

Closing the Loop: Joint Rain Generation and Removal via Disentangled Image Translation

Yuntong Ye^{1,2}, Yi Chang^{2,*}, Hanyu Zhou¹, Luxin Yan¹

¹National Key Laboratory of Science and Technology on Multispectral Information Processing, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, China

²AI Center, Peng Cheng Laboratory, Shenzhen, China

{yuntongye, yichang, hyzhou, yanluxin}@hust.edu.cn

Abstract

Existing deep learning-based image deraining methods have achieved promising performance for synthetic rainy images, typically rely on the pairs of sharp images and simulated rainy counterparts. However, these methods suffer from significant performance drop when facing the real rain, because of the huge gap between the simplified synthetic rain and the complex real rain. In this work, we argue that the rain generation and removal are the two sides of the same coin and should be tightly coupled. To close the loop, we propose to jointly learn real rain generation and removal procedure within a unified disentangled image translation framework. Specifically, we propose a bidirectional disentangled translation network, in which each unidirectional network contains two loops of joint rain generation and removal for both the real and synthetic rain image, respectively. Meanwhile, we enforce the disentanglement strategy by decomposing the rainy image into a clean background and rain layer (rain removal), in order to better preserve the identity background via both the cycle-consistency loss and adversarial loss, and ease the rain layer translating between the real and synthetic rainy image. A counterpart composition with the entanglement strategy is symmetrically applied for rain generation. Extensive experiments on synthetic and real-world rain datasets show the superiority of proposed method compared to state-of-the-arts.

1. Introduction

Rain is a common weather phenomenon which dramatically degrades the quality of images and affects many computer vision tasks such as detection [18] and segmentation [1]. The forward rain generation procedure [20, 36, 8, 12, 17] is usually simplified as:

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

*Corresponding Author

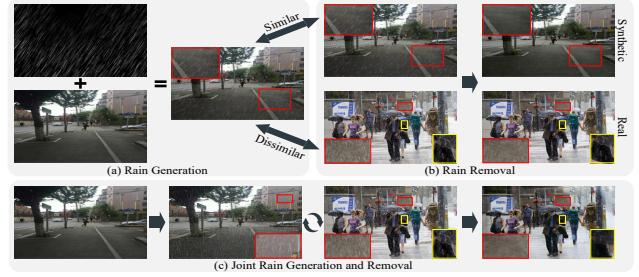


Figure 1. Illustration of rain generation and removal. (a) The hand-crafted simplified rain generation model. (b) Although the simulated rain has been well removed, the huge gap between synthetic training images and real-world testing images leads a significant performance drop in existing learning-based method JORDER [36]. (c) The proposed method learns the complex rain model from the data-driven perspective. We propose to jointly learn the rain generation and removal within a unified framework, as so to better bridges the domain gap between the real and synthetic rain.

where \mathbf{O} , \mathbf{B} , \mathbf{R} denote the rainy image, clean background and rain layer [Fig. 1(a)]. Image deraining is formulated as an ill-posed inverse problem of the rain generation (1), aiming to recover the clean image \mathbf{B} from rainy image \mathbf{O} .

Recently, the deep learning-based deraining methods have achieved remarkable performance, benefiting from powerful representation of convolutional neural network. Fu *et al.* [8] introduced the end-to-end residual CNN for rain streaks removal task. Latter, more sophisticated CNN architectures [36, 19, 21, 27, 17, 34, 28, 6] have sprung up with tremendous progress, and the visual deraining results are impressive. Unfortunately, it is widely recognized that although the trained models can achieve satisfactory results on the synthetic rain, they cannot well generalize to the real rain [Fig. 1(b)] because of the huge gap between the simplified synthetic rain and the complex real rain.

To address this problem, a number of works have been proposed for better real rain removal. The first category

starts from the intuitive rain generation perspective, in which the key idea of this research line is to make the simulated rain as real as possible in supervised manner. They tend to incorporate more complicated visual appearance of rain into consideration, such as streaks, haze and occlusion degradation factors within a comprehensive rain model [36, 21, 17, 12]. However, these hand-crafted generation models cannot well accommodate the complicated distribution of the real rain, due to the varied angle, location, depth, intensity, density, length, width and so on. Also, researchers try to generate the pair of the real rain and clean image from the videos [29] or the rendering technique [11]. On the one hand, the clean image generation is cumbersome; on the other hand, the generated clean image is still pseudo-label, not the oracle clean image. The second category directly resorts to the real image from the domain adaptation perspective [31, 38]. The key idea is to enforce additional constraint between the real and simulated deraining results. These semi-supervised/unsupervised methods can alleviate the domain gap by learning from the real rain, but they neglect the physical rain generation procedure.

The previous methods either focus on the rain generation or rain removal, few of them have noticed a simple yet importance problem that the rain generation and rain removal are of equal importance and should be tightly coupled. The rain removal is a typical inverse problem for the rain generation. A better rain generation model would definitely benefit for real rain removal, or vice versa shown in Fig. 1(c).

In this work, we bridge the gap between the rain generation and rain removal in an end-to-end learning framework. We bypass the difficulty of explicitly designing the sophisticated rain degradation model. Instead, our philosophy is to learn from real rainy image so as to approximate the real degradation implicitly. Specifically, we propose a bidirectional disentangled translation network [Fig. 2], in which each unidirectional network contains two stages of rain generation and removal for both the real and synthetic rain image, respectively. We observe that, in the image translation between the real rain image and simulated rain image, the background clean image layer is consistent while only the rain layer is changed. Instead of directly translating images from synthetic to real domain, this motivates us to preserve the identity in the image background while focus on transforming the simpler rain layer between the real ones and simulated ones. We employ both the self-consistency loss and adversarial loss for the image background. We summarize the main contributions as follows:

- We propose a novel image deraining algorithm which jointly learns the rain generation (forward) and rain removal (inverse) in a unified framework. Compared with the hand-crafted rain generation model, learning physical degradation from real rainy image could offer better approximation to the real rain in an implicit

manner. Moreover, the rain generation and removal would benefit greatly from each other, thus improving the generalization for the real rainy images.

- We employ the disentanglement strategy in which the consistent background is well preserved by the self-consistency loss and adversarial loss. In contrast to previous methods which directly transform the simulated rain image to real rain image, the proposed method gets rid of the identity background and concentrates on the simpler rain layer translating, which significantly ease the difficulty of bridging the gap between real rain and simulated rain image.
- We conduct extensive experiments on both synthetic and real-world datasets, which perform favorably against the state-of-the-art methods and consistently superior on real-world rainy images.

2. Related Work

2.1. Rain Generation

Rain generation is the basis for the rain removal. Nayar and Garg [9, 10] firstly developed a geometric and photometric model with detailed analysis about real rain appearance. Latter, the simple yet intuitive rain image decomposition model in Eq. (1) has been widely used in the optimization-based methods [20, 43, 2]. To better accommodate the complicated visual appearance of real rain, Yang *et al.* [36] and Liu *et al.* [21] extended the simple additive model (rain streaks) to the heavy rain model (distant veiling/haze effect) and occlusion rain model (close rain occlusions), respectively. Along this direction, Hu *et al.* [12] have reached a very high level by taking both streaks, haze and occlusion degradation factors into a comprehensive rain model. Another research line tries to generate the rain image from the real rain. Wang *et al.* [29] proposed a semi-automatic method on rain sequence to generate paired high-quality clean-rainy images. Li *et al.* [18] provided real rain dataset in driving and surveillance without clean counterparts. The rendering-based methods [11] have also been proposed to simulate the rain as real as possible.

However, these complicated hand-crafted rain models still can not accurately reflect the raining procedure of the real physical world, due to the highly complex visual appearance of the real rain. In this work, compared with previous methods, we bypass the difficulty of explicitly designing the sophisticated rain degradation model. Instead, our philosophy is to learn from real rainy image so as to approximate the real degradation implicitly.

2.2. Rain Removal

Rain removal is a highly ill-posed inverse problem to rain generation. Pioneer works design hand-crafted priors to decompose a rainy image into the background and rain layer

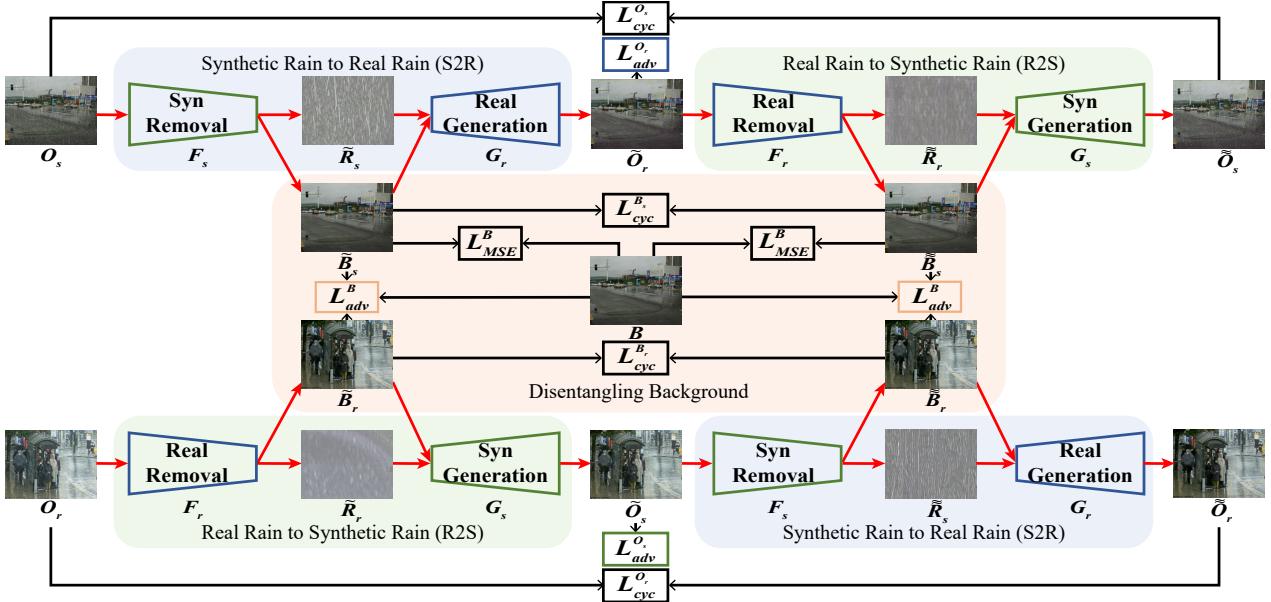


Figure 2. Overview of the proposed joint rain generation and removal (JRGR) framework. The proposed method consists of synthetic to real (S2R) and real to synthetic (R2S) modules, aiming to bridge the gap between synthetic and real rainy images. Instead of the directly translating, we embed the rain removal and generation into each module, which further improves the robustness of the proposed method. To better preserve the identity background image, we introduce both the self-consistency loss and adversarial loss for the clean background.

[14, 4, 22, 20, 43]. Recently, the deep learning-based methods have achieved remarkable progress [7, 19, 39, 30, 24, 12, 17, 31, 38, 37], benefiting from powerful representation of the CNN. Fu *et al.* [8] firstly introduced an end-to-end residual CNN for rain streaks removal. Yang *et al.* [36] jointly detected and removed the rain in a multi-task network. Hu *et al.* [12] designed a depth-attention network to handle both the rain and haze. Albeit the impressive performance of the deraining methods on synthetic rain datasets, these methods suffer a significant performance drop in real-world scene due to the gap between synthetic training data and real testing data. To address this issue, utilizing both the synthetic paired data and real unpaired data, Wei *et al.* [31] designed a semi-supervised framework via a weight-sharing network in which a Gaussian mixture model (GMM) is proposed to regularize the real rain layer. Inspired by [31], Yasarla *et al.* [38] further introduced Gaussian Process (GP) to model the real rain. In this work, we propose to jointly learn the rain generation and rain removal in disentangle image translation framework.

2.3. Generative Adversarial Networks (GAN)

Recently, GAN has attracted extensive attention for image deraining due to its great power of synthesizing photo-realistic images [23, 40, 41, 17]. The generator learns the mapping from the rain image to the clean image, and then the discriminator tries to distinguish between fake/real samples during the train process. Recently, the CycleGAN [42] based domain adaptation methods have been introduced for

the real rain/haze removal with unpaired data [33, 26], aiming to translate the rainy/hazy image domain into the clean image domain in an unsupervised manner. The purposes are still on the rain/haze removal. Typically, Shao *et al.* [26] utilized the architecture of CycleGAN + syn/real haze removal to generate hazy-clean pairs from unpaired real images, and finally focus on the dehazing. In this work, we emphasize the equal importance of rain removal and generation, in which the CycleGAN is formed as the internal cycles (removal \leftrightarrow generation) within the external cycles (synthetic \leftrightarrow real rainy images) for the synergy of removal and generation. Moreover, we propose a novel disentanglement strategy by performing the translations on the rain layers instead of images, which significantly eases the translation since the rain space is much simpler than image space.

3. Proposed Method

3.1. Framework Architecture

Given an real rainy image O , our goal is to estimate the clean background B . The huge gap between the synthetic training images O_s and real testing image O_r , leads significant performance drop in full supervised learning-based methods. In this work, we propose to jointly learn real rain generation and removal procedure within a unified disentangled image translation framework, as shown in Fig. 2.

Specifically, the overall architecture of the JRGR includes both external and internal cycles. The external cycles performs the synthetic and real cycle-consistency respec-

tively, while the internal cycles performs the rain removal and generation cycle-consistency. On one hand, the proposed JRGR has constructed strong relationship between the real and synthetic rainy image via iteratively rain generation and removal (S2R and R2S modules). On the other hand, the overall architecture has great benefit for the feature propagation in both tasks. The S2R module translates the synthetic rainy image into a real rainy image via the disentanglement and entanglement, while R2S module performs the reverse process.

Compared with previous direct image-to-image translation methods, the proposed method JRGR employs the disentangle strategy for better translation. The key importance of the disentanglement is how to choose a proper space with interpretability. The DRIT [16] embeds the images onto two spaces: a domain-invariant content space capturing the shared information and a domain-specific attribute space. However, the two spaces are not clearly defined and constrained. On the contrary, JRGR disentangles the rainy image into the background and rain with clear physical meaning via both MSE and adversarial losses. Moreover, the conventional translation methods treat the two spaces equally, while we treat the spaces asymmetrically, in which the rain space is much simpler than image space. Not that, the rain layer is obtained by subtraction of rainy image and disentangled background. Thus we keep the background consistent while translate only between the real and synthetic rain space. Consequently, JRGR automatically exploits clean backgrounds, synthetic rain layers and real rain layers during the removal and generation, thus possessing the generalization ability in real-world deraining.

3.2. Loss Functions

In this section, we give a detail description about the loss functions. We apply unsupervised adversarial loss, self-supervised cycle-consistency loss and supervised MSE loss for each intermediate removal and generation output.

Adversarial Loss. To enforce the disentanglement of clean background which has no access to the ground truth, we employ a background discriminator D_B which attempts to distinguish the decomposed backgrounds and the photo-realistic background inputs provided by synthetic data. Meanwhile, the removal sub-network attempts to fool the discriminator by generating more realistic backgrounds. We impose the adversarial loss for removal sub-networks in both S2R and R2S modules:

$$\begin{aligned} \mathcal{L}_{adv}^B(D_B, F_s, F_r) = & \mathbb{E}_{O_s}[\log(1 - D_B(F_s(\mathbf{O}_s)))] + \\ & \mathbb{E}_{O_r}[\log(1 - D_B(F_r(\mathbf{O}_r)))] + \alpha \mathbb{E}_B[\log D_B(\mathbf{B})], \end{aligned} \quad (2)$$

where F_s and F_r denote synthetic and real rain removal sub-networks. α serves as the balance weight. In the cycles, another R2S and S2R modules are proposed on the generated synthetic and real rainy images $\tilde{\mathbf{O}}_r$ and $\tilde{\mathbf{O}}_s$ which form the closed-loop. Thus the background adversarial loss is

further applied to the decomposed backgrounds $\tilde{\mathbf{B}}_r$ and $\tilde{\mathbf{B}}_s$. The final background adversarial loss is written as:

$$\begin{aligned} \mathcal{L}_{adv}^B(D_B, F_s, F_r, G_s, G_r) = & \alpha \mathbb{E}_B[\log D_B(\mathbf{B})] \\ & + \mathbb{E}_{O_s}[\log(1 - D_B(F_s(\mathbf{O}_s))) + \log(1 - D_B(F_r(G_r(F_s(\mathbf{O}_s))))))] \\ & + \mathbb{E}_{O_r}[\log(1 - D_B(F_r(\mathbf{O}_r))) + \log(1 - D_B(F_s(G_s(F_r(\mathbf{O}_r))))))] \end{aligned} \quad (3)$$

When the discriminator is updated, the background \mathbf{B} is randomly chosen in the former inputs, following the strategy in [42]. By imposing Eq. 3, we constrain all the decomposed backgrounds in the same domain as the clean input \mathbf{B} .

After the disentanglement of backgrounds, we obtain the rain layers by subtraction. Since the gap between synthetic and real rainy images mainly exists in the rain layers, we perform the translation between rain layers by the rain generation sub-network. The translated rain layers are then entangled with the backgrounds to generate rainy images. For the generated synthetic and real images, we apply two rainy image discriminators to produce the adversarial losses. Taking the real rain generation as an example, the real rainy inputs \mathbf{O}_r are utilized to train the discriminator D_{O_r} and generation sub-network G_r in an adversarial manner:

$$\begin{aligned} \mathcal{L}_{adv}^{O_r}(D_{O_r}, F_s, G_r) = & \mathbb{E}_{O_r}[\log D_{O_r}(\mathbf{O}_r)] \\ & + \mathbb{E}_{O_s}[\log(1 - D_{O_r}(G_r(F_s(\mathbf{O}_s))))], \end{aligned} \quad (4)$$

A similar adversarial loss $\mathcal{L}_{adv}^{O_s}$ is imposed on synthetic generation sub-network G_s and the discriminator D_{O_s} to enforce the generation of synthetic rainy image $\tilde{\mathbf{O}}_s$.

Cycle-consistency Loss. Although the adversarial losses guarantee the generated images and the target images in the same domain, the content information in rainy images may be degraded during the generation. Benefiting from the closed-loop in our framework, we employ four cycle-consistency losses which self-supervise the removal and generation sub-networks to preserve the content information. Firstly, while S2R module generates real rainy images from synthetic rainy images, the reconstructed synthetic output which is generated by the latter reverse process of R2S module should be consistent with the synthetic input. Thus we define the cycle-consistency loss as:

$$\mathcal{L}_{cyc}^{O_s}(F_s, F_r, G_s, G_r) = \mathbb{E}_{O_s}[\|\mathbf{O}_s - G_s(F_r(G_r(F_s(\mathbf{O}_s))))\|_1]. \quad (5)$$

Meanwhile, for the reconstruction of \mathbf{O}_r , a similar cycle-consistency loss $\mathcal{L}_{cyc}^{O_r}$ is applied to train the other cycle.

Secondly, the generated real rainy images $\tilde{\mathbf{O}}_r$ and the original synthetic rainy images \mathbf{O}_s should possess consistent backgrounds. The cycle-consistency loss is written as:

$$\mathcal{L}_{cyc}^{B_s}(F_s, F_r, G_r) = \mathbb{E}_{O_s}[\|F_s(\mathbf{O}_s) - (F_r(G_r(F_s(\mathbf{O}_s))))\|_1]. \quad (6)$$

In a similar fashion, the cycle-consistency loss $\mathcal{L}_{cyc}^{B_r}$ is defined for the decomposed backgrounds $\tilde{\mathbf{B}}_r$ and $\tilde{\mathbf{B}}_s$.

MSE Loss. In addition to the adversarial losses and the cycle-consistency losses, for the synthetic rainy images which have ground truth background counterparts, we utilize the MSE loss to supervise the training of synthetic re-

removal sub-network, which is written as:

$$\begin{aligned} \mathcal{L}_{mse}^{B_s}(F_s, F_r, G_r) &= \|F_s(\mathbf{O}_s) - \mathbf{B}\|_2^2 \\ &\quad + \|F_r(G_r(F_s(\mathbf{O}_s))) - \mathbf{B}\|_2^2. \end{aligned} \quad (7)$$

Full Objective. The full objective function contains the adversarial loss, cycle-consistency loss, and the MSE loss as follow:

$$\begin{aligned} \mathcal{L}(F_s, F_r, G_s, G_r, D_B, D_{O_s}, D_{O_r}) &= \lambda_{adv}(L_{adv}^B + L_{adv}^{D_{O_s}} + L_{adv}^{D_{O_r}}) \\ &\quad + \lambda_{cyc}(L_{cyc}^{O_r} + L_{cyc}^{O_s} + L_{cyc}^B + L_{cyc}^{B_s}) + \lambda_{mse}L_{mse}^{B_s}. \end{aligned} \quad (8)$$

By imposing the full objective function, we alleviate the problem that paired real rainy-clean images are not accessible. In our framework, each removal and generation sub-network is constrained by at least one unsupervised adversarial loss and one self-supervised cycle-consistency loss to guarantee the domain of the outputs while preserve the content information, which consequently guides the framework to exploit the clean backgrounds and rain layers.

3.3. Implementation Details

The framework is implemented using the PyTorch with four RTX 2080Ti GPUs. We utilize the U-Net [25] as the removal and rain generation sub-networks. Subtraction and summation are followed to obtain the disentangled rain layers and generated rainy images. The PatchGAN [13, 15] is utilized to construct the discriminators. We empirically set the balance weight α , λ_{adv} , λ_{cyc} and λ_{mse} as 4, 10, 1, 10.

The synthetic and real images are randomly cropped into 256×256 as input of each sub-network. We first pre-train the synthetic removal sub-network and real removal sub-network on paired synthetic dataset for 100 epochs with the learning rate 0.0001. Then we jointly train JRGR using paired synthetic data and unpaired real data for 200 epochs, in which the learning rates of generation sub-networks are set as 0.0001 while divided by 10 and 100 for real and synthetic removal sub-network. The Adam optimizer is adopted with batch size 16. In testing, we apply the real removal sub-network to output clean backgrounds.

4. Experiments

4.1. Datasets and Experimental Setting

Datasets. We conduct experiments on both synthetic and real rain datasets to evaluate the proposed method.

- **Cityscape.** We utilize two different synthetic rain datasets, RainCityscape [12] and Rendering [11], which synthesize rain and haze with different models on the Cityscape dataset [5]. To simulate the real situation, during training, we use 1400 paired rainy-clean images in RainCityscape as synthetic data and 1400 unpaired rainy images in RainRendering with no access to the clean counterparts as real data. 175 rainy images in RainRendering are used for testing.

- **RainHQ.** RainHQ dataset is a high quality real rain dataset which provides paired rainy-clean patches via semi-automatic labelling methods [29]. We use 2000 paired synthetic rainy-clean images following the strategy in [8] and 2000 unpaired rainy images for training. 500 real rainy images are utilized for testing.

- **RealRain.** We also collect unpaired real world rainy images and clean backgrounds with large field of view from the datasets provided by [36, 32, 40, 3] and Google search. We utilize totally 400 real rainy images in RealRain as the unpaired rainy images and 400 paired synthetic rainy images to train the framework. 88 real rainy images are utilized for test.

Experimental Setting. We compare JRGR with (1) supervised deraining methods DDN [8], JORDER-E [35], RESCAN [19], DAF [12] and the GAN-based generation method pix2pix [13], which are trained with paired synthetic rainy-clean images and tested in real rainy images; (2) unsupervised method Cycle GAN [42] trained with real rainy images and clean backgrounds provided by synthetic data; (3) semi-supervised methods SIRR [31] and Syn2Real [38] which are trained with both synthetic and real data. PSNR and SSIM are utilized for quantitative evaluation.

4.2. Experiments on Synthetic Images

Figure 3 shows the visual results on synthetic dataset Rendering [11]. For the supervised methods, they cannot well generalize on Rendering dataset due to the different rain streak appearances in training set RainCityscape and testing set Rendering. As a result, the rain streaks exist in most deraining results. SSIR learns the rain streaks in Rendering in a semi-supervised manner and removes the rain streaks in the sky. However, the hand-craft designed GMM is limited to model the diverse rain such as the long rain streak on the floor. The unsupervised Cycle GAN generates artifacts on the floor which degrades the visualization quality in deraining result. The proposed JRGR learns the distribution of backgrounds and rain layers during removal and generation, which consequently generates the satisfactory deraining results. The quantitative results in Table 1 also demonstrate the superiority of our proposed method.

4.3. Experiments on Real Images

In Fig. 4, we visualize the rain removal results on real rain dataset RainHQ [29], in which the rain is more complex than Rendering. State-of-the-art deraining methods have difficulty in dealing with real-world rain and leave the rain streaks unremoved. The artifacts in Cycle GAN degrade deraining results. The proposed method possesses the clean and smooth backgrounds. Quantitative comparison is shown in Table 1. The performance of DAF [12] significantly drops without re-training using depth. Cycle GAN possesses a comparable quantitative result. By imposing

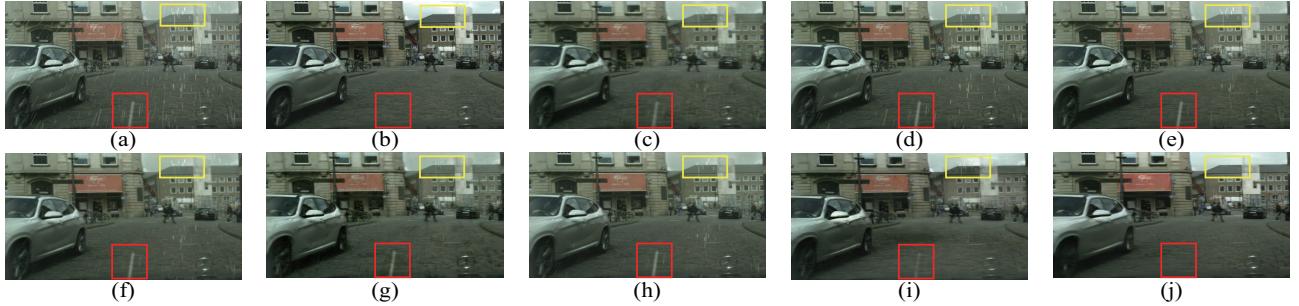


Figure 3. Visualization of deraining results on Rendering dataset. (a) Rainy image. (b) Clean image. Deraining results by (c) DDN, (d) JORDER-E, (e) RESCAN (f) DAF, (g) SSIR, (h) Syn2Real, (i) Cycle GAN, (j) JRGR.

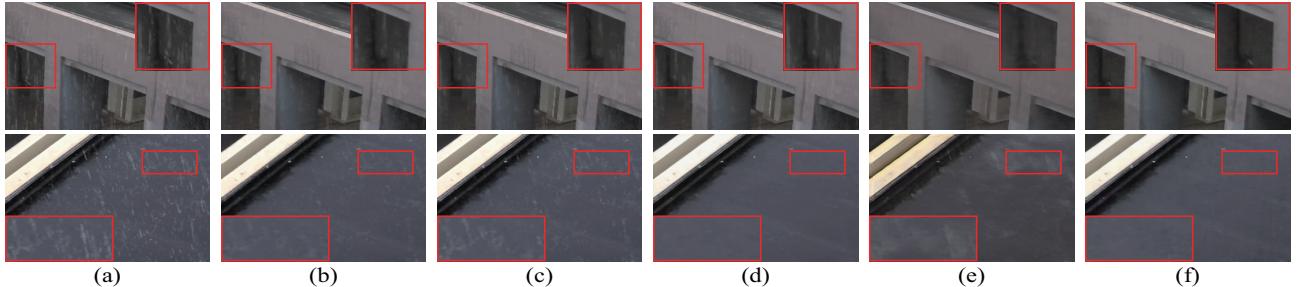


Figure 4. Visualization of deraining results on RainHQ dataset. (a) Rainy image. Deraining results by (b) DDN, (c) JORDER-E, (d) Syn2Real, (e) Cycle GAN, (f) JRGR.

Table 1. Quantitative comparison with state-of-art methods on synthetic and real datasets.

Dataset	Rendering [11]		RainHQ [29]	
	PSNR	SSIM	PSNR	SSIM
DDN [8]	24.12	0.8781	33.74	0.9112
RESCAN [19]	24.89	0.9101	33.11	0.9253
DAF [12]	25.23	0.8827	24.15	0.8523
JORDER-E [35]	25.64	0.8767	33.28	0.9406
pix2pix [13]	25.21	0.8833	32.43	0.9148
Cycle GAN [42]	26.29	0.8921	33.54	0.9127
SSIR [31]	25.08	0.8853	30.78	0.8668
Syn2Real [38]	25.32	0.8871	33.14	0.9183
JRGR	27.51	0.9132	35.59	0.9498

disentanglement and translation on the rain layer, the proposed method obtains more natural results and achieves the best performance in terms of PSNR and SSIM.

Furthermore, we show the visual results on RealRain dataset in Fig. 5. DDN, JORDER-E and Syn2Real tend to leave the rain streak on deraining results. More artifacts are produced by Cycle GAN when facing the complex scene. The proposed JRGR removes most of the rain, while preserves the details of backgrounds.

4.4. Ablation Study

In this section, ablation study on RainHQ is conducted to evaluate the effectiveness of losses and training strategy.

Table 2. The ablation study for different loss functions and training strategies on real rain dataset RainHQ. Init-1 and Init-2 denote two different training strategies.

Strategy	PSNR	SSIM
w/o L_{adv}^B	27.38	0.8926
w/o L_{adv}^O	28.15	0.9012
w/o L_{cyc}	26.87	0.8902
w/o L_{MSE}	31.24	0.9144
Init-1	33.97	0.9305
Init-2	34.54	0.9424
Proposed	35.59	0.9498

Effectiveness of Losses. Denoting L_{adv}^O as the rainy image adversarial losses, i.e., $L_{adv}^{O_s}$ and $L_{adv}^{O_t}$, while L_{cyc} as the cycle-consistency losses, we investigate the effectiveness of the loss functions in Table 2. The performance drops in the first four rows demonstrate the benefits brought by the adversarial, cycle-consistency and MSE loss.

Effectiveness of Training Strategy. We also study the effectiveness of training strategy in Table 2. Init-1 trains the sub-networks in JRGR together with no pre-training. Init-2 pre-trains the synthetic removal sub-network on the paired synthetic dataset and then trains the sub-networks together, in which the learning rate of synthetic removal sub-network is divided by 100. The proposed method further pre-trains the real removal sub-network and then trains the sub-networks together, in which the learning rate of real

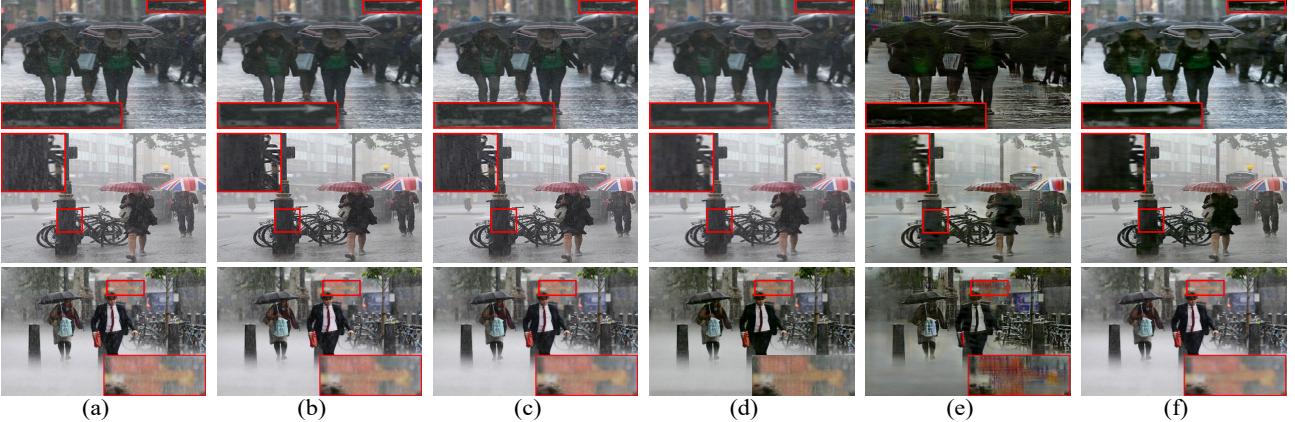


Figure 5. Visualization of deraining results on RealRain dataset. (a) Rainy image. Deraining results by (b) DDN, (c) JORDER-E, (d) Syn2Real, (e) Cycle GAN, (f) JRGR.

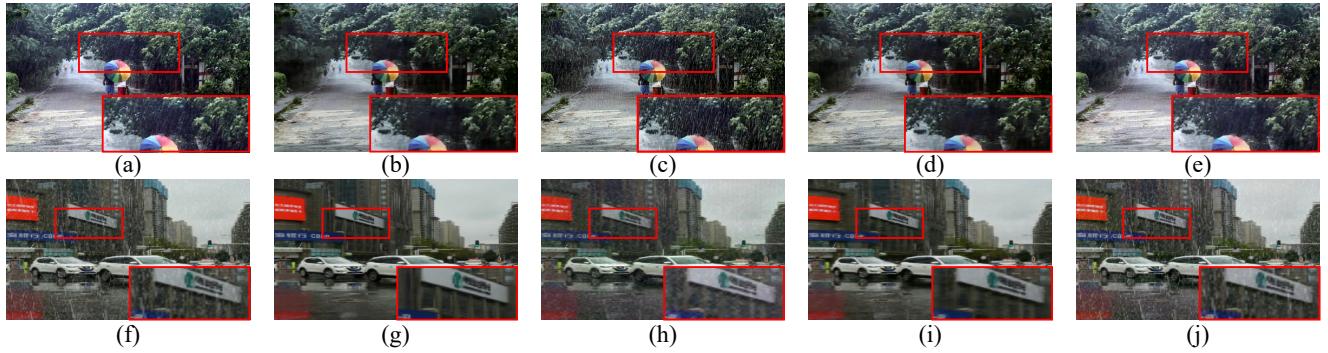


Figure 6. The effectiveness of the joint removal-generation framework. (a) Real input, (b) decomposed background from (a), (c) generated synthetic rainy image, (d) decomposed background from (c), (e) reconstructed real input, (f) synthetic input, (g) decomposed background from (f), (h) generated real rainy image, (i) decomposed background from (h), (j) reconstructed synthetic input. By the mutual effect of removal and generation sub-networks, JRGR obtains good performance in each intermediate output.

removal sub-network is divided by 10. The quantitative results show the initialization of the synthetic and real removal sub-networks is necessary.

4.5. Discussion

Analysis of Joint Rain Removal and Generation. We investigate all the intermediate removal and generation output of JRGR. In Fig. 6, all the estimated backgrounds (b), (d), (g), (i) are clean without rain streaks. The rain streaks in generated synthetic rainy image are in a synthetic style [Fig. 6(c)], while the generated real rainy image [Fig. 6(h)] possesses more photo-realistic rain which is similar with the input [Fig. 6(a)]. The final outputs [(Fig. 6(e), (j))] successfully reconstruct the inputs [Fig. 6(a), (f)]. The proposed JRGR could provide physical meaning results with good performance via the adversarial and cycle-consistency losses which guarantee both the translation performance and identity preservation. In Fig. 7, we present more high quality realistic rainy images generated by JRGR, which includes both the natural rain veiling and streaks.

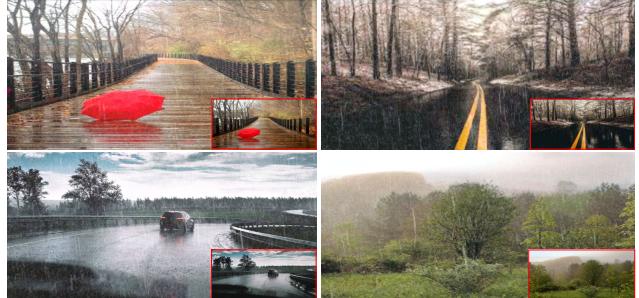


Figure 7. Visualization results of generated real rains by JRGR.

Analysis of Rain Layer Translation. To study the performance of rain layer translation, we visualize the features of synthetic and real rain layers using t-SNE in Fig. 8. On the one hand, the separation of the synthetic and real rain layer feature demonstrates the huge gap in synthetic and real rain, which is also illustrated by the appearance difference of the corresponding rain layer visualizations. On the other hand, the decomposed and generated rain layers are

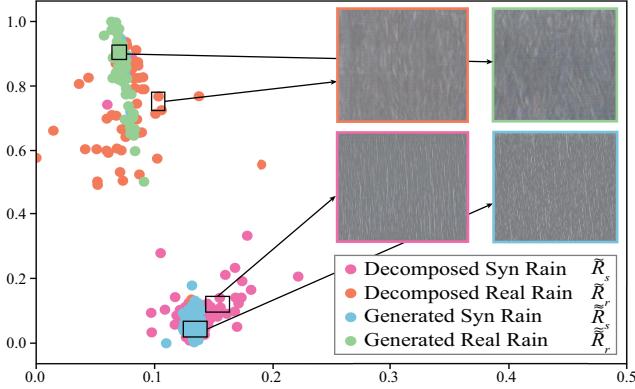


Figure 8. Visualization results of t-SNE on the decomposed and generated rain layers in real and synthetic data.



Figure 9. The superiority of translation between rain layers. Top: CycleGAN directly translates background into rainy image leaving obvious artifact. Bottom: JRGR preserves the background and focuses on the translation of rain layer, obtaining natural result.

aligned in both synthetic and real rain domain (green with orange, blue with pink). The visualized decomposed and generated rain layers are also similar, which demonstrates that our rain generators successfully perform the translation between synthetic and real rain layers.

Furthermore, to show the superiority of performing translation between simpler rain layers, we compare JRGR with Cycle GAN which directly performs the translation between backgrounds and rainy images in Fig. 9. Cycle GAN pays attention to the most discriminative features between backgrounds and rainy images, which may not be the rain but the other factors such as textures and colors. On the contrary, the backgrounds of JRGR are well decomposed and preserved during removal, which makes it possible for the generator to focus on the translation between rain layers. That is the main reason why the JRGR could obtain more natural results than that of the Cycle GAN.

Limitation. The JRGR mainly bridges the inter-domain gap between synthetic and real rainy images. However, there still exists the intra-domain gap within real rainy images. For example, the real rainy city and forest images contain the different color and texture characteristics. These intra-domain gaps in the real rainy image would lead to the performance drop. In Fig. 10, we show the deraining ex-



Figure 10. The limitation of our JRGR for intra-domain gap. (a) Examples of backgrounds in three different training synthetic datasets. (b) JRGR tends to generate results with different color styles when trained with different datasets. JRGR fails when the training and testing backgrounds are significantly different.

amples by three datasets with different backgrounds, i.e., city under overcast sky, city under clear sky and forests. When the backgrounds behind real and synthetic data are significantly different, the proposed method fails to generate good deraining results. For example, the method trained by overcast sky dataset obtains the best performance for the backgrounds are most similar with the rainy images. Definitely, the comprehensive datasets would alleviate the problem from the dataset perspective. We would like to tackle the inter and intra-domain gaps simultaneously in future.

5. Conclusion

In this paper, we have proposed a disentangled image translation framework which jointly learns rain generation and removal to address the real-world deraining problem. Specifically, a bidirectional translation network are constructed by the removal and generation sub-networks with tightly coupled generation and removal modules. We further employ the disentangled strategy on the rain image by decomposing the rainy image into clean background and rain layer, so as to preserve the identity background and ease the translation with only the rain layer. The removal and generation sub-networks, constrained by the adversarial and cycle-consistency losses, mutually affect each other and consequently endow the framework with the generalization ability of dealing with real rain. The extensive experiments on synthetic and real rain datasets demonstrate the superiority of the proposed framework.

Acknowledgements. This work was supported by National Natural Science Foundation of China under Grant No. 61971460, China Postdoctoral Science Foundation under Grant 2020M672748, National Postdoctoral Program for Innovative Talents BX20200173 and Industrial Technology Development Program grant JCKY2018204B068.

References

- [1] Chris H Bahnsen and Thomas B Moeslund. Rain removal in traffic surveillance: Does it matter? *IEEE Trans. Intell. Transp. Syst.*, 20(8):2802–2819, 2018. [1](#)
- [2] Y. Chang, L. Yan, and S. Zhong. Transformed low-rank model for line pattern noise removal. In *Int. Conf. Comput. Vis.*, pages 1726–1734, 2017. [2](#)
- [3] J. Chen, C. Tan, J. Hou, L. Chau, and H. Li. Robust video content alignment and compensation for rain removal in a cnn framework. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6286–6295, 2018. [5](#)
- [4] Y. Chen and C. Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *Int. Conf. Comput. Vis.*, pages 1968–1975, 2013. [3](#)
- [5] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3213–3223, 2016. [5](#)
- [6] Y. Du, J. Xu, X. Zhen, M. Cheng, and L. Shao. Conditional variational image deraining. *IEEE Trans. Image Process.*, 2020. [1](#)
- [7] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Trans. Image Process.*, 26(6):2944–2956, 2017. [3](#)
- [8] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley. Removing rain from single images via a deep detail network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3855–3863, 2017. [1, 3, 5, 6](#)
- [9] K. Garg and S. Nayar. When does a camera see rain? In *Int. Conf. Comput. Vis.*, pages 1067–1074, 2005. [2](#)
- [10] K. Garg and S. Nayar. Vision and rain. *Int. J. Comput. Vis.*, 75(1):3–27, 2007. [2](#)
- [11] S. Halder, J. Lalonde, and R. Charette. Physics-based rendering for improving robustness to rain. In *Int. Conf. Comput. Vis.*, pages 10203–10212, 2019. [2, 5, 6](#)
- [12] X. Hu, C. Fu, L. Zhu, and P. Heng. Depth-attentional features for single-image rain removal. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8022–8031, 2019. [1, 2, 3, 5, 6](#)
- [13] P. Isola, J. Zhu, T. Zhou, and A. Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1125–1134, 2017. [5, 6](#)
- [14] L. Kang, C. Lin, and Y. Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Trans. Image Process.*, 21(4):1742–1755, 2011. [3](#)
- [15] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, and Z. Wang. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4681–4690, 2017. [5](#)
- [16] H. Lee, H. Tseng, J. Huang, M. Singh, and M. Yang. Diverse image-to-image translation via disentangled representations. In *Eur. Conf. Comput. Vis.*, pages 35–51, 2018. [4](#)
- [17] R. Li, L. Cheong, and R. Tan. Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1633–1642, 2019. [1, 2, 3](#)
- [18] S. Li, I. Araujo, W. Ren, Z. Wang, E. Tokuda, R. Junior, R. Cesar-Junior, J. Zhang, X. Guo, and X. Cao. Single image deraining: A comprehensive benchmark analysis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3838–3847, 2019. [1, 2](#)
- [19] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *Eur. Conf. Comput. Vis.*, pages 254–269, 2018. [1, 3, 5, 6](#)
- [20] Y. Li, R. Tan, X. Guo, J. Lu, and M. Brown. Rain streak removal using layer priors. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2736–2744, 2016. [1, 2, 3](#)
- [21] J. Liu, W. Yang, S. Yang, and Z. Guo. Erase or fill? deep joint recurrent rain removal and reconstruction in videos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3233–3242, 2018. [1, 2](#)
- [22] Y. Luo, Y. Xu, and H. Ji. Removing rain from a single image via discriminative sparse coding. In *Int. Conf. Comput. Vis.*, pages 3397–3405, 2015. [3](#)
- [23] R. Qian, R. Tan, W. Yang, J. Su, and J. Liu. Attentive generative adversarial network for raindrop removal from a single image. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2482–2491, 2018. [3](#)
- [24] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng. Progressive image deraining networks: A better and simpler baseline. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3937–3946, 2019. [3](#)
- [25] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015. [5](#)
- [26] Y. Shao, L. Li, W. Ren, C. Gao, and N. Sang. Domain adaptation for image dehazing. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2808–2817, 2020. [3](#)
- [27] G. Wang, C. Sun, and A. Sowmya. Erl-net: Entangled representation learning for single image de-raining. In *Int. Conf. Comput. Vis.*, pages 5644–5652, 2019. [1](#)
- [28] H. Wang, Q. Xie, Q. Zhao, and D. Meng. A model-driven deep neural network for single image rain removal. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3103–3112, 2020. [1](#)
- [29] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, and R. Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12270–12279, 2019. [2, 5, 6](#)
- [30] Z. Wang, J. Li, and G. Song. Dtdn: Dual-task de-raining network. In *ACM Int. Conf. Multimedia*, pages 1833–1841, 2019. [3](#)
- [31] W. Wei, D. Meng, Q. Zhao, Z. Xu, and Y. Wu. Semi-supervised transfer learning for image rain removal. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3877–3886, 2019. [2, 3, 5, 6](#)
- [32] W. Wei, L. Yi, Q. Xie, Q. Zhao, D. Meng, and Z. Xu. Should we encode rain streaks in video as deterministic or stochastic? In *Int. Conf. Comput. Vis.*, pages 2516–2525, 2017. [5](#)

- [33] Y. Wei, Z. Zhang, Y. Wang, J. Fan, S. Yan, and M. Wang. Deraincyclegan: A simple unsupervised network for single image deraining and rainmaking. *arXiv preprint arXiv:1912.07015*, 2020. [3](#)
- [34] W. Yang, J. Liu, S. Yang, and Z. Guo. Scale-free single image deraining via visibility-enhanced recurrent wavelet learning. *IEEE Trans. Image Process.*, 28(6):2948–2961, 2019. [1](#)
- [35] W. Yang, R. Tan, J. Feng, Z. Guo, S. Yan, and J. Liu. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(6):1377–1393, 2019. [5, 6](#)
- [36] W. Yang, R. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan. Deep joint rain detection and removal from a single image. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1357–1366, 2017. [1, 2, 3, 5](#)
- [37] R. Yasarla and V. Patel. Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image deraining. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8405–8414, 2019. [3](#)
- [38] R. Yasarla, V. Sindagi, and V. Patel. Syn2real transfer learning for image deraining using gaussian processes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2726–2736, 2020. [2, 3, 5, 6](#)
- [39] H. Zhang and V. Patel. Density-aware single image deraining using a multi-stream dense network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 695–704, 2018. [3](#)
- [40] H. Zhang, V. Sindagi, and V. Patel. Image de-raining using a conditional generative adversarial network. *IEEE Trans. Circuit Syst. Video Technol.*, 2019. [3, 5](#)
- [41] H. Zhu, X. Peng, J. Zhou, S. Yang, V. Chanderasekh, L. Li, and J. Lim. Singe image rain removal with unpaired information: A differentiable programming perspective. In *AAAI*, volume 33, pages 9332–9339, 2019. [3](#)
- [42] J. Zhu, T. Park, P. Isola, and A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Int. Conf. Comput. Vis.*, pages 2223–2232, 2017. [3, 4, 5, 6](#)
- [43] L. Zhu, C. Fu, D. Lischinski, and P. Heng. Joint bi-layer optimization for single-image rain streak removal. In *Int. Conf. Comput. Vis.*, pages 2526–2534, 2017. [2, 3](#)