An Efficient Multiscale Spatial Rearrangement MLP Architecture for Image Restoration

Xia Hua, Zezheng Li, and Hanyu Hong

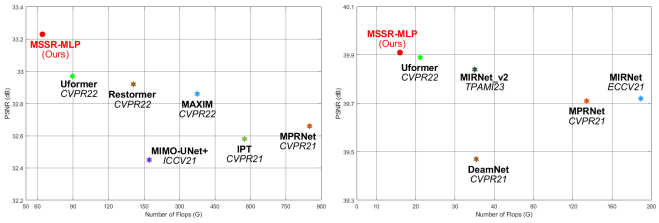
***Abstract*— The effective use of long-range information can yield improved network performance, which is very important for image restoration. Although local window-based models have linear complexity and can be feasibly applied to process high-resolution images, a single-scale window has a limited receptive field and is less efficient for encoding long-range context information. To address this issue, this paper presents a single-stage multiscale spatial rearrangement multilayer perceptron (MSSR-MLP) architecture that can obtain information at different scales within a local window. Specifically, we propose a simple and efficient spatial rearrangement module (SRM) that moves information outside the local window to the inside of the local window so that long-range dependencies can be modeled using only a window-based fully connected (FC) layer. The SRM can extend the local receptive field of a window-based FC layer without introducing additional parameters and FLOPs. Utilizing several spatial rearrangement modules with different step sizes, we design an efficient multiscale spatial rearrangement MLP architecture for image restoration. This design aggregates multiscale information to achieve improved restoration quality while maintaining a low computational cost. Extensive experiments conducted on several image restoration tasks demonstrate the efficiency and effectiveness of our method. For example, it requires only ∼4.3% of the FLOPs needed by SwinIR for Gaussian gray image denoising, ∼13.9% of the FLOPs needed by C^{2}PNet for single-image dehazing and ∼18.9% of the FLOPs needed by MAXIM for single-image motion deblurring but achieves better performance on each of these restoration tasks.**

***Index Terms*— Image restoration, spatial rearrangement, efficient multilayer perceptron model.**

I. Introduction

IMAGE restoration, which aims to restore a high-quality image from its low-quality version, has long been an important research topic in computer vision. Due to the ill-posed nature of the image restoration task, a variety of effective priors [1], [2], [3] such as the hyper-Laplacian prior for clear images [4] and deep residual prior for blur kernels [2], have been designed to obtain acceptable solutions. In recent years, deep convolutional neural networks (CNNs) have been extensively studied, and CNN-based models have demonstrated state-of-the-art performance on various image restoration tasks [5], [6], [7]. With the progress achieved in deep learning, researchers have found that models with multiscale structures can provide comprehensive encodings of the multiscale information contained in images while maintaining manageable computation and memory complexity levels [8]. As a pioneering work, DeepDeblur [9] was the first multiscale convolutional neural network introduced for this purpose, and it achieved state-of-the-art performance on image deblurring tasks. Inspired by the success of DeepDeblur [9], various CNN-based models have been designed to aggregate multiscale information for improving the performance attained in restoration tasks. For instance, Li et al. [10] proposed a multiscale residual network to fully capture multiscale image features and achieved state-of-the-art performance on the image superresolution task. Cho et al. [11] designed an asymmetric feature fusion (AFF) module to efficiently merge multiscale information for image deblurring.

These multiscale CNN-based models achieve better performance than that of traditional model-based methods but have a limited ability to capture long-range dependencies [13]. To address this issue, some researchers have used self-attention (SA) mechanisms to design transformer-based models [14] for image restoration due to the highly dynamic weights and globally dependent capture ability of self-attention. The earliest transformer-based image restoration models directly employed global self-attention, inevitably incuring quadratic



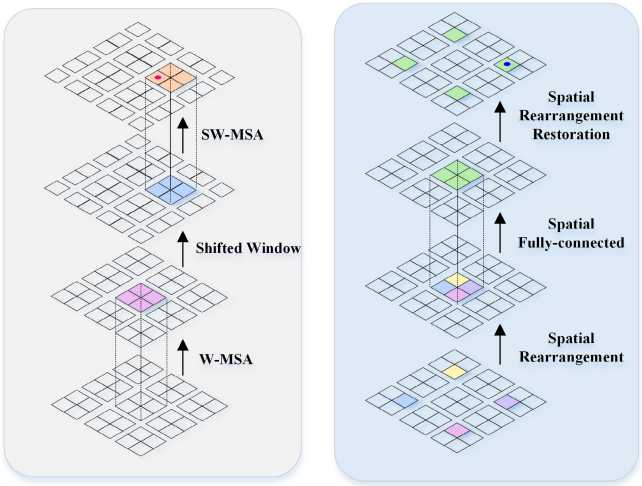
(a) Deblurring (b) Denoising

Fig. 1. PSNR vs. computational cost analyses conducted on the GoPro dataset [9] (left) and the SIDD dataset [12] (right). Our MSSR-MLP achieves state-of-the-art performance while being computationally efficient.The FLOPs are computed for image sizes of 256×256 in the image deblurring task and 128×128 in the image denoising task.

Manuscript received 30 March 2023; revised 24 August 2023 and 31 October 2023; accepted 8 November 2023. Date of publication 25 December 2023; date of current version 29 December 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 61801337 and Grant 62171329 and in part by the Wuhan Knowledge Innovation Special Project under Grant 2022010801010351. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Lisimachos P. Kondi. *(Corresponding authors: Hanyu Hong; Zezheng Li.)*

The authors are with the School of Electrical and Information Engineering, Wuhan Institute of Technology, Wuhan 430205, China, and also with the Hubei Key Laboratory of Optical Information and Pattern Recognition, Wuhan 430074, China (e-mail: hedahuaxia@wit.edu.cn; lzz924241978@gmail.com; hhyhong@wit.edu.cn).

Digital Object Identifier 10.1109/TIP.2023.3341700



(a) Shifted Window (b) Spatial Rearrangement

Fig. 2. The whole spatial rearrangement (left) and window shifting (right) processes. W-MSA and SW-MSA are multihead self-attention layers with regular and shifted window settings, respectively. The orange area represents the receptive field of the red dot after performing window shifting, and the green area represents the receptive field of the blue dot after conducting spatial rearrangement. It can be clearly observed that our module can capture long-range interactions outside the window, while the shifted window strategy cannot escape the scope of the local window.

computational complexity and hindering their applicability to high-resolution images. Therefore, local window-based transformers [15], [16] have been proposed to achieve improved efficiency. These methods partition the input features into nonoverlapping windows and limit the self-attention computation to the local window, which significantly reduces the computational complexity required for high-resolution input features. To introduce interactions between neighboring nonoverlapping windows, the shifted window strategy (see Fig. 2(a)) can be adopted, but this approach still cannot eliminate the local scope restriction introduced by window partitioning.

More recently, some studies have shown that self-attention may not be critical and that it can be replaced by a simple multilayer perceptron [17]. Along this line, many MLP architectures [18], [19], [20] have been developed for image-domain tasks resulting in promising performance. Although MLP models without self-attention seem to be simpler than transformers, the fully connected (FC) layers that serve as the core components of MLP models have the same computational complexity as the self-attention mechanisms in transformers. To reduce the incurred computational overhead, the window-based mechanisms used in transformers can also be applied in MLP models [21]. However, a window-based FC layer can only model the dependencies within a given local window. Other researchers have tried to improve the modeling ability of these methods in local window environments. The multi-axis MLP (MAXIM) [22] exchanges information across windows through the introduction of a global branch; however, this requires additional parameters and computations, and the dependencies in each MLP block are not rich enough to capture sufficient contextual information.

In summary, an inconvenient trade-off is formed between the ability to model long-range dependencies and computational efficiency. The reason why a window-based mechanism can reduce the incurred computational cost is that window partitioning limits the number of tokens considered in a self-attention mechanism or fully connected layer. Therefore, linear complexity can be preserved as long as window partitioning is performed before the self-attention mechanism or fully connected layer is executed.

From the above analysis, a question naturally arises: Can we use a small local window to capture dependencies with different ranges rather than just local dependencies? This would have two advantages, which are described as follows. First, the complexity of a local window-based model is linear with respect to the input resolution, making it more suitable for processing high-resolution images. Second, the existing methods have shown that operations with single-scale receptive field have low efficiency in terms of encoding context information, which is not conducive to image restoration [23]. However, if we could capture dependencies with different ranges through a fixed local window, this would provide us with more accurate and contextually enriched feature representations, thereby yielding improved model performance.

Motivated by this question, we design an efficient spatial rearrangement module (SRM), which shifts features outside the window to the inside of the window in accordance with a given step size before implementing the window partitioning operation so that long-distance information can be captured through a small spatial window. Since this spatial rearrangement destroys semantic information, the corresponding reverse spatial rearrangement (spatial rearrangement restoration) operation must be carried out after the window-based FC layer to accurately restore the original pixel arrangement. Fig. 2 illustrates a comparison between our spatial rearrangement module and the shifted window strategy. We can see that our module can capture long-range interactions outside the window, while the shifted window strategy cannot escape the scope of the local window.

With the SRM, through different rearrangement steps, information at different scales can be obtained in a small spatial window. We can efficiently aggregate this multiscale information by applying multiple SRMs with different step sizes for rearrangement purposes. This operation is called multiscale spatial rearrangement (MSSR). Due to the preservation of the window-based approach, the complexity of MSSR is still linear with respect to the input features. On the basis of MSSR and a channel MLP, we construct a multiscale spatial rearrangement MLP (MSSR-MLP) block and embed it into a U-shaped multiscale structure to obtain an efficient multiscale spatial rearrangement MLP architecture.

Finally, we apply our MSSR-MLP architecture to image restoration tasks, including image denoising, deblurring, deraining, and dehazing, and achieve state-of-the-art results on ten datasets with far fewer FLOPs than those needed by competing models, as shown in Fig. 1. Our main contributions can be summarized as follows.

- We propose a simple and efficient spatial rearrangement module (SRM) that moves information outside the local

window to the inside of the local window so that long-range dependencies can be efficiently modeled using only a window-based FC layer. The SRM extends the local receptive field of the window-based FC layer without introducing additional parameters and FLOPs.

- Due to the high efficiency of the SRM, we design an efficient multiscale spatial rearrangement (MSSR) MLP architecture for image restoration using multiple SRMs with different step sizes. The proposed model aggregates multiscale information to achieve improved restoration quality while maintaining a low computational cost.
- Extensive experiments conducted on several image restoration tasks demonstrate the efficiency and effectiveness of our method. For example, it requires only ∼4.3% of the FLOPs needed by SwinIR for Gaussian gray image denoising, ∼13.9% of the FLOPs needed by C^2PNet for single-image dehazing, ∼18.9% of the FLOPs needed by MAXIM for single-image motion deblurring and ∼45.4% of the FLOPs needed by Restormer for real single-image deraining but achieves better performance on each of these restoration tasks.

II. Related Work

In this section, we briefly introduce the relevant models in the literature, including CNN-based models, transformer-based models and MLP-based models.

A. CNN-Based Models

With the success of AlexNet [24], CNN-based models have come to be widely used in image restoration tasks [9], [11], [25], [26], [27], and they attain better performance each year. This progress can be attributed to their novel architectures and efficient module designs. For instance, many successful module designs are based on the U-shaped architecture, [11], [26], [28] which supports hierarchical multiscale information flows while maintaining high computational efficiency. Cho et al [11]. designed an efficient multi-input and multioutput U-shaped network and achieved favorable image deblurring results. Some researchers have attempted to design unified models for image restoration [6]. Pan et al. [7] proposed a physical model-constrained GAN model that generates much sharper images and can be effectively used for a variety of image restoration and low-level vision tasks. A general dual convolutional neural network (DualCNN) [5] with two parallel branches was designed to recover structures and details in an end-to-end manner and this approach can be easily integrated into existing CNNs. A residual dense network (RDN) [6] makes full use of hierarchical features to recover high-quality images. Recently, many attention modules have been implemented in low-level vision tasks, such as image deblurring [29] and image denoising [30]. A U-shaped structure is also selected as the basic architecture for our method, in addition to the application of our novel MSSR-MLP as a basic building block.

B. Transformer-Based Vision Models

Transformer-based models, in which an attention mechanism is used to model the long-range dependency relationships contained in the input data, were first proposed for NLP tasks [31]. As an expansion of the application scope of transformers, the vision transformer (ViT) [32] was developed for applications in the image classification field. To make vision transformers more efficient on training data, Touvron et al. [33] proposed several training strategies to obtain better results on much smaller datasets (e.g., ImageNet-1K [34]). However, the large computational cost incurred by self-attention in transformer models still remains to be addressed. The Swin transformer [21] computes self-attention based on nonoverlapping local windows, thus making the computational complexity of the self-attention calculations linear instead of quadratic with respect to the image size. Recently, transformer-based models have exhibited great potential for completing low-level vision tasks. For instance, Uformer [16] makes use of window-based self-attention to reduce the required computational cost and applies depthwise convolution in a feedforward network to further improve its ability to capture local features. CSformer [35] was the first approach to incorporated the recent popular masked autoencoder (MAE) pretraining technique into low-level vision tasks, and it achieved state-of-the-art performances on several image restoration tasks. Other transformer-based models [13], [15], [36] have also attained state-of-the-art performance on image restoration tasks.

C. MLP-Based Vision Models

Recently, simple and efficient MLP-based models [17], [18], [19], [22], [37], [38] using fully connected layers with nonlinear activation functions have become increasingly popular in the image processing field. One of the main contributions of these models is their ability to alleviate the large computational costs incurred by self-attention in transformers. For instance, MLP-Mixer [17] is a pure multilayer perceptron that does not include any convolutional layers. Liu et al. [18] proposed a spatial gating unit (g-MLP) to improve the performance of MLP architectures. CycleMLP [37] captures spatial information by using a cyclic fully connected layer that can handle variable image scales. In addition, by axially moving the channels of the feature map, AS-MLP [19] can obtain information flows from different axial directions and capture local dependencies. Similarly, S2-MLP [20] uses a spatial-shift-based MLP for interwindow communication. For low-level vision tasks, multiaxis MLP and cross-gating MLP blocks are used to build the backbone network structure of MAXIM [22] to balance the relationship between global and local features.

III. Method

Our goal is to design an efficient and effective MLP model for image restoration tasks. In this section, we first introduce the core module, the SRM. Then, we provide the details of the MSSR-MLP block, which contains two main components: an MSSR module and a channel-MLP. The MSSR module is designed based on the SRM. Finally, we describe the overall pipeline of our MSSR-MLP architecture, in which the MSSR-MLP block is the basic component.

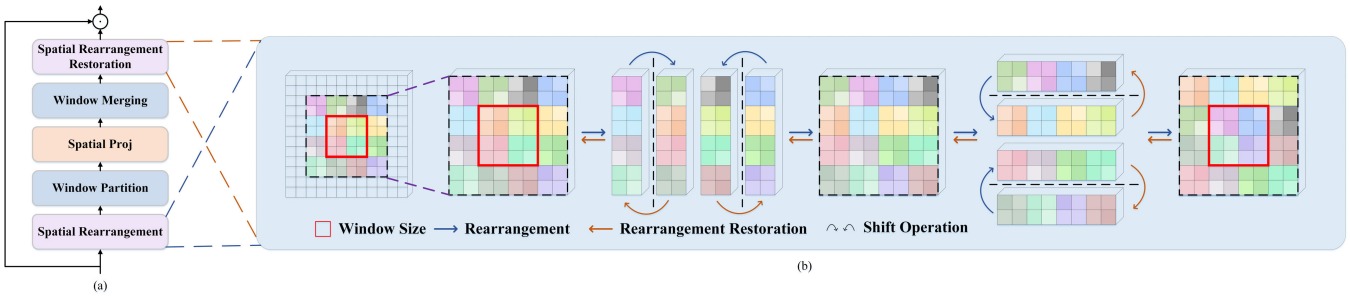


Fig. 3. (a) The spatial rearrangement module (SRM). (b) The detailed processes of spatial rearrangement and spatial rearrangement restoration. The SRM shifts features outside the window to the inside of the window in accordance with a given step size before performing the window partitioning operation so that long-range dependencies can be efficiently modeled using only a window-based FC layer. The SRM extends the local receptive field of the window-based FC layer without introducing additional parameters and FLOPs. As we can see, the 2×2 features at the four corners are moved into a 4×4 local window by an SRM with a step size of 2; therefore, after a spatial FC layer, the receptive field of the 4×4 local window is enlarged to 8×8.

A. Spatial Rearrangement Module

The use of a window-based spatial FC layer can significantly reduce the computational complexity of high-resolution feature mapping, but it is difficult to model multiscale information with a fixed window, and using multiscale windows (MSA) to extract multiscale information requires additional parameters and calculations, especially when the window size becomes very large. To obtain information at different scales within a fixed local window, we propose a **S**patial **R**earrangement **M**odule (**SRM**). As shown in Fig. 3(a), the SRM contains five units: a spatial rearrangement unit, a window partitioning unit, a spatial projection unit, a window merging unit and a spatial rearrangement restoration unit. To refine the spatial information obtained from spatial projection (the spatial FC layer), we apply a linear gating mechanism [22] after the spatial rearrangement restoration unit. The key unit of the SRM is the spatial rearrangement unit; this unit can generate information at different scales in local windows after performing window partitioning. In the following, we describe the specifics of the spatial rearrangement process.

The spatial rearrangement process consists of two steps: width-direction rearrangement and height-direction rearrangement. Because both follow the same process, we take width-direction rearrangement as an example. Width-direction rearrangement is implemented by shifting all spatial regions along a specific direction with a given step size s, as illustrated in Fig. 3(b) (s = 2 along the height direction). Given a feature map $X \in \mathbb{R}^{C \times H \times W}$, we assume that the local window size is $M \times M$. The input feature map X is split into multiple strip regions along the height direction, that is, $X = [X_1, X_2, \ldots, X_n]$, where $X_i \in \mathbb{R}^{C \times H \times M/2}$. Then, every two adjacent strip regions are divided into a group, i.e., $X = [G_1, G_2, \ldots, G_{n/2}]$, where $G_i = [X_{2i-1}, X_{2i}]$ and $G_i \in \mathbb{R}^{C \times H \times M}$. Along the width direction, a window-based spatial FC layer can capture only the features within the range of 0 to M. To enlarge the receptive field of the local window, for each group along the width direction, we rearrange the two strips immediately outside a group to the inside of that group. Taking G_i as an example, G_i is $[X_{2i}, X_{2i}]$ before performing spatial rearrangement and becomes $[X_{2i-2}, X_{2i+1}]$ after conducting spatial rearrangement with a step size of $s = M/2$. This operation is carried out simultaneously for all groups. Subsequently, we continue on to height-direction rearrangement. The process of height-direction rearrangement is similar to that of width-direction rearrangement except for the direction of the splitting and rearrangement operations, as shown in Fig. 3(b). To restore the original arrangement of the rearranged features to avoid losing the original information after implementing spatial projection, we choose to pad the features by copying the boundary parts in accordance with the step size before conducting spatial rearrangement and remove the padded parts before performing spatial projection.

Then, we split the rearranged features into nonoverlapping local small windows. Different long-range dependency relationships learned in the same window can be regarded as information of different scales. Therefore, each window contains spatial information at the scale corresponding to the step size used for rearrangement in the spatial rearrangement unit. We apply spatial projection to these windows to extract the scale information from the windows. Then, via the window merging unit and spatial rearrangement restoration unit, which perform the reverse processes of window partitioning and spatial rearrangement respectively, these windows are restored back to the features of the initial pixel arrangement. Finally, we apply a linear gating mechanism [22] to refine the spatial information captured from spatial projection. The linear gating mechanism can be described as follows:

$$G(x) = x \odot f_{W,b}(x) \tag{1}$$

where $\odot$ denotes the elementwise product operation; x and $G(x)$ denote the input and output, respectively, of linear gating, and $f_{W,b}$ denotes the fully connected layer used for spatial projection. Moreover, the computational cost is still only quadratic with respect to the small window size in the SRM. The spatial rearrangement unit introduces only a small additional computational cost.

B. Multiscale Spatial Rearrangement MLP

How to efficiently utilize multiscale information in an MLP model is an important research question. To address this question, we propose an MSSR-MLP block, which is illustrated in Fig. 4(b). We construct the block using two main designs: a multiscale spatial rearrangement (MSSR) module and a channel-MLP. The total process of the MSSR-MLP

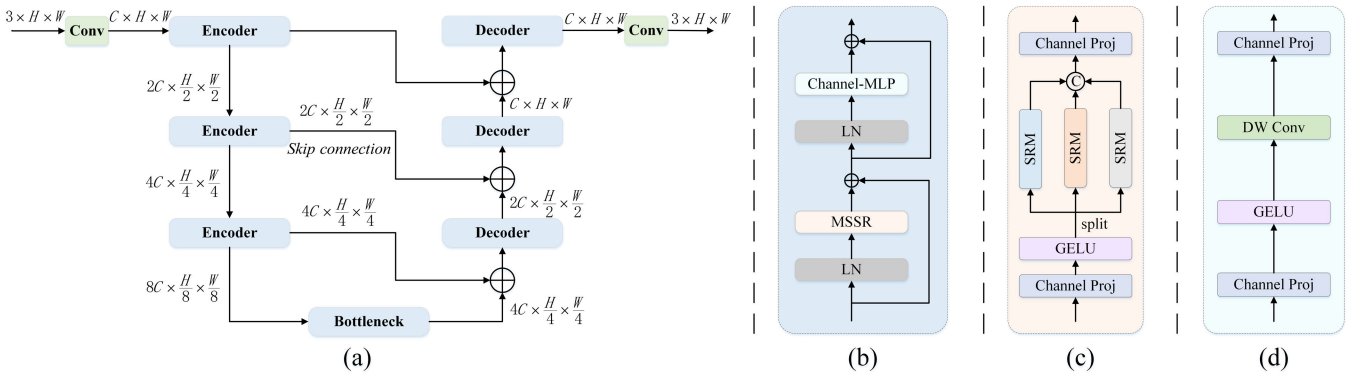


Fig. 4. (a) Architecture of the MSSR-MLP for image restoration. (b) Multiscale spatial rearrangement MLP (MSSR-MLP) block. (c) Multiscale spatial rearrangement (MSSR) module. (d) Channel-MLP.

block can be described as follows:

$$X_1 = MSSR(LN(X_0)) + X_0$$
$$X_2 = Channel\text{-}MLP(LN(X_1)) + X_1 \quad (2)$$

where X_0, X_1, and X_2 denote the input and output feature maps of the MSSR module and the output feature map of the channel-MLP, respectively, and LN denotes layer normalization [13]. In the following, we provide the details of the MSSR module and channel-MLP separately.

1) Multiscale Spatial Rearrangement (MSSR) Module: As shown in Fig. 4(c), the key element in the MSSR module is the SRM. Based on Section III-A, we combine several SRMs with different rearrangement step sizes to obtain multiscale information. To be more specific, given an input feature map, we first apply channel projection to extract information along the channel dimension and then use GELU [43] activation to achieve enhanced nonlinearity. Then, we split the feature map into three independent parts along the channel dimension. Subsequently, we apply SRMs with different step sizes to each part and obtain spatial information with different scales from each SRM. Finally, the outputs of these parts are concatenated along the channel dimension and fused via channel projection to generate multiscale information.

2) Channel-MLP: We follow a previous state-of-the-art MLP model to apply a channel-mixing MLP [17], which contains two fully connected layers with a GELU activation layer [43] between them for channel projection purposes to enhance the communication among different channels. As previously reported [44], local information and neighboring pixels are crucial for image restoration [45]. To make the channel-mixing MLP focus on local information, we add a depthwise convolution operation between two fully connected layers, as done in some recent state-of-the-art works [16], [44]. The details of the channel-MLP can be seen in Fig. 4(d).

C. Overall Pipeline

As shown in Fig. 4(a), using the MSSR-MLP block as the basic component, we build a single-stage multiscale spatial rearrangement MLP **(MSSR-MLP)** architecture, which is a U-shaped multiscale structure with skip connections between the encoder and decoder. In more detail, given an input $I \in R^{3\times H\times W}$, we first apply a 3×3 convolutional layer using LeakyReLU to generate low-level feature maps $X_0 \in \mathbb{R}^{C\times H\times W}$ as the encoder inputs. To leverage the multiscale structure, each encoder, decoder and bottleneck stage contains multiple MSSR-MLP blocks that aim to mix multiscale long-range spatial information. The MSSR-MLP blocks apply spatial projection on the feature maps through nonoverlapping windows to reduce required the computational cost. To obtain information at different scales within a fixed single-scale local window, the MSSR-MLP blocks perform spatial rearrangement on the feature maps by setting different rearrangement step sizes in our proposed spatial rearrangement module. This operation introduces no additional computational cost but aggregates the features to obtain multiscale information for image restoration. In addition, a downsampling layer follows each encoder stage and an upsampling layer is placed before each decoder stage. In the downsampling and upsampling layers, we apply a 3×3 convolution with a stride of 2 and a pixel-shuffling operation [26], respectively.

For the output of the last decoder stage, we apply a 3×3 convolutional layer to generate a residual image $R \in \mathbb{R}^{3\times H\times W}$ and then add the original input to obtain the restored image: $I' = I + R$. In addition, we use a content loss [9] and a frequency loss [11] to optimize our MSSR-MLP architecture. The content loss measures the distance between the restored images and the ground truths:

$$L_{content} = ||I_s - I_r||_1 \quad (3)$$

where I_s denotes the ground truth and I_r denotes the restored images. The frequency loss measures the distance between the restored images and the ground truths in the frequency domain:

$$L_{freq} = ||\mathcal{F}(I_s) - \mathcal{F}(I_r)||_1 \quad (4)$$

where $\mathcal{F}$ denotes the fast Fourier transform (FFT), which transfers the image signal to the frequency domain. Then, the total loss consists of the content loss and the frequency loss, which can be described as follows:

$$L_{total} = L_{content} + \lambda L_{freq} \quad (5)$$

where we experimentally set $\lambda = 0.1$.

IV. Experiments

In this section, we first evaluate our MSSR-MLP on several image restoration tasks, including single-image motion deblurring, defocus deblurring, image deraining, image denoising and image dehazing, and then conduct several ablation studies to verify the effectiveness of the core components of our model.

A. Experimental Setup

1) Datasets and Measurements: We evaluate the effectiveness and efficiency of the MSSR-MLP in several image restoration tasks on ten datasets: GoPro [9], HIDE [51] (for motion deblurring), DPDD [42] (for defocus deblurring), Set12 [52], BSD68 [53] (for grayscale image Gaussian denoising), SIDD [12], DND (for image denoising), SPAD [46] (for image deraining) and SOTS [54] (for image dehazing). To conduct quantitative comparisons, we mainly use PSNR and SSIM metrics for quality evaluation purposes in all restoration tasks; we add the MAE and LPIPS [55] for defocus deblurring, and DEHAZEfr [56] and DHQI [57] for dehazing.

2) Training Settings: In this section, we first introduce the general settings of our framework and then specify the different experimental settings utilized for different image restoration tasks. Our framework is end-to-end trainable, and no pretraining is needed. We train the MSSR-MLP architecture on 256×256 randomly cropped image patches using the PyTorch toolbox [58]. We apply the Adam optimizer [59] with the default parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 1 \times 10^{-8}$) to optimize the network parameters. The initial learning rate can be set higher if the network contains layer normalization (LN) [13]; this step can make the training process more stable. Therefore, we set the initial learning rate to 1×10^{-3} and subsequently decrease the learning rate to 1×10^{-6} via the cosine annealing strategy [60]. To improve the generalization ability of our framework, we implement data augmentation by randomly flipping and rotating the input images. We set the window size in all MSSR-MLP blocks to 4×4 for the comparisons with the recently proposed state-of-the-art methods and to 8×8 for the ablation studies. In addition to these common experimental settings, different image restoration tasks require different training settings. In the following, we describe the different training details applied for each task.

- **Single-Image Motion Deblurring.** Following the previous state-of-the-art methods [16], [22], [26], we train the MSSR-MLP only on the GoPro [9] dataset and directly evaluate it on the GoPro [9] and HIDE [51] testing datasets. We randomly crop the images in the training dataset to 256×256 patches for training and testing on the full-resolution test image pairs. The batch size is set to 4, and the total number of training epochs is 3000.
- **Dual-Pixel Defocus Deblurring.** We crop the DPD [42] training samples to 60% overlapping 512×512 patches for training. Following previous methods [16], [42], we apply the Sobel filter to discard the 30% of the patches with the lowest sharpness energy. For evaluation purposes, we test our MSSR-MLP on the full-size image pairs. The total training process lasts for 200 epochs, and the batch size is set to 1.
- **Real Single-Image Deraining.** For the real image deraining task, we train our model on the SPAD [46] dataset, which contains 638492 training image pairs. For evaluation purposes, we test our MSSR-MLP on the full-size image pairs. We randomly crop the training image pairs to 256×256 patches. We train our model for only 10 epochs with a batch size of 4 due to the sufficiently large size of the dataset and the fast convergence of our model. To make the training process more stable, we decrease the learning rate as the iterative procedureons progress.
- **Real Single-Image Denoising.** We train our model on the SIDD [12] dataset for the real image denoising task. For evaluation purposes, we directly evaluate our model on 1280 patches with sizes of 256×256 from the SIDD [12] dataset and 1000 patches with sizes of 512×512 from the DND [66] benchmark dataset. The results obtained on DND are evaluated online. Following a common training strategy used in recent state-of-the-art works [16], [30], [67], we randomly crop the SIDD [12] training image pairs to 128×128 patches. We train the model for 250 epochs with a batch size of 16.
- **Grayscale Single-Image Gaussian Denoising.** For the grayscale single-image Gaussian denoising task, we use randomly cropped patches derived from a combination of 800 DIV2K [68] images, 2650 Flickr2K images [69], 400 BSD500 images [70] and 4744 WED images [71] for training [13], [15]. The batch size is 16, and the patch size is 128×128. Consistent with the existing methods, we train and test a separate model for each noise level, where the noise levels are set to 15, 25 and 50. We test our model on the Set12 [52] and BSD68 datasets [53].
- **Single-Image Dehazing.** Following the previously developed state-of-the-art methods [72], we separately train and test our model on indoor and outdoor scenes. We select the full ITS dataset (13990 training image pairs) to train the indoor model and evaluate it on the indoor scenes (500 test image pairs) of the SOTS dataset [54]. We train the outdoor model on the OTS dataset (313950 training image pairs) and evaluate it on the outdoor scenes (500 test image pairs) of the SOTS dataset [54]. We randomly crop the training image pairs to 256×256 patches for training purposes. We train our model on ITS for 300 epochs and on OTS for 25 epochs with a batch size of 4.

3) Architectural Configuration: We design two MSSR-MLP architecture variants called MSSR-MLP-S (small) and MSSR-MLP-B (base) by embedding different numbers of MSSR-MLP blocks in each encoder, bottleneck and decoder stage. To conduct quantitative comparisons with the recently proposed state-of-the-art methods, we choose MSSR-MLP-B for training and testing. In the ablation experiments, we choose MSSR-MLP-S for training and testing. The details are listed as follows.

- MSSR-MLP-S: number of channels = 36, encoder depths: {1, 2, 4}, bottleneck: 4, decoder depths: {1, 2, 4}

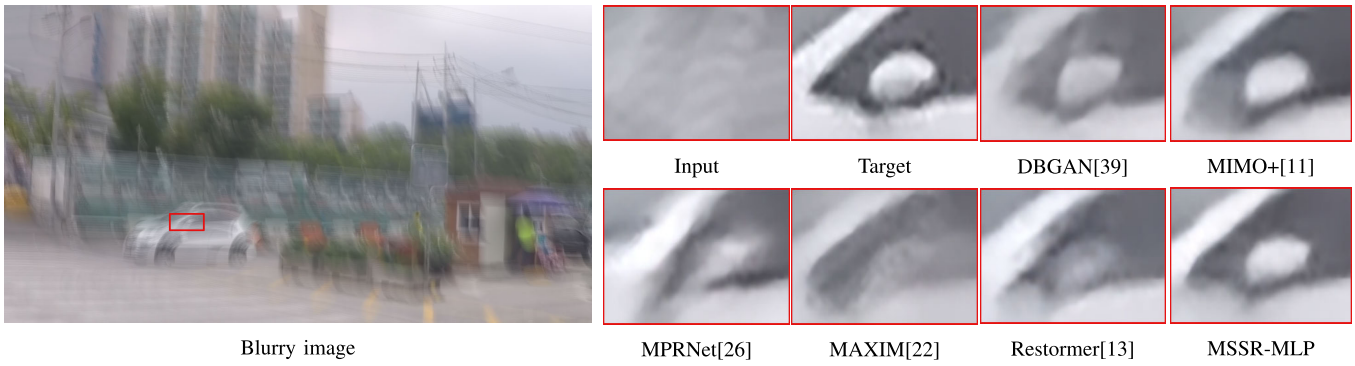


Fig. 5. Visual comparisons with the state-of-the-art methods on the GoPro dataset [9] for the motion deblurring task. Our MSSR-MLP is more effective at removing blur while preserving the original information.

TABLE I

QUANTITATIVE RESULTS OBTAINED ON THE GOPRO DATASET [9] FOR THE MOTION DEBLURRING TASK. FOR NAFNET [43], (+0.61) AND (+0.004) DENOTE THE IMPROVEMENTS ACHIEVED BY USING THE ADVANCED PATCH-BASED TESTING STRATEGY [43].THE BEST AND 2ND BEST RESULTS ARE HIGHLIGHTED AND UNDERLINED, RESPECTIVELY. THE FLOPS ARE COMPUTED FOR AN IMAGE WITH A SIZE OF 256 × 256

	CNN-based models					Transformer-based models			MLP-based models	
Method	DBGAN [39]	DMPHN [73]	MIMO-UNet+ [11]	MPRNet [26]	NAFNet [43]	IPT [14]	Restormer [13]	KiT [36]	MAXIM [22]	**MSSR-MLP**
Params(M)	11.59	21.7	16.11	20.13	67.8	-	26.13	-	22.20	15.68
FLOPs(G)	759.85	678.56	154.41	777.01	65	594	140.99(**+220%**)	**-**	339.2(**+530%**)	64.06
PSNR↑	31.10	31.20	32.45	32.66	33.08(+0.61)	32.58	32.92	32.70	32.86	**33.23**
SSIM↑	0.942	0.940	0.957	0.959	**0.963(+0.004)**	-	0.961	0.959	0.961	0.962

TABLE II

MOTION DEBLURRING RESULTS OBTAINED ON THE HIDE DATASET [51]. WE TRAIN OUR MODEL ON THE GOPRO DATASET [9] AND DIRECTLY EVALUATE IT ON THE HIDE DATASET [51]. THE FLOPS ARE COMPUTED FOR AN IMAGE WITH A SIZE OF 256 × 256

Method	SRN [74]	DMPHN [73]	Gao et al. [75]	Suin et al. [29]	MIMO-UNet+ [11]	HINet [25]	Uformer [16]	MPRNet [26]	Restormer [13]	**MSSR-MLP**
Params(M)	10.25	21.7	-	23.0	16.11	88.67	50.88	20.13	26.13	15.68
FLOPs(G)	108.63	678.56	-	536.74	154.41	170.72	89.46	777.01(**+1213%**)	140.99(**+220%**)	64.06
PSNR↑	28.36	29.09	29.11	29.98	29.99	30.32	30.83	30.96	**31.22**	30.96
SSIM↑	0.915	0.924	0.913	0.930	0.930	0.932	**0.952**	0.939	0.942	0.939

- MSSR-MLP-B: number of channels = 42, encoder depths: {2, 4, 12}, bottleneck: 4, decoder depths: {2, 4, 12}.

4) Testing Details: During the testing process, we pad the test images to a size that is a multiple of 64 by using symmetric padding on the boundary and crop the padded images back to the original size after the inference step. We note that for some methods, the advanced local-patch-based testing strategy [77] is adopted during testing. However, because this strategy can only be applied for models that contain global operations such as global average pooling or global attention [77], it is not a general testing strategy. To ensure fair comparisons, the restoration results obtained by all models in our experiments are inferred directly on the full-resolution images without using any special testing strategy; this is a common testing approach that is applicable for most state-of-the-art models.

B. Main Results

1) Single-Image Motion Deblurring Results: For the motion deblurring task, we train the MSSR-MLP on the GoPro [9] training set, which contains 2103 image pairs, and test it on two synthetic datasets, the GoPro [9] test set and HIDE [51], which contain 1111 and 2025 image pairs, respectively. Table I and Table II report the quantitative comparison results. Notably, our MSSR-MLP outperforms the recently proposed state-of-the-art methods by at least 0.15 dB on GoPro [9] test set and achieves competitive results on the HIDE [51] dataset. Therefore, the results demonstrate the acceptable generalization ability of our network. It is worth mentioning that our model requires fewer FLOPs to achieve this superior or competitive performance. For example, Restormer and MPRNet require $\sim 2\times$ and $\sim 12\times$ as many FLOPs as our proposed method, respectively, which indicates that our method achieves a better trade-off between efficiency and effectiveness. In addition, we provide visual results in Fig. 5. As the Fig. 5 shows, the restored images produced by our method are much sharper and closer to the ground truths than those of the competing models. Moreover, our MSSR-MLP requires fewer FLOPs than all other methods.

2) Dual-Pixel Defocus Deblurring Results: Defocus deblurring is performed on the DPDD [42] dataset, which contains 350 high-resolution image pairs for training and 76 image pairs for testing. Table III illustrates that the MSSR-MLP achieves state-of-the-art performance in terms of most of the evaluation metrics. In addition, our method requires fewer FLOPs than

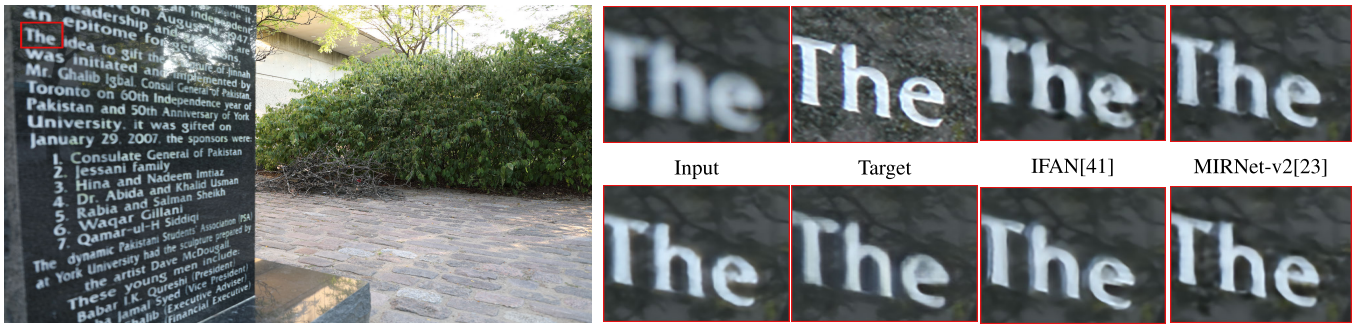


Fig. 6. Visual comparisons conducted on the DPDD dataset [42] for the defocused deblurring task. The restoration results of the MSSR-MLP are much sharper than those of the other methods.

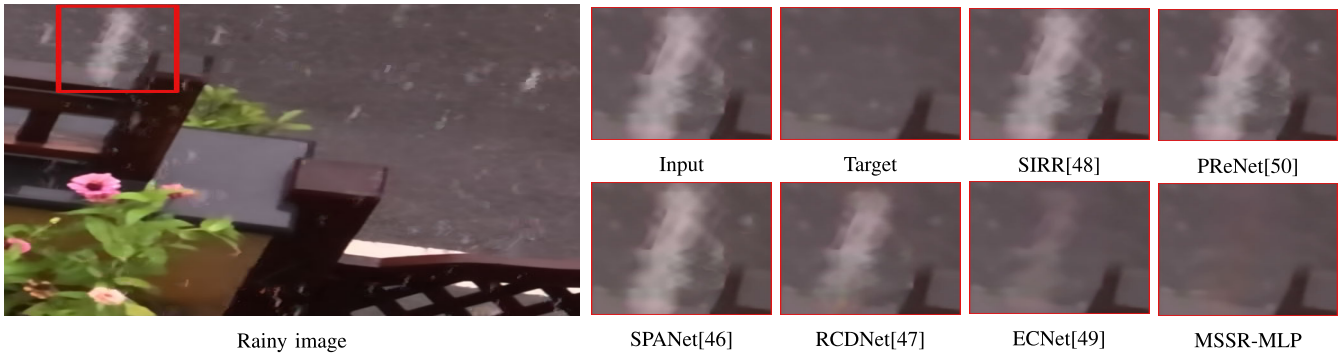


Fig. 7. Visual comparisons conducted on the SPAD dataset [46] for the image deraining task. The MSSR-MLP can remove real rain more effectively than the other methods.

TABLE III

DUAL-PIXEL DEFOCUS DEBLURRING COMPARISON RESULTS OBTAINED ON THE DPDDDATASET [42]. THE FLOPS ARE COMPUTED FOR AN IMAGE WITH A SIZE OF 256 × 256

Method	Params(M)	FLOPs(G)	PSNR↑	SSIM↑	MAE↓	LPIPS↓
DPDNet[42]	-	-	25.13	0.786	0.041	0.223
RDPD[76]	-	-	25.39	0.772	0.040	0.255
Uformer[16]	50.88	89.57	25.65	0.795	0.039	0.243
IFAN[41]	10.48	29.89	25.99	0.804	0.037	0.207
DRBNet[40]	11.69	49.20	26.33	0.811	0.037	**0.154**
Restormer[13]	26.13	141.38(**+220%**)	26.66	0.833	0.035	0.155
MIRNet-v2[23]	5.86	104.49(**+163%**)	26.20	0.816	0.037	0.180
MSSR-MLP	15.68	64.21	**26.72**	**0.835**	**0.035**	0.155

the recently developed state-of-the-art methods. For instance, Restormer requires ∼ 2× as many FLOPs as our method, which demonstrates the efficiency of our proposed method. Fig. 6 shows the qualitative results obtained for the dual pixel defocus deblurring task. Our method is more effective than the other approaches at removing defocus blur while preserving image details.

3) Real Single-Image Deraining Results: Image deraining experiments are performed on the real-world rainy SPAD dataset [46], which contains 638492 rainy/clean image pairs for training and 1000 image pairs for testing. We compute the PSNR and SSIM metrics by using the Y channel in the YCbCr color space according to previous methods [13], [16], [26]. As shown in Table IV, our model is the only approach that surpasses a PSNR of 48 dB, and the second-best method, Restormer [13], requires ∼ 2× as many FLOPs as our method, which strongly indicates that our method achieves the best balance between efficiency and effectiveness. Fig. 7 shows that our method can remove real rain much better than the competing models.

4) Grayscale Single-Image Gaussian Denoising Results: For the grayscale single-image Gaussian denoising task, we evaluate our model on Set12 (12 test images) and BSD68 (68 test images) at different noise levels following the previously developed state-of-the-art methods [13], [15]. Table V shows the quantitative grayscale image denoising results. In addition to the PSNR results, we also list the numbers of parameters and FLOPs required by our method and the compared methods as measures of their efficiency. Our model requires significantly fewer FLOPs while achieving state-of-the-art performance in comparison with the other methods. For instance, our model achieves better performance than DRUNet [84] and SwinIR [15], which require ∼ 4× and ∼ 23× as many FLOPs, respectively. Fig. 8 shows visual comparisons among the results obtained for the grayscale Gaussian denoising task. Visually, our model is more effective at removing Gaussian noise than the other approaches.

5) Real Single-Image Denoising Results: To train our model for real image denoising tasks, we select 320 high-resolution images from the SIDD [12] dataset and evaluate its performance on 1280 patches from the SIDD [12] test dataset and 1000 patches from the DND [66] benchmark dataset. Table VI shows that our MSSR-MLP obtains competitive

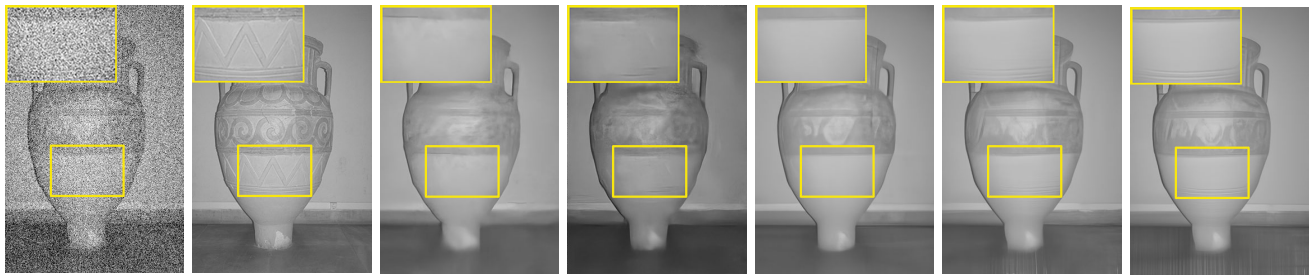


Fig. 8. Visual comparisons conducted on the BSD68 dataset [53] for the grayscale single-image Gaussian denoising task. The MSSR-MLP can generate much cleaner denoising results than the other methods.

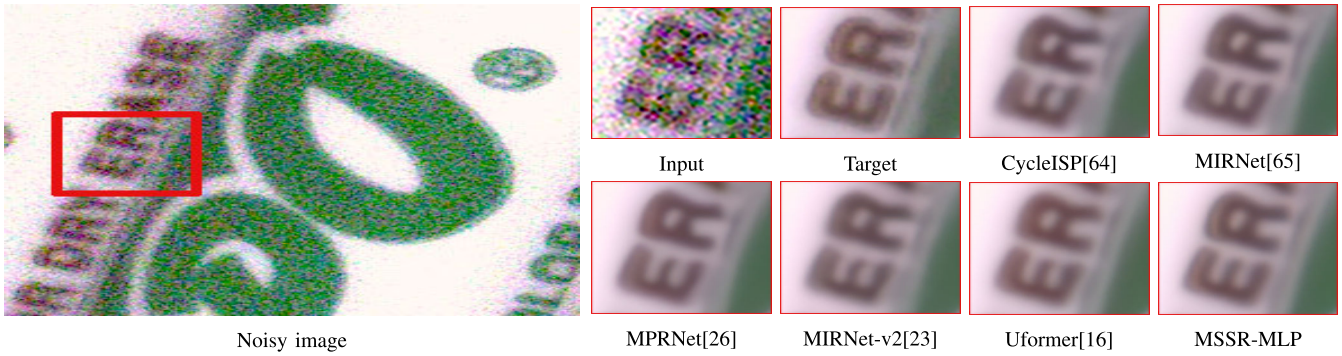


Fig. 9. Visual comparisons conducted on the SIDD dataset [12] for the image denoising task. The MSSR-MLP can generate much cleaner denoising results than the other methods.

TABLE IV

RESULTS OBTAINED ON THE SPAD DATASET [46] FOR THE REAL IMAGE DERAINING TASK. OUR METHOD ACHIEVES SIGNIFICANT GAINS OVER THE PREVIOUSLY DEVELOPED STATE-OF-THE-ART METHODS. THE FLOPS ARE COMPUTED FOR AN IMAGE WITH A SIZE OF 256 × 256

Method	SIRR [48]	RESCAN [78]	PReNet [50]	SPANet [46]	RCDNet [47]	SPDNet [79]	MPRNet [26]	ECNet [49]	Uformer [16]	Restormer [13]	**MSSR-MLP**
Params(M)	-	0.15	0.17	0.28	3.17	3.32	20.13	2.46	50.88	26.13	15.68
FLOPs(G)	-	32.32	66.58	26.68	194.22	96.92	777.01	15.86	89.46(**+140%**)	140.99(**+220%**)	64.06
PSNR↑	35.31	38.11	40.17	40.24	41.47	43.55	43.64	44.32	47.84	<u>47.98</u>	**48.64**
SSIM↑	0.9411	0.9707	0.9816	0.9811	0.9834	0.9875	0.9844	0.9913	<u>0.9925</u>	0.9921	**0.9936**

TABLE V

GRAYSCALE SINGLE-IMAGE GAUSSIAN DENOISING RESULTS OBTAINED ON THE SET12 [52] AND BSD68 DATASETS [53] WITH DIFFERENT NOISE-LEVELS. σ REFERS TO THE NOISE LEVEL, WITH A LARGER VALUE DENOTING A HIGHER NOISE LEVEL. THE FLOPS ARE COMPUTED FOR AN IMAGE WITH A SIZE OF 128 × 128

Method		DnCNN[52]	IRCNN[[80]	FFDNet[81]	NLRN[82]	RDN[6]	RRC[83]	MWCNN[62]	DAGL[61]	DRUNet[84]	SwinIR[15]	CODE[63]	**MSSR-MLP**
#Params		0.56M	0.19M	0.49M	0.34M	22.0M	-	16.15M	5.73M	32.64M	11.43M	12.18M	15.68M
FLOPs		18.22G	6.09G	3.98G	1382.08G	-	-	28.95G	64.39G	71.71G(**+448%**)	373.02G(**+2333%**)	22.52G(**+141%**)	15.99G
Set12	σ =15	32.86	32.76	32.75	33.16	-	-	33.15	33.28	33.25	<u>33.36</u>	33.33	**33.36**
	σ =25	30.44	30.37	30.43	30.80	-	-	30.79	30.93	30.94	<u>31.01</u>	<u>31.01</u>	**31.04**
	σ =50	27.18	27.12	27.32	27.64	27.60	26.90	27.74	27.81	27.90	27.91	<u>27.93</u>	**28.00**
BSD68	σ =15	31.73	31.63	31.63	31.88	-	-	31.86	31.93	31.91	<u>31.97</u>	31.96	**31.97**
	σ =25	29.23	29.15	29.19	29.41	-	-	29.41	29.46	29.48	29.50	<u>29.51</u>	**29.53**
	σ =50	26.23	26.19	26.29	26.47	26.41	26.14	26.53	26.51	<u>26.59</u>	26.58	26.58	**26.64**

results compared those of Restormer, which is the state-of-the-art method; it also achieves 0.19 dB and 0.02 dB gains over MIRNet [65] and Uformer [16], respectively, which represent the previous best CNN method and window-based transformer method. Although our method is the second-best approach in terms of the PSNR metric, it requires the fewest FLOPs. For example, Restormer and Uformer require $\sim 2\times$ and $\sim 1.3\times$ as many FLOPs as our method, respectively, which indicates that our method achieves a better balance between efficiency and effectiveness than the other methods. As seen from Fig. 9, our MSSR-MLP can effectively remove noise and produce restoration results there are closer to the original sharp image while maintaining textural details.

6) Single-Image Dehazing Results: To investigate the proposed method on the image dehazing task, we conduct dehazing experiments on the RESIDE [54] dataset. As shown

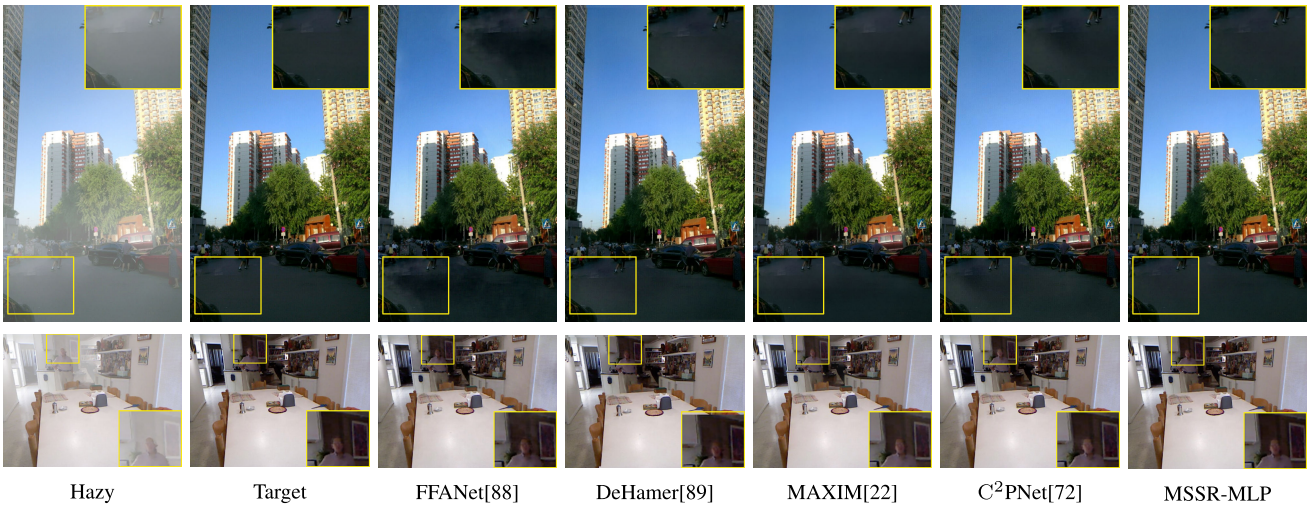


Fig. 10. Top row: dehazing results obtained for the SOTS [54] outdoor scenes. Bottom row: dehazing results obtained for the SOTS indoor scenes. Visual comparisons are conducted on the SOTS dataset [54] for the image dehazing task. Our proposed MSSR-MLP performs better than the other state-of-the-art methods.

TABLE VI

DENOISING RESULTS OBTAINED ON THE BENCHMARK SIDD [12] AND DND [66] DATASETS. OUR MODEL IS ONLY TRAINED ON SIDD [12] AND DIRECTLYTESTED ON DND [66]. "*" INDICATES THAT CSFORMER HAS NOT BEEN PRE-TRAINED. THE FLOPS ARE COMPUTED FOR AN IMAGE WITH A SIZE OF 128 × 128

Method	Params(M)	FLOPs(G)	SIDD[12]		DND[66]	
			PSNR↑	SSIM↑	PSNR↑	SSIM↑
BM3D[85]	-	-	25.65	0.685	34.51	0.851
RIDNet[86]	1.5	38.71	24.5	0.914	39.26	0.953
VDN[87]	7.8	39.28	12.37	0.909	39.38	0.952
DANet[67]	9.1	3.71	39.30	0.916	39.47	0.955
CycleISP[64]	2.83	47.32	39.52	0.957	39.56	0.956
MPRNet[26]	15.74	147.03	39.71	0.958	39.80	0.954
MIRNet[65]	31.79	196.61	39.72	0.959	39.88	0.956
NBNet[30]	13.3	22.2	39.75	0.959	39.89	0.955
MIRNet-v2[23]	5.86	35.09	39.84	0.959	39.86	0.955
CSformer*[35]	-	-	40.00	0.960	40.00	0.956
Uformer[16]	50.88	20.36(**+127%**)	39.89	0.960	39.98	0.955
Restormer[13]	26.13	35.25(**+220%**)	**40.02**	0.960	**40.03**	0.956
MSSR-MLP	15.68	16.01	39.91	**0.960**	39.96	**0.956**

in Table VII, our method surpasses PSNRs of 43 dB and 39 dB on indoor scenes and outdoor scenes, respectively, and is the only approach to do so. Moreover, compared with the recently proposed state-of-the-art methods, our model requires far fewer FLOPs. For example, MAXIM and C^{2}PNet require $\sim 3\times$ and $\sim 7\times$ as many FLOPs as our method, respectively. We also use DHQI [57] and DEHAZEfr [56], which were specifically designed for image dehazing, to quantitatively measure the quality of the dehazing results. As shown in Table VII, in addition to the traditional PSNR and SSIM metrics, our method still performs better than the other competing methods in terms of DHQI [57] and DEHAZEfr [56] in both indoor and outdoor scenes, which can convincingly demonstrate the effectiveness of our method. These results demonstrate that our model achieves the best balance between efficiency and effectiveness. Fig. 10 exhibits comparisons among the visual results obtained for the single-image dehazing task, from which it is seen that our model performs better than the other models.

C. Examples With Different Step Sizes

Fig. 11 shows the whole spatial rearrangement and spatial rearrangement restoration processes at step sizes of 0, 2 and 4. We can observe that the receptive field in the 4×4 local window is enlarged to 8×8 and 12×12 with step sizes of 2 and 4, respectively.

D. Computational Overhead Comparisons

In Table VIII and Table IX, we list the numbers of parameters, FLOPs and inference times of the recently developed state-of-the-art image deblurring and image denoising methods to further demonstrate the efficiency of our model. Note that for the image deblurring and denoising tasks, the FLOPs are calculated on 256×256 and 128×128 images, respectively, and the inference time is computed on the full-size images. As seen in these tables, our model achieves comparable performance to that of the other methods on GoPro [9] and SIDD [12]. Although our PSNR is lower than that of Restormer on SIDD [12], the MSSR-MLP has a faster speed and achieved better performance on GoPro with only ∼60% and ∼45.4% as many parameters and FLOPs, respectively. This demonstrates the efficiency and effectiveness of our model.

E. Ablation Study

To demonstrate the effects of the core components in the MSSR-MLP block, we conduct several ablation experiments based on the MSSR-MLP-S architecture. We train MSSR-MLP-S for 1000 epochs on the GoPro [9] training dataset with the batch size set to 4. For evaluation purposes, we select the PSNR and FLOPs as the evaluation metrics and test them on the GoPro [9] test dataset with full-resolution image pairs.

TABLE VII

IMAGE DEHAZING RESULTS OBTAINED ON THE SOTS DATASET [54]. THE FLOPS ARE COMPUTED ON AN IMAGE WITH A SIZE OF 256 × 256

Method	Params(M)	FLOPs(G)	SOTS-indoor				SOTS-outdoor			
			PSNR	SSIM	DHQI	DEHAZEfr	PSNR	SSIM	DHQI	DEHAZEfr
PFDN(AAAI20)[90]	11.27	50.46	32.68	0.976	-	-	-	-	-	-
MSBDN(CVPR20)[91]	31.35	41.54	33.67	0.985	-	-	33.48	0.982	-	-
FFA-Net(AAAI20)[88]	4.456	287.8	36.39	0.989	69.04	0.977	33.57	0.984	66.97	0.977
DeHamer(CVPR22)[89]	132.45	48.93	36.63	0.988	69.23	0.981	35.18	0.986	63.58	0.966
PMNet(ECCV22)[92]	18.90	81.13	38.41	0.990	-	-	34.74	0.985	-	-
MAXIM-2S(CVPR22)[22]	14.1	216.4(**+338%**)	38.11	0.991	69.53	0.983	34.19	0.985	66.55	0.964
C^2PNet(CVPR23)[72]	7.17	462.24(**+722%**)	42.56	0.995	69.49	0.991	36.68	0.990	53.67	0.958
MSSR-MLP	15.68	64.06	**43.01**	**0.995**	**69.53**	**0.991**	**39.02**	**0.992**	**67.44**	**0.985**

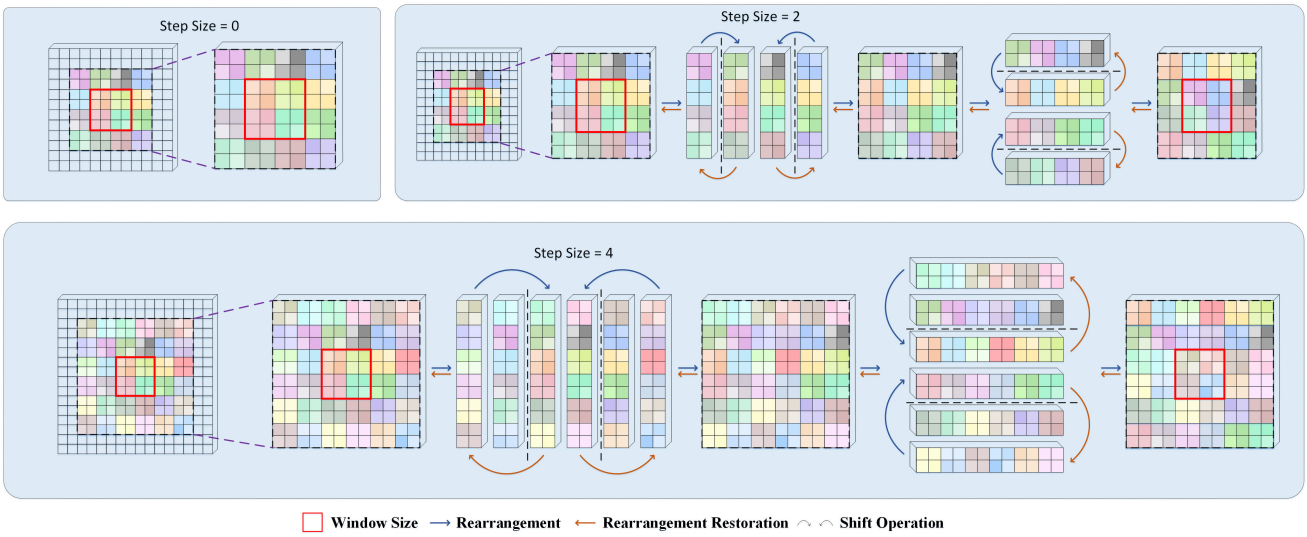


Fig. 11. The detailed processes of spatial rearrangement and spatial rearrangement restoration with different step sizes. The receptive field of the 4 × 4 local window is enlarged to 8 × 8 and 12 × 12 after performing spatial rearrangement with step sizes of 2 and 4, respectively.

TABLE VIII

OVERALL COMPUTATIONAL OVERHEADS COMPARISON FOR THE IMAGE DEBLURRING(GOPRO) TASK [9]. THE FLOPS ARE COMPUTED FOR AN IMAGE WITH A SIZE OF 256 × 256. THE INFERENCE TIMES ARE COMPUTED ON THE FULL-SIZE IMAGE

Method	Params(M)	FLOPs(G)	Time(s)	PSNR(dB)↑
Restormer[13]	26.13	140.99	1.172	32.92
Uformer[16]	50.88	89.46	0.883	32.97
MAXIM[22]	22.2	339.2	-	32.86
MPRNet[26]	20.13	777.01	1.127	32.66
DMPHN[73]	21.7	678.56	1.071	31.20
MSSR-MLP	**15.68**	**64.06**	**0.570**	**33.23**

TABLE IX

OVERALL COMPUTATIONAL OVERHEADS COMPARISON FOR THE IMAGE DENOISING (SIDD) [12] TASK. THE FLOPS ARE COMPUTED FOR AN IMAGE WITH A SIZE OF 128 × 128.THE INFERENCE TIMES ARE COMPUTED ON THE FULL-SIZE IMAGE

Method	Params(M)	FLOPs(G)	Time(s)	PSNR(dB)↑
Restormer[13]	26.13	35.25	0.067	**40.02**
Uformer[16]	50.88	22.36	0.065	39.89
MPRNet[26]	15.74	147.03	0.096	39.71
MIRNet[65]	31.79	196.61	0.097	39.72
MIRNet-v2[23]	**5.86**	35.09	0.060	39.84
MSSR-MLP	15.68	**16.01**	**0.059**	39.91

1) Window Shifting vs. Spatial Rearrangement: The shifted window strategy used in window-based transformer models can introduce interactions between neighboring nonoverlapping windows. However, it cannot fully overcome the local nature of the scope imposed by window partitioning. In contrast, our spatial rearrangement module (SRM) shifts information located outside the local window to the inside of the local window so that long-range dependencies can be efficiently modeled using only a window-based FC layer. We compare the shifted window method with our proposed spatial rearrangement module using MSSR-MLP-S as a backbone, as shown in Fig. 12. In Table X, the SRM achieves 0.25 dB and 0.001 gains in the PSNR and SSIM

TABLE X

PERFORMANCE COMPARISONS BETWEEN THE SHIFTED WINDOW METHOD AND OUR SPATIAL REARRANGEMENT MODULE IN AN MLP BLOCK AND A TRANSFORMER BLOCK

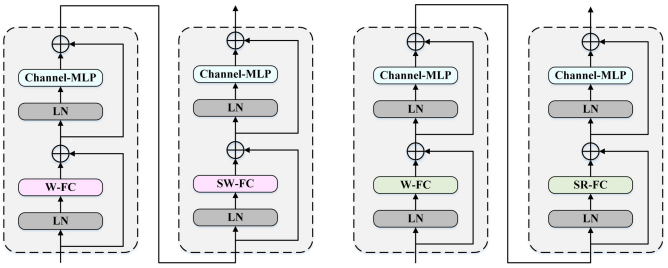
Method	Description	PSNR↑	SSIM↑	Params(M)	FLOPs(GMacs)
MLP	SW-FC	31.41	0.947	6.1	18.67
	SR-FC	**31.66**	0.948	6.1	18.67
Transformer	SW-MSA	31.32	0.946	7.12	20.87
	SR-MSA	31.64	**0.949**	7.12	20.87

TABLE XI
EFFECTS OF THE SPATIAL REARRANGEMENT RESTORATION UNIT

Method	Params	PSNR↑	SSIM↑
w/o the spatial rearrangement restoration unit	8.33	31.79	0.950
w the spatial rearrangement restoration unit	8.33	**31.94**	**0.952**

TABLE XII
EFFECTS OF THE LINEAR GATING MECHANISM

SRM	Params(M)	PSNR↑	SSIM↑
w/o the linear gating mechanism	6.10	31.45	0.947
w the linear gating mechanism	6.10	**31.52**	**0.948**

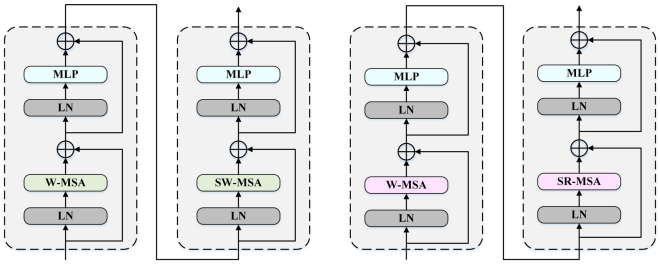


(a) Two successive MLP blocks with shifted window (b) Two successive MLP blocks with spatial rearrangement

Fig. 12. Architecture comparisons between MLP blocks with window shifting and spatial rearrangement. (a) Two successive MLP blocks with window shifting, where W-FC and SW-FC are fully connected layers with regular and shifted windowing settings, respectively. (b) Two successive MLP blocks with spatial rearrangement, where SW-FC is replaced by a fully connected layer with spatial rearrangement (SR-FC).

TABLE XIII
SINGLE-SCALE AND MULTISCALE EVALUATION RESULTS. THE LAST FIVE ROWS SHOW THE PERFORMANCE COMPARISONS BETWEEN MULTI-SCALE WINDOWS AND MULTISCALE SPATIAL REARRANGEMENT FOR OBTAINING MULTISCALE INFORMATION. K: THE SCALE LEVEL OF EACH BUILDING BLOCK. S: THE STEP SIZE IN SRM. THE FLOPS ARE COMPUTED FOR AN IMAGE WITH A SIZE OF 256 × 256

Method	K	Window	PSNR↑	Params (M)	FLOPs (G)
SSW-MLP	1	8	31.52	6.10	18.64
		16	31.74	7.21	21.58
		24	31.58	12.01	28.95
		32	31.49	24.92	33.36
MSW-MLP	2	[8,16]	31.81	8.33	25.62
	3	[8,16,24]	31.64	15.35	39.98
		[8,16,32]	31.89	28.26	44.38
MSSR-MLP	2	8, S=4	31.82	7.22	22.68
	3	8, S=4, 8	**31.94**	8.33	26.72



(a) Two successive transformer blocks with shifted window (b) Two successive transformer blocks with spatial rearrangement

Fig. 13. Architecture comparisons between transformer blocks with window shifting and spatial rearrangement. (a) Two successive transformer blocks with window shifting, where W-MSA and SW-MSA are multihead self-attention layers with regular and shifted windowing settings, respectively. (b) Two successive transformer blocks with spatial rearrangement, where SW-MSA is replaced by a multihead self-attention layer with spatial rearrangement (SR-MSA).

TABLE XIV
SINGLE-SCALE AND MULTISCALE PERFORMANCE COMPARISONS BETWEEN THE WINDOW-BASED TRANSFORMER BLOCK [16] AND MSSR-MLP BLOCK. K: THE SCALE LEVEL OF EACH BUILDING BLOCK. S: THE STEP SIZE IN SRM. THE FLOPS ARE COMPUTED FOR AN IMAGE WITH A SIZE OF 256 × 256

Method	K	Window	PSNR↑	Params (M)	FLOPs (G)
SSW-Transformer[16]	1	8	31.32	7.12	22.68
		16	31.29	7.17	28.56
SSW-MLP	1	8	31.52	6.10	18.64
		16	31.74	7.21	21.58
MSW-Transformer[16]	2	[8,16]	31.54	10.33	39.70
	3	[8,16,24]	31.68	13.11	65.01
MSSR-MLP	2	8, S=4	31.82	7.22	22.68
	3	8, S=4, 8	**31.94**	8.33	26.72

metrics, respectively, over the shifted window method for a window-based MLP. Similarly, in Fig. 13, we extend the spatial rearrangement approach to a Transformer model by alternating between self-attention and spatially rearranged self-attention. For a window-based transformer, our SRM can also achieve increases of 0.32 dB and 0.003 in the PSNR and SSIM metrics, respectively, over the shifted window method, as shown in Table X.

2) Is the Spatial Rearrangement Restoration Unit Necessary?: Since the relative positions of the pixels are crucial for constructing semantic information, the output obtained after the spatial FC layer should be passed through the spatial rearrangement restoration unit to ensure accurate alignment with the original input before applying the gating mechanism. Table XI shows that the spatial rearrangement restoration unit achieves a gain of 0.15 dB for the model.

3) Is the Linear Gating Mechanism Important?: Table XII shows the performance achieved when the linear gating mechanism is and is not applied to refine the information in the SRM. The linear gating mechanism [22] can yield an improvement of 0.07 dB for the image deblurring task.

4) Effects of Multiscale Spatial Rearrangement: To evaluate the effects of multiscale spatial rearrangement in our MSSR-MLP block, we replace all multiscale spatial rearrangement (MSSR) modules in our model with single-scale window (SSW)-based FC layers and multiscale window (MSW)-based FC layers, while leaving all other parts of the model unchanged, resulting in an SSW-MLP and an MSW-MLP, respectively. The MSW-MLP uses a window with a different size for each scale branch, while our MSSR-MLP uses a fixed-size window with a different rearrangement step size for each scale branch. The results are reported in Table XIII. Although

TABLE XV

MORE SINGLE-SCALE AND MULTISCALE EVALUATION RESULTS WITH DIFFERENT WINDOW SIZES. THE LAST ELEVEN ROWS SHOW THE PERFORMANCE COMPARISONS BETWEEN MULTISCALE WINDOWS AND MULTISCALE SPATIAL REARRANGEMENT FOR GENERATING MULTISCALE INFORMATION. K: THE SCALE LEVEL OF EACH BUILDING BLOCK. S: THE STEP SIZE IN SRM. THE FLOPS ARE COMPUTED FOR AN IMAGE WITH A SIZE OF 256×256

Method	K	Window	PSNR↑	Params (M)	FLOPs (G)
SSW-MLP	1	4	31.55	6.03	17.94
		6	31.51	6.05	18.21
		8	31.52	6.10	18.64
		10	31.55	6.21	19.19
		16	31.74	7.21	21.58
MSW-MLP	2	[8,16]	31.81	8.33	25.62
	3	[8,16,24]	31.64	15.35	39.98
		[8,16,32]	31.89	28.26	44.38
MSSR-MLP	2	4, S =2	32.00	7.08	21.20
		6, S =3	31.80	7.12	21.82
		8, S =4	31.82	7.22	22.68
		10, S =5	31.85	7.43	23.78
	3	4, S =2, 4	**32.08**	8.12	24.54
		6, S =3, 6	31.93	8.18	25.43
		8, S =4, 8	31.94	8.33	26.72
		10, S =5, 10	31.97	8.66	28.37

the multiscale window-based MSW-MLP performs better than the SSW-MLP, it incurs much greater increases in the numbers of parameters and FLOPs than our MSSR-MLP. At the scale level of three, our MSSR-MLP achieves better performance than the MSW-MLP with only 29.48% of its parameters and 60.21% of its computational cost. This study indicates the cost-effectiveness of multiscale spatial rearrangement in comparison with the multiscale window mechanism.

5) MSW-Transformer vs. the MSSR-MLP: To further demonstrate the efficiency and effectiveness of the MSSR-MLP block, we compare the proposed model with a window-based transformer [7] from two perspectives: first, we compare the performances of the MLP (SSW-MLP) and transformer (SSW-Transformer) blocks with fixed-scale windows; second, the performances of the multiscale spatial rearrangement (MSSR-MLP) block and the multiscale window transformer (MSW-Transformer) block are compared, where MSW-Transformer uses a window with a different size for each scale branch, while our MSSR-MLP uses a fixed-size window with a different rearrangement step size for each scale branch. For fair comparisons, we adapt the default settings of the MSSR-MLP-S architecture. As shown in Table XIV, for a fixed window scale, our SSW-MLP achieves better performance than SSW-Transformer with a lower computational cost and fewer parameters. This comparative advantage is more obvious at the multiscale level. At the scale level of three, compared to MSW-Transformer, our MSSR-MLP achieves better performance with 58.9% fewer FLOPs. This shows that the MSSR-MLP architecture can efficiently obtain multiscale information to achieve improved image restoration performance over that of MSW-Transformer.

TABLE XVI

EXTRA QUALITY ASSESSMENT COMPARISONS CONDUCTED ON THE SPAD DATASET FOR THE IMAGE DERAINING TASK

Method	SPANet[46]	RCDNet[47]	ECNet[49]	Restormer[13]	MSSR-MLP
BPRI	0.039	0.044	0.055	0.059	**0.066**
BMPRI	28.04	33.24	37.20	38.85	**40.11**
UCA	1.011	1.002	1.008	1.021	**1.032**

TABLE XVII

EXTRA QUALITY ASSESSMENT COMPARISONS CONDUCTED ON THE SIDD DATASET FOR THE IMAGE DENOISING TASK

Method	CycleISP[64]	MPRNet[26]	MIRNet[65]	Restormer[13]	MSSR-MLP
BPRI	0.017	0.019	0.023	0.019	**0.026**
BMPRI	30.62	30.74	32.07	30.31	**32.14**
UCA	0.966	0.965	0.964	0.967	**0.967**

6) Multiscale Spatial Rearrangement with Other Window Sizes: To demonstrate the generalization ability of multiscale spatial rearrangement, we conduct several experiments on the MSSR-MLP block with other window sizes. Table XV reports that applying multiscale spatial rearrangement with a 6×6 window or a 10×10 window can also yield improvements over single-scale windows. In multiscale comparisons, multiscale spatial rearrangement processes with other window sizes can also perform better than the MSW-MLP with fewer parameters and FLOPs.

F. Discussion

1) Performance Comparison With Additional Metrics: In our study, we mainly use the traditional PSNR and SSIM as image quality assessment (IQA) metrics for the paired datasets. However, existing research indicates that the PSNR and SSIM may not be sufficient for effectively measuring visual quality [93], [94], [95]. We introduce additional quantitative IQA metrics, including the BPRI [96], BMPRI [97] and UCA [95], to further assess the achieved restoration performance. The BPRI and BMPRI can sensitively reflect content sharpness/noisiness without using reference images [96]. The UCA [95] is designed to focus on the human perceptual characteristic variations in different contents by adaptive multiscale weighting. Therefore, they are suitable for evaluating the quality of recovered images [97]. The higher the BPRI, BMPRI and UCA are, the clearer the corresponding image. The overall results are shown in Table XVI and Table XVII. Notably, the MSSR-MLP outperforms the competing models, which again demonstrates the effectiveness of our approach.

2) Effectiveness of the MSSR-MLP With Visual Attention: Attention in human perception usually refers to the ability of the human visual system to adaptively process visual information and focus on regions of interest [98], [99], [100]. For nonuniform degraded images, visual attention can locate the degradation regions and their magnitudes, and embedding visual attention into the model may improve the its performance.

To explore whether incorporating visual attention can further improve the proposed method, we plug a well-known

TABLE XVIII
THE RESULTS OF AN ABLATION STUDY CONDUCTED ON VISUAL ATTENTION

	PSNR	SSIM	Params	FLOPs
w/o visual attention	31.94	0.952	8.33	26.72
w visual attention	**32.03**	**0.953**	9.51	29.83

visual attention module used in image restoration tasks, the dual attention unit (DAU) [65], into our MSSR-MLP block and train it on MSSR-MLP-S for an ablation study. As shown in Table XVIII, DAU brings 0.09 dB and 0.001 improvements in terms of the PSNR and SSIM, respectively, which can demonstrate the effectiveness of visual attention embedding. However, the additional overheads cannot be ignored, as an extra 1.18 M parameters and 2.11 G additional FLOPs may break the satisfactory trade-off between the performance and efficiency of our proposed model. Therefore, we will consider designing a lightweight plug-and-play visual attention module in the future.

V. CONCLUSION

In this paper, we propose a single-stage multiscale spatial rearrangement MLP (MSSR-MLP) architecture for image restoration. The core component in the MSSR-MLP is the spatial rearrangement module (SRM), which can generate information at different scales in a local window by setting different step sizes. In addition, the SRM extends the local receptive field without introducing additional parameters and FLOPs. Accordingly, by using several SRMs with different step sizes, we design an MSSR-MLP block that can aggregate multiscale information at a low computational cost. Extensive experiments conducted on various benchmark datasets demonstrate that our MSSR-MLP achieves state-of-the-art performance in several image restoration tasks, including single-image motion deblurring, defocus deblurring, image deraining and image denoising, while requiring fewer FLOPs than the competing models.

REFERENCES

[1] Y. Li, J. Hu, G. Ni, and T. Zeng, "Deep CNN denoiser prior for blurred images restoration with multiplicative noise," *Inverse Problems Imag.*, vol. 17, no. 3, pp. 726–745, 2023.

[2] X. Tang, X. Zhao, J. Liu, J. Wang, Y. Miao, and T. Zeng, "Uncertainty-aware unsupervised image deblurring with deep residual prior," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 9883–9892.

[3] V. Lempitsky, A. Vedaldi, and D. Ulyanov, "Deep image prior," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9446–9454.

[4] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-Laplacian priors," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1033–1041.

[5] J. Pan et al., "Dual convolutional neural networks for low-level vision," *Int. J. Comput. Vis.*, vol. 130, no. 6, pp. 1440–1458, Jun. 2022.

[6] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 7, pp. 2480–2495, Jul. 2021.

[7] J. Pan et al., "Physics-based generative adversarial models for image restoration and beyond," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 7, pp. 2449–2462, Jul. 2021.

[8] P. Zhang et al., "Multi-scale vision longformer: A new vision transformer for high-resolution image encoding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2978–2988.

[9] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 257–265.

[10] J. Li, F. Fang, K. Mei, and G. Zhang, "Multi-scale residual network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 517–532.

[11] S.-J. Cho, S.-W. Ji, J.-P. Hong, S.-W. Jung, and S.-J. Ko, "Rethinking coarse-to-fine approach in single image deblurring," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4621–4630.

[12] A. Abdelhamed, S. Lin, and M. S. Brown, "A high-quality denoising dataset for smartphone cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1692–1700.

[13] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M. Yang, "Restormer: Efficient transformer for high-resolution image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5718–5729.

[14] H. Chen et al., "Pre-trained image processing transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12294–12305.

[15] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "SwinIR: Image restoration using Swin transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 1833–1844.

[16] Z. Wang, X. Cun, J. Bao, W. Zhou, J. Liu, and H. Li, "Uformer: A general U-shaped transformer for image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 17683–17693.

[17] I. Tolstikhin et al., "MLP-mixer: An all-MLP architecture for vision," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 24261–24272.

[18] H. Liu, Z. Dai, D. So, and Q. V. Le, "Pay attention to MLPs," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 9204–9215.

[19] D. Lian, Z. Yu, X. Sun, and S. Gao, "As-MLP: An axial shifted MLP architecture for vision," 2021, *arXiv:2107.08391*.

[20] T. Yu, X. Li, Y. Cai, M. Sun, and P. Li, "S^2-MLP: Spatial-shift MLP architecture for vision," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 3615–3624.

[21] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[22] Z. Tu et al., "MAXIM: Multi-axis MLP for image processing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5759–5770.

[23] S. W. Zamir et al., "Learning enriched features for fast image restoration and enhancement," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 1934–1948, Feb. 2023.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[25] L. Chen, X. Lu, J. Zhang, X. Chu, and C. Chen, "HINet: Half instance normalization network for image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 182–192.

[26] S. W. Zamir et al., "Multi-stage progressive image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14816–14826.

[27] X. Hua, M. Li, J. Fei, J. Liu, Y. Shi, and H. Hong, "Dynamic scene deblurring with continuous cross-layer attention transmission," *Pattern Recognit.*, vol. 143, Nov. 2023, Art. no. 109719.

[28] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.

[29] M. Suin, K. Purohit, and A. N. Rajagopalan, "Spatially-attentive patch-hierarchical network for adaptive motion deblurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3603–3612.

[30] S. Cheng, Y. Wang, H. Huang, D. Liu, H. Fan, and S. Liu, "NBNet: Noise basis learning for image denoising with subspace projection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4894–4904.

[31] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[32] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[33] H. Touvron et al., "ResMLP: Feedforward networks for image classification with data-efficient training," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 5314–5321, Apr. 2023.

[34] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[35] H. Duan et al., "Masked autoencoders as image processors," 2023, *arXiv:2303.17316*.

[36] H. Lee, H. Choi, K. Sohn, and D. Min, "KNN local attention for image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 2129–2139.

[37] S. Chen, E. Xie, C. Ge, D. Liang, and P. Luo, "CycleMLP: A MLP-like architecture for dense prediction," 2021, *arXiv:2107.10224*.

[38] Y. Ma, F. Lin, S. Wu, S. Tian, and L. Yu, "PRSeg: A lightweight patch rotate MLP decoder for semantic segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 11, pp. 6860–6871, Nov. 2023.

[39] K. Zhang et al., "Deblurring by realistic blurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2734–2743.

[40] L. Ruan, B. Chen, J. Li, and M. Lam, "Learning to deblur using light field generated and real defocus images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 16283–16292.

[41] J. Lee, H. Son, J. Rim, S. Cho, and S. Lee, "Iterative filter adaptive network for single image defocus deblurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2034–2042.

[42] A. Abuolaim and M. S. Brown, "Defocus deblurring using dual-pixel data," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 111–126.

[43] L. Chen, X. Chu, X. Zhang, and J. Sun, "Simple baselines for image restoration," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 17–33.

[44] H. Wu et al., "CvT: Introducing convolutions to vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 22–31.

[45] T. Huang, S. Li, X. Jia, H. Lu, and J. Liu, "Neighbor2Neighbor: Self-supervised denoising from single noisy images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14776–14785.

[46] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, and R. W. H. Lau, "Spatial attentive single-image deraining with a high quality real rain dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12262–12271.

[47] H. Wang, Q. Xie, Q. Zhao, and D. Meng, "A model-driven deep neural network for single image rain removal," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3100–3109.

[48] W. Wei, D. Meng, Q. Zhao, Z. Xu, and Y. Wu, "Semi-supervised transfer learning for image rain removal," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3872–3881.

[49] Y. Li, Y. Monno, and M. Okutomi, "Single image deraining network with rain embedding consistency and layered LSTM," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 3957–3966.

[50] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, "Progressive image deraining networks: A better and simpler baseline," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3932–3941.

[51] Z. Shen et al., "Human-aware motion deblurring," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5571–5580.

[52] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[53] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis.*, 2001, pp. 416–423.

[54] B. Li et al., "Benchmarking single-image dehazing and beyond," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 492–505, Jan. 2019.

[55] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.

[56] X. Min et al., "Quality evaluation of image dehazing methods using synthetic hazy images," *IEEE Trans. Multimedia*, vol. 21, no. 9, pp. 2319–2333, Sep. 2019.

[57] X. Min, G. Zhai, K. Gu, X. Yang, and X. Guan, "Objective quality evaluation of dehazed images," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 2879–2892, Aug. 2019.

[58] A. Paszke et al., "Automatic differentiation in PyTorch," in *Proc. Adv. Neural Inf. Process. Syst. Workshops*, 2017, pp. 1–4.

[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[60] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," 2016, *arXiv:1608.03983*.

[61] C. Mou, J. Zhang, and Z. Wu, "Dynamic attentive graph learning for image restoration," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4308–4317.

[62] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, "Multi-level wavelet-CNN for image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1–12.

[63] H. Zhao, Y. Gou, B. Li, D. Peng, J. Lv, and X. Peng, "Comprehensive and delicate: An efficient transformer for image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 14122–14132.

[64] S. W. Zamir et al., "CycleISP: Real image restoration via improved data synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2693–2702.

[65] S. W. Zamir et al., "Learning enriched features for real image restoration and enhancement," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 492–511.

[66] T. Plötz and S. Roth, "Benchmarking denoising algorithms with real photographs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2750–2759.

[67] Z. Yue, Q. Zhao, L. Zhang, and D. Meng, "Dual adversarial network: Toward real-world noise removal and noise generation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 41–58.

[68] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1122–1131.

[69] X. Wang, L. Xie, C. Dong, and Y. Shan, "Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 1905–1914.

[70] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[71] K. Ma et al., "Waterloo exploration database: New challenges for image quality assessment models," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 1004–1016, Feb. 2017.

[72] Y. Zheng, J. Zhan, S. He, J. Dong, and Y. Du, "Curricular contrastive regularization for physics-aware single image dehazing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 5785–5794.

[73] H. Zhang, Y. Dai, H. Li, and P. Koniusz, "Deep stacked hierarchical multi-patch network for image deblurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5971–5979.

[74] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, "Scale-recurrent network for deep image deblurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8174–8182.

[75] H. Gao, X. Tao, X. Shen, and J. Jia, "Dynamic scene deblurring with parameter selective sharing and nested skip connections," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3843–3851.

[76] A. Abuolaim, M. Delbracio, D. Kelly, M. S. Brown, and P. Milanfar, "Learning to reduce defocus blur by realistically modeling dual-pixel data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2269–2278.

[77] X. Chu, L. Chen, C. Chen, and X. Lu, "Improving image restoration by revisiting global information aggregation," in *Proc. 17th Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 53–71.

[78] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha, "Recurrent squeeze-and-excitation context aggregation net for single image deraining," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 254–269.

[79] Q. Yi, J. Li, Q. Dai, F. Fang, G. Zhang, and T. Zeng, "Structure-preserving deraining with residue channel prior guidance," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4218–4227.

[80] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2808–2817.

[81] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.

[82] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang, "Non-local recurrent network for image restoration," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1680–1689.

[83] Z. Zha, X. Yuan, B. Wen, J. Zhou, J. Zhang, and C. Zhu, "From rank estimation to rank approximation: Rank residual constraint for image restoration," *IEEE Trans. Image Process.*, vol. 29, pp. 3254–3269, 2020.
[84] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, "Plug-and-play image restoration with deep denoiser prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6360–6376, Oct. 2022.
[85] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.
[86] S. Anwar and N. Barnes, "Real image denoising with feature attention," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3155–3164.
[87] Z. Yue, H. Yong, Q. Zhao, D. Meng, and L. Zhang, "Variational denoising network: Toward blind noise modeling and removal," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1690–1701.
[88] X. Qin, Z. Wang, Y. Bai, X. Xie, and H. Jia, "FFA-Net: Feature fusion attention network for single image dehazing," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 11908–11915.
[89] C. Guo, Q. Yan, S. Anwar, R. Cong, W. Ren, and C. Li, "Image dehazing transformer with transmission-aware 3D position embedding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5802–5810.
[90] J. Dong and J. Pan, "Physics-based feature dehazing networks," in *Proc. 16th Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 188–204.
[91] H. Dong et al., "Multi-scale boosted dehazing network with dense feature fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2154–2164.
[92] T. Ye et al., "Perceiving and modeling density is all you need for image dehazing," 2021, *arXiv:2111.09733*.
[93] X. Min et al., "Screen content quality assessment: Overview, benchmark, and beyond," *ACM Comput. Surv.*, vol. 54, no. 9, pp. 1–36, Dec. 2022.
[94] G. Zhai and X. Min, "Perceptual image quality assessment: A survey," *Sci. China Inf. Sci.*, vol. 63, no. 11, Nov. 2020.
[95] X. Min, K. Ma, K. Gu, G. Zhai, Z. Wang, and W. Lin, "Unified blind quality assessment of compressed natural, graphic, and screen content images," *IEEE Trans. Image Process.*, vol. 26, no. 11, pp. 5462–5474, Nov. 2017.
[96] X. Min, K. Gu, G. Zhai, J. Liu, X. Yang, and C. W. Chen, "Blind quality assessment based on pseudo-reference image," *IEEE Trans. Multimedia*, vol. 20, no. 8, pp. 2049–2062, Aug. 2018.
[97] X. Min, G. Zhai, K. Gu, Y. Liu, and X. Yang, "Blind image quality estimation via distortion aggravation," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 508–517, Jun. 2018.
[98] X. Min, G. Zhai, J. Zhou, X.-P. Zhang, X. Yang, and X. Guan, "A multimodal saliency model for videos with high audio-visual correspondence," *IEEE Trans. Image Process.*, vol. 29, pp. 3805–3819, 2020.
[99] X. Min, G. Zhai, J. Zhou, M. C. Q. Farias, and A. C. Bovik, "Study of subjective and objective quality assessment of audio-visual signals," *IEEE Trans. Image Process.*, vol. 29, pp. 6054–6068, 2020.
[100] X. Min, G. Zhai, K. Gu, and X. Yang, "Fixation prediction through multimodal analysis," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 13, no. 1, pp. 1–23, Feb. 2017.

Xia Hua received the Ph.D. degree in pattern recognition and intelligent systems from the Huazhong University of Science and Technology (HUST), China, in 2014. He is currently an Associate Professor with the School of Electrical and Information Engineering, Wuhan Institute of Technology, Hubei, China. His current research interests include image restoration, pattern recognition, and fast algorithm of digital signal processing.

Zezheng Li received the B.S. degree from the School of Electrical Engineering and Information, Wuhan Institute of Technology, Wuhan, China, in 2021, where he is currently pursuing the master's degree. His current research interests include deep learning, image restoration, and image processing.

Hanyu Hong received the Ph.D. degree from the Institute for Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan, China, in 2004. He is currently a Professor and the Dean of the School of Electrical and Information Engineering, Wuhan Institute of Technology, Wuhan. His current research interests include image analysis, pattern recognition, image reconstruction, pattern recognition, and computer graphics.