This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2952690, IEEE Transactions on Image Processing

Semi-Supervised Image Dehazing

Lerenhan Li*, Yunlong Dong*, Wenqi Ren, Jinshan Pan, Changxin Gao, Nong Sang†, Ming-Hsuan Yang

***Abstract*—We present an effective semi-supervised learning algorithm for single image dehazing. The proposed algorithm applies a deep Convolutional Neural Network (CNN) containing a supervised learning branch and an unsupervised learning branch. In the supervised branch, the deep neural network is constrained by the supervised loss functions, which are mean squared, perceptual, and adversarial losses. In the unsupervised branch, we exploit the properties of clean images via sparsity of dark channel and gradient priors to constrain the network. We train the proposed network on both the synthetic data and real-world images in an end-to-end manner. Our analysis shows that the proposed semi-supervised learning algorithm is not limited to synthetic training datasets and can be generalized well to real-world images. Extensive experimental results demonstrate that the proposed algorithm performs favorably against the state-of-the-art single image dehazing algorithms on both benchmark datasets and real-world images.**

***Index Terms*—Image dehazing, Deep learning, Semi-supervised learning.**

I. Introduction

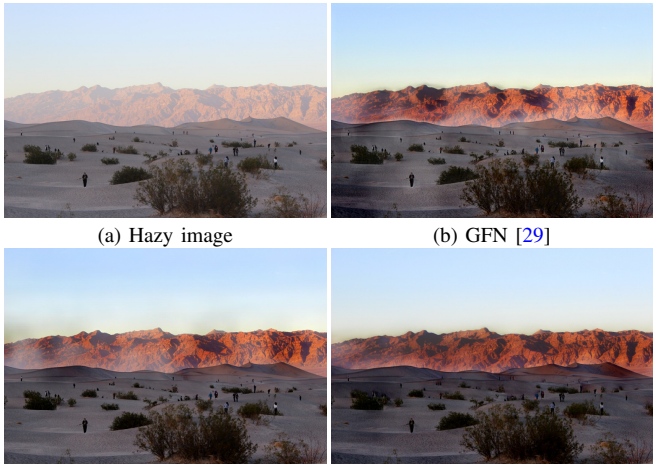
SINGLE image dehazing aims to recover the clean image from a hazy one. It has been an active research effort in the vision and graphics community due to the challenges in problem formulation, regularization, and optimization. Mathematically, the hazing process [7] can be formulated as

$$I(x) = J(x)t(x) + A(1 - t(x)), \tag{1}$$

where $I(x)$, $J(x)$, A, and $t(x)$ denote a hazy image, clean image, global atmospheric light, and a transmission map, respectively. When the haze is homogeneous, the transmission map $t(x)$ can be expressed as $t(x) = e^{-\beta d(x)}$, where $d(x)$ is the scene depth and β is the medium extinction coefficient. As only the hazy image $I(x)$ is available, the problem is ill-posed.

To make the problem well posed, existing algorithms usually make assumptions on the clean images [7], [6], e.g., the dark channel prior [7] and color-line priors [6]. As image priors often involve non-convex and non-linear terms, such approaches entail high computational loads.

To overcome this problem, deep convolutional neural networks (CNNs) have been used for image dehazing [2], [28], [43], [15], [29], [18], [30]. Typically, deep CNNs are used either to estimate transmission maps [2], [28], [43] or predict clean images directly [15], [29], [18]. These methods are efficient and outperform the hand-crafted prior based algorithms with significant performance gains. However, deep learning based approaches usually require a large number of ground-truth images for training. Most of them are trained on synthetic hazy datasets (e.g., NYU Depth dataset [37] and Make3D dataset [33], [35], [34]). As these synthetic hazy datasets contain limited image categories and image depths, the performance of existing deep learning based algorithms is usually limited to synthetic training datasets and cannot be well generalized to real-world hazy images.



(a) Hazy image (b) GFN [29] (c) Ours w/o unsupervised branch (d) Ours

Fig. 1. Dehazed results on a real-world hazy image. Our semi-supervised method generates cleaner images with fewer artifacts and color distortion.

To this end, we propose a semi-supervised learning network for image dehazing using both synthetic and real-world hazy images. Specifically, we design a deep network consisted of a supervised branch and an unsupervised branch, both of which share the weights during the training process. The supervised branch is trained on synthetic hazy images while the unsupervised one is trained on real hazy images. In the supervised branch, we apply labeled losses such as mean squared loss, perceptual loss, and adversarial loss to train the network with the difference between estimated results and ground-truths. To avoid the supervised branch over-fitting the training dataset, we exploit the properties of clear images via the dark channel (DC) [7] and image gradients such as total variation (TV) [32] to constrain the unsupervised branch. The whole network is trained on both the synthetic data and real-world images in an end-to-end manner. With the semi-supervised learning approach, our network performs favorably against the state-of-

* denotes equal contributions
† corresponding author
L. Li, Y. Dong, C. Gao, and N. Sang are with the National Key Laboratory of Science and Technology on Multi-spectral Information Processing, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, 430074, China (e-mail: lrhli@hust.edu.cn; dyl@hust.edu.cn; cgao@hust.edu.cn; nsang@hust.edu.cn).
W. Ren is with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, 100049, China (e-mail: rwq.renwenqi@gmail.com).
J. Pan is with Intelligent Media Analysis Group, School of Computer Science, Nanjing University of Science and Technology, Nanjing, 210094, China (e-mail: sdluran@gmail.com).
M.-H. Yang is with the School of Engineering, University of California at Merced, Merced, CA, 95534, USA (e-mail: mhyang@ucmerced.edu).

the-art dehazing approaches. Figure 1 shows an example on a real-world hazy image, where the proposed method generates a cleaner image.

The contributions of this work are as follows:

- We propose a semi-supervised algorithm to learn the relationship between synthetic and real-world hazy images. The proposed network consists of a supervised branch and an unsupervised branch.
- We exploit conventional image priors as unlabeled losses to train the unsupervised branch with real training data.
- We conduct extensive experiments and demonstrate that the proposed semi-supervised dehazing method performs favorably against the state-of-the-art dehazing approaches both on the synthetic datasets as well as real hazy images.

II. Related Work

In this section, we discuss the prior based and learning based single image dehazing algorithms, and semi-supervised learning methods for low-level vision tasks.

A. Prior-based Single Image Dehazing

Prior based methods focus on exploiting statistical properties of images to estimate transmission maps and atmospheric light. Tan [38] proposes a contrast-maximization image restoration method based on the observations that images with enhanced visibility (or clear-day images) have more contrast than images plagued by bad weather, and air light whose variation mainly depends on the distance of objects to the viewer, tends to be smooth. He et al. [7] develop an image restoration method by enforcing the sparsity on the dark channel of a recovered image based on the observations that the dark channel of the clean image is sparser than that of the hazy image. Several approaches have since been developed to improve efficiency and performance of image restoration based on the dark channel prior [40], [21], [19], [23], [41]. In addition, Zhu et al. [45] estimate the scene depth of the hazy images and remove the haze based on a color attenuation prior. Fattal [6] develops a color-line prior based on the observation that small image patches typically exhibit a one-dimensional distribution in the RGB color space. Similarly, Berman et al. [1] approximate the colors of a clean image by distinct chromatic properties and use them as the prior on haze-free images.

However, prior based image restoration methods usually entail solving non-convex optimization problems with computationally expensive steps. Furthermore, it may not perform well when the assumed priors do not hold for some specific scenes [7].

B. Learning-based Single Image Dehazing

Numerous deep CNN models have been proposed for low-level vision problems such as super-resolution [9], [12], denoising [20], [27], and image deblurring [17], [36], [39]. A number of methods [2], [28], [43] apply deep CNNs to estimate the transmission maps and atmospheric light first, and then recover the clean image by an element-wise division based on the degradation model in (1). Ren et al. [28] learn the mapping from hazy inputs to transmission maps using a coarse-to-fine strategy. On the other hand, Zhang and Patel [43] propose to estimate transmission maps by a densely connected pyramid network while estimating the atmospheric light via a U-Net [31]. However, these approaches may introduce artifacts and color distortion when the estimated transmission maps and atmospheric light are not accurate. To remedy this, some end-to-end methods [15], [29], [18] do not estimate the transmission map or atmospheric light, and learn to recover the clean image directly. Ren et al. [29] present a gated fusion network by fusing three images derived from the original hazy input (e.g., white balanced, contrast enhanced, and gamma corrected), which may suffer from color distortion due to pre-processing. Li et al. [18] develop a method to restore clean images by training conditional Generative Adversarial Networks (GANs).

The aforementioned learning methods use synthetic images and may not perform well on real images due to the domain gap. On the other hand, a number of algorithms apply unpaired data to train the network based on the physics model [42] or cycle GAN [4]. However, only applying the unlabeled or unpaired data is less effective than using the labeled or paired data to train the network. Different from the existing CNN based approaches, our network is trained on both synthetic and real data in a semi-supervised manner, and adapts to different image domains.

C. Semi-supervised Learning.

A few semi-supervised learning methods [11], [13] have been recently proposed to solve the low-level vision tasks. Kuznietsov et al. [11] train a deep network to predict depth maps by adding the image alignment error and regularization cost to enforce smoothness of the estimated depth maps. Without any prior assumptions in optical flow estimation, Lai et al. [13] propose a discriminator to distinguish the flow warp error between labeled and unlabeled data. In these approaches, the design of reconstruction errors in the unsupervised branches is based on the domain-specific knowledge, which cannot be directly applied to image dehazing. Thus, we propose to train the unsupervised branch with conventional dark channel and total variation loss functions.

III. Proposed Algorithm

Existing deep CNN-based image dehazing algorithms are usually developed within the supervised learning framework, which is limited to the specific synthetic training data. In this work, we address this problem with a semi-supervised learning approach. Specifically, we train a deep CNN for image dehazing using a labeled dataset $\{I_i, \hat{J}_i\}_{i=1}^{N_l}$ and an unlabeled dataset $\{I_i\}_{i=1}^{N_u}$, where N_l and N_u denote the numbers of the labeled and unlabeled training images, respectively. In addition, I_i and $\hat{J}_i$ denote the i-th hazy image and the corresponding ground-truth clean image. We train a network to learn haze-free images J from hazy inputs I:

$$J = \mathcal{G}(I), \tag{2}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2952690, IEEE Transactions on Image Processing

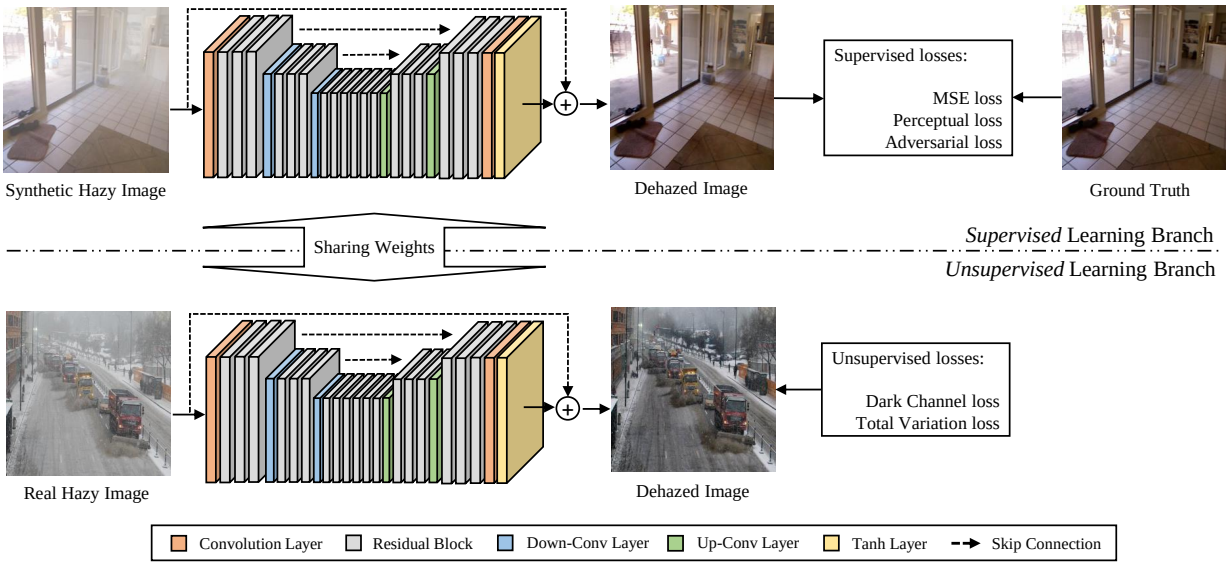


Fig. 2. Proposed semi-supervised learning framework for single image dehazing. The proposed method consists of two branches sharing the same weights. The supervised branch is trained using labeled synthetic data and loss functions based on mean squared, perceptual, and adversarial errors. The unsupervised branch is trained using unlabeled real data and loss functions based on dark channel loss and total variation.

TABLE I
CONFIGURATIONS OF THE PROPOSED NETWORK. "CONV" DENOTES THE CONVOLUTION LAYER, "RES" DENOTES THE RESIDUAL BLOCK, "UPCONV" DENOTES THE UP-SAMPLE LAYER BY TRANSPOSED CONVOLUTION OPERATOR, "TANH" DENOTES THE NON-LINEAR TANH LAYER, AND "SUM" DENOTES THE SUMMATION OPERATION. WE APPLY THE SUMMATION OPERATION AS THE SKIP CONNECTION METHOD.

Layer	Conv1	Res2-4	Conv5	Res6-8	Conv9	Res10-15	Upconv16	Res17-19	Upconv20	Res21-23	Conv24	Tanh
In_channels	3	64	64	128	128	256	256	128	128	64	64	3
Out_channels	64	64	128	128	256	256	128	128	64	64	3	3
Kernel size	7	-	5	-	3	-	4	-	4	-	7	-
Stride	1	-	2	-	2	-	2	-	2	-	1	-
Pad	3	-	1	-	1	-	1	-	1	-	3	-
Sum	-	-	-	-	-	-	Res8	-	Res4	-	-	Input

TABLE II
CONFIGURATIONS OF THE RESIDUAL BLOCKS. FOR EACH BLOCK, THE NUMBERS OF INPUT CHANNELS AND OUTPUT CHANNELS ARE THE SAME AS THOSE IN TABLE I.

Layer	Conv1	ReLU2	Conv3
Kernel size	3	-	3
Stride	1	-	1
Pad	1	-	1
Sum	-	-	Input

TABLE III
ARCHITECTURE AND CONFIGURATIONS OF THE DISCRIMINATOR. FOR EACH CONVOLUTION LAYER, THERE FOLLOWS A NON-LINEAR RELU LAYER EXCEPT FOR CONV5. WE APPLY THE INSTANCE NORMALIZATION (IN) LAYER TO NORMALIZE THE FEATURE MAPS.

Layer	Conv1	Conv2	Conv3	Conv4	Conv5
In_channels	3	64	128	256	512
Out_channels	64	128	256	512	1
Kernel size	4	4	4	4	4
Stride	2	2	2	1	1
Pad	2	2	2	2	2
Norm	-	IN	IN	IN	-

where $\mathcal{G}(\cdot)$ denotes the proposed network consisting of a supervised branch $\mathcal{G}_s$ and an unsupervised branch $\mathcal{G}_u$. Both branches share the same weights during the training. We summarize the proposed network in Figure 2 and present the detailed network architecture in the following section.

A. Network Architecture

We use an encoder-decoder architecture with skip connections which has been shown effective for low-level tasks [36], [13], [39]. We show the architecture and configurations of the proposed network in Figure 2 and Table I. The encoder contains three scales and each consists of three stacked residual blocks. Similar to the work by Nah et al. [22], we do not use any normalization layer in the residual blocks. The configurations of the residual blocks are shown in Table II. The numbers of the input and output channels of the residual blocks are the same as those in Table I.

We use the `Stride-Conv` layer to down-sample the feature maps from the previous scale by $1/2$. The decoder contains three scales and each is also stacked by three residual

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2952690, IEEE Transactions on Image Processing

blocks. We use the `Transposed-Conv` layer to up-sample the features by the factor of 2. Each convolution layer is followed by a non-linear ReLU layer except for Conv24. We skip-connect the feature maps by the summation operation. In addition, we use the residual learning to learn the difference between hazy and clean images. We analyze the effect of each component of the network in Section V-D.

For adversarial learning, we construct a discriminator by a conventional classifier, stacked by a branch of convolution, non-linear ReLU, and instance normalization layers. The details of the discriminator are shown in Table III.

B. Training Losses

We use supervised and unsupervised losses to train the corresponding branches.

1) Supervised Losses: We use the mean squared loss to ensure the predicted image J is close to the ground-truth $\hat{J}$:

$$L_c = \frac{1}{N_l}\sum_{i=1}^{N_l}\left\|\mathbf{J}_i - \hat{\mathbf{J}}_i\right\|_2, \tag{3}$$

where N_l denotes the number of labeled data in a mini-batch. In addition, $\mathbf{J}$ and $\hat{\mathbf{J}}$ represent the vector forms of the predicted image J and the corresponding ground-truth image $\hat{J}$, respectively. To generate photo-realistic images, we also use the perceptual loss based on the pre-trained VGG-19 network [8]:

$$L_p = \frac{1}{N_l}\sum_{i=1}^{N_l}\left\|\mathbf{F}_{J_i} - \mathbf{F}_{\hat{J}_i}\right\|_2, \tag{4}$$

where $\mathbf{F}_{J_i}$ and $\mathbf{F}_{\hat{J}_i}$ denote the vector forms of the feature maps w.r.t. the predicted image J and its corresponding ground-truth $\hat{J}$, respectively. The feature maps are from the `conv3-3` layer of the VGG-19 network that is pre-trained on the ImageNet [3].

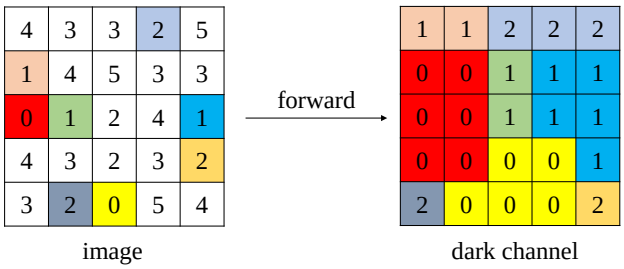
To generate sharp and visually pleasing images, we follow the GAN model [14] and build a discriminator $\mathcal{D}_{is}$ to distinguish whether an image is produced by the generator $\mathcal{G}$ (i.e., J) or from the ground-truth of labeled data (i.e., $\hat{J}$). The adversarial loss can be expressed as:

$$L_a = \mathbb{E}_{\hat{J}}\left[\log \mathcal{D}_{is}(\hat{J})\right] + \mathbb{E}_J\left[\log(1 - \mathcal{D}_{is}(J))\right]. \tag{5}$$

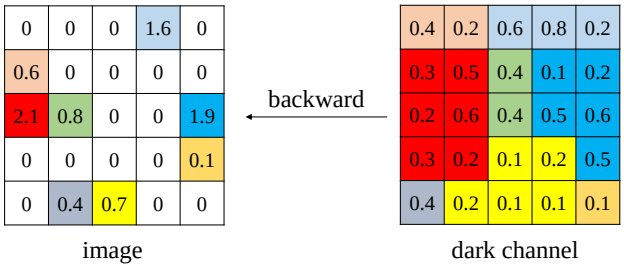
2) Unsupervised Losses: We use the total variation and dark channel losses to enforce the unsupervised branch to generate images that have the same statistical properties as clean images. The total variation loss, an ℓ_1-regularization gradient prior on the predicted images by the unsupervised branch, is applied to preserve structures and details:

$$L_t = \frac{1}{N_u}\sum_{i=1}^{N_u}(\|\nabla_h \mathbf{J}_i\|_1 + \|\nabla_v \mathbf{J}_i\|_1), \tag{6}$$

where ∇_h and ∇_v represent the horizontal and vertical differential operation matrices, respectively.



(a) `forward` of $D(\cdot)$



(b) `backward` of $D(\cdot)$

Fig. 3. A toy example of implementing the `forward` and `backward` steps for the dark channel operation [24]. In the `forward` pass, the numbers in the matrix represent pixels values in the image. We compute the dark channel by finding the minimum at each overlapped patch and replacing the central pixel with the minimum. In the `backward` pass, the numbers denote the propagated gradient values. We collect the gradients from the corresponding pixels. We mark the pixels sharing the same dark pixel value with the same color for illustration purpose.

The dark channel of clean images [7], [24] has been shown to be sparser than that of the hazy ones. It can be expressed by:

$$D(I) = \min_{y\in N(x)}\left[\min_{c\in\{r,g,b\}} I^c(y)\right], \tag{7}$$

where x and y are pixel coordinates, I^c denotes c-th color channel, and $N(x)$ is an image patch centered at x. Motivated by this, we apply an ℓ_1-regularization to constrain the sparsity of the dark channel of the predicted images:

$$L_d = \frac{1}{N_u}\sum_{i=1}^{N_u}\|\mathbf{D}_{J_i}\|_1, \tag{8}$$

where $\mathbf{D}_{J_i}$ denotes the vector form of the dark channel of the predicted image J_i.

Although the dark channel has been shown effective to remove the haze by adding the constraint on the clean image [7], it is challenging to embed into the learning network due to the highly non-convex and non-linear term. We apply the look-up table scheme [24] to implement the `forward` and `backward` step of the dark channel operation. Figure 3 shows a toy example on how the operations are carried out using the look-up table scheme.

We apply a matrix of 5×5 to represent a single channel image. Based on (7), in the `forward` stage, the dark channel of the image is computed as:

$$D(I) = \min_{y\in N(x)}\left[I(y)\right], \tag{9}$$

where the size of the patch $N(y)$ is set as 3×3. As shown in Figure 3(a), the dark pixels of each patch are marked by

Algorithm 1 Training details at each iteration.

Input: Initialized generator $\mathcal{G} = \{\mathcal{G}_s, \mathcal{G}_u\}$, Synthetic training data N_l, Real training data from N_u
Output: Updated generator $\mathcal{G}$.

while $i < \text{iter}_{\max}$ **do**
  randomly choose hazy/clean image pair $I_s/\hat{J}$ from N_l
  randomly choose hazy image I_r from N_u
  obtain J_s by $J_s = \mathcal{G}(I_s)$
  obtain J_r by $J_r = \mathcal{G}(I_r)$
  obtain L_c by (3) from $\{J_s, \hat{J}\}$
  obtain L_p by (4) from $\{J_s, \hat{J}\}$
  obtain L_a by (5) from $\{J_s, \hat{J}\}$
  obtain L_d by (8) from J_r
  obtain L_t by (6) from J_r
  `back propagate` $\mathcal{G}_s$ by $L_c + L_p + L_a$
  `back propagate` $\mathcal{G}_u$ by $L_d + L_t$
  $i \leftarrow i + 1$
end while

different colors. We pad the image repeatedly to handle the boundary issues. In the `backward` stage, the dark pixels of the image collect the propagating gradients of the corresponding pixels by summation.

3) Overall Loss Function: We combine supervised losses, unsupervised losses, and the adversarial loss to train the proposed network:

$$L = L_c + \lambda L_p + \gamma L_t + \mu L_d + \eta L_a, \tag{10}$$

where λ, γ, μ, and η are the positive weights of each loss function.

C. Semi-Supervised Training Details

As the supervised branch and unsupervised branch share the same architecture as well as weights, we update weights iteratively during the training process. We first randomly choose a batch of N_l labeled samples and compute the difference between dehazed images and ground-truths by labeled losses. Meanwhile, we randomly choose a batch of N_u unlabeled samples and compute the unlabeled losses. We then use the labeled/unlabeled losses to update the parameters of the supervised/unsupervised branch by back propagation, respectively. We present the training details when updating the generator in Algorithm 1.

IV. Experimental Results

A. Implementation Details

We alternatively update the generator and discriminator by updating one while fixing the other. More specifically, we update the discriminator once after updating the generator five times. When updating the generator, we optimize the network parameters in a semi-supervised way. We use the Pytorch toolbox [25] and Adam [10] solver to optimize both the generator and discriminator. We set $\beta_1 = 0.9$, $\beta_2 = 0.99$, and the weight decay as 10^{-4}. The network is trained for 300 epochs. The learning rate is set to be 10^{-4} at the first 150 epochs, and decreased linearly to 10^{-6} within the following 150 epochs by $l_r = 10^{-4} - \frac{10^{-4}-10^{-6}}{150}(E - 150)$, where E denotes the number of the training epoch.

TABLE IV
Quantitative evaluations (PSNR/SSIM) on three benchmark datasets [16]. Our method performs favorably against the state-of-the-art dehazing algorithms.

(PSNR/SSIM)	RESIDE-C	HazeRD	SOTS
Haze	14.00/0.59	14.01/0.39	13.95/0.64
DCP [7]	14.60/0.75	14.01/0.39	15.49/0.64
MSCNN [28]	17.54/0.81	15.57/0.42	17.57/0.81
DehazeNet [2]	20.06/0.83	15.54/0.41	21.14/0.85
AOD-Net [15]	18.33/0.82	15.63/0.45	19.06/0.85
DCPDN [43]	19.19/0.82	16.12/0.34	19.39/0.65
GFN [29]	19.85/0.80	13.98/0.37	22.30/0.88
PDN [41]	18.93/0.84	15.21/0.43	19.69/0.82
CycleGAN [4]	17.33/0.57	15.64/0.43	17.78/0.72
baseline	**23.54**/0.90	15.48/0.42	24.33/0.88
Ours	23.15/**0.91**	**16.55/0.47**	**24.44/0.89**

We train the network by randomly choosing both labeled and unlabeled samples from the RESIDE dataset [16], which contains the ITS (Indoor Training Set), OTS (Outdoor Training Set), SOTS (Synthetic Object Testing Set), URHI (Unlabeled real Hazy Images), and RTTS (real Task-driven Testing Set). For labeled data, we select 4000 synthetic hazy images, 2000 from the ITS set and 2000 from the OTS set. For unlabeled data, we randomly choose 2000 real hazy images from the URHI dataset. We set the batch size to 4, and apply the following strategies to randomly augment the training data: 1) flipping horizontally and vertically, 2) rotating for $-90°$ or $90°$, and 3) adding Gaussian noise with the sigma of 0.01. Then we randomly crop the images to the size of 256×256 and normalize the pixel values to $[-1, 1]$.

We set the patch size as 35×35 when computing the DC loss. The loss weights are set as: $\lambda = 10^{-2}$, $\gamma = 10^{-5}$, $\mu = 10^{-5}$, and $\eta = 10^{-3}$. We train our network on an Nvidia GTX 1080 GPU and it takes three days to converge. The source code and pre-trained model will be made publicly available on the project website: https://sites.google.com/view/lerenhanli/homepage/semi_su_dehazing.

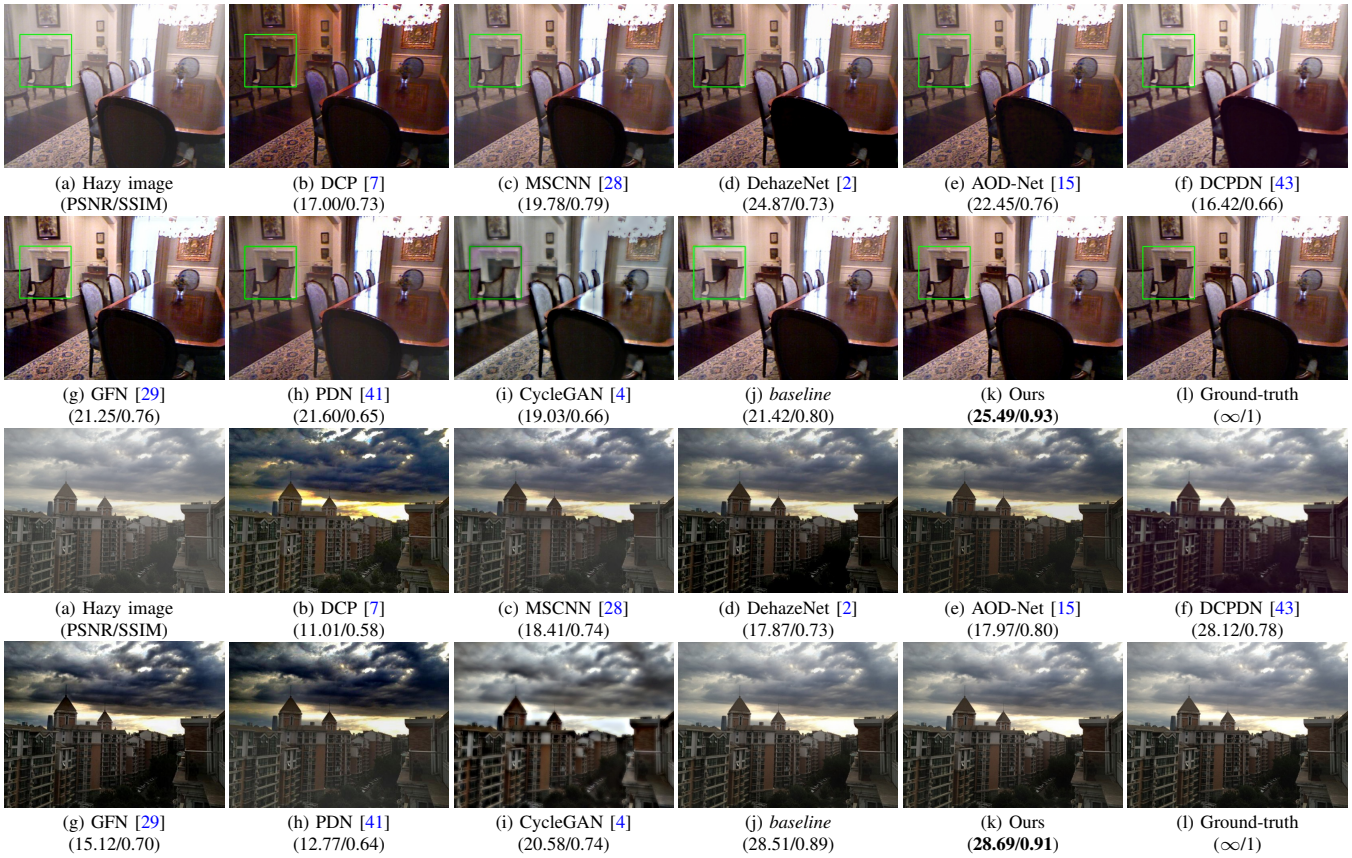
B. Evaluation Settings

We evaluate the performance of the proposed method against the state-of-the-art dehazing approaches including DCP [7], MSCNN [28], DehazeNet [2], AOD-Net [15], DCPDN [43], GFN [29], CycleGAN [4], and PDN [41]. To better understand the proposed semi-supervised method, we retrain a network which only contains the supervised branch as the *baseline* model. For fair comparisons in the loss function, we apply all the proposed loss function (10) on the labeled data including supervised and unsupervised losses.

C. Evaluations on Synthetic Datasets

We use three benchmark datasets [16], [44] to evaluate the proposed method, including RESIDE-C, HazeRD, and SOTS datasets. The RESIDE-C dataset contains 100 indoor and 100 outdoor synthetic hazy images randomly chosen from the ITS

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2952690, IEEE Transactions on Image Processing



(a) Hazy image (PSNR/SSIM) | (b) DCP [7] (17.00/0.73) | (c) MSCNN [28] (19.78/0.79) | (d) DehazeNet [2] (24.87/0.73) | (e) AOD-Net [15] (22.45/0.76) | (f) DCPDN [43] (16.42/0.66)

(g) GFN [29] (21.25/0.76) | (h) PDN [41] (21.60/0.65) | (i) CycleGAN [4] (19.03/0.66) | (j) *baseline* (21.42/0.80) | (k) Ours (**25.49/0.93**) | (l) Ground-truth (∞/1)

(a) Hazy image (PSNR/SSIM) | (b) DCP [7] (11.01/0.58) | (c) MSCNN [28] (18.41/0.74) | (d) DehazeNet [2] (17.87/0.73) | (e) AOD-Net [15] (17.97/0.80) | (f) DCPDN [43] (28.12/0.78)

(g) GFN [29] (15.12/0.70) | (h) PDN [41] (12.77/0.64) | (i) CycleGAN [4] (20.58/0.74) | (j) *baseline* (28.51/0.89) | (k) Ours (**28.69/0.91**) | (l) Ground-truth (∞/1)

Fig. 4. Dehazed results on the RESIDE-C [16] dataset. Our method generates cleaner results with less artifacts and color distortion.

(a) Hazy image | (b) DCPDN [43] | (c) GFN [29] | (d) PDN [41] | (e) *baseline* | (f) Ours

Fig. 5. A challenging dehazing example from the HazeRD [44] dataset. Our result looks more pleasing and cleaner than the others. (In the dataset, "Sky" is masked before generating the hazy images.)

and OTS databases. We note the selected images do not appear in our training data. As shown in Figure 4 and Table IV, the proposed method generates results with cleaner structures and details.

We then evaluate our method on the HazeRD [44] and SOTS [16] datasets. As shown in Figure 5-6 and Table IV, our algorithm performs favorably against the state-of-the-art dehazing methods.

We note that the network trained only with unpaired data [4] does not perform well due to the limited learning capability on the unlabeled data. Furthermore, the *baseline* model (i.e., purely supervised) performs slightly better than the proposed semi-supervised model in terms of PSNR on the RESIDE-C dataset. However, the *baseline* does not perform as robustly as the proposed algorithm on each synthetic dataset. The proposed semi-supervised method can be well generalized to the images where the image categories and scenes are different from the training dataset.

D. Evaluations on Real Images

We evaluate the proposed method against the state-of-the-art approaches on real hazy images. Figure 7 shows that our method recovers cleaner and visually more pleasing images than the state-of-the-art approaches. To better understand the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2952690, IEEE Transactions on Image Processing

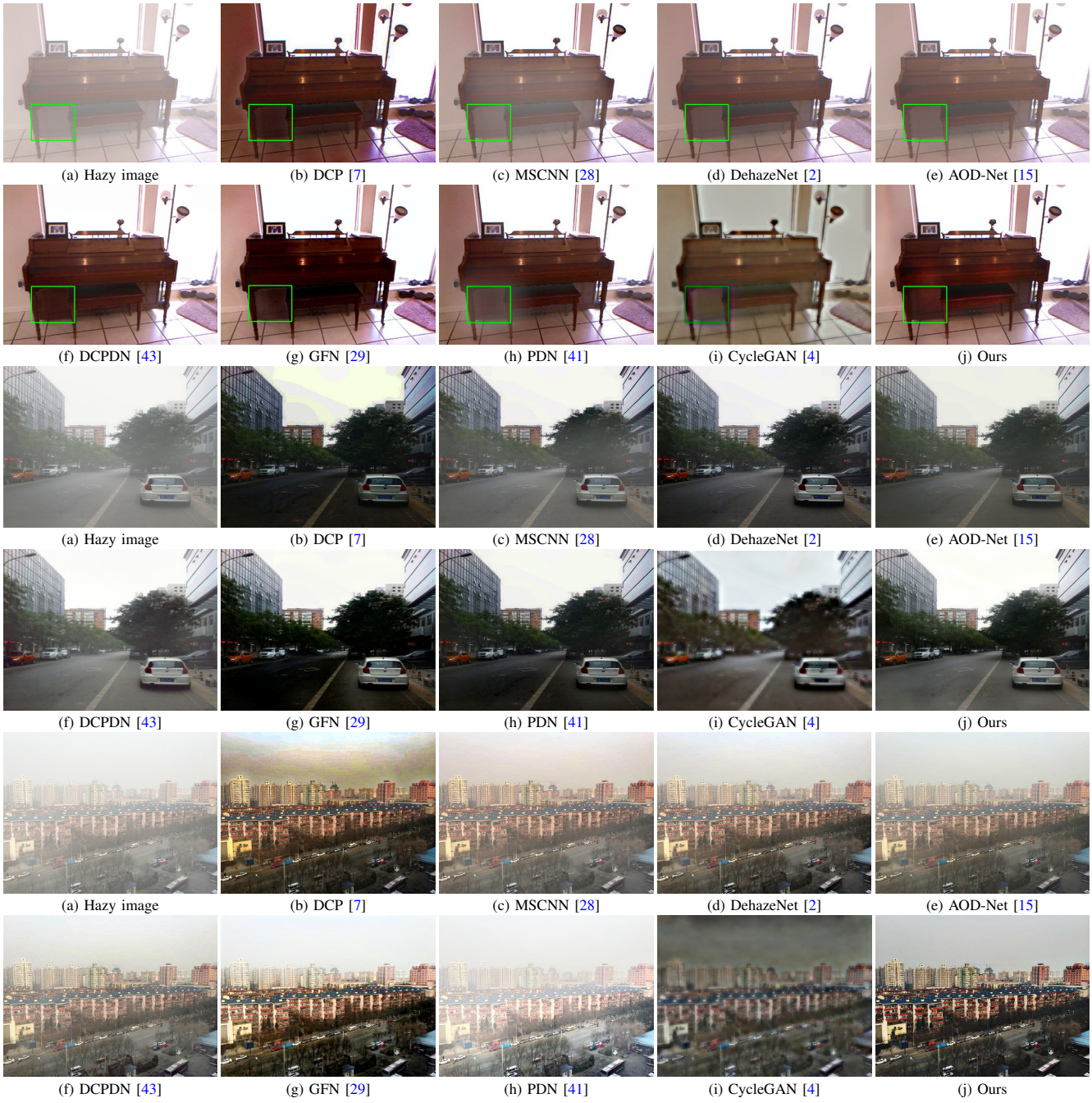


Fig. 6. Dehazed results on the SOTS [16] dataset. Our results are cleaner and have less color distortion.

performance of our method on real images, we apply a task-driven evaluation presented by Li et al. [16]. We test our method on the RTTS [16] dataset, which contains 4322 real-world images annotated with object categories and bounding boxes. After restoring the clean images, we apply a pre-trained Faster R-CNN [26] to detect objects of interests, and compute the mean Average Precision (mAP) of each method. Table V shows the proposed method performs favorably against the other approaches for object detection on the RTTS dataset.

The proposed method outperforms the *baseline* (i.e., purely supervised) model in both visual and task-driven evaluations. This shows that the proposed semi-supervised method is effective in learning the domain gap between synthetic data and real-world images, thus alleviating the over-fitting problems.

E. Run Time

We evaluate the run time of the proposed algorithm with comparisons to the other approaches. We randomly sample 100 hazy images of 512×512 pixels and compute the average processing time of each method. All the methods are carried out on a desktop computer with an Intel(R) Xeon(R) CPU E5-2670 v3@2.30GHz, 32 GB RAM, and an Nvidia GTX 1080 GPU. As DCP is based on a conventional optimization method, we evaluate the execution time on the CPU without any GPU acceleration. As shown in Table VI, our method performs competitively with state-of-the-art approaches.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2952690, IEEE Transactions on Image Processing

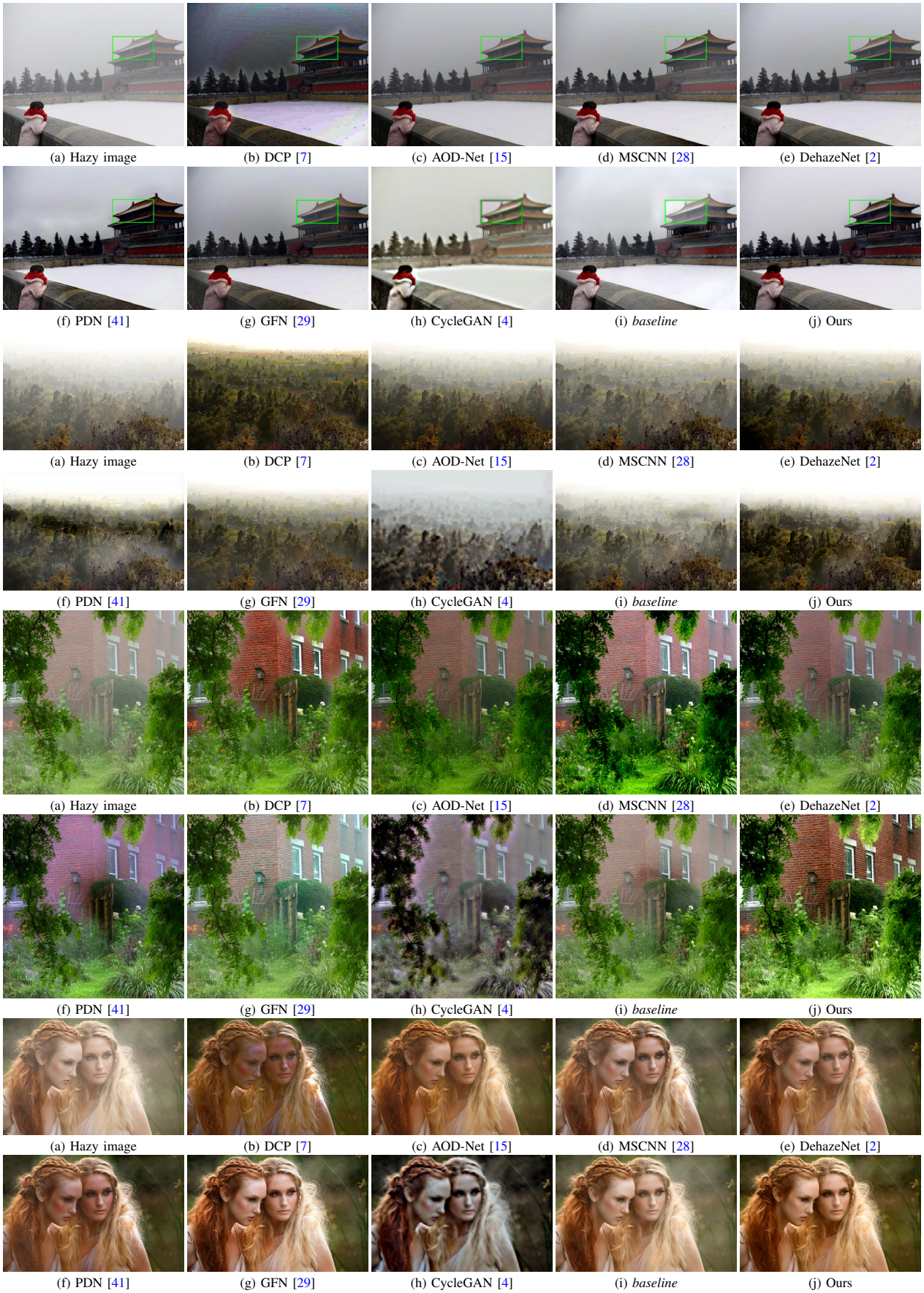


Fig. 7. Dehazed results on real hazy images. Our results are cleaner and look more pleasing.

(a) Hazy image (b) Ours w/o unlabeled data (c) Ours w/ 500 unlabeled data (d) Ours w/ 1000 unlabeled data (e) Ours w/ 2000 unlabeled data

Fig. 8. Sensitivity to the amount of unlabeled data. With the increasing of the unlabeled data, the result looks more pleasing.

(a) Hazy image (b) L_c only (c) L_p only (d) $L_c + L_p$ (e) $L_p + L_c + L_a$

Fig. 9. Visual comparisons on different labeled losses. The combination of all labeled losses leads to a better result.

(a) Hazy image (b) w/o L_d nor L_t (c) L_d only (d) L_t only (e) $L_d + L_t$

Fig. 10. Visual comparisons on different unlabeled loss functions. The combination of DC and TV losses leads to a better result.

TABLE V
OBJECT DETECTION RESULTS ON THE RTTS [16] DATASET. WE APPLY FASTER R-CNN TO DETECT OBJECTS OF INTERESTS ON DEHAZED IMAGES. FASTER R-CNN IS TRAINED ON THE VOC2007 [5] DATASET. THE DETECTION TASK FAVORS THE PROPOSED METHOD MOST AMONG THE OTHER ALGORITHMS.

	mAP (%)
Hazy	37.58
DCP [7]	40.58
MSCNN [28]	41.34
DehazeNet [2]	40.54
AOD-Net [15]	37.47
GFN [29]	58.11
DCPDN [43]	61.28
PDN [41]	62.30
CycleGAN [4]	42.53
baseline	55.27
Ours	**62.61**

V. ANALYSIS AND DISCUSSIONS

We analyze the proposed method with ablation studies in this section.

A. Semi-Supervised Learning

As our network architecture includes a supervised branch as the *baseline*, it is of great interest to understand how the semi-supervised learning formulation facilitates dehazing the images. As shown in Section IV, the proposed semi-supervised learning method helps to remove haze from real-world images while the baseline method with purely supervised learning is less effective (Table V and Figure 7). We note that the method with supervised learning can generate clean images on the synthetic test dataset which is generated in the same way as the training dataset. However, this method does not effectively remove haze from real images. This confirms our analysis that the method with supervised learning is usually limited to specific synthetic training datasets. To better understand the effectiveness of semi-supervised learning, we train the proposed network on different number of the labeled samples and evaluate it against the *baseline* model. Specifically, when training the supervised branch network, we use 4000, 2000, and 1000 labeled samples and use the same number (i.e., 2000) of unlabeled samples to train the unsupervised branch. For fair comparisons, we use 2000 labeled samples to train an unsupervised branch in the *baseline* model. For evaluation on dehazing real images, we use the RTTS [16] dataset to compute the mAP (%) on the dehazed results of each model. As shown in Table VII, the proposed semi-supervised method performs more robustly to the number of the labeled samples than the *baseline* model. These results also show the effect of using unlabeled samples in the proposed algorithm.

Furthermore, we evaluate the sensitivity to the amount of unlabeled data. Specifically, when training the unsupervised branch, we apply 0, 500, 1000, and 2000 unlabeled samples and use the same amount (i.e., 4000) of labeled samples to train the supervised branch. We note that the proposed method training without unlabeled data degrades to a supervised

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2952690, IEEE Transactions on Image Processing

TABLE VI
AVERAGE RUNTIME (SECONDS) FOR AN IMAGE WITH THE SIZE OF 512×512 PIXELS. WE APPLY THE PUBLICLY SOURCE CODE OF ALL THE METHODS. ALL THE METHODS ARE CARRIED OUT ON A DESKTOP COMPUTER WITH AN INTEL(R) XEON(R) CPU E5-2670 V3@2.30GHZ, 32 GB RAM, AND AN NVIDIA GTX 1080 GPU.

	Platform	Run time
DCP [7]	Matlab (CPU)	1.30
MSCNN [28]	MatConvnet (GPU)	3.21
DehazeNet [2]	Matlab (GPU)	1.85
AOD-Net [15]	Caffe (GPU)	0.40
DCPDN [43]	Pytorch (GPU)	1.53
GFN [29]	Matlab (GPU)	5.67
PDN [41]	Matlab (GPU)	7.32
CycleGAN [4]	TensorFlow (GPU)	2.51
Ours	Pytorch (GPU)	1.12

TABLE VII
TASK-DRIVEN EVALUATIONS (MAP, %) ON THE RTTS DATASET WITH DIFFERENT AMOUNT OF LABELED DATA. THE PROPOSED SEMI-SUPERVISED METHOD PERFORMS MORE ROBUSTLY THAN THE *baseline* MODEL.

Amount of the labeled data	Purely supervised	Semi-supervised
4000	58.60	62.61
2000	50.12	60.31
1000	45.36	60.26

learning approach. We quantitatively evaluate the proposed method on the synthetic dataset (i.e., SOTS) and the real-world image dataset (i.e., RTTS). As shown in Table VIII, the proposed model performs similarly on the synthetic dataset while the performance on the real dataset becomes better with the amount of unlabeled data increasing. Figure 8 shows the visual results with the different amount of unlabeled training data. The proposed network performs less effectively when training with fewer unlabeled samples.

B. Supervised Losses

We analyze the effect of each supervised loss on the SOTS dataset. We fix the unlabeled losses (i.e., TV loss and DC loss) in the unsupervised branch and evaluate the proposed method using three labeled losses. As shown in Table IX and Figure 9, the method using the combination of MSE, perceptual, and adversarial losses performs well against the other alternatives. Solely using L_c or L_p in the proposed model introduces over-smooth results (Figure 9(b)) and checkerboard artifacts (Figure 9(c)), respectively. Using L_a in the proposed method facilitates generating the results visually more pleasing and closer to real images.

C. Unsupervised Losses

As the performance of the unsupervised learning branch is constrained by the loss functions, we analyze the effect of each component. For fair comparisons, we fix the labeled loss functions in the supervised branch and evaluate the effect of each unlabeled loss by removing one while keeping the other one. We retrain the network with the same training dataset as

TABLE VIII
QUANTITATIVE EVALUATIONS WITH DIFFERENT AMOUNT OF LABELED DATA. THE PERFORMANCE ON THE SYNTHETIC DATASET ARE ROBUST WHILE THE PERFORMANCE ON THE REAL DATASET ARE SENSITIVE TO THE UNLABELED DATA.

Amount of the unlabeled data	SOTS (PSNR/SSIM)	RTTS (mAP, %)
0	23.65/0.86	53.48
500	24.37/0.88	58.97
1000	24.41/0.89	60.79
2000	24.44/0.89	62.61

TABLE IX
QUANTITATIVE RESULTS ON DIFFERENT LABELED LOSS FUNCTIONS. THE COMBINATION OF ALL THE LABELED LOSSES PERFORMS WELL AGAINST THE OTHER ALTERNATIVES.

	PSNR	SSIM
L_c only	23.99	0.79
L_p only	21.56	0.67
$L_p + L_c$	24.43	0.81
$L_p + L_c + L_a$	**24.44**	**0.89**

presented in Algorithm 1 and quantitatively evaluate the performance on the SOTS and RTTS dataset. Table X shows that the combination of L_d and L_t generates better results. Solely using L_d or L_t tends to introduce undesirable artifacts in the dehazed images as shown in Figure 10. More specifically, only using L_d does not generate clean images since simply constraining the sparsity on the dark channel tends to generate artifacts with dark pixels as shown in Figure 10(b). On the other hand, the scheme of only using L_t is less effective as this constraint smooths image details. Although the quantitative results decrease, they are still comparable with state-of-the-art dehazing approaches. We note that the model without using any loss function in the unsupervised branch is degraded to a supervised model, which is the same as the one in Table VIII of Section V-A. Our *baseline* model performs slightly lower than the full model in terms of PSNR and SSIM. The main reason is that we apply the proposed unlabeled losses on the supervised branch, which also demonstrates the effectiveness of the proposed unlabeled loss functions.

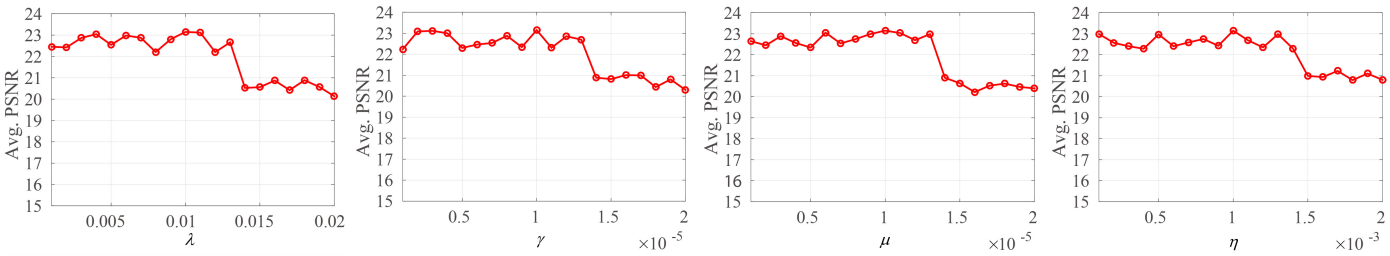
D. Hyper-Parameters

There are four hyper-parameters in the proposed method (λ, γ, μ, and η in the loss function (10)). For sensitivity analysis, we evaluate the proposed model by varying one hyper-parameter while fixing the others. We retrain the proposed network with the same training data as mentioned in Section IV-A and compute the PSNR values on the RESIDE-C dataset. The results in Figure 11 demonstrate that the proposed method is insensitive to changes in these parameters within a sensible range.

E. Network Architecture

To better understand the effect of each component in the proposed network, we conduct several ablation studies to analyze several design choices. Specifically, we compare

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2952690, IEEE Transactions on Image Processing



(a) Effect of λ (b) Effect of γ (c) Effect of μ (d) Effect of η

Fig. 11. Sensitivity analysis on the four positive weights, λ, γ, μ, and η in the loss function. The proposed method is insensitive to changes in these hyper-parameters within a sensible range.

(a) Hazy image (b) GFN [29] (c) DCPDN [43] (d) Ours

Fig. 12. Visual results on a noisy hazy image (noise level: 10%). The proposed method can remove the haze as well as image noise.

TABLE X
QUANTITATIVE RESULTS ON DIFFERENT UNLABELED LOSS FUNCTIONS. WE EVALUATE THE PSNR AND SSIM ON THE SOTS DATASET AND MAP ON THE RTTS DATASET. THE COMBINATION OF L_d AND L_t LEADS TO THE BETTER RESULTS.

	PSNR	SSIM	mAP (%)
w/o L_d nor L_t	23.65	0.86	53.48
baseline model	24.33	0.88	55.27
L_d only	20.65	0.71	62.34
L_t only	21.30	0.75	61.23
$L_d + L_t$	**24.44**	**0.89**	**62.61**

TABLE XI
ANALYSIS ON DIFFERENT TRAINING STRATEGIES. THE PROPOSED MODEL PERFORMS WELL AGAINST THE OTHER ALTERNATIVES.

	PSNR	SSIM
w/o skip connection	21.17	0.83
w/o residual learning	22.32	0.87
skip conn. via concat.	23.01	0.88
our full model	**23.15**	**0.91**

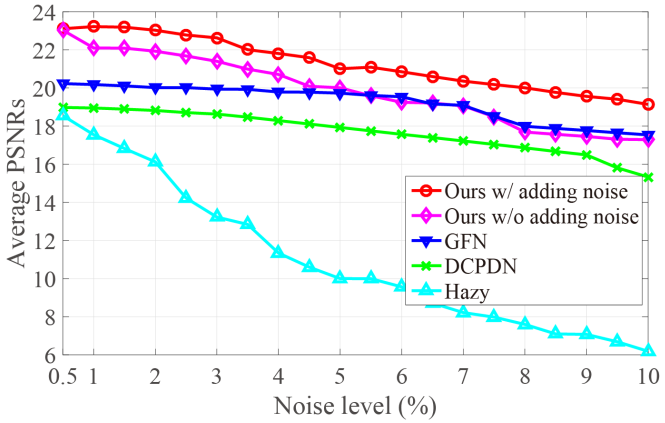


Fig. 13. Quantitative evaluations in terms of PSNR on different noise levels. Our method consistently performs better than the other algorithms [29], [43] at each noise level.

the proposed network with the following models: i) without skip connections between the encoder and decoder, ii) without residual learning, iii) using concatenation instead of summation to skip connect the encoder and the decoder. Table XI shows that all these network designs are crucial to the performance of the proposed model. The model with skip connections reuses the features from the encoder, which facilitates the training on deep networks. The residual learning ensures the network focuses on predicting the details instead of the pixel values. Finally, the proposed model performs slightly better than the scheme using concatenation as the skip connection.

F. Additive Noise

We use the RESIDE-C dataset to evaluate the proposed method against noise by adding random Gaussian noise of 0.5% to 10% to all the test images. We note that DCPDN [43] does not apply noise for data augmentation during the training process. In order to compare our method with DCPDN fairly, we retrain the proposed model without adding noise in the training data. As shown in Figure 13, the proposed method consistently performs well even when the noise level is high. Furthermore, adding noise to the training data can improve the performance of handling the image noise. Figure 12 shows that our method can remove the image haze and noise while state-of-the-art deahzing algorithms [29], [43] are less effective.

(a) Hazy image (b) DCP [7]

(c) GFN [29] (d) Ours

Fig. 14. Limitations on a severe hazy image. The propose method cannot effectively recover the structures and details when the image suffers from the severe haze.

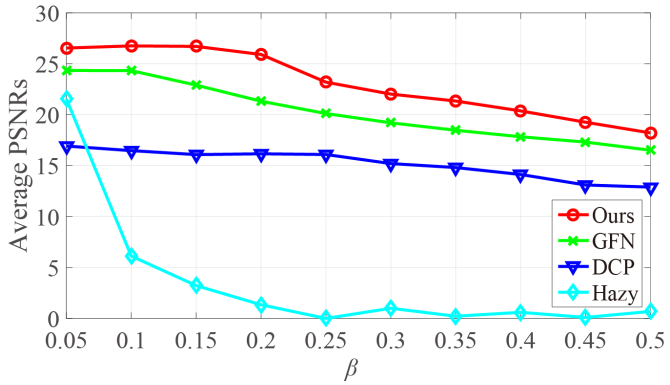


Fig. 15. Performance analysis on different haze concentration. Our method becomes less effective when the images suffer from severe haze.

G. Limitations

Figure 14 shows that our approach performs less effective when the image suffers from severe haze. In order to systematically understand the performance of the proposed method under different concentrations of image haze, we randomly select 10 clean images from the SOTS dataset and generate synthetic hazy images by the atmospheric model (1). We generate different haze concentrations by using different β (from 0.05 to 0.5 with an interval of 0.05). Then, we evaluate the performance of the proposed method and compare it with the state-of-the-art approaches (DCP [7] and GFN [29]) on 100 synthesized hazy images in terms of PSNR. As shown in Figure 15, although all the methods do not perform robustly with the increasing of hazy concentration, the proposed semi-supervised method obtains higher PSNR values than the competitors. Our future work will focus on removing severe haze by transferring sharp scenes into hazy images from clean pictures.

VI. Conclusions

In this work, we propose a novel semi-supervised learning algorithm for single image dehazing as the feature domains of the synthetic and real-world images are different. On the one hand, we use labeled loss functions to train the supervised branch on the synthetic data with ground-truth labels. On the other hand, we train the unsupervised branch with real data and unlabeled loss functions based on commonly used image priors including dark channel and total variation. Extensive experimental results demonstrate that the proposed algorithm performs favorably against the state-of-the-art dehazing methods both on synthetic and real hazy images.

Acknowledgment

This work is partially supported by the National Natural Science Foundation of China (Nos. 61433007, 61901184, 61872421, and 61922043), the Natural Science Foundation of Jiangsu Province (No. BK20180471) , and the National Science Foundation CAREER (No. 1149783). Wenqi Ren is supported in part by the CCF-DiDi GAIA (YF20180101) and Zhejiang Labs International Talent Fund for Young Professionals. Lerenhan Li is supported by a scholarship from China Scholarship Council.

References

[1] D. Berman, S. Avidan, et al. Non-local image dehazing. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[2] B. Cai, X. Xu, K. Jia, Q. Chunmei, and D. Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 2016. 1, 2, 5, 6, 7, 8, 9, 10

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 4

[4] D. Engin, A. Genç, and H. K. Ekenel. Cycle-dehaze: Enhanced cyclegan for single image dehazing. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018. 2, 5, 6, 7, 8, 9, 10

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. 9

[6] R. Fattal. Dehazing using color-lines. *ACM Transactions on Graphics*, 2014. 1, 2

[7] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011. 1, 2, 4, 5, 6, 7, 8, 9, 10, 12

[8] J. Justin, A. Alexandre, and F.-F. Li. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016. 4

[9] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[10] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv*, 2014. 5

[11] Y. Kuznietsov, J. Stückler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[12] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 2

[13] W.-S. Lai, J.-B. Huang, and M.-H. Yang. Semi-supervised learning for optical flow with generative adversarial networks. In *Neural Information Processing Systems*, 2017. 2, 3

[14] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 4

[15] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng. Aod-net: All-in-one dehazing network. In *IEEE International Conference on Computer Vision*, 2017. 1, 2, 5, 6, 7, 8, 9, 10

[16] B. Li, W. Ren, D. Fu, D. Tao, D. Feng, W. Zeng, and Z. Wang. Benchmarking single-image dehazing and beyond. *IEEE Transactions on Image Processing*, 2019. 5, 6, 7, 9

[17] L. Li, J. Pan, W.-S. Lai, C. Gao, N. Sang, and M.-H. Yang. Learning a discriminative prior for blind image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[18] R. Li, J. Pan, Z. Li, and J. Tang. Single image dehazing via conditional generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2

[19] Y. Li, R. T. Tan, and M. S. Brown. Nighttime haze removal with glow and multiple light colors. In *IEEE International Conference on Computer Vision*, 2015. 2

[20] X. Mao, C. Shen, and Y.-B. Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Neural Information Processing Systems*, 2016. 2

[21] G. Meng, Y. Wang, J. Duan, S. Xiang, and C. Pan. Efficient image dehazing with boundary constraint and contextual regularization. In *IEEE International Conference on Computer Vision*, 2013. 2

[22] S. Nah, T. Hyun Kim, and K. Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

[23] K. Nishino, L. Kratz, and S. Lombardi. Bayesian defogging. *International Journal of Computer Vision*, 2012. 2

[24] J. Pan, D. Sun, H. Pfister, and M.-H. Yang. Blind image deblurring using dark channel prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 4

[25] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *Neural Information Processing Systems Workshops*, 2017. 5

[26] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems*, 2015. 7

[27] W. Ren, S. Liu, L. Ma, Q. Xu, X. Xu, X. Cao, J. Du, and M.-H. Yang. Low-light image enhancement via a deep hybrid network. *IEEE Transactions on Image Processing*, 2019. 2

[28] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang. Single image dehazing via multi-scale convolutional neural networks. In *European Conference on Computer Vision*, 2016. 1, 2, 5, 6, 7, 8, 9, 10

[29] W. Ren, L. Ma, J. Zhang, J. Pan, X. Cao, W. Liu, and M.-H. Yang. Gated fusion network for single image dehazing. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 5, 6, 7, 8, 9, 10, 11, 12

[30] W. Ren, J. Zhang, X. Xu, L. Ma, X. Cao, G. Meng, and W. Liu. Deep video dehazing with semantic segmentation. *IEEE Transactions on Image Processing*, 2018. 1

[31] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, 2015. 2

[32] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 1992. 1

[33] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Neural Information Processing Systems*, 2006. 1

[34] A. Saxena, S. H. Chung, and A. Y. Ng. 3-d depth reconstruction from a single still image. *International Journal of Computer Vision*, 2008. 1

[35] A. Saxena, M. Sun, and A. Y. Ng. Learning 3-d scene structure from a single still image. In *IEEE International Conference on Computer Vision*, 2007. 1

[36] Z. Shen, W.-S. Lai, T. Xu, J. Kautz, and M.-H. Yang. Deep semantic face deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2, 3

[37] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, 2012. 1

[38] R. T. Tan. Visibility in bad weather from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 2

[39] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia. Scale-recurrent network for deep image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2, 3

[40] J.-P. Tarel and N. Hautiere. Fast visibility restoration from a single color or gray level image. In *IEEE International Conference on Computer Vision*, 2009. 2

[41] D. Yang and J. Sun. Proximal dehaze-net: A prior learning-based deep network for single image dehazing. In *European Conference on Computer Vision*, 2018. 2, 5, 6, 7, 8, 9, 10

[42] X. Yang, Z. Xu, and J. Luo. Towards perceptual image dehazing by physics-based disentanglement and adversarial training. In *AAAI Conference on Artificial Intelligence*, 2018. 2

[43] H. Zhang and V. M. Patel. Densely connected pyramid dehazing network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 5, 6, 7, 9, 10, 11, 12

[44] Y. Zhang, L. Ding, and G. Sharma. Hazerd: an outdoor scene dataset and benchmark for single image dehazing. In *IEEE International Conference on Image Processing*, 2017. 5, 6

[45] Q. Zhu, J. Mai, L. Shao, et al. A fast single image haze removal algorithm using color attenuation prior. *IEEE Transactions on Image Processing*, 2015. 2

Lerenhan Li is currently a Ph.D. candidate in the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, supervised by Prof. Nong Sang. He was a joint-training Ph.D. student in the Department of Electrical Engineering and Computer Science, University of California at Merced, supervised by Prof. Ming-Hsuan Yang. He received the B. Eng. degree in the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. His current research interests focus on low level computer vision problems and deep learning.

Yunlong Dong Yunlong Dong is currently a Ph.D. candidate in the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, supervised by Prof. Ye Yuan. He received the B. Eng. degree in the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. His current research interests focus on low level computer vision problems and deep learning.

Wenqi Ren is an assistant professor in Institute of Information Engineering, Chinese Academy of Sciences, China. He received his Ph.D degree from Tianjin University in 2017. During 2015 to 2016, he was a joint-training Ph.D. student in Electrical Engineering and Computer Science at the University of California, Merced, CA, USA. His research interest includes image/video analysis and enhancement, and related vision problems.

Jinshan Pan is a professor of School of Computer Science and Engineering, Nanjing University of Science and Technology. He received the Ph.D. degree in computational mathematics from the Dalian University of Technology, China, in 2017. He was a joint-training Ph.D. student in School of Mathematical Sciences at and Electrical Engineering and Computer Science at University of California, Merced, CA, USA from 2014 to 2016. His research interest includes image deblurring, image/video analysis and enhancement, and related vision problems.

Changxin Gao is currently an associate professor with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. He received the Ph.D. degree in pattern recognition and intelligent systems from Huazhong University of Science and Technology in 2010. His research interests are pattern recognition and surveillance video analysis.

Nong Sang is currently a professor with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. He received his PhD degree in pattern recognition and intelligent control in 2000 from Huazhong University of Science and Technology. His research interests include pattern recognition, computer vision, and neural networks.

Ming-Hsuan Yang is a professor of Electrical Engineering and Computer Science with the University of California, Merced, CA, USA. He received the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign, USA, in 2000. He received the NSF CAREER Award in 2012, and the Google Faculty Award in 2009. He is a fellow of the IEEE and a senior member of the ACM.