Holistic Attention-Fusion Adversarial Network for Single Image Defogging
Wei Liu, Cheng Chen, Rui Jiang, Tao Lu and Zixiang Xiong

Abstract—Adversarial learning-based image defogging methods have been extensively studied in computer vision due to their remarkable performance. However, most existing methods have limited defogging capabilities for real cases because they are trained on the paired clear and synthesized foggy images of the same scenes. In addition, they have limitations in preserving vivid color and rich textual details in defogging. To address these issues, we develop a novel generative adversarial network, called holistic attention-fusion adversarial network (HAAN), for single image defogging. HAAN consists of a Fog2Fogfree block and a Fogfree2Fog block. In each block, there are three learning-based modules, namely, fog removal, color-texture recovery, and fog synthetic, that are constrained each other to generate high quality images. HAAN is designed to exploit the self-similarity of texture and structure information by learning the holistic channel-spatial feature correlations between the foggy image with its several derived images. Moreover, in the fog synthetic module, we utilize the atmospheric scattering model to guide it to improve the generative quality by focusing on an atmospheric light optimization with a novel sky segmentation network. Extensive experiments on both synthetic and real-world datasets show that HAAN outperforms state-of-the-art defogging methods in terms of quantitative accuracy and subjective visual quality.

Index Terms—Image defogging, holistic attention-fusion mechanism, Fog2Fogfree block, sky segmentation network.

I. INTRODUCTION

In outdoor natural scenes, the captured digital images often experience quality degradation due to bad weather conditions, such as fog, smoke and haze. The resulting foggy images suffer from low contrast, distorted colors, and severe texture information loss, which adversely impact most existing computer vision applications such as image alignment [55], object tracking [42], and remote sensing [14]. It is thus necessary to design effective single image defogging algorithms to restore the content, color, and texture details from the foggy images. Existing image defogging methods can be roughly divided into three classes: prior-based methods [17], [4], [58], [21], [2], fusion-based methods [11], [12], [13], [44], and learning-based methods [5], [40], [11], [53], [50], [52]. Most prior-based methods remove the fog from an image by using or improving an atmospheric scattering model (ASM) [31]. Although these methods work well for some scenes, they often fail to handle other scenes that do not meet the predetermined prior assumptions. For instance, a dark-channel prior does not work well for restoring the sky or white building since there is no dark channel in these areas. Traditional fusion-based defogging methods usually need to combine several derived inputs from the original foggy image to fuse a fine-weight map to remove the fog. The limitation of these methods is that the derived inputs fail to reflect the inherent correlation between the depth information and the foggy image, leading to poor defogging performance when the fog density is high.

In recent years, many learning-based defogging methods have been proposed to address the disadvantages of prior- and fusion-based methods by leveraging the powerful feature extraction ability and spatial mapping capacities of convolutional neural networks (CNN). These methods can be grouped into two categories: supervised and unsupervised defogging methods: for former requires pairs of fog and fog-free images in network training, whereas the latter does not. In addition, supervised defogging mainly includes ASM based defogging networks (ASMDN) [5] [39] [30] [50] [23] and end-to-end defogging networks (EDN) [40] [45] [57] [52] [50]. In ASM, transmission and atmospheric light are two most important parameters. The accuracy of their estimation greatly affects the quality of the recovered image, with estimation inaccuracy resulting in undesired artifacts such as color distortion, halo, and over enhancement. ASMDN exploits CNN to estimate these two parameters before using ASM for defogging. However, ASMDN still suffers if transmission approximations are inaccurate due to the colors of the objects in the scene are inherently similar to the atmospheric lights. Thus, EDN was proposed to directly remove the fog form foggy images by using CNN without ASM. EDN has gradually become the mainstream algorithm for defogging. Although ASMDN and EDN have rectified some shortcomings of traditional methods (e.g., prior-based ones), training their networks require a large number of fog-fogfree image pairs, which are difficult to obtain in practice. Therefore, for a clear image, ASM is usually adopted to synthesize the corresponding foggy image. This synthesis method is simple and efficient, but the distribution of the generated fog is relatively uniform, whereas the atmosphere is heterogeneous. CNN based models trained with this synthesized dataset are not best suited for the foggy images in real-world scenes. A suitable learning-based defogging method not only needs to have good performance, but also fine texture details and high fidelity.

Unsupervised defogging methods [47] [11] [20] [54] have been proposed to address the above issues because they are trained with unpaired real-world foggy and clear images. However, the defogged images generated through these methods usually have low contrast and quality. In this work,
we propose a novel unsupervised defogging method, called holistic attention-fusion adversarial network (HAAN), for improved performance. HAAN shares a similar structure with CycleGAN that it also includes two transformation blocks that we call Fog2FogFree and FogFree2Fog blocks. In each block, there are three modules: a Fog Removal Module (FRM), a Fog Synthetic Module (FSM), and a Color-Texture Recovery Module (CTRM), as shown in Fig. 1, which constitute a dual mapping path. Inside the Fog2Fogfree block, a defogging path is constructed by FRM and FSM (FRM→FSM) to map the image from the fog to fog-free domain, and a color-texture recovery path for defogged result is constructed by CTRM and FSM (CTRM→FSM) to improve image details. Similarly, inside the Fogfree2Fog block, a defogging path is constructed by FRM and FSM (FRM→FSM) to map the image from the fog to foggy domain, and a color-texture recovery path for defogged result is constructed by CTRM and FSM (CTRM→FSM) to improve image details.

In this ill-conditioned equation, both \( T(x) \) and \( A \) are unknown in practice. Thus, if we want to recover \( J \) from \( I \), we need to first determine \( A \) and \( T \). Most state-of-the-art methods either assume prior knowledge of them or use CNN to estimate them before attempting to recover the fog-free image \( J(x) \) according (1). Moreover, in practice, \( T(x) \) represents the medium transmission map, which is related to the depth of the scene via

\[
t(x) = e^{-\beta d(x)}
\]

where \( \beta \) stands for the atmosphere scattering parameter and \( d \) is the depth of the scene.

### B. Supervised methods

As mentioned before, learning-based supervised defogging methods can be mainly divided into two groups: ASMDN and EDN.

**Defogging via ASMDN.** These methods focus on estimating the transmission map \( T(x) \) and atmospheric light \( A \) through CNN before recovering the foggy image by using ASM. Cai et al. [3] proposed a deep CNN based method to extract the multi-features from foggy image to optimize \( T(x) \) with a nonlinear activation function. Similarly, a multiscale network was designed by Ren et al. [29] to estimate \( T(x) \) and \( A \). Pang et al. [36] designed a unified network to jointly estimating \( T(x) \), \( A \) and fog-free images. In [56], Zhu et al. proposed a DehazeGAN, which combines a dense coarse-scale network with a fine-scale network to extract multiscale features from the foggy image, and they designed estimators for \( T(x) \) and \( A \) from these features. Different from most estimation methods, Li et al. [23] reformulated (1) to combine \( T(x) \) and \( A \) as an integrated variable before proposing a light-weight CNN to generate the clean image with this new formulation. Compared with the traditional prior-based methods, ASMDN obtains better estimates of \( T(x) \) and \( A \). However, artifacts will be present in defogged images when these two parameters are estimated incorrectly.

**Defogging via EDN.** These methods directly remove the fog from foggy images to generate fog-free images through CNNs without using ASM. Based on the derived features of an input foggy image, Ren et al. [40] employed an end-to-end trainable gated fusion network for single image defogging that can effectively learn the inherent correlation between fog-related features and the transmission maps. Wang et al. [45] proposed an atmospheric illumination prior in which the luminance channel in the YCrCb color space is mainly disturbed by the atmospheric light in foggy weather. Based on this prior, they adopt an end-to-end multiscale trainable model to restore the Y channel of a foggy image. In [32], the authors presented a Dual-Path Recurrent network, which has two parallel branches to recover the image by simultaneously learning the characteristics of the basic content and details of foggy images. Zhang et al. [50] proposed a densely connected pyramid encoder to extract multiscale image features from foggy inputs before using a density-aware generator to obtain fog-free images; they also constructed a Laplacian pyramid decoder to refine the defogged images. Different from the above methods, Qu et al. [37] treated image defogging as an image-to-image translation problem and proposed an enhanced
Fig. 1. The overall architecture of the proposed HAAN. It consists of two main components: 1) a Fog2Fogfree Block; 2) a Fogfree2Fog Block. Each block has a dual-path mapping mechanism. Inside the Fog2Fogfree block, the bottom sequential units constitute a defogging path to map foggy images to fogfree images, and the top sequential units constitute a color-texture recovery path for defogged result by fusion several derived inputs. Similarly, inside the Fogfree2Fog block, the top sequential units constitute a synthesizing path to map fogfree images to foggy images, and the bottom sequential units constitute a color-texture recovery path for synthesized results.

pix2pix defogging network. This method constructs a multi-resolution generative adversarial network and an enhancer to translate a foggy image to a clear one.

C. Unsupervised methods

Although supervised learning-based defogging methods have achieved remarkable results, they require fog-fogfree image pair to train the network. In practice, it is difficult to synthesize fog around distant objects because the depth information is hard to estimate. Moreover, it is a time-consuming and labor-intensive task to synthesize a large number of foggy images with different fog concentrations. Therefore, unsupervised learning-based defogging methods are better suited for real-world defogging tasks. These methods not only learn the domain correlation between the foggy and clear images in the natural scenes, but also effectively reduce the cost of data preparation. Training on unpaired data, Yang et al. [47] first proposed a disentangled dehazing network consists of three generators and a multi-scale discriminator to produce defogged results from the foggy images. By introducing a perceptual loss function, Engin et al. [11] employed an enhanced CycleGAN [57] to directly generate defogging results. Inspired by [57], we proposed in one of our previous works [26] a network with a two-stage transformation path based on CycleGAN to map foggy images to the fog-free domain; the network in [26] consists of two generators, two enhancers, and a discriminator. In [54], Zhao proposed a two-stage RefineDNet to combine the merits of prior- and learning-based approaches to achieve both visibility restoration and realness improvement. The methods above have a common characteristic: they all use an adversarial generator (AG) to recover foggy images. We thus refer to them as AG-based methods, which typically tend to generate low-quality defogged images with loss of texture detail and low contrast.

D. Attention Mechanism

Attention mechanism has become a high-profile technology in the field of computer vision since it can guide deep neural networks to focus on areas with most valuable information to advance the discriminative learning ability [18] [20]. It has been successfully applied to the low-level tasks such as image super-resolution, deraining, and defogging. Niu et al. [34] proposed a holistic image super-resolution attention network consists of a layer attention module and a channel-spatial attention module to learn the correlation information among layers, channels, and positions. In [20], Jiang et al. presented an improved attention-guided network to remove rain streaks from the input image by first decomposing rain streaks into multiple rain layers and then fusing them through a mixed attention block. To reduce artifacts such as over- and under-defogging introduced by retinex based networks, Li et al. [25] proposed a channel and spatial attention mechanism that can automatically adjust the channel- and pixel-wise attention weights. Liu [27] proposed GridDehazeNet to address the bottleneck effect of information flow in the hierarchical architecture while allowing efficient information exchange across different scales. Although the above methods can leverage the feature extraction ability of neural networks, they all train on paired image. There exist few studies on application of attention mechanisms to unpaired training due to diversity in training samples in unsupervised learning. However, attention mechanisms can be introduced into unsupervised learning if their strong feature extraction capability is able to effectively compensate for sample diversity in training.
III. PROPOSED MODEL

A. Overview

As illustrated in Fig. 1, our proposed HAAN consists of a Fog2Fogfree block and a Fogfree2Fog block. In these two blocks, there are three modules, including Fog Removal Module (FRM, $M_R$), Fog Synthetic Module (FSM, $M_S$) and Color-Texture Recovery Module (CTRM, $M_{CTR}$). As can be seen from Fig. 1 different modules combine to form four different mapping path, which are defogging path (composed of $M_R$ and $M_S$), synthesizing path (composed of $M_S$ and $M_R$), color-texture recovery path for defogged result (composed of $M_{CTR}$ and $M_S$ ) and color-texture recovery path for synthesized result (composed of $M_R$ and $M_{CTR}$). Therefore, our method forms a game learning process of confrontation by these four paths to achieve the mapping transformation from the foggy images to the fog-free images domain. Moreover, the functions of $M_R$, $M_S$ and $M_{CTR}$ are mainly completed by our proposed Defogging generator, Synthesizing generator and Holistic Attention-Fusion generator respectively.

For Fog2Fogfree block, its dual-path consists of a defogging path and a color-texture recovery path for defogged result. Given an input real-world foggy image $I_{rf}$, we first obtain the defogged image $I_{df}^{rf}$ by

$$ I_{df}^{rf} = M_R(I_{rf}) $$

(3)

Then, $I_{df}^{rf}$ is fed to fog synthetic module to reconstruct the first foggy image $I_{rcf}^{rf}$,

$$ I_{rcf}^{rf} = M_S(M_R(I_{rf})) = M_S(I_{df}^{rf}) $$

(4)

The above process we call it defogging path. To further improve the performance of $M_R$, the color-texture recovery path for defogged result is proposed. This path first takes the defogged image $I_{df}^{rf}$ and several derived images of the $I_{rf}$ as the input, and generated a recovered result $I_{rr}^{rf}$,

$$ I_{rr}^{rf} = M_{CTR}(I_{df}^{rf}, I_{df}^{rf}) $$

(5)

where $I_{df}^{rf}$ denotes the derived images of $I_{rf}$. In this paper, we select three derived images for $I_{rf}$. Then, we put this result again into fog synthetic module to reconstruct the second foggy image $I_{rcf}^{2}$ by

$$ I_{rcf}^{2} = M_S(M_{CTR}(I_{df}^{rf}, I_{df}^{rf})) = M_S(I_{rr}^{rf}) $$

(6)

Finally, we can obtain two cycle consistency loss functions for foggy images among $I_{rf}$, $I_{rcf}^{rf}$, and $I_{rcf}^{2}$. Note that, the defogged result obtained by fog removal module is our goal. The color-texture recovery path for defogged result is to constrain the fog removal module to generate results with better color fidelity and texture details.

For Fogfree2Fog block, there still has a dual-path, which consists of a synthesizing path and a color-texture recovery path for synthesized result. Given an input real-world fogfree image $I_{rf}$, the synthesizing path is first fed it to fog synthetic module to generate a synthesized result $I_{sf}$, and then use $I_{sf}$ as the input of the fog removal module to generate the reconstruct fogfree module to generate the reconstruct fogfree image $I_{rr}^{rf}$,

$$ I_{sf} = M_S(I_{rf}) $$

$$ I_{rr}^{rf} = M_R(I_{sf}) $$

(7)

For color-texture recovery path for synthesized result, we first take the reconstruct fogfree image $I_{rr}^{rf}$ and the derived images $I_{df}^{rf}$ produced by $I_{sf}$ as input. Then we recovery this result by using color-texture recovery module.

$$ I_{rr}^{rf} = M_{CTR}(I_{rr}^{rf}, I_{df}^{rf}) $$

(8)

where $I_{rr}^{rf}$ is the recovered image for synthesized result $I_{rf}$. Similarly, two cycle consistency loss functions for fogfree images are constructed among $I_{rf}$, $I_{rr}^{rf}$, and $I_{rf}^{sf}$.

B. Network Architecture

As discussed in III-A each module has a major generator to perform its corresponding functions (as shown in Fig. 1), which are Defogging generator for $M_R$, Synthesizing generator for $M_S$, and Holistic Attention-Fusion generator for $M_{CTR}$. Moreover, another major component of our HAAN is the discriminator.

![Fig. 2. The architecture of the defogging generator.](image)

Defogging generator: As can be seen from Fig. 2 we show the architecture details of the defogging generator. Inspired by [10], we design a densely-residual connected encoder-decoder for remove the fog from the foggy image. Compared with the structure of densely connect alone, introducing the residually connect block in our architecture, the generator can not only better learn the correlation information from shallow layers to deep layers, but also can better preserve the image content and texture details for defogged results. As shown in Fig. 2 the encoder contains two convolution blocks(ConvBlocks) and two dense blocks (DenseBlocks), which including a series of convolutional, concatenation, batch normalization(BN), pooling layers and ReLU layers. After the encoder, six residual blocks (ResBlocks) are connected. For encoder, given a foggy image $I_f$, we first extract the shallow feature $F_C^1 \in \mathbb{R}^{64 \times H \times W}$ by the first convolution block:

$$ F_C^1 = C_{7 \times 7}(I_f), $$

$$ C_{k \times k} = \delta(\eta(C_{k \times k}(.))) $$

(9)

where $C_{k \times k}(.)$ denotes the ConvBlock operation, including convolution operation $C_{k \times k}$ where kernel size is $k \times k$, batch normalization function $\eta$, and rectified linear unit (ReLU) activation function $\delta$. Then, we fed $F_C^1$ to the first DenseBlock to distort the first encoding feature $F_{en}^1 \in \mathbb{R}^{128 \times H \times W}$:

$$ F_{en}^1 = \left[ \mathcal{P}(F_C^1), \mathcal{P}(C_{1 \times 1}(\eta(D_{3 \times 3}(F_C^1))) \right] $$

(10)
where $D_{k \times k}(\cdot)$ is the DenseBlock operation, which has three ConvBlocks (as shown in Fig. 2). $\mathcal{P}$ represents concatenation operation, $\mathcal{P}$ is the pooling operation. Finally, $F_{en}^{1}$ is fed to the second ConvBlock and DenseBlock to obtain the final encoding feature $F_{en}^{2} \in \mathbb{R}^{256 \times H/4 \times W/4}$.

$$F_{en}^{2} = [\mathcal{P}(F_{en}^{2}), \mathcal{P}(C_{1 \times 1}(\sigma(D_{3 \times 3}(F_{en}^{2}))))]$$

where $F_{en}^{2} = C_{3 \times 3}(F_{en}^{1})$. Moreover, we know that increasing the network depth can improve the representational ability of the network. However, to avoid the problem of gradient disappearance caused by increasing the network depth, we use the residual block to further refine the encoding feature $F_{en}^{2}$. The residual features extracted from the ResBlock is expressed as follows:

$$F_{re}^{i} = \mathcal{R}_{k \times k}^{i}(F_{re}^{i-1}), i \in [1, N]$$

where $F_{re}^{0} = F_{en}^{2}$, $\mathcal{R}_{k \times k}(\cdot)$ represents the i-th ResBlock operation, $N$ is the number of the ResBlock. In this work, $N = 6$.

For decoder, as shown in Fig. 2, it contains two upsample blocks (UpsampleBlocks) and two convolution blocks(ConvBlocks). Inside decoder, to preserve the image content and recover the details from the input, we concatenate the shallow features. In our decoder, after each upsampling, its output is spliced with that of the convolution block in the encoding stage, and then the convolution block in the decoding stage is used to refine the splicing result to obtain the decoding feature. The first decoding feature $F_{de}^{1} \in \mathbb{R}^{128 \times H/2 \times W/2}$ is formed by

$$F_{de}^{1} = \delta(C_{1 \times 1}(C_{3 \times 3}(\mathcal{U}_{k \times k}(F_{re}^{1}), F_{en}^{2})))$$

where $\mathcal{U}_{k \times k}(\cdot)$ is the UpsampleBlock operation where kernel size is $k \times k$. Similarly, the second decoding feature $F_{de}^{2} \in \mathbb{R}^{64 \times H \times W}$ is got by

$$F_{de}^{2} = \delta(C_{1 \times 1}(C_{3 \times 3}(\mathcal{U}_{k \times k}(F_{de}^{1}), F_{en}^{2})))$$

Then, through a convolution operation and a Tanh activation function, we obtain the defogged result $I_{df} \in \mathbb{R}^{3 \times H \times W}$.

$$I_{df} = \tau(C_{7 \times 7}(F_{de}^{2}))$$

where $\tau$ denotes the Tanh activation function.

**Synthesizing generator:** In practice, due to the diversity and complexity of fog distribution, it is very difficult to directly use CNN to synthesize the foggy image. Thus, as shown in Fig. 3, the ASM is embedded in our synthesizer generator. We design two CNN-based models to optimize the transmission $T$ and atmospheric light $A$ respectively. Especially for atmospheric light $A$, we propose a sky segmentation model (SSM, as shown in Fig. 5). To obtain finer transmission $T$, we use 5 convolution layers. As shown in Fig. 3 we first take a clear image ($J$) and the maximum pixel value in its channel as the input, and then extract the roughly feature by using a convolution operation. After this operation, we use three convolutional neural networks with ReLU function to refine the roughly feature. Finally, the transmission $T$ is generated by a convolution network with a sigmoid function.

$$T = \sigma(C_{3 \times 3}(\mathcal{M}(\mathcal{M}(C_{9 \times 9}([J, \text{Max}(J)]))))$$

where $\mathcal{M}$ denotes a $3 \times 3$ convolution neural network with a ReLU function, $\sigma$ denotes the sigmoid function. $\text{Max}(\cdot)$ means to take the maximum pixel value for a image. Fig. 4 illustrates three transmission maps generated by our proposed network. It can be seen that each transmission map can effectively estimate the depth information of the scene and preserve richer texture details.

To optimize $A$, in our previous work [26], we propose a sky prior. For the foggy image with sky region, we assume that the depth of the sky is regarded as infinity, and define it as the intensity of pixels in the area of maximum fog density. Then, the average value of the sky region is described as $A$. For more details about this derivation process, please refer to our previous work [26]. Therefore, how to divide the sky region from the foggy image is crucial. Instead of using the existing technology [16] [22] [35] to segment the image to obtain the sky region, we propose a CNN-based sky segmentation model (SSM).

As shown in Fig. 3 SSM consists of an Enhance Block and a Segmentation Block. Each block adopts the encoder-decoder network. For a outdoor image, we observed that the boundary between sky area and non-sky area was often blurred. Thus, we first enhance the outdoor image to improve the edge information by using an enhance network, which can sharpness this boundary and label the sky region. Inside enhance block, we also use the dense connection to improve the texture details of the extracted features. For an input outdoor image $I_{M}$, the output of each layer in encoding stage is $E_{M}^{i}$, and $D_{M}^{i}$ is the...
output of each layer in decoding stage. The encoding process is defined as: 
\[ E^k_e = \delta(C_{4x4}(E^{i-1}_e)) \]  
(17)
where \( E^0_e = I_M \). The decoding process is defined as:
\[
\begin{align*}
D^1_e &= \mathcal{U}_{4x4}(E^0_e), \\
D^{k+1}_e &= \mathcal{U}_{4x4}([D^k_e, \delta(I^{m-k}_e)], k \in [1, n - 1])
\end{align*}
\]  
(18)
where \( i \) and \( j \) are layer indices \((i = 1, ..., m; j = 1, ..., n, m \) and \( n \) are the number of layers). In this work, \( m = n = 3 \). Finally, the enhanced image \( E_M \) is generated by concatenating all encoding and decoding features followed by two convolution layers.

In segmentation block, we add residual blocks in addition to encoding network and decoding network (as shown in Fig. 3). Similarly, we defined the outputs of each layer in encoding, decoding, and residual network as \( E^i_1, D^i_1, \) and \( R^k \) respectively \((i = 1, ..., m_1; j = 1, ..., n_1; k = 1, ..., K; m_1, n_1 \) and \( K \) are the number of layers). In this block, \( m_1 = 3, n_1 = 2, \) and \( K = 6 \). Therefore, the encoding process in segmentation block is formulated as:
\[ E^i_s = \delta(\eta(C_{7x7}(E^i_M))), E^s_2 = \mathcal{M}(E^1_s), E^s_3 = \mathcal{M}(E^2_s) \]
(19)
For \( R^k_s \), we use the equation 12 to obtain. Note that, \( \mathcal{F}_{re} = E^2_s \). Then, the decoding process is formulated as:
\[ D^1_s = \mathcal{U}_{3x3}(R^0_s), D^2_s = \mathcal{U}_{3x3}(\mathcal{M}([D^1_s, E^1_s])) \]
(20)
After the above process, the final output segmentation image \((O_M \in [-1, 1])\) is defined as:
\[ O_M = \tau(C_{7x7}(\mathcal{M}([D^1_s, E^1_s]))) \]
(21)
Thus, the atmospheric light \( A \) is calculated as follows:
\[ A = mean(O_M \odot I_M) \]
(22)
where \( mean(.) \) denotes an average filter, \( \odot \) denotes the element-wise product. However, due to the small capacity of our GPU, we used a pre-trained SSM to segment the sky region to improve the training efficiency of our HAAN. Please refer to our online address (https://github.com/ChengChen-ai/Sky-Segmentation) for details about our SSM. Moreover, we proposed a foggy-sky image dataset (FSID) for sky segmentation research. It contains 900 natural outdoor scenes, including traffic scenes, landscape, water scenes, etc. In addition, similar to [26], if the image has no sky region, we followed by the dark channel [17] that picked the maximum of the top 0.1 percent brightest pixels in the image as \( A \).

**Holistic Attention-Fusion generator:** To further improve the ability of transformation to map fog images to a fogfree image domain through the defogging generator, the output of \( M_R \) and three derived images of the foggy image are input into a color-texture recovery network to refine correlated information for generating a better-recovered image with a color coherence and exactness of detail. We refer to this network as Holistic Attention-Fusion generator, as shown in Fig. 5. Inspired by [40], we also use the gated fusion strategy to recover the color and visibility of the defogged image with three derived inputs of the foggy image, such as white balanced image \( I_{wb} \), contrast enhanced image \( I_{ce} \), and gamma corrected image \( I_{gc} \).

In past studies, white balance has been found to have the potential to restore color in scenes, which is intended to eliminate chromatic aberrations caused by atmospheric colors. In this paper, we use the technology based on the gray world hypothesis [40] to obtain the white balance feature map of foggy images. The formula is as follows:
\[
\begin{align*}
I(\hat{R}) &= I(R) \ast k_r, \\
I(\hat{G}) &= I(G) \ast k_g, \\
I(\hat{B}) &= I(B) \ast k_b
\end{align*}
\]
(23)
\[ I_{wb} = [I(\hat{R}), I(\hat{G}), I(\hat{B})] \]
(24)
where, \( k_r, k_g \) and \( k_b \) represent the gain coefficients of R, G, and B channels respectively. \( I_{wb} \) represents the result of a white balance operation.

To further improve the contrast of the results, we introduce the following two derived inputs. For \( I_{ce} \), we exploit the Ancuti [11] proposed method to improve the global visibility of haze areas in foggy images.
\[ I_{ce} = \mu(I - \bar{T}) \]
(25)
where \( I \) is the foggy input, \( \bar{T} \) is the average luminance value of \( I, \mu \) is a linearly increase factor, \( \mu = 2(0.5 + \bar{T}) \). However, it can be seen from [25] that as the value of \( \bar{T} \) increases, \( (I - \bar{T}) \) may be negative, resulting in the dark area in foggy image tending to black after contrast enhancement. To address this problem, we adopt gamma correction:
\[ I_{gc} = \alpha I^\gamma \]
(26)
As discussed in [40], we also set \( \alpha = 1 \), and the decoding gamma correction \( \gamma = 2.5 \). Through gamma correction, we not only enhanced the visibility of the original foggy image, but also effectively avoided the appearance of the severe dark area.

After obtaining these derived inputs, how to fuse them together to optimize the defogged result is the key of this paper. In [40], however, Ren et al. ignored the influence of channel features and spatial features of each layer in the network on confidence maps respectively, resulting in the loss of partial texture details. Thus, we design a simple but effective method combining channel attention and spatial attention to fuse the confidence maps of each input to recover a higher quality image from foggy image, which can not only learns the hierarchical-specific knowledge from all preceding layers but also reduces the feature redundancy effectively. As shown
in Fig. 5, we first concatenated each input image according to the channel direction to obtain the concatenation feature map \( F_{ct} \):

\[
F_{ct} = [I_{df}, I_{ec}, I_{gm}, I_{wb}] \tag{27}
\]

where \( I_{df} \) is the output of \( M_R \). Then, a layer of convolutional network is used to initialize the global spatial information of the concatenation feature map to highlight the effective information of each channel. Thereafter, we use a channel attention network to assign weights to different channels of \( F_{ct} \). For channel attention network, we first use global average pooling and maximum pooling to compress the global spatial features of channels into two different spatial feature expressions: \( g_{avg} \in \mathbb{R}^{C \times 1 \times 1} \) and \( g_{max} \in \mathbb{R}^{C \times 1 \times 1} \). Then, we generate the channel-attention map \( W_{ct} \in \mathbb{R}^{C \times 1 \times 1} \) through a shared network:

\[
W_c = \sigma(C_{3 \times 3}((C_{3 \times 3}(g_{avg}^c))) + C_{3 \times 3}((C_{3 \times 3}(g_{max}^c)))) \quad c \in \{df, wb, ec, gm\} \tag{28}
\]

where \( W_{df} \) is the weight of \( I_{df} \), \( W_{wb} \) is the white balanced weight, \( W_{ec} \) is the contrast enhanced weight, and \( W_{gm} \) is the gamma correction weight. Finally, we get the preliminary fusion feature \( F_{ct}^* \) through the following formula:

\[
F_{ct}^* = W_c \cdot F_{ct} = (W_{df}W_{ec}W_{gm}W_{wb}) \cdot (I_{df}I_{ec}I_{gm}I_{wb}) = W_{df} \odot I_{df} + W_{ec} \odot I_{ec} + W_{gm} \odot I_{gm} + W_{wb} \odot I_{wb} \tag{29}
\]

where \( \odot \) denotes inner-vector product. To further uptake of channel dimension information, we propose a spatial attention network, which can contain responses from all dimensions of the feature map \( F_{ct}^* \). It pays more attention to the location features information by assigning greater weight to high-frequency signals and dense fog areas. Along the channel direction, we calculate the mean value \( g_{avg}^{s} \in \mathbb{R}^{1 \times H \times W} \) and maximum value \( g_{max}^{s} \in \mathbb{R}^{1 \times H \times W} \) of \( F_{ct}^* \) in the spatial dimension to effectively highlight fog relevant feature to preserve more texture details for generated result. Then we concatenate these two spatial features and put them into a convolution layer and a sigmoid function to obtain the weight \( W_s \in \mathbb{R}^{1 \times H \times W} \) of \( F_{ct}^* \):

\[
W_s = \sigma(C_{3 \times 3}(g_{avg}^s, g_{max}^s)) \tag{30}
\]

The final attention-fusion feature is calculated as follows:

\[
F_{fusion} = W_s \odot F_{ct}^* \tag{31}
\]

### C. Loss Function

To train our proposed holistic attention-fusion adversarial defogging model, our loss function consists of three terms: 1) adversarial loss \( L_{adv} \), \( L_{CTR} \), \( L_{sadv} \); 2) cycle-consistency loss \( L_{c1} \), \( L_{c2} \); 3) perceptual loss \( L_{per} \). The total loss of our network is presented as:

\[
L_{Total} = \lambda_1 L_{adv} + \lambda_2 L_{CTR} + \lambda_3 L_{sadv} + \lambda_4 L_{c1} + \lambda_5 L_{c2} + \lambda_6 L_{per} \tag{32}
\]

where \( \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5 \) and \( \lambda_6 \) are the positive weights, which are used to balance the importance of the corresponding loss.

#### Adversarial Loss:

We considered three losses of adversarial learning including \( L_{adv} \), \( L_{CTR} \) and \( L_{sadv} \), where \( L_{adv} \) and \( L_{CTR} \) fooled the discriminator \( D_f \) by encouraging Defogging generator and Holistic Attention-Fusion generator to recover high-quality clean image, and \( L_{sadv} \) aims to fool the \( D_f \) using Synthesizer generator to synthesize a more realistic foggy image. These adversarial losses are defined as:

\[
L_{adv}(M_R, D_f) = \mathbb{E}_{y_r \sim Y_R} [\log D_f(y_r)] + \mathbb{E}_{x_r \sim X_R} [\log(1 - D_f(M_R(x_r)))] \tag{33}
\]

\[
L_{CTR}(M_R, M_{CTR}, D_f) = \mathbb{E}_{y_r \sim Y_R} [\log D_f(y_r)] + \mathbb{E}_{x_r \sim X_R} [\log(1 - D_f(M_{CTR}(M_R(x_r), I_d^r)))] \tag{34}
\]

\[
L_{sadv}(M_S, D_f) = \mathbb{E}_{y_r \sim Y_R} [\log D_f(y_r)] + \mathbb{E}_{y_r \sim Y_R} [\log(1 - D_f(M_S(y_r)))] \tag{35}
\]

where \( X_R \) represents the real-world fog image domain and \( Y_R \) refers to the real-world clear image domain; \( x_r \) and \( y_r \) represent the input image from \( X_R \) and \( Y_R \) respectively. \( M_R(\cdot) \), \( M_{CTR}(\cdot) \) and \( M_S(\cdot) \) represent the results generated by Fog Removal Module, Color-Texture Recovery Module and Fog Synthetic Module respectively; \( I_d^r \) denotes the derived images of \( x_r \).

#### Cycle-consistency Loss:

We applied a mean-square-error loss to make the final output of each path in the both Fog2Fogfree and Fogfree2Fog block close to the corresponding initial input. For Fog2Fogfree block, we obtained cycle-consistency loss function for foggy images among \( I_r, I_{rf} \), and \( I_{rf}^r \):

\[
L_{c1} = \| I_r - I_{rf}^r \|^2 + \| I_{rf} - I_{rf}^r \|^2 \tag{36}
\]
where $I_{rf}$ is the input of real-world foggy image, $I_{rcf}$ and $I_{rtf}$ represent the output of the defogging path and the color-texture recovery path respectively. For Fogfree2Fog block, cycle-consistency loss function for fogfree images is constructed among $I_{rf}$, $I_{rcf}$, and $I_{rtf}$:

$$L_{c2} = \|I_{rf} - I_{rcf}\|^2 + \|I_{rf} - I_{rtf}\|^2 \quad (37)$$

where $I_{rf}$ is the input of real-world clean image, $I_{rcf}$ and $I_{rtf}$ denote the output of the synthesizing path and the color-texture recovery path respectively.

Perceptual Loss: To better extract more texture details from foggy images, we introduce a perceptual loss function, which is constructed with the pretrained VGG19 [45] to compare the original input image with the reconstructed image in the feature space. This objective is defined as:

$$L_{per} = \|\phi_i(I_{rf}) - \phi_i(I_{rcf})\|^2 + \|\phi_i(I_{rf}) - \phi_i(I_{rtf})\|^2 + \|\phi_i(I_{rf}) - \phi_i(I_{rcf})\|^2 + \|\phi_i(I_{rf}) - \phi_i(I_{rtf})\|^2 \quad (38)$$

where $\phi_i$ denotes the feature map in $i^{th}$ layer of VGG19 network.

IV. EXPERIMENTS AND DISCUSSION
A. Implementation Setting and Details
1) Evaluation methods
In this section, we compare the defogged performance of our proposed approach with that of eleven state-of-the-art methods. Among them, some are trained using paired data, including the DehazeNet [5] (TIP 2016), a densely connected pyramid dehazing network (DCPDN) [49] (CVPR 2018), an enhanced pix2pix dehazing network (EPDN) [57] (CVPR 2019), a gated context aggregation network (GCANet) [6] (WACV 2019), a domain adaptation for image dehazing (DA-Dehaze) [41] (CVPR 2020), a multi-scale boosted dehazing network (MSBDN) [9] (CVPR 2020) and a principled synthetic-real dehazing (PSD) [7] (CVPR 2021). While other methods use unpaired data for training, including the Cycle-Dehaze [11] (CVPR Workshops 2018), an image dehazing and exposure (IDE) [21] (TIP 2021), a weakly supervised dehazing Refinement framework (RefineDNet) [54] (TIP 2021) and a dehazing via decomposing transmission map into density and depth ($D^3$) [48] (CVPR 2022).

2) Implementation details
Due to our proposed HAAN uses unsupervised training, we randomly choose both outdoor synthetic and real-world unpaired hazy images from the RESIDE dataset [24]. The dataset is widely used, which is divided into five subsets: Indoor Training Set (ITS), Outdoor Training Set (OTS), Synthetic Object Testing Set (SOTS), Unannotated real Hazy Images (URHI) and real Task-driven Testing Set (RTTS). We train the network by randomly selecting 10000 synthetic foggy images from the OTS and 8000 real foggy images from the URHI and RTTS.

Our HAAN is implemented by PyTorch 1.8.0 with one NVIDIA GeForce GTX 3060 GPU. During training, we resize all the images to $256 \times 256$ and normalize the pixel values to [-1,1]. The models are trained using ADAM optimizer with exponential decay rates $\beta$ equal to 0.999, meanwhile the learning rate and batchsize are initialised to $1 \times 10^{-4}$ and 2, respectively. We set the total number of epoch to 15 for 270000 iterations. The perceptual loss in (35) is set using the latent features of the 2nd and 5th layers from the fixed pretrained VGG19, and empirically initialise default values of $\lambda_1$, $\lambda_2$, $\lambda_3$, $\lambda_4$, $\lambda_5$ and $\lambda_6$ in (32) are set to 10, 10, 10, 5, 5, 1 respectively.

3) Evaluation datasets and metrics
We evaluate the proposed method on two synthetic datasets (SOTS, HazeRD) and two real-world datasets (O-HAZE, LIVE). For the synthetic datasets, SOTS comes from the RESIDE dataset, which contains 500 indoor scenes and 500 outdoor scenes. HazeRD synthesized by Zhang et al. [53] which contains 75 synthetic outdoor foggy images of different fog concentrations. The real-world foggy dataset O-HAZE [3] used in the NTIRE 2018 image dehazing challenge. It contains 45 outdoor real-world hazy images, and the LIVE dataset provided by Choi et al. [8] which contains 500 outdoor real-world foggy images.

To evaluate the performance of our method, we utilize reference indicators and non-reference indicators as the evaluative metrics. Like most defogging papers, the Structure Similarity (SSIM) [46], the Peak Signal to Noise Ratio (PSNR) [19] and the Learned Perceptual Image Patch Similarity (LPIPS) [51] are usually used as criteria to evaluate the quantitative performance of defogging methods. In addition, as the corresponding ground truth of real-world foggy images is extremely difficult to available, the performance of defogged results on these real-world images was assessed by no-
parameter models including Fog Aware Density Evaluator (FADE) [8], the Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) [29], Natural Image Quality Evaluator (NIQE) [30] and Blind Assessment based on Visibility Enhancement (BAVE) [15]. Among them, FADE predicts the visibility of a single foggy image to perceive the fog density. The lower the value of the FADE, the stronger the defogged ability. BRISQUE uses natural scene statistics to quantify losses of “naturalness” in the image due to the distortions. The score of the BRISQUE ranges from 0 to 100, the closer to 0 the better the defogged quality. NIQE is a space domain natural scene statistics-based quality aware collection for predicting the quality of defogged image. The lower the value of the NIQE, the better the defogged effect. For BAVE, we consider the evaluation indicator $\gamma$, which represents the mean ratio of the gradients at visible edges. The higher the value of $\gamma$, the better the defogged performance.

B. Ablation Studies

To demonstrate the effectiveness of our HAAN architecture, we perform a series of ablation analyses on the different modules of our proposed HAAN. We mainly consider the following four factors: 1) w/o VGG: our full model without the VGG perceptual losses; 2) w/o DC: our full model without the proposed densely-residual connected encoder-decoder; 3) w/o ASM: our full model without the atmospheric scattering model; 4) w/o HAG: our full model without the holistic attention-fusion generator. Moreover, we also compare our method to CycleGAN [57] and Cycle-Dehaze [11].

TABLE I

|         | PSNR [19] | SSIM [46] |
|---------|-----------|-----------|
| CycleGAN [57] | 14.7115  | 0.5024    |
| Cycle-Dehaze [11] | 20.9274  | 0.7431    |
| w/o VGG   | 21.2083  | 0.9073    |
| w/o DC    | 22.5896  | 0.8795    |
| w/o ASM   | 23.2111  | 0.8467    |
| w/o HAG   | 25.7661  | 0.9331    |
| HAAN      | 29.3991  | 0.9646    |

The qualitative and quantitative results are shown in Fig. 7 and Table I. It demonstrates that HAAN achieves the best defogging performance in terms of visual effect and quantitative accuracy. As can be seen in Fig. 7(b), the CycleGAN’s results have the worst defogging performance, which have the serious color distortion and the texture details are completely lost. Compared to CycleGAN, we can see from Fig. 7(c) that the texture information of the Cycle-Dehaze’s results are restored better due to the VGG perceptual loss can preserve more image content. However, as shown in the zoom-in regions in Fig. 7(c), the color board (1st row) and the sign (2nd row) are blurred. Fig. 7(d) shows the results obtained by using HAAN without VGG perceptual losses. We can see that there is a significant amount of fog in the defogged results and the sign is also blurred. In addition, without using the proposed DC module, there is a serious chessboard phenomenon in some regions.

Fig. 8. Qualitative comparison between the proposed HAAN and the state-of-the-art methods on the SOTS dataset.
This indicates that our method can achieve the best defo- 
ginning performance. To further demonstrate the effectiveness 
of our HAAN, we quantitative compare on SOTS on outdoor 
dataset for different configurations. As shown in Table I, our 
HAAN has the best values of PSNR and SSIM.

C. Performance Comparison with State-of-the-arts

In this subsection, to verify the effectiveness of our proposed 
method, we qualitatively and quantitatively compared against 
the state-of-the-arts methods on both synthetic and real-world 
datasets.

1) Comparison on synthesized foggy images

Fig. 8 and Fig. 9 illustrate the defogging results on SOTS and 
HazeRD by our method and other methods, respectively. In Fig. 8(b)-(c),(h)-(j),(l), there is a significant amount of fog in the defogged results of DehazeNet, DCPDN, PSD, Cycle-Dehaze, IDE, RefineDNet, and D4. As can be seen in Fig. 8(h)-(j), although PSD, Cycle-Dehaze, and IDE are able to improve image

(e.g., the color board and the sign in Fig. 7(e)). It demonstrates 
that the proposed DC module could help the network preserve 
richer image content and texture details. As shown in Fig. 7(l), 
the defogged results generated by our HAAN without ASM 
module have color distortion phenomenon. The reason is due 
to the atmospheric scattering model can guide the synthesizing 
network to generate a more natural foggy image to improve the 
mapping ability from foggy image domain to fogfree image 
domain. Finally, compared to the defogged results generated 
by our HAAN without HAG module (as shown in Fig. 7(g)), 
our HAAN’s results (as shown in Fig. 7(h)) are clearer and 
more pleasing in visual effect. It demonstrates that HAG can 
recover the color-texture information of the defogged results.

Moreover, the above analysis is also reflected in indicator 
NIQE. It can be seen from Fig. 7, the NIQE values of our 
method are the smallest, followed by w/o VGG, w/o ASM, 
w/o DC, w/o HAG, Cycle-Dehaze, and CycleGAN. This indicates that our method can achieve the best defo-

| Dataset | SOTS [24] | HazeRD [53] |
|---------|-----------|-------------|
| Metric  | PSNR [19] | SSIM [46] | LPIPS [51] | FADE [8] | BRISQUE [29] | NIQE [30] | PSNR [19] | SSIM [46] | LPIPS [51] | FADE [8] | BRISQUE [29] | NIQE [30] |
| DehazeNet [5] | 23.483 | 0.8716 | 0.0524 | 0.5374 | 10.9584 | 3.6732 | 1.5170 | 15.7240 | 0.8234 | 0.1513 | 0.8533 | 24.6961 | 4.0871 | 2.3394 |
| DCPDN [49] | 25.8069 | 0.9214 | 0.1087 | 0.5365 | 30.8489 | 4.1004 | 1.4108 | 14.5950 | 0.874 | 0.1313 | 1.1316 | 15.1890 | 3.8303 | 1.3136 |
| EPDN [37] | 21.0540 | 0.8716 | 0.0718 | 0.9340 | 21.8789 | 3.8303 | 1.3136 | 17.3982 | 0.8234 | 0.1513 | 0.8533 | 25.9086 | 5.8394 | 2.3394 |
| GCANet [6] | 27.4001 | 0.9380 | 0.0161 | 0.6438 | 17.3982 | 3.4035 | 1.5959 | 13.5029 | 0.874 | 0.1313 | 1.1316 | 15.1890 | 3.3586 | 1.4635 |
| DA-Dehaze [41] | 21.0540 | 0.8716 | 0.0718 | 0.9340 | 21.8789 | 3.8303 | 1.3136 | 17.3982 | 0.8234 | 0.1513 | 0.8533 | 25.9086 | 5.8394 | 2.3394 |
| MSBDN [9] | 27.4001 | 0.9380 | 0.0161 | 0.6438 | 17.3982 | 3.4035 | 1.5959 | 13.5029 | 0.874 | 0.1313 | 1.1316 | 15.1890 | 3.3586 | 1.4635 |
| PSD [7] | 27.4001 | 0.9380 | 0.0161 | 0.6438 | 17.3982 | 3.4035 | 1.5959 | 13.5029 | 0.874 | 0.1313 | 1.1316 | 15.1890 | 3.3586 | 1.4635 |
| HAAN | 27.4001 | 0.9380 | 0.0161 | 0.6438 | 17.3982 | 3.4035 | 1.5959 | 13.5029 | 0.874 | 0.1313 | 1.1316 | 15.1890 | 3.3586 | 1.4635 |

Fig. 9. Qualitative comparison between the proposed HAAN and the state-of-the-art methods on the HazeRD dataset.
brightness, some results appeared varying degrees of coloration and over-enhancement. Furthermore, the defogged results generated by DCPDN [49] (as shown in the second row of Fig. 9(c)) have completely lost most of the structural information when processing images with dense fog. We can observe that the defogged results of EPDN [37], GCANet [6], DA-Dehaze [41] and RefineDNet [54] (as shown in Fig. 8(d)-(f),(k) and Fig. 9(d)-(f),(k)) exhibit low brightness and artifacts, which is caused by excessive defogging. For the defogged results in Fig. 8(g) and Fig. 8(m), both MSBDN [9] and our HAAN can thoroughly remove the fog while retaining a clear texture. However, the defogged results of MSBDN [9] (as shown in Fig. 9(g)) appear over-enhancement in the sky region. In comparison, our model can remain sharper edge contours and the results (as shown in Fig. 9(m)) are closer to the ground truth images visually.

To quantitatively compare the recovery quality of the different techniques, Table I summarizes the mean value of the seven metrics on the outdoor datasets of SOTS [24] and HazeRD [53]. Although Our method is trained unpaired, it still outperforms the competitors including supervised ones, in terms of most reference and non-reference metrics. Especially, on both synthetic datasets, the proposed method achieves the best values in terms of PSNR, SSIM and BRISQUE, which demonstrates that the superiority of our method in terms of color recovery and generation quality of defogging results.
3) Comparison on real-world foggy images

We evaluate our method on real foggy images, which are provided by the O-HAZE [3], LIVE [8] and previous methods. Fig. 10 shows the defogged results of qualitative comparison between our HAAN and the other methods on the real hazy images. As can be seen in Fig. 10(b), although DehazeNet [5] is able to remove fog from misty images, the defogged results still have some remaining fog artifacts. As shown in Fig. 10(c), most of the fog is removed in the DCPDN’s results, and some results appeared serious checkerboard effect (as shown in the first row of Fig. 10(c)). The results of EPDN [37] and GCANet [6] have no remaining fog, however, some defogged images significantly suffer from over-dehaze resulting in regional darkening (as shown in the first row of Fig. 10(d) and Fig. 10(e)). The defogged results of RefineDNet [54] and D* [48] (as shown in Fig. 10(k)-(l)) also showed low contrast. As can be seen in Fig. 10(f)-(g), there is some fog remain in the defogged results of DA-Dehaze [41] and MSBDN [9]. The defogged images generated by PSD [7] and Cycle-Dehaze [11] (such as the clothes in the third row of Fig. 10) are distorted. Similar to the defogged results from synthetic images, as shown in the first and last row of Fig. 10(j), the color distortion phenomenon still exists in the results by using IDE [21] to process the real-world foggy images. In contrast, the results of our method HAAN (as shown in Fig. 10(m)) are more natural and clearer than other methods. The reason is that our method uses unpaired real-world foggy dataset for training, and the proposed attention mechanism network can pay more attention to fog relevant features to preserve richer image content and texture details for defogged results. We performed the quantitative comparison of defogged results on the two real-world datasets of O-HAZE [3] and LIVE [8]. As shown in Table III, it is clear that HAAN outperforms state-of-the-art techniques in terms of PSNR and SSIM, which proves our method has quite competitive in single image defogging. Our method has higher PSNR than others, which indicates that our method welo restores texture details. Meanwhile, the BRISQUE scores and NIQE values of our results are lower than other methods, which means that our defogged results have higher fidelity. Moreover, our method has a smaller FADE than others, which demonstrates that our defogged results are clearer.

V. Concluding Remarks

In this work, we presented HAAN to exploit the self-similarity of multi-foggy relevant features and the complementary texture/structure information to improve the quality of defogged images based on a cyclic generative adversarial framework. We devised three basic modules (FRM, FSM, and CTRM) to constrain each other to generate high-quality images from fog/fogfree domain to fogfree/fog domain by using four mapping paths. We also explored several derived images of foggy inputs to guide our channel-spatial attention to cooperatively retrieve more foggy related contextual information to enhance details in defogged images. Finally, we used a novel sky segmentation network to optimize the atmospheric light to guide the FSM to generate more natural foggy images to improve the transformation mapping ability from foggy to fog-free images by FRM. Extensive qualitative and quantitative comparisons demonstrate that HAAN delivers superior defogging performance over the state of the art.

### REFERENCES

[1] C. O. Ancuti and C. Ancuti. Single image dehazing by multi-scale fusion. *IEEE Transactions on Image Processing*, 22(8):3271–3282, 2013.

[2] C. O. Ancuti, C. Ancuti, C. Hermans, and P. Bekaert. A fast semi-inversion approach to detect and remove the haze from a single image. In *Asian Conference on Computer Vision*, pages 501–514. Springer, 2010.

[3] C. O. Ancuti, C. Ancuti, R. Timofte, and C. De Vleeschouwer. O-haze: a dehazing benchmark with real hazy and haze-free outdoor images. *ArXiv e-prints*, 5(6), 2018.

[4] 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.

[5] 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, 2016.

[6] D. Chen, M. He, Q. Fan, J. Liao, L. Zhang, D. Hou, L. Yuan, and G. Hua. Gated context aggregation network for image dehazing and deraining. In *2019 IEEE winter conference on applications of computer vision (WACV)*, pages 1375–1383. IEEE, 2019.

[7] Z. Chen, Y. Wang, Y. Yang, and D. Liu. Psd: Principled synthetic-to-real dehazing guided by physical priors. In *Proceedings of the IEEE/ICCV conference on computer vision and pattern recognition*, pages 7180–7189, 2021.

[8] L. K. Choi, J. You, and A. C. Bovik. Referenceless prediction of perceptual fog density and perceptual image dehazing. *IEEE Transactions on Image Processing*, 24(11):3888–3901, 2015.

[9] H. Dong, J. Pan, L. Xiang, Z. Hu, X. Zhang, F. Wang, and M.-H. Yang. Multi-scale boosted dehazing network with dense feature fusion. In *Proceedings of the IEEE/ICCV conference on computer vision and pattern recognition*, pages 2157–2167, 2020.

### TABLE III

| Dataset | O-HAZE | LIVE |
|---------|--------|-------|
| Metric | PSNR [19] | SSIM [46] | LPIPS [51] | FADE [8] | BRISQUE [29] | NIQE [30] | τ [15] | PSNR [19] | SSIM [46] | LPIPS [51] | FADE [8] | BRISQUE [29] | NIQE [30] | τ [15] |
| DehazeNet [5] | 18.5655 | 0.7384 | 0.2566 | 0.3811 | 11.6304 | 2.6174 | 1.2240 | - | - | - | - | - | - | - |
| DCPDN [27] | 18.5372 | 0.7268 | 0.2754 | 0.7051 | 21.1931 | 3.4793 | 1.4379 | - | - | - | - | - | - | - |
| EPDN [27] | 19.0154 | 0.7822 | 0.1999 | 0.3905 | 18.1234 | 3.0177 | 1.7658 | - | - | - | - | - | - | - |
| GCANet [6] | 17.9407 | 0.7443 | 0.1917 | 0.2714 | 21.7128 | 2.8813 | 1.9354 | - | - | - | - | - | - | - |
| DA-Dehaze [41] | 19.5841 | 0.7279 | 0.2505 | 0.3527 | 15.9735 | 2.7854 | 2.1407 | - | - | - | - | - | - | - |
| MSBDN [9] | 15.1640 | 0.5733 | 0.2656 | 0.5195 | 18.2713 | 4.7255 | 1.5547 | - | - | - | - | - | - | - |
| PSD [7] | 13.4581 | 0.5520 | 0.2523 | 0.6392 | 14.5644 | 2.4693 | 3.2222 | - | - | - | - | - | - | - |
| RefineDNet [54] | 17.2068 | 0.7997 | 0.2446 | 0.3261 | 14.6594 | 2.3629 | 3.6548 | - | - | - | - | - | - | - |
| Cycle-Dehaze [11] | 15.3640 | 0.5733 | 0.2446 | 0.3261 | 14.6594 | 2.3629 | 3.6548 | - | - | - | - | - | - | - |
| Ridge [11] | 14.8387 | 0.4867 | 0.3463 | 0.3816 | 12.9541 | 2.7142 | 2.7009 | - | - | - | - | - | - | - |
| RefineDNet [54] | 17.2068 | 0.7997 | 0.2446 | 0.3261 | 14.6594 | 2.3629 | 3.6548 | - | - | - | - | - | - | - |

* Table III: Average PSNR, SSIM, LPIPS, FADE, BRISQUE, NIQE, τ of Defogged Results on Real-world Datasets O-HAZE and LIVE. The Top Two Performance Values Are Highlighted in Red and Blue.*
