Dense Scene Information Estimation Network for Dehazing

Tiantong Guo, Xuelu Li, Venkateswararao Cherukuri, Vishal Monga
The Pennsylvania State University, The Department of Electrical Engineering, University Park, PA, USA
tiantong@ieee.org, xuelu@psu.edu, vmc5164@psu.edu, vmonga@engr.psu.edu

Abstract

Image dehazing continues to be one of the most challenging inverse problems. Deep learning methods have emerged to complement traditional model based methods and have helped define a new state of the art in achievable dehazed image quality. Yet, practical challenges remain in dehazing of real-world images where the scene is heavily covered with dense haze, even to the extent that no scene information can be observed visually. Many recent dehazing methods have addressed this challenge by designing deep networks that estimate physical parameters in the haze model, i.e. ambient light (A) and transmission map (t). The inverse of the haze model may then be used to estimate the dehazed image. In this work, we develop two novel network architectures to further this line of investigation. Our first model, denoted as At-DH, designs a shared DenseNet based encoder and two distinct DenseNet based decoders to jointly estimate the scene information viz. A and t respectively. This in contrast to most recent efforts (include those published in CVPR’18) that estimate these physical parameters separately. As a natural extension of At-DH, we develop the AtJ-DH network, which adds one more DenseNet based decoder to jointly recreate the haze-free image along with A and t. The knowledge of (ground truth) training dehazed/clean images can be exploited by a custom regularization term that further enhances the estimates of model parameters A and t in AtJ-DH. Experiments performed on challenging benchmark image datasets of NTIRE’19 and NTIRE’18 demonstrate that At-DH and AtJ-DH can outperform state-of-the-art alternatives, especially when recovering images corrupted by dense haze.

1. Introduction

Haze is an atmospheric phenomenon in which dust, smoke, and other particulates obscure the clarity. Haze can potentially incur dramatic degradation to the visibility of scenes captured, esp. in inclement weather. While hazing leads to a loss of quality for visual consumption, the presence of haze in digital images has further negative effects on high-level computer vision tasks such as detection and recognition[1] of objects.

Dehazing refers to the technique of using algorithmic methods to process hazed images and produce (dehazed) images with high visual quality. The need for effective dehazing algorithms has been fueled by emerging applications in photography from mobile devices, autonomous driving and navigation systems, etc.

Many methods have been proposed to counteract the negative influence of haze and their starting point is the classical haze model introduced by [2] as below:

\[
I = J \cdot t + A \cdot (1 - t)
\]

where \(\cdot\) represents element-wise multiplication, \(I\) is the observed hazy image, \(J\) is the true scene radiance, \(A\) is the ambient light intensity, and \(t\) is the transmission map. \(t\) is a distance-dependent factor that affects the fraction of light which is able to reach the camera sensor. \(t\) can be expressed as \(t = e^{-\beta d}\), where \(\beta\) represents the attenuation coefficient of the atmosphere and \(d\) represents the scene depth. The image dehazing task is essentially a process of recovering \(J\) based on \(I\), 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 solution choices for any given hazy image as the input. It is also clear that if the information about \(d\) is provided, the task of recovering \(J\) will become much easier. This depth information is however rarely available in practice.

Image dehazing has been actively researched over the past two decades [3, 4, 5, 6, 7, 8, 9, 10, 11]. Existing work can be categorized into multi-image and single image dehazing. Restricted by the availability of the parameters that describe the scene information, much early research focused on multi-image dehazing [12, 13]. The availability of many images of the same scene under different weather/environmental conditions is however often unrealistic.

As a result, in recent efforts, single image dehazing has gained popularity [14, 15]. Most existing single image dehazing methods attempt to recover \(J\) based on \(I\) via the estimation of \(t\), including applying dark channel priors and color regularizers, etc. Performance benefits of
these techniques over early methods has been demonstrated
[16, 17, 18, 19, 20, 21, 22].

Recently, Deep Learning (DL) methods have been shown to achieve success in imaging inverse problems such as image super-resolution [23, 24], deblurring [25], and inpainting [26]. For single image dehazing, DL methods usually require I and J as training pairs to learn mappings between I and J, I and t, and/or I and A. A detailed review of related literature is provided in Sec. 2. These methods take advantage of the powerful generalization ability of the deep networks to model the non-linear mapping function. However, as the real-world I and J pairs are hard to come by, synthetic datasets are often used during training.

In real-world scenes, the haze level can be much denser than is the case in most synthetic training datasets. The NTIRE2019-Dehaze Challenge dataset is representative of real-world challenges. Even if parts of the image are well recovered, no explicit visual information can be often be seen in areas with the densest haze.

In this paper, we propose a Dense Scene Information Estimation Network to directly tackle this issue. Our first model, denoted as At-DH, designs a shared DenseNet based encoder and two distinct DenseNet based decoders to jointly estimate the scene information viz. A and t respectively. State of the art success in dehazing has been shown by approaches that use a custom designed deep network for estimating model parameters – such as [27] – however, they design separate networks to estimate A and t.

To incorporate more structural information in the learning process and overcome the difficulty of estimating t and A in dense haze, we develop an extension of At-DH called the AtJ-DH network, which adds one more DenseNet based decoder to jointly recreate the haze-free image along with A and t. The knowledge of (ground truth) training dehazed/clean images can be exploited by a custom regularization term that further enhances the estimates of model parameters A and t in AtJ-DH. Experiments performed on challenging benchmark image datasets of NTIRE’19 and NTIRE’18 demonstrate that At-DH and AtJ-DH can outperform state-of-the-art alternatives, especially when it comes to dense haze. Based on PSNR and SSIM results, the AtJ-DH ranks 1st and At-DH ranks 3rd in the NTIRE’19 Dehazing Challenge [28].

2. Related Work

Multiple deep learning based methods have been proposed to estimate t and A from I to reconstruct J. For example, Cai [29], introduced an end-to-end CNN network to estimate t with a novel BReLU unit. More recently, Ren [30] also proposed a multi-scale deep neural network to estimate t. However, these methods all have the limitation that they only put t into consideration in their CNN frameworks without introducing any other essential information.

Li [1] proposed an all-in-one dehazing network to address the problem, where a linear transpose is leveraged to encode t and A into one variable.

The success of Generative Adversarial Networks (GANs) [31] in synthesizing realistic images has led researchers to explore more possibilities in the applications of GANs. In [32], to further incorporate the mutual structure information between the estimated t and the dehazed result, the authors proposed a joint-discriminator based on GAN to decide whether the corresponding dehazed image and the estimated transmission map are real or fake. In [27], the authors present a multi-scale image dehazing method using perceptual pyramid deep network based on the recently popular dense blocks and residual blocks. The method involves an encoder-decoder structure with a pyramid pooling module in the decoder to incorporate contextual information of the scene while decoding. In [33], the authors proposed a bi-directional consistency loss to estimate t and A to reconstruct J based on their specially designed fully convolutional neural network. In [34], a bilinear convolutional neural network is used to estimate t and A with the help of the proposed bilinear composition loss function which directly model the correlations between t, A and J.

3. Scene Information Estimation Network

Our proposed deep network framework consists of a shared encoder and multiple decoders. The encoder serves as a general feature extractor and the decoders are trained to estimate the scene information based on the features extracted from the encoder. Inspired by the success of dense networks in various applications including dehazing [32], we build our encoder and decoders with DenseNet blocks.

3.1. Network Building Blocks

The proposed network is targeted to estimate the scene information from corrupted images with dense haze. The estimation system along with the refinement blocks are constructed by using the following building blocks:

1) Encoder: the encoder is constructed based on the Densely Connected Network (DCN) [35].

2) Decoder: the decoder has a similar structure as the encoder but with more batch normalization layers.

3) Refinement blocks: the refinement blocks in [32] are used for refine the outputs at different scales.

The encoder blocks from ‘Base.0’ to ‘Dense.4’ in Table 1 are initialized with pre-trained parameters from [35], which is originally trained for image classification. These connected pre-trained blocks also have the ability to obtain representative features in dehazing tasks. The dense blocks in the latter half of the DCN from [35] are not utilized since the features obtained through it will be mapped into a lower-dimension feature space only for classification. To make the pre-trained dense blocks be better applied for dehazing, we
append the encoder with newly added blocks ‘Trans.4’ and ‘Res.4’ to enlarge the features generated by ‘Dense.4’. It can help the encoder preserve more spatial feature information from the pre-trained dense blocks which is then utilized by decoders for further enhancement in dehazing.

Table 2 provides more details about the structure of the decoder. The decoder is built to exploit the extracted features from the encoder for dehazing. It contains 4 ‘Trans’ blocks, which stand for transformation block. Each transformation block essentially plays the role of reordering and enlarging the refined image/features. The reordering process is accomplished by a 1 × 1 convolutional layer, and the enlargement is done by the upsampling layer 2. We added new residual blocks [36] in between two successive dense blocks to incorporate more high-frequency information which can assist in recovering more details in the recovered image. In addition to that, batch normalization layers are added to the dense blocks to normalize the training data so that the manifold of the network parameters will be smoothed and the network will enjoy better training stability[37].

The bottom row of Table 2 details the structure of refinement blocks as suggested by [27, 32]. These blocks first use average pooling layers at spatial size of 32 × 32, 16 × 16, 8 × 8, and 4 × 4 to extract local average information. Then, the 1 × 1 convolutional layer refines the outputs and an upsampling layer is used to enlarge the image into the desired spatial size. These enlarged locally reorganized images are then appended together and passed through the final refinement layer which uses several convolutional layers to eliminate the blocking artifacts. This refinement practice would allow the image information to be merged and retouched at different scales.

### 3.2. At-DH Network

With the aforementioned building blocks, the proposed At-DH network structure is described in Table 3 and shown in the dash-lined part in Fig. 1. The encoder in At-DH is constructed as described in Table 1 and the Decoder_A and Decoder_B are constructed as illustrated in Table 2. The features extracted by the encoder are densely connected to the decoder. As shown in Table 2, ‘Dense.5’ utilizes the concatenated outputs from ‘Res.4’ and ‘Trans.2’ to have better information flow as suggested in [35]. Similar connection is used in ‘Dense.6’ as well.

In the At-DH model, the two decoders are trained to generate the A and ë such that combined with the input hazy image I, the At-DH network can generate haze-free images.
Figure 1: The proposed the At-DH (within dashlines) and AtJ-DH network. A shared encoder is employed to extract image features that are subsequently exploited by the two decoders (Decoder.A and Decoder.t) to jointly estimate the physical model parameters. AtJ-DH employs an additional decoder, i.e. Decoder.J which estimates the haze-free image. Employing custom regularization in the learning of AtJ-DH using the ground truth further enhances the estimates A and t. The architectural details of the encoder and decoder are listed in Table 1. At-DH and AtJ-DH network architectures are listed Table 3 and 4.

### Table 3: At-DH Model Structure

<table>
<thead>
<tr>
<th>Input Structure</th>
<th>Encoder</th>
<th>Decoder.A</th>
<th>Decoder.t</th>
</tr>
</thead>
<tbody>
<tr>
<td>As in Table 1</td>
<td>As in Table 2 with X = 3</td>
<td>As in Table 2 with X = 1</td>
<td>As in Table 2 with X = 1</td>
</tr>
</tbody>
</table>

### Table 4: AtJ-DH Model Structure

<table>
<thead>
<tr>
<th>Input Structure</th>
<th>Encoder</th>
<th>Decoder.A</th>
<th>Decoder.t</th>
<th>Decoder.J</th>
</tr>
</thead>
<tbody>
<tr>
<td>As in Table 1</td>
<td>As in Table 2 with X = 3</td>
<td>As in Table 2 with X = 3</td>
<td>As in Table 2 with X = 3</td>
<td>As in Table 2 with X = 3</td>
</tr>
</tbody>
</table>

by the inverse haze model as following:

\[
\hat{J} = \frac{I - \hat{A} \cdot (1 - \hat{t})}{\hat{t}}
\]  (2)

where \( \hat{J} \) is the reconstructed haze-free image, \( \hat{A} := f_A(I) \) and \( \hat{t} := f_t(I) \). \( f_A(\cdot) \) and \( f_t(\cdot) \) denote the outputs of Decoder.A and Decoder.t, respectively. We can also reconstruct the hazy image using the decoders’ outputs and the haze-free image during the training by the inverse haze-model as following:

\[
\hat{I} = J \cdot \hat{t} - \hat{A} \cdot (1 - \hat{t})
\]  (3)

where \( \hat{I} \) is the self-reconstructed hazy image. The training of these two decoders is accomplished by minimizing the following loss function:

\[
L = L_{\ell_2} + \alpha L_{vgg}
\]  (4)

where the \( L_{\ell_2} \) is the (re)construction loss between the hazy images (I and  \( I \)) and the haze-free images (  \( J \) and  \( J \)) during the training:

\[
L_{\ell_2} = \|J - J\|^2 + \|I - I\|^2
\]  (5)

where the  \( J \) and  \( I \) are constructed following Eqs. (2) and (3). Another important loss term in Eq. (4) is the perceptual loss, which is used to ensure that the overall image content is agree with the ground-truth image. To achieve this, the high-level features extracted from the pre-trained VGG network are used. The VGG based regularizer is given by:

\[
L_{vgg} = \sum_{i=1}^{3} \|g_i(\hat{J}) - g_i(J)\|^2 + \sum_{i=1}^{3} \|g_i(\hat{I}) - g_i(I)\|^2
\]  (6)

where the \( g_i(\cdot) \) represents the operator formed by the pre-trained feature extraction layers in the \( vgg16 \) model that ends with layers ReLu1.1, ReLu2.2, and ReLu3.3. The \( L_{\ell_2} \) and \( L_{vgg} \) are balanced by the parameter \( \alpha \).

### 3.3. AtJ-DH Network

The At-DH with self-regularization generates dehazed images with enhanced visual quality. However, for areas in images where the haze is so dense that any scene information in  \( I \) is visually obscured, the two decoders for  \( A \) and  \( t \) do not suffice. Hence, we extend the proposed ‘At’ model by introducing a new J-decoder. Since the J-decoder is trained by treating the hazy observation and corresponding ground truth as training pairs, it is able to provide real scene information during the training process.

Our proposed AtJ-DH network is visually illustrated in Fig. 1 and its architecture is detailed in Table 4. Decoder.A and Decoder.t are trained to estimate the  \( A \) and  \( t \) in the haze model while Decoder.J is trained to directly recover
the haze-free image. The outputs of Decoder.J are concatenated together and sent to the refinement block which is described as Refine.10 to Refine.13 units in Table 2. Several convolutional layers are used to eliminate blocking artifacts that may be generated by the refinement blocks.

In the AtJ-DH network, the outputs of the three decoders are defined as \( A := f_A(I), t := f_t(I) \) and \( J_{\text{direct}} := f_J(I) \) (7)

Compared with the At-DH network, the AtJ-DH network has an additional Decoder.J (Eq. (7)), whose job is to recover the haze-free image directly. By adding Decoder.J, the network would have the ability to make reconstructed scene information be more consistent with ground truth. The separate network system formed by the encoder and Decoder.J make it possible for Decoder.J to generate real scene details even in areas which are full of dense haze.

During the training, Decoder.J provides further guidelines for Decoder.A and Decoder.t to produce better estimations of the scene information. The training of the AtJ-DH is regularized by three loss terms:

- Training Decoder.J by minimizing:
  \[
  L_J = \|J_{\text{direct}} - J\|_2^2 + \beta \sum_{i=1}^{3} \|g_i(J_{\text{direct}}) - g_i(J)\|_2^2
  \]
  (8)
  where the first term is the standard \( \ell_2 \) loss function that is commonly used in regression problems. The second term is the VGG based perceptual loss similar to Eq. (6). With the help of this loss function, Decoder.J essentially learns to recreate the haze-free image based on the hazy input. The ground truth information introduced by training Decoder.J will be later utilized to guide Decoder.A and Decoder.t to better estimate \( A \) and \( t \) which preserve the scene information. \( \beta \) is a weight parameters that controls the relative importance of the loss terms.

- Training Decoder.A and Decoder.t by minimizing:
  \[
  L_{At} = \|\hat{J} - J\|_2^2 + \gamma \|I - \hat{I}\|_2^2
  \]
  + \( \kappa \sum_{i=1}^{3} \|g_i(\hat{J}) - g_i(J)\|_2^2 + \kappa \sum_{i=1}^{3} \|g_i(\hat{I}) - g_i(I)\|_2^2
  \]
  (9)
  where \( \hat{I} \) and \( \hat{J} \) are given in Eq. (3) and (2). The \( L_{At} \) is the same as the loss function of At-DH network, which consists of the \( \ell_2 \) reconstruction loss and the vgg perceptual loss. The \( \gamma \) and \( \kappa \) serve as the trade-off parameters to balance different loss terms selected by cross-validation [38].

- Training Decoder.A, Decoder.t, and Decoder.J together by minimizing:
  \[
  L_{AtJ} = \|J_{\text{direct}} \cdot \hat{t} + (1 - \hat{t}) \cdot \hat{A} - I\|_2^2
  \]
  (10)

As we discussed above, when the haze is too dense, it will be difficult to use Decoder.A and Decoder.t alone to estimate all the scene information. By combining with Decoder.J, Decoder.A and Decoder.t can be learned to reconstruct the haze-free image based on the guidance from the ground truth information introduced by training Decoder.J. It is noted that Decoder.J, A, and t are trained jointly at each step so that the learned information from all three decoders can be integrated for next training steps.

For both networks, the final dehazed images are generated using the estimated t and A following Eq. (2).

### 4. Dataset, Training, and Test Procedure

#### 4.1. Datasets

To train At-DH and AtJ-DH, we use the NTIRE2019-Dehaze dataset [39]. 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 training data consists of 45 hazy images (with dense haze generated for both indoor and outdoor environments) and their corresponding ground truth (haze-free) images with exactly the same scene information.

To learn a network with more powerful generalization ability, we also include the NTIRE2018-Dehaze dataset [40] in training. Compared to the NTIRE2018 dataset, the haze is much denser in the NTIRE2019 dataset. We developed a synthetic method to thicken the haze in NTIRE2018 training images in order to reach the same haze level as in the NTIRE2019. We generated the synthetic dense-haze image following the practice demonstrated in [29] as below:

\[
I_{syn} = I \cdot t + (1 - t) \cdot A
\]
where, \( I_{syn} \) is the synthetic dense-haze image, \( I \) is the NTIRE2018 haze image. \( A \) is chosen to be a fixed value to reduce the uncertainty in variable learning. For indoor images, we set \( A = [0.6, 0.6, 0.6] \) and for outdoor images we set \( A = [0.80, 0.81, 0.86] \) to give a blueish atmosphere light which is estimated by using method described in [21]. The \( \gamma \) is selected uniformly between 0.01 and 0.3 to add dense haze as smaller \( \gamma \) yields more atmosphere information. These values of \( \gamma \) and \( \kappa \) are chosen according to the results of the cross-validation during the development phase. As shown in Fig. 2, the aforementioned synthetic method can generate images with dense haze. During the training, patches of size \( 512 \times 512 \) are extracted from the training images. The augmentations are used as the combination of the following options: i) horizontal flip, rotation

---

3For RGB channels, \( A \) is set to 0.80, 0.81, and 0.86 respectively.
by 90°, 180°, and 270°; 2) scale to 0.7, 0.8, and 0.9 of the original image size. Also, the images (whole image, not patches) are resized to 512 × 512 and applied the same augmentation strategies on these resized images and included them for training.

4.2. Training

Learning the complete network for once is challenging and can result instability in training as the network is highly parameterized. Therefore, we adapted a two-stage training strategy as described below:

Stage 1 - Pre-training of Encoder: We first pre-train the encoder by combining it with a single decoder to make sure the output of the decoder can be reconstructed as the ground truth. The structure of the encoder and the decoder is the same as described in Tables 1 and 2. In this stage, we used the NTIRE19 and synthetically dense-haze images from NTIRE18 to train the encoder for 80 epochs.

Stage 2 - At/AtJ-DH Training: In this stage we combine the encoder trained from stage 1 with newly constructed decoders for estimating A, t, and J. For At-DH, the decoders are as shown in Table 3 and trained using the losses detailed in Section 3.2. For AtJ-DH, the decoders are as shown in Table 4 and trained using the losses detailed in Section 3.3. The training is conducted on two datasets. For the first 50 epochs the training data is the same as in Stage 1. For the next 70 epochs, the training data is only from NTIRE19.

Adam optimizer [41] with initial learning rate of $1 \times 10^{-4}$ is used for training. The learning rate is reduced to its 70% for every 35 epochs and reset in stage 2.

4.3. IRCNN Post-processing

While At/AtJ-DH is already able to recover scene information from the dense-haze input, we also used an optional IRCNN [42] denoiser with $\sigma = 15$ to further improve the results visually. IRCNN method combines the benefits of both model-based and learning-based techniques for image restoration applications. In this dehazing problem, we use the pre-trained CNN denoiser and incorporate it as a post processing unit after the output obtained by our proposed At/AtJ-DH framework. We use At/AtJ-DH+ to indicate that the post processing is present. Detailed results obtained via At/AtJ-DH and At/AtJ-DH+ are listed in Section 5.2.

5. Experimental Results

In this section we present the experimental results of our proposed At/AtJ-DH, that is, ablation study and comparison w.r.t. state-of-the-art methods. The evaluation metrics used to quantify the performance are Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [43].

5.1. Ablation Study

The performances of different configurations of the proposed scene estimation network are investigated in this section. Table 6, 5, and 7 report the results of At-DH and AtJ-DH methods. As shown in the tables, by adding DecoderJ, AtJ-DH further improves both the PSNR and SSIM scores, and produces visually pleasing images (see Fig. 5).

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 [44, 45].

State-of-the-art Methods

The state-of-the-art methods included in the comparisons are: CVPR’09 [17, 46], TIP’15 [47], ECCV’16 [30], TIP’16 [29], CVPR’16 [22], ICCV’17 [1], CVPR’18 [27], and CVPRW’18 [32].

Evaluation Datasets

The comparisons are conducted on the I-HAZE (indoor) and O-HAZE (outdoor) validation datasets [40]. Each of the dataset contains 5 pairs of haze and haze-free image pairs. Detailed acquisition methods of these real-world hazy image pairs are discussed in [40].

Fig. 3 and 4 show the experimental results of the state-of-the-art methods compared with AtJ-DH conducted on NTIRE2018 indoor and outdoor validation datasets. It can be found that AtJ-DH generated much more visually pleasing results. As shown in Table 5 and 6, At/AtJ-DH outperforms other state-of-the-art methods when evaluated on PSNR and SSIM. With the help of the post processing procedure described in Section 4.3, the final dehazed images generated by AtJ-DH+ can obtain the highest scores.

5.3. NTIRE-2019 Dehazing Challenge

The haze presented in images from the NTIRE2019-Dehaze dataset is much denser than 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 the dense haze has covered almost all the scene information. Since AtJ-DH can estimate more accurate scene information from the dense-haze image, the dehazed images generated by AtJ-DH are much more visually pleasing. We evaluate the quantitative performances of the methods on the NTIRE2019 validation set since their ground-truth images are made available [39]. As shown in Table 7, AtJ-DH outperforms all the other state-of-the-art methods. If the post processing procedure described in Section 4.3 is adapted, the final dehazed images generated by AtJ-DH+ can achieve the highest PSNR and SSIM scores.

Table 8 includes the top-6 methods from the contest. It is found that At-DH/+ and AtJ-DH/+ are among the top-performing methods in the NTIRE2019-Dehazing Challenge and outperform other methods by a noticeable margin.

4Code is available at the project page: http://signal.ee.psu.edu/research/ATJDH.html
6. Conclusion

We focus on developing deep learning architectures that estimate physical parameters in the haze model jointly as opposed to separately. Our At-DH network, uses a shared DenseNet encoder and two distinct DenseNet decoders to jointly estimate the scene information \( A \) and \( t \). As a natural extension of At-DH, we develop the AtJ-DH network, which adds one more DenseNet decoder to jointly recreate the haze-free image along with \( A \) and \( t \). Experiments performed on challenging benchmark image datasets of NTIRE’19 and NTIRE’18 demonstrate that At(J)-DH can outperform state-of-the-art alternatives. Notably, in NTIRE’19 results At(J)-DH methods offer nearly a 1 dB gain in PSNR over the next best method. In AtJ-Dh we recover the final dehazed image using enhanced estimates of \( A \) and \( t \) in an inverse haze model; this dehazed image may be fused intelligently with the output of Decoder.J in the AtJ-DH network to further enhance performance.
References


[10] C. Chen, M. N. Do, and J. Wang, “Robust image and video dehazing with visual artifact suppression via gradient resid-