MobileDeRainGAN: An Efficient Semi-Supervised Approach to Single Image Rain Removal for Task-Driven Applications

Ruphan Swaminathan *
Columbia University
rs4203@columbia.edu

Pradyot Korupolu
Ottonomy Inc
pradyot.korupolu@ottonomy.io

Abstract

Rain removal is an essential task in computer vision, particularly for applications such as autonomous navigation to function seamlessly during rain. However, most existing single-image deraining algorithms are limited by their inability to generalize on diverse real-world rainy images, the need for real-time processing, and the lack of task-driven metric enhancement. This paper proposes MobileDeRainGAN, an efficient semi-supervised algorithm that addresses these challenges. The proposed approach includes a novel latent bridge network and multi-scale discriminator that effectively removes rain-related artifacts at different scales. Our cross-domain experiments on Rain1400 and RainCityscapes datasets demonstrate substantial improvements over state-of-the-art methods in terms of generalization and object detection scores in a semi-supervised setting. Furthermore, our approach is significantly faster and can run in real-time even on edge devices. Overall, our proposed MobileDeRainGAN algorithm offers a significant improvement in rain removal performance on real-world images while being efficient, scalable, and suitable for real-world applications.

1. Introduction

Rainfall can severely degrade the quality of images captured during adverse weather conditions, resulting in artifacts such as occlusion. Rain removal plays a critical role in computer vision applications such as autonomous navigation and surveillance, where image quality is paramount for accurate analysis and decision making. The popular benchmark datasets and models used in the field tend to ignore images with rain, leading to a performance drop during rainfall. Therefore, removing the artifacts caused by rain from images is critical to ensure seamless performance of the algorithms such as object recognition, detection, and segmentation [36, 37] even during rainfall.

*Work done while interning at Ottonomy Inc

Figure 1. Comparison of synthetic rain images with real-world rain. Typical synthetic rain images (top row) lack diversity while real-world rain (bottom row) exhibits diverse patterns, often with the same image in terms of streak size, direction and appearance.

Over the last two decades, rain removal research has made significant progress. However, it faces several challenges, including i) the lack of incorporating real-world datasets during training, which results in a domain gap, ii) computationally expensive forward passes that impede real-time performance, and iii) the neglect of evaluating application-oriented metrics. Similar research problems like raindrop and haze removal have tackled the first challenge by creating setups to collect real data. Qian et al. [34] curated a paired image dataset for raindrop removal by capturing image pairs with and without a glass plane sprinkled with water droplets. Ancuti et al. [1] created a paired image dataset called O-HAZE by generating haze using professional haze machines. However, such a paired image dataset doesn't exist for rain removal. Most works address this issue by synthetically adding rain to clean images [7], resulting in poor performance when tested on real-world rain images. Conversely, [40] proposed an alternative solution by artificially removing rain streaks from rainy images to generate a paired dataset using human supervision. However, the dataset had a fundamental limitation of lacking diversity in the rain patterns [4]. Some works [42] also address this challenge by developing sophisticated models that include different components of rain in images but they can never entirely capture the real-world rain model due to a large

amount of diversity. Most existing works fail to tackle the second challenge and have a poor inference time, rendering it impractical to be used as a preprocessing step for tasks like real-time object detection. Only a few works, such as LPNet [8], have focused on introducing lightweight components into the architecture design. The third major challenge is that, while existing works acknowledge the applications of rain removal to benefit tasks like autonomous navigation and surveillance, they are largely neglected during evaluation. Almost all rain removal algorithms are evaluated only using traditional image quality metrics such as PSNR and SSIM and fail to consider the performance improvement in object recognition, detection and segmentation. It is even possible for a performance drop in such tasks after deraining due to undesirable alterations introduced in the non-rainy background, as shown in later sections. For instance, Rai et al. [35] observed that the mIoU score for semantic segmentation after rain removal was significantly lower than that of the original rainy images.

In recent years, various Generative Adversarial Network (introduced by Goodfellow et al. [11]) based approaches have been proposed to address the problem of image artifacts caused by rain [21,28,50]. Although these approaches have demonstrated promising results, GAN-based models often make more modifications to the image than necessary, resulting in overcompensation. This effect is illustrated in Fig. 4 and quantitatively evaluated in later sections. While this may not seem like a significant drawback when assessing the results qualitatively, it can cause problems when using the overcompensated image as input for other neural networks, such as an object detection model. In fact, the overcompensation acts as a form of an adversarial attack, as the minor changes can deceive benchmarked models. Furthermore, most of these works evaluate their results on labeled synthetic datasets and overlook task-driven metrics, which can limit their practical usefulness in real-world applications.

This paper proposes MobileDeRainGAN, a novel approach to single image deraining using generative adversarial learning. To address the challenge of real-time performance, we draw inspiration from MobileNetv2 [38] and use inverted residual blocks in the generator network. To bridge the domain gap between labeled and unlabeled data, we introduce a latent bridge network. Furthermore, we employ a multi-scale discriminative network to produce globally coherent images while simultaneously attending to finer rain artifacts. To accommodate different hardware and application requirements, we demonstrate a trade-off between image quality and inference time. In summary, the key contributions of this paper are as follows:

- We propose MobileDeRainGAN, which is a semi-supervised deraining algorithm that effectively incorporates real-world unlabeled data during training, unlike many existing works in the literature.
- Our proposed method achieves real-time performance even on edge devices, making it highly suitable for practical applications.
- Through cross-domain experiments and evaluation of application-based metrics, we demonstrate that MobileDeRainGAN is a useful preprocessing step for autonomous navigation and surveillance systems, paving the way for further advancements in this field.

2. Related Works

Removing rain from images has been an extensively studied problem in the literature and received significant attention over the past two decades. The problem is broadly classified into two categories. The first category is single image deraining, which uses the spatial context within a single image to reconstruct a rain-free image. Earlier approaches depended on image priors and used techniques like image decomposition [16], sparse coding [27], non-local mean filtering [17] and Gaussian mixture models [24]. The second category is video or multi-image deraining, which utilizes both spatial and temporal context in a sequence of images. Some of the first works [9, 10] in the field were video-based methods that relied on photometric properties of rain for detection and averaging them across the temporal domain for removal.

Recently, deep learning-based approaches have become the standard for image deraining due to their effectiveness in exploiting large amounts of data. Yang et al. [43] proposed a recurrent neural network that jointly detects and removes rain by decomposing them into multiple layers of rain streaks of different shapes and directions. Fu et al. [6] presented a convolutional neural network-based method to learn a mapping between rain and rain-free image pairs. Zhang et al. [50] explore using a conditional generative adversarial network to obtain higher quality reconstruction. Fu et al. [7] proposed a deep detail network, an end-to-end CNN, to learn a mapping function between the residual layer and detail layer. Zhang and Patel [49] employed a density-aware multi-stream densely connected CNN to classify rain density to guide rain removal from images. Li et al. [22] proposed a contextual dilated network with attention to predict stage-wise residual iteratively. Chen et al. [3] formulated a video-based method for spatial-temporal content alignment and a convolutional neural network to reconstruct rain-free background from a rainy image. Yang et al. [44] proposed SLDNet to self-learn rain streak removal and recover rain-free background using temporal correlation and consistency from image sequences.

While video-based methods tend to self-supervise the learning process due to temporal consistency, the single

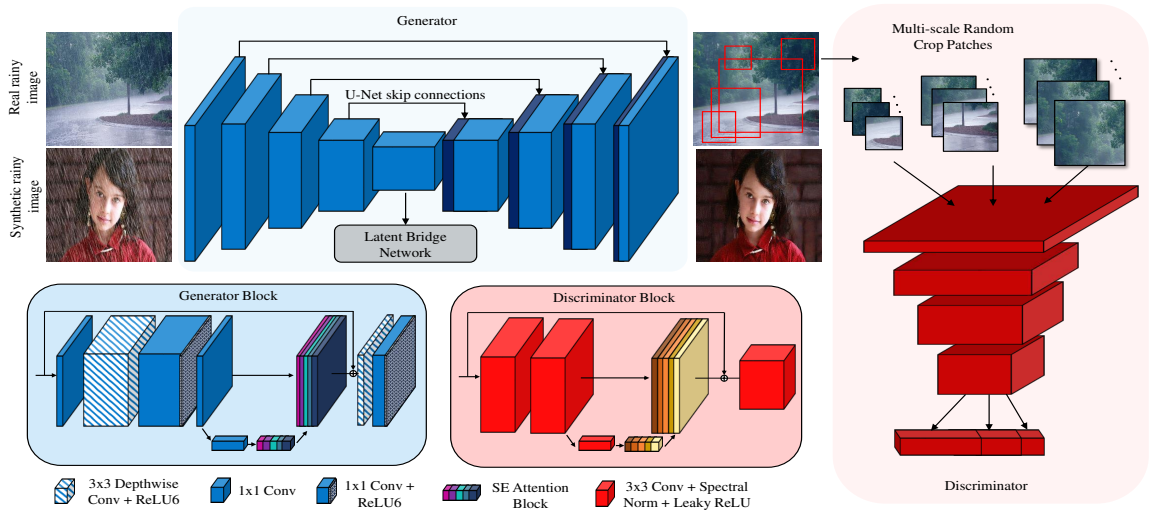

Figure 2. Overview of the MobileDeRainGAN network. Synthetic and real rainy images are passed to the generator to remove rain-related artifacts. The resulting latent spaces are bridged using a latent bridge network for domain adaptation. The multi-scale discriminator incorporates unlabeled data and improves image quality at multiple scales with priority to smaller scales due to the nature of the problem.

image-based methods rely on exploiting the relationship between rain and rain-free image pairs. Due to the problem's nature, most single-image-based methods use synthetically generated data for training and have achieved excellent results. However, due to the inherent domain gap, they tend to perform sub-optimally during testing using real-world images. Recently, semi-supervised approaches have been proposed to deal with the problem by incorporating unlabeled real data into the training phase. Wei et al. [41] proposed SIRR, a semi-supervised transfer learning approach to eliminate the bias-to-supervised-sample issue. Inspired by the work, Yasarla et al. [46] proposed Syn2Real for semi-supervised transfer learning using Gaussian processes. Yue et al. [47] proposed a video-based semi-supervised method to generate realistic rain by training a dynamical rain generator on unlabeled real data.

3. Proposed Method

Each sample in the dataset consists of two pairs of images, referred to as the labeled and unlabeled image pairs, denoted by I_L and I_{UL}, respectively. I_L and I_{UL} are from two different domains, typically synthetic and real-world rain images. The labeled and unlabeled image pairs are denoted as $I_L = \{I_L^R, I_L^C\}$ and $I_{UL} = \{I_{UL}^R, I_{UL}^C\}$, respectively. The labeled rainy image I_L^R, has a one-to-one correspondence with the labeled clean image I_L^C, while this correspondence doesn't hold for the unlabeled image pairs. Note that I_{UL}^C can be substituted by I_L^C. However, having I_{UL}^C from a similar distribution as I_{UL}^R improves the stability of adversarial training. The overview of MobileDeRainGAN shown in Fig. 2 consists of a conditional generative adversarial network. The generator is an image-to-image translational network that removes the artifacts caused by rain. The discriminator helps to improve the quality and minimize artifacts in an unsupervised setting. We also introduce a new network referred to as the latent bridge network to generalize rain removal between the two domains. The following sections describe each of these components in detail.

3.1. Generator

The generator network is a CNN-based encoder-decoder architecture to facilitate image-to-image translation. The network uses a UNet backbone that utilizes skip connections for long-range information sharing. Each block in the network comprises the following three subblocks: An inverted residual block, a depth-wise separable convolution layer for changing the channel depth and a squeeze and excitation block for channel attention. MobileNetV2 [38] introduced inverted residual blocks with a residual connection between the thin bottleneck layers to minimize computation. A squeeze and excitation block introduces channel attention in the inverted residual block. Compared to other forms of attention, channel attention introduces the least amount of overhead in memory and speed, making it suitable for real-time applications. The computation required can be adjusted by varying the expansion ratio in the inverted residual block. No normalization layers are used in the entire generator network, as it was found to hurt performance severely. We show in the later sections that it is important to treat the labeled and unlabeled image pairs differently to preserve the semantic information in the non-rainy

background. Normalization layers prevent this by reducing the internal covariate shift and rescaling feature maps to fit both distributions. The equations governing the generator:

$$[Z_L,\ Z_{UL}] = [G^{enc}(I_L^R),\ G^{enc}(I_{UL}^R)] \tag{1}$$

$$[I_L^{DR},\ I_{UL}^{DR}] = [G^{dec}(Z_L),\ G^{dec}(Z_{UL})] \tag{2}$$

where G^{enc} and G^{dec} are the encoder and decoder of the generator. Z_L and Z_{UL} are the latent spaces corresponding to labeled and unlabeled rainy images. I_L^{DR} and I_{UL}^{DR} are the derained images corresponding to labeled and unlabeled rainy images.

3.2. Discriminator

The generator, as a stand-alone, can only be trained with the labeled data using a supervised loss function. However, with the discriminator, the unpaired image data can be utilized in an unsupervised setting. Like PatchGAN [14], the discriminative network is fully convolutional to suit varying input sizes. Iizuka et al. [13] proposed GLCIC and introduced context discriminators for using global and local discriminators to improve image inpainting significantly. The global discriminator assesses the quality of the synthesized image as a whole, while the local discriminator improves local consistency. Unlike GLCIC, which uses separate discriminators, this work utilizes a single discriminator that analyzes the image on multiple scales. The inspiration for this comes from the human visual cortex, which does not use different modules for perception [20, 29]. Rather, using a single network can benefit from the flow of complementary information to assess image quality at different scales. As shown in Fig. 2, the discriminator uses randomly cropped patches of varying scales. We define a set C consisting of n_c crop ratios, where n_c refers to the number of scales of the multi-scale discriminator.

$$C\ =\ \{c_i\}_{i=1}^{n_c} \tag{3}$$

$$(h_i, w_i) = \left(\frac{H}{c_i},\ \frac{W}{c_i}\right) \forall\ i \in [i, n_c] \tag{4}$$

where h, w and H, W are the heights and widths of the cropped patches and the input image, respectively. The following equations govern the output of the discriminator:

$$P_i(I)\ =\ RandPatches(h_i,\ w_i,\ c_i) \tag{5}$$

$$\varphi_i(I)\ =\ AdptPool(D(P_i(I))) \tag{6}$$

$$\varphi(I)\ =\ \varphi_1(I) \oplus \varphi_2(I) \oplus \cdots \varphi_{n_c}(I) \tag{7}$$

The patch $P_i(I) \rightarrow \mathbb{R}^{c_i \times 3 \times h_i \times w_i}$ is a batch of c_i (from Eq. (4)) randomly cropped RGB patches of size $3 \times h_i \times w_i$ each, from the image I. The vector $\varphi_i(I) \rightarrow \mathbb{R}^{c_i \times 1 \times 1 \times 1}$ is the intermediate output obtained by adaptive average pooling the output when the patch $P_i(I)$ is passed to discriminator D. The final output $\varphi(I)$, is a concatenation of all the intermediate outputs. Thus, the contribution of various patches to the final output is proportional to the crop ratio and inversely proportional to the size of the patch. It is favorable to have more contribution from smaller patches than larger patches as the degradation due to rain occurs more at a smaller scale. Each discriminator block consists of a residual block with channel attention followed by a convolutional layer for increasing channel depth. The discriminator only sees images from the unlabeled image domain; hence we use a normalization layer after each convolutional layer. Spectral normalization [31] is chosen as it improves the training stability of GANs by alleviating vanishing and exploding gradients [25].

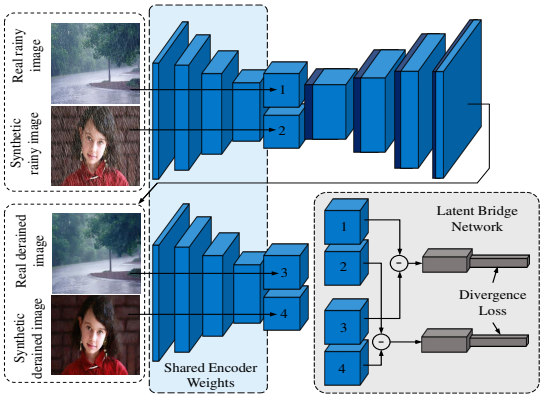

Figure 3. Configuration of the latent bridge network. The latent spaces corresponding to rainy and derained images are subtracted to obtain a rain-related latent space. The divergence between rain-related latent spaces of labeled and unlabeled data are minimized to achieve domain adaptation.

3.3. Latent Bridge Network

As shown in Fig. 1, synthetic and real rain have widely different characteristics. Thus, a deraining algorithm performing well on synthetic data might often fail to remove rain from real-world images. To overcome this, a latent bridge network is introduced. The goal of this bridge network is to force the encoder of the generator to be indifferent to real and synthetic rain. Encoder-decoder networks use the latent space as a bottleneck to condense useful information [2]. A rain removal network encoder-decoder network thus encodes rain features in its latent space [23]. So, an ideal encoder should be able to distill real and synthetic images to produce latent spaces that belong to the same distribution. Achieving this would generalize rain removal on real images guided by synthetic rain removal in a semi-supervised setting. One way to enforce this shared distri-

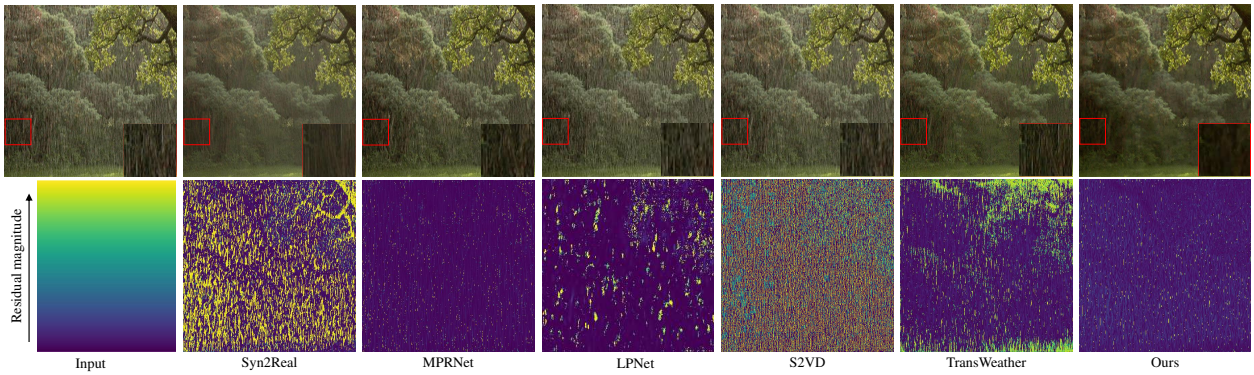

Figure 4. Derained real-world images (top) and their residuals (bottom). Most methods overcompensate rain removal by affecting the non-rainy background layer of the image. This subtle modification mostly invisible to the naked eye leads to a drop in task-driven metrics compared to the rainy image. Red boxes show zoomed view to compare rain removal.

bution could be to minimize a loss like Kullback–Leibler divergence [39] between the two latent spaces. However, initial experiments showed that this hugely affects deraining performance, especially if the initial distribution of synthetic and real images are very different. Fig. 3 shows a workaround by using the difference between latent spaces of rainy and derained images to target parts of the latent space that contribute the most in encoding rain. These differences between synthetic and real images are used to minimize Kullback–Leibler divergence loss.

3.4. Loss

The overall loss of the network is a weighted sum of two supervised and three unsupervised losses. The supervised losses directly guide the network to minimize L1 loss and perceptual quality loss, given the synthetic input and ground truth image pair. The unsupervised losses facilitate the incorporation of unpaired real data. The losses include adversarial loss, bridge loss and consistency loss.

$$\begin{aligned}\mathcal{L}_{total} &= \mathcal{L}_{L1} + \lambda_{percep}\mathcal{L}_{percep} \\ &+ \lambda_{adv}\mathcal{L}_{adv} + \lambda_{bridge}\mathcal{L}_{bridge} \\ &+ \lambda_{const}\mathcal{L}_{const} \end{aligned} \tag{8}$$

Consistent with existing works, L1 loss performed better than L2 loss which caused blurring artifacts. The pretrained VGG16 network on ImageNet is used as a feature extractor at layers 3, 8 and 15 to minimize the perceptual loss on synthetic data. Eq. (9) and Eq. (10) show the formulation of both the supervised losses.

$$\mathcal{L}_{L1} = \left| I_L^{DR} - I_L^C \right| \tag{9}$$

$$\mathcal{L}_{percep} = \sum_{i=3,8,15} \left| VGG_i(I_L^{DR}) - VGG_i(I_L^C) \right|^2 \tag{10}$$

Adversarial loss helps to utilize the unpaired real image dataset to make the generator synthesize better quality derained images. The bridge loss uses the difference between the rainy and derained latent spaces of synthetic and real images to enforce a constraint on feature extraction by the encoder. Finally, consistency loss restricts changes to clean and derained images when passed into the generator to minimize undesirable modifications to derained images. Eq. (11) - Eq. (13) show the formulation of all the unsupervised losses.

$$\begin{aligned}\mathcal{L}_{adv} &= \min_G \max_D \Big[log(\varphi(I_{UL}^C)) \\ &+ log(1 - \varphi(I_{UL}^{DR})) \Big] \end{aligned} \tag{11}$$

$$\begin{aligned}\mathcal{L}_{bridge} &= \Big[Z_L - G^{enc}(I_L^{DR}) \Big] \\ &\Big[log\Big(Z_L - G^{enc}(I_L^{DR}) \Big) \\ &- log\Big(Z_{UL} - G^{enc}(I_{UL}^{DR}) \Big) \Big] \end{aligned} \tag{12}$$

$$\begin{aligned}\mathcal{L}_{const} &= \Big| G^{dec}\Big(G^{enc}(I_{UL}^{DR} \oplus I_{UL}^C) \Big) \\ &- (I_{UL}^{DR} \oplus I_{UL}^C) \Big| \end{aligned} \tag{13}$$

4. Experiments and Results

4.1. Datasets

Hu et al. [12] introduced the RainCityscapes dataset consisting of 10,620 images by modeling synthetic rain by studying the effect of rainfall based on scene depth to improve realism. Unlike traditional synthetic rain datasets, which contain just a few types of rain streaks usually added using Photoshop [33], RainCityscapes use depth information in the Cityscapes dataset [5] to add fog-like layers causing varying object visibility with depth from the camera. To show the effectiveness of our method using cross-domain experiments, we also use the Rain1400 synthetic

Type	Dataset/Model	Venue	PSNR (↑)	SSIM (↑)	mAP (↑)	Inference (fps) (↑)
	Clean [5]	CVPR16	∞	1.00	13.81	-
	Rainy [12]	CVPR19	16.94	0.81	9.23	-
Fully Supervised	UMRL [45]	CVPR19	16.26	0.76	7.23	4.67
	Yang et al. [42]	CVPR19	16.90	0.68	9.60	1.64
	LPNet [8]	TNNLS20	17.30	0.81	9.81	**22.47**
	MPRNet [48]	CVPR21	17.19	**0.83**	8.36	2.17
	TransWeather [15]	CVPR22	19.73	**0.87**	**10.52**	15.18
Un/Semi-Supervised	CycleGAN [52]	ICCV17	**23.38**	0.79	6.49	6.38
	Syn2Real [46]	CVPR20	19.74	0.79	7.64	5.55
	S2VD [47]	CVPR21	16.96	0.81	9.24	14.43
	Ours	-	**25.98**	**0.87**	**11.53**	**33.29**

Table 1. Quantitative comparison of various methods on the RainCityscapes dataset. PSNR and SSIM metrics quantify the capability of domain adaptation, mAP@0.5 score quantifies task-driven performance and inference speed shows the ability to deploy in real-time systems. **Red** and **Blue** correspond to first and second best results. (↑) indicates higher is better.

Dataset/Model	Venue	mAP (↑)	BRISQUE (↓)
Rainy	-	6.03	27.40
UMRL [45]	CVPR19	5.75	23.29
Yang et al. [42]	CVPR19	5.68	34.28
LPNet [8]	TNNLS20	**6.27**	25.87
MPRNet [48]	CVPR21	5.43	24.00
TransWeather [15]	CVPR22	6.16	23.94
Syn2Real [46]	CVPR20	6.25	**22.61**
S2VD [47]	CVPR21	5.98	25.62
Ours	-	**7.32**	**21.68**

Table 2. Comparison of task-driven (mAP@0.5) and no-reference (BRISQUE) metrics on a small-scale real-world dataset with 100 rainy images collected from the internet. **Red** and **Blue** correspond to first and second best results. (↑) and (↓) indicate higher and lower is better, respectively.

Model	PSNR	SSIM (↑)	mAP (↑)
Generator	18.37	0.80	5.98
+ Semi-supervision (disc.)	24.10	0.84	7.50
+ Channel Attention	24.64	0.83	8.51
+ Consistency Loss	24.93	0.84	8.61
+ Multi-scale Discriminator	25.42	0.85	8.82
+ Latent Bridge Network	25.98	0.87	11.53

Table 3. Ablation Study on the proposed architecture to analyze contribution of the key components of the network. Each row adds a new component to the model along with the ones in the rows above. Last row is the full MobileDeRainGAN architecture.

dataset, which is of similar size but with a completely different model of rain and image distribution. Apart from the increased complexity of the rain model, RainCityscapes dataset also provides the advantage of using bounding box labels from the CityPersons dataset [51] to evaluate object detection performance after rain removal. Fu et al. [7] introduced Rain1400 containing 14,000 images using 14 different types of rain streaks with 1000 images. We also use real-world rainy images collected from stock footage, Google image search and images from [19, 50]. Finally, we manually label bounding boxes for 100 real-world rainy images to quantify deraining performance on real-world rain.

4.2. Implementation Details

The network is implemented using PyTorch [32], trained on an NVIDIA Tesla V100 with a batch size of 4 for 100 epochs. We use a learning rate of 0.0002 with the Adam optimizer [18]. The learning rate is decayed by a factor of 0.8 every 20 epochs. The weights for the supervised and unsupervised losses (Eq. (8)) are $[\lambda_{percep}, \lambda_{adv}, \lambda_{bridge}, \lambda_{const}] = [0.05, 0.05, 0.1, 0.1]$. The set C of crop ratios (Eq. (3)) is $[1, 4, 16]$. Additional details can be found in the supplementary material.

4.3. Cross-Domain Analysis

This paper proposes a setup for quantitative evaluation of semi-supervised methods using cross-domain analysis. We train the model using Rain1400 in a supervised fashion while incorporating RainCityscapes in an unsupervised fashion. As both these datasets are significantly different, rain removal performance on RainCityscapes provides a good quantitative estimate of the model's ability to generalize to a different distribution. The results are tabulated using PSNR, SSIM and mAP@0.5 scores of the YOLOv5 model as metrics. To ensure fairness in testing, the semi-supervised and unsupervised methods are trained using the RainCityscapes dataset without labels, while the supervised methods are evaluated on the pretrained models open-sourced by the authors, as they can't be finetuned on an unlabeled dataset.

We compare the proposed method with several existing methods like UMRL [45], LPNet [8], MPRNet [48], Yang et al. [42], TransWeather [15], CycleGAN [52], Syn2Real [46] and S2VD [47]. The results in Tab. 1 show that the proposed method outperforms the existing methods by a signif-

Figure 5. Sample qualitative results on real-world images. Red boxes show zoomed in patches for better comparison.

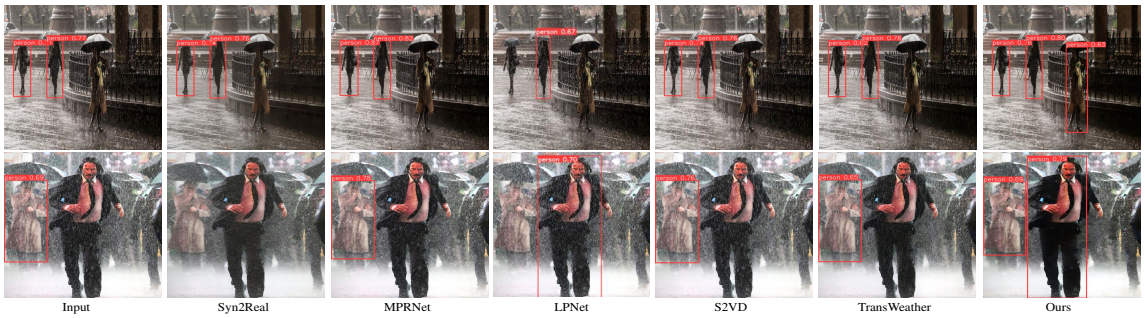

Figure 6. Sample results of object detection using YOLOv5 model after deraining using various algorithms. We show that even in the case of partial deraining during heavy rainfall (bottom row), the detection confidence increases.

icant margin. Note that some works like TransWeather use a diverse dataset, including different weather patterns, yet have a significant performance drop on unseen unlabeled datasets where finetuning is infeasible. Consistent with the observations in [19] and [35], there is a drop or only minor improvement in the mAP score after deraining using most of the existing works. Using deraining as a preprocessing step for autonomous navigation and surveillance applications becomes impractical when rainy images have a better mAP score. Note that for CycleGAN, the mAP score is the lowest despite having the second highest PSNR metric. Thus, conventional image quality metrics used to report performance in the literature aren't necessarily a good indicator of deraining performance for application-based goals. Fig. 4 shows the residuals from deraining for various methods. The residuals show that the semantic structure of the non-rainy background is affected which is subtle to the naked eye, but hurts the performance of task-driven metrics more than what deraining can compensate.

4.4. Performance and Limitations on Real Rain

The proposed method can achieve domain adaptation from not just one synthetic data to another but also from synthetic data to real data. To support this argument, we train the model using a combination of Rain1400 and RainCityscapes as the labeled dataset and the collected real-world rainy images as the unlabeled dataset. Fig. 5 illustrates the qualitative results of deraining on real-world rainy images. It can be seen that the proposed method outperforms the other methods by producing results with better visual quality. Fig. 6 shows the results of object detection after deraining using YOLOv5. The bottom row of Fig. 6 is a challenging real-world scenario with heavy fog and rain obstructing the objects of interest. Even though the all models under consideration only partially derain in this case, including ours, we can observe that the object detection model has a relatively higher confidence. We also quantitatively compare the performance in terms of a no-reference metric BRISQUE [30] and object detection mAP score at 0.5 IoU on a small subset of manually labeled images. The results are tabulated in Tab. 2.

5. Discussion

Snow Removal: A robust model must adapt to different scenarios with limited data and training time. To test this, we transfer learn the model trained on RainCityscapes and real-world rain with Snow100K [26]. Snow 100K contains 100K synthetic snow images and 1329 real images. We take 1000 images each from synthetic and real snow data and transfer learn the model for 10 epochs. Even with a relatively small dataset and training period the model generalizes quickly to the new scenario. Synthetic desnowing resulted in a PSNR score of 26.8 dB PSNR. While state-of-

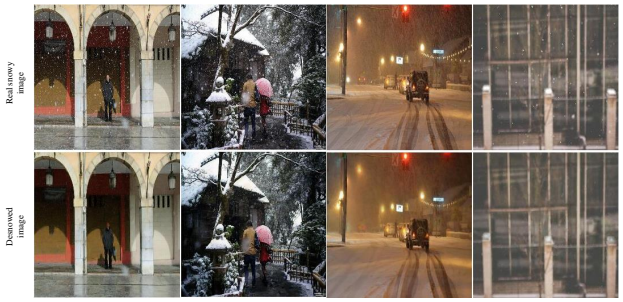

Figure 7. Sample qualitative results of densowing on real snowy images. This demonstrates the robustness of the model as it adapts to the new weather conditions with less training data and time using transfer learning.

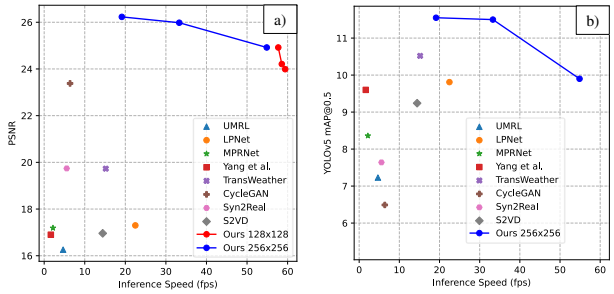

Figure 8. Performance curves of MobileDeRainGAN vs other methods. For our networks, we use expansion ratios 3, 6 and 12 for all resolutions. a) Inference speed vs PSNR score. b) Inference speed vs mAP@0.5 for YOLOv5.

the-art models have surpassed this score, we must note that the model just used 1% of the synthetic dataset. Yet, the model removes snow from real snowy images in a robust manner. Fig. 7 shows sample qualitative results on desnowing real snowy images. The supplementary video attached shows improvement in object detection after desnowing in a real-world video captured from a surveillance camera.

Ablation Study: An ablation study is conducted to understand the contributions from the major components of the architecture. Similar to the quantitative analysis section, we use Rain1400 as the labeled dataset and RainCityscapes as the unlabeled dataset. First, we use a baseline generator model trained on Rain1400 and evaluate on RainCityscapes. Without incorporating the RainCityscapes data in a semi-supervised setting, the results are poor. Then we use the baseline generator-discriminator model. Then we add SE block for channel attention at the end of each generator and discriminator block. We then add consistency loss during the training phase with a coefficient of 0.1. Then we introduce a multi-scale discriminator looking at patches of size 16, 64 and 256. Finally, we introduce the latent bridge network into the generator architecture for domain adaptation. This final configuration is the MobileDeRainGAN architecture used during the analysis in the previous sections. The performance metrics are tabulated in Tab. 3 and show the significance of each component used. Note that, while all components slightly improve PSNR/SSIM metrics, the latent bridge network significantly improves the task-driven metric contributing to the novelty of the model.

Inference Times: Facilitating autonomous navigation in real-time during rainfall requires deraining as a preprocessing step. To this end, we evaluate the feed-forward time for deraining using various models on an NVIDIA Jetson AGX Xavier. The calculation excludes any data loading time and only uses the feed-forward time for a 256x256 image. The results tabulated in Tab. 1 show that the proposed method significantly outperforms the existing methods. Note, that this speed is for the standard model and a trade off between speed and performance can produce faster variants.

Performance Analysis: Achieving a tradeoff between speed and performance metrics can be useful in choosing a model variant based on the hardware used for deployment. Fig. 8 shows the performance of different variants of the proposed model against other works. For a given resolution of input image, three variants of the model are obtained by varying the expansion ratio as 3, 6 and 12. The figure also acts as an ablation for choosing expansion ratio as 6 and input image resolution as 256x256 as it achieves real-time inference while maximizing the performance metrics. Note that for fair , part b) of the figure does not include images of resolution 128x128 due to significant drop in mAP caused by resolution drop.

6. Conclusion

In this paper, we introduced MobileDeRainGAN, a semi-supervised domain adaptation method that removes rain from real images with significantly better results than existing state-of-the-art methods while using a fraction of the computational resources of existing methods. Our approach is based on a novel latent bridge network and multi-scale discriminator that builds off of existing semi-supervised and GAN-based works to tackle the issue of over and under deraining to achieve better task-driven metrics. Being significantly faster than existing methods allows us to trade-off between speed and performance based on hardware. Additionally, our experiments demonstrate that our approach can also be generalized for snow removal. Overall, our work has important implications for real-world applications like autonomous navigation, that require high-quality image processing in real-time under adverse weather conditions. We believe that our findings will inspire further research in this area, leading to the development of more practical and effective approaches for image restoration in challenging environments.

References

[1] Codruta O. Ancuti, Cosmin Ancuti, Radu Timofte, and Christophe De Vleeschouwer. O-haze: A dehazing benchmark with real hazy and haze-free outdoor images. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 867–8678, 2018. 1

[2] Chenghao Chen and Hao Li. Robust representation learning with feedback for single image deraining. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021. 4

[3] Jie Chen, Cheen-Hau Tan, Junhui Hou, Lap-Pui Chau, and He Li. Robust video content alignment and compensation for rain removal in a cnn framework. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6286–6295, 2018. 2

[4] Jaewoong Choi, Dae Ha Kim, Sanghyuk Lee, Sang Hyuk Lee, and Byung Cheol Song. Synthesized rain images for deraining algorithms. *Neurocomputing*, 492:421–439, 2022. 1

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016. 5, 6

[6] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing*, 26(6):2944–2956, 2017. 2

[7] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1715–1723, 2017. 1, 2, 6

[8] Xueyang Fu, Borong Liang, Yue Huang, Xinghao Ding, and John Paisley. Lightweight pyramid networks for image deraining. *IEEE transactions on neural networks and learning systems*, 31(6):1794–1807, 2020. 2, 6

[9] K. Garg and S.K. Nayar. Detection and removal of rain from videos. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I, 2004. 2

[10] Kshitiz Garg and Shree K. Nayar. Vision and rain. *Int. J. Comput. Vis.*, 75(1):3–27, 2007. 2

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. 2

[12] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. Depth-attentional features for single-image rain removal. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8014–8023, 2019. 5, 6

[13] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Trans. Graph.*, 36(4), jul 2017. 4

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017. 4

[15] Jeya Maria Jose Valanarasu, Rajeev Yasarla, and Vishal M. Patel. Transweather: Transformer-based restoration of images degraded by adverse weather conditions. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2343–2353, 2022. 6

[16] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Transactions on Image Processing*, 21(4):1742–1755, 2012. 2

[17] Jin-Hwan Kim, Chul Lee, Jae-Young Sim, and Chang-Su Kim. Single-image deraining using an adaptive nonlocal means filter. In *2013 IEEE International Conference on Image Processing*, pages 914–917, 2013. 2

[18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. 6

[19] 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. Single image deraining: A comprehensive benchmark analysis. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3833–3842, 2019. 6, 7

[20] Wu Li, Valentin Piëch, and Charles D. Gilbert. Perceptual learning and top-down influences in primary visual cortex. *Nature neuroscience*, 7(6):651–657, 2004. 4

[21] Xuelong Li, Kai Kou, and Bin Zhao. Weather GAN: multi-domain weather translation using generative adversarial networks. *CoRR*, abs/2103.05422, 2021. 2

[22] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *ECCV (7)*, volume 11211 of *Lecture Notes in Computer Science*, pages 262–277. Springer, 2018. 2

[23] Yizhou Li, Yusuke Monno, and Masatoshi Okutomi. Single image deraining network with rain embedding consistency and layered lstm. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2022. 4

[24] Yu Li, Robby T. Tan, Xiaojie Guo, Jiangbo Lu, and Michael S. Brown. Rain streak removal using layer priors. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2736–2744, 2016. 2

[25] Zinan Lin, Vyas Sekar, and Giulia Fanti. Why spectral normalization stabilizes gans: Analysis and improvements, 2020. 4

[26] Yun-Fu Liu, Da-Wei Jaw, Shih-Chia Huang, and Jenq-Neng Hwang. Desnownet: Context-aware deep network for snow removal. *IEEE Transactions on Image Processing*, 27(6):3064–3073, 2018. 7

[27] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3397–3405, 2015. 2

[28] Takuro Matsui and Masaaki Ikehara. Gan-based rain noise removal from single-image considering rain composite models. *IEEE Access*, 8:40892–40900, 2020. 2

[29] Justin N. J. McManus, Wu Li, and Charles D. Gilbert. Adaptive shape processing in primary visual cortex. *Proceedings of the National Academy of Sciences*, 108(24):9739–9746, 2011. 4

[30] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012. 7

[31] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks, 2018. 4

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019. 6

[33] Steve Patterson. Adding rain to a photo with photoshop, Oct 2012. 5

[34] Rui Qian, Robby T. Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2482–2491, 2018. 1

[35] Shyam Nandan Rai, Rohit Saluja, Chetan Arora, Vineeth N Balasubramanian, Anbumani Subramanian, and C. V. Jawahar. Fluid: Few-shot self-supervised image deraining. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 418–427, 2022. 2, 7

[36] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 1

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. 1

[38] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. 2, 3

[39] Jonathon Shlens. Notes on kullback-leibler divergence and likelihood, 2014. 5

[40] 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 *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12262–12271, 2019. 1

[41] Wei Wei, Deyu Meng, Qian Zhao, Zongben Xu, and Ying Wu. Semi-supervised transfer learning for image rain removal. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3872–3881, 2019. 3

[42] Wenhan Yang, Jiaying Liu, and Jiashi Feng. Frame-consistent recurrent video deraining with dual-level flow. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1661–1670, 2019. 1, 6

[43] 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 Transactions on Pattern Analysis and Machine Intelligence*, 42(6):1377–1393, 2020. 2

[44] Wenhan Yang, Robby T. Tan, Shiqi Wang, and Jiaying Liu. Self-learning video rain streak removal: When cyclic consistency meets temporal correspondence. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1717–1726, 2020. 2

[45] Rajeev Yasarla and Vishal M. Patel. Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image de-raining. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019. 6

[46] Rajeev Yasarla, Vishwanath A. Sindagi, and Vishal M. Patel. Syn2real transfer learning for image deraining using gaussian processes. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2723–2733, 2020. 3, 6

[47] Zongsheng Yue, Jianwen Xie, Qian Zhao, and Deyu Meng. Semi-supervised video deraining with dynamical rain generator. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 642–652, 2021. 3, 6

[48] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14821–14831, June 2021. 6

[49] He Zhang and Vishal M. Patel. Density-aware single image de-raining using a multi-stream dense network. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 695–704, 2018. 2

[50] 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*, 30(11):3943–3956, 2020. 2, 6

[51] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Citypersons: A diverse dataset for pedestrian detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4457–4465, 2017. 6

[52] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017. 6