UConNet: Unsupervised Controllable Network for Image and Video Deraining

Junhao Zhuang
University of Electronic Science and Technology of China
Chengdu, China
junhaozhuang@foxmail.com

Yisi Luo
University of Electronic Science and Technology of China
Chengdu, China
yisiluo1221@foxmail.com

Xile Zhao*
University of Electronic Science and Technology of China
Chengdu, China
xlzhao122003@163.com

Taixiang Jiang
Southwestern University of Finance and Economics
Chengdu, China
taixiangjiang@gmail.com

Bichuan Guo
Tsinghua University
Beijing, China
guobichuan@gmail.com

ABSTRACT

Image deraining is an important task for subsequent multimedia applications in rainy weather. Traditional deep learning-based methods rely on the quantity and diversity of training data, which is hard to cover all complex real-world rain scenarios. In this work, we propose the first Unsupervised Controllable Network (UConNet) to flexibly tackle different rain scenarios by adaptively controlling the network at the inference stage. Specifically, our unsupervised network takes the physics-based regularizations as the unsupervised loss function. Then, we sensibly derive the relationship between trade-off parameters of the loss function and the weightings of feature maps. Based on this relationship, our learned UConNet can be flexibly customized for different rain scenarios by controlling the weightings of feature maps at the inference stage. Alternatively, these weightings can also be efficiently determined by a learned weightings recommendation network. Extensive experiments for image and video deraining show that our method achieves very promising effectiveness, efficiency, and generalization abilities as compared with state-of-the-art methods.

CCS CONCEPTS

• **Computing methodologies** → **Reconstruction**.

KEYWORDS

Unsupervised, controllable network, image and video deraining

ACM Reference Format:
Junhao Zhuang, Yisi Luo, Xile Zhao, Taixiang Jiang, and Bichuan Guo. 2022. UConNet: Unsupervised Controllable Network for Image and Video Deraining. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3503161.3547772

*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MM '22, October 10–14, 2022, Lisboa, Portugal
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3547772

1 INTRODUCTION

Removing rain from a single image is a classical and challenging low-level visual task [24, 52]. The classical rain degradation model [33, 58] assumes that the rainy image $\mathbf{O}$ is an addition of the clean image $\mathbf{B}$ and the rain $\mathbf{R}$, i.e.,

$$\mathbf{O} = \mathbf{B} + \mathbf{R}. \tag{1}$$

The goal of image deraining is to estimate the underlying clean image $\mathbf{B}$ and rain $\mathbf{R}$ from the observed $\mathbf{O}$, which is a typical ill-posed inverse problem. Existing image deraining methods can be roughly categorized into three classes: model-based methods, deep learning-based methods, and their integrations. In this paper, we are interested in the last one: how to elegantly and cleverly combine the model-based and deep learning-based methods to utilize their respective advantages—the generalization abilities of model-based methods and the representation abilities and efficiency of deep learning-based methods. This is curial to make the deraining algorithm more generalized and efficient for real world applications.

Model-Based Deraining Methods In the era before deep learning, the deraining task was usually formulated as optimization problems [1, 7], where the modeling of the clean image and rain streaks are especially important. Early techniques included sparse coding [18, 29, 62], which approximated different layers by discriminative codes over learned dictionaries. The low-rankness [1, 6, 20, 62] was widely adopted since both the rain streaks and clean images were supposed to exhibit some low-rank structures. The non-local means filter [19] was also developed for deraining. Li et al. [26] utilized Gaussian mixture models to depict the complex rain streaks. Some directional prior of rain streaks and clean images [7, 69] was further considered for deraining. However, model-based methods have two limitations. First, these methods are based on shallow operators, which may lack representation abilities for real-world complex rain scenarios. Second, many model-based methods are not efficient enough, since they rely on iterative optimization algorithms, such as alternating direction method of multipliers (ADMM) [7], alternating minimization [26], and greedy pursuit [29].

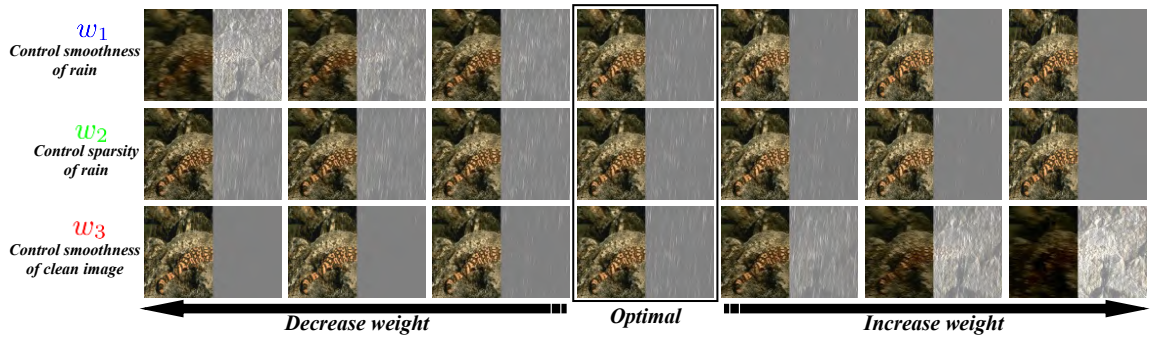

Figure 1: The deraining results and residual rain maps. Our learned UConNet has three controllable weightings $\{w_i\}_{i=1}^3$ at the inference stage. We control each weighting (by increasing or decreasing its value) and fix others to achieve different deraining results. UConNet can handle different rain scenarios by controlling these weightings, leading to high generalization abilities.

Deep Learning-Based Deraining Methods In recent years, deep learning-based methods were extensively studied for deraining. Based on convolutional neural networks (CNNs), these methods studied some popular deep learning architectures including attention modules [13, 31, 37, 42, 66], variational autoencoder [9], aggregation network [3, 25], fusion network [15, 16, 38], generative adversarial network [32, 48], recurrent network [34, 54, 67], and some deep learning strategies such as representation learning [2, 4, 39], residual learning [55, 67, 68], transfer learning [47, 56], semi-supervised learning [14, 57], to name but a few [12, 35, 46, 49, 50]. Apart from these, other deep learning-based methods focused more on the structures of rainy images, such as vapor and haze removal [43, 44], details preserving [8, 10, 11], rain detection [51], rain density estimation [45, 63], rain generation [41, 58], and others [23, 30, 33, 60, 65]. However, all these deep learning-based methods learn a deep network as a deterministic mapping, i.e., at the inference stage, the learned network is uncontrollable and is expected to directly map the rainy image to a clean one without manual adjustment. When encountered with out-of-distribution data, these uncontrollable networks would unavoidably lack generalization abilities and their performance could be unstable.

Integrations of Model and Deep Learning Recently, there are few attempts to integrate the model-based and deep learning-based methods for deraining. Liu et al. [28] combined the optimization with deep networks under the plug-and-play framework for image deraining, which was based on iterative optimization. Wang et al. [40] unfolded an optimization process to a deep neural network for interpretable deep image deraining. Yu et al. [59] proposed an unsupervised image deraining method by leveraging the model-based regularizations as the unsupervised loss of the network. These strategies [40, 59] achieve good interpretation and enhance the generalizations abilities of deep networks. However, the learned networks [40, 59] are still deterministic mappings and are uncontrollable at the inference stage.

In real-world rain scenarios, the structures of rain streaks and images are complex. The learned deraining networks highly depend on the quantity and diversity of training data, which is hard to cover all the different complex rain scenarios and thus their generalization abilities are limited. To address this issue, we aim to design a controllable network that can be adaptively tunned at the inference stage to meet different rain scenarios, which can largely enhance the generalization abilities of the deraining network.

In this work, we build an unsupervised image deraining network which is the first one that can be controlled at the inference stage for deraining. The key methodology is that we estabilish a relationship between trade-off parameters of the loss function $\{\beta_i\}_{i=1}^3$ and weightings of some feature maps $\{w_i\}_{i=1}^3$ (see Eq. (9)). This relationship allows us to control the weightings $\{w_i\}_{i=1}^3$ at the inference stage, which is equivalent to controlling the underlying trade-off parameters $\{\beta_i\}_{i=1}^3$, see Fig. 1. Our network is unsupervisedly trained using hand-crafted regularizations (see. Sec. 2.1) and thus is termed as the Unsupervised Controllable Network (UConNet). The contributions of this paper are:

- We propose the UConNet for image deraining. First, we design physics-based regularizations as the unsupervised loss function by solely using observed rainy images as training data. Then, we establish a relationship between trade-off parameters of the loss function and weightings of some feature maps. At the inference stage, the learned UConNet can handle different rain scenarios by adaptively controlling these weightings or using the suggested weightings by the weightings recommendation network, leading to high generalization abilities. Moreover, the learned UConNet can be flexibly controlled to balance the rain removal and details preserving, leading to promising deraining results.
- We evaluate UConNet on benchmark synthetic and real-world datasets. The effectiveness, efficiency, and generalization abilities of UConNet are demonstrated quantitatively and qualitatively. We also extend our UConNet to video deraining by further incorporating the optical flow. Since rain scenarios in videos would vary in different frames, the demand for controllability is even more urgent, which exactly fits the advantages of our UConNet. The experimental results show that our method achieves state-of-the-art performances as compared with other video deraining methods.

2 THE PROPOSED METHOD

In this section, we first introduce the unsupervised loss function for image deraining. Then, we propose the controllable module by establishing a relationship between trade-off parameters of the loss function and weightings of feature maps. Finally, we present the rain direction estimation network and content-aware weightings recommendation network to improve the applicability of UConNet. The overall flowchart of our UConNet is illustrated in Fig. 3.

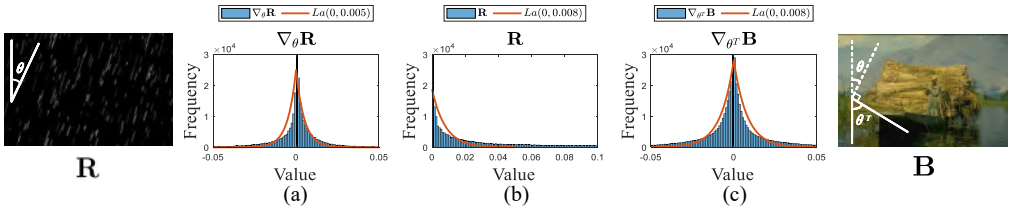

Figure 2: Justifications of our physics-based regularizations. (a) The gradient map of rain streaks R along the rain direction θ follows the Laplacian distribution. (b) The rain streaks R follow the Laplacian distribution. (c) The gradient map of the clean image B along the orthogonal direction of the rain direction, i.e., θ^T, follows the Laplacian distribution. Here, the real rain streaks R are captured by a camera in the black background [22].

2.1 Unsupervised Loss Function

2.1.1 Problem Formulation. Suppose that we are given a rainy image $\mathbf{O} \in \mathbb{R}^{H\times W}$. Our goal is to estimate the underlying $\mathbf{B}$ and $\mathbf{R}$ by maximizing the posterior $P(\mathbf{B},\mathbf{R}|\mathbf{O})$ (with the assumption that $\mathbf{B}$ and $\mathbf{R}$ are independent):

$$\arg\max_{\mathbf{B},\mathbf{R}} \log P(\mathbf{B},\mathbf{R}|\mathbf{O}) = \arg\max_{\mathbf{B},\mathbf{R}} \log P(\mathbf{O}|\mathbf{B},\mathbf{R}) + \log P(\mathbf{B}) + \log P(\mathbf{R}). \tag{2}$$

According to the basic rain model (1), the log likelihood $\log P(\mathbf{O}|\mathbf{B},\mathbf{R})$ is re-formulated as a fidelity constraint $\mathbf{O} = \mathbf{B} + \mathbf{R}$:

$$\arg\max_{\mathbf{B},\mathbf{R}} \log P(\mathbf{B}) + \log P(\mathbf{R}) \quad \text{s.t.} \quad \mathbf{O} = \mathbf{B} + \mathbf{R}, \tag{3}$$

where $P(\mathbf{B})$ and P($\mathbf{R}$) respectively correspond to the regularization term of clean image and rain streaks.

2.1.2 Regularization Terms. We form the unsupervised loss function by elaborately designing three regularization terms:

- **(i)** The piece-wise smoothness of rain. We discover that the rain streaks are piece-wise smooth along their landing directions (see Fig. 2(a)). Specifically, suppose that the rain direction is θ, it is reasonable to assume that each pixel of the gradient map of $\mathbf{R}$ along the direction θ follows i.i.d. Laplacian distributions with zero mean, i.e., $(\nabla_\theta \mathbf{R})_{i,j} \sim La(0, \frac{1}{\beta_1})$ $(i = 1, 2, \cdots, H.\ j = 1, 2, \cdots, W)$, where ∇_θ denotes the derivative operator along the direction θ (see supplementary materials for detailed definition of ∇_θ). This forms the first regularization term $\beta_1\|\nabla_\theta \mathbf{R}\|_{\ell_1}$.
- **(ii)** The sparsity of rain. We discover that rain streaks appear in few places in the image and thus are sparse (see Fig. 2(b)). So rain streaks also follow the Laplacian distribution $(\mathbf{R})_{i,j} \sim La(0, \frac{1}{\beta_2})$, which forms the second regularization term $\beta_2\|\mathbf{R}\|_{\ell_1}$.
- **(iii)** The piece-wise smoothness of clean image. We notice that the clean image $\mathbf{B}$ is piece-wise smooth along the orthogonal direction of the rain direction θ (see Fig. 2(c)). Thus, the gradient map of $\mathbf{B}$ along the direction θ^T follows the Laplacian distributions $(\nabla_{\theta^T}\mathbf{B})_{i,j} \sim La(0, \frac{1}{\beta_3})$, where θ^T denotes the orthogonal direction of θ. This forms th third regularization term $\beta_3\|\nabla_{\theta^T}\mathbf{B}\|_{\ell_1}$.

According to the above analysis, the final optimization model corresponding to (3) is formulated as

$$\arg\min_{\mathbf{B},\mathbf{R}} \beta_1\|\nabla_\theta \mathbf{R}\|_{\ell_1} + \beta_2\|\mathbf{R}\|_{\ell_1} + \beta_3\|\nabla_{\theta^T}\mathbf{B}\|_{\ell_1}, \ \text{s.t.}\ \mathbf{O} = \mathbf{B} + \mathbf{R}, \tag{4}$$

which can be equivalently formulated as

$$\arg\min_{\mathbf{R}} \beta_1\|\nabla_\theta \mathbf{R}\|_{\ell_1} + \beta_2\|\mathbf{R}\|_{\ell_1} + \beta_3\|\nabla_{\theta^T}(\mathbf{O}-\mathbf{R})\|_{\ell_1}, \tag{5}$$

where the optimization variable is the rain $\mathbf{R}$ and the clean image is obtained via $\mathbf{B} = \mathbf{O} - \mathbf{R}$.

2.1.3 Formulations of Unsuperviseed Loss Function. In traditional model-based methods, iterative algorithms such as the ADMM [7] can be directly adopted to solve model (5). However, directly solving model (5) using numerical algorithms may lask representation abilities for real-world complex rain scenarios, since the model and algorithms are based on shallow operators. Also, iterative algorithms require many iterations and large computational costs.

To address these issues, we employ a deep CNN and train it using the three regularization terms in (5). The learned network can be regarded as a fast solver of (5). Given the rainy image $\mathbf{O}$ and the rain direction θ, we hope that the learned network $\mathcal{N}_\omega(\mathbf{O},\theta)$ (parameterized by ω) can map $\mathbf{O}$ and θ to the approximate solution of model (5):

$$\mathcal{N}_\omega(\mathbf{O},\theta) \approx \arg\min_{\mathbf{R}} \beta_1\|\nabla_\theta \mathbf{R}\|_{\ell_1} + \beta_2\|\mathbf{R}\|_{\ell_1} + \beta_3\|\nabla_{\theta^T}(\mathbf{O}-\mathbf{R})\|_{\ell_1}, \tag{6}$$

where the network output is expected to be the rain map $\mathcal{N}_\omega(\mathbf{O},\theta) = \mathbf{R}$ and the clean image is obtained via $\mathbf{B} = \mathbf{O} - \mathbf{R}$.

To learn such a neural network $\mathcal{N}(\mathbf{O},\theta)$, we only need to collect abundant rainy images $\{O_n\}_{n=1}^N$ with their corresponding rain directions $\{\theta_n\}_{n=1}^N$, and form an unsupervised loss function as

$$\begin{aligned}\mathcal{L}_\omega = &\frac{1}{N}\sum_{n=1}^{N} \beta_1\|\nabla_{\theta_n}\mathcal{N}_\omega(\mathbf{O}_n,\theta_n)\|_{\ell_1} + \beta_2\|\mathcal{N}_\omega(\mathbf{O}_n,\theta_n)\|_{\ell_1} + \\ &\beta_3\|\nabla_{\theta_n^T}(\mathbf{O}_n - \mathcal{N}_\omega(\mathbf{O}_n,\theta_n))\|_{\ell_1},\end{aligned} \tag{7}$$

where N denotes the total number of training samples. By minimizing the loss (7), $\mathcal{N}_\omega$ can be unsupervisedly learned as a fast approximator of the solution of (5). Given the approximator network $\mathcal{N}_\omega$, only one inference forward propagation is needed for deraining, which is obtained by computing $\mathcal{N}_\omega(\mathbf{O},\theta)$ (the rain layer) and $\mathbf{O} - \mathcal{N}_\omega(\mathbf{O},\theta)$ (the clean image).

2.2 Controllable Network

Although the formulation of the unsupervised loss function (7) has taken the respect advantages of model-based methods (generalization abilities) and deep learning (efficiency and representation abilities), the loss function (7) has a critical issue: the determination of the trade-off parameters $\{\beta_i\}_{i=1}^3$. A simple strategy is to manually set the trade-off parameters as some deterministic values in the

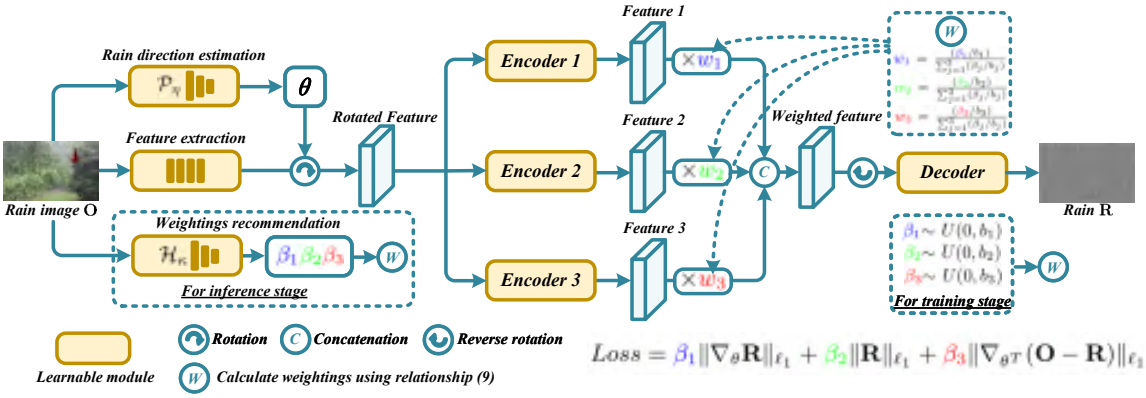

Figure 3: The flowchart of our UConNet $\mathcal{N}_\omega$. At the training stage, we learn a relationship between trade-off parameters $\{\beta_i\}_{i=1}^3$ and weightings of feature maps $\{w_i\}_{i=1}^3$. At the inference stage, given a rainy image, we can control the weightings $\{w_i\}_{i=1}^3$ to handle different rain scenarios.

training phase [59]. This strategy implies that at the inference stage, the network can only provide deraining results corresponding to fixed $\{\beta_i\}_{i=1}^3$, since the trade-off parameters are not controllable at the inference stage. This limits its generalization abilities for different rain scenarios. The underlying challenge of building a controllable network is how to embed the trade-off parameters into the forward propagation of the network.

To address this issue, we derive a relationship between the trade-off parameters $\{\beta_i\}_{i=1}^3$ and weightings of feature maps $\{w_i\}_{i=1}^3$ (see Fig. 3). The ideal result is that at the inference stage, by changing the weightings $\{w_i\}_{i=1}^3$, the network $\mathcal{N}_\omega$ can output the deraining result corresponding to different trade-off parameters $\{\beta_i\}_{i=1}^3$, and thus can handle different rain scenarios. To achieve the controllability, in the training phase, we randomly select the values of $\{\beta_i\}_{i=1}^3$ for each training sample. Specifically, for the n-th training sample, we select the trade-off parameters $\{\beta_{i,n}\}_{i=1}^3$ as

$$\beta_{i,n} \sim U(0, b_i) \ (i = 1, 2, 3), \tag{8}$$

where $U(0, b_i)$ $(i = 1, 2, 3)$ are the uniform distributions with $\{b_i\}_{i=1}^3$ denote the parameters of the uniform distributions. Then, we assign the weightings of feature maps according to the following analytical relationship:

$$w_{i,n} = \frac{(\beta_{i,n}/b_i)}{\sum_{j=1}^3 (\beta_{j,n}/b_j)} \ (i = 1, 2, 3), \tag{9}$$

where the scaling factor $1/b_i$ ensures that the three weightings $\{w_{i,n}\}_{i=1}^3$ are in similar orders of magnitude. Based on this relationship, we re-formulate the unsupervised loss function (7) as

$$\begin{aligned} \mathcal{L}_\omega = & \frac{1}{N}\sum_{n=1}^{N} \beta_{1,n}\|\nabla_{\theta_n}\mathcal{N}_\omega(\mathbf{O}_n, \theta_n, \{w_{i,n}\}_{i=1}^3)\|_{\ell_1} + \\ & \beta_{2,n}\|\mathcal{N}_\omega(\mathbf{O}_n, \theta_n, \{w_{i,n}\}_{i=1}^3)\|_{\ell_1} + \\ & \beta_{3,n}\|\nabla_{\theta_n{}^T}(\mathbf{O}_n - \mathcal{N}_\omega(\mathbf{O}_n, \theta_n, \{w_{i,n}\}_{i=1}^3))\|_{\ell_1}. \end{aligned} \tag{10}$$

Here, the inputs of the network $\mathcal{N}_\omega$ are the rainy image $\mathbf{O}$, the rain direction θ, and the three weightings $\{w_i\}_{i=1}^3$. The output $\mathcal{N}_\omega(\mathbf{O}_n, \theta_n, \{w_{i,n}\}_{i=1}^3)$ is the estimated rain map $\mathbf{R}$.

At the inference stage, we can explicitly control the weightings $\{w_i\}_{i=1}^3$ to obtain different rain maps $\mathbf{R}$ (see Fig. 1 for examples). The underlying methodology is that controlling the weightings $\{w_i\}_{i=1}^3$ is equivalent to controlling the trade-off parameters $\{\beta_i\}_{i=1}^3$ under the relationship (9), where different trade-off parameters $\{\beta_i\}_{i=1}^3$ correspond to different rain maps $\mathbf{R}$. The controllability brings significant advantages that the network is not deterministic but can be manually controlled to meet different rain scenarios, which ensures high generalization abilities.

In our work, we set $b_1 = 28$, $b_2 = 1$, and $b_3 = 16$ for all training samples to ensure the balance between different regularizations (b_1, b_2, and b_3 respectively correspond to the smothness of rain streaks, the sparsity of rain streaks, and the smoothness of the clean image). We adopt the Adam optimizer for training $\mathcal{N}_\omega$. The learning rate is set to 0.0001 and the batch size is 16. We train 10 epochs. The training datasets are specified in the experiments.

2.2.1 Network Architectures. As shown in Fig. 3, the UConNet $\mathcal{N}_\omega$ consists of a feature extraction network, three encoders, and one decoder. In the following, we respectively introduce their structures.

- The feature extraction network maps the input rainy image of size $1 \times 1 \times H \times W$ (We respectively feed each channel of the RBG image into the network) to a feature tensor of size $1 \times 16 \times H \times W$. It consists of four residual blocks (each residual block consists of two convolutional layers, batch normalization, and ReLU activation). Before the encoding phase, the feature tensor is rotated according to the given rain direction θ. The rotation in the redundant feature space can thoroughly avoid information loss as compared to the rotation in the image space [17, 59].
- The encoding stage contains three parallel encoders. The input feature tensor is processed separately and multiplied by the corresponding weightings $\{w_i\}_{i=1}^3$, and then concatenated together to form a weighted feature tensor of size $1 \times 48 \times H \times W$. Each encoder is composed of four residual blocks. The reverse rotation is employed to rotate the feature tensor back before the decoding stage.
- The decoder consists of seven residual blocks, which map the feature tensor to the desired rain streaks of size $1 \times 1 \times H \times W$.

2.3 Rain Direction Estimation and Weightings Recommendation Networks

Although the learned UConNet can be adaptively controlled to handle different rain scenarios, it needs manual estimations of the rain direction θ and the weightings $\{w_i\}_{i=1}^3$. Therefore, we introduce the rain direction estimation network and weightings recommendation network to ease the burden of rain direction estimation and weightings selection at the inference stage. These two networks have simpler structures than the main UConNet, and thus we use supervised training strategies to learn these two networks.

2.3.1 Rain Direction Estimation Network. To estimate the rain direction θ of each rainy image $\mathbf{O}$, we propose to learn a rain direction estimation network $\mathcal{P}_\eta$ parameterized by η. The loss function for training the rain direction estimation network $\mathcal{P}_\eta$ is

$$\mathcal{L}_\eta = \frac{1}{M}\sum_{n=1}^{M}(\mathcal{P}_\eta(\mathbf{O}_n) - \theta_n)^2, \tag{11}$$

where θ_n denotes the real rain direction of $\mathbf{O}_n$ and $\mathcal{P}_\eta(\mathbf{O}_n)$ denotes the estimated rain direction in the n-th sample. M denotes the total number of training samples. The learned $\mathcal{P}_\eta$ can be used to provide the estimated rain direction θ to the main network $\mathcal{N}_\omega(\mathbf{O}, \theta, \{w_i\}_{i=1}^3)$ for deraining, see Fig. 3.

The network $\mathcal{P}_\eta$ is consists of four downsampling layers, which reduce the spatial dimension of data from 128×128 to 1×1. Each downsampling layer is composed of one max pooling layer for downsampling and two convolutional layers. Each convolutional layer is followed by batch normalization and ReLU activation. A convolutional operator is added in the last layer to reduce the channel size to 1. We select $M = 8000$ training samples from the DID-MDN dataset [63]. Specifically, we first randomly perform synthetic rain streaks with known rain directions $\{\theta_n\}_{n=1}^M$ on all clean images in the DID-MDN dataset. Then, we randomly crop a 128×128 patch in each synthetic rainy image as the input $\mathbf{O}_n$. We use the Adam optimizer for training. The learning rate of training $\mathcal{P}_\eta$ is 0.0001 and the batch size is 64.

2.3.2 Content-aware Weightings Recommendation Network. We remark that the training of UConNet $\mathcal{N}_\omega$ dose not need the manual selection of weightings $\{w_i\}_{i=1}^3$, since the weightings are randomly selected at the training stage. However, at the inference stage, the weightings $\{w_i\}_{i=1}^3$ are controllable and need manual selection.

To ease the burden of selecting appropriate weightings $\{w_i\}_{i=1}^3$ at the inference stage, we learn a content-aware weightings recommendation network $\mathcal{H}_\kappa$ (parameterized by κ) to automatically suggest the appropriate weightings for each given rainy image. Since there is no ground truth labels for the optimal weightings $\{w_i\}_{i=1}^3$ for each rainy image, it is hard to directly train the weightings recommendation network $\mathcal{H}_\kappa$. To address this issue, we turn to use the ground truth clean image labels and adopt the learned UConNet $\mathcal{N}_\omega$ to train the recommendation network $\mathcal{H}_\kappa$. Specifically, using the well-trained UConNet $\mathcal{N}_\omega$, the loss function for training $\mathcal{H}_\kappa$ is

$$\mathcal{L}_\kappa = \frac{1}{P}\sum_{n=1}^{P}\|(\mathbf{O}_n - \mathcal{N}_\omega(\mathbf{O}_n, \theta_n, \{w_{i,n}\}_{i=1}^3)) - \mathbf{B}_n\|_F^2, \tag{12}$$

Table 1: The average results on R100L dataset. UConNet (a) denotes that the weightings $\{w_i\}_{i=1}^3$ are fixed. UConNet (b) denotes that the weightings are suggested by the weightings recommendation network. UConNet (c) denotes that the weightings are tunned to obtain the best PSNR value.

Metric	Rainy	UConNet (a)	UConNet (b)	UConNet (c)
PSNR	25.83	29.03	<u>31.28</u>	**31.40**
SSIM	0.8300	0.9231	<u>0.9351</u>	**0.9372**
Time (second)	--	0.1214	0.1512	2.1095

Table 2: The average results on R100L dataset. UConNet (d) denotes that $w_1 = 0$ and w_2, w_3 are searched to obtain the best PSNR value. UConNet (e) denotes that $w_2 = 0$ and w_1, w_3 are searched to obtain the best PSNR value. UConNet (f) denotes that $w_3 = 0$ and w_1, w_2 are searched to obtain the best PSNR value. UConNet (c) denotes that all weightings are searched to obtain the best PSNR value.

Metric	Rainy	UConNet (d)	UConNet (e)	UConNet (f)	UConNet (c)
PSNR	25.832	29.64	<u>30.14</u>	25.832	**31.40**
SSIM	0.8300	0.9019	<u>0.9092</u>	0.8300	**0.9372**
Time (second)	--	0.9826	1.0103	1.1201	2.1095

where

$$w_{i,n} = \frac{(\beta_{i,n}/b_i)}{\sum_{j=1}^3(\beta_{j,n}/b_j)}\ (i = 1, 2, 3),\ \{\beta_{i,n}\}_{i=1}^3 = \mathcal{H}_\kappa(\mathbf{O}_n), \tag{13}$$

and $\{\mathbf{O}_n, \mathbf{B}_n\}_{n=1}^P$ denote P pairs of rainy images and corresponding ground-truth clean images. The loss function (12) minimizes the mean square error between the estimed clean image $\mathbf{O}_n - \mathcal{N}_\omega(\mathbf{O}_n, \theta_n, \{w_{i,n}\}_{i=1}^3)$ and the ground truth clean image $\mathbf{B}_n$, where the weightings $\{w_{i,n}\}_{i=1}^3$ are obtained using the weightings recommendation network $\mathcal{H}_\kappa$. Here, the parameters of the UConNet $\mathcal{N}_\omega$ are fixed. We only update the parameters of the weightings recommendation network $\mathcal{H}_\kappa$ using the loss function (12).

We use the Adam optimizer for training $\mathcal{H}_\kappa$. The learning rate is 0.0001 and the batch size is 32. The training data contains $P = 8000$ pairs of rainy images and clean images in the DID-MDN dataset [63]. The network structure of $\mathcal{H}_\kappa$ contains two serial sub-networks. The first sub-network is the fixed feature extraction network and the three encoders of the UConNet $\mathcal{N}_\omega$ (see Fig. 3), which results in a feature tensor of size $1 \times 48 \times H \times W$. The second sub-network contains seven downsampling layers, which map the feature tensor to the suggested trade-off parameters $\{\beta_i\}_{i=1}^3$, which further calculate the suggested weightings through relationship (9). Here, the downsampling layer has the same structure as that of the direction estimation network $\mathcal{P}_\eta$.

2.4 The Overall Deraining Framework

By combining the backbone UConNet with the two auxiliary networks, we obtain an overall deraining framework as shown in Fig. 3. Specifically, given an observed rainy image $\mathbf{O}$, we first use the learned rain direction estimation network $\mathcal{P}_\eta$ to estimate the rain direction θ. Specifically, we collect eight 128×128 patches in the rainy image $\mathbf{O}$ as independent inputs of $\mathcal{P}_\eta$, where the final rain direction θ is determined as the median number of the eight output directions. With the estimated rain direction θ, we can feed θ and $\mathbf{O}$ into the learned UConNet $\mathcal{N}_\omega$. Here, the weightings of feature

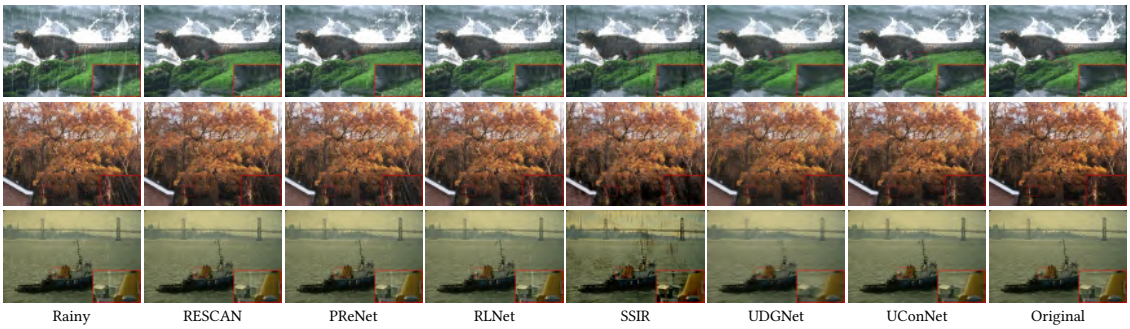

Figure 4: The deraining results by different methods on three synthetic rainy images in the R100L [51], Rain1400 [11], and Rain12 [26] datasets.

maps $\{w_i\}_{i=1}^3$ can be either manually tuned to achieve satisfactory performances or using the suggested weightings by the recommendation network $\mathcal{H}_\kappa$. In experiments, we choose the latter strategy for efficiency consideration.

Overall, the proposed UConNet can efficiently handle different rain scenarios by changing the controllable weightings, and thus achieves high generalization abilities.

3 EXPERIMENTS

3.1 Ablation Study

3.1.1 Effectiveness of Controllable Weightings. To validate the effectiveness of the controllability of weightings $\{w_i\}_{i=1}^3$ at the inference stage, we consider three variants of our UConNet: (a) The UConNet using fixed weightings $w_1 = 0.5$, $w_2 = 0.25$, and $w_3 = 0.25$. (b) The UConNet using the weightings recommendation network $\mathcal{H}_\kappa$ to provide the values of weightings, which is used in the main experiments. (c) The weightings of UConNet are searched by the greedy algorithm to obtain the best PSNR value for each rainy image. We test these three variants on the R100L dataset (The backbone UConNet is trained on the DID-MDN dataset [63]) and report the average results and inference time in Table. 1. We can observe that UConNet (c) can achieve the best PSNR value since it is based on adaptive tunning for each rainy image, which is reasonable. UConNet (a) is not that effective since the fixed weightings limit its applicability for different rainy images. UConNet (b) also obtains satisfactory results, while its efficiency is higher than UConNet (c), since the weightings recommendation network $\mathcal{H}_\kappa$ can provide the suggested weightings in less time. These results validate the effectiveness of the controllable weightings, which help the UConNet to achieve higher generalization abilities for different rainy images.

3.1.2 Effectiveness of Different Regularization Terms. To test the effectiveness of different regularization terms in the loss function (10), we respectively set each weighting w_i $(i = 1, 2, 3)$ to 0 (Note that w_1 corresponds to the smoothness of rain streaks, w_2 corresponds to the sparsity of rain streaks, and w_3 corresponds to the smoothness of the clean image) and tune other weightings to obtain the best PSNR value. The average results are shown in Table 2. We can observe that without the smooth regularization of rain (UConNet (d) with $w_1 = 0$), the performance is degraded. The same phenomenon happens when the spasity of rain is neglected (UConNet (e) with $w_2 = 0$). When the smooth regularization of the clean image is neglected (UConNet (f) with $w_3 = 0$), the PSNR and SSIM values of the deraining results are the same as that of the rainy images. This is reasonable, since without the smooth regularization of clean images, the network output (the rain map) is only being constrained to be smooth and sparse, and thus the network output is a zero matrix, making the deraining result being the same as the rainy image. These analyses show that each controllable weighting is important to improving the performance of UConNet.

Table 3: The quantitative results and average inference time by different methods on different datasets.

Dataset	Metric	Rainy	RESCAN	PReNet	RLNet	SSIR	UDGNet	Model (5)	UConNet
R100L	PSNR	25.33	29.95	27.92	29.41	26.23	26.84	27.81	**31.28**
	SSIM	0.8300	0.8972	0.8682	0.8979	0.8409	0.8482	0.8792	**0.9351**
Rain1400	PSNR	23.65	26.15	24.39	25.53	23.14	22.93	23.93	**26.78**
	SSIM	0.7561	0.8187	0.7717	0.8038	0.7169	0.7740	0.7910	**0.8491**
Rain12	PSNR	28.82	30.80	29.64	30.31	29.65	27.49	29.03	**32.39**
	SSIM	0.8355	0.8680	0.8439	0.8596	0.8650	0.8589	0.8589	**0.9142**
Parameters ($\times 10^6$)		––	0.2	0.2	5.6	0.05	1.2	––	0.9
FLOPs ($\times 10^{11}$)		––	0.76	1.6	3.1	1.8	2.7	––	1.2
Time (second)		––	0.0220	0.1521	0.2022	0.2514	0.0888	4.8735	0.1781

3.2 Comparisons with State-of-the-Arts

Baselines We consider five state-of-the-art competing image deraining methods: RESCAN [25] (ECCV 2018, supervised), PReNet [35] (CVPR 2019, supervised), RLNet [2] (CVPR 2021, supervised), SSIR [47] (CVPR 2019, semi-supervised), and UDGNet [59] (ACM MM 2021, unsupervised). Moreover, we include the unsupervised optimization model (5), which is solved by the ADMM (See supplementary materials for details of the ADMM), as another baseline.

Training data We adopt 4000 clean images in the DID-MDN dataset [63] as training data. Since the training of our rain direction estimation network needs the real rain direction, we cannot use the rainy images in DID-MDN, where the rain directions are unknown. Instead, we re-perform synthetic rain streaks on the clean images in the DID-MDN dataset, where the rain streaks are generated by the screen blend model [29] with randomly selected length, direction, width, and intensity for each image. Each clean image is used to generate two rainy images. The supervised methods RESCAN, PReNet, and RLNet are re-trained using these 8000 pairs of simulated rainy images and ground truth clean images. The semi-supervised SSIR is re-trained using these 8000 pairs of images and their own real-world rainy images [47]. The unsupervised UDGNet and our UConNet are trained with only 8000 simulated rainy images, while UDGNet needs some other clean images for the adversarial

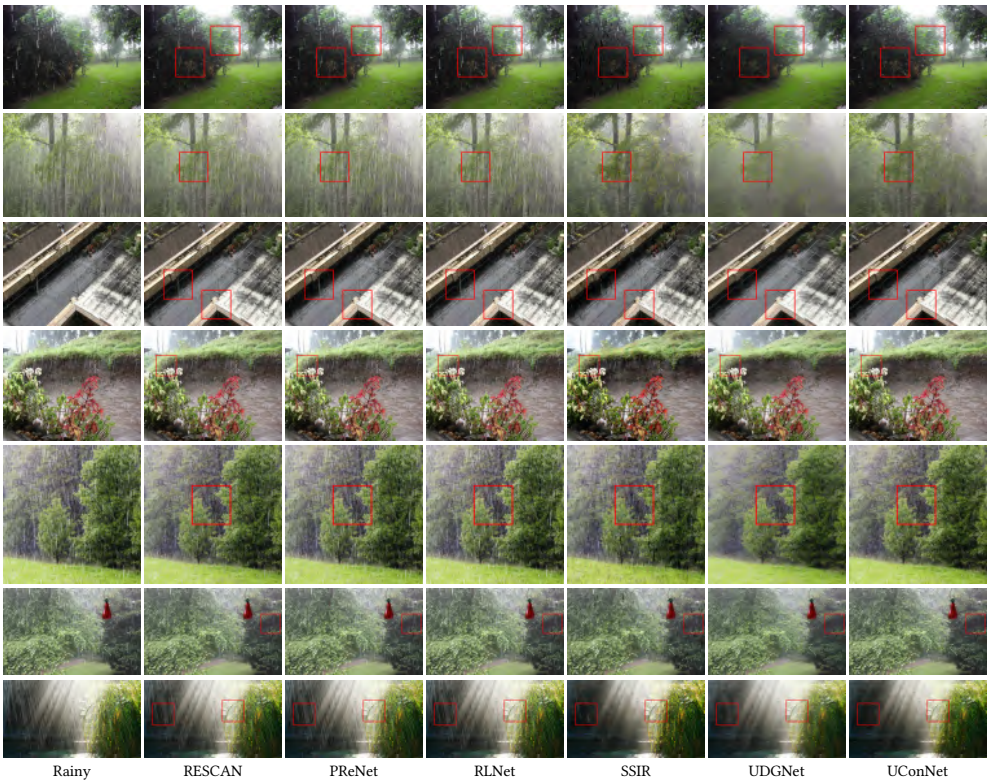

Figure 5: The deraining results by different methods on real-world rainy images. Please increase screen brightness.

loss [59]. Moreover, both UDGNet and our UConNet need the real rain directions data to train the direction estimation network.

Testing data To test the generalization abilities of our method, we use three benchmark testing datasets: R100L [51] (containing 200 pairs of rain and clean images), Rain1400 [11] (containing 1400 pairs of rain and clean images), and Rain12 [26] (containing 12 pairs of rain and clean images). We evaluate the results using peak signal-to-noise ratio (PSNR) and structure similarity (SSIM). For real-world data, we collect rainy images with large field of view from datasets [11, 26, 51] and online search. A total number of 38 real-world rainy images with different rain scenarios are included to test the generalization abilities of our method.

3.3 Experimental Results

The quantitative results on synthetic rainy images are reported in Table 3. Although our UConNet achieves better results than other supervised methods, we remark that we only adopt a relatively small scale dataset (i.e., the DID-MDN dataset with 8000 pairs of images) to train the compared supervised methods, and the training data and testing data are in different domains. These factors relatively restrict the performance of supervised methods. However, since our unsupervised method is also trained with only 8000 rainy images and still attains better performances, it is reasonable to say that our unsupervised UConNet has higher generalization abilities than compared supervised methods under a small scale training dataset. Also, our UConNet outperforms the unsupervised UDGNet, which reflects the superiority of the controllability of our method. Meanwhile, UConNet shows better performances than directly solving the model (5), which reveals that the deep network has higher representation abilities to learn natural image statistics inside training data. The inference time of UConNet is short, which makes it more applicable in real scenarios as compared with model-based methods, e.g., the model (5), which takes several seconds to process a rainy image. Also, the number of parameters and FLOPs of our method are acceptable as compared with other deep learning methods. Some qualitative results on synthetic rainy images are displayed in Fig. 4. We can observe that our UConNet can both better remove the rain streaks and preserve the image details as compared to other methods thanks to the controllable weightings of UConNet, where different rainy images are assigned with different weightings to achieve the optimal deraining effect.

The deraining results on real-world rainy images are shown in Fig. 5. From these figures, we can observe that supervised methods cannot totally remove the real-world rain streaks due to the domain shift between real-world rainy images and training data. As compared, the unsupervised UDGNet and our UConNet are more applicable in real-world scenarios. However, the UDGNet is a deterministic mapping and cannot be tunned for different rainy images. As compared, our method has controllable weightings at the inference stage, which is more flexible to handle different rain scenarios. As a result, according to the visual results in the red-boxes of Fig. 5, we can observe that our UConNet can better preserve the image details, while UDGNet suffers from over smoothness. More visual results can be found in supplementary materials.

3.4 Extension to Video Deraining

Compared with single image, videos have consistence along the temporal mode. To tackle the video deraining task, we only need to

Figure 6: The deraining results by different methods on a synthetic rainy video in the NTURain dataset (The first row) and real-world rainy videos (The last two rows).

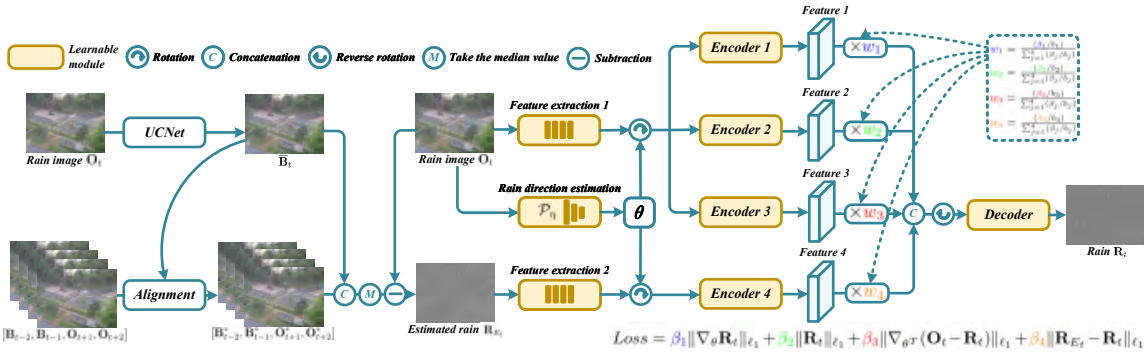

Figure 7: The flowchart of our UConNet-V for video deraining. Based on UConNet, we introduce an optical flow network [36] to align adjacent frames. The weightings $\{w_i\}_{i=1}^4$ are controllable at the inference stage to achieve high generalization abilities.

introduce an extra pre-trained optical flow network to help capture the temporal consistence [36], which extracts the optical flow of videos to align adjacent frames. The flowchart of our UConNet for video deraining (termed as UConNet-V) is illustrated in Fig. 7, where the weightings $\{w_i\}_{i=1}^4$ are also controllable at the inference stage (see supplementary materials for details of the structure and training of our method for video deraining).

We compare our video deraining method with OTMS-CSC [21], J4RNet [27], SpacCNN [5], SLDNet [53], S2VD [61], and ESTINet [64] on synthetic and real-world rainy videos in the NTURain testing dataset [5]. We use the pre-trained models of SpacCNN, S2VD, ESTINet, and J4RNet, where SpacCNN, S2VD, ESTINet, and our UConNet-V are trained on the NTURain training dataset [5]. J4RNet is trained on the RainSynLight25 and RainSynComplex25 datasets [27]. OTMS-CSC is a model-based method. The results of SLDNet [53] are obtained from its generous authors. We train UConNet-V for 10 epochs with learning rate 0.0001 and batch size 16. The results are illustrated in Table 4 and Fig. 6. Here, J4RNet and SpacCNN are implemented with Caffe, thus we are unable to calculate the number of parameters and FLOPs. From Table 4, we can observe that UConNet-V achieves competitive results compared to state-of-the-art methods. We note that although S2VD and ESTINet achieve promising results for synthetic rainy videos, they fail to remove real-world rain streaks (see Fig. 6) due to the domain shift between real-world data and training data. In contrast, our UConNet-V has better generalization abilities since it does not rely on paired training data and can also be controlled at the inference stage.

Table 4: The quantitative results and average inference time per frame by different video deraining methods. The best and second-best values are highlighted.

Dataset	Metric	Rainy	OTMS-CSC	J4RNet	SpacCNN	SLDnet	S2VD	ESTINet	UConNet-V
NTURain (a)	PSNR	28.74	31.61	28.86	31.95	34.50	**36.18**	36.10	35.09
	SSIM	0.9355	0.9523	0.9259	0.9389	0.9536	0.9630	**0.9672**	0.9661
NTURain (b)	PSNR	30.64	25.32	31.09	34.23	35.05	**38.51**	36.14	36.64
	SSIM	0.9015	0.8229	0.9427	0.9558	0.9540	**0.9736**	0.9649	0.9657
Parameters ($\times 10^6$)		––	––	––	––	4.0	0.5	29.9	1.3
FLOPs ($\times 10^{11}$)		––	––	––	––	51.0	3.8	54.2	4.5
Time (second)		––	4.02	5.94	2.91	1.49	0.05	1.41	1.18

4 CONCLUSION

In this work, we propose the UConNet for image and video deraining. Our UConNet learns a relationship between trade-off parameters of the loss function and weightings of feature maps. At the inference stage, the weightings can be adaptively controlled to handle different rain scenarios, resulting in high generalization abilities. Extensive experimental results validate the effectiveness, generalization abilities, and efficiency of UConNet. In future work, we can consider applying the controllable network to more low-level visual tasks, e.g., denoising and super-resolution.

Acknowledgement This research is supported by NSFC (No. 61876203, 12171072, 12001446), the Applied Basic Research Project of Sichuan Province (No. 2021YJ0107), the Key Project of Applied Basic Research in Sichuan Province (No. 2020YJ0216), the National Key Research and Development Program of China (No. 2020YFA0714001), and the Fundamental Research Funds for the Central Universities (No. JBK2202049, JBK2102001).

REFERENCES

[1] Yi Chang, Luxin Yan, and Sheng Zhong. 2017. Transformed Low-Rank Model for Line Pattern Noise Removal. In *ICCV*. 1735–1743.

[2] Chenghao Chen and Hao Li. 2021. Robust Representation Learning with Feedback for Single Image Deraining. In *CVPR*. 7738–7747.

[3] Dongdong Chen, Mingming He, Qingnan Fan, Jing Liao, Liheng Zhang, Dongdong Hou, Lu Yuan, and Gang Hua. 2019. Gated Context Aggregation Network for Image Dehazing and Deraining. In *WACV*. 1375–1383.

[4] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. 2021. Pre-Trained Image Processing Transformer. In *CVPR*. 12294–12305.

[5] Jie Chen, Cheen-Hau Tan, Junhui Hou, Lap-Pui Chau, and He Li. 2018. Robust Video Content Alignment and Compensation for Rain Removal in a CNN Framework. In *CVPR*. 6286–6295.

[6] Yi-Lei Chen and Chiou-Ting Hsu. 2013. A Generalized Low-Rank Appearance Model for Spatio-temporally Correlated Rain Streaks. In *ICCV*. 1968–1975.

[7] Liang-Jian Deng, T. Huang, Xi-Le Zhao, and Tai-Xiang Jiang. 2018. A directional global sparse model for single image rain removal. *Applied Mathematical Modelling* 59 (2018), 662–679.

[8] Sen Deng, Mingqiang Wei, Jun Wang, Yidan Feng, Luming Liang, Haoran Xie, Fu Lee Wang, and Meng Wang. 2020. Detail-recovery Image Deraining via Context Aggregation Networks. In *CVPR*. 14548–14557.

[9] Yingjun Du, Jun Xu, Xiantong Zhen, Ming-Ming Cheng, and Ling Shao. 2020. Conditional Variational Image Deraining. *IEEE Transactions on Image Processing* 29 (2020), 6288–6301.

[10] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. 2017. Clearing the Skies: A Deep Network Architecture for Single-Image Rain Removal. *IEEE Transactions on Image Processing* 26, 6 (2017), 2944–2956.

[11] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. 2017. Removing Rain from Single Images via a Deep Detail Network. In *CVPR*. 1715–1723.

[12] Xueyang Fu, Borong Liang, Yue Huang, Xinghao Ding, and John Paisley. 2020. Lightweight Pyramid Networks for Image Deraining. *IEEE Transactions on Neural Networks and Learning Systems* 31, 6 (2020), 1794–1807.

[13] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. 2019. Depth-Attentional Features for Single-Image Rain Removal. In *CVPR*. 8014–8023.

[14] Huaibo Huang, Aijing Yu, and Ran He. 2021. Memory Oriented Transfer Learning for Semi-Supervised Image Deraining. In *CVPR*. 7728–7737.

[15] Kui Jiang, Zhongyuan Wang, Peng Yi, Chen Chen, Baojin Huang, Yimin Luo, Jiayi Ma, and Junjun Jiang. 2020. Multi-Scale Progressive Fusion Network for Single Image Deraining. In *CVPR*. 8343–8352.

[16] Kui Jiang, Zhongyuan Wang, Peng Yi, Chen Chen, Guangcheng Wang, Zhen Han, Junjun Jiang, and Zixiang Xiong. 2021. Multi-Scale Hybrid Fusion Network for Single Image Deraining. *IEEE Transactions on Neural Networks and Learning Systems* (2021). https://doi.org/10.1109/TNNLS.2021.3112235

[17] Tai-Xiang Jiang, Ting-Zhu Huang, Xi-Le Zhao, Liang-Jian Deng, and Yao Wang. 2019. FastDeRain: A Novel Video Rain Streak Removal Method Using Directional Gradient Priors. *IEEE Transactions on Image Processing* 28, 4 (2019), 2089–2102.

[18] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu. 2012. Automatic Single-Image-Based Rain Streaks Removal via Image Decomposition. *IEEE Transactions on Image Processing* 21, 4 (2012), 1742–1755.

[19] Jin-Hwan Kim, Chul Lee, Jae-Young Sim, and Chang-Su Kim. 2013. Single-image deraining using an adaptive nonlocal means filter. In *ICIP*. 914–917.

[20] Jin-Hwan Kim, Jae-Young Sim, and Chang-Su Kim. 2015. Video Deraining and Desnowing Using Temporal Correlation and Low-Rank Matrix Completion. *IEEE Transactions on Image Processing* 24, 9 (2015), 2658–2670.

[21] Minghan Li, Xiangyong Cao, Qian Zhao, Lei Zhang, and Deyu Meng. 2021. Online Rain/Snow Removal From Surveillance Videos. *IEEE Transactions on Image Processing* 30 (2021), 2029–2044.

[22] Minghan Li, Qi Xie, Qian Zhao, Wei Wei, Shuhang Gu, Jing Tao, and Deyu Meng. 2018. Video Rain Streak Removal by Multiscale Convolutional Sparse Coding. In *CVPR*. 6644–6653.

[23] Ruoteng Li, Loong-Fah Cheong, and Robby T. Tan. 2019. Heavy Rain Image Restoration: Integrating Physics Model and Conditional Adversarial Learning. In *CVPR*. 1633–1642.

[24] Siyuan Li, Iago Breno Araujo, Wenqi Ren, Zhangyang Wang, Eric K. Tokuda, Roberto Hirata Junior, Roberto Cesar-Junior, Jiawan Zhang, Xiaojie Guo, and Xiaochun Cao. 2019. Single Image Deraining: A Comprehensive Benchmark Analysis. In *CVPR*. 3833–3842.

[25] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. 2018. Recurrent Squeeze-and-Excitation Context Aggregation Net for Single Image Deraining. In *ECCV*. 262–277.

[26] Yu Li, Robby T. Tan, Xiaojie Guo, Jiangbo Lu, and Michael S. Brown. 2016. Rain Streak Removal Using Layer Priors. In *CVPR*. 2736–2744.

[27] Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming Guo. 2018. Erase or Fill? Deep Joint Recurrent Rain Removal and Reconstruction in Videos. In *CVPR*. 3233–3242.

[28] Risheng Liu, Zhiying Jiang, Long Ma, Xin Fan, Haojie Li, and Zhongxuan Luo. 2018. Deep Layer Prior Optimization for Single Image Rain Streaks Removal. In *ICASSP*. 1408–1412.

[29] Yu Luo, Yong Xu, and Hui Ji. 2015. Removing Rain from a Single Image via Discriminative Sparse Coding. In *ICCV*. 3397–3405.

[30] Siqi Ni, Xueyun Cao, Tao Yue, and Xuemei Hu. 2021. Controlling the Rain: from Removal to Rendering. In *CVPR*. 6324–6333.

[31] Bo Pang, Deming Zhai, Junjun Jiang, and Xianming Liu. 2020. Single Image Deraining via Scale-Space Invariant Attention Neural Network. In *ACM MM*. 375–383.

[32] Rui Qian, Robby T. Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. 2018. Attentive Generative Adversarial Network for Raindrop Removal from A Single Image. In *CVPR*. 2482–2491.

[33] Ruijie Quan, Xin Yu, Yuanzhi Liang, and Yi Yang. 2021. Removing Raindrops and Rain Streaks in One Go. In *CVPR*. 9143–9152.

[34] Dongwei Ren, Wei Shang, Pengfei Zhu, Qinghua Hu, Deyu Meng, and Wangmeng Zuo. 2020. Single Image Deraining Using Bilateral Recurrent Network. *IEEE Transactions on Image Processing* 29 (2020), 6852–6863.

[35] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. 2019. Progressive Image Deraining Networks: A Better and Simpler Baseline. In *CVPR*. 3932–3941.

[36] Zachary Teed and Jia Deng. 2020. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *ECCV*. 402–419.

[37] Cong Wang, Yutong Wu, Zhixun Su, and Junyang Chen. 2020. Joint Self-Attention and Scale-Aggregation for Self-Calibrated Deraining Network. In *ACM MM*. 2517–2525.

[38] Cong Wang, Xiaoying Xing, Yutong Wu, Zhixun Su, and Junyang Chen. 2020. DCSFN: Deep Cross-Scale Fusion Network for Single Image Rain Removal. In *ACM MM*. 1643–1651.

[39] Guoqing Wang, Changming Sun, and Arcot Sowmya. 2019. ERL-Net: Entangled Representation Learning for Single Image De-Raining. In *ICCV*. 5643–5651.

[40] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. 2020. A Model-Driven Deep Neural Network for Single Image Rain Removal. In *CVPR*. 3100–3109.

[41] Hong Wang, Zongsheng Yue, Qi Xie, Qian Zhao, Yefeng Zheng, and Deyu Meng. 2021. From Rain Generation to Rain Removal. In *CVPR*. 14791–14801.

[42] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson W.H. Lau. 2019. Spatial Attentive Single-Image Deraining With a High Quality Real Rain Dataset. In *CVPR*. 12262–12271.

[43] Yinglong Wang, Dong Gong, Jie Yang, Qinfeng Shi, Anton van denHengel, Dehua Xie, and Bing Zeng. 2021. Deep Single Image Deraining via Modeling Haze-Like Effect. *IEEE Transactions on Multimedia* 23 (2021), 2481–2492.

[44] Yinglong Wang, Yibing Song, Chao Ma, and Bing Zeng. 2020. Rethinking Image Deraining via Rain Streaks and Vapors. In *ECCV*. 367–382.

[45] Yetao Wang, Xile Zhao, Taixiang Jiang, Liangjian Deng, Yi Chang, and Tingzhu Huang. 2021. Rain Streaks Removal for Single Image via Kernel-Guided Convolutional Neural Network. *IEEE Transactions on Neural Networks and Learning Systems* 32, 8 (2021), 3664–3676.

[46] Zheng Wang, Jianwu Li, and Ge Song. 2019. DTDN: Dual-Task De-Raining Network. In *ACM MM*. 1833–1841.

[47] Wei Wei, Deyu Meng, Qian Zhao, Zongben Xu, and Ying Wu. 2019. Semi-Supervised Transfer Learning for Image Rain Removal. In *CVPR*. 3872–3881.

[48] Yanyan Wei, Zhao Zhang, Yang Wang, Mingliang Xu, Yi Yang, Shuicheng Yan, and Meng Wang. 2021. DerainCycleGAN: Rain Attentive CycleGAN for Single Image Deraining and Rainmaking. *IEEE Transactions on Image Processing* 30 (2021), 4788–4801.

[49] Wenhan Yang, Jiaying Liu, Shuai Yang, and Zongming Guo. 2019. Scale-Free Single Image Deraining Via Visibility-Enhanced Recurrent Wavelet Learning. *IEEE Transactions on Image Processing* 28, 6 (2019), 2948–2961.

[50] Wenhan Yang, Robby T. Tan, Jiashi Feng, Zongming Guo, Shuicheng Yan, and Jiaying Liu. 2020. Joint Rain Detection and Removal from a Single Image with Contextualized Deep Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 6 (2020), 1377–1393.

[51] Wenhan Yang, Robby T. Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. 2017. Deep Joint Rain Detection and Removal from a Single Image. In *CVPR*. 1685–1694.

[52] Wenhan Yang, Robby T. Tan, Shiqi Wang, Yuming Fang, and Jiaying Liu. 2021. Single Image Deraining: From Model-Based to Data-Driven and Beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 11 (2021), 4059–4077.

[53] Wenhan Yang, Robby T. Tan, Shiqi Wang, and Jiaying Liu. 2020. Self-Learning Video Rain Streak Removal: When Cyclic Consistency Meets Temporal Correspondence. In *CVPR*. 1717–1726.

[54] Youzhao Yang and Hong Lu. 2019. Single Image Deraining via Recurrent Hierarchy Enhancement Network. In *ACM MM*. 1814–1822.

[55] Rajeev Yasarla and Vishal M. Patel. 2019. Uncertainty Guided Multi-Scale Residual Learning-Using a Cycle Spinning CNN for Single Image De-Raining. In *CVPR*. 8397–8406.

[56] Rajeev Yasarla, Vishwanath A. Sindagi, and Vishal M. Patel. 2020. Syn2Real Transfer Learning for Image Deraining Using Gaussian Processes. In *CVPR*. 2723–2733.
[57] Rajeev Yasarla, Vishwanath A. Sindagi, and Vishal M. Patel. 2021. Semi-Supervised Image Deraining Using Gaussian Processes. *IEEE Transactions on Image Processing* 30 (2021), 6570–6582.
[58] Yuntong Ye, Yi Chang, Hanyu Zhou, and Luxin Yan. 2021. Closing the Loop: Joint Rain Generation and Removal via Disentangled Image Translation. In *CVPR*. 2053–2062.
[59] Changfeng Yu, Yi Chang, Yi Li, Xile Zhao, and Luxin Yan. 2021. Unsupervised Image Deraining: Optimization Model Driven Deep CNN. In *ACM MM*. 2634–2642.
[60] Weijiang Yu, Zhe Huang, Wayne Zhang, Litong Feng, and Nong Xiao. 2019. Gradual Network for Single Image De-Raining. In *ACM MM*. 1795–1804.
[61] Zongsheng Yue, Jianwen Xie, Qian Zhao, and Deyu Meng. 2021. Semi-Supervised Video Deraining with Dynamical Rain Generator. In *CVPR*. 642–652.
[62] He Zhang and Vishal M. Patel. 2017. Convolutional Sparse and Low-Rank Coding-Based Rain Streak Removal. In *WACV*. 1259–1267.
[63] He Zhang and Vishal M. Patel. 2018. Density-Aware Single Image De-raining Using a Multi-stream Dense Network. In *CVPR*. 695–704.
[64] Kaihao Zhang, Dongxu Li, Wenhan Luo, Wenqi Ren, and Wei Liu. 2022. Enhanced Spatio-Temporal Interaction Learning for Video Deraining: A Faster and Better Framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022). doi=10.1109/TPAMI.2022.3148707.
[65] Kaihao Zhang, Wenhan Luo, Wenqi Ren, Jingwen Wang, Fang Zhao, Lin Ma, and Hongdong Li. 2020. Beyond Monocular Deraining: Stereo Image Deraining via Semantic Understanding. In *ECCV*. Springer, 71–89.
[66] Zheyu Zhang, Yurui Zhu, Xueyang Fu, Zhiwei Xiong, Zheng-Jun Zha, and Feng Wu. 2021. Multifocal Attention-Based Cross-Scale Network for Image De-Raining. In *ACM MM*. 3673–3681.
[67] Yupei Zheng, Xin Yu, Miaomiao Liu, and Shunli Zhang. 2022. Single-Image Deraining via Recurrent Residual Multiscale Networks. *IEEE Transactions on Neural Networks and Learning Systems* 33, 3 (2022), 1310–1323.
[68] Lei Zhu, Zijun Deng, Xiaowei Hu, Haoran Xie, Xuemiao Xu, Jing Qin, and Pheng-Ann Heng. 2021. Learning Gated Non-Local Residual for Single-Image Rain Streak Removal. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 6 (2021), 2147–2159.
[69] Lei Zhu, Chi-Wing Fu, Dani Lischinski, and Pheng-Ann Heng. 2017. Joint Bi-layer Optimization for Single-Image Rain Streak Removal. In *ICCV*. 2545–2553.