NonLocal Channel Attention for NonHomogeneous Image Dehazing

Kareem Metwaly
kareem@psu.edu

Xuelu Li
xuelu@psu.edu

Tiantong Guo
tiantong@ieee.org

Vishal Monga
vmonga@engr.psu.edu

Abstract

The emergence of deep learning methods that complement traditional model-based methods has helped achieve a new state-of-the-art for image dehazing. Many recent methods design deep networks that either estimate the haze-free image (J) directly or estimate physical parameters in the haze model, i.e. ambient light (A) and transmission map (t) followed by using the inverse of the haze model to estimate the dehazed image. However, both kinds of methods fail in dealing with non-homogeneous haze images where some parts of the image are covered with denser haze and the other parts with shallower haze. In this work, we develop a novel neural network architecture that can take benefits of the aforementioned two kinds of dehazed images simultaneously by estimating a new quantity — a spatially varying weight map (w). w can then be used to combine the directly estimated J and the results obtained by the inverse model. In our work, we utilize a shared DenseNet-based encoder, and four distinct DenseNet-based decoders that estimate J, A, t, and w jointly. A channel attention structure is added to facilitate the generation of distinct feature maps of different decoders. Furthermore, we propose a novel dilation inception module in the architecture to utilize the non-local features to make up the missing information during the learning process. Experiments performed on challenging benchmark datasets of NTIRE'20 and NTIRE'18 demonstrate that the proposed method –namely, AtJwD– can outperform many state-of-the-art alternatives in the sense of quality metrics such as SSIM, especially in recovering images under non-homogeneous haze.

1. Introduction

Haze is a common atmospheric phenomenon caused by the floating particles in the atmosphere which can scatter or absorb lights. It dramatically degrades the visibility and details of scenes in captured outdoor images. Consequently, it also affects computer vision tasks that excessively depend on captured images, such as classification and segmentation [29]. Many methods have been proposed to reduce the negative impact from haze by utilizing a mathematical model introduced by [33], which can be described by the equation:

$$\mathbf{I}(x) = \mathbf{J}(x)\mathbf{t}(x) + \mathbf{A}(1 - \mathbf{t}(x)) \quad (1)$$

where $\mathbf{I}$ is the observed haze image, $\mathbf{J}$ is the true scene radiance, $\mathbf{A}$ is the global atmospheric light indicating the intensity of the ambient light, $\mathbf{t}$ is the transmission map and x is the pixel location. Transmission map is the distance-dependent factor that affects the fraction of light which is able to reach the camera sensor. When the atmospheric light $\mathbf{A}$ is homogeneous, the transmission map can be expressed as $\mathbf{t}(x) = e^{-\beta \mathbf{d}(x)}$, where β represents the attenuation coefficient of the atmosphere and $\mathbf{d}$ represents the scene depth. Most existing single image dehazing methods attempt to recover the clear image or scene radiance $\mathbf{J}$ based on the observed hazy image $\mathbf{I}$ via estimation of the transmission map $\mathbf{t}$. In fact, the image dehazing task is essentially a process of recovering $\mathbf{J}$ based on the observation of $\mathbf{I}$, which would inevitably lead to a heavily ill-posed problem according to Eq. (1). It can be observed from Eq. (1) that there are multiple possibilities for the choice of the solution when given a hazy image as the input. Having dense-haze in certain regions of the image implies a significantly small value (close to 0) for $\mathbf{t}$ and large value (close to 1) for $\mathbf{A}$ in those regions.

Existing dehazing work can be categorized into multi-image and single image dehazing. The limited availability of the parameters describing the scene information pushes early research to focus on multi-image dehazing [36, 37, 12, 31, 44, 45]. However, it is often unrealistic to capture many images of the same scene under different weather/environmental conditions, besides the problem of aligning multiple images with such limited scene information. As a result, single image dehazing has gained popularity recently where most work tries to reconstruct $\mathbf{J}$ through $\mathbf{I}$ and the estimated parameters $\mathbf{t}$ and $\mathbf{A}$ [39, 38, 23, 28].

Deep learning techniques are well-known for their excellent performance in image inverse problems such as single image super-resolution [47, 22], image deblurring [35], and image inpainting [50]. For single image dehazing, deep learning techniques also bring significant improvements in performance. These techniques usually require many pairs of hazy and haze-free images to either learn a mapping between them directly or to estimate $\mathbf{t}$ and/or $\mathbf{A}$ first then reconstruct the dehazed image utilizing Eq. (1).

For images with non-homogeneous haze, the haze level may vary from one region to another. One great example is the dataset used in NTIRE'20 dehazing challenge, which has non-homogeneous haze with sharp changes in terms of haze level from certain regions to others. Existing state-of-the-art methods fail to provide good performance when dealing with this dataset. In this paper, we propose a Non-Local Channel Attention Estimation Network to tackle the issues brought by non-homogeneity. The proposed network has a U-net [43] like structure with DenseNet blocks [26] embedded in it. The complete architecture consists of one shared encoder, three bottlenecks and four decoders. The encoder is used to extract representative features from the hazy input and the bottlenecks help bifurcate the feature extraction flow. Three decoders are used to obtain the estimated values $\hat{\mathbf{A}}$, $\hat{\mathbf{t}}$, $\hat{\mathbf{J}}_{\text{direct}}$ of $\mathbf{A}$, $\mathbf{t}$, and $\mathbf{J}$, respectively. The fourth decoder is designed to estimate w — a spatially varying weight map used to combine the dehazed result $\hat{\mathbf{J}}_{\text{AT}}$ reconstructed by $\hat{\mathbf{A}}$ and $\hat{\mathbf{t}}$ and the directly estimated haze free image $\hat{\mathbf{J}}_{\text{direct}}$. It is observed that $\hat{\mathbf{J}}_{\text{direct}}$ has more satisfying values in the dense haze regions where $\mathbf{t}$ is close to 0 and $\mathbf{A} >> \mathbf{J}$ – pixel-wise. Since, $\hat{\mathbf{t}}$ and $\hat{\mathbf{A}}$ will be close to extreme values while the direct estimation process of $\hat{\mathbf{J}}_{\text{direct}}$ can still recreate missing features consistently with the remaining available features. On contrary, in regions where there are light haze, $\hat{\mathbf{J}}_{\text{AT}}$ performs better than $\hat{\mathbf{J}}_{\text{direct}}$ since the accurate estimations of $\mathbf{A}$ and $\mathbf{t}$ can complement each other hence preserve more sharp information in reconstruction. The channel attention structure is added to facilitate the different decoders to extract different feature maps after receiving the features learned through the shared encoder.

Furthermore, in order to better preserve the information especially in the regions where the dense haze and light haze are concatenated, we propose a novel dilation inception module which successfully fill the gap regions in feature maps. Customized regularized loss terms are constructed to further enhance the parameter estimation. The experiments on the challenging NTIRE'18 and NTIRE'20 datasets show that the proposed method gives better results compared with other state-of-the-art alternatives. In NTIRE'20 non-homogeneous dehazing challenge[7], the proposed AtJwD and AtJwD+ (see Section 5 for details) obtain highly competitive dehazing results with AtJwD in particular achieving the best performance in terms of LPIPS metric [16] and the second best in terms of SSIM metric [24].

2. Related Work

Most deep learning based singe image dehazing methods try to reconstruct the haze-free image $\hat{\mathbf{J}}$ by using the inverse function of Eq. (1) with estimated $\hat{\mathbf{A}}$ and $\hat{\mathbf{t}}$ through $\mathbf{I}$. For instance, Ren *et al*. [42] proposed a multi-scale deep neural network to estimate $\hat{\mathbf{t}}$ and Cai *et al*. [11] introduces an end-to-end CNN network to estimate $\hat{\mathbf{t}}$ with a novel BReLU unit. More recently, Guo *et al*. [21] have developed a network to jointly estimate $\hat{\mathbf{t}}$ and $\hat{\mathbf{A}}$. Li *et al*. [29] proposed an all-in-one dehazing network to estimate $\hat{\mathbf{t}}$ and $\hat{\mathbf{A}}$. Some authors [18, 20] addressed the issue of color distortion in the earlier CNN-based work by presenting a multi-stage CNN. Liu *et al*. [32] developed a CNN based iterative algorithm to iteratively find $\hat{\mathbf{A}}$ and $\hat{\mathbf{t}}$. Similarly, Li *et al*. [30] have proposed a sophisticated model to gradually estimate the parameters of the physical model starting from the easier regions and going through the more difficult ones. Deng *et al*. [15] used a multi-model fusion for dehazing. Chen *et al*. [13] developed an adaptive-distillation based network to selectively change regions with higher haze level. Moreover, a high resolution auto-encoder network for dehazing has been proposed by Bianco *et al*. [9].

In addition to this, many people choose to use GAN [19] based architectures to improve their results [53, 41, 17]. For instance, Zhang *et al*. [53] proposed an end-to-end dehazing method to combine the parameter estimation and dehazing all together by utilizing a joint discriminator in GAN. Qu *et al*. [41] have proposed an enhanced Pix2Pix network based on GAN for dehazing, which can reinforce the dehazing effect in both color and details. Furthermore, Dudhane *et al*. [17] uses a residual inception module in their GAN architecture to learn integrated features related to haze-removal.

3. Proposed Method

To overcome the challenge in non-homogeneous image dehazing, we propose a specially designed deep network that can make benefits of regions with different haze levels. The proposed network treats distinct levels of haze differently. It utilizes a U-Net like structure with pre-trained dense blocks embedded in it. It has a shared encoder and multiple decoders to estimate different parameter values.

3.1. Weighted ensemble estimation

As described in Section 1, $\hat{\mathbf{J}}_{\text{direct}}$ has better performance than $\hat{\mathbf{J}}_{\text{AT}}$ in regions with dense haze and vice versa for regions with shallow haze. As a result, our proposed network is targeted to utilize the ensemble of both estimates in different regions. Hence, the proposed network is mainly composed by the following building blocks: **1)** One shared encoder, which is constructed based on densely connected network [26], **2)** Three bottleneck blocks used to bifurcate specific feature flows for decoders, **3)** Four separate decoders which have similar structures as the encoder. Skip connections are used between the encoder and the decoders as in U-net. The complete network structure is shown in Fig. 1.

Encoder: The detailed structure of the shared encoder is shown in Table 1. It consists of three pre-trained dense blocks borrowed from DenseNet-121 [26] with transition blocks in between. We obtained the pretrained network parameters of these blocks from PyTorch framework [40].

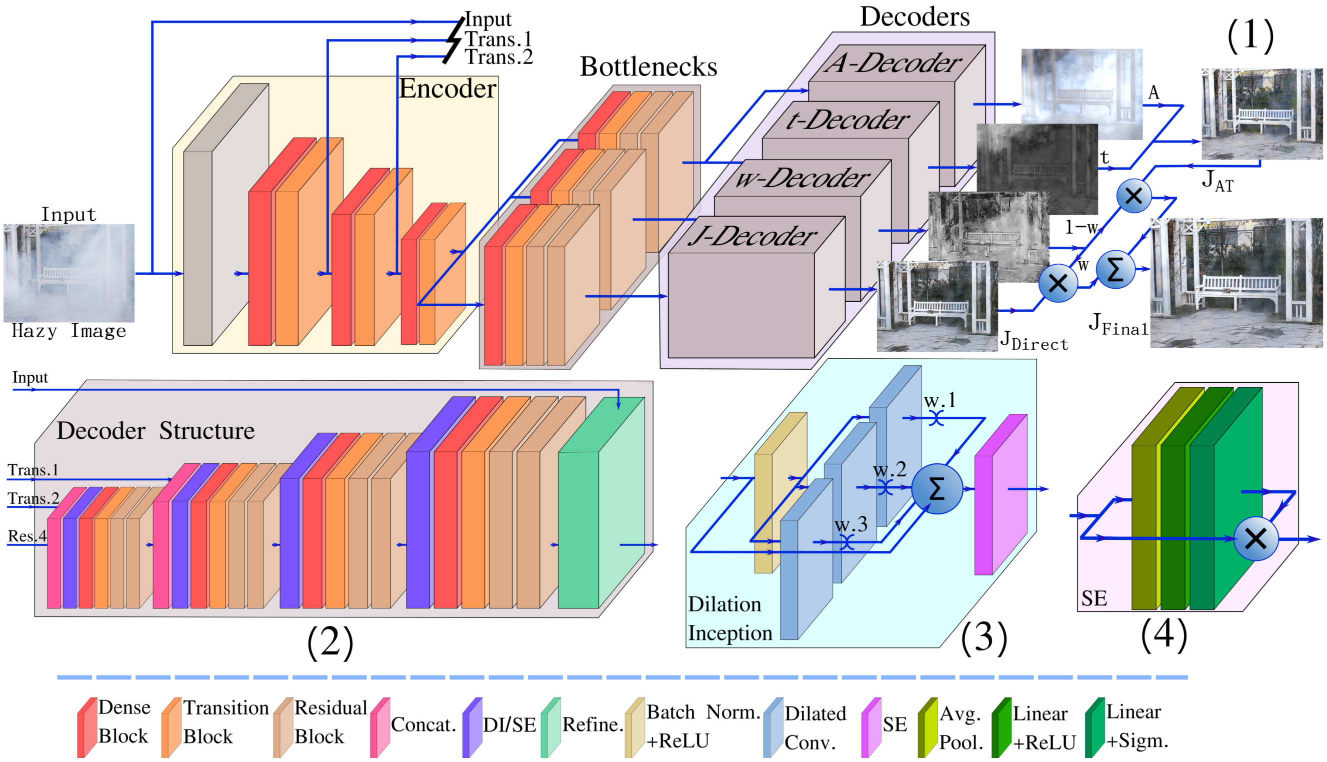

Figure 1: The proposed 'AtJwD' network architecture. All decoders are identical except for J-Decoder which has extra layers for non-local features utilization (Dilation Inception–DI). A-, t- and w-decoders only use SE layers.

DenseNet-121 was originally proposed for classification problem. We utilize the first three dense blocks from it in order to extract representative features even with limited training data. It is pretrained over ImageNet dataset [14] which is a very big dataset designed for classification.

Bottleneck: The bottleneck structure is used to connect the encoder and decoders. Its detailed structure is shown in Table 3. Different bottleneck structures connect to different decoders according to the characteristics of decoders. We use a shared bottleneck between A- and t-decoders as they contribute to the same estimation $\hat{\mathbf{J}}_{\text{AT}}$, which reduces the number of parameters in the network.

Decoders: the network architecture includes four decoders: A-, t-, J- and w-decoders to predict the estimated values $\hat{\mathbf{A}}$, $\hat{\mathbf{t}}$, $\hat{\mathbf{J}}_{\text{direct}}$ of $\mathbf{A}$, $\mathbf{t}$, and $\mathbf{J}$, respectively and a spatially varying weight map w used to give different weights when combining the ensemble outputs of $\hat{\mathbf{J}}_{\text{AT}}$ and $\hat{\mathbf{J}}_{\text{direct}}$. The decoders share similar structures as the encoder but have different intermediate structures from each other. In A-, t- and w-decoders, Squeeze and Excitation (SE) layers [25] are added at the middle of the structure. SE, detailed in Table 4, is a channel attention module which enable the three decoders to learn specific feature maps corresponding to their own characteristics while at the same time enjoying the benefits of complement learning brought by sharing the same encoder. For the J-decoder, we add a specially designed structure — dilation inception module which we will describe in detail in next section. Table 2 shows the details of the decoders.

To make the benefits of both $\hat{\mathbf{J}}_{\text{AT}}$ and $\hat{\mathbf{J}}_{\text{direct}}$, we first obtain the values of $\hat{\mathbf{J}}_{\text{AT}}$ through the physical model using estimated $\hat{\mathbf{A}}$ and $\hat{\mathbf{t}}$ as below:

$$\hat{\mathbf{J}}_{\text{AT}}(x) = \frac{\mathbf{I}(x) - \hat{\mathbf{A}}(x)(1-\hat{\mathbf{t}}(x))}{\hat{\mathbf{t}}(x)+\epsilon} \tag{2}$$

where $\mathbf{I}$ is the input hazy image. ϵ is a small value for numerical stability to avoid division by zero and x is the pixel location. Then we combine the output of J-decoder: $\hat{\mathbf{J}}_{\text{direct}}$ and $\hat{\mathbf{J}}_{\text{AT}}$ using the estimate w from w-decoder as below:

$$\hat{\mathbf{J}}_{\text{total}}(x) = w(x)\cdot\hat{\mathbf{J}}_{\text{direct}}(x) + (1-w(x))\cdot\hat{\mathbf{J}}_{\text{AT}}(x) \tag{3}$$

We constrain the value of w between 0 and 1 by using a Sigmoid activation layer at the end of w-decoder to prevent blobs of saturation and burns caused by large values after combination, in addition to training stabilization.

3.2. Dilation Inception Module

As mentioned in Section 3.1, we add a specially designed Dilation Inception Module in the middle of $J-$decoder. The main function of the proposed module is to take advantage of the non-local information nearby to complement

Table 1: Encoder Structure

	Base	Dense.1	Trans.1	Dense.2	Trans.2	Dense.3	Trans.3
Input	input patch/image	Base	Dense.1	Trans.1	Dense.2	Trans.2	Dense.3
Structure	$\begin{bmatrix}7\times 7\text{ conv.}\\ 3\times 3\text{ max-pool}\end{bmatrix}$	$\begin{bmatrix}1\times 1\text{ conv.}\\ 3\times 3\text{ conv.}\end{bmatrix}\times 6$	$\begin{bmatrix}1\times 1\text{ conv.}\\ 2\times 2\text{ avg-pool}\end{bmatrix}$	$\begin{bmatrix}1\times 1\text{ conv.}\\ 3\times 3\text{ conv.}\end{bmatrix}\times 12$	$\begin{bmatrix}1\times 1\text{ conv.}\\ 2\times 2\text{ avg-pool}\end{bmatrix}$	$\begin{bmatrix}1\times 1\text{ conv.}\\ 3\times 3\text{ conv.}\end{bmatrix}\times 24$	$\begin{bmatrix}1\times 1\text{ conv.}\\ 2\times 2\text{ avg-pool}\end{bmatrix}$
Output	$64\times 64\times 64$	$64\times 64\times 256$	$32\times 32\times 128$	$32\times 32\times 512$	$16\times 16\times 256$	$16\times 16\times 1024$	$8\times 8\times 512$

Table 2: Decoder Structure, C is the number of output channels which depends on the functionality of the decoder

	Dense.5	Trans.5	Res.5	Dense.6	Trans.6	Res.6
Input	[Res.4, Trans.2]	Dense.5	Trans.5	[Trans.1, Res.5]	Dense.6	Trans.6
Structure	$\begin{bmatrix}\text{SE/Dilation }(R=16)\\ \text{batch norm}\\ 3\times 3\text{ conv.}\end{bmatrix}\times 7$	$\begin{bmatrix}1\times 1\text{ conv.}\\ \text{upsample }2\end{bmatrix}$	$\begin{bmatrix}3\times 3\text{ conv.}\\ 3\times 3\text{ conv.}\end{bmatrix}\times 2$	$\begin{bmatrix}\text{SE/Dilation }(R=16)\\ \text{batch norm}\\ 3\times 3\text{ conv.}\end{bmatrix}\times 7$	$\begin{bmatrix}1\times 1\text{ conv.}\\ \text{upsample }2\end{bmatrix}$	$\begin{bmatrix}3\times 3\text{ conv.}\\ 3\times 3\text{ conv.}\end{bmatrix}\times 2$
Output	$16\times 16\times 640$	$32\times 32\times 128$	$32\times 32\times 128$	$32\times 32\times 384$	$64\times 64\times 64$	$64\times 64\times 64$

	Dense.7	Trans.7	Res.7	Dense.8	Trans.8	Res.8
Input	Res.6	Dense.7	Trans.7	Res.7	Dense.8	Trans.8
Structure	$\begin{bmatrix}\text{batch norm}\\ 3\times 3\text{ conv.}\end{bmatrix}\times 7$	$\begin{bmatrix}1\times 1\text{ conv.}\\ \text{upsample }2\end{bmatrix}$	$\begin{bmatrix}3\times 3\text{ conv.}\\ 3\times 3\text{ conv.}\end{bmatrix}\times 2$	$\begin{bmatrix}\text{batch norm}\\ 3\times 3\text{ conv.}\end{bmatrix}\times 7$	$\begin{bmatrix}1\times 1\text{ conv.}\\ \text{upsample }2\end{bmatrix}$	$\begin{bmatrix}3\times 3\text{ conv.}\\ 3\times 3\text{ conv.}\end{bmatrix}\times 2$
Output	$64\times 64\times 128$	$128\times 128\times 32$	$128\times 128\times 32$	$128\times 128\times 64$	$256\times 256\times 16$	$256\times 256\times 16$

	Refine.9	Refine.10	Refine.11	Refine.12	Refine.13	Output.14
Input	[Input, Res.8]	Refine.9	Refine.9	Refine.9	Refine.9	[Refine.9.10.11.12.13]
Structure	$\begin{bmatrix}\text{SE/Dilation }(R=3)\\ 3\times 3\text{ conv.}\end{bmatrix}$	$\begin{bmatrix}32\times 32\text{ avg-pool}\\ 1\times 1\text{ conv.}\\ \text{upsample}\end{bmatrix}$	$\begin{bmatrix}16\times 16\text{ avg-pool}\\ 1\times 1\text{ conv.}\\ \text{upsample}\end{bmatrix}$	$\begin{bmatrix}8\times 8\text{ avg-pool}\\ 1\times 1\text{ conv.}\\ \text{upsample}\end{bmatrix}$	$\begin{bmatrix}4\times 4\text{ avg-pool}\\ 1\times 1\text{ conv.}\\ \text{upsample}\end{bmatrix}$	3×3 conv.
Output	$256\times 256\times 20$	$256\times 256\times 1$	$256\times 256\times 1$	$256\times 256\times 1$	$256\times 256\times 1$	$256\times 256\times C$

Table 3: Bottleneck Structure

	Dense.4	Trans.4	Res.4
Input	Trans.3	Dense.4	Trans.4
Structure	$\begin{bmatrix}1\times 1\text{ conv.}\\ 3\times 3\text{ conv.}\end{bmatrix}\times 16$	$\begin{bmatrix}1\times 1\text{ conv.}\\ \text{upsample }2\end{bmatrix}$	$\begin{bmatrix}3\times 3\text{ conv.}\\ 3\times 3\text{ conv.}\end{bmatrix}\times 2$
Output	$8\times 8\times 768$	$16\times 16\times 128$	$16\times 16\times 128$

the missing information in local regions during the learning process. Its explicit structure is shown in Fig. 1 (3). It can be observed that the module is mainly constructed by several layers with dilation convolution[51], which is designed to increase receptive view (global view) of the network exponentially with linear parameter accretion. In our dilation inception module, first, the input features will pass a structure consisting of a Batch Normalization layer and a ReLU layer to normalize the input features and increase the non-linearity of the module. Then, the output is fed to N dilated convolution layers in parallel, each with a kernel size $= 3\times 3$, a stride of 1 and different dilation values from 1 to N. In our work, we set $N = 3$, since although a larger value for N can increase the perceptual quality of the whole image, it may unfortunately reduce the fidelity of the local regions. This is caused by the fact that utilization of non-local information will affect the utilization of local information at the same time. Next, each output from the dilated convolution layers is multiplied by a trainable weight and added to the input features. This guarantees that the dilated layers will not have a negative impact to the input during the learning process. As it only adds up features to the input according to the corresponding multiplied weight. Finally, we pass the new generated features through an SE layer to re-calibrate feature channels according to their respective importance. The details of the Dilation Inception Module can be seen in Table 5.

3.3. Customized Loss Function

In addition to the network structure, we designed a customized loss function $\mathcal{L}$ for the training process to obtain satisfying results from each decoder:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_p \mathcal{L}_p + \lambda_s \mathcal{L}_s + \lambda_A \mathcal{L}_{\text{std}} \tag{4}$$

$$\begin{aligned}\mathcal{L}_{rec} =& \|\hat{\mathbf{J}}_{\text{total}} - \mathbf{J}\|_2^2 \\ &+ \lambda_c \left(\|\hat{\mathbf{J}}_{\text{direct}} - \mathbf{J}\|_2^2 + \|\hat{\mathbf{J}}_{\text{AT}} - \mathbf{J}\|_2^2 \right)\end{aligned} \tag{5}$$

$$\mathcal{L}_p = \|G(\hat{\mathbf{J}}_{\text{total}}) - G(\mathbf{J})\|_2^2 \tag{6}$$

$$\mathcal{L}_s = 1 - \mathbf{SSIM}(\hat{\mathbf{J}}_{\text{total}}, \mathbf{J}) \tag{7}$$

$$\mathcal{L}_{\text{std}} = \sigma_A^2 \tag{8}$$

where $\mathcal{L}_{rec}$ is the reconstruction loss between the different reconstructed dehazed images $\hat{\mathbf{J}}_{\text{total}}$, $\hat{\mathbf{J}}_{\text{AT}}$, $\hat{\mathbf{J}}_{\text{direct}}$ and the ground truth $\mathbf{J}$, which ensures that each decoder is able to generate its expected estimated parameters. $\mathcal{L}_p$ is the perceptual loss obtained by pushing the outputs of feature extraction layers of a pre-trained VGG16 [46, 53] (G is the function representing the feature extraction module in the VGG model) to be as similar as possible when using $\hat{\mathbf{J}}_{\text{total}}$ and $\mathbf{J}$ as the inputs. $\mathcal{L}_s$ is used to maximize the value of **SSIM**, which is refereed as Multi-Scale Structure Similarity (MS-SSIM)[49]. By maximizing the value of **SSIM**, more detailed structural information of the input image can be preserved during the learning process of different parameters. We also regularize A by minimizing its variance σ_A^2 through $\mathcal{L}_{std}$ to prevent generating extreme values through out the image. λ_c, λ_p, λ_s and λ_A are hyper parameters used to balance the contribution of each loss term.

Table 4: SE Layer with a reduction factor R

	Pool	Lin.0	Lin.1	Multiplication
Input	X of shape $h \times w \times C$	Pool	Lin.0	[Lin.1, X]
Structure	$\begin{bmatrix}\text{Adaptive-Avg-Pool2D} \\ \text{Squeeze}\end{bmatrix}$	$\begin{bmatrix}\text{Linear}\,(C, C/R) \\ \text{ReLU}\end{bmatrix}$	$\begin{bmatrix}\text{Linear}\,(C/R, C) \\ \text{Sigmoid}\end{bmatrix}$	$\begin{bmatrix}\text{Broadcast of Lin.1 to X} \\ \text{Element-wise Multiplication}\end{bmatrix}$
Output	$1 \times C$	$1 \times C/R$	$1 \times C$	$h \times w \times C$

Table 5: Dilation Inception with N Layers and R Reduction, $w.k$ are the trainable parameters for dilated convolution layers

	Pre	D.1	D.2	$\cdots$	D.N	Post
Input	X of shape $h \times w \times C$	Pre	Pre	$\cdots$	Pre	$\text{X} + \sum_{k=1}^{N} w.k \times \text{D}.k$
Structure	$\begin{bmatrix}\text{batch norm} \\ \text{ReLU}\end{bmatrix}$	3×3 conv. dilation $= 1$	3×3 conv. dilation $= 2$	$\cdots$	3×3 conv. dilation $= N$	SE(R)
Output	$h \times w \times C$	$h \times w \times C$	$h \times w \times C$	$\cdots$	$h \times w \times C$	$h \times w \times C$

The details about how the layers in the network are connected can either be inferred from the tables provided in this paper or from the code made available online[1].

4. Dataset, Training, and Test Procedure

4.1. Dataset

In training for non-homogeneous haze, we used the training set provided in NTIRE'20 competition[3]. The images were collected by a professional camera including professional fog generators, so as to capture the same scene under both conditions (with and without haze). The haze distribution of the training images are non-homogeneous, namely, part of the regions in the images are covered with dense haze and others are with light haze. Haze level changes abruptly in some regions, which inhibits the accurate estimation of dehazed images. The dataset consists of 45 pairs of (non-homogeneous) hazy images and haze-free ground-truths. It also includes 5 images for validation and another 5 for testing without any ground truth data. The statistical experimental results shown in experiments section are obtained by submitting the generated images to the rank board provided by NTIRE'20 dehazing challenge organizers[2].

To learn a network with more powerful generalization ability, we randomly selected 10 images from the NTIRE2018- and NTIRE2019-Dehaze datasets [1, 6], and add them in training set with NTIRE'20 dehazing training set. The number of external training images are chosen based on the consideration of importing more similar structural information meanwhile avoiding learning excessive specific information from the external training images. During the training, patches of size 256×256 are extracted from the training images. The augmentations are used as the combination of the following options: **1)** horizontal flip, rotation by 90° , 180° , and 270° ; **2)** the images (whole image, not patches) are resized to 256×256 and applied the same augmentation strategies on these resized images and included them for training.

4.2. Training

In order to better utilize the pre-trained DenseNet modules embedded in our network and avoid unstable results in training a highly parameterized network. We adopted a two-stage training strategy as described below:

Stage 1 - Freezing the Encoder: In the first stage we freeze the parameters of the encoder and only allow the parameters of the three bottlenecks and the four decoders to be updated with big learning rate. This can provide a reasonable initialization for the parameters of the bottlenecks and the decoders, since the fixed pre-trained parameters in the encoder can ensure the bottlenecks and decoders to learn some reasonable initialization values by passing through representative features generated by the fixed-parameters encoder. Furthermore, randomly initialized parameters in bottlenecks and decoders can cause the over-learning of parameters in the encoder if not freezed, hence losing the benefits brought by the pre-trained module.

Stage 2 - Unfreezing the Encoder: Starting from epoch 40, the parameters of the encoder are unfreezed, and the whole network is trained together with a small learning rate. Since at this moment, the parameters of the whole network are in some good neighborhood of their optimum values. Having a larger learning rate may diverge the values of the parameters from that optimum point.

Adam optimizer [27] with initial learning rate of 1×10^{-4} and 1×10^{-6} is used for training in stages 1 and 2 respectively. The learning rate is reduced to its 70% after every 10 epochs. We set the values of $\lambda_c, \lambda_p, \lambda_s$ and λ_A to $0.7, 0.5, 0.5$ and 0.01 respectively by cross validation [34].

4.3. Optional Post-processing

An optional post-processing procedure that we utilized is the IRCNN [54] de-nosier with $\sigma = 15$ to further improve the visual results. IRCNN method enjoys both the benefits of model based techniques and learning based techniques for image restoration applications. In our dehazing prob-

[1] https://drive.google.com/drive/folders/1qsLmB_9HyqE7EhxRkhssNUeq9RVpuYXr?usp=sharing

[2] https://competitions.codalab.org/competitions/22236#results

(a) Hazy-Input (b) 'AtJw' (c) 'AtJwD'

Figure 2: Output of our network (b) without and (c) with dilation for validation image examples from NTIRE'20

lem, we apply the pre-trained CNN de-noiser and incorporate it as a post processing unit for the preliminary output obtained by our proposed framework. This is reported in the main manuscript. Albeit, it is noteworthy that over the test set, IRCNN did not have any significant effect. However, during validation, we have noticed that it gave a consistent marginal improvement by about 0.1 dB in PSNR. More discussion is provided in the experimental results section.

5. Experimental Results

In this section we present the experimental results of our proposed AtJwD network. Both an ablation study over different components of our network, outputs and loss terms and comparisons w.r.t state-of-the-art methods are presented. The evaluation metrics used to quantify the performance are Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [48].

5.1. Ablation Study

Effects of Dilation Inception and Color Channel Attention module: We denote the model by reducing dilation inception modules to SE layers by 'AtJw'. In other words, 'AtJw' has the same network structure as 'AtJwD' with four identical decoders that only have SE layers. We also denote the same network structure without SE layer as 'AtJw-'. In Table 6, we show a performance comparison between the three models. We also show in Fig. 2 two zoomed-in patches from NTIRE'20 validation set and the results of 'AtJw' and 'AtJwD'. It is clear that 'AtJwD' performs significantly better both in statistics and visualisation due to its rich utilization of non-local features and adaption of channel attention modules. Through experimentation, we have also noticed that increasing N (the number of dilated convolution layers) increases the perceptual quality but it may negatively impact fidelity depending on the haze level.

Table 6: Ablation study for SE and Dilation Inception over the validation set provided in NTIRE'20.

	AtJw-	AtJw	AtJwD
PSNR (dB) ↑	19.12	19.23	**19.38**
SSIM ↑	0.63	**0.65**	**0.65**

Effects of fusing $\hat{\mathbf{J}}_{\text{direct}}$ and $\hat{\mathbf{J}}_{\text{AT}}$: Table 7 shows how merging the two resultant images through the physical model $\hat{\mathbf{J}}_{\text{AT}}$ and the direct decoder $\hat{\mathbf{J}}_{\text{direct}}$ boosts the numerical results significantly. This is predictable since the physical model estimation performs well in shallow hazy regions, and the direct estimation performs better in the dense hazy regions. The generated value of w (the weight map), that is used to merge the two generated images, confirms this conclusion.

Table 7: 'AtJwD' vs. 'J'-only or 'At'-only

		PSNR	SSIM	Time/Epoch
AtJwD	$\hat{\mathbf{J}}_{\text{direct}}$	18.34	0.58	30 minutes
	$\hat{\mathbf{J}}_{\text{AT}}$	18.83	0.59	
	$\hat{\mathbf{J}}_{\text{total}}$	**19.38**	**0.65**	
J-only		18.29	0.58	15 minutes
At-only		18.51	0.59	20 minutes

Comparison between 'AtJwD', 'J'-only and 'At'-only Table 7 also provides a comparison between outputs from AtJwD and two other separate estimation networks. The first one has one encoder and one decoder to generate $\hat{\mathbf{J}}_{\text{direct}}$ directly and the other one has one encoder and two decoders to generate $\mathbf{J}$ through estimating $\mathbf{A}$ and $\mathbf{t}$. It is clear that having a shared encoder gives better results since the estimation of different parameters can provide complementary information to each other during the training process. In addition to that, according to the results shown in Table 7, we can infer that estimating $\mathbf{J}$ and $\hat{\mathbf{J}}_{\text{AT}}$ separately will consume much more training time not to mention that also need another network to estimate w.

Effects of different loss terms: We also show the effect of removing different loss terms from the total loss in Eq. (4) in Table 8. It is clear that the total customized loss increases the performance. The perceptual loss and SSIM losses have a noticeable effect on the SSIM results while the STD loss has some effect over the PSNR results.

Table 8: Ablation study over different loss terms

$\mathcal{L}_{rec}$	✓	✓	✓	✓
$\mathcal{L}_p$		✓	✓	✓
$\mathcal{L}_s$			✓	✓
$\mathcal{L}_{\text{STD}}$				✓
PSNR	19.03	19.20	19.31	**19.38**
SSIM	0.60	0.61	**0.65**	**0.65**

5.2. Comparison with State-of-the-art Methods

This section illustrates the comparisons between our proposed methods with the state-of-the-art methods on real-world benchmark data sets I-HAZE, and O-HAZE [4, 5]. Although AtJwD has similar performance over NTIRE'19 [2], we do not show it due to limitations in paper size.

State-of-the-art Methods The state-of-the-art methods included in the comparisons are: TIP'15 [56], ECCV'16 [42], TIP'16 [11], CVPR'16 [8], ICCV'17 [29], CVPR'18 [52], CVPRW'18 [53] and CVPRW'19 [21].

Evaluation Datasets The comparisons are conducted on the I-HAZE (indoor) and O-HAZE (outdoor) validation datasets [4, 5]. Each of the dataset contains 5 pairs of

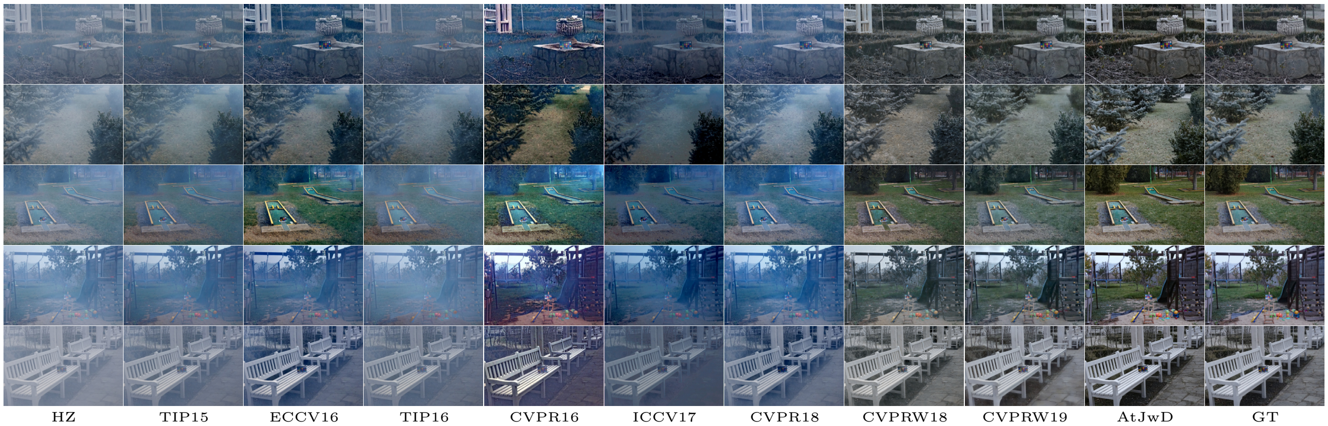

Figure 3: The visual results of NTIRE2018-outdoor validation dataset.

Figure 4: The visual results of NTIRE2018-indoor validation dataset.

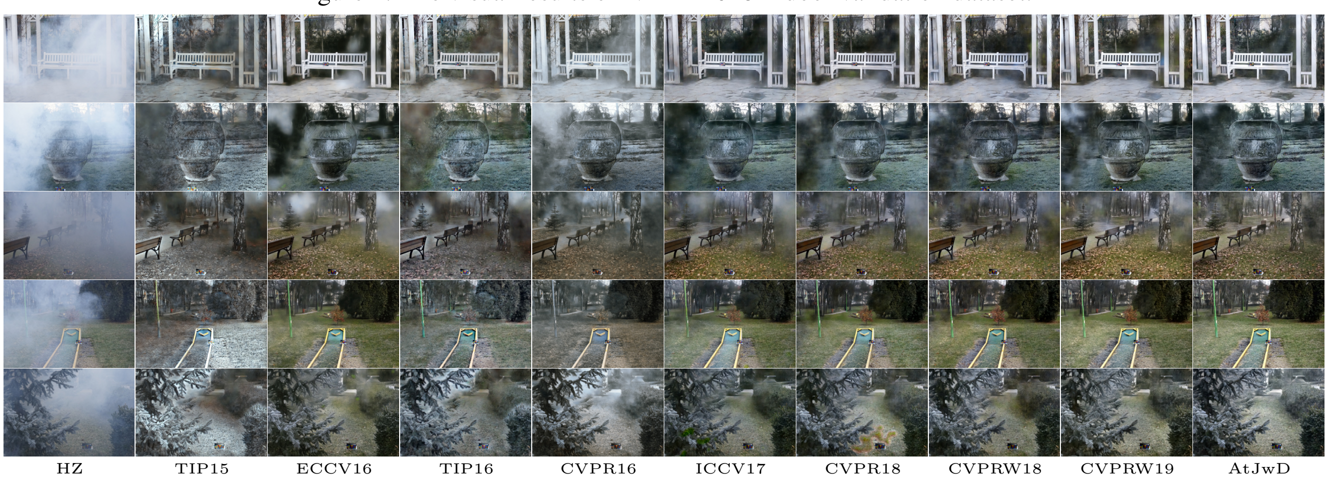

Figure 5: The visual results of NTIRE2020 validation dataset.

haze and haze-free image pairs. Detailed acquisition methods of these real-world hazy image pairs are discussed in [4, 5]. Figs. 3 and 4 show the experimental results of the state-of-the-art methods compared with AtJwD conducted on NTIRE2018 indoor and outdoor validation datasets. It can be found that AtJwD generates much more visually pleasing results. As shown in Tables 9 and 10, AtJwD outperforms other state-of-the-art methods when evaluated on PSNR and SSIM. AtJwD+ refers to the output after applying the post-processing processing step discussed in Section 4.3. The post processing had a small but stable improvement in the PSNR values but had no effect over the SSIM values.

Table 9: The PSNR/SSIM of different methods over NTIRE2018-outdoor validation dataset.

method	36.*png*	37.*png*	38.*png*	39.*png*	40.*png*	avg.
TIP15 [56]	17.4660/0.4976	16.1686/0.4533	15.1391/0.1796	14.7964/0.4131	16.3732/0.5683	15.9887/0.4224
TIP16 [42]	16.5891/0.4862	15.7593/0.4334	13.2500/0.1890	12.7816/0.3935	16.5339/0.5597	14.9828/0.4123
CVPR16 [11]	16.9236/0.4267	14.9854/0.4776	15.5448/0.3390	17.6496/0.4751	17.0424/0.5350	16.4292/0.4507
ICCV17 [29]	17.0951/0.4516	16.4676/0.3886	16.1153/0.1194	15.0439/0.3388	15.9477/0.5043	16.1339/0.3606
CVPR18 [52]	17.1374/0.4385	15.2847/0.4173	14.6555/0.1143	15.2353/0.3530	17.7805/0.5198	16.0187/0.3686
CVPRW18 [53]	24.6703/0.7288	22.4079/0.6551	23.7469/0.7199	21.9055/0.6296	22.2878/0.6822	23.0037/0.6831
CVPRW19 [21]	27.0772/0.8154	24.0295/0.7513	23.9662/0.7991	22.5974/0.7555	24.4090/0.8025	24.4159/0.7847
AtJwD	27.2630/0.8309	24.3895/0.7655	24.0893/0.8146	22.8518/0.7338	24.6910/0.8507	**24.65692/0.7991**
AtJwD+	27.3800/0.8309	24.3991/0.7709	24.1082/0.8209	22.8537/0.7420	24.6981/0.8507	**24.65692/0.7991**

Table 10: The PSNR/SSIM of different methods over NTIRE2018-indoor validation dataset.

method	26.*png*	27.*png*	28.*png*	29.*png*	30.*png*	avg.
TIP15 [56]	13.1816/0.6581	16.6858/0.3952	11.5135/0.5590	17.1496/0.7803	15.7567/0.3215	14.8574/0.5428
TIP16 [42]	10.1699/0.5498	14.5147/0.3094	13.3890/0.6349	11.9041/0.5369	15.5312/0.3412	13.1018/0.4744
CVPR16 [11]	12.4147/0.4800	14.7990/0.3639	13.2925/0.5489	14.6639/0.5296	13.9293/0.4057	13.8199/0.4656
ICCV17 [29]	10.8313/0.6185	16.8387/0.3943	12.7391/0.4692	15.3688/0.8054	17.2741/0.3095	14.6104/0.5194
CVPR18 [52]	15.3106/0.6283	16.0856/0.3512	9.8470/0.5540	22.2085/0.8013	15.4517/0.1977	15.7807/0.5065
CVPRW18 [53]	14.2680/0.6778	20.8952/0.7533	18.4479/0.6983	20.5845/0.8154	16.4299/0.5445	18.1251/0.6978
CVPRW19 [21]	20.5938/0.8760	22.9991/0.8490	19.9912/0.8313	22.9211/0.9001	18.5186/0.8056	21.0048/0.8524
AtJwD	22.8168/0.9016	23.1603/0.8590	20.2137/0.8511	23.6192/0.9208	20.5301/0.8458	**22.0680/0.8756**
AtJwD+	22.8503/0.9018	23.1960/0.8598	20.3737/0.8622	23.6214/0.9220	20.6101/0.8458	**22.1303/0.8783**

Table 11: The PSNR/SSIM of different methods over NTIRE2020 **validation** dataset.

method	avg. over 5 images
TIP15 [56]	14.59/0.55
TIP16 [42]	15.94/0.57
CVPR16 [11]	16.13/0.60
ICCV17 [29]	17.97/0.62
CVPR18 [52]	17.90/0.63
CVPRW18 [53]	18.23/0.62
CVPRW19 [21]	18.45/0.64
AtJwD	**19.38/0.65**
AtJwD+	**19.39/0.65**

Table 12: Comparison between our network with the dilation inception module (AtJwD) and without (AtJw) against some other methods participating in the competition over the **test** set provided in NTIRE'20.

	method 1	method 2	AtJw	AtJwD
PSNR (dB) ↑	19.70	19.22	19.92	**20.10**
SSIM ↑	0.68	0.66	0.68	**0.69**
LPIPS[55] ↓	0.301	0.266	0.269	**0.265**
PI[10] ↓	2.985	3.267	2.888	**2.883**

5.3. NTIRE-2020 Dehazing Challenge

The haze presented in images from the NTIRE2020-Dehaze dataset is non-homogeneous compared to images in the previous literature. As shown in Fig. 5, the state-of-the-art methods' performances drop largely when applied to the dataset due to the reason that non-homogeneity makes it difficult to have a good estimate of the physical parameters or to directly estimate the dehazed image each on its own as most of the state-of-the-art methods do. Since AtJwD can estimate both accurately from the non-homogeneous haze image, the dehazed images generated by AtJwD are much more visually pleasing. We evaluate the quantitative performances of the methods on the NTIRE2020 validation set through the competition server [7]. As shown in Table 11, AtJwD outperforms all the other state-of-the-art methods. Table 12 includes a comparison between AtJwD, AtJw and some other top methods from the contest. It is found that AtJwD/AtJw are among the top performing methods. Specifically, they outperform all other methods in terms of perceptual quality metrics (AtJwD ranked first in terms of LPIPS metric and second in terms of SSIM). We noticed over the testset, the post processing step didn't have any effect on the PSNR nor SSIM values. In fact, it negatively impacted perceptual metrics (LPIPS and PI).

6. Conclusion

We focus on developing deep learning architecture that estimates physical parameters in the haze model. Our AtJwD network, uses a shared DenseNet encoder and four distinct decoders to jointly estimate the scene information viz. A and t., the haze-free scene directly and the fusing weight between them. We use a channel attention scheme to generate different feature maps and a novel Dilation Inception module at the direct decoder to generate missing features at densely-hazed regions using non-local features. Experiments performed on challenging benchmark image datasets of NTIRE'20 and NTIRE'18 demonstrate that AtJwD can outperform state-of-the-art alternatives. Notably, in NTIRE'20 results where the haze is non-homogeneous, AtJwD outperforms the competing methods.

References

[1] C. Ancuti, C. O. Ancuti, and R. Timofte. NTIRE 2018 Challenge on Image Dehazing: Methods and Results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 891–901, 2018. 5

[2] C. O. Ancuti, C. Ancuti, M. Sbert, and R. Timofte. Dense haze: A benchmark for image dehazing with dense-haze and haze-free images. In *IEEE International Conference on Image Processing*, 2019. 6

[3] C. O. Ancuti, C. Ancuti, and R. Timofte. NH-HAZE: An image dehazing benchmark with nonhomogeneous hazy and haze-free images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 5

[4] C. O. Ancuti, C. Ancuti, R. Timofte, and C. De Vleeschouwer. I-HAZE: a dehazing benchmark with real hazy and haze-free indoor images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018. arXiv: 1804.05091. 6, 7

[5] C. O. Ancuti, C. Ancuti, R. Timofte, and C. De Vleeschouwer. O-HAZE: A Dehazing Benchmark With Real Hazy and Haze-Free Outdoor Images. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pages 754–762, 2018. 6, 7

[6] C. O. Ancuti, C. Ancuti, R. Timofte, L. V. Gool, L. Zhang, and M.-H. Yang. NTIRE 2019 Image Dehazing Challenge Report. *The IEEE Conference on Computer Vision and Pattern Recognition Workshops*, page 13, June 2019. 5

[7] C. O. Ancuti, C. Ancuti, F.-A. Vasluianu, R. Timofte, et al. Ntire 2020 challenge on nonhomogeneous dehazing. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 2, 8

[8] D. Berman, S. Avidan, et al. Non-local image dehazing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1674–1682, 2016. 6

[9] S. Bianco, L. Celona, F. Piccoli, and R. Schettini. High-Resolution Single Image Dehazing Using Encoder-Decoder Architecture. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2

[10] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, and L. Zelnik-Manor. 2018 pirm challenge on perceptual image super-resolution. In *Proceedings of the IEEE European Conference on Computer Vision Workshop*, 2018. 8

[11] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao. DehazeNet: An End-to-End System for Single Image Haze Removal. *IEEE Transactions on Image Processing*, 25(11):5187–5198, Nov. 2016. arXiv: 1601.07661. 2, 6, 8

[12] L. Caraffa and J.-P. Tarel. Stereo Reconstruction and Contrast Restoration in Daytime Fog. In K. M. Lee, Y. Matsushita, J. M. Rehg, and Z. Hu, editors, *Computer Vision – ACCV 2012*, Lecture Notes in Computer Science, pages 13–25, Berlin, Heidelberg, 2013. Springer. 1

[13] S. Chen, Y. Chen, Y. Qu, J. Huang, and M. Hong. Multi-Scale Adaptive Dehazing Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 3

[15] Z. Deng, L. Zhu, X. Hu, C.-W. Fu, X. Xu, Q. Zhang, J. Qin, and P.-A. Heng. Deep Multi-Model Fusion for Single-Image Dehazing. In *2019 IEEE/CVF Inter. Conf. on Computer Vision*, pages 2453–2462, Oct. 2019. ISSN: 2380-7504. 2

[16] A. Dosovitskiy and T. Brox. Generating Images with Perceptual Similarity Metrics based on Deep Networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 658–666. Curran Associates, Inc., 2016. 2

[17] A. Dudhane, H. S. Aulakh, and S. Murala. RI-GAN: An End-To-End Network for Single Image Haze Removal. *The IEEE Conference on Computer Vision and Pattern Recognition Workshops*, page 10, June 2019. 2

[18] A. Dudhane and S. Murala. C^ 2msnet: A novel approach for single image haze removal. In *Winter Conf. on Applications of Computer Vision*, pages 1397–1404. IEEE, 2018. 2

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. . Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[20] T. Guo, V. Cherukuri, and V. Monga. Dense '123' Color Enhancement Dehazing Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, page 9, 2019. 2

[21] T. Guo, X. Li, V. Cherukuri, and V. Monga. Dense Scene Information Estimation Network for Dehazing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 2, 6, 8

[22] T. Guo, H. S. Mousavi, T. H. Vu, and V. Monga. Deep wavelet prediction for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017. 1

[23] N. Hautière, J.-P. Tarel, and D. Aubert. Towards fog-free in-vehicle vision systems through contrast restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 1

[24] A. Hore and D. Ziou. Image quality metrics: PSNR vs. SSIM. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369. IEEE, 2010. 2

[25] J. Hu, L. Shen, and G. Sun. Squeeze-and-Excitation Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, June 2018. ISSN: 1063-6919. 3

[26] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger. Densely Connected Convolutional Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2261–2269, Honolulu, HI, July 2017. IEEE. 2

[27] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations*, San Diego, CA, USA, 2015. 5

[28] J. Kopf, B. Neubert, B. Chen, M. Cohen, D. Cohen-Or, O. Deussen, M. Uyttendaele, and D. Lischinski. Deep photo: model-based photograph enhancement and viewing, Dec. 2008. 1

[29] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng. AOD-Net: All-in-One Dehazing Network. In *2017 IEEE International Conference on Computer Vision*, pages 4780–4788, Venice, Oct. 2017. IEEE. 1, 2, 6, 8

[30] Y. Li, Q. Miao, W. Ouyang, Z. Ma, H. Fang, C. Dong, and Y. Quan. LAP-Net: Level-Aware Progressive Network for Image Dehazing. In *2019 IEEE/CVF International Conference on Computer Vision*, pages 3275–3284, Oct. 2019. ISSN: 2380-7504. 2

[31] Z. Li, P. Tan, R. T. Tan, D. Zou, S. Zhiying Zhou, and L.-F. Cheong. Simultaneous video defogging and stereo reconstruction. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 4988–4997, 2015. 1

[32] Y. Liu, J. Pan, J. Ren, and Z. Su. Learning Deep Priors for Image Dehazing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2492–2500, 2019. 2

[33] E. J. McCartney. Optics of the atmosphere. Scattering by molecules and particles. *Wiley Series in Pure and Applied Optics, New York: Wiley, 1976*, 1976. 1

[34] V. Monga, editor. *Handbook of Convex Optimization Methods in Imaging Science*. Springer International Publishing, Cham, 2018. 5

[35] S. Nah, T. Hyun Kim, and K. Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3883–3891, 2017. 1

[36] S. G. Narasimhan and S. K. Nayar. Chromatic framework for vision in bad weather. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, volume 1, pages 598–605. IEEE, 2000. 1

[37] S. G. Narasimhan and S. K. Nayar. Contrast restoration of weather degraded images. *IEEE Transactions on Pattern Analysis and Machine Int.*, 25(6):713–724, 2003. 1

[38] S. G. Narasimhan and S. K. Nayar. Interactive (de) weathering of an image using physical models. In *Proceedings of the IEEE Workshop Color and Photometric Methods in Computer Vision*, volume 6, page 1. France, 2003. 1

[39] J. P. Oakley and B. L. Satherley. Improving image quality in poor visibility conditions using a physical model for contrast degradation. *IEEE Transactions on Image Processing*, 7(2):167–179, 1998. 1

[40] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 2

[41] Y. Qu, Y. Chen, J. Huang, and Y. Xie. Enhanced Pix2pix Dehazing Network. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8152–8160, Long Beach, CA, USA, June 2019. IEEE. 2

[42] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang. Single image dehazing via multi-scale convolutional neural networks. In *Proceedings of the IEEE European Conference on Computer Vision*, pages 154–169. Springer, 2016. 2, 6, 8

[43] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conf. on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2

[44] Y. Y. Schechner, S. G. Narasimhan, and S. K. Nayar. Instant dehazing of images using polarization. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, volume 1, pages 325–332, 2001. 1

[45] S. Shwartz, E. Namer, and Y. Y. Schechner. Blind haze separation. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, volume 2, pages 1984–1991, 2006. 1

[46] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations*, 2014. 4

[47] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, L. Zhang, B. Lim, S. Son, H. Kim, S. Nah, K. M. Lee, et al. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pages 1110–1121. IEEE, 2017. 1

[48] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003. 6

[49] Z.-S. Xiao. A Multi-scale Structure SIMilarity metric for image fusion qulity assessment. In *2011 International Conference on Wavelet Analysis and Pattern Recognition*, pages 69–72, July 2011. ISSN: 2158-5695. 4

[50] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *Proc. of Advances in Neural Information Processing Systems*, pages 341–349, 2012. 1

[51] F. Yu and V. Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. *International Conference on Learning Representations*, 2015. 4

[52] H. Zhang and V. M. Patel. Densely connected pyramid dehazing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3194–3203, 2018. 6, 8

[53] H. Zhang, V. Sindagi, and V. M. Patel. Multi-scale Single Image Dehazing Using Perceptual Pyramid Deep Network. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1015–101509, Salt Lake City, UT, USA, June 2018. IEEE. 2, 4, 6, 8

[54] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, pages 3929–3938, 2017. 5

[55] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, Salt Lake City, UT, June 2018. IEEE. 8

[56] Q. Zhu, J. Mai, and L. Shao. A fast single image haze removal algorithm using color attenuation prior. *IEEE Trans. on Image Processing*, 24(11):3522–3533, 2015. 6, 8