 8) or at least second best (1 out of 8) perfor-

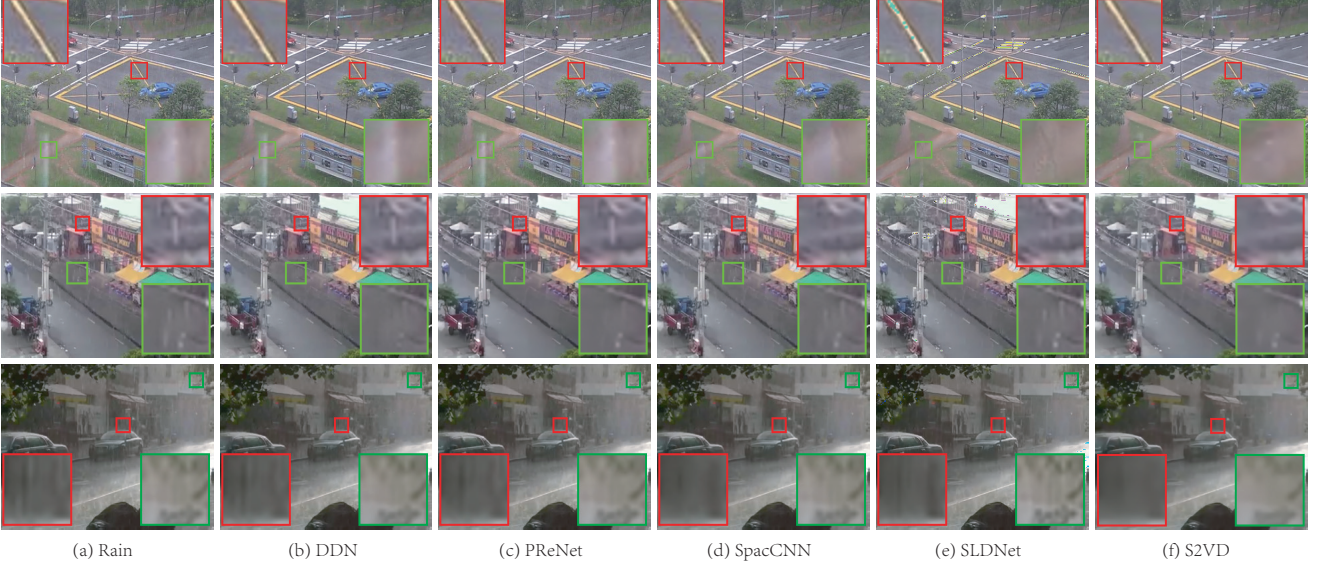


Figure 5. Visual comparisons of different methods on three typical real testing images in *NTURain* [6] (the 1st row) and [31] (the 2nd and 3rd row). From left to right: (a) rainy image, (b)-(f) deraining results by DDN, PReNet, SpacCNN, SLDNet and our S2VD.

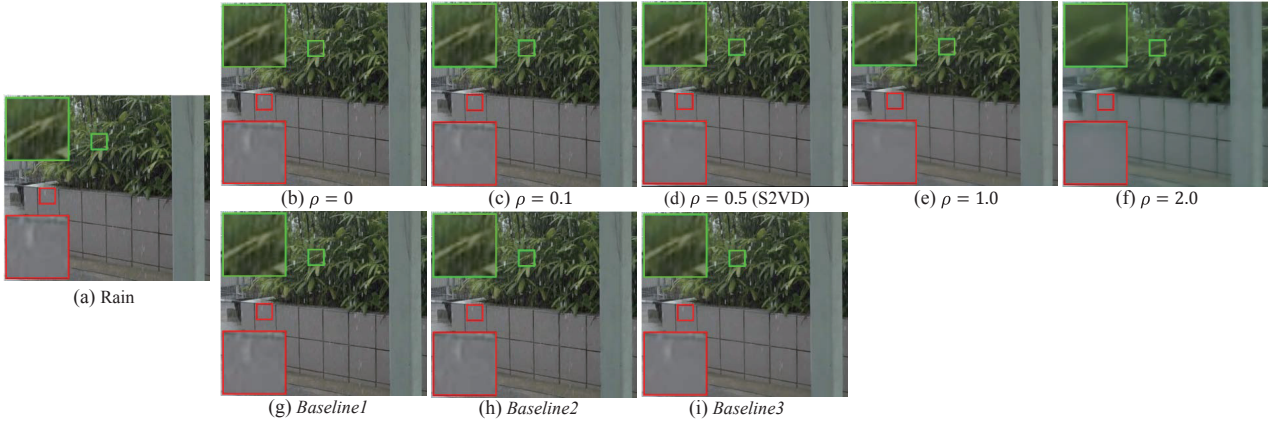


Figure 6. Comparisons of S2VD under different settings. The top row: (a) rainy image, (b)-(f) deraining results of S2VD with different ρ values. The bottom row: (g) *Baseline1*, (h) *Baseline1*, (i) *Baseline3*, and the definitions of these baselines can be seen in Sec. 4.2.2.

mance in all cases. Comparing with current SOTAs (SpacCNN or SLDNet), it achieves at least 2.5dB PSNR and 0.01 SSIM gain. And the visual results are shown in Fig. 4. Note that we only display the results of DL-based methods due to page limitations. It can be observed that: 1) The derained result of PReNet still contains some rain streaks. 2) DDN and SpacCNN both lose some image contents. 3) SLDNet can not finely preserve the original color maps. However, our S2VD evidently alleviate such deficiencies and obtains the closest result to ground truth, which indicates the effectiveness of our proposed semi-supervised deraining model.

4.1.2 Evaluation on Real Data

To further test the generalization of S2VD in real tasks, we test it on two kinds of real rainy videos, i.e., the real testing

Table 2. Average PSNR/SSIM results of S2VD on the synthetic testing dataset of *NTURain* under different ρ values.

Metrics	ρ				
	0	0.1	0.5	1	2
PSNR	38.18	38.05	37.37	35.50	31.55
SSIM	0.9719	0.9713	0.9683	0.9519	0.8947

data set in *NTURain* and several other real rainy videos in [31]. Note that the former is included in our training set as unlabeled data, but the second is not. Fig. 5 illustrates typical deraining results by different methods on such two kinds of data sets. It can be seen that S2VD obviously achieves the best visual results comparing with other methods. Especially, the superiorities in the second data set substantiates that S2VD is able to handle the real rainy videos even that do not appear in the unlabeled data set, such generalization capability should be potentially useful in real deraining task.

Table 3. Average PSNR/SSIM results of three baselines and S2VD on the synthetic testing dataset of *NTURain*.

Metrics	Methods			
	<i>Baseline1</i>	<i>Baseline2</i>	<i>Baseline3</i>	S2VD
PSNR	36.11	37.12	37.96	37.37
SSIM	0.9602	0.9673	0.9717	0.9683

4.2. Additional Analysis

4.2.1 Sensitiveness of hyper-parameter ρ

The hyper-parameter ρ in Eqs. (2) or (3) controls the relative importance of MRF prior in S2VD. The quantitative performance on the synthetic testing data set and the qualitative performance on the real testing data set of *NTURain* under different ρ values are listed in Table 2 and Fig. 6, respectively. On one hand, when ρ becomes gradually larger, the performance on the synthetic testing set tends to decrease as shown in Table 2, since the constraint led by the ground truth in Eq. (3) becomes weaker step by step. On the other hand, MRF prior is able to prevent the derainer over-fitting onto the synthetic data and thus improve the generalization capability in real case, which is sufficiently verified by the visual comparisons in Fig. 6. Comprehensively considering these two aspects, we simply set ρ as 0.5.

4.2.2 Ablation Studies

As shown in Eq. (14), our S2VD degenerates into the Mean Square Error (MSE) loss when $\varepsilon_0 \rightarrow 0$. Comparing with such special case, our model introduces one more likelihood term, one more MRF regularizer and the semi-supervised learning paradigm. To clarify the effect of each part, we compare S2VD with three baselines as follows: 1) *Baseline1*: We only train the derainer with MSE loss on labeled data set as the first baseline. 2) *Baseline2*: We train S2VD with $\varepsilon_0^2 = 1e-6$ and $\rho = 0$ only on labeled data set so as to justify the marginal gain brought up by the likelihood term comparing with MSE (i.e., *Baseline1*). 3) *Baseline3*: On the basis of *Baseline2*, we introduce the MRF regularizer by setting $\rho = 0.5$ as the third baseline.

The quantitative comparisons on synthetic testing data set of *NTURain* are listed in Table 3, and the visual results on real testing data set are also displayed in Fig. 6. In summary, we can see that: 1) The performance improvement (1.01dB PSNR and 0.0071 SSIM) of *Baseline2* beyond *Baseline1* substantiates that the likelihood term plays a substantial role in our model. 2) Under the supervised learning manner, MRF prior is beneficial to our model both in the synthetic and real cases according to the performance of *Baseline3*. 3) Obviously, the addition of unlabeled data in S2VD increase the generalization capability on real task as shown in Fig. 6 (d) and (i). However, it leads to a little deterioration of the performance on synthetic data, mainly because the large gap between the rain types contained in

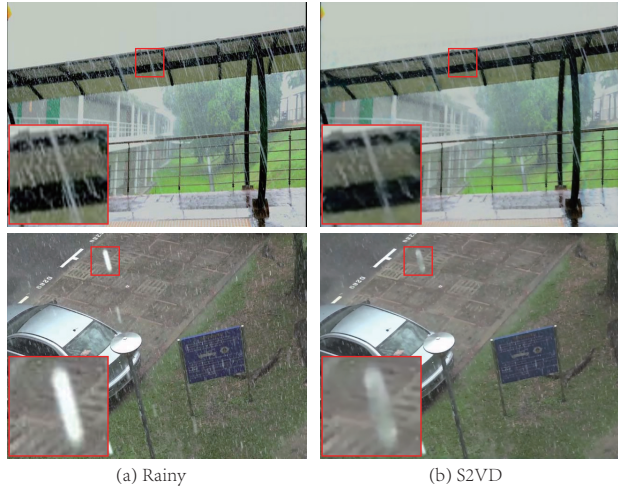


Figure 7. Two typical failed deraining examples by our method. The 1st row represents the large camera motion case, while the 2nd row is with heavy rain streaks.

the synthetic labeled and unlabeled real data sets.

4.2.3 Limitation and Future Direction

Although achieving impressive deraining results as shown above, our method may still fails in some real scenarios, e.g., large camera motion between adjacent frames and heavy rain streaks as shown in Fig. 7. That's mainly because the adopted MRF prior for unlabeled real data is not strong enough to guarantee satisfactory deraining results in such complex cases. Therefore, it is necessary to exploit better prior knowledge in order to handle more general real deraining task in the future.

5. Conclusion

In this paper, we have constructed a dynamic rain generator based on the spatial-temporal process in statistics. With such generator, a semi-supervised video deraining method is proposed. Specifically, we elaborately model the rain layer using such rain generator, which is able to facilitate the rain removal task. In order to handle the generalization issue in real cases, we propose a semi-supervised learning manner to exploit the common knowledge underlying the synthetic labeled and real unlabeled data sets. Besides, a Monte Carlo based EM algorithm is designed to solve it. Extensive experimental results demonstrated the effectiveness of the proposed video deraining method. We believe that our work can benefit to the research of rain removal in computer vision community.

Acknowledgement: This research was supported by the National Key R&D Program of China (2020YFA0713900), the China NSFC projects under contracts 11690011, 61721002, U1811461, 62076196.

References

- [1] Peter C Barnum, Srinivasa Narasimhan, and Takeo Kanade. Analysis of rain and snow in frequency space. *International journal of computer vision*, 86(2-3):256, 2010. 2
- [2] Jérémie Bossu, Nicolas Hautière, and Jean-Philippe 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. 2
- [3] Nathan Brewer and Nianjun Liu. Using the shape characteristics of rain to identify and remove rain from video. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 451–458. Springer, 2008. 2
- [4] Yi Chang, Luxin Yan, and Sheng Zhong. Transformed low-rank model for line pattern noise removal. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1726–1734, 2017. 3
- [5] Duan-Yu Chen, Chien-Cheng Chen, and Li-Wei Kang. Visual depth guided color image rain streaks removal using sparse coding. *IEEE transactions on circuits and systems for video technology*, 24(8):1430–1455, 2014. 3
- [6] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6286–6295, 2018. 1, 2, 3, 6, 7
- [7] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2147–2156, 2016. 1
- [8] Yi-Lei Chen and Chiou-Ting 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
- [9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, volume 1, pages 886–893, 2005. 1
- [10] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977. 5
- [11] Gianfranco Doretto, Alessandro Chiuso, Ying Nian Wu, and Stefano Soatto. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, 2003. 4
- [12] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing*, 26(6):2944–2956, 2017. 3
- [13] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3855–3863, 2017. 3, 6
- [14] Xueyang Fu, Borong Liang, Yue Huang, Xinghao Ding, and John Paisley. Lightweight pyramid networks for image deraining. *IEEE transactions on neural networks and learning systems*, 2019. 3
- [15] K. Garg and S.K. Nayar. Detection and removal of rain from videos. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages 528–535, 2004. 1, 2
- [16] Kshitiz Garg and Shree K Nayar. When does a camera see rain? In *Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1*, volume 2, pages 1067–1074. IEEE, 2005. 1, 2
- [17] Kshitiz Garg and Shree K Nayar. Photorealistic rendering of rain streaks. *ACM Transactions on Graphics (TOG)*, 25(3):996–1002, 2006. 2
- [18] Kshitiz Garg and Shree K Nayar. Vision and rain. *International Journal of Computer Vision*, 75(1):3–27, 2007. 2
- [19] Shuhang Gu, Deyu Meng, Wangmeng Zuo, and Lei Zhang. Joint convolutional analysis and synthesis sparse representation for single image layer separation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1708–1716, 2017. 3
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 4
- [21] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. Depth-attentional features for single-image rain removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3
- [22] Kui Jiang, Zhongyuan Wang, Peng Yi, Chen Chen, Baojin Huang, Yimin Luo, Jiayi Ma, and Junjun Jiang. Multi-scale progressive fusion network for single image deraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3
- [23] 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 *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2827, 2017. 1, 2
- [24] Tai-Xiang Jiang, Ting-Zhu Huang, Xi-Le Zhao, Liang-Jian Deng, and Yao Wang. Fastderain: A novel video rain streak removal method using directional gradient priors. *IEEE Transactions on Image Processing*, 28(4):2089–2102, 2019. 1, 2, 6
- [25] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE transactions on image processing*, 21(4):1742–1755, 2011. 3
- [26] Jin-Hwan Kim, Chul Lee, Jae-Young Sim, and Chang-Su Kim. Single-image deraining using an adaptive nonlocal means filter. In *2013 IEEE International Conference on Image Processing*, pages 914–917. IEEE, 2013. 3
- [27] Jin-Hwan Kim, Jae-Young Sim, and Chang-Su Kim. Video deraining and desnowing using temporal correlation and low-rank matrix completion. *IEEE Transactions on Image Processing*, 24(9):2658–2670, 2015. 2

- [28] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *ICLR 2015 : International Conference on Learning Representations 2015*, 2015. 6
- [29] Paul Langevin. On the theory of brownian motion. 1983. 5
- [30] Guanbin Li, Xiang He, Wei Zhang, Huiyou Chang, Le Dong, and Liang Lin. Non-locally enhanced encoder-decoder network for single image de-raining. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1056–1064, 2018. 3
- [31] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6644–6653, 2018. 1, 3, 7
- [32] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 254–269, 2018. 3
- [33] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown. Rain streak removal using layer priors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2736–2744, 2016. 3
- [34] Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming Guo. D3r-net: Dynamic routing residue recurrent network for video rain removal. *IEEE Transactions on Image Processing*, 28(2):699–712, 2018. 1, 3
- [35] Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming Guo. Erase or fill? deep joint recurrent rain removal and reconstruction in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3233–3242, 2018. 1, 3
- [36] Peng Liu, Jing Xu, Jiafeng Liu, and Xianglong Tang. Pixel based temporal analysis using chromatic property for removing rain from videos. *Computer and information science*, 2(1):53–60, 2009. 1, 2
- [37] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3397–3405, 2015. 3, 6
- [38] Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda G. Shapiro, and Hannaneh Hajishirzi. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 561–580, 2018. 1
- [39] Pan Mu, Jian Chen, Risheng Liu, Xin Fan, and Zhongxuan Luo. Learning bilevel layer priors for single image rain streaks removal. *IEEE Signal Processing Letters*, 26(2):307–311, 2018. 3
- [40] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3937–3946, 2019. 3, 6
- [41] Weihong Ren, Jiandong Tian, Zhi Han, Antoni Chan, and Yandong Tang. Video desnowing and deraining based on matrix decomposition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2838–2847, 2017. 1, 2
- [42] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986. 6
- [43] Varun Santhaseelan and Vijayan K Asari. Utilizing local phase information to remove rain from video. *International Journal of Computer Vision*, 112(1):71–89, 2015. 2
- [44] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *ICLR 2014 : International Conference on Learning Representations (ICLR) 2014*, 2014. 6
- [45] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 4
- [46] Chaoyue Wang, Chang Xu, Chaohui Wang, and Dacheng Tao. Perceptual adversarial networks for image-to-image transformation. *IEEE Transactions on Image Processing*, pages 4066–4079, 2018. 3
- [47] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A model-driven deep neural network for single image rain removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3
- [48] Hong Wang, Zongsheng Yue, Qi Xie, Qian Zhao, and Deyu Meng. From rain removal to rain generation. *arXiv preprint arXiv:2008.03580*, 2020. 4
- [49] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12270–12279, 2019. 2, 3
- [50] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6
- [51] Wei Wei, Deyu Meng, Qian Zhao, Zongben Xu, and Ying Wu. Semi-supervised transfer learning for image rain removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3
- [52] 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 *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2535–2544, 2017. 1, 2
- [53] Jianwen Xie, Ruiqi Gao, Zilong Zheng, Song-Chun Zhu, and Ying Nian Wu. Learning dynamic generator model by alternating back-propagation through time. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5498–5507, 2019. 4, 5
- [54] Wenhan Yang, Jiaying Liu, and Jiashi Feng. Frame-consistent recurrent video deraining with dual-level flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1661–1670, 2019. 1, 3

- [55] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1357–1366, 2017. 3
- [56] Wenhan Yang, Robby T Tan, Shiqi Wang, and Jiaying Liu. Self-learning video rain streak removal: When cyclic consistency meets temporal correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1720–1729, 2020. 1, 3, 6
- [57] Rajeev Yasarla and Vishal M. Patel. Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image de-raining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3
- [58] He Zhang and Vishal M Patel. Convolutional sparse and low-rank coding-based rain streak removal. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1259–1267. IEEE, 2017. 3
- [59] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 695–704, 2018. 3
- [60] He Zhang, Vishwanath Sindagi, and Vishal M. Patel. Image de-raining using a conditional generative adversarial network. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017. 3
- [61] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018. 4
- [62] Xiaopeng Zhang, Hao Li, Yingyi Qi, Wee Kheng Leow, and Teck Khim Ng. Rain removal in video by combining temporal and chromatic properties. In *2006 IEEE international conference on multimedia and expo*, pages 461–464. IEEE, 2006. 2
- [63] Yupei Zheng, Xin Yu, Miaomiao Liu, and Shunli Zhang. Residual multiscale based single image deraining. In *BMVC*, page 147, 2019. 3
- [64] Lei Zhu, Chi-Wing Fu, Dani Lischinski, and Pheng-Ann Heng. Joint bi-layer optimization for single-image rain streak removal. In *Proceedings of the IEEE international conference on computer vision*, pages 2526–2534, 2017. 3