This WACV 2022 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

FLUID: Few-Shot Self-Supervised Image Deraining

Shyam Nandan Rai[1] Rohit Saluja[1] Chetan Arora[3] Vineeth N Balasubramanian[2]
Anbumani Subramanian[1] C.V. Jawahar[1]
[1]CVIT - IIIT Hyderabad, India [2]IIT Hyderabad, India [3]IIT Delhi, India
[1]shyamnandanraii@gmail.com, [1]rohit.saluja@research.iiit.ac.in, [2]vineethnb@iith.ac.in,
[3]chetan@cse.iitd.ac.in, [1]{anbumani, jawahar}@iiit.ac.in

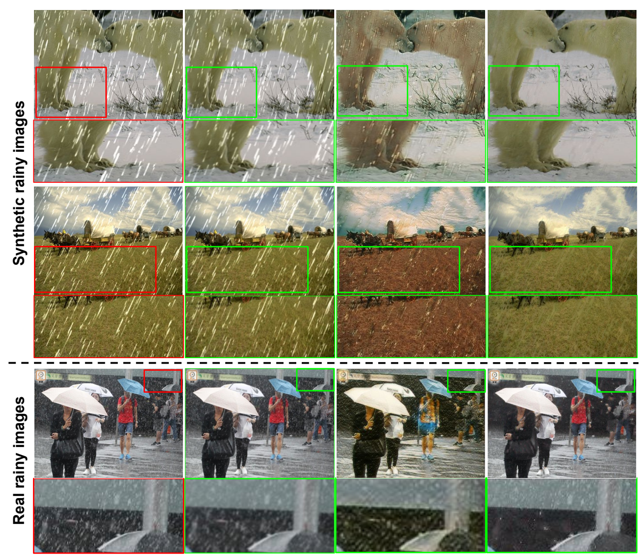
Abstract

Self-supervised methods have shown promising results in denoising and dehazing tasks, where the collection of the paired dataset is challenging and expensive. However, we find that these methods fail to remove the rain streaks when applied for image deraining tasks. The method's poor performance is due to the explicit assumptions: (i) the distribution of noise or haze is uniform and (ii) the value of a noisy or hazy pixel is independent of its neighbors. The rainy pixels are non-uniformly distributed, and it is not necessarily dependant on its neighboring pixels. Hence, we conclude that the self-supervised method needs to have some prior knowledge about rain distribution to perform the deraining task. To provide this knowledge, we hypothesize a network trained with minimal supervision to estimate the likelihood of rainy pixels. This leads us to our proposed method called FLUID: Few Shot Self-Supervised Image Deraining.

We perform extensive experiments and comparisons with existing image deraining and few-shot image-to-image translation methods on Rain 100L and DDN-SIRR datasets containing real and synthetic rainy images. In addition, we use the Rainy Cityscapes dataset to show that our method trained in a few-shot setting can improve semantic segmentation and object detection in rainy conditions. Our approach obtains a mIoU gain of 51.20 *over the current best-performing deraining method. [Project Page]*

1. Introduction

Deep learning models require large-scale datasets to learn a computer vision task. Applications such as autonomous navigation systems use many paired images to generalize across different adverse weather conditions. The collection of such a dataset is an expensive and tedious task. Self-supervised methods [16] were introduced, which gave marginal performance behind the supervised methods on various downstream tasks while avoiding the dependency on large-scale labeled datasets.



Input Image N2S [47] Yasarla *et al*. [50] **FLUID**

Figure 1. Image deraining results by the self-supervised method: N2S [47] (col. 2), semi-supervised method: Yasarla *et al*. [50] (col. 3), and our method: FLUID (col. 4). We can observe N2S [47] fail to remove rainy streaks due to poor prior knowledge about rain distribution, whereas, Yasarla *et al*. [50] suffer from image artifacts due to its sensitivity to the training sample choice. For a fair comparison, we train all the baselines methods in a few-shot unsupervised setting.

Recently, self-supervised methods have shown good results for image denoising [1, 19, 47] and image dehazing [21]. However, these methods explicitly mention that the following assumptions are made to perform denoising or dehazing tasks: *(a) noise or haze is uniformly distributed across an image, and (b) a noisy or hazy pixel value is independent of its neighboring pixels.* Applying such self-supervised techniques for image deraining gives poor results (Figure 1: col. 2) because a rainy pixel might/might not depends on nearby pixels, and rain pixels are non-uniformly distributed in the image, unlike haze and noise. This concludes that self-supervised techniques require prior knowledge about rain distribution to perform image deraining tasks. To provide this knowledge to the self-supervised network, we hypothesize a network trained with minimal supervision to estimate the likelihood of rainy pixels. This leads us to the proposed method: Few-Shot Self- Supervised Image Deraining (FLUID). Figure 1 shows the FLUID deraining results in comparison with self-supervised and semi-supervised methods on real and synthetic images.

The FLUID framework consists of three stages. In the first stage, we train a Probability Estimation Network (PEN) trained in a few-shot setting that predicts the pixel-wise rain likelihood in an image. The trained PEN network helps to provide prior knowledge about rain distribution. In PEN, we predict pixel-wise rain likelihood instead of learning non-rainy pixels. The network can learn to predict rainy pixels independent of textural information present in training images. Recent semi-supervised deraining method [46, 50] performs poorly when trained in a few-shot unsupervised setting. This is because the objective function minimizes the loss between the rainy and clean image pair, enabling to learn the textural image information. This claim becomes evident in Figure 1: col. 3 where we compare our method's performance with the semi-supervised method [50]. This brings us to the conclusion that semi-supervised methods are sensitive to the choice of training samples which is evident from the color shift caused by the choice of the training image.

In the next stage, we use the trained PEN to predict the pixel-wise rain probability that helps identify and mask the rainy regions in the images. We then fill the masked area using image inpainting. The inpainted output acts as a prior to a Self-Supervised Network (SSN). In the last stage, we pass the inpainted output to the SSN. With sufficient prior knowledge about the rain distribution, the SSN can further derain the image and remove image artifacts and blurriness introduced by image inpainting. The efficacy of our proposed model is evaluated on Rain 100L and DDN-SIRR having real and synthetic rainy images. We show extensive qualitative and quantitative comparison with image deraining methods and few-shot image-to-image translation methods. Further, the FLUID framework was also employed in improving downstream tasks: semantic segmentation and object detection.

In summary, our key contributions are as follows:

- This is the first data-driven image deraining method in a few-shot setting to the best of our knowledge.
- Train a Probability Estimation Network that estimates the pixel-wise likelihood of rain. The output of trained PEN provides prior knowledge about rain distribution to a Self-Supervised Network.
- Perform extensive experiments on multiple datasets containing natural and synthetic images to show our method's deraining ability. Our ablation study establishes that our method's performance is consistent irrespective of the choice of the training samples.
- Demonstrates that using derained images from the FLUID framework significantly improves semantic segmentation and object detection compared to existing deraining approaches.

2. Related Works

Single Image Deraining: Single image deraining [49] is the task of generating rain-free images that have been extensively researched over the past few decades. There are also video-based deraining techniques [23, 40, 54], but single image deraining is more challenging due to temporal information's unavailability. We can divide all the single deraining methods into two categories: *model-based* and *deep-learning based* methods.

Model-based methods or non-deep learning methods utilize dictionary learning [4, 29], prior-based [56], sparsity-based model [7, 45], and mixture-model based [26] to get the derained images. However, the methods mentioned above struggle to generalize over variations in rainy streaks. Recently, deep-learning models have shown state-of-the-art performance in various computer vision tasks due to efficient feature learning. Leveraging the advantage of deep learning models, Yang *et al*. [48] proposed a deep network that can detect and remove rain. Later, new approaches were proposed, which were based on Convolutional Neural Network (CNN) [25, 51], generative models [52], and physics-driven models [24].

However, the methods mentioned earlier tend to fail when tested on real rainy images. Wei *et al*. [46] proposed an efficient semi-supervised approach that used synthetic rainy pair images and unlabeled real rainy images. This approach, without proper initialization, will lead to sub-optimal results [50]. Yasarla *et al*. [50] presented an improved semi-supervised method that used the Gaussian process to leverage the information from unlabeled real rainy images while training. However, these methods perform poorly in few-shot unsupervised settings as they are sensitive to the training image pairs.

Few-Shot Image-to-Image Translation: Few-shot learning for image classification [10, 32] is a widely studied problem. Recently, Liu *et al*. [28] proposed a method to generate images of unseen classes with only a few samples provided at the testing phase. Later, other few-shot generation methods were proposed for face reenactment [12], interactive video stylization [43], and font style transfer [22]. However, we find that when trained in a few-shot unsupervised setting, the few-shot methods

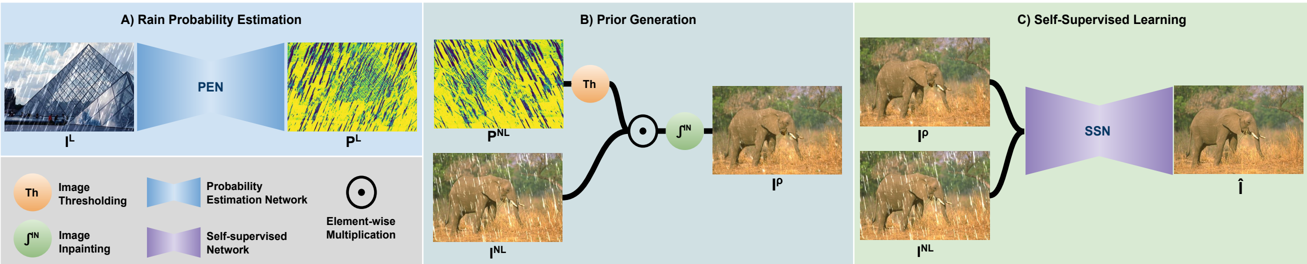


Figure 2. **Overview of FLUID framework:** The FLUID framework consists of three stages. A) *Rain Probability Estimation:* We train a Probability Estimation Network (PEN) that predicts the pixel-wise rain probability of an image. The trained PEN network helps in the generation of prior knowledge for the Self-Supervised Network (SSN). B) *Prior Generation:* We pass unpaired rainy images I^{NL} to estimate pixel-wise rain probability in this stage. The predicted pixel-wise rain probability map values are thresholded Th by giving 0 to rainy pixel and 1 to non-rainy. Then, we perform element-wise multiplication between the I^{NL} and its corresponding thresholded probability map P^{NL}. As a result, the rainy regions are masked out. We then fill the masked area through image inpainting IN. The inpainted output I^{ρ} acts as a prior for SSN. C) *Self-Supervised Learning:* Finally, the generated prior I^{ρ} trains the SSN that refines the results further by minimizing the image artifacts introduced by image inpainting and the tiny rain streaks that are undetected by PEN.

struggle to minimize the artifacts by adverse weather conditions.

Self-Supervised Learning: Self-supervised learning [16] refers to the learning of visual features from the unlabeled dataset. This framework trains a network to solve the pretext task using the pseudo-labels generated from a dataset without human supervision. Doerschet *et al*. [8] proposed the first self-supervised learning method that used a pretext task of predicting image patches' relative position, which improves object detection tasks. Later self-supervised approaches used the pretext tasks such as solving jigsaw puzzles [33], image rotation estimation [18], super-resolution [20], colorization [53], and inpainting [34].

Recent self-supervised denoising methods such as Noise2Void [19], Noise2Self [1], and Noise2Same [47] does not depend on prior noise information for denoising. Although, the availability of noise information further improved the performance. The success of self-supervised models in denoising motivated us to use such frameworks for image deraining.

3. FLUID: Few-Shot Self-Supervised Image Deraining

3.1. Overview

We begin the formulation of the framework by a set of rainy images: $I^L = \{I_i^L : i = 1, 2, \ldots, n\}$ and the corresponding clean images: $I = \{I_i : i = 1, 2, \ldots, n\}$. The value of n in our framework is 1, 3, and 5. The unpaired rainy image set without the clean image is denoted by: $I^{NL} = \{I_i^{NL} : i = 1, 2, \ldots, m\}$, where $m \gg n$. Firstly, we train a Probability Estimation Network (PEN) on I^L and I to get the pixel-wise rain probability of an image. We then use the trained PEN network to get pixel-wise

Figure 3. **Visualization of output at various stages:** Left to Right: *col. 1*: input rainy image, *col. 2*: PEN output, *col. 3*: masking of rain by taking the dot product between rainy image and PEN output, *col. 4*: output obtained from image inpainting, *col. 5*: refined output from Self-Supervised Network.

rain probability estimation of I^{NL} denoted by P^{NL}. Next, we perform element-wise multiplication between the I^{NL} and corresponding P^{NL}. As a result, the rainy regions are masked out and filled through image inpainting. After that, we use the inpainted output as the prior denoted by I^{ρ}. The generated labels I^{ρ} suffer from image artifacts and blurriness due to image inpainting and have tiny rain streaks that are undetected by PEN. Finally, we train a SSN to minimize such image artifacts introduced by image inpainting and further derain the image. Figure 2 illustrates the overview of our method, and Figure 3 shows the output at various stages on the images of the Rain 100L dataset. We now discuss the individual stage of the FLUID framework in detail.

3.2. Probability Estimation Network

Consider a rainy image I_i^L having pixel value at x to be $I_{i(x)}^L$ and the probability of the pixel being rainy is $P_r(x)$. Now, we need to learn a function f_P that estimates the conditional probability of the pixel at location x to be rainy given $I_{i(x)}^L$ which is formulated as:

$$f_P(x) = P_r(x/I_{i(x)}^L) \tag{1}$$

We learn the function $f_P(x)$ by training a UNet [38] to estimate pixel-wise rain probability. We train the UNet on

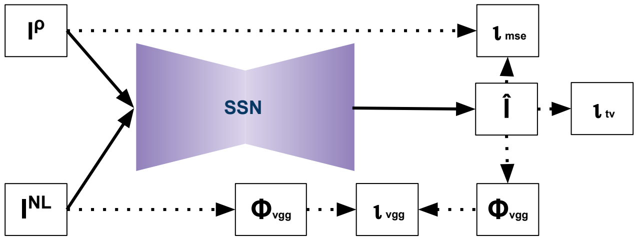


Figure 4. **Self-Supervised Network (SSN):** We train SSN on a combination of three losses that are l_{mse}, l_{tv}, and l_{vgg}. We use l_{mse} to get the average content of priors I^ρ. l_{tv} is then used to minimize the small rainy streaks. We use l_{vgg} to improve the high-frequency information in the output image $\hat{I}$. The dotted line represents the flow of input to a loss function, and the solid line shows the flow of input and output to the SSN.

binary cross-entropy loss which is given by:

$$l_{PEN} = \frac{1}{N}\sum_{x=1}^{N} P^L_{i(x)}.log(f_P(x)) + (1 - P^L_{i(x)}).log(1 - f_P(x)) \tag{2}$$

$P^L_{i(x)}$ represents the given ground truth rain probability of $I^L_{i(x)}$ at location x and N is the total number of pixel. In PEN, we predict pixel-wise rain likelihood instead of learning non-rainy pixels since rainy streaks are mostly textureless. Hence, the trained PEN will predict rainy pixels independent of textural information present in training images. Data augmentation plays a significant role in improving the rain detection capabilities of PEN across various rain patterns shown in Supp. Sec. 1. Figure 3: col. 2 shows the rain streaks predicted by PEN.

3.3. Prior Generation

We now use the trained PEN to generate the priors I^ρ for SSN. We pass the unpaired rainy images I^{NL} through trained PEN to generate pixel-wise rain probability. The output probability map inferred from PEN is thresholded th to 0 for rainy pixels and 1 for rainy pixels, which results in P^{NL}. We then perform element-wise multiplication between I^{NL} and P^{NL} that masks the rainy regions. Figure 3: col. 3 shows the masked image. Now, we fill the masked areas by image inpainting f_{IN} that gives the prior for SSN shown in Figure 3: col. 4. The entire process can be formulated as:

$$I^\rho = f_{IN}(P^{NL} \odot I^{NL}) \tag{3}$$

We used a statistical inpainting method by Damelin *et al.* [6] for the image inpainting task. We did not use pre-trained inpainting network for inpainting as they can give biased results based on the training dataset.

3.4. Self-Supervised Learning

The generated prior have blurry regions introduced by image inpainting and have tiny rainy streaks undetected by PEN. Now, we use SSN to improve the quality of prior and further derained the image to get the final derained image $\hat{I}$. Firstly, we pass I^ρ and I^{NL} through the SSN as an input. Next, we calculate mean square loss l_{mse} between I^ρ and $\hat{I}$ to retrieve the average prior content. Now, we use total variation loss [2] denoted by l_{tv} on $\hat{I}$ to minimize the tiny streaks which are undetected by the PEN. l_{tv} and l_{mse} smoothens the output image which reduces the high frequency detail. Hence, we use VGG loss [20] denoted by l_{vgg} that adds high-frequency details into $\hat{I}$. We calculate the l_{vgg} using features map denoted by ϕ_{vgg} obtained from ReLU activation layers of the pretrained VGG16 [42] network. Figure 4 illustrates the flow of input and output to the SSN along with various training losses. The final objective is given by:

$$l_{SSN}((P^{NL}_i; I^{NL}_i), \hat{I}_i) = l_{mse} + \lambda_1 l_{tv} + \lambda_2 l_{vgg} \tag{4}$$

$$l_{SSN}((P^{NL}_i; I^{NL}_i), \hat{I}_i) = \frac{1}{N}\sum_{x=1}^{N} ||I^\rho_{i(x)} - \hat{I}_{i(x)}||_2 + \frac{\lambda_1}{N}\sum_{x=1}^{N} ||\hat{I}_{i(x)}||_{tv} + \frac{\lambda_2}{M}\sum_{x=1}^{M} ||\phi_{vgg}(I^{NL}_{i(x)}) - \phi_{vgg}(\hat{I}_{i(x)})||_2 \tag{5}$$

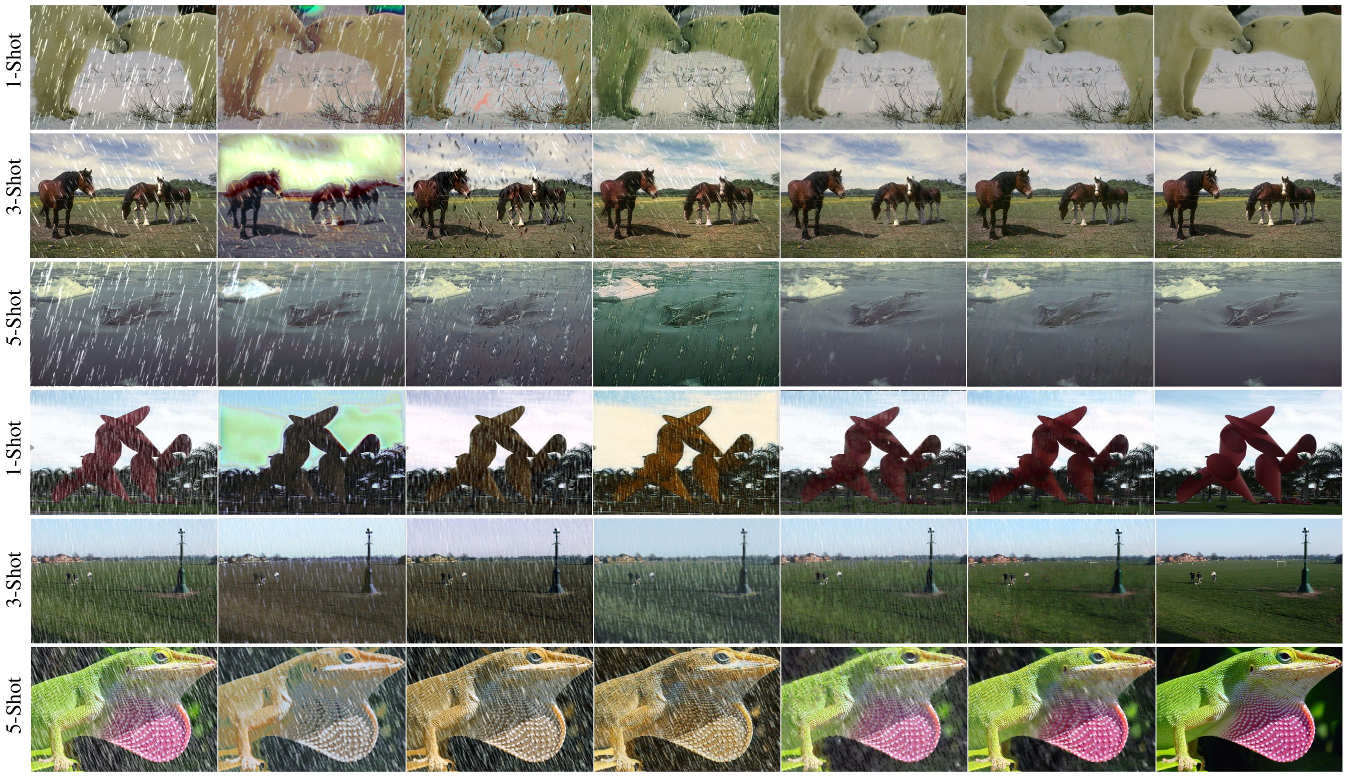
$I^\rho_{i(x)}$ and $\hat{I}_{i(x)}$ represent the value of prior and its corresponding SSN output at pixel x, respectively. M denotes ϕ_{vgg} dimension. λ_1 and λ_2 are the hyperparameter empirically estimated during the network's training. Figure 3: col. 5 shows the SSN output.

4. Experimentation

4.1. Experimental Settings

4.1.1 Dataset and Evaluation

Rain 100L: Yang *et al.* [48] synthesized the dataset using the rain streak rendering method by Garg *et al.* [11] on the clean images of BSD200 [30]. It consists of 200 pairs of training images and 100 pairs of test images. We divide the training image pairs into two parts for our experiments: 5 image pairs for training and 195 image pairs for validation.
DDN-SIRR: This dataset consists of synthetic rainy and rain-free image pairs and unpaired natural rainy images created by Wei *et al.* [46]. The rain-free images were taken from the UCID [41] dataset. We use the synthetic dataset in our few-shot experiments by randomly choosing five image pairs for training and 400 image pairs for validation and testing. Further, we test the trained model on a set of 100 real rainy images having dense and sparse rain streak.
Rainy Cityscapes: Halder *et al.* [13] proposed a physics-based rain rendering method to inject rain into the clean images realistically.Using this method, Halder *et al.* [13] creates a rainy cityscapes dataset consisting of rain and rain-free images of Cityscapes [5]. We use this dataset to show improvement in semantic segmentation. We randomly



Input Rainy Image **ID-CGAN [52]** **Wei *et al*. [46]** **Yasarla *et al*. [50]** **Ours** **Rainy2Clean** **Ground Truth**

Figure 5. **Qualitative deraining results in few-shot setting:** *Row 1-3:* Qualitative results on Rain 100L dataset. *Row 4-6:* Qualitative results on DDN-SIRR dataset.

Input Rainy Image **Ground Truth** **FLUID (Ours)** **CycleGAN [55]** **MUNIT [14]** **FUNIT [28]** **COCO-FUNIT [39]**

Figure 6. **Qualitative comparison with image-to-image translation methods.** More qualitative results in supplementary material.

choose five training image pairs, 300 validation image pairs, and 200 image pairs for the test.

Evaluation Metrics: We use Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) to evaluate the performance of deraining methods for the synthetic datasets as the ground-truth is available. For natural rainy images, we use Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) [31].

4.1.2 Training Details

We train the PEN on twenty thousand epochs with batch size 1. The initial learning rate is $1e-4$, which is reduced to $1e-5$ after ten thousand epochs. We train the SSN for 500 epochs with a learning rate of $1e-3$ and a batch size of 16. While training both the networks, the input is given by randomly cropping 128×128 image patch, which is randomly rotated between $(-180°, 180°)$. The values of λ_1 and λ_2 in Eq 5 were empirically found best to be $1e-3$ and 0.04. We use the value of Th to be 0.95.

4.1.3 Baselines

Since there is no previous few-shot image deraining work, we baseline FLUID performance with a) few-shot/unsupervised/supervised image-to-image translation methods and b) semi/fully supervised deraining methods. We train all the methods in a few-shot unsupervised setting for a fair comparison, i.e., only a few rainy/clean image pairs were given, rest were unpaired during training.

Dataset	RESCAN [25] *(ECCV'18)*	ID-CGAN [52] *(TCSVT'19)*	Wei *et al.* [46] *(CVPR'19)*	Yasarla *et al.* [50] *(CVPR'20)*	**Ours**	Rainy2Clean (Upper-Bound)
Rain 100L [1Shot]						
Val	16.51 / 0.5676	17.64 / 0.6619	21.49 / 0.7117	22.81 / 0.7298	24.31 / 0.8156	-
Test	16.37 / 0.5510	17.01 / 0.6391	20.94 / 0.7021	22.26 / 0.7229	23.87 / 0.7724	27.52 / 0.9180
Rain 100L [3Shot]						
Val	17.32 / 0.5800	17.93 / 0.6707	22.62 / 0.7269	23.01 / 0.7604	25.79 / 0.8317	-
Test	16.91 / 0.5772	17.48 / 0.6544	22.17 / 0.7195	22.42 / 0.7596	25.54 / 0.8260	27.52 / 0.9180
Rain 100L [5 Shot]						
Val	18.07 / 0.6035	19.37 / 0.6965	23.91 / 0.7811	23.97 / 0.7832	26.97 / 0.8643	-
Test	17.44 / 0.5993	18.66 / 0.6821	23.77 / 0.7751	23.59 / 0.7703	26.87 / 0.8615	27.52 / 0.9180
DDN-SIRR [1 Shot]						
Val	14.38 / 0.4631	16.11 / 0.5530	18.73 / 0.6013	19.51 / 0.6313	21.92/ 0.6808	-
Test	11.35 / 0.3173	15.30 / 0.5127	17.26 / 0.5849	19.65 / 0.6512	21.83 / 0.6781	24.13 / 0.7802
DDN-SIRR [3 Shot]						
Val	16.70 / 0.5427	18.69 / 0.5962	19.51 / 0.6257	20.74 / 0.6537	22.23 / 0.6918	-
Test	16.58 / 0.5639	18.13 / 0.6159	19.34 / 0.6381	20.09 / 0.6485	21.97 / 0.6749	24.13 / 0.7802
DDN-SIRR [5 Shot]						
Val	17.72 / 0.5843	19.07 / 0.6287	20.88 / 0.6517	21.08 / 0.6709	22.27 / 0.6992	-
Test	17.34 / 0.5702	18.82 / 0.6119	20.16 / 0.6449	20.84 / 0.6667	22.07 / 0.6841	24.13 / 0.7802

Table 1. **Quantitative comparison (PSNR/SSIM):** Results in blue background shows the performance of our method. Orange and green background shows the results of supervised and semi-supervised methods in few-shot setting. Gray background shows the results when we train the network with all the training samples.

	Method	PSNR / SSIM
Full Supv.	ID [17] *(TIP'12)*	23.13 / 0.70
	CNN [9] *(ICCV'13)*	23.70 / 0.81
	DSC [29] *(ICCV'15)*	24.16 / 0.87
	LP [26] *(CVPR'16)*	25.91 / 0.89
5 Shot	DerainDrop [35] *(CVPR'18)*	15.69 / 0.53
	RESCAN [25] *(ECCV'18)*	17.44 / 0.59
	SPANet [44] *(CVPR'19)*	18.46 / 0.65
	ID-CGAN [52] *(TCSVT'19)*	18.66 / 0.68
	FLUID (Ours)	**26.87 / 0.86**

Table 2. Result comparison of deraining method with FLUID on Rain 100L dataset. The methods above the dotted line are trained on full training dataset and the below are trained in 5-shot unsupervised setting. Supv. denotes supervision.

Method	PSNR / SSIM
UNIT [27] *(NIPS'17)*	7.80 / 0.053
Pix2Pix [15] *(ICCV'17)*	10.73 / 0.14
COCO-FUNIT [39] *(ECCV'20)*	14.80 / 0.35
FUNIT [28] *(ICCV'19)*	15.79 / 0.40
MUNIT [14] *(ECCV'19)*	15.79 / 0.40
CycleGAN [55] *(CVPR'17)*	16.64 / 0.49
FLUID (Ours)	**23.87 / 0.77**

Table 3. Quantitative comparison of FLUID with image-to-image translation methods in 1-shot unsupervised setting.

We compare our framework performance with i) supervised: Pix2Pix [15], ii) unsupervised: UNIT [27], CycleGAN [55], and MUNIT [14], and iii) few-shot: FUNIT [28] and COCO-FUNIT [39] image-to-image translation methods. Next, we baseline FLUID with semi-supervised deraining methods proposed by Wei *et al*. [46] and Yasarla *et al*. [50]. We also train supervised deraining methods ID [17], CNN [9], DSC [29], LP [26], DerainDrop [35], SPANet [44], RESCAN [25], and ID-CGAN [52] as our baselines. We create an upper-bound baseline ***Rainy2Clean*** by training the SSN network with full supervision. Rainy2Clean shows deraining results when we have access to the entire dataset.

4.2. Results

From the results shown in Table 3, we observe that the proposed FLUID framework outperforms the image-to-image translation methods: Pix2Pix [15], UNIT [27], CycleGAN [55], MUNIT [14], FUNIT [28] and COCO-FUNIT [39]. Qualitative results in Figure 6 show that our method can minimize the rain streaks, whereas the baselined methods suffer from image artifacts. Table 2 shows the performance comparison of our method with the supervised methods: ID [17], CNN [9], DSC [29], LP [26], DerainDrop [35], SPANet [44], RESCAN [25], and ID-CGAN [52] trained on Rain 100L dataset. We can observe in Table 2 (row: 1-4, 9), our method trained only on 5-shot setting achieves better PSNR compared to initial deraining methods: ID [17], CNN [9], DSC [29], and LP [26] which were trained on the entire dataset. We also observe in Table 2 (row: 5-9), our model significantly outperforms recent deraining methods: DerainDrop [35], SPANet [44], RESCAN [25], and ID-CGAN [52] in 5-shot setting. Quantitatively, we get $8.21/0.18$ PSNR/SSIM gain over the best-supervised method.

Next, we compare our proposed method with Wei *et al*. [46], Yasarla *et al*. [50], RESCAN [25], ID-CGAN [52], and Rainy2Clean on the test set of Rain 100L and DDN-

Method	BRISQUE Score ↓
Rainy Image	32.28
Yasarla *et al*. [50]	31.93
Ours	30.67
Rainy2Clean	27.89

Table 4. **Performance evaluation on real rainy images (DDN-SSIR).** (↓) indicates lower the score better the performance.

Figure 7. **Real rainy (DDN-SIRR) results in 5-shot setting.** (a) Input rainy image. (b) Results from semi-supervised method: Yasarla *et al*. [50]. (c) Derained output from our proposed method. (d) Results from the method trained on all training samples: Rainy2Clean. We observe that our approach shows performance close to Rainy2Clean. Notably, it works better in removing rainy streaks, as observed in the bottom row results.

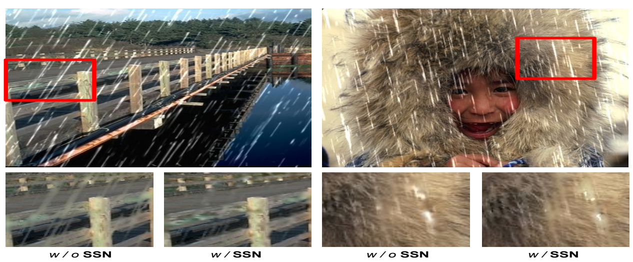


Figure 8. **Deraining performance with and without self-supervised network:** We observe SSN can minimize the image artifacts by inpainting and the rain streaks left undetected by PEN.

SIRR dataset in 1-shot, 3-shot, and 5-shot setting. Figure 5 and Table 1 shows the qualitative and quantitative results. We observe that our model outperforms the other deraining methods in the few-shot settings. We find that semi-supervised methods [46, 50] struggle to remove the rain and cannot retain the input image statistics. This is because of the poor latent representation learned by their supervised networks. The visual results of fully supervised methods [25, 52] are lower than semi-supervised methods as they do not have the choice to improve their latent representation of their model using real rainy images. Figure 7 and Table 4 show our method's performance on real rainy images of the DDN-SIRR dataset trained in the 5-shot setting. We observe in Figure 7, our method acts more effectively in removing the rain streaks than Rainy2Clean.

4.3. Effectiveness of SSN

We investigate the effectiveness of using SSN in our deraining framework by defining various methods with different losses (Section 3). *w/o* SSN: Train without SSN. M^1: Train with SSN on l_{mse}. M^2: Train with SSN on l_{mse} and l_{tv}. *w/* SSN: Train with SSN on l_{mse}, l_{tv}, and l_{vgg}. We trained all the methods on Rain 100L dataset and presented the results in Table 6. We can see *w/* SSN shows the best performance demonstrating the effectiveness of the combination of loss used in SSN. In Figure 8, we observe that *w/* SSN can minimize the image artifacts due to image impainting and the rain streaks that are undetected by PEN.

	Yasarla *et al*. [50]	Ours	Rainy2Clean
Pair 1	19.08	23.87	27.52
Pair 2	20.91	23.62	27.52
Pair 3	16.59	24.07	27.52

Table 5. **Quantitative generalization performance:** Shows the consistent quantitative performance (PSNR) of our method on Rain 100L dataset trained on different training image pair.

Method	Loss	PSNR
w/o SSN	n/a	22.91
M^1	l_{mse}	23.17
M^2	$l_{mse} + l_{tv}$	23.38
w/ SSN	$l_{mse} + l_{tv} + l_{vgg}$	**23.87**

Table 6. **Ablative study on SSN:** Shows incremental improvement by adding SSN with l_{mse}, l_{tv}, and l_{vgg}.

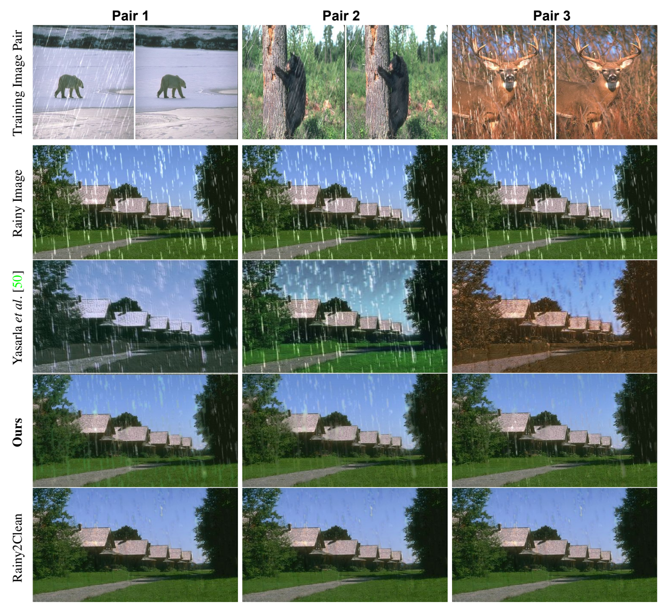


Figure 9. **Qualitative generalization performance:** We show that our method gives a consistent deraining performance on the Rain 100L dataset, irrespective of the training image pair while training in a 1-shot setting. Whereas, Yasarla *et al*. [50] is sensitive to training image pair, which we prominently observed in Pair 3.

4.4. Generalization

We demonstrate our proposed method's performance consistency irrespective of the training pair in a 1-shot set-

Method	road	swalk	build.	wall	fence	pole	tlight	sign	veg.	terrain	sky	person	rider	car	truck	bus	train	mbike	bicycle	mIoU
Rainy Images	83.66	59.67	79.04	27.41	42.76	42.07	48.79	53.09	79.95	68.30	74.68	68.08	48.70	69.38	57.92	65.78	34.06	58.06	47.82	58.38
Yasarla *et al*. [50]	31.99	9.68	42.71	0.12	3.48	1.17	3.93	0.17	38.90	13.19	7.70	12.36	3.17	38.03	5.48	1.92	1.31	7.85	8.76	12.21
Ours	87.34	65.61	82.91	39.80	48.94	46.65	51.60	58.82	83.41	70.53	80.49	69.81	56.24	72.62	63.10	71.04	44.14	59.17	52.63	63.41
Rainy2Clean	96.41	78.01	85.82	64.80	54.86	53.76	55.52	62.88	86.17	79.38	83.90	74.42	59.26	91.29	64.64	77.87	44.92	65.22	57.19	70.33
IoU Gain	55.35	55.93	40.20	39.68	45.46	45.48	47.67	58.65	44.51	57.34	72.79	57.45	53.07	34.59	57.62	69.12	42.83	51.32	43.87	**51.20**

Table 7. **Classwise Semantic Segmentation Results:** We show the segmentation results of ERFNet on derained images obtained by various deraining methods on Rainy Cityscapes. Rainy Images denote the semantic segmentation performance on rainy images. IoU Gain is the gain in IoU by our method compared with Yasarla *et al*. [50]. We observe that the derained images obtained from Yasarla *et al*. [50] suffer large performance loss due to the generalization issue (discussed in subsection 4.4). IoU Gain in blue background shows Group 3 [3] classes, that are most critical for autonomous navigation system.

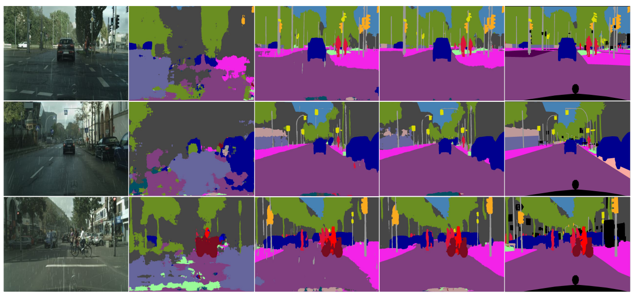


Figure 10. **Semantic Segmentation Results:** Visual segmentation results obtained by ERFNet on images derained from different methods.

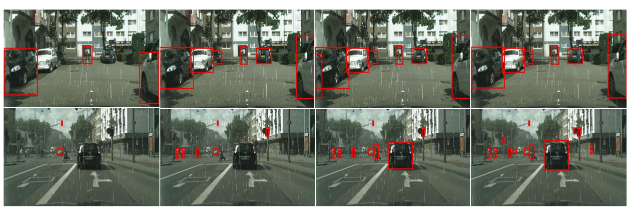


Figure 11. **Object Detection Results:** Detection results obtained from various deraining methods on Rainy Cityscapes dataset.

ting. We randomly choose three pairs of rainy and rain-free images shown in Figure 9 from the Rain 100L dataset. In Figure 9, We find that Yasarla *et al*. [50] struggles to remove rain and retain the input image statistics that can be prominently observed in the Pair 3 result. The reason behind the decreased performance is Yasarla *et al*. [50] relies on a large number of paired rain and rain-free images to learn its hidden representation. The learned hidden representation is further refined by using real-world images. Since the model has only access to a single training pair, it cannot learn robust hidden representation for deraining. In contrast, our proposed method shows steady performance across all the training image pairs. We observe similar behavior in Table 5 quantitatively.

5. Applications

In this section, we employ FLUID in improving semantic segmentation and object detection under rainy conditions. We use Rainy Cityscapes *et al*. [13] dataset to perform the deraining experiments by training all the deraining methods in a 5-shot unsupervised setting. We then pass the derained output images through a semantic segmentation network to show the improvement.

We use ERFNet [37] to perform semantic segmentation. Next, we compare the performance of FLUID with Yasarla *et al*. [50], and Rainy2Clean. Figure 10 and Table 7 shows the qualitative and quantitative results, respectively. We observe that the derained images obtained from Yasarla *et al*. [50] suffer significant performance loss due to the generalization issue (discussed in subsection 4.4) faced by Yasarla *et al*. [50]. We also observe significant improvement in those classes, which are important for an autonomous driving system that belongs to Group 3, according to Chen *et al*. [3]. In Table 7, IoU gain results with blue background show the improvement in Group 3 classes. Next, we perform object detection on the derained images of Rainy Cityscapes from the various deraining methods. We use Faster R-CNN [36] as our detection model. In Figure 11, we observe that our method's detection results show improvement compared to Yasarla *et al*. [50].

6. Conclusion

This work identifies self-supervised methods that struggle to perform image deraining tasks due to poor prior knowledge of rain distribution. We address this problem by proposing a network that learns to estimate the likelihood of rainy pixels with minimal supervision. We also show that our method improves computer vision tasks: semantic segmentation and object detection critical for autonomous applications. This work has opened the doors to further research of restoring images that are taken in adverse weather. It would be interesting to examine our method's applicability for improving images in other adverse weather modalities such as snow.

Acknowledgement

This work was partly funded by IHub-Data at IIIT-Hyderabad and DST through the IMPRINT program.

References

[1] Joshua Batson and Loic Royer. Noise2Self: Blind Denoising by Self-Supervision. In *ICML*, 2019. 1, 3

[2] Tony Chan, Selim Esedoglu, Frederick Park, A Yip, et al. Recent Developments in Total Variation Image Restoration. *Mathematical Models of Computer Vision*, 2005. 4

[3] Bi-ke Chen, Chen Gong, and Jian Yang. Importance-Aware Semantic Segmentation for Autonomous Driving System. In *IJCAI*, 2017. 8

[4] Duan-Yu Chen, Chien-Cheng Chen, and Li-Wei Kang. Visual Depth Guided Color Image Rain Streaks Removal Using Sparse Coding. In *2012 International Symposium on Intelligent Signal Processing and Communications Systems*, 2012. 2

[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 *CVPR*, 2016. 4

[6] Steven B Damelin and NS Hoang. On Surface Completion and Image Inpainting by Biharmonic Functions: Numerical Aspects. *International Journal of Mathematics and Mathematical Sciences*, 2018, 2018. 4

[7] Liang-Jian Deng, Ting-Zhu Huang, Xi-Le Zhao, and Tai-Xiang Jiang. A directional global sparse model for single image rain removal. *Applied Mathematical Modelling*, 2018. 2

[8] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised Visual Representation Learning by Context Prediction. In *ICCV*, 2015. 3

[9] David Eigen, Dilip Krishnan, and Rob Fergus. Restoring An Image Taken Through a Window Covered with Dirt or Rain. In *ICCV*, 2013. 6

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *ICML*. PMLR, 2017. 2

[11] Kshitiz Garg and Shree K Nayar. Photorealistic Rendering of Rain Streaks. *TOG*, 2006. 4

[12] Sungjoo Ha, Martin Kersner, Beomsu Kim, Seokjun Seo, and Dongyoung Kim. MarioNETte: Few-Shot Face Reenactment Preserving Identity of Unseen Targets. In *AAAI*, 2020. 2

[13] Shirsendu Sukanta Halder, Jean-François Lalonde, and Raoul de Charette. Physics-Based Rendering for Improving Robustness to Rain. In *ICCV*, 2019. 4, 8

[14] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 172–189, 2018. 5, 6

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017. 6

[16] Longlong Jing and Yingli Tian. Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey. *TPAMI*, 2020. 1, 3

[17] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu. Automatic Single-Image-Based Rain Streaks Removal via Image Decomposition. *TIP*, 2011. 6

[18] Nikos Komodakis and Spyros Gidaris. Unsupervised Representation Learning by Predicting Image Rotations. In *ICLR*, 2018. 3

[19] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void - Learning Denoising from Single Noisy Images. In *CVPR*, 2019. 1, 3

[20] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In *CVPR*, 2017. 3, 4

[21] Boyun Li, Yuanbiao Gou, Shuhang Gu, Jerry Zitao Liu, Joey Tianyi Zhou, and Xi Peng. You Only Look Yourself: Unsupervised and Untrained Single Image Dehazing Neural Network. *arXiv preprint arXiv:2006.16829*, 2020. 1

[22] Chenhao Li, Yuta Taniguchi, Min Lu, and Shin'ichi Konomi. Few-Shot Font Style Transfer Between Different Languages. In *WACV*, 2021. 2

[23] Minghan Li, Qi Xie, Qian Zhao, Wei Wei, Shuhang Gu, Jing Tao, and Deyu Meng. Video Rain Streak Removal By Multiscale Convolutional Sparse Coding. In *CVPR*, 2018. 2

[24] Ruoteng Li, Loong-Fah Cheong, and Robby T Tan. Heavy Rain Image Restoration: Integrating Physics Model and Conditional Adversarial Learning. In *CVPR*, 2019. 2

[25] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent Squeeze-and-Excitation Context Aggregation Net for Single Image Deraining. In *ECCV*, 2018. 2, 6, 7

[26] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown. Rain Streak Removal Using Layer Priors. In *CVPR*, 2016. 2, 6

[27] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, pages 700–708, 2017. 6

[28] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-Shot Unsupervised Image-to-Image Translation. In *ICCV*, 2019. 2, 5, 6

[29] Yu Luo, Yong Xu, and Hui Ji. Removing Rain From a Single Image via Discriminative Sparse Coding. In *ICCV*, 2015. 2, 6

[30] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In *ICCV*, 2001. 4

[31] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-Reference Image Quality Assessment in the Spatial Domain. *TIP*, 2012. 5

[32] Tsendsuren Munkhdalai and Hong Yu. Meta Networks. In *ICML*. PMLR, 2017. 2

[33] Mehdi Noroozi and Paolo Favaro. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. In *ECCV*, 2016. 3

[34] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context Encoders: Feature Learning by Inpainting. In *CVPR*, 2016. 3

[35] Rui Qian, Robby T. Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive Generative Adversarial Network for Raindrop Removal From a Single Image. In *CVPR*, 2018. 6

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016. 8

[37] Eduardo Romera, José M Alvarez, Luis M Bergasa, and Roberto Arroyo. Erfnet: Efficient Residual Factorized Convnet for Real-Time Semantic Segmentation. *IEEE TITS*, 2017. 8

[38] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015. 3

[39] Kuniaki Saito, Kate Saenko, and Ming-Yu Liu. Coco-funit: Few-shot unsupervised image translation with a content conditioned style encoder. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 382–398. Springer, 2020. 5, 6

[40] Varun Santhaseelan and Vijayan K Asari. Utilizing Local Phase Information to Remove Rain from Video. *IJCV*, 2015. 2

[41] Gerald Schaefer and Michal Stich. UCID: An Uncompressed Color Image Database. In *Storage and Retrieval Methods and Applications for Multimedia 2004*. International Society for Optics and Photonics, 2003. 4

[42] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[43] Ondřej Texler, David Futschik, Michal Kučera, Ondřej Jamriška, Šárka Sochorová, Menclei Chai, Sergey Tulyakov, and Daniel Sỳkora. Interactive Video Stylization Using Few-Shot Patch-Based Training. *TOG*, 2020. 2

[44] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. Spatial Attentive Single-Image Deraining with a High Quality Real Rain Dataset. In *CVPR*, 2019. 6

[45] Yinglong Wang, Shuaicheng Liu, Chen Chen, and Bing Zeng. A Hierarchical Approach for Rain or Snow Removing in a Single Color Image. *TIP*, 2017. 2

[46] 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, 6, 7

[47] Yaochen Xie, Zhengyang Wang, and Shuiwang Ji. Noise2Same: Optimizing a self-supervised bound for image denoising. In *Advances in Neural Information Processing Systems*, volume 33, pages 20320–20330, 2020. 1, 3

[48] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep Joint Rain Detection and Removal from a Single Image. In *CVPR*, 2017. 2, 4

[49] Wenhan Yang, Robby T Tan, Shiqi Wang, Yuming Fang, and Jiaying Liu. Single Image Deraining: From Model-Based to Data-Driven and Beyond. *TPAMI*, 2020. 2

[50] Rajeev Yasarla, Vishwanath A Sindagi, and Vishal M Patel. Syn2real Transfer Learning for Image Deraining using Gaussian Processes. In *CVPR*, 2020. 1, 2, 5, 6, 7, 8

[51] He Zhang and Vishal M Patel. Density-aware Single Image De-raining using a Multi-stream Dense Network. In *CVPR*, 2018. 2

[52] 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. 2, 5, 6, 7

[53] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful Image Colorization. In *ECCV*, 2016. 3

[54] Xiaopeng Zhang, Hao Li, Yingyi Qi, Wee Kheng Leow, and Teck Khim Ng. Rain Removal in Video by Combining Temporal and Chromatic Properties. In *ICME*, 2006. 2

[55] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017. 5, 6

[56] Lei Zhu, Chi-Wing Fu, Dani Lischinski, and Pheng-Ann Heng. Joint Bi-layer Optimization for Single-Image Rain Streak Removal. In *ICCV*, 2017. 2