Unpaired Learning for Deep Image Deraining with Rain Direction Regularizer

Yang Liu [1] Ziyu Yue [1] Jinshan Pan[2] Zhixun Su[1]
[1]Dalian University of Technology [2]Nanjing University of Science and Technology
https://lewisyangliu.github.io/projects/UDRDR

Abstract

We present a simple yet effective unpaired learning based image rain removal method from an unpaired set of synthetic images and real rainy images by exploring the properties of rain maps. The proposed algorithm mainly consists of a semi-supervised learning part and a knowledge distillation part. The semi-supervised part estimates the rain map and reconstructs the derained image based on the well-established layer separation principle. To facilitate rain removal, we develop a rain direction regularizer to constrain the rain estimation network in the semi-supervised learning part. With the estimated rain maps from the semi-supervised learning part, we first synthesize a new paired set by adding to rain-free images based on the superimposition model. The real rainy images and the derained results constitute another paired set. Then we develop an effective knowledge distillation method to explore such two paired sets so that the deraining model in the semi-supervised learning part is distilled. We propose two new rainy datasets, named RainDirection and Real3000, to validate the effectiveness of the proposed method. Both quantitative and qualitative experimental results demonstrate that the proposed method achieves favorable results against state-of-the-art methods in benchmark datasets and real-world images.

1. Introduction

Single image deraining aims to estimate a rain-free image from a rainy image. It is a classical image processing problem, which has been an active research topic in the vision and graphics communities within the last decade. As numerous real-world tasks (e.g., traffic detection and environmental monitoring) require high-quality images, and the rainy environment usually leads to deprecated images, it is of great interest to develop an effective algorithm to recover rain-free images.

Rain is a complex atmospheric phenomenon. Images with rain may lose color fidelity and suffer visual occlusions as a result of rain streaks (drops) move through the background scene. Mathematically, the raining process can be formulated by the following linear superimposition model:

$$O(x) = B(x) + R(x), \tag{1}$$

where O is the rainy image, B is the rain-free image, and R is the rain map, which describes the distribution and motion of the rain. x denotes the pixel coordinate. As only O is available, we need to recover B and R simultaneously. This problem is highly ill-posed because many different pairs of B and R give rise to the same O, e.g., a nearly all-zeros matrix for R and a rainy image for B, where the networks do not succeed in estimating the rain map due to the complicated distribution and motion of the rain.

To make image deraining problem well-posed, existing methods usually make some assumptions on the rain map and rain-free image. For example, the low rank representation model [1], the sparse representation model [14], Gaussian mixture model [11] have been proposed to model the rain maps or rain-free images to help rain removal. Jiang et al. [7] explore gradient operators based on the sparsity prior for rain maps. Instead of making some assumptions on the rain map and rain-free image, several methods focus on developing end-to-end trainable networks for image deraining [2, 3, 30, 22, 19, 10]. To better explore the domain knowledge, several deep learning-based methods impose constraints on the rain map or/and derained images in different perspectives, e.g., joint rain location detection in [26], rain density label assigned in [29], rain kernels describing the repetitive local patterns of rain streaks in [21]. These methods usually need paired data to train deep neural networks. However, the pixel-wise content loss used in these methods tends to smooth details, while GAN based losses may generate fake structures. In [19, 13], Negative SSIM loss is used to model the local structure similarity between the derained image and ground truth one. However, the approach of only imposing constraints on the derained images neglects the distribution and motion of rain streaks. Thus, it is necessary to explore the properties of the rain map for better rain removal.

We find that the rain map of the natural rainy images usually has a linear shape with similar directions in a local image patch. The rain map is usually of high contrast with sharp edges as the pixels covered by rain present high intensity while the pixels without rain is close to zeros. Therefore we propose a rain direction regularizer to model the properties of the rain map for the rain map estimation so that consistent and sharp rain structures can be preserved. The proposed rain direction regularizer extracts gradient, which corresponding parallel and perpendicular directions of the rain maps to truthfully preserve the local consistency and sharpness of structures of rain maps.

To further improve the generalization ability for handling real rainy images, several methods [23, 28, 24, 13] reveal the gap existed between the synthetic and real rainy images due to the complicated situations of real rain influenced by the distribution and motion of the rain. Wang et al. [22] construct a paired real rain dataset by fusing rainy image sequences. However, we observe that structure and detail disalignment are bring into this dataset. Lin et al. [13] synthesize real rain maps with filtered images to distill a rain map estimation network, yet the derained results are inferior to contemporaneous state-of-the-art methods according to their released experimental results.

To handle real images, we develop a semi-supervised learning method with the knowledge distillation strategy. The semi-supervised learning estimates rain maps and derained images constrained by the rain direction regularizer. With the estimated rain maps, we can synthesize a new paired set by applying the estimated rain maps to reference clear images based on the raining model (1). Then, the knowledge distillation part distills the derained model in the semi-supervised learning part to facilitate rain removal tasks. In addition, considering the unavailability of the rain direction label of existing datasets, we construct a new dataset RealDirection, of which each rainy image is assigned with a direction label. A Real3000 dataset is also proposed as the existing publicly available real rainy dataset is small-scale. We conduct comprehensive experiments and show that the proposed method achieves promising results for rain removal tasks.

The main contributions of this work are as follows:

- We propose a simple and effective rain direction regularizer to preserve the structures and sharpness of the rain maps based on the local properties of rain maps.
- We develop an unpaired learning method for rain removal by incorporating semi-supervised learning and knowledge distillation strategy. Given an unpaired set of synthetic and real rainy images. The semi-supervised learning part estimates rain maps and derained images. The derained results are further improved by the knowledge distillation part.
- We propose the RainDirection dataset and Real3000 dataset to train the proposed method. Both quantitative and qualitative evaluations on the benchmark datasets and real-world images demonstrate the effectiveness of the proposed method.

2. Related Work

We have witnessed significant advances in single image deraining in recent years. A comprehensive review is beyond the scope of this work. We present a brief review of the most related ones within proper contexts in this section.

Single image deraining is a classical ill-posed problem. Various methods make some assumptions and regularizations on the separated rain map and rain-free image. Chen and Hsu [1] reveal that a rainy scene usually contains similar patterns of rain streaks in different local patches. Then they propose a low-rank model to characterize the appearance of rain streaks. Luo et al. [14] introduce a sparsity-based regularization strategy to help rain removal, which assumes that both the rain map and rain-free image could be sparsely modeled. Li et al. [11] show that the rainy image patch mainly contains the annoying effect of rain streaks. Then they learn a rain streak layer prior for layer separation based on GMM. One of the limitations is that they tend to over-smooth the resulting images. Different from the gradient operators used in Fastderain [7] which is based on the hand-crafted sparsity prior of rain maps and need to detect the direction of the rain streaks to shift the image. We synthesize a large dataset with accurate direction labels. Our proposed rain direction regularizer is differentiable and can be naturally plugged into the networks to train with modern deep learning techniques jointly.

Recently deep learning-based methods [2, 3, 26, 29, 17, 18, 21, 6] have made remarkable progress in image deraining problem. Numerous regularizers and prior knowledge have been applied to the network to better estimate the rain map and derained image. In [26], the rain location information is considered to help estimate rain streaks. In [29], each rainy image is assigned with a density label, which is used to supervise the rain residual during training. Wang et al. [21] introduce rain kernels which describe the repetitive local patterns of rain streaks. As per-pixel luminance value based losses tend to smooth the edges, Negative SSIM loss is adopted in [19, 13] on the derained images to improve structure similarity. We note that existing methods rarely explicitly explore the rain direction information in the task of rain map estimation, which actually encodes important structures and motion information of rain streaks.

The deep learning based methods discussed above mainly train with paired synthetic data in a fully-supervised manner. Several recent semi-supervised learning based methods [23, 28, 24] have been proposed to process real rainy images. The basic idea behind the semi-supervised learning strategy is to train the network with synthetic data in a supervised manner and real data in an unsupervised manner, and the architecture and parameters are shared in both stages. The supervised parts of these methods are similar. The networks directly learn the mapping from the synthetic rainy image to the corresponding ground truth rain-free image with per-pixel luminance value based losses. As for the unsupervised part, [23] introduces a GMM likelihood term with a K-L Regularizer, while [28] using Gaussian processes, and GAN is used in [24]. [22] proposes a paired real rain dataset by fusing multi-frame rainy images and develops an attention-based supervised network to tackle this problem. Different from these approaches, we introduce a rain direction regularizer to help rain map estimation with an unpaired distillation strategy to further boost the performance for rain removal. We note that our approach is different from a recently proposed method [13] for synthesizing real rain in model formulation, solving process, and learning goals. Our method achieves favorable results against these methods as shown in Sec. 4.

3. Proposed Algorithm

Our proposed algorithm for image deraining contains the semi-supervised learning and knowledge distillation part, which mainly involves three stages. Figure 1 shows an overview of the algorithm. First, we propose a rain direction regularizer based on the properties of the rain streaks to constraint the deep neural network on estimating rain maps. Second, the reconstruction network restores derained images using the estimated rain maps. The networks for the synthetic and real input in the first and second stage are trained in a semi-supervised learning manner. Third, with the estimated rain maps, we synthesize a new paired data for the real images and develop a knowledge distillation method to explore these two paired sets such that the deraining model in the semi-supervised learning part is distilled. All three stages are trained jointly with an unpaired set of synthetic images and real rainy images. We present the details of each component of our algorithm in the following.

3.1. Rain map estimation with direction regularizer

The first stage of our proposed algorithm is to estimate the rain maps from the unpaired input rainy images $\{x_{syn}; x_{real}\}$, where x_{syn} and x_{real} denotes a synthetic rainy image and a real rainy image picked randomly from two independent datasets, respectively.

As pointed in [26, 3, 17], a rain map usually encodes high-frequency details (residual) and location information of rain streaks (drops) within the rainy image, which could be decoded via a series of parameterized mapping functions, e.g., deep CNNs. Considering the good capability in modeling residual information and approximating mapping functions, we use an effective residual network [5, 12] to estimate a rain map from a rainy image directly. Formally, the process to estimate the rain maps could be formulated as:

$$z_{syn} = \mathcal{F}_1(x_{syn}), \tag{2}$$

$$z_{real} = \mathcal{F}_1(x_{real}), \tag{3}$$

where $\mathcal{F}_1$ is a deep residual network taking x_{syn} and x_{real} as the inputs; z_{syn} and z_{real} are the network outputs which are the estimated rain maps of x_{syn} and x_{real}, respectively.

To regularize the network $\mathcal{F}_1$, a commonly used method is to ensure that the estimated rain map z_{syn} is close to the ground truth rain map z_{gt} by:

$$\mathcal{L}_{content} = f(z_{syn} - z_{gt}), \tag{4}$$

where f is usually taken as L_1-norm. However, this commonly used content loss (4) does not consider the local structures of the rain streaks and tends to generate over-smoothed results as demonstrated in [9, 15].

We note that the rain map of the natural rainy images often observed linear shapes with similar directions in a local image patch. Moreover, the rain map is usually of high contrast with sharp edges as the pixels covered by rain present high intensity while the pixels without rain are close to zeros. Hence, we propose a rain direction regularizer to help preserve the structures and shape of rain maps. The proposed regularizer is defined as:

$$\mathcal{L}_{direction} = \sum_{k_d \in \{k_{\parallel}, k_{\perp}\}} f(k_d \otimes z_{syn} - k_d \otimes z_{gt}) \tag{5}$$

k_d are directional filter kernels used to extract the local gradients by a convolutional operation. Our proposed RainDirection dataset contains rainy images with corresponding rain direction labels. During training, two predefined kernels correspond to the parallel (denoted as $k_{\parallel}$) and perpendicular (denoted as $k_{\perp}$) direction of the rain streaks in rain map are used to calculate (5). f is taken as L_1-norm.

The proposed rain direction regularizer can model the neighboring information of rain maps so that the local properties of the rain map could be preserved more truthfully. We predefine 18 $l \times l$ (l = 5) sized directional filter kernels corresponding to the angles $\{0°, 10°, ..., 170°\}$ which improves the robustness to rain direction with small variation. The filter kernels k_d are differentiable and can be naturally plugged into the networks to train with modern deep learning techniques jointly. We note that the vertical and horizontal image gradient operators based on the first-order difference method are a special case of our rain direction regularizer. Although the image gradient operators also help preserve edge information, rain motion information, i.e., rain direction, has not been considered. Figure 2 presents two simple cases to illustrate our motivation. Thanks to the rain direction regularizer, the resulting rain maps (b, e) are closer to the ground truth (c, f) against (a, d). (b, e) preserve truthful and sharp structures of rain streaks.

3.2. Derained image reconstruction

With the learned rain maps from the first stage, the reconstruction network in the second stage could restore the derained images according to the rain model (1). We implement this process as:

$$y_{syn} = \mathcal{F}_2(x_{syn} - z_{syn}), \tag{6}$$

$$y_{real} = \mathcal{F}_2(x_{real} - z_{real}), \tag{7}$$

where the deep residual network $\mathcal{F}_2$ taking $x_{syn} - z_{syn}$ and $x_{real} - z_{real}$ as the inputs; y_{syn} and y_{real} are the network outputs which are the estimated derained images of x_{syn} and x_{real}, respectively.

The reconstruction loss is used to regularize the network $\mathcal{F}_2$ with the supervision of clean image y_{gt}:

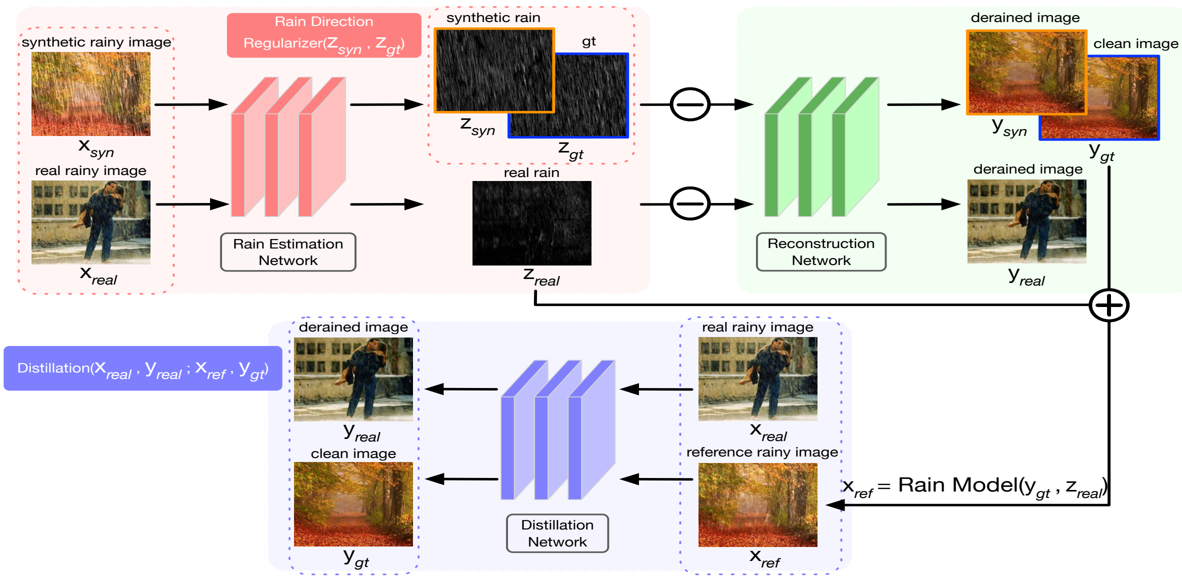


Figure 1. An overview of the proposed method. Given an unpaired input rainy images $\{x_{syn}; x_{real}\}$, the first stage is to estimate the rain maps constrained by the rain direction regularizer. Then the second stage reconstruct the derained images with $\{z_{syn}; z_{real}\}$ according to the inverse process of (1). The estimated rain map z_{real} is used to generate a new rainy image x_{ref} by Eq. (9). Such two paired sets $\{x_{real}, y_{real}\}$ and $\{x_{ref}, y_{gt}\}$ are used to distill the third stage to facilitate the real rain removal tasks.

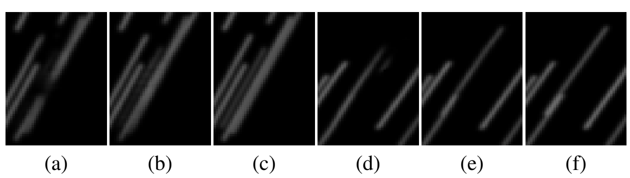


Figure 2. Simple cases to illustrate our motivation. (a, d), (b, e) and (c, f) denote the resulting rain maps estimated with $\mathcal{L}_{content}$ only, $\mathcal{L}_{content} + \lambda\mathcal{L}_{direction}$, and the ground truth, respectively. λ is a weight parameter. The rain direction regularizer $\mathcal{L}_{direction}$ helps preserve structure consistency and sharpness of rain streaks.

$$\mathcal{L}_{recons} = f(y_{syn} - y_{gt}), \tag{8}$$

where f is taken as L_1-norm in our experiments.

3.3. Knowledge distillation for rain removal

We use the results from the rain estimation and reconstruction networks (i.e., the semi-supervised part) to distill the network of the third stage. The first two stages serve as the teacher part while the third stage is the student part. With the help of rain direction regularizer, the teacher part can learn a decent derained image y_{syn} from x_{syn} supervised by y_{gt}, and z_{real} as well as y_{real} from x_{real} via a semi-supervised manner. Given the reference clean image y_{gt}, we use the estimated rain map z_{real} to generate a new rainy image by:

$$x_{ref} = y_{gt} + z_{real}. \tag{9}$$

Thus, we obtain a new image pair $\{x_{ref}, y_{gt}\}$. And our knowledge distillation is achieved by:

$$\mathcal{L}_{distill} = f(\mathcal{F}_3(x_{real}) - y_{real}) + \phi g(\mathcal{F}_3(x_{ref}) - y_{gt}), \tag{10}$$

where f and g are both taken as $L1$-norm; $\mathcal{F}_3$ denotes a deep residual network; ϕ is a positive weight parameter balancing the importance of each part. Note that x_{real} and y_{real} are the input and output image of the teacher part.

The knowledge distillation is used to explore the useful information in $\{x_{ref}, y_{gt}\}$ taught by $\{x_{real}, y_{real}\}$ so that the rain removal process for real-world cases can be further improved. Section 5.2 will illustrate its effectiveness.

3.4. Joint training with synthetic and real images

Our method adopts both synthetic and real rainy images to alternatingly update the rain estimation network $\mathcal{F}_1$, reconstruction network $\mathcal{F}_2$ and the distillation network $\mathcal{F}_3$ during training. Each stage takes the outputs from previous stages as the inputs. Given the unpaired input rainy images $\{x_{syn}; x_{real}\}$, $\mathcal{F}_1$ and $\mathcal{F}_2$ are firstly trained with the synthetic image x_{syn} which are regularized by the losses defined in (4) and (5), and in (8), respectively. As the ground truth rain map and clean image of the real rainy image x_{real} are unavailable. Similar to [23, 4], we use the TV regularization term L_{TV} on the derained image y_{real} to constrain the smoothness of the background scene. The cycle-consistent regularizer L_{cycle} [32] is also adopted so that the result of derained image y_{real} added back to the estimated rain map z_{real} should close to the input rainy image x_{real}. After obtaining the unpaired image pair $\{x_{real}, y_{real}\}$ and $\{x_{ref}, y_{gt}\}$, $\mathcal{F}_3$ is trained in a fully-supervised manner by the distillion loss defined in (10) where $\phi = 10$. The overall objective of our network is $L_{content} + 1.2L_{direction} + L_{recons} + 0.001L_{TV} + 0.01L_{cycle} + L_{distill}$. All the weighting parameters are set by sensitivity analysis, e.g., our method in Figure 8 achieves the highest PSNR value when $\lambda = 1.2$ for the $L_{direction}$.

We design the semi-supervised part to distill the third stage and only the knowledge distillation part is used for test. We note that the semi-supervised part could be replaced with other pretrained state-of-the-art image deraining models with the rain direction regularizer to further improve rain map estimation. As for the datasets to train the knowledge distillation part, only an unpaired set of real-world rainy and clean images is needed, i.e., x_{real} and y_{gt} are unpaired. And the unpaired image pair $\{x_{real}, y_{real}\}$ and $\{x_{ref}, y_{gt}\}$ can be synthesized during training. Such strategy effectively facilitates real rain removal tasks.

4. Experimental Results

We compare our method against state-of-the-art image deraining methods on the publicly available benchmark datasets and our proposed datasets. Due to the comprehensive experiments performed, we only show a small portion of the results in the main paper. Please visit the homapege for more extensive results where we release our source code, trained models and datasets to the public.

4.1. Datasets

The datasets for training and test are twofold. Rain100H [26], Rain100L [26], DID-Data [29] and our proposed RainDirection contain synthetic rainy images with ground truth rain-free images, while SIRR-Data [23], Real200 [24] and our proposed Real3000 contain real rainy images without rain-free images. Rain100H and Rain100L dataset is proposed in [26] covering different rain streak distribution. DID-Data dataset[29] containing rain-density label information, e.g., light, medium, and heavy rain conditions.

We propose a new dataset named RainDirection dataset to validate the effectiveness of the rain direction regularizer. Even though existing large-scale datasets could reflect rainy weather conditions well, they lack the corresponding rain direction label information. The rainy images in RainDirection are obtained by adding clean images from Flick2K and DIV2K dataset [20] with synthetic labeled rain maps according to the rain model (1). Each rainy image is assigned with a direction label. These direction labels are used to calculate (5) during training. The training and test set of RainDirection contains 2920 and 430 images, respectively.

We also propose another new dataset named Real3000 dataset. Although several real datasets have been proposed in previous works, the quantity is small, e.g., SIRR-Data [23] (147 images) and Real200 [24] (200 images). It is hard to say that such small-scale datasets are sufficient to train networks with millions of trainable parameters. The large-scale Real3000 dataset contains 3000 real rainy images without ground truth images which are collected mainly from the internet and captured by a Canon EOS 6D camera. The training and test set contains 2700 and 300 diverse natural outdoor images, respectively. Real3000 dataset covers various rain conditions, e.g., sparse and dense rain with different directions and shapes.

4.2. Implementation details

The rain map estimation network $\mathcal{F}_1$, the reconstruction network $\mathcal{F}_2$ and the distillation network $\mathcal{F}_3$ could be any existing deep CNNs. We adopt a small-scale EDSR [12] network in our method and share the same architecture for $\mathcal{F}_1$, $\mathcal{F}_2$ and $\mathcal{F}_3$, of which the number of residual blocks is 16, and the number of filters is 64, 64 and 128, respectively.

In the learning process, we use the ADAM optimizer [8] with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-4}$. The minibatch size is set to be 12. The learning rate is initialized as 8×10^{-4}, which is halved every 40 epochs. The parameters are initialized randomly, and the network converges well after 200 epochs. We empirically set the weight parameter of three stages to be 1.0. The entire network is trained using the Pytorch framework.

During training, the synthetic and real images are randomly cropped in an unpaired manner. For an RGB image of size $h \times w \times c$, the size of the rainy image O, the rain map R, and the rain-free image B remain to be $h \times w \times c$. The mathematical operations and CNNs used in our method do not change the dimensions during training and test.

4.3. Comparisons with the state-of-the-arts

To evaluate the performance of the proposed algorithm, we compared it against state-of-the-art algorithms including GMM [11], Clear [2], DDN [3], JORDER [26] (JORDER-E [25]), DID-MDN [29], UMRL [27], PReNet [19], SIRR [23], O'er [13], Syn2Real [28], RCDNet [21]. We adopt 4 measurement metrics including PSNR, SSIM, Learned Perceptual Image Patch Similarity (LPIPS) [31], and Natural Image Quality Evaluator (NIQE) [16]. Higher PSNR and SSIM values indicate higher image quality, while lower LPIPS and NIQE values indicate higher perceptual quality. We strictly employ the same settings for all the evaluated methods on the Y channel in YCbCr space similar to previous works, except that three channels of RGB space are adopted for the DID-Data dataset. The proposed method can achieve promising results against the state-of-the-art methods in benchmark datasets and real images.

Benchmark datasets with ground truth. Table 1 shows the quantitative evaluations on different benchmark datasets, where the results of the state-of-the-art methods are obtained using the corresponding publicly available codes and models for fair comparisons. The proposed method achieves favorable derained images as well as rain maps against state-of-the-art methods in terms of PSNR, SSIM and LPIPS. All of the compared methods are strictly evaluated with the same settings. We also find some results of the methods retested by us in Table 1 are better than their published papers. For example, the PSNR/SSIM on DID-Data dataset of UMRL [27] tested by us is 30.54/0.93, while it is 29.77/0.92 in their paper.

Figure 3 and Figure 4 show some derained results from the Rain100H and Rain100L datasets by the evaluated

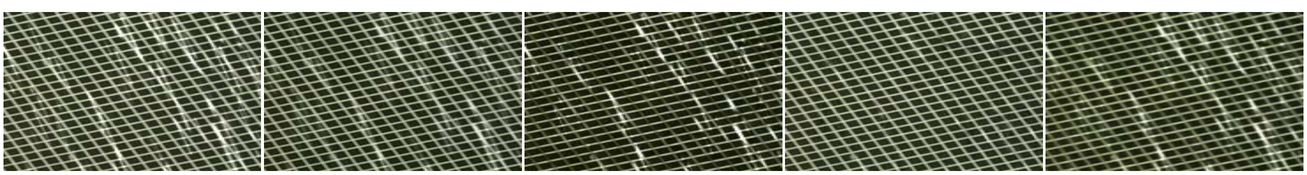
Table 1. Quantitative evaluations for the state-of-the-art deraining methods on the benchmark datasets Rain100H [26], Rain100L [26], DID-Data [29] and the proposed RainDirection dataset in terms of PSNR, SSIM and LPIPS. Our method performs favorably against the state-of-the-art image deraining methods. The bold values indicate the best performance. (The results of JORDER [26] on RainDirection dataset is retrained using their code of TPAMI version [25].)

Datasets	Rain100H	Rain100L	DID-Data	RainDirection
Derained Image	PSNR/SSIM/LPIPS	PSNR/SSIM/LPIPS	PSNR/SSIM/LPIPS	PSNR/SSIM/LPIPS
GMM [11]	15.23/0.4511/0.573	29.06/0.8720/0.224	24.50/0.8320/0.229	22.92/0.7709/0.355
Clear [2]	13.85/0.4430/0.557	27.39/0.8748/0.215	22.05/0.8396/0.162	22.25/0.8428/0.248
DDN [3]	17.90/0.5621/0.459	29.73/0.9171/0.151	28.37/0.8999/0.136	28.04/0.8746/0.223
JORDER [26]	26.69/0.8347/0.191	36.72/0.9739/0.026	25.52/0.8759/0.179	30.10/0.9064/0.165
DID-MDN [29]	17.36/0.6103/0.443	25.76/0.8597/0.248	27.97/0.9107/0.104	25.70/0.7974/0.350
UMRL [27]	18.21/0.5354/0.465	27.44/0.8717/0.225	30.54/0.9308/**0.074**	26.08/0.7947/0.342
PReNet [19]	29.46/0.8979/0.122	37.42/0.9784/0.019	26.42/0.8889/0.147	28.94/0.8972/0.181
RCDNet [21]	31.28/0.9081/0.104	39.97/0.9856/0.012	27.69/0.8973/0.132	30.39/0.9054/0.170
Ours	**32.62/0.9230/0.090**	**40.54/0.9872/0.009**	**33.07/0.9536**/0.076	**31.35/0.9181/0.153**
Rain Map	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
GMM [11]	15.26/0.3932	29.11/0.7971	26.30/0.6864	22.57/0.6889
Clear [2]	14.24/0.2344	28.83/0.8016	25.16/0.6569	23.85/0.7337
DDN [3]	17.91/0.5287	29.75/0.8380	30.35/0.8282	27.46/0.8383
JORDER [26]	26.54/0.9083	36.61/0.9467	25.79/0.6858	29.29/0.8816
DID-MDN [29]	17.51/0.6220	25.93/0.7152	30.19/0.8376	26.43/0.7812
UMRL [27]	18.21/0.5377	27.52/0.7601	32.84/0.8779	26.73/0.7790
PReNet [19]	29.45/0.9358	37.41/0.9536	25.69/0.6785	28.28/0.8627
RCDNet [21]	31.27/0.9483	39.95/0.9716	29.59/0.8140	29.53/0.8827
Ours	**32.62/0.9590**	**40.53/0.9741**	**34.88/0.9014**	**30.48/0.8978**

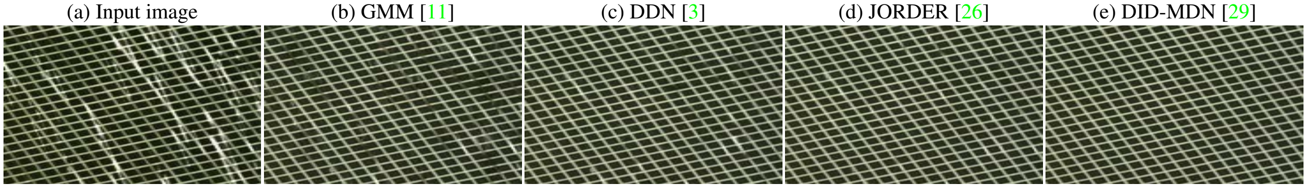
(a) Input image (b) GMM [11] (c) DDN [3] (d) JORDER [26] (e) DID-MDN [29]

(f) UMRL [27] (g) PReNet [19] (h) RCDNet [21] (i) Ours (j) Ground Truth

Figure 3. Example results on Rain100H dataset [26]. Our method restores a more clean image with truthful face structures.



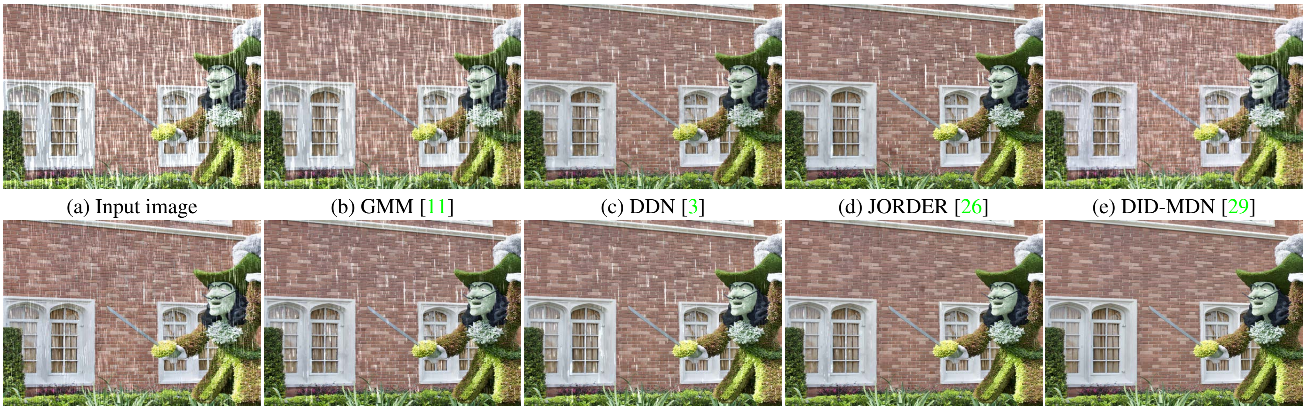
(a) Input image (b) GMM [11] (c) DDN [3] (d) JORDER [26] (e) DID-MDN [29]



(f) UMRL [27] (g) PReNet [19] (h) RCDNet [21] (i) Ours (j) Ground Truth

Figure 4. Example results on Rain100L dataset [26]. Our method reconstructs more accurate and consistent structures.

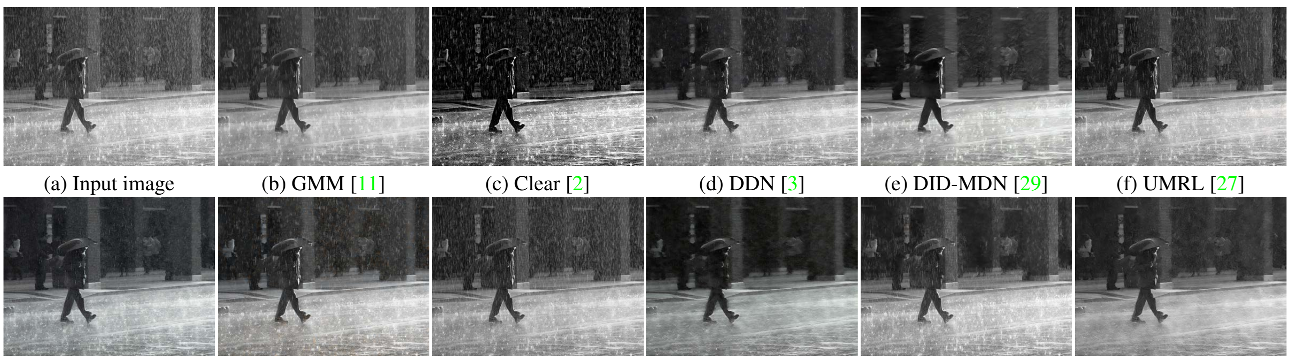
methods. The Gaussian mixture model (GMM) based method [11] does not restore the derained images well and generates the results with significant rain residues. DDN [3] develops a deep detail network for rain removal. However, this method does not restore human faces or important structures well as shown in Figure 3 (c) and Figure 4 (c). JORDER [26] jointly estimates rain locations and rain maps and achieves effective results, but obvious artifacts still exist in the results (Figure 3 (d)). DID-MDN [29] introduces rain density labels while UMRL [27] develops rain streak location information to help rain removal. However, both of them do not remove the rain from the input images well.



(a) Input image (b) GMM [11] (c) DDN [3] (d) JORDER [26] (e) DID-MDN [29]

(f) UMRL [27] (g) PReNet [19] (h) RCDNet [21] (i) Ours (j) Ground Truth

Figure 5. Example results on the RainDirection dataset. Our algorithm recovers a high-quality image with clearer structures and details.



(a) Input image (b) GMM [11] (c) Clear [2] (d) DDN [3] (e) DID-MDN [29] (f) UMRL [27]

(g) PReNet [19] (h) SIRR [23] (i) O'er Me [13] (j) JORDER-E [25] (k) RCDNet [21] (l) Ours

Figure 6. Example results in real-world images. In this challenging case, our method removes the rain streaks (drops) effectively with fewer rain residues against the compared methods.

Table 2. Quantitative evaluations of the compared methods on the proposed Real3000 dataset (300 real test images) in terms of NIQE.

Methods	GMM [11]	DID-MDN [29]	PReNet [19]	SIRR [23]	O'er Me [13]	JORDER-E [25]	Syn2Real [28]	RCDNet [21]	Ours
NIQE	3.7119	3.8162	3.8363	3.2664	3.5511	3.4530	3.1973	3.4401	3.0558

PReNet [19] progressively tackles the deraining problem in multiple stages with a shallow yet effective ResNet. RCDNet [21] also introduces an iterative algorithm. This method converges well after 20 stages. However, artifacts still exist and the structures are not restored well as presented in Figure 3 (h) and Figure 4 (h). In contrast, our proposed method recovers more clean images with truthful structures.

Figure 5 shows some derained results on our proposed RainDirection dataset. We fine-tune the pre-trained models of other CNN-based methods on the proposed RainDirection dataset and choose their best models for evaluation. We note that state-of-the-art methods do not remove the rain streaks well. In contrast, the proposed algorithm restores high-quality images with truthful image details and structures against the others.

Real examples. We further evaluate another real example from [30] where the heavy rain streaks in the air and the raindrops on the ground both exist. Figure 6 shows that the state-of-the-art methods [11, 2, 3, 29, 27, 19, 23, 13, 25, 21] do not restore the derained images well and the results contain rain residues or color distortion. In contrast, our method removes both of the rain streaks and raindrops effectively and restores promising derained images. Table 2 summarizes the quantitative evaluations using the no-reference score NIQE on the proposed Real3000 dataset. Table 2 indicates that our method can achieve high-quality images against the state-of-the-art methods on real-world images.

Table 3. Average running time comparisons (in seconds) on 100 RGB images of 512 x 512 pixels.

Methods	SIRR [23]	JORDER-E [25]	RCDNet [21]	Ours
Platform	TensorFlow	PyTorch	PyTorch	PyTorch
Avg. time	0.15	0.36	0.79	0.17
Model size	0.06M	4.2M	3.2M	4.9M

Running time. We evaluate the average running time on randomly selected 100 RGB images of 512 x 512 pixels on a computer with an Intel Core i9-9940X CPU and an NVIDIA TITAN RTX GPU. Table 3 shows that our method takes slightly more running time compared to SIRR [23] due to a larger model size. JORDER-E [25] and RCDNet [21] take more time than ours because their networks have multiple stages. Our method only needs to evaluate the distillation network in testing.

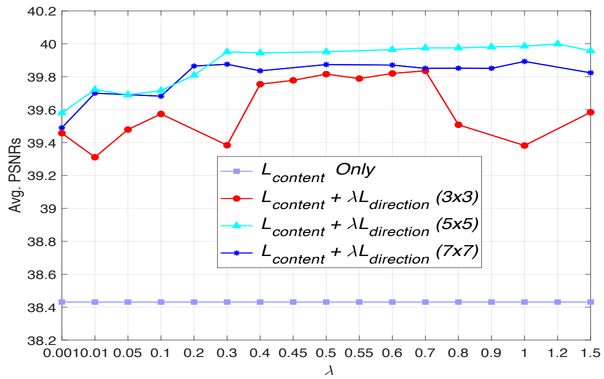


Figure 7. Sensitivity analysis on the hyper-parameters λ and directional filer kernel size of the rain direction regularizer.

Table 4. Effectiveness of the rain direction regularizer (5) for rain removal tasks on RainDirection dataset.

Methods	w/o (5)	Ours
PSNR/SSIM/LPIPS	30.34/0.9083/0.166	31.35/0.9181/0.153

5. Analysis and Discussions

In this section, we further analyze the effectiveness of the components via conducting ablation studies.

5.1. Effectiveness of the rain direction regularizer

We introduce a rain direction regularizer to help preserve the structures and shape of rain maps. To demonstrate the effectiveness of the rain direction regularizer, we analyze the sensitivity of the trade-off parameter λ and directional filter kernel size of the rain direction regularizer. Our proposed rain direction regularizer $L_{direction}$ is different from $L_{content}$ although both the f defined in them are taken as L_1-norm. Figure 7 shows that our method is not sensitive to the change of the hyper-parameter λ within a wide range on the validation set. When $\lambda = 0$, our approach reduces to using the $L_{content}$ only and the results decrease by nearly 1.6 dB in Avg. PNSR. Compared to using the $L_{content}$ only or the directional filter of 3×3 kernel size (reduces to using a horizontal and vertical gradient operator), 5×5 or 7×7 kernel size helps achive better performance. Table 4 demonstrates that using the rain direction regularizer can achieve better derained images in terms of PSNR, SSIM, and LPIPS on the RainDirection dataset. These experiments indicate that the rain direction regularizer effectively helps preserve the structures and shape for image rain removal.

5.2. Effectiveness of the knowledge distillation

The semi-supervised part serves as the teacher part. The knowledge distillation strategy (10) explores the useful information from the synthesized paired data $\{x_{ref}, y_{gt}\}$ and $\{x_{real}, y_{real}\}$. To demonstrate its effectiveness, we disable (10) in the proposed method and retrain with the same settings for fair comparisons. Figure 8 (c) shows the result using the knowledge distillation, where the rain streaks are removed well. In contrast, the result w/o (10) contains artifacts and color distortion as shown in Figure 8 (b). The comparisons in Table 5 demonstrates that using the knowledge distillation can achieve better derained images in terms of PSNR, SSIM, and LPIPS on Ran100L [26] dataset. The distillation network benefits from the complementary information encoded within the synthesized data, thus further improve the results of rain removal tasks.

(a) Real rainy input (b) w/o (10) (c) Ours

Figure 8. Effectiveness of the knowledge distillation in a real case.

Table 5. Effectiveness of the knowledge distillation strategy (10) for rain removal tasks on Rain100L [26] dataset.

Methods	w/o (10)	Ours
PSNR/SSIM/LPIPS	39.15/0.9662/0.013	40.54/0.9741/0.009

5.3. Relations with the most related methods

Recently, a data distillation method has been proposed for rain removal tasks [13]. This approach develops a data distillation strategy to learn real rain maps for rain removal tasks. Our method is significantly different from this approach in model formulation, solving process and learning goals. We differently employ a semi-supervised strategy with an effective rain direction regularizer to estimate the rain map and derained image. The knowledge distillation part and rain estimation module are decoupled, exploring complementary information within the synthesized pseudo paired data and trained in an unpaired learning manner. The quantitative evaluations in Figure 6 and Table 2 further demonstrate that the proposed approach is more effective for rain removal tasks than the state-of-the-art algorithms including this method.

6. Conclusions

In this paper, we have proposed an effective unpaired learning based method for image rain removal. The proposed method explores the simple and well-established layer separation principles according to the linear superimposition model. It mainly consists of a semi-supervised part and a knowledge distillation part, trained by unpaired synthetic and real data. We have developed a rain direction regularizer to help rain map estimation and preserve local consistency and sharpness of the rain streaks. Two new datasets named RainDirection and Rain3000 are proposed to validate the effectivness. Qualitative and quantitative experiments demonstrate that the proposed method achieves promising results and performs favorably against state-of-the-art methods on benchmark datasets and real images.

Acknowledgements: This work is supported in part by National Natural Science Foundation of China (No. 61976041), National Key R&D Program of China (No. 2018AAA0100300).

References

[1] Yi-Lei Chen and Chiou-Ting Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *ICCV*, pages 1968–1975, 2013. 1, 2

[2] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John W Paisley. Clearing the Skies - A Deep Network Architecture for Single-Image Rain Removal. *IEEE TIP*, 26(6):2944–2956, 2017. 1, 2, 5, 6, 7

[3] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John W Paisley. Removing rain from single images via a deep detail network. In *CVPR*, pages 1715–1723, 2017. 1, 2, 3, 5, 6, 7

[4] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *CVPR*, pages 1712–1722, 2019. 4

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, pages 770–778, 2016. 3

[6] 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 *CVPR*, pages 8346–8355, 2020. 2

[7] 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. *TIP*, 28(4):2089–2102, 2019. 1, 2

[8] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv*, 2014. 5

[9] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *arXiv*, 2016. 3

[10] Ruoteng Li, Robby T. Tan, and Loong-Fah Cheong. All in one bad weather removal using architectural search. In *CVPR*, pages 3175–3185, 2020. 1

[11] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown. Rain Streak Removal Using Layer Priors. In *CVPR*, pages 2736–2744, 2016. 1, 2, 5, 6, 7

[12] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced Deep Residual Networks for Single Image Super-Resolution. In *CVPR*, pages 136–144, 2017. 3, 5

[13] Huangxing Lin, Yanlong Li, Xueyang Fu, Xinghao DIng, Yue Huang, and John Paisley. Rain O'er Me: Synthesizing Real Rain to Derain with Data Distillation. *IEEE TIP*, 2020. 1, 2, 3, 5, 7, 8

[14] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. In *ICCV*, pages 3397–3405, 2015. 1, 2

[15] Cheng Ma, Yongming Rao, Yean Cheng, Ce Chen, Jiwen Lu, and Jie Zhou. Structure-preserving super resolution with gradient guidance. In *CVPR*, pages 7769–7778, 2020. 3

[16] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. Making a 'completely blind' image quality analyzer. *IEEE Signal Processing Letters*, 2013. 5

[17] Jinshan Pan, Sifei Liu, Deqing Sun, Jiawei Zhang, Yang Liu, Jimmy S J Ren, Zechao Li, Jinhui Tang, Huchuan Lu, Yu-Wing Tai, and Ming-Hsuan Yang. Learning Dual Convolutional Neural Networks for Low-Level Vision. In *CVPR*, pages 3070–3079, 2018. 2, 3

[18] Dongwei Ren, Wei Shang, Pengfei Zhu, Qinghua Hu, Deyu Meng, and Wangmeng Zuo. Single image deraining using bilateral recurrent network. *IEEE Transactions on Image Processing*, 29:6852–6863, 2020. 2

[19] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive Image Deraining Networks: A Better and Simpler Baseline. In *CVPR*, pages 3937–3946, 2019. 1, 2, 5, 6, 7

[20] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, Lei Zhang, Bee Lim, et al. Ntire 2017 challenge on single image super-resolution: Methods and results. In *CVPR Workshops*, 2017. 5

[21] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A Model-driven Deep Neural Network for Single Image Rain Removal. pages 3103–3112, 2020. 1, 2, 5, 6, 7

[22] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson W.H. Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *CVPR*, pages 12262–12271, 2019. 1, 2

[23] Wei Wei, Deyu Meng, Qian Zhao, Zongben Xu, and Ying Wu. Semi-supervised transfer learning for image rain removal. In *CVPR*, 2019. 2, 4, 5, 7

[24] Yanyan Wei, Zhao Zhang, Haijun Zhang, Jie Qin, and Mingbo Zhao. Semi-deraingan: A new semi-supervised single image deraining network. *arXiv*, 2020. 2, 5

[25] Wenhan Yang, Robby T. Tan, Jiashi Feng, Zongming Guo, Shuicheng Yan, and Jiaying Liu. Joint Rain Detection and Removal from a Single Image with Contextualized Deep Networks. *IEEE TPAMI*, pages 1377–1393,, 2020. 5, 6, 7

[26] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep Joint Rain Detection and Removal from a Single Image. pages 1685–1694, 2017. 1, 2, 3, 5, 6, 7, 8

[27] Rajeev Yasarla and Vishal M Patel. Uncertainty Guided Multi-Scale Residual Learning-using a Cycle Spinning CNN for Single Image De-Raining. In *CVPR*, pages 8405–8414, 2019. 5, 6, 7

[28] Rajeev Yasarla, Vishwanath A. Sindagi, and Vishal M. Patel. Syn2Real transfer learning for image deraining using gaussian processes. In *CVPR*, 2020. 2, 5, 7

[29] He Zhang and Vishal M Patel. Density-aware Single Image De-raining using a Multi-stream Dense Network. In *CVPR*, pages 695–704, 2018. 1, 2, 5, 6, 7

[30] 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*, 2019. 1, 7

[31] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 5

[32] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *arXiv*, 2017. 4