Syn2Real Transfer Learning for Image Deraining using Gaussian Processes

Rajeev Yasarla* Vishwanath A. Sindagi* Vishal M. Patel
Johns Hopkins University
Department of Electrical and Computer Engineering, Baltimore, MD 21218, USA
{ryasarl1, vishwanathsindagi, vpatel36}@jhu.edu

Abstract

Recent CNN-based methods for image deraining have achieved excellent performance in terms of reconstruction error as well as visual quality. However, these methods are limited in the sense that they can be trained only on fully labeled data. Due to various challenges in obtaining real world fully-labeled image deraining datasets, existing methods are trained only on synthetically generated data and hence, generalize poorly to real-world images. The use of real-world data in training image deraining networks is relatively less explored in the literature. We propose a Gaussian Process-based semi-supervised learning framework which enables the network in learning to derain using synthetic dataset while generalizing better using unlabeled real-world images. Through extensive experiments and ablations on several challenging datasets (such as Rain800, Rain200H and DDN-SIRR), we show that the proposed method, when trained on limited labeled data, achieves on-par performance with fully-labeled training. Additionally, we demonstrate that using unlabeled real-world images in the proposed GP-based framework results in superior performance as compared to existing methods.

1. Introduction

Images captured under rainy conditions are often of poor quality. The artifacts introduced by rain streaks adversely affect the performance of subsequent computer vision algorithms such as object detection and recognition [12, 28, 41, 4]. With such algorithms becoming vital components in several applications such as autonomous navigation and video surveillance [37, 25, 36], it is increasingly important to develop sophisticated algorithms for rain removal.

The task of rain removal is plagued with several issues such as (i) large variations in scale, density and orientation of the rain streaks, and (ii) lack of real-world labeled training data. Most of the existing work [58, 9, 23, 51, 60, 6, 52, 20] in image deraining have largely focused towards addressing the first issue. For example, Fu *et al*. [9] developed an end-to-end method which focuses on high frequency detail during training a deraining network. In another work, Zhang and Patel [58] proposed a density-aware multi-steam densely connected network for joint rain den-

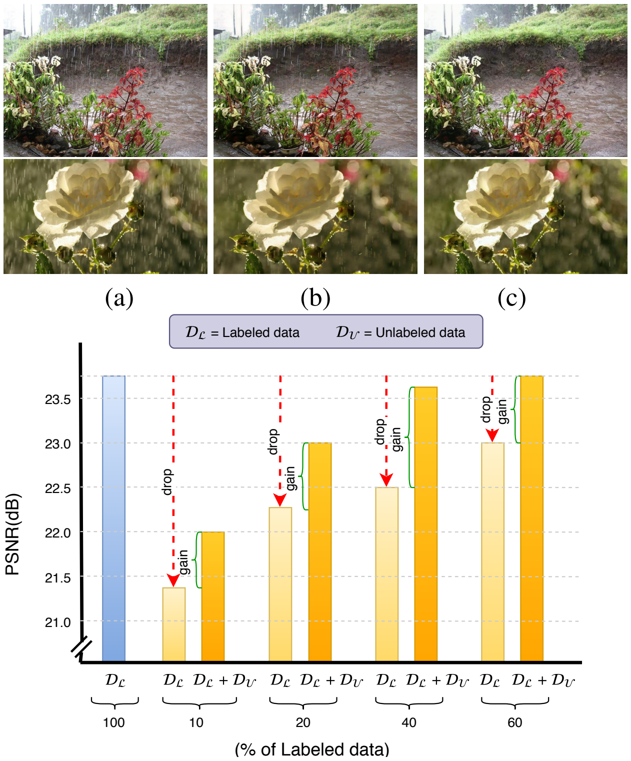

Figure 1. *Top row*: (a) Input rainy image. (b) Output from a network trained using only the synthetic data. (c) Output from a network trained using the synthetic data and unlabeled real-world data. This shows better generalization. *Bottom row*: Results from Semi-supervised learning (SSL) experiments. Reducing the amount of labeled data used for training results in the performance drop. Using the proposed SSL framework, we are able to recover the performance.

*equal contribution

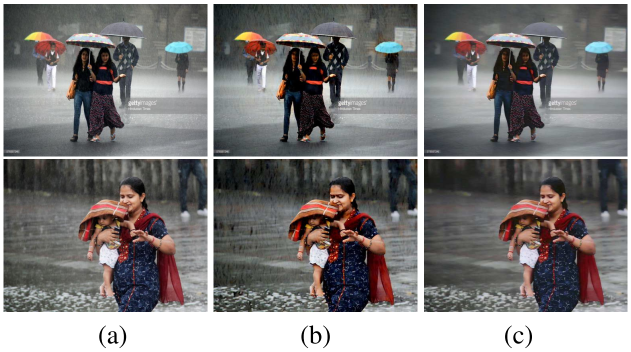

Figure 2. Derained results. (a) Input rainy images. (a) SSIR output [49]. (c) Our output. It can be observed that the proposed method achieves better deraining.

sity estimation and deraining. Li *et al.* [21] incorporated context information through recurrent neural networks for rain removal. More recently, Ren *et al.* [40] introduced a progressive ResNet that leverages dependencies of features across stages. While these methods have achieved superior performance in obtaining high-quality derained images, they are inherently limited due to the fact that they are fully-supervised networks and they can only leverage fully-labeled training data. However, as mentioned earlier, obtaining labeled real-world training data is quite challenging and hence, existing methods typically train their networks only on synthetically generated rain datasets [57, 51].

The use of synthetic datasets results in sub-optimal performance on the real-world images, typically because of the distributional-shift between synthetic and rainy images [4]. Despite this gap in performance, this issue remains relatively unexplored in the literature.

Recently, Wei *et al.* [49] proposed a semi-supervised learning framework (SIRR) where they simultaneously learn from labeled and unlabeled data for the purpose of image deraining. For training on the labeled data, they use the traditional mean absolute error loss between predictions and ground-truth (GT). For unlabeled data, they model the rain residual (difference between the input and output) through a likelihood term imposed on a Gaussian mixture model (GMM). Furthermore, they enforce additional consistency that the distribution of synthetic rain is closer to that of real rain by minimizing the Kullback-Leibler (KL) divergence between them. This is the first method to formulate the task of image deraining in a semi-supervised learning framework that can leverage unlabeled real-world images to improve the generalization capabilities. Although this method achieves promising results, it has the following drawbacks: (i) Due to the multi-modal nature of rain residuals, the authors assume that they can be modeled using GMM. This is true only if the actual residuals are being used to compute the GMM parameters. However, the authors use the predicted rain residuals of real-world (unlabeled) images over training iterations for modeling the GMM. The same model is then used to compute the likelihood of the predicted residuals (of unlabeled images) in the subsequent iterations. Hence, if the GMM parameters learned during the initial set of iterations are not accurate, which is most likely the case in the early stages of training, it will lead to sub-optimal performance. (ii) The goal of using the KL divergence is to bring the synthetic rain distribution closer to the real rain distribution. As stated earlier, the predictions of real rain residuals will not be accurate during the earlier stages of training and hence, minimizing the discrepancy between the two distributions may not be appropriate. (iii) Using GMM to model the rain residuals requires one to choose the number of mixture components, rendering the model to be sensitive to such choices.

Inspired by Wei *et al.* [49], we address the issue of incorporating unlabeled real-world images into the training process for better generalization by overcoming the drawbacks of their method. In contrast to [49], we use a non-parametric approach to generate supervision for the unlabeled data. Specifically, we propose a Gaussian-process (GP) based semi-supervised learning (SSL) framework which involves iteratively training on the labeled and unlabeled data. The labeled learning phase involves training on the labeled data using mean squared error between the predictions and the ground-truth. Additionally, inputs (from labeled dataset) are projected onto the latent space, which are then modeled using GP. During the unlabeled training phase, we generate pseudo-GT for the unlabeled inputs using the GP modeled earlier in the labeled training phase. This pseudo GT is then used to supervise the intermediate latent space for the unlabeled data. The creation of the pseudo GT is based on the assumption that unlabeled images, when projected to the latent space, can be expressed as a weighted combination of the labeled data features where the weights are determined using a kernel function. These weights indicate the uncertainty of the labeled data points being used to formulate the unlabeled data point. Hence, minimizing the error between the unlabeled data projections and the pseudo GT reduces the variance, hence resulting in the network weights being adapted automatically to the domain of unlabeled data. Fig. 1 demonstrates the results of leveraging unlabeled data using the proposed framework. Fig. 2 compares the results of the proposed method with SIRR [49]. One can clearly see that our method is able to provide better results as compared to SIRR [49].

To summarize, this paper makes the following contributions:

- We propose a non-parametric approach for performing SSL to incorporate unlabeled real-world data into the training process.

- The proposed method consists of modeling the intermediate latent space in the network using GP, which is then used to create the pseudo GT for the unlabeled data. The pseudo GT is further used to supervise the network at the intermediate level for the unlabeled data.
- Through extensive experiments on different datasets, we show that the proposed method is able to achieve on-par performance with limited training data as compared to network trained with full training data. Additionally, we also show that using the proposed GP-based SSL framework to incorporate the unlabeled real-world data into the training process results in better performance as compared to the existing methods.

2. Related work

Image deraining is an extensively researched topic in the low-level computer vision community. Several approaches have been developed to address this problem. These approaches are classified into two main categories: single image-based techniques [58, 9, 23, 51, 60, 54] and video-based techniques [59, 11, 44, 27, 17, 26]. A comprehensive analysis of these methods can be found in [19].

Single image-based techniques typically consume a single image as the input and attempt to reconstruct a rain-free image from it. Early methods for single image deraining either employed priors such as sparsity [56, 29] and low-rank representation [3] or modeled image patches using techniques such as dictionary learning [2] and GMM [42]. Recently, deep learning-based techniques have gained prominence due to their effectiveness in ability to learn efficiently from paired data. Video-based deraining techniques typically leverage additional information by enforcing constraints like temporal consistency among the frames.

In this work, we focus on single image-based deraining that specifically leverages additional unlabeled real-world data. Fu et al. [7] proposed a convolutional neural network (CNN) based approach in which they learns a mapping from a rainy image to the clean image. Zhang et al. [57] introduced generative adversarial network (GAN) for image deraining that resulted in high quality reconstructions. Fu et al. [9] presented an end-to-end CNN called, deep detail network, which directly reduces the mapping range from input to output. Zhang and Patel [58] proposed a density-aware multi-stream densely connected CNN for joint rain density estimation and deraining. Their network first classifies the input image based on the rain density, and then employs an appropriate network based on the predicted rain density to remove the rain streaks from the input image. Wang et al. [48] employed a hierarchical approach based on estimating different frequency details of an image to obtain the derained image. Qian *et al*. [38] proposed a GAN to remove rain drops from camera lens. To enable the network focus on important regions, they injected attention map into the generative and discriminating network. Li et al. [21] proposed a convolutional and recurrent neural network-based method for single image deraining that incorporates context information. Recently, Li *et al*. [18] and Hu *et al*. [13] incorporated depth information to improve the deraining quality. Yasarla and Patel [53] employed uncertainty mechanism to learn location-based confidence for the predicted residuals. Wang *et al*. [47] proposed a spatial attention network that removes rain in a local to global manner.

3. Background

In this section, we provide a formulation of the problem statement, followed by a brief description of key concepts in GP.

3.1. Single image de-raining

Existing image deraining methods assume the additive model where the rainy image (x) is considered to be the superposition of a clean image (y) and a rain component (r), *i.e*.,

$$x = y + r. \tag{1}$$

Single image deraining task is typically an inverse problem where the goal is to estimate the clean image y, given a rainy image x. This can be achieved by learning a function that either (i) directly maps from rainy image to clean image [5, 8, 60, 56], or (ii) extracts the rain component from the rainy image which can then be subtracted from the rainy image to obtain the clean image [9, 58, 22]. We follow the second approach of estimating the rain component from a rainy image.

3.2. Semi-supervised learning

In semi-supervised learning, we are given a labeled dataset of input-target pairs ($\{x, y\} \in \mathcal{D}_{\mathcal{L}}$) sampled from an unknown joint distribution $p(x, y)$ and unlabeled input data points $x \in \mathcal{D}_{\mathcal{U}}$ sampled from $p(x)$. The goal is to learn a function $f(x|\theta)$ parameterized by θ that accurately predicts the correct target y for unseen samples from $p(x)$. The parameters θ are learned by leveraging both labeled and unlabeled datasets. Since the labeled dataset consists of input-target pairs, supervised loss functions such as mean absolute error or cross entropy are typically used to train the networks. The unlabeled datapoints form $\mathcal{D}_{\mathcal{U}}$ are used to augment $f(x|\theta)$ with information about the structure of $p(x)$ like shape of the data manifold [35] via different techniques such as enforcing consistent regularization [15], virtual adversarial training [34] or pseudo-labeling [16].

Following [49], we employ the semi-supervised learning framework to leverage unlabeled real-world data to obtain better generalization performance. Specifically, we consider the synthetically generated rain dataset consisting of input-target pairs as the labeled dataset $\mathcal{D}_{\mathcal{L}}$ and real-world unlabeled images as the unlabeled dataset $\mathcal{D}_{\mathcal{U}}$. In contrast to [49], we follow the approach of pseudo-labeling to leverage the unlabeled data.

3.3. Gaussian processes

A Gaussian process (GP) $f(v)$ is an infinite collection of random variables, of which any finite subset is jointly Gaussian distributed. A GP is completely specified by its mean function and covariance function which are defined as follows

$$m(v) = \mathbb{E}[f(v)], \tag{2}$$

$$K(v, v') = \mathbb{E}\left[(f(v) - m(v))\left(f(v') - m(v')\right)\right], \tag{3}$$

where $v, v' \in \mathcal{V}$ denote the possible inputs that index the GP. The covariance matrix is constructed from a covariance function, or kernel, K which expresses some prior notion of smoothness of the underlying function. GP can then be denoted as follows

$$f(v) \sim \mathcal{GP}(m(v), K(v, v') + \sigma_\epsilon^2 I). \tag{4}$$

where is I identity matrix and σ_ϵ^2 is the variance of the additive noise. Any collection of function values is then jointly Gaussian as follows

$$f(V) = \left[f(v_1), \ldots, f(v_n)\right]^T \sim \mathcal{N}\left(\mu, K(V, V') + \sigma_\epsilon^2 I\right) \tag{5}$$

with mean vector and covariance matrix defined by the GP as mentioned earlier. To make predictions at unlabeled points, one can compute a Gaussian posterior distribution in closed form by conditioning on the observed data. The reader is referred to [39] for a detailed review on GP.

4. Proposed method

As shown in Fig. 3, the proposed method consists of a CNN based on the UNet structure [43], where each block is constructed using a Res2Block [10]. The details of the network architecture are provided in the supplementary material. In summary, the network is made up of an encoder ($h(x, \theta_{enc})$) and a decoder ($g(z, \theta_{dec})$). Here, the encoder and decoder are parameterized by θ_{enc} and θ_{dec}, respectively. Furthermore, x is the input to the network which is then mapped by the encoder to a latent vector z. In our case, x is the rainy image from which we want to remove the rain streaks. The latent vector is then fed to the decoder to produce the output r, which in our case is the rain streaks. The rain streak component is then subtracted form the rainy image (x) to produce the clean image (y), *i.e.*,

$$y = x - r, \tag{6}$$

where

$$r = g(h(x, \theta_{enc}), \theta_{dec}). \tag{7}$$

In our problem formulation, the training dataset is $\mathcal{D} = \mathcal{D}_{\mathcal{L}} \cup \mathcal{D}_{\mathcal{U}}$, where $\mathcal{D}_{\mathcal{L}} = \{x_l^i, y_l^i\}_{i=1}^{N_l}$ is a labeled training set consisting of N_l samples and $\mathcal{D}_{\mathcal{U}} = \{x_u^i\}_{i=1}^{N_u}$ is a set consisting of N_u unlabeled samples. For the rest of the paper, $\mathcal{D}_{\mathcal{L}}$ refers to labeled "*synthetic*" dataset and $\mathcal{D}_{\mathcal{U}}$ refers to unlabeled "*real-world*" dataset, unless otherwise specified.

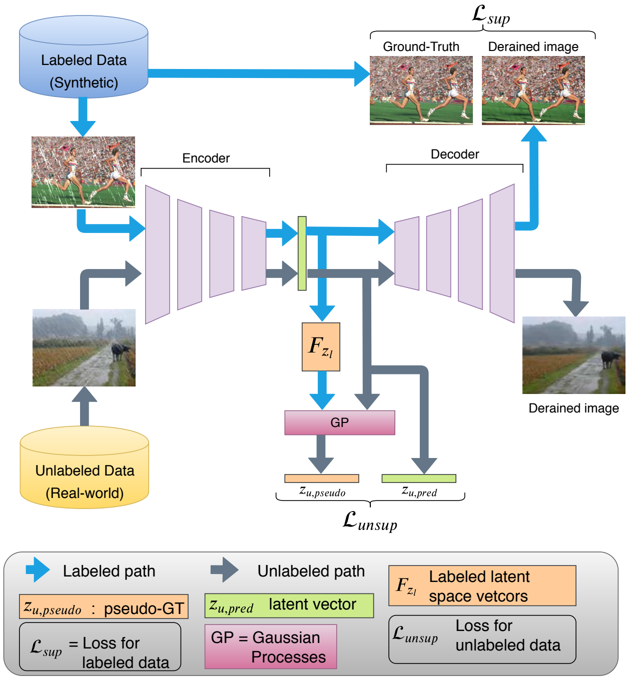

Figure 3. Overview of the proposed GP-based SSL framework. We leverage unlabeled data during learning. The training process consists of iterating over labeled data and unlabeled data. During the labeled training phase, we use supervised loss function consisting of l_1 error and perceptual loss between the prediction and targets. In the unlabeled phase, we jointly model the labeled and unlabeled latent vectors using GP to obtain the pseudo-GT for the unlabeled sample at the latent space. We use this pseudo-GT for supervision.

The goal of the proposed method is to learn the network parameters by leveraging both labeled ($\mathcal{D}_{\mathcal{L}}$) and unlabeled dataset ($\mathcal{D}_{\mathcal{U}}$). The training process iterates over labeled and unlabeled datasets. The network parameters are learned by minimizing (i) the supervised loss function ($\mathcal{L}_{sup}$) in the labeled training phase, and (ii) the unsupervised loss function ($\mathcal{L}_{unsup}$) in the unlabeled training phase. For the unlabeled training phase, we generate pseudo GT using GP formulation, which is then used in the unsupervised loss function. The two training phases are described in detail in the following sections.

4.1. Labeled training phase

In this phase, we use the labeled data $\mathcal{D}_{\mathcal{L}}$ to learn the network parameters. Specifically, we minimize the following supervised loss function

$$\mathcal{L}_{sup} = \mathcal{L}_1 + \lambda_p \mathcal{L}_p, \tag{8}$$

where λ_p is a constant, and $\mathcal{L}_1$ and $\mathcal{L}_p$ are l_1-loss and perceptual loss [14, 55] functions, respectively. They are defined as follows

$$\mathcal{L}_1 = \|y_l^{pred} - y_l\|_1, \tag{9}$$

$$\mathcal{L}_p = \|\Phi_{VGG}(y_l^{pred}) - \Phi_{VGG}(y_l)\|_2^2, \tag{10}$$

where $y_l^{pred} = g(z, \theta_{dec})$ is the predicted output, y_l is the ground-truth, $z = h(x, \theta_{enc})$ is the intermediate latent space vector and $\Phi_{VGG}(\cdot)$ represents the pre-trained VGG-16 [46] network. For more details on the perceptual loss, please refer to supplementary material.

In addition to minimizing the loss function, we also store the intermediate feature vectors z_l^i's for all the labeled training images x_l^i's in a matrix F_{z_l}. That is $F_{z_l} = \{z_l^i\}_{i=1}^{N_l}$. It is used later in the unlabeled training phase to generate the pseudo-GT for the unlabeled data. In our case, z_l^i is a vector of size $1 \times M$, where M = 32,768 for the network in our proposed method. Thus F_{z_l} is a matrix of size $N_l \times M$.

4.2. Unlabeled training phase

In this phase, we leverage the unlabeled data $\mathcal{D_U}$ to improve the generalization performance. Specifically, we provide supervision at the intermediate latent space by minimizing the error between the predicted latent vectors and the pseudo-GT obtained by modeling the latent space vectors of the labeled sample images F_{z_l} and z_u^{pred} jointly using GP.

Pseudo-GT using GP: The training occurs in an iterative manner, where we first learn the weights using the labeled data ($\mathcal{D_L}$) followed by weight updates using the unlabeled data ($\mathcal{D_U}$). After the first iteration on $\mathcal{D_L}$, we store the latent space vectors of the labeled data in a list F_{z_l}. These vectors lie on a low dimension manifold. During the unlabeled phase, we project the latent space vector (z_u) of the unlabeled input onto the space of labeled vectors $F_{z_l} = \{z_l^i\}_{i=1}^{N_l}$. That is, we express the unlabeled latent space vector z_u^k corresponding to the k^{th} training sample from $\mathcal{D_U}$ as

$$z_u^k = \sum_{i=1}^{N_l} \alpha_i z_l^i + \epsilon, \tag{11}$$

where α_i are the coefficients, and ϵ is additive noise $\mathcal{N}(0, \sigma_\epsilon^2)$.

With this formulation, we can jointly model the distribution of the latent space vectors of the labeled and the unlabeled samples using GP. Conditioning the joint distribution will yield the following conditional multi-variate Gaussian distribution for the unlabeled sample

$$P(z_u^k | \mathcal{D_L}, F_{z_l}) = \mathcal{N}(\mu_u^k, \Sigma_u^k), \tag{12}$$

where

$$\mu_u^k = K(z_u^k, F_{z_l})[K(F_{z_l}, F_{z_l}) + \sigma_\epsilon^2 I]^{-1} F_{z_l}, \tag{13}$$

$$\Sigma_u^k = K(z_u^k, z_u^k) - K(z_u^k, F_{z_l})[K(F_{z_l}, F_{z_l}) + \sigma_\epsilon^2 I]^{-1} K(F_{z_l}, z_u^k) + \sigma_\epsilon^2 \tag{14}$$

where σ_ϵ^2 is set equal to 1, K is defined by the kernel function as follows

$$K(Z, Z)_{k,i} = \kappa(z_u^k, z_l^i) = \frac{\langle z_u^k, z_l^i \rangle}{|z_u^k| \cdot |z_l^i|}. \tag{15}$$

Note that F_{z_l} contains the latent space vectors of all the labeled images, $K(F_{z_l}, F_{z_l})$ is a matrix of size $N_l \times N_l$, and $K(z_u^k, F_{z_l})$ is a vector of size $1 \times N_l$. Using all the vectors may not be necessarily optimal for the following reasons: (i) These vectors will correspond to different regions in the image with a wide diversity in terms of content and density/orientation of rain streaks. It is important to consider only those vectors that are similar to the unlabeled vector. (ii) Using all the vectors is computationally prohibitive. Hence, we use only N_n nearest labeled vectors corresponding to an unlabeled vector. More specifically, we replace F_{z_l} by $F_{z_l,n}$ in Eq. (11)-(14). Here $F_{z_l,n} = \{z_l^j : z_l^j \in nearest(z_u^k, F_{z_l}, N_n)\}$ with $nearest(p, Q, N_n)$ being a function that finds top N_n nearest neighbors of p in Q.

We use the mean predicted by Eq. (13) as the pseudo-GT ($z_{u,pseudo}^k$) for supervision at the latent space level. By minimizing the error between $z_{u,pred}^k = h(x_u, \theta_{enc})$ and $z_{u,pseudo}^k$, we update the weights of the encoder $h(\cdot, \theta_{enc})$, thereby adapting the network to unlabeled data which results in better generalization. We also minimize the prediction variance by minimizing Eq. (14). Using GP we are approximating z_u^k, latent vector of an unlabeled image using the latent space vectors in F_{z_l}, by doing this we may end up computing incorrect pseudo-GT predictions because of the dissimilarity between the latent vectors. This dissimilarity is due to different compositions in rain streaks like different densities, shapes, and directions of rain streaks. In order to address this issue we minimize the variance $\Sigma_{u,n}^k$ computed between z_u^k and the N_n nearest neighbors in the latent space vectors using GP. Additionally, we maximize the variance $\Sigma_{u,f}^k$ computed between z_u^k and the N_f farthest vectors in the latent space using GP, in order to ensure that the latent vectors in F_{z_l} are dissimilar to the unlabeled vector z_u^k and do not affect the GP prediction, as defined below

$$\Sigma_{u,f}^k = K(z_u^k, z_u^k) - K(z_u^k, F_{z_l,f})[K(F_{z_l,f}, F_{z_l,f}) + \sigma_\epsilon^2 I]^{-1} K(F_{z_l,f}, z_u^k) + \sigma_\epsilon^2, \tag{16}$$

where $F_{z_l,f}$ is the matrix of N_f labeled vectors that are farthest from z_u^k.

Thus, the loss used during training using the unlabeled data is defined as follows

$$\mathcal{L}_{unsup} = \|z_{u,pred}^k - z_{u,pseudo}^k\|_2 + \log \Sigma_{u,n}^k + \log(1 - \Sigma_{u,f}^k), \tag{17}$$

where $z_{u,pred}^k$ is the latent vector obtained by forwarding an unlabeled input image x_u^k through the encoder h, *i.e.*, $z_{u,pred}^k = h(x_u, \theta_{enc})$, $z_{u,pseudo}^k = \mu_u^k$ is the pseudo-GT latent space vector (see Eq. (13)), and $\Sigma_{u,n}^k$ is the variance obtained by replacing F_{z_l} in Eq. (14) with $F_{z_l,n}$.

Table 1. Effect of using unlabeled real-world data in training process on DDN-SIRR dataset. Evaluation is performed on synthetic dataset similar to [49]. Proposed method achieves better gain in PSNR as compared to SIRR[49] in the case of both Dense and Sparse categories. $\mathcal{D}_{\mathcal{L}}$ indicates training using only labeled dataset and $\mathcal{D}_{\mathcal{L}} + \mathcal{D}_{\mathcal{U}}$ indicates training using both labeled and unlabeled dataset.

Dataset	Input	Methods that use only synthetic dataset							Methods that use synthetic and real-world dataset					
		DSC [30]	LP [24]	JORDER [51]	DDN [9]	JBO [60]	DID-MDN [58]	UMRL [53]	SIRR [49] (CVPR '19)			Ours		
		(ICCV '15)	(CVPR '16)	(CVPR '17)	(CVPR '17)	(CVPR '17)	(CVPR '18)	(CVPR '19)	$\mathcal{D}_{\mathcal{L}}$	$\mathcal{D}_{\mathcal{L}} + \mathcal{D}_{\mathcal{U}}$	Gain	$\mathcal{D}_{\mathcal{L}}$	$\mathcal{D}_{\mathcal{L}} + \mathcal{D}_{\mathcal{U}}$	Gain
Dense	17.95	19.00	19.27	18.75	19.90	18.87	18.60	20.11	20.01	21.60	1.59	20.24	**22.36**	**2.12**
Sparse	24.14	25.05	25.67	24.22	26.88	25.24	25.66	26.94	26.90	26.98	0.08	26.15	**27.26**	**1.11**

4.3. Total loss

The overall loss function used for training the network is defined as follows

$$\mathcal{L}_{total} = \mathcal{L}_{sup} + \lambda_{unsup}\mathcal{L}_{unsup}, \tag{18}$$

where λ_{unsup} is a pre-defined weight that controls the contribution from $\mathcal{L}_{sup}$ and $\mathcal{L}_{unsup}$.

4.4. Training and implementation details

We use the UDeNet network that is based on the UNet style encoder-decoder architecture [43] with a slight difference in the building blocks. Details of the network architecture are provided in the supplementary material. The network is trained using the Adam optimizer with a learning rate of 0.0002 and batchsize of 4 for a total of 60 epochs. Furthermore, we reduce the learning rate by a factor of 0.5 at every 25 epochs. We use $\lambda_p = 0.04$ (Eq. (8)), $\lambda_{unsup} = 1.5 \times 10^{-4}$ (Eq. (18)), $N_n = 64$ and $N_f = 64$. During training, the images are randomly cropped to the size of 256×256. Ablation studies with different hyperparameter values are provided in supplementary material.

5. Experiments and results

In this section, we present the details of the datasets and various experiments conducted to demonstrate the effectiveness of the proposed framework. Specifically, we conducted two sets of experiments. In the first set, we analyze the effectiveness of using the unlabeled real-world data during training using the proposed framework. Here, we compare the performance of our method with a recent SSL framework for image deraining (SIRR) [49]. In the second set of experiments, we evaluate the proposed method by training it on different percentages of the labeled data.

5.1. Datasets

Rain800: This dataset was introduced by Zhang *et al.* [57] and it contains a total of 800 images. The train split consists of 700 real-world clean images, with 500 images chosen randomly from the first half of the UCID dataset [45] and 200 images chosen randomly from the BSD-500 train set [1]. The test set consists of a total of 100 images, with 50 images chosen randomly from the second half of the UCID dataset and the rest 50 chosen randomly from the test set of the BSD-500 dataset. The authors generate the corresponding rainy images by synthesizing rain-streaks of different intensities and orientations.

Rain200H: Yang et al. [51] collected images from BSD200 [31] to create 3 datasets: Rain12, Rain200L and Rain200H. Following [22], we use the most difficult one, Rain200H, to evaluate our model. The images for the training set are collected from the BSD300 dataset. Rain streaks with different orientations are synthesized using photo-realistic techniques. There are 1,800 synthetic image pairs in the Rain200H train set, and 200 pairs in the test set.

DDN-SIRR dataset: Wei *et al.* [49] constructed a dataset consisting of labeled synthetic training set and unlabeled real-world dataset. This dataset is constructed specifically to evaluate semi-supervised learning frameworks. The labeled training set is borrowed from Fu *et al.* [9] and it consists of 9,100 image pairs obtained by synthesizing different types of rain streaks on the clean images from the UCID dataset [45]. The unlabeled real-world synthetic train set comprises of images collected from [50, 51, 57] and Google image search. Furthermore, the test set consists of two categories: (i) Dense rain streaks, and (ii) Sparse rain streaks Each test set consists of 10 images.

5.2. Use of real-world data

The goal of this experiment is to analyze the effect of using unlabeled real-world data along with labeled synthetic dataset in the training framework. Following the protocol set by [49], we use the "labeled synthetic" train set from the DDN-SIRR dataset as $\mathcal{D}_{\mathcal{L}}$ and the "real-world" train set from the DDN-SIRR dataset as $\mathcal{D}_{\mathcal{U}}$. Evaluation is performed on (i) Synthetic test set from DDN-SIRR, and (ii) Real-world test set from DDN-SIRR.

Results on synthetic test set: The evaluation results on the synthetic test set are shown in Table. 1. Similar to [49], we use PSNR as the evaluation metric. We compare the proposed method with several existing approaches such as DSC [30], LP [24], JORDER [51], DDN [9], JBO [60] and DID-MDN [58]. These methods can use only synthetic dataset. Since the proposed method has the ability to leverage unlabeled real-world data, it is able to achieve significantly better results as compared to the existing approaches.

Furthermore, we also compare the performance of our method with a recent GMM-based semi-supervised deraining method (SIRR) [49]. It can be observed from Table

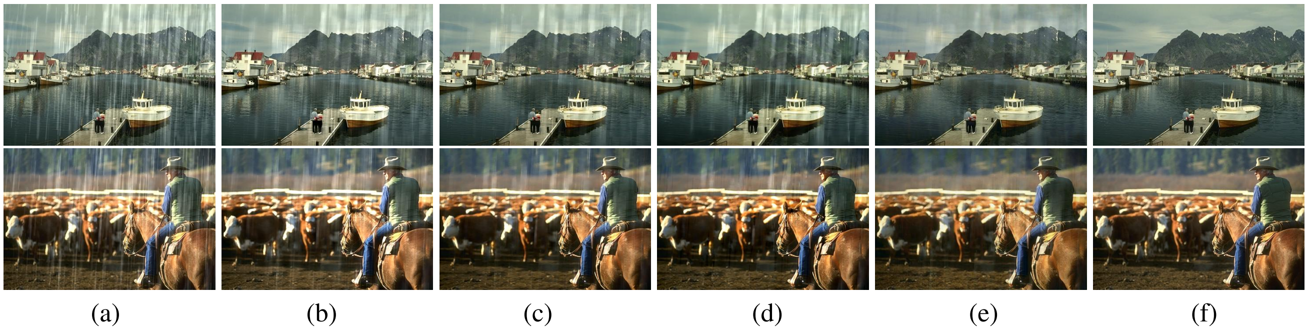

Figure 4. Qualitative results on DDN-SIRR **synthetic** test set. (a) Input rainy image (b) DID-MDN [58](CVPR '18) (c) DDN [9](CVPR '17) (d) SIRR [49](CVPR '19) (e) Ours (f) ground-truth image.

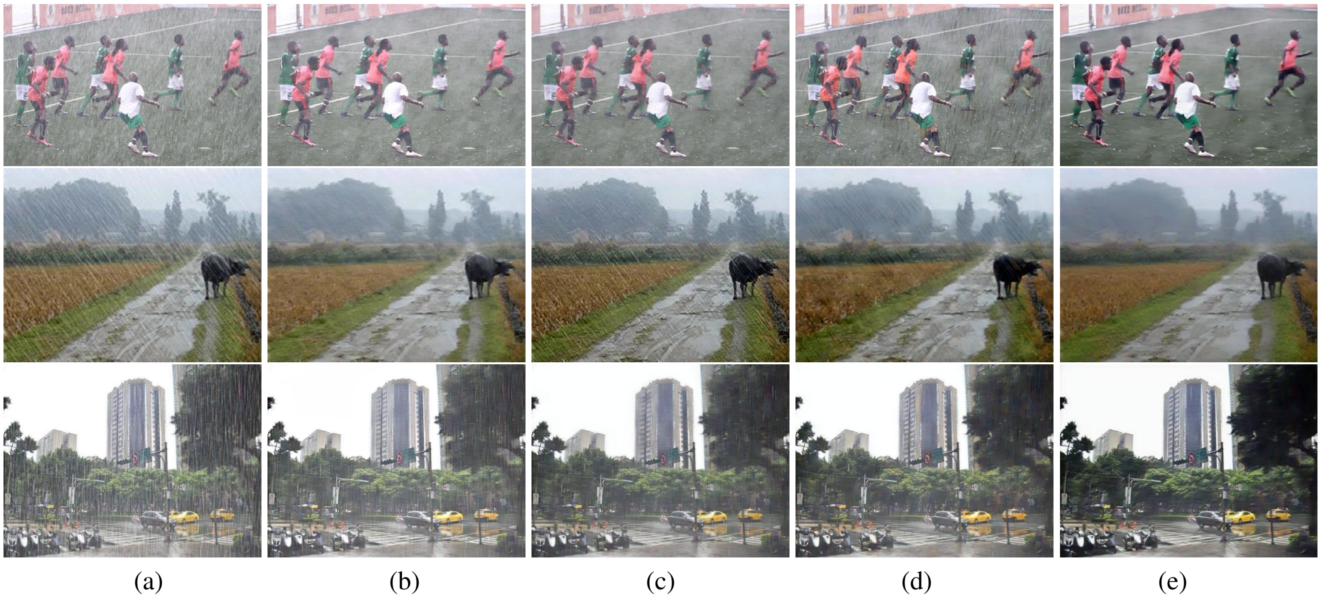

Figure 5. Qualitative results on DDN-SIRR **real-world** test set. (a) Input rainy image (b) DID-MDN [58] (c) DDN [9] (d) SIRR [49] (e) Ours.

1 that the proposed method outperforms SIRR with significant margins. Additionally, we also illustrate the gains[1] achieved due to the use of additional unlabeled real-world data by both the methods. The proposed method achieves greater gains as compared to SIRR, which indicates that the proposed method has better capacity to leverage unlabeled data.

Qualitative results on the test set are shown in Fig. 4. As can be seen from this figure, the proposed method achieves better quality reconstructions as compared to the existing methods.

Results on real-world test set: Similar to [49], we evaluate the proposed method on the real-world test set from DDN-SIRR. We use no-reference quality metrics NIQE [33] and BRISQUE [32] to perform quantitative comparison. The results are shown in Table. 2. We compare the performance of our method with SIRR [49] which also leverages unlabeled data. It can be observed that the proposed method achieves better performance than SIRR. Note that lower scores indicate better performance. Furthermore, the proposed method is able to achieve better gains with the use of unlabeled data as compared to SIRR.

From these experiments, we can conclude that the proposed GP-based framework when leverages unlabeled real-world data results in better generalization as compared to not using the unlabeled data.

5.3. Ablation study: SSL experiments

In this set of experiments, we analyze the capacity of the proposed method to leverage unlabeled data by varying the amount of labeled data used for training the network. Since, the goal is to evaluate the method quantitatively, we use synthetic datasets (Rain800 and Rain200H) for these experiments. Specifically, we run 5 experiments where we train the network on 10%, 20%, 40%, 60% and 100% of

[1] The gain is computed by subtracting the performance obtained using only $\mathcal{D}_{\mathcal{L}}$ from the performance obtained using $\mathcal{D}_{\mathcal{L}} + \mathcal{D}_{\mathcal{U}}$.

Table 2. Effect of using unlabeled real-world data in training process on the DDN-SIRR dataset. Evaluation is performed on the **real-world** test set of DDN-SIRR dataset using no-reference quality metrics (NIQE and BRISQUE). Note that lower scores indicate better performance.

Metrics	Input	SIRR [49]			Ours		
		$\mathcal{D}_{\mathcal{L}}$	$\mathcal{D}_{\mathcal{L}}+\mathcal{D}_{\mathcal{U}}$	Gain	$\mathcal{D}_{\mathcal{L}}$	$\mathcal{D}_{\mathcal{L}}+\mathcal{D}_{\mathcal{U}}$	Gain
NIQE	4.671	3.86	3.84	0.02	3.85	3.78	0.07
BRISQUE	31.37	26.61	25.29	1.32	25.77	22.95	2.82

Table 3. SSL experiments on Rain800 [57] dataset: The percentage of labeled data used for training is varied between 10% and 100%. Consistent gains are observed when unlabeled data is leveraged using the proposed method as compared to the use of only labeled data.

$\mathcal{D}_{\mathcal{L}}$ %	PSNR			SSIM		
	$\mathcal{D}_{\mathcal{L}}$	$\mathcal{D}_{\mathcal{L}}+\mathcal{D}_{\mathcal{U}}$	Gain	$\mathcal{D}_{\mathcal{L}}$	$\mathcal{D}_{\mathcal{L}}+\mathcal{D}_{\mathcal{U}}$	Gain
10%	21.31	22.02	0.71	0.729	0.750	0.021
20%	22.28	22.95	0.67	0.752	0.768	0.016
40%	22.61	23.60	0.99	0.761	0.788	0.027
60%	22.96	23.70	0.74	0.775	0.795	0.020
100%	23.74	–	–	0.799	–	–

Table 4. SSL experiments on Rain200H [51] dataset: The percentage of labeled data used for training is varied between 10% and 100%. Consistent gains are observed when unlabeled data is leveraged using the proposed method as compared to the use of only labeled data.

$\mathcal{D}_{\mathcal{L}}$ %	PSNR			SSIM		
	$\mathcal{D}_{\mathcal{L}}$	$\mathcal{D}_{\mathcal{L}}+\mathcal{D}_{\mathcal{U}}$	Gain	$\mathcal{D}_{\mathcal{L}}$	$\mathcal{D}_{\mathcal{L}}+\mathcal{D}_{\mathcal{U}}$	Gain
10%	22.92	23.64	0.72	0.742	0.767	0.025
20%	23.22	24.00	0.78	0.755	0.776	0.021
40%	23.84	24.75	0.91	0.772	0.794	0.022
60%	24.32	25.26	0.94	0.782	0.808	0.026
100%	25.27	–	–	0.810	–	–

the dataset as the labeled data $\mathcal{D}_{\mathcal{L}}$. The rest of the dataset is leveraged as the unlabeled data $\mathcal{D}_{\mathcal{U}}$. We use PSNR and SSIM metrics for this ablation study.

The results on the Rain800 test and Rain200H set are shown in Table 3 and 4, respectively. From these tables, we make following observations: (i) Reducing the amount of labeled data leads to significant drop in performance as compared to using 100% of the data as the labeled data. For example, the performance drops from 23.74 dB when using 100% data to 22.6dB after reducing the labeled data to 40%. (ii) By using unlabeled data in the proposed SSL framework, we are able to achieve improvements as compared to using only labeled data. (iv) The gain in performance obtained due to the use of unlabeled data is consistent across different amounts of labeled data. (iii) Finally, the proposed method with just 60% labeled data (and unlabeled data) is able to achieve performance that is comparable to that achieved by using 100% labeled data.

Fig. 6 and 7 show sample qualitative results when using 10% and 40% labeled data, respectively. It can be observed that using additional unlabeled data results in better performance as compared to using only labeled data.

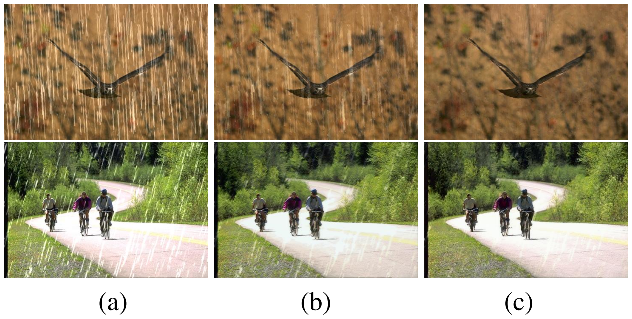

Figure 6. Results of experiments with 10% labeled data on Rain200H (a) Input rainy image (a) Using only labeled data (c) Using labeled and unlabeled data.

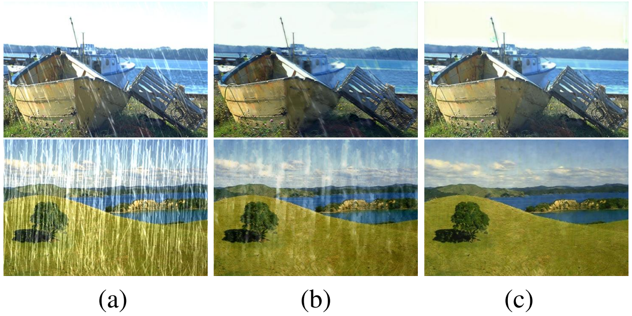

Figure 7. Results of experiments with 40% labeled data on Rain200H (a) Input rainy image (a) Using only labeled data (c) Using labeled and unlabeled data.

From these experiments, we can conclude that the proposed method can effectively leverage unlabeled data even with minimal amount of labeled training data. Additionally, only a fraction of the labeled data is sufficient to obtain performance similar to that when using 100% labeled data.

5.4. Ablation study: Hyperparameters

We also conduct a detailed ablation study to analyze the effects of different hyperparameters present in the proposed method. Due to space constraints, these results along with more qualitative visualizations are provided in supplementary material.

6. Conclusion

We presented a GP-based SSL framework to leverage unlabeled data during training for the image deraining task. We use supervised loss functions such as l_1 and the perceptual loss to train on the labeled data. For unlabeled data, we estimate the pseudo-GT at the latent space by jointly modeling the labeled and unlabeled latent space vectors using the GP. The pseudo-GT is then used to supervise for the unlabeled samples. Through extensive experiments on several datasets such as Rain800, Rain200H and DDN-SIRR, we demonstrate that the proposed method is able to achieve better generalization by leveraging unlabeled data.

References

[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010. 6

[2] H S Bhadauria and M L Dewal. Online dictionary learning for sparse coding. *In: International Conference on Machine Learning(ICML)*, pages 689–696, 2009. 3

[3] Y. Chen and C. Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *2013 IEEE International Conference on Computer Vision*, pages 1968–1975, Dec 2013. 3

[4] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3339–3348, 2018. 1, 2

[5] David Eigen, Dilip Krishnan, and Rob Fergus. Restoring an image taken through a window covered with dirt or rain. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 633–640, 2013. 3

[6] Zhiwen Fan, Huafeng Wu, Xueyang Fu, Yue Hunag, and Xinghao Ding. Residual-guide feature fusion network for single image deraining. *arXiv preprint arXiv:1804.07493*, 2018. 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 Transactions on Image Processing*, 26:2944–2956, 2017. 3

[8] 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. 3

[9] X Fu, J Huang, D Zeng, X Ding, Y Liao, and J 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, 3, 6, 7

[10] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2net: A new multi-scale backbone architecture. *arXiv preprint arXiv:1904.01169*, 2019. 4

[11] K Garg and S K Nayar. Vision and rain. *In: International Journal of Computer Vision*, 75:3–27, 2007. 3

[12] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1

[13] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. Depth-attentional features for single-image rain removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8022–8031, 2019. 3

[14] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. 2016. 4

[15] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016. 3

[16] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. 3

[17] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[18] Ruoteng Li, Loong-Fah Cheong, and Robby T Tan. Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1633–1642, 2019. 3

[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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3838–3847, 2019. 3

[20] Siyuan Li, Wenqi Ren, Jiawan Zhang, Jinke Yu, and Xiaojie Guo. Single image rain removal via a deep decomposition–composition network. *Computer Vision and Image Understanding*, 2019. 1

[21] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. *In: European Conference on Computer Vision(ECCV)*, pages 262–277, 2018. 2, 3

[22] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 254–269, 2018. 3, 6

[23] Y Li, R T Tan, X Guo, J Lu, and M S Brown. Rain streak removal using layer priors. *In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2736–2744, 2016. 1, 3

[24] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown. Rain streak removal using layer priors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2736–2744, 2016. 6

[25] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 641–656, 2018. 1

[26] Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming Guo. D3r-net: Dynamic routing residue recurrent network for video rain removal. *IEEE Transactions on Image Processing*, 28(2):699–712, 2018. 3

[27] Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming Guo. Erase or fill? deep joint recurrent rain removal and reconstruction in videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[28] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1

[29] Y Luo, Y Xu, and H Ji. Removing rain from a single image via discriminative sparse coding. *In:IEEE International*

Conference on Computer Vision(ICCV), pages 3397–3405, 2013. 3

[30] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3397–3405, 2015. 6

[31] David Martin, Charless Fowlkes, Doron Tal, Jitendra Malik, et al. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Iccv Vancouver:, 2001. 6

[32] 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

[33] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a completely blind image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2012. 7

[34] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018. 3

[35] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pages 3235–3246, 2018. 3

[36] Asanka G Perera, Yee Wei Law, and Javaan Chahl. Uav-gesture: a dataset for uav control and gesture recognition. In *European Conference on Computer Vision*, pages 117–128. Springer, 2018. 1

[37] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018. 1

[38] Rui Qian, Robby T. Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[39] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003. 4

[40] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: a better and simpler baseline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3937–3946, 2019. 2

[41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1

[42] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10:19–41, 2000. 3

[43] 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*, pages 234–241. Springer, 2015. 4, 6

[44] Varun Santhaseelan and Vijayan Asari. Utilizing local phase information to remove rain from video. *In: International Journal of Computer Vision*, 112, 2015. 3

[45] Gerald Schaefer and Michal Stich. Ucid: An uncompressed color image database. In *Storage and Retrieval Methods and Applications for Multimedia 2004*, volume 5307, pages 472–480. International Society for Optics and Photonics, 2003. 6

[46] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[47] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12270–12279, 2019. 3

[48] Y Wang, S Liu, C Chen, and B Zeng. A hierarchical approach for rain or snow removing in a single color image. *IEEE Transactions on Image Processing*, 26:3936–3950, 2017. 3

[49] Wei Wei, Deyu Meng, Qian Zhao, Zongben Xu, and Ying Wu. Semi-supervised transfer learning for image rain removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3877–3886, 2019. 2, 3, 4, 6, 7, 8

[50] Wei Wei, Lixuan Yi, Qi Xie, Qian Zhao, Deyu Meng, and Zongben Xu. Should we encode rain streaks in video as deterministic or stochastic? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2516–2525, 2017. 6

[51] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1357–1366, 2017. 1, 2, 3, 6, 8

[52] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Shuicheng Yan, and Zongming Guo. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 1

[53] Rajeev Yasarla and Vishal M Patel. Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image de-raining. *arXiv preprint arXiv:1906.11129*, 2019. 3, 6

[54] R. Yasarla and V. M. Patel. Confidence measure guided single image de-raining. *IEEE Transactions on Image Processing*, 29:4544–4555, 2020. 3

[55] Hang Zhang and Kristin Dana. Multi-style generative network for real-time transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. 4

[56] H. Zhang and Vishal M Patel. Convolutional sparse and lowrank coding-based rain streak removal. *7 IEEE Winter Conference In Applications of Computer Vision(WACV)*, pages 1259–1267, 2017. 3

[57] He Zhang and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *arXiv preprint arXiv:1701.05957*, 2017. 2, 3, 6, 8

[58] H. Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, abs/1802.07412, 2018. 1, 3, 6, 7

[59] Xiaopeng Zhang, Hao Li, Yingyi Qi, Wee Kheng Leow, and Teck Khim Ng. Rain removal in video by combining temporal and chromatic properties. *In: IEEE International Conference on Multimedia and Expo*, pages 461–464, 2006. 3

[60] L Zhu, C W Fu, D Lischinski, and P A Heng. Joint bilayer optimization for single-image rain streak removal. *In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2534, 2017. 1, 3, 6