International Journal of Analytical and Organic Chemistry
https://trpub.org/index.php/IJAOC
Vol 1, Issue 1, 2026

DEA-Net: Single Image Dehazing Based on Detail-Enhanced Convolution and Content-Guided Attention

G. Narendra Kumar
SQL DBA at Cognizant, Secunderabad, Telangana, India
E-mail: narendrasqldba@gmail.com

Abstract—Single image dehazing is a challenging ill-posed problem which estimates latent haze-free images from observed hazy images. Some existing deep learning based methods are devoted to improving the model performance via increasing the depth or width of convolution. The learning ability of convolutional neural network (CNN) structure is still under-explored. In this paper, a detail-enhanced attention block (DEAB) consisting of the detail-enhanced convolution (DEConv) and the content-guided attention (CGA) is proposed to boost the feature learning for improving the dehazing performance. Specifically, the DEConv integrates prior information into normal convolution layer to enhance the representation and generalization capacity. Then by using the re-parameterization technique, DEConv is equivalently converted into a vanilla convolution with NO extra parameters and computational cost. By assigning unique spatial importance map (SIM) to every channel, CGA can attend more useful information encoded in features. In addition, a CGA- based mixup fusion scheme is presented to effectively fuse the features and aid the gradient flow. By combining above mentioned components, we propose our detail-enhanced attention network (DEA-Net) for recovering high-quality haze-free images. Extensive experimental results demonstrate the effectiveness of our DEA-Net, outperforming the state-of-the-art (SOTA) methods by boosting the PSNR index over 41 dB with only 3.653 M parameters. The source code of our DEA-Net will be made available at https://github.com/cecret3350/DEA-Net.

Index Terms—Image dehazing, Detail-enhanced convolution, Content-guided attention, Fusion scheme.

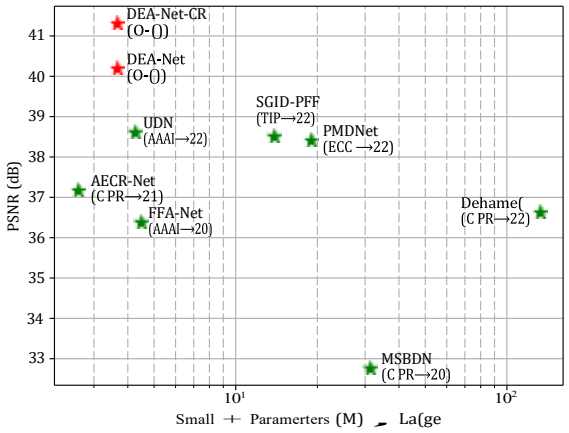


Fig. 1. Graph of PSNR vs. number of parameters. We compare our DEA- Net with some state-of-the-art methods (after 2020). The results are tested on SOTS-indoor dataset. Note that AECR-Net adopts a sharing strategy to reduce the number of parameters.

I. Introduction

IMAGES captured under hazy scenes usually suffer from noticeable visual quality degradation in contrast or color distortion [1], leading to significant performance drop when inputting to some high-level vision tasks (e.g., object detection, semantic segmentation). Haze-free images are highly demanded or required among these tasks. Therefore, single image dehazing, which aims to recover the clean scene from the corresponding hazy image, has attracted significant attention among both the academic and industrial communities over the past decade. As a fundamental low-level image restoration task, image dehazing can be the pre-processing step of the subsequent high-level vision tasks. In this paper, we attempt to develop an effective algorithm to remove the haze and recover the details from the hazy input.

Recently, with the rapid development of deep learning, convolution neural network (CNN) based dehazing methods achieve superior performance [2]–[6]. Earlier CNN-based methods [2], [7], [8] first estimate the transmission map and the atmospheric light separately, and then utilize the atmospheric scattering model (ASM) [9] to derive the haze-free images. Typically, the transmission map is supervised by the ground truth, which is used for synthesizing the training dataset. However, inaccurate estimation of the transmission map or the atmospheric light would significantly influence the image restoration results. More recently, some methods [6], [10], [11] prefer to predict the latent haze-free images in an end-to-end manner since it tends to achieve promising results.

However, there still exists two main issues:

(1) Less effectiveness of vanilla convolution. Previous works [12]–[14] prove that well-designed priors like dark channel prior [12], [15], non-local haze-line prior [13], and color attenuation prior [14], are helpful for recovering missing information. Most of existing dehazing methods [5], [6], [16] adopt classical convolution layers for feature extraction without utilizing these priors. However, vanilla convolutions search the vast solution space without any constrains, which

Z. Chen, Z. He and Z.M. Lu are with School of Aeronautics and Astronautics, Zhejiang University (e-mail: zeweihe@zju.edu.cn).

This work was funded by China Postdoctoral Science Foundation funded project (No. 2022M712792).

†The first two authors contribute equally to this work.

*Corresponding author: Zhe-Ming Lu.

This paper was produced by the IEEE Publication Technology Group. They are in Piscataway, NJ.

Manuscript received April 19, 2021; revised August 16, 2021.

to some extend may limit the expressive ability (or modeling capacity). In addition, some transformer-based methods [17] expand the receptive field to the whole image for mining long-distance dependencies. They can enhance the expressive ability (or modeling capacity) at the cost of complex training strategy and tedious hyper-parameter tuning. Also, the prohibitive computational cost and vast GPU memory occupation cannot be ignored. In this regard, the ideal solution is to embed the well-designed priors into CNN for improving the feature learning capability.

(2) Haze non-uniformity. There are two kinds of non-uniformity in the dehazing problem: uneven haze distribution in image level and channel-wise haze difference in feature level. To cope with the first one, Qin et al. [5] employed a pixel attention (i.e., spatial attention) to generate a spatial importance map (SIM), which can adaptively indicating the importance levels of different pixel locations. Through this discriminative strategy, the FFA-Net model treats thin and thick haze regions unequally. Similarly, Ye et al. [11] tried to model the density of haze distribution via a density estimation module, which essentially is also a spatial attention. However, seldom researchers paid attention to the non-uniformity in feature level, which remains to be unexploited. The channel attention used in [5] can produce a channel-wise attention vector [1] to indicate the importance level of each channel, which fails to consider the contextual information in the spatial dimensions. The haze information is encoded into the feature maps after applying convolution layers. Different channels in the feature space have different meanings depending on the role of the filters applied. In this regard, we argue that spatial importance maps should be channel-specific, and consider two kinds of non-uniformity (image level and feature level) simultaneously.

To address above mentioned issues, we design a detail-enhanced attention block (DEAB), which consists of a detail-enhanced convolution (DEConv) and a content-guided attention (CGA) mechanism. The DEConv contains five convolution layers (four difference convolutions [18] and one vanilla convolution), which are parallel deployed for feature extraction. Specifically, a central difference convolution (CDC), an angular difference convolution (ADC), a horizontal difference convolution (HDC), and a vertical difference convolution (VDC) are adopted to integrate traditional local descriptors into the convolution layer, thus can enhance the representation and generalization capacity. In difference convolutions, the pixel differences in the image are firstly calculated, and then convolved with the convolution kernel to generate the output feature maps. The strategy of pixel pair's difference calculation can be designed to explicitly encode prior information into CNN. For instance, HDC and VDC explicitly encode the gradient prior into the convolution layers via learning beneficial gradient information.

Moreover, the sophisticated attention mechanism (i.e., CGA) is a two-step attention generator, which can produce the coarse spatial attention map firstly and then refine it to the fine version. Specifically, given certain input feature maps, we utilize the spatial attention mechanism presented in [19] and the channel attention presented in [20] to generate the initial SIMs (i.e., the coarse version). Then, the initial SIMs are refined according to every channel of input feature maps to produce final SIMs. By using the content of input features to guide the generation of SIMs, CGA can focus on the unique part of features in each channel. It is worth mentioning that CGA as a universal basic block can be plug into neural networks to improve the performance in various image restoration tasks.

Besides the improvements mentioned above, we re-parameterize the learned kernel weights of the parallel convolutions to reduce the number of parameters and accelerate the training the testing process. The five parallel convolutions are simplified into one vanilla convolution layer with applying some constrains to the kernel weights and by using the linear property of convolution layers. Therefore, the proposed DEConv can extract richful features for improving dehazing performance while keeping the number of parameters and computational cost equal to the vanilla convolution. Fig. 1 shows the efficiency and effectiveness of our method.

Following [6], [10], [21], [22], we also adopt a U-net-like framework to make the major time-consuming convolution computations in the low-resolution space. Among them, the fusion of shallow and deep features is widely used. Feature fusion can enhance the information flow from shallow layers to deep ones, which is effective for feature preserving and gradient back-propagation. The information encoded in the shallow features is tremendously different from the information encoded in the deep features, since the diverse receptive fields. One single pixel in the deep features are originated from a region of pixels in the shallow features. Simple addition or concatenation operation is unable to solve the receptive field mismatch problem. We further propose a CGA-based mixup scheme to adaptively fuse the low-level features in the encoder part with corresponding high-level features, by modulating the features via learned spatial weights.

The diagram of our proposed method are shown in Fig. 2. We term the proposed single image dehazing model as DEA-Net by introducing the **D**etail-**E**nhanced **A**ttention block (DEAB) with the **D**etail-**E**nhanced convolution and the content-guided **A**ttention.

To conclude, we have following main contributions:

- We design a detail-enhanced convolution (DEConv), which contains parallel vanilla and difference convolutions. To the best of our knowledge, it's the first time that difference convolutions are introduced to solve the image dehazing problem. By encoding prior information into normal convolution layer, the representation and generalization capacity of DEConv is enhanced for improving dehazing performance. In addition, we equivalently convert the DEConv into a normal convolution with **NO** extra parameters and computational cost via using re-parameterization technique.
- We propose a novel attention mechanism called content-guided attention (CGA) to generate the channel-specific SIMs in a coarse-to-fine manner. By using input fea-

[1]The global average pooling (GAP) operation reduces the spatial dimensions to one point.

tures to guide the generation of SIMs, CGA assigns unique SIM to every channel, making the model attend significant regions of each channel. Thus, more useful information encoded in features can be emphasized to effectively improve the performance. Moreover, a CGA- based mixup fusion scheme is presented to effectively fuse the low-level features in the encoder part with corresponding high-level features.

- By combining DEConv and CGA, and using CGA-based mixup fusion scheme, we propose our detail-enhanced attention network (DEA-Net) for reconstructing high-quality haze-free images. DEA-Net shows superior performance over the state-of-the-art dehazing methods on multiple benchmark datasets, achieving more accurate results with faster inference speed.

The remainder of this paper is organized as follows. We first review a number of deep learning-based dehazing methods in Sec. II. Sec. III describes the proposed EDA-Net model in detail, and Sec. IV shows the experimental results. Finally, Sec. V concludes this paper.

II. Related Work

A. Single Image Dehazing

For single image dehazing, existing methods can be mainly divided into two categories. One is to manually generalize the statistical discrepancy between the hazy and haze-free images as empirical priors. Another one aims to directly or indirectly learning the mapping function based on large-scale datasets. We usually term the former as the prior-based methods and the latter as the data-driven methods.

The prior-based methods are the pioneers of image dehazing. They usually rely on atmospheric scattering model (ASM) [9] and handcraft priors. The widely known priors include dark channel prior (DCP) [12], [15], non-local prior (NLP) [13], color attenuation prior (CAP) [14], etc. He et al. [12], [15] proposed DCP based on a key observation - most local patches in haze-free outdoor images contain some pixels which have very low intensities in at least one color channel, which can help estimate the transmission map. CAP [14] starts from the HSV color model, and establishes a linear relationship between depth and the difference of brightness and saturation. Berman et al. [13] found that pixel clusters of haze-free images will become haze-lines when haze presents. These prior-based methods have achieved promising dehazing results. However, they tend to work well only in specific scenes which happen to satisfy their assumptions.

Recently, with the rising of deep learning, researchers focused on data-driven methods, since they can achieve better performance. Earlier data-driven methods usually perform dehazing based on the physical model. For instance, DehazeNet [2] and MSCNN [7] utilize CNNs to estimate the transmis- sion map. Then, AOD-Net [3] rewrites the ASM and esti- mates atmospheric light together with transmission map. Later, DCPDN [8] estimates the transmission map and atmospheric light by two different networks. However, the cumulative errors introduced by inaccurate estimations of transmission map and atmospheric light may cause the performance degradation.

To avoid this, more recent works tend to recover the haze-free image from the hazy image directly without the help of the physical model. GFN [23] gates and fuses three enhanced images from original hazy inputs to generate the haze-free images. GridDehazeNet [24] utilizes a three-stage attention-based grid network to recover the haze-free images. MSBDN [10] utilizes boosting strategy and back-projection technique to enhance the feature fusion. FFA-Net [5] introduces the feature attention mechanism (FAM) to dehazing network to deal with different types of information. AECR-Net [6] reuses the feature attention block (FAB) [5] and proposes a novel contrastive regularization, which can benefit from both positive samples and negative samples. UDN [22] analyzes two types of uncertainty in image dehazing, and utilizes them to increase the dehazing performance. PMDNet [11] and Dehamer [17] adopt transformer to build long-range dependencies and per- form dehazing with the guidance of haze density. However, as data-driven methods develop and dehazing performance improves, the complexity of dehazing networks also increases. Different from previous works, we rethink the deficiencies of vanilla convolution in image dehazing and design a novel convolution operator by combining well-designed priors into CNN for improving the feature learning capability. We also dig deeper into the unexploited non-uniformity of haze in feature level.

B. Difference Convolution

The origin of difference convolutions can be traced back to the local binary pattern (LBP) [25], which encodes the pixel differences in the local patch to a decimal number for texture classification. Since the success of CNNs in com- puter vision tasks, Xu et al. [26] proposed the local binary convolution (LBC) which encodes the pixel differences by using non-linear activation functions and linear convolution layers. Recently, Yu et al. [27] proposed the central difference convolution (CDC) to directly encode the pixel differences with completely learnable weights. Later, various forms of dif- ference convolutions have been proposed, such as cross central difference convolution [28] and pixel difference convolution [29]. Considering the nature of the difference convolution for capturing gradient-level information, we firstly introduce it to single image dehazing for improving the performance.

III. Methodology

As shown in Fig. 2, our DEA-Net consists of three parts: encoder part, feature transform part, and decoder part. As the core of our DEA-Net, the feature transform part adopts stacked detail-enhanced attention blocks (DEABs) to learn haze-free features. There are three levels in the hierarchical structure, and we employ different blocks in different levels to extract corresponding features (level 1&2: DEB, level 3: DEAB). Given a hazy input image $\mathbf{I} \in \mathbf{R}^{3\times H\times W}$, the goal of DEA-Net is to restore the corresponding haze-free image $\mathbf{J} \in \mathbf{R}^{3\times H\times W}$.

A. Detail-enhanced Convolution

In single image dehazing domain, previous methods [5], [6], [16] usually utilize vanilla convolution (VC) layers for

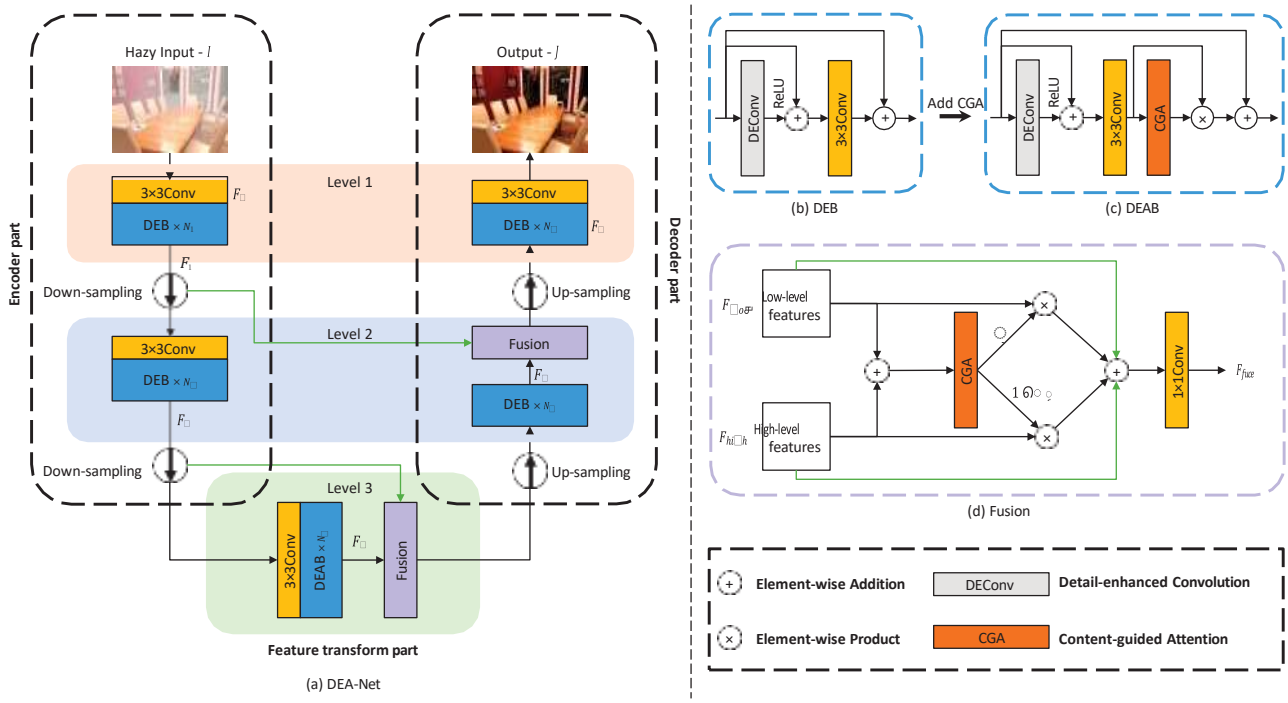


Fig. 2. The overall architecture of our proposed detail-enhanced attention network (DEA-Net), which is a three-level encoder-decoder-like architecture. DEA-Net contains three parts: encoder part, feature transform part, and decoder part. We deploy detail-enhanced attention blocks (DEABs) in the feature transform part, and deploy detail-enhanced blocks (DEBs) in rest parts.

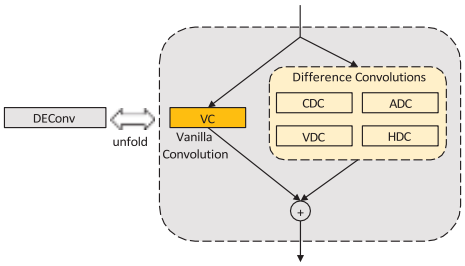


Fig. 3. Detail-enhanced convolution (DEConv). It contains five parallel deployed convolution layers including: a vanilla convolution (VC), a central difference convolution (CDC), an angular difference convolution (ADC), a horizontal difference convolution (HDC), and a vertical difference convolution (VDC).

feature extraction and learning. Normal convolution layers search the vast solution space without any constrains (even start from random initialization), restricting the expressive ability or modeling capacity. Then we notice that the high- frequency information (e.g., edges and contours) is of great significance in recovering an image captured under the hazy scene. Based on this, some researchers [8], [21], [30] adopted the edge prior in the dehazing model to help restore sharper contours. Inspired by their works [8], [30], we design a detail- enhanced convolution (DEConv) layer (see in Fig. 3), which can integrate well-designed priors into vanilla convolution layers.

Before elaborating the proposed DEConv in detail, we first recap the difference convolution (DC). Previous works [27]–[29], [31] usually describe the difference convolution as the convolution of pixel differences (the pixel differences are firstly calculated, and then convolved with the kernel weights to generate feature maps), which can enhance the representation and generalization capacity of vanilla convolution. Central difference convolution (CDC) and angular difference convolution (ADC) are two kinds of typical DCs, and implemented by re-arranging learned kernel weights to save computational cost and memory consumption [29]. It proves to be effective for edge detection [29] and face anti-spoofing tasks [27], [28], [31]. **To the best of our knowledge, it is the first time that we introduce DC to solve the single image dehazing problem.**

In our implementation, we employ five convolution layers (four DCs [18] and one vanilla convolution), which are parallel deployed for feature extraction. In DCs, the strategy of pixel pair's difference calculation can be designed to explicitly encode prior information into CNN. For our DEConv, besides the central difference convolution (CDC) and the angular difference convolution (ADC), we derive the horizontal difference convolution (HDC) and the vertical difference convolution (VDC) to integrate traditional local descriptors (like Sobel [32], Prewitt [33], or Scharr [34]) into the convolution layer. As shown in Fig. 4, taking HDC as an example, the horizontal gradient is firstly calculated by computing the differences of selected pixel pairs. After training, we re-arrange the learned kernel weights equivalently, and apply convolution directly to the untouched input features. Note that, the equivalent kernel has the similar format of traditional local descriptors (the sum of horizontal weights equals to zero). Horizontal kernels of Sobel [32], Prewitt [33], and Scharr [34] can be

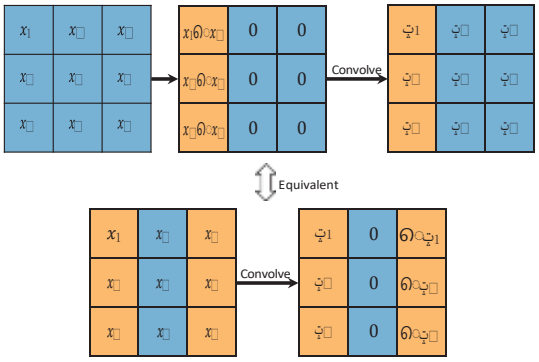


Fig. 4. The derivation of horizontal difference convolution (HDC).

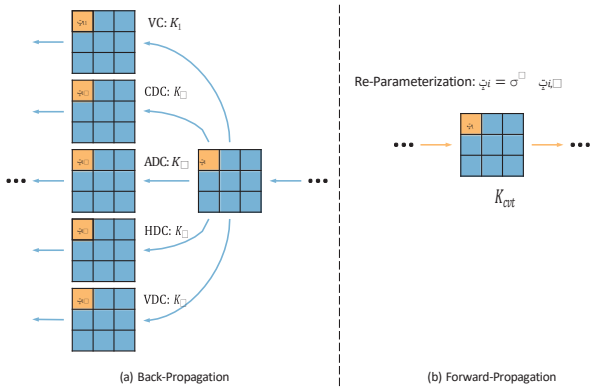


Fig. 5. The process of the re-parameterization technique.

regarded as the special case of the equivalent kernel. VDC has the similar derivation by changing the horizontal gradient to corresponding vertical counterpart. Both HDC and VDC explicitly encode the gradient prior into the convolution layers to enhance the representation and generalization capacity via learning beneficial gradient information.

In our design, the vanilla convolution serves to obtain the intensity-level information while the difference convolutions are used to enhance gradient-level information. We simply add the learned features together to obtain the output of DEConv. We trust more sophisticated designs of the way for calculating the pixel difference can further benefit image restoration task, which is not the main direction of this paper.

However, deploying five parallel convolution layers for feature extraction will undesirably cause the increase of parameters and inference time. We seek to exploit the additivity of convolution layers for simplifying the parallel deployed convolutions into a single standard convolution. We notice a useful property of the convolution: if several 2D kernels with the identical size operate on the same input with the same stride and padding to produce outputs, and their outputs are summed up to obtain the final output, we can add up these kernels on the corresponding positions to obtain an equivalent kernel which will produce the identical final output. Surprisingly, our DEConv exactly fits this situation. Given the input features F_{in}, DEConv can output F_{out} with identical computational cost and inference time to a vanilla convolution layer by utilizing re-parameterization technique. The formula is as follows (the biases are omitted for simplification):

$$F_{out} = \text{DEConv}(F_{in}) = \sum_{i=1}^{5} F_{in} * K_i \quad (1)$$

$$= F_{in} * (\sum_{i=1}^{5} K_i) = F_{in} * K_{cvt},$$

phase, the kernel weights of the parallel convolutions are fixed and the converted kernel weights are calculated by adding up them on the corresponding positions. Note that, the re-parameterization technique can accelerate the training and testing process simultaneously, since both of them contain the forward-propagation phase.

Compare with the vanilla convolution layer, the proposed DEConv can extract more richful features while maintains the parameter size, and introduces no extra computational cost and memory burdens in the inference stage. More discussions about DEConv can be found in Sec. IV-C1.

B. Content-guided Attention

Feature attention module (FAM) consists of a channel attention and a spatial attention, which are sequentially placed to calculate the attention weights in channel and spatial dimensions. The channel attention calculates a channel-wise vector, i.e., $W_c \in R^{C\times1\times1}$, to re-calibrate the features. The spatial attention calculates a spatial importance map (SIM), i.e., $W_s \in R^{H\times W}$ to adaptively indicate the importance levels of different regions. The FAM treats different channels and pixels unequally, improving the dehazing performance.

However, the spatial attention inside FAM can only address the uneven haze distribution in image level, and ignore the uneven distribution in feature level. The channel attention inside FAM models the channel-wise differences without considering the contextual information. With the expanding of feature channels, the image-level haze distribution information is encoded into the feature maps. Different channels in the feature space have different meanings depending on the role of the filters applied. That means for every channel of features, the haze information is unevenly spread across the spatial dimensions. The channel-specific SIMs are desired in t

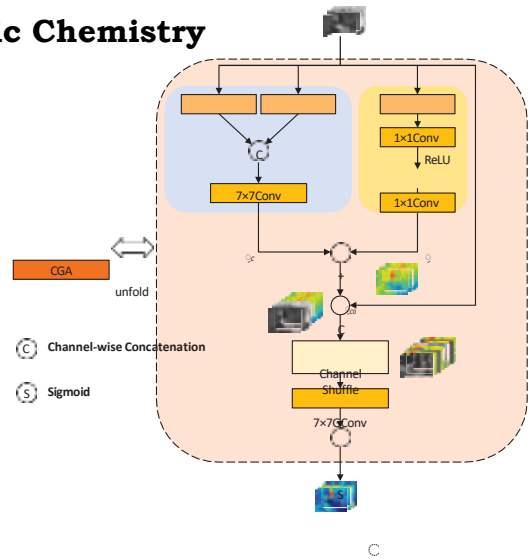


Fig. 6. The diagram of content-guided attention (CGA). CGA is a coarse-to-fine process: the coarse version of SIMs (i.e., $W_{coa} \in R^{C\times H\times W}$) is generated firstly and then every channel is refined by the guidance of input features.

The detailed procedures of CGA are illustrated in Fig. 6, let $X \in R^{C\times H\times W}$ denotes the proceeding input features, the goal of CGA is to generate channel-specific SIMs (i.e., $W \in R^{C\times H\times W}$), which has the identical dimensions with X.

We first compute the corresponding W_c and W_s by following [19], [20].

$$W_c = C_{1\times1}(\max(0, C_{1\times1}(X^c_{GAP}))), \quad (2)$$

$$W_s = C_{7\times7}([X^s_{GAP}, X^s_{GMP}]),$$

where $\max(0, x)$ denotes the ReLU activation function, $C_{k\times k}(\cdot)$ denotes a convolution layer with $k \times k$ kernel size, $[\cdot]$ denotes the channel-wise concatenation operation. X^c_{GAP}, X^s_{GAP}, and X^{GMP}_s denote the features processed by global average pooling operation across the spatial dimensions, global average pooling operation across the channel dimension, and global max pooling operation across the channel dimension, respectively. To reduce the number of parameters and limit the model complexity, the first 1×1 convolution reduces the channel dimension from C to $\frac{C}{r}$ (r refers to the reduction ratio), and the second 1×1 convolution expands it back to C.

In our implementation, we opt to reduce the channel dimension to a fixed value (i.e., 16) by setting r to $\frac{C}{16}$.

Then we fuse W_c and W_s together via a simple addition operation, which follows broadcasting rules, to obtain the coarse SIMs $W_{coa} \in R^{C\times H\times W}$. We experimentally find the product operation can achieve similar results.

$$W_{coa} = W_c + W_s, \quad (3)$$

In order to obtain the final refined SIMs W, every channel

The CGA assigns unique SIM to every channel, guiding the model to focus on significant regions of each channel. Therefore, more useful information encoded in features can be emphasized to effectively improve the dehazing performance. As shown in the right part of Fig. 2, combining proposed DEConv with the CGA, we propose the main block of our DEA-Net, i.e., detail-enhanced attention block (DEAB). By removing the CGA part, we obtain the detail enhanced block (DEB).

C. CGA-based Mixup Fusion Scheme

Following [6], [10], [21], [22], we adopt the encoder-decoder-like (or U-Net-like) architecture for our DEA-Net. We observe that fusing the features from the encoder part with that from the decoder part is an effective trick in dehazing and other low-level vision tasks [6], [10], [36], [37]. Low- level features (e.g., edges and contours), which have a non-negligible role for recovering haze-free images, gradually lose their impact after passing through many intermediate layers. Feature fusion can enhance the information flow from shallow layers to deep ones, which is beneficial for feature preserving and gradient back-propagation. The simplest way for fusion is element-wise addition, which is adopted in many previous approaches [10], [11], [21]. Later, Wu et al. [6] applied the adaptive mixup operation to adjust the fusion proportion via self-learned weights, which is more flexible than the addition.

However, there exists a receptive field mismatch problem in above mentioned fusion schemes. The information encoded in the shallow features is tremendously different from the information encoded in the deep features, since they have the totally different receptive fields. One single pixel in the deep features are originated from a region of pixels in the shallow features. Simple addition or concatenation operation or mixup operation fails to address the mismatch before fusion.

To mitigate this problem, we further propose a CGA-based mixup scheme to adaptively fuse the low-level features in the encoder part with corresponding high-level features, by modulating the features via learned spatial weights.

Fig. 2 (d) shows the details of proposed CGA-based mixup fusion scheme. The core part is that we opt to employ the CGA to calculate the spatial weights for feature modulation. The low-level features in the encoder part and corresponding high- level features are fed into the CGA to calculate the weights, and then combined by a weighted summation method. We also add the input features via skip connections to mitigate gradient

D. Overall Architecture

By combining (1) DEConv, (2) CGA, and (3) CGA-based mixup fusion scheme together, we propose our DEA-Net with DEAB and DEB as the basic blocks. As shown in Figure 2, our DEA-Net is a three-level encoder-decoder-like (or U-Net-like) architecture, which consists of three parts: encoder part, feature transform part, and decoder part. There are two down-sampling operations and two up-sampling operations in our DEA-Net. The down-sampling operation halves the spatial dimensions and doubles the number of channels. It is realized through a normal convolution layer by setting the value of stride to 2 and setting the number of output channels to 2 times of input channels. The up-sampling operation can be regarded as the inverse form of the down-sampling operation, which is realized through a deconvolution layer. The dimensional size of level 1, level 2, and level 3 are $C \times H \times W$, $2C \times \frac{H}{2} \times \frac{W}{2}$, and $4C \times \frac{H}{4} \times \frac{W}{4}$, respectively. In our implementation, we set the value of C to 32. Previous methods [6], [22] transform the features only in the low-resolution space, resulting in information loss, which is non-trivial for the detail-sensitive task like dehazing. Differently, we deploy feature extraction blocks from level 1 to level 3. Specifically, we opt to employ different blocks in different levels (level 1&2: DEB, level 3: DEAB). For feature fusion, we fuse the features after the down-sampling operations and corresponding features before the up-sampling operations (highlighted with green arrow lines in Fig. 2). Finally, we simply employ a 3×3 convolution layer at the end to obtain the dehazing result J.

The DEA-Net is trained by minimizing the pixel-wise difference between the predicted haze-free image J and the corresponding ground truth GT. In our implementation, we choose $L1$ loss function (i.e., mean absolute error) to drive the training.

$$L_{L1} = ||J - GT||_1, \quad (6)$$

IV. Experiment

A. Datasets and Metrics

Datasets. In our implementation, we train and test our proposed DEA-Net on synthetic and real-captured datasets. REalistic Single Image DEhazing (RESIDE) [38] is a widely-used dataset, which contains five subsets: Indoor Training Set (ITS), Outdoor Training Set (OTS), Synthetic Objective Testing Set (SOTS), Real-world Task-driven Testing Set (RTTS), and Hybrid Subjective Testing Set (HSTS). We select ITS and OTS in the training phase and select SOTS in the testing phase. Note that, the SOTS is divided into two subsets (i.e., SOTS-indoor and SOTS-outdoor) for evaluating the models separately trained on ITS and OTS. ITS contains 1399 indoor clean images and for every clean image, 10 simulated hazy images are generated based on the physical scattering model. As for OTS, we pick around 296K images for the training process [2]. SOTS-indoor and SOTS-outdoor contain 500 indoor and 500 outdoor testing images, respectively. In addition, Haze4K dataset [39], which contains 3000 synthetic training images and 1000 synthetic testing images, is also employed to evaluate our DEA-Net. Besides, some real-captured hazy images are utilized to further verify the effectiveness on real scenes.

Evaluation Metrics. Peak signal-to-noise-ratio (PSNR) and structural similarity index (SSIM) [40], which are commonly used to measure the image quality among the computer vision community, are utilized for dehazing performance evaluation. For a fair comparison, we calculate the metrics based on the RGB color images without cropping pixels.

B. Implementation Details

We implement the proposed DEA-Net model on PyTorch deep learning platform with a single NVIDIA RTX3080Ti GPU. We deploy DEB, DEB, and DEAB in level 1, level 2, and level 3, respectively. The number of blocks deployed on different stages $[N_1, N_2, N_3, N_4, N_5]$ is set to $[4, 4, 8, 4, 4]$. The DEA-Net is optimized using Adam [41] optimizer and β_1, β_2, ε are set to default values, i.e., 0.9, 0.999, $1e^{-8}$. Moreover, the initial learning rate and the batch size are set to $1e^{-4}$ and 16, respectively. Cosine annealing strategy [42] is adopted to adjust the learning rate from the initial value to $1e^{-6}$. To train the model, we randomly crop patches from the original images with size 256×256, then two data augmentation techniques are adopted including: 90° or 180° or 270° rotation and vertical or horizontal flip. In the whole training phase, the model is trained for 1500K iterations, and it takes roughly 5 days to train our DEA-Net on ITS.

C. Ablation Study

To demonstrate the effectiveness of our proposed DEA- Net, we investigate on the designs and effects of (1) Detail-enhanced convolution (DEConv), (2) Content-guided attention (CGA), and (3) CGA-based mixup fusion scheme. The contribution of each components are analyzed via ablation experiments.

1) DEConv: We first construct the baseline model by deploying classical residual block (RB) [43] in level 3, and this model is denoted as Base_RB. As a popular basic block used in dehazing domain, we also employ the feature attention block (FAB) from [5] in level 3. The hyper-parameters are set to the default values as described in the original paper. We term this model as our second baseline, Base_FAB.

To extract more effective features, we revise the block by introducing DEConv into RB and FAB. As illustrated in Fig. 7, the first vanilla convolution layer is replaced with the proposed DEConv in both RB and FAB. The blocks deployed in level 3 are indicated as $RB_{w/\ DEConv}$ and $FAB_{w/\ DEConv}$, respectively. The corresponding models are denoted as Model_RB_D and Model_FAB_D.

For a fair comparison, all of the four blocks (i.e., RB, FAB, $RB_{w/\ DEConv}$, and $FAB_{w/\ DEConv}$) are cascaded for 6 times in level 3, and the same fusion scheme is used (i.e., Mixup [5]). For convenience, we omit the blocks in level 1 and level 2, and train the models for only 500K iterations with initial learning rate is set to $2e^{-4}$ (These settings are with the ablation study). The experimental results are tested on the same testing dataset

[2]Following [24], data cleaning is applied since the intersection of training and testing datasets.

TABLE I
ABLATION STUDY OF DECONV AND CGA. ALL THE EXPERIMENTS ARE CONDUCTED ON SOTS-INDOOR [38] DATASET.

Model		Base_RB	Base_FAB	Model_RB_D	Model_FAB_D	Model_DEAB	Model_FAB_D_CBAM
Setting	Level 1	–	–	–	–	–	–
	Level 2	–	–	–	–	–	–
	Level 3	RB	FAB	$RB_{w/\ DEConv}$	$FAB_{w/\ DEConv}$	$FAB_{w/\ DEConv\ \&\ CGA}$ (DEAB)	$FAB_{w/\ DEConv\ \&\ CBAM}$
	Attention	–	FAM	–	FAM	CGA	CBAM [19]
PSNR (dB)		30.74	33.07	31.01	33.67	35.17	34.68
SSIM		0.9729	0.9824	0.9739	0.9840	0.9866	0.9857
# Param. (K)		2105	2143	4467	4505	4569	4493

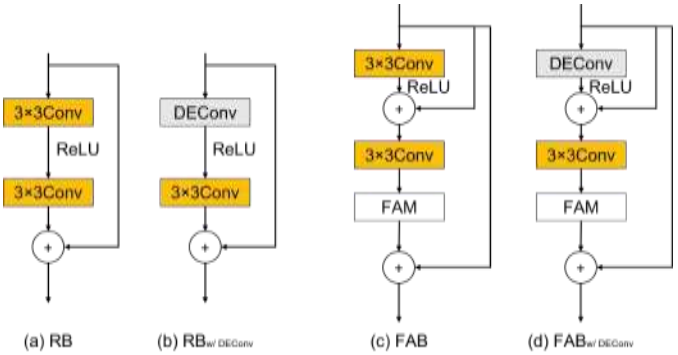


Fig. 7. The schematic diagrams of RB, $RB_{w/\ DEConv}$, FAB, and $FAB_{w/\ DEConv}$. The first vanilla convolution layer in RB/FAB is replaced with the proposed DEConv to generate the $RB_{w/\ DEConv}$/$FAB_{w/\ DEConv}$.

(i.e., SOTS-Indoor [38] dataset). Although metrics are lower than the completely trained models reported in Table. V, the trends and values are consistent and meaningful.

The performance of all these aforementioned models is summarized in Table. I. Replacing the vanilla convolution layer with the parallel convolution layers (i.e., DEConv) brings 0.27 and 0.6 dB improvement in terms of PSNR on RB and FAB, respectively. By comparing Model_FAB_D with Base_FAB, the results indicate that DEConv can definitely improve the values of metrics (i.e., PSNR and SSIM) at the cost of around twice number of parameters (4505 K vs. 2143 K). That is very unfriendly and may cause failure under some memory-limited situations, prohibiting the usage of the DEConv on mobile or embedded devices.

In order to deal with the problem, we equivalently transform the DEConv into a standard 3×3 convolution by adding up the learned kernel weights in the same positions (i.e., re-parameterization). Table. II shows the comparative results of the number of parameters (# Param.), the number of floating-point operations (# FLOPs) and inference time of Model_FAB_D before and after the re-parameterization opera- tion. We can clearly see that the re-parameterization operation simplifies the parallel structure without triggering performance drop. In particular, after the simplification, Model_FAB_D still achieves 0.6 dB performance improvement when comparing with Base_FAB and no extra overhead is introduced.

In addition, we also explore the designs of parallel convolution layers from only a single vanilla convolution layer (i.e., FAB) to two parallel vanilla convolution layers, then to the complete DEConv (i.e., $FAB_{w/\ DEConv}$). As shown in Table. III, adding a parallel vanilla convolution layer to the FAB causes a 0.15 dB performance drop. The underlying reason behind this may be the training difficulty due to redundant features extracted by the identical layers. On the contrary, adding a parallel CDC layer to the FAB boosts the performance. The experimental results verify that by embedding traditional prior information, difference convolution (DC) layers can effectively extract more representative features. We also observe that by adding more parallel DC streams for feature extraction, the performance gradually improves from 33.07 dB to 33.67 dB in terms of PSNR. Similar trend can be observed in terms of SSIM. Based on the discussions above, we choose Model_FAB_D with basic block $FAB_{w/\ DEConv}$ for the following study.

TABLE II
THE COMPARATIVE RESULTS OF THE NUMBER OF PARAMETERS (# PARAMETERS), THE NUMBER OF FLOATING-POINT OPERATIONS (# FLOPS) AND INFERENCE TIME OF Model_FAB_D BEFORE AND AFTER THE RE-PARAMETERIZATION OPERATION. RE-PA. IS SHORT FOR THE RE-PARAMETERIZATION OPERATION.

	Model_FAB_D		Base_FAB
	w/o Re-Pa.	w/ Re-Pa.	–
# Param. (K)	4505	2143	2143
# FLOPs (G)	23.72	9.23	9.23
inference time (ms)	4.53	1.76	1.76
PSNR (dB)	33.67	33.67	33.07

TABLE III
THE EXPERIMENTAL RESULTS ON DESIGNS OF PARALLEL CONVOLUTION LAYERS. ✓✓MEANS THE SAME CONVOLUTION LAYER IS USED TWICE WITHIN TWO PARALLEL STREAMS. THE METRICS ARE TESTED ON SOTS-INDOOR [38] DATASET.

Design	FAB	+ vanilla	+ DC	+ DC	$FAB_{w/\ DEConv}$
Vanilla Conv.	✓	✓✓	✓	✓	✓
+ CDC			✓	✓	✓
+ HDC & VDC				✓	✓
+ ADC					✓
PSNR	33.07	32.92	33.23	33.43	**33.67**
SSIM	0.9824	0.9820	0.9826	0.9833	**0.9840**

2) CGA: Further, we investigate the effectiveness of the proposed two-step coarse-to-fine attention mechanism (i.e., CGA). As mentioned in Section I, CGA generates channel-specific spatial importance maps (SIMs) to indicate the im-

portant regions of individual channel. We compare the CGA with the other attention mechanisms such as feature attention module (FAM) used in many dehazing methods [5], [6], [16] and the common convolutional block attention module (CBAM) [19]. Both FAM and CBAM contain sequential channel attention and spatial attention with slightly different implementations.

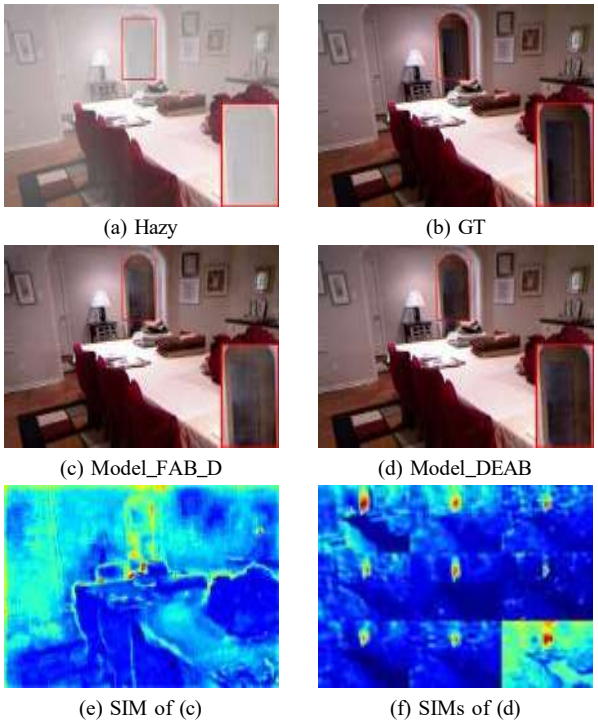
Model_FAB_D cascades $FAB_{w/\ DEConv}$ blocks in level 3 and inside the $FAB_{w/\ DEConv}$ block, the FAM is adopted. Then, we combine CGA and CBAM into $FAB_{w/\ DEConv}$ block to gener- ate the $FAB_{w/\ DEConv\ \&\ CGA}$ (i.e., DEAB) and $FAB_{w/\ DEConv\ \&\ CBAM}$, respectively, and the corresponding models are denoted as Model_DEAB and Model_FAB_D_CBAM.

The spatial attention used in FAM or CBAM learns the SIM with only one single channel to indicate the important regions of the input features with relatively larger number of channels. Such approaches neglect the specificity of each channel of features and somehow restrict the powerful representation ability of CNNs. As shown in the right three columns of Table. I, Model_DEAB outperforms both Model_FAB_D and Model _FAB_D_CBAM by 1.5 dB and 1.01 dB in terms of PSNR. The results indicate that CGA can better re-calibrate the features via learning channel-specific SIMs to pay attention to channel-wise haze distribution difference.

Fig. 8 visually illustrates the SIMs learned by CGA and FAM and the corresponding processing results. As we can see from Fig. 8e, one-channel SIM obtained by FAM can indicate the uneven haze distribution (to some extend). However, it is not accurate enough (e.g., the red chairs region) due to the mix of some contour patterns. By using the content of input features to guide the generation of SIMs, CGA can learn more accurate spatial weights. Fig. 8f shows eight randomly selected channels of SIMs, and the average map of all SIMs (right bottom). The channel-specific SIMs treat different channels of features with different spatial weights, which can better guide the model to focus on critical regions. Fig. 8c and Fig. 8d are the corresponding results. We observe that the arched door region (highlighted by the red rectangle) recovered by Model_FAB_D has obvious haze residual.

3) CGA-based Mixup Fusion Scheme: We further perform ablation study to verify the effectiveness of proposed CGA-based mixup fusion scheme. We utilize Model_DEAB with mixup fusion scheme from AECR-Net [6] as the baseline, and then evaluate another two schemes: element-wise addi- tion [10], [11], [21] and proposed CGA-based mixup. Their models are referred to Model_DEAB_A and Model_DEAB_C. The comparative results are shown in Table. IV. From these results, we see that addition achieves very similar performance with mixup (0.06 dB higher PSNR and 0.002 lower SSIM). Addition is a special case of mixup with the constant weights, and we experimentally find that the initial values have a significant impact on the performance of mixup. Note that, our proposed CGA-based mixup fusion scheme achieves the best performance in terms of PSNR and SSIM.

In addition, we deploy feature extraction blocks in level 1 and 2 to further improve the performance. By deploying residual block (RB) in level 1 and level 2 (we refer this model to Model_MS), the performance improves by a large margin



(a) Hazy (b) GT (c) Model_FAB_D (d) Model_DEAB (e) SIM of (c) (f) SIMs of (d)

Fig. 8. Visual comparisons of FAM and our proposed CGA. We show the learned SIMs and corresponding results.

(2.52 dB in terms of PSNR). It means transforming features in high-resolution space even full-resolution space can repair the lost information, which is critical for image regression. Our final DEA-Net-S achieves 39.16 dB in terms of PSNR and 0.9921 in terms of SSIM by deploying DEB in level 1 and 2. The suffix '-S' denotes the model is trained with the settings in ablation study, which is a simplified version. For Model_MS and DEA-Net-S, $[N_1, N_2, N_3, N_4, N_5]$ is set to [3, 3, 6, 3, 3]. It is worth mentioning that we omit the CGA in level 1 and level 2 (simplify DEAB into DEB) by taking the model complexity into account and to avoid complex hyper-parameter tuning (e.g., the reduction ratio).

D. Comparisons with SOTA Methods

In this section, we compare our proposed DEA-Net with 4 earlier dehazing approaches including DCP [12], DehazeNet [2], AOD-Net [3], GFN [23] and 8 recent state-of-the-art (SOTA) single image dehazing methods including FFA-Net [5], MSBDN [10], DMT-Net [39], AECR-Net [6], SGID-PFF [21], UDN [22], PMDNet [11], Dehamer [17] on SOTS-Indoor, SOTS-Ourdoor, Haze4K datasets. We report three DEA-Net variants including DEA-Net-S with the settings in ablation study (i.e., the final model of Table. IV), DEA-Net with normal settings, and DEA-Net-CR with normal settings and the contrastive regularization (CR) from AECR-Net [6]. DEA-Net-CR has identical setting of CR with AECR-Net [6]. Note that CR will not increase additional parameters and inference time, since it can be directly removed in the testing phase. For others, we adopt the official released codes or evaluation results of these methods for fair comparisons if

TABLE IV
ABLATION STUDY OF CGA-BASED MIXUP FUSION SCHEME. WE COMPARE IT WITH ELEMENT-WISE ADDITION AND MIXUP [6]. ALL THE EXPERIMENTS ARE CONDUCTED ON SOTS-INDOOR [38] DATASET.

Model		Model_DEAB_A	Model_DEAB	Model_DEAB_C	Model_MS	DEA-Net-S
Setting	Level 1	–	–	–	RB	DEB
	Level 2	–	–	–	RB	DEB
	Level 3	DEAB	DEAB	DEAB	DEAB	DEAB
	Fusion scheme	Addition	Mixup	CGA-based Mixup	CGA-based Mixup	CGA-based Mixup
PSNR (dB)		35.23	35.17	35.40	37.92	39.16
SSIM		0.9864	0.9866	0.9875	0.9915	0.9921

(a) Hazy (b) DCP [12] (c) GDN [24] (d) FFA [5] (e) AECR-Net [6] (f) Ours (g) GT

Fig. 9. Visual comparisons of various methods on synthetic SOTS-indoor [38] dataset. Please zoom in on screen for a better view.

(a) Hazy (b) DCP [12] (c) GDN [24] (d) FFA [5] (e) Dehamer [17] (f) Ours (g) GT

Fig. 10. Visual comparisons of various methods on synthetic SOTS-outdoor [38] dataset. Please zoom in on screen for a better view.

(a) Hazy (b) DCP [12] (c) GDN [24] (d) FFA [5] (e) AECR-Net [6] (f) Dehamer [17] (g) Ours

Fig. 11. Dehazing results of various methods on real-world hazy images. Please zoom in on screen for a better view.

TABLE V
QUANTITATIVE COMPARISONS OF VARIOUS DEHAZING METHODS ON SOTS-INDOOR, SOTS-OURDOOR, AND HAZE4K. WE REPORT PSNR, SSIM, NUMBER OF PARAMETERS (# PARAM.), NUMBER OF FLOATING-POINT OPERATIONS (# FLOPS), AND RUNTIME TO PERFORM COMPREHENSIVE COMPARISONS. THE SIGN "-" DENOTES THE DIGIT IS UNAVAILABLE. **BOLD** AND UNDERLINED INDICATE THE BEST AND THE SECOND BEST RESULTS, RESPECTIVELY.

Method	SOTS-indoor [38]		SOTS-outdoor [38]		Haze4K [39]		Overhead		
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	# Param. (M)	# FLOPs (G)	Runtime (ms)
(TPAMI'10) DCP [12]	16.61	0.8546	19.14	0.8605	14.01	0.76	-	-	-
(TIP'16) DehazeNet [2]	19.82	0.8209	27.75	0.9269	19.12	0.84	0.008	0.5409	0.9932
(ICCV'17) AOD-Net [3]	20.51	0.8162	24.14	0.9198	17.15	0.83	0.0018	0.1146	0.3159
(CVPR'18) GFN [23]	22.30	0.8800	21.55	0.8444	-	-	0.4990	14.94	-
(AAAI'20) FFA-Net [5]	36.39	0.9886	33.57	0.9840	26.97	0.95	4.456	287.5	47.98
(CVPR'20) MSBDN [10]	32.77	0.9812	34.81	0.9857	22.99	0.85	31.35	**24.44**	9.826
(ACMMM'21) DMT-Net [39]	-	-	-	-	28.53	0.96	51.79	75.56	26.83
(CVPR'21) AECR-Net [6]	37.17	0.9901	-	-	-	-	**2.611**	52.20	-
(TIP'22) SGID-PFF [21]	38.52	0.9913	30.20	0.9754	-	-	13.87	152.8	20.92
(AAAI'22) UDN [22]	38.62	0.9909	34.92	0.9871	-	-	4.250	-	-
(ECCV'22) PMDNet [11]	38.41	0.9900	34.74	0.9850	33.49	0.98	18.90	-	-
(CVPR'22) Dehamer [17]	36.63	0.9881	35.18	0.9860	-	-	132.4	48.93	14.12
(Ours) DEA-Net-S	39.16	0.9921	-	-	-	-	2.844	24.88	**5.632**
(Ours) DEA-Net	40.20	0.9934	36.03	0.9891	33.19	**0.99**	3.653	32.23	7.093
(Ours) DEA-Net-CR	**41.31**	**0.9945**	**36.59**	**0.9897**	**34.25**	**0.99**	3.653	32.23	7.093

they are publicly available, otherwise we retrain them using the same training datasets.

Quantitative Analysis. Table. V shows quantitative evaluation results (PSNR and SSIM indexes) of our DEA-Nets and other state-of-the-art methods on SOTS [38] and Haze4K [39]. As we can see, even our DEA-Net-S achieves the best performance with 39.16 dB PSNR and 0.9921 SSIM on SOTS-indoor than the alternatives. Further, our DEA-Net and DEA-Net-CR improve the performance by a large margin on both SOTS-indoor and SOTS-outdoor. On Haze4K dataset, our DEA-Net and DEA-Net-CR achieve the best SSIM (0.9869 and 0.9885). We round the results to two decimals to keep consistent with [39]. Our DEA-Net-CR ranks first in all comparisons on SOTS and Haze4K.

In addition, we adopt number of parameters (# Param.), number of floating-point operations (# FLOPs), and runtime as the major indicators of computational efficiency. The earlier dehazing methods contain very small parameters sizes at the cost of a big performance drop. Compared with recent SOTA methods, our DEA-Nets run fastest with acceptable # Param. and # FLOPs. Any one of our DEA-Net variants can rank sec- ond best in terms of # Param. and # FLOPs. This implies our DEA-Nets can reach a good trade-off between performance
and model complexity. Note that # FLOPs and runtime are measured on color images with 256×256 resolution.

Qualitative Analysis. Fig. 9 shows the visual compar- isons between our DEA-Net and previous SOTA methods on synthetic SOTS-indoor dataset. Our proposed DEA-Net can recover sharper and clearer contours or edges, and the results obtained by DEA-Net contains less haze residuals. Fig. 10 shows the visual comparisons on synthetic SOTS- ourdoor dataset. We observe that in outdoor scenes, the results of our DEA-Net are closest to the ground truth than the other alternatives. We also test our DEA-Net on real-world hazy images, and compare the results with various SOTA methods. As shown, the other methods either remain haze on the processed results or produce color deviations and artifacts. On the contrary, our DEA-Net can output more visually pleasing dehazing results.

V. CONCLUSION

In this paper, we propose a DEA-Net to deal with the challenging single image dehazing problem. Specifically, we design the detail-enhanced convolution (DEConv) by introducing the difference convolution to integrate local descriptors into normal convolution layer. Compare with vanilla convo- lution, DEConv has enhanced representation and generaliza- tion capacity. In addition, the DEConv can be equivalently converted into a vanilla convolution without triggering extra parameters and computational cost. Then, we design a sophisticated attention mechanism termed content-guided attention (CGA), which assigns unique spatial importance map (SIM) to every channel. With CGA, more useful information encoded in features can be emphasized. Based on CGA, we further present a fusion scheme to effectively fuse low-level features in the encoder part with corresponding high-level features. Extensive experiments show that our DEA-Net achieves state-of-the-art results quantitatively and qualitatively.

REFERENCES

[1] R. T. Tan, "Visibility in bad weather from a single image," in CVPR. IEEE, 2008, pp. 1–8.
[2] 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, vol. 25, no. 11, pp. 5187–5198, 2016.
[3] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng, "Aod-net: All-in-one dehazing network," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 4770–4778.
[4] P. Li, J. Tian, Y. Tang, G. Wang, and C. Wu, "Deep retinex network for single image dehazing," IEEE Transactions on Image Processing, vol. 30, pp. 1100–1115, 2020.
[5] X. Qin, Z. Wang, Y. Bai, X. Xie, and H. Jia, "Ffa-net: Feature fusion attention network for single image dehazing," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 07, 2020, pp. 11 908–11 915.

[6] H. Wu, Y. Qu, S. Lin, J. Zhou, R. Qiao, Z. Zhang, Y. Xie, and L. Ma, "Contrastive learning for compact single image dehazing," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 10 551–10 560.
[7] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang, "Single image dehazing via multi-scale convolutional neural networks," in European conference on computer vision. Springer, 2016, pp. 154– 169.
[8] H. Zhang and V. M. Patel, "Densely connected pyramid dehazing network," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 3194–3203.
[9] S. Narasimhan and S. Nayar, "Contrast restoration of weather degraded images," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 6, pp. 713–724, 2003.
[10] 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/CVF conference on computer vision and pattern recognition, 2020, pp. 2157–2167.
[11] T. Ye, M. Jiang, Y. Zhang, L. Chen, E. Chen, P. Chen, and Z. Lu, "Perceiving and modeling density is all you need for image dehazing," arXiv preprint arXiv:2111.09733, 2021.
[12] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," IEEE transactions on pattern analysis and machine intelligence, vol. 33, no. 12, pp. 2341–2353, 2010.
[13] D. Berman, S. Avidan et al., "Non-local image dehazing," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1674–1682.
[14] Q. Zhu, J. Mai, and L. Shao, "A fast single image haze removal algorithm using color attenuation prior," IEEE transactions on image processing, vol. 24, no. 11, pp. 3522–3533, 2015.
[15] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," 2009, pp. 1956–1963.
[16] H. Wu, J. Liu, Y. Xie, Y. Qu, and L. Ma, "Knowledge Transfer Dehazing Network for NonHomogeneous Dehazing," in CVPR Workshop. IEEE, 2020, pp. 1975–1983.
[17] C.-L. Guo, Q. Yan, S. Anwar, R. Cong, W. Ren, and C. Li, "Image dehazing transformer with transmission-aware 3d position embedding," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5812–5820.
[18] Z. Yu, C. Zhao, Z. Wang, Y. Qin, Z. Su, X. Li, F. Zhou, and G. Zhao, "Searching central difference convolutional networks for face anti-spoofing," in CVPR, 2020, pp. 5294–5304.
[19] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 3–19.
[20] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in CVPR, 2018, pp. 7132–7141.
[21] H. Bai, J. Pan, X. Xiang, and J. Tang, "Self-guided image dehazing using progressive feature fusion," IEEE Transactions on Image Processing, vol. 31, pp. 1217–1229, 2022.
[22] M. Hong, J. Liu, C. Li, and Y. Qu, "Uncertainty-driven dehazing network," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 1, pp. 906–913, Jun. 2022. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/19973
[23] W. Ren, L. Ma, J. Zhang, J. Pan, X. Cao, W. Liu, and M.-H. Yang, "Gated fusion network for single image dehazing," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 3253–3261.
[24] X. Liu, Y. Ma, Z. Shi, and J. Chen, "Griddehazenet: Attention-based multi-scale network for image dehazing," in Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 7314–7323.
[25] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," IEEE Transactions on pattern analysis and machine intelligence, vol. 24, no. 7, pp. 971–987, 2002.
[26] F. Juefei-Xu, V. Naresh Boddeti, and M. Savvides, "Local binary convolutional neural networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 19–28.
[27] Z. Yu, C. Zhao, Z. Wang, Y. Qin, Z. Su, X. Li, F. Zhou, and G. Zhao, "Searching central difference convolutional networks for face anti-spoofing," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5295–5305.
[28] Z. Yu, Y. Qin, H. Zhao, X. Li, and G. Zhao, "Dual-cross central difference network for face anti-spoofing," in Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, Z.-H. Zhou, Ed. International Joint Conferences on Artificial Intelligence Organization, 8 2021, pp. 1281–1287, main Track. [Online]. Available: https://doi.org/10.24963/ijcai.2021/177
[29] Z. Su, W. Liu, Z. Yu, D. Hu, Q. Liao, Q. Tian, M. Pietika¨inen, and L. Liu, "Pixel difference networks for efficient edge detection," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 5117–5127.
[30] C. Wang, H.-Z. Shen, F. Fan, M.-W. Shao, C.-S. Yang, J.-C. Luo, and L.-J. Deng, "Eaa-net: A novel edge assisted attention network for single image dehazing," Knowledge-Based Systems, vol. 228, p. 107279, 2021.
[31] Z. Yu, J. Wan, Y. Qin, X. Li, S. Z. Li, and G. Zhao, "Nas-fas: Static-dynamic central difference network search for face anti-spoofing," IEEE transactions on pattern analysis and machine intelligence, vol. 43, no. 9, pp. 3005–3023, 2020.
[32] I. Sobel, G. Feldman et al., "A 3x3 isotropic gradient operator for image processing," a talk at the Stanford Artificial Project in, pp. 271–272, 1968.
[33] J. M. Prewitt et al., "Object enhancement and extraction," Picture processing and Psychopictorics, vol. 10, no. 1, pp. 15–19, 1970.
[34] H. Scharr, "Optimal operators in digital image processing," Ph.D. dissertation, 2000.
[35] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6848–6856.
[36] Z. He, G. Fu, Y. Cao, Y. Cao, J. Yang, and X. Li, "Eskn: Enhanced selective kernel network for single image super-resolution," Signal Processing, vol. 189, p. 108274, 2021.
[37] Z. He, D. Chen, Y. Cao, J. Yang, Y. Cao, X. Li, S. Tang, Y. Zhuang, and Z. ming Lu, "Single image super-resolution based on progressive fusion of orientation-aware features," Pattern Recognition, vol. 133, p. 109038, 1 2023.
[38] B. Li, W. Ren, D. Fu, D. Tao, D. Feng, W. Zeng, and Z. Wang, "Benchmarking single-image dehazing and beyond," IEEE Transactions on Image Processing, vol. 28, no. 1, pp. 492–505, 2018.
[39] Y. Liu, L. Zhu, S. Pei, H. Fu, J. Qin, Q. Zhang, L. Wan, and W. Feng, "From synthetic to real: Image dehazing collaborating with unlabeled real data," in Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 50–58.
[40] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity." IEEE Transactions on Image Processing, vol. 13, pp. 600–612, 4 2004.
[41] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in ICLR, 2015, pp. 1–15.
[42] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in CVPR. IEEE, 6 2019, pp. 558–567.
[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.