Comprehensive and Delicate: An Efficient Transformer for Image Restoration

Haiyu Zhao*, Yuanbiao Gou*, Boyun Li, Dezhong Peng, Jiancheng Lv, Xi Peng†
College of Computer Science, Sichuan University.
{haiyuzhao.gm, gouyuanbiao, liboyun.gm, pengx.gm}@gmail.com;
{pengdz, lvjiancheng}@scu.edu.cn

Abstract

Vision Transformers have shown promising performance in image restoration, which usually conduct window- or channel-based attention to avoid intensive computations. Although the promising performance has been achieved, they go against the biggest success factor of Transformers to a certain extent by capturing the local instead of global dependency among pixels. In this paper, we propose a novel efficient image restoration Transformer that first captures the superpixel-wise global dependency, and then transfers it into each pixel. Such a coarse-to-fine paradigm is implemented through two neural blocks, i.e., condensed attention neural block (CA) and dual adaptive neural block (DA). In brief, CA employs feature aggregation, attention computation, and feature recovery to efficiently capture the global dependency at the superpixel level. To embrace the pixel-wise global dependency, DA takes a novel dual-way structure to adaptively encapsulate the globality from superpixels into pixels. Thanks to the two neural blocks, our method achieves comparable performance while taking only ∼6% FLOPs compared with SwinIR.

1. Introduction

Image restoration aims to recover the high-quality image from its degraded version, and huge success has been achieved by plentiful methods in the past years [22, 30, 35, 54–56, 60]. In the era of deep learning, Convolutional Neural Networks (CNNs) have shown promising performance in image restoration [25, 52, 67] thanks to the inductive biases of weight sharing and spatial locality [12]. However, although a number of studies have shown the effectiveness of CNNs, it has suffered from the following limitations [12], *i.e.*, i) non-dynamic weights of CNNs limit the model capacity of instance adaption, and ii) the sparse connections of CNNs limit the capture of global dependency.

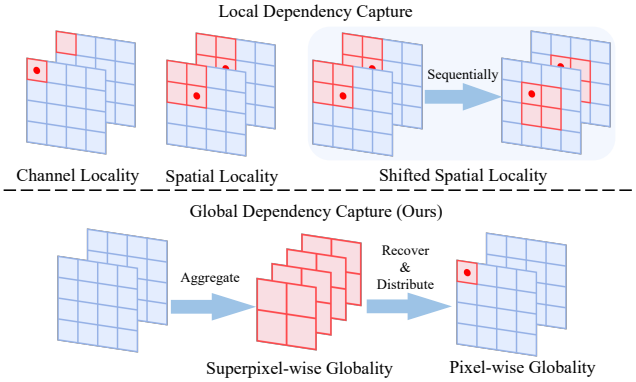

Figure 1. Illustration of the dependency capture in existing vision Transformers and ours. The red boxes refer to the dependency capture range of a given pixel marked by a red point. We generally summarize the dependency capture in existing vision Transformers as the three ways above the dashed line. Obviously, they could only capture the dependency in a local range. In contrast, our method could obtain a pixel-wise global dependency through superpixel-wise dependency computation and distribution.

To overcome these limitations, some solutions [51, 60, 67, 68] have been specifically established, of which Transformer-based methods [3, 5, 28, 47, 53] have achieved huge success thanks to their high capacities of dynamic weighting and global dependency capturing. Towards the image-restoration-specific Transformers, the biggest roadblock might be the unacceptable cost caused by the global attention computation. Therefore, some efficient attentions have been proposed to trade-off the efficiency and dependency range, *e.g.*, local window attention [49], shifted window attention [22], and channel attention [56]. Although the promising performance has been achieved, these Transformers have still suffered from the following limitations. First, the computation costs are still very high, thus limiting their applications in the mobile scenarios. Second, the attention mechanisms could only capture the dependency from a given range, and thus such a locality might not fully explore the potential of Transformer.

*indicates equal contribution.
†Corresponding author.

In practice, it is daunting to develop an efficient image restoration Transformer while directly capturing the global dependency, since it necessarily introduces intensive computations, which goes against efficiency. Different from existing studies focusing on efficient attention mechanisms, this paper resolves the contradiction from a novel perspective. To be specific, our basic idea is to adaptively aggregate the features at pixel level into a lower dimensional space of superpixel to remove the redundancy in channel [13] and space [50] domains. Through the feature aggregation, the dimension is remarkably reduced, and thus the attention could be computed in a global way with acceptable computation costs. After that, the feature recovery is performed to restore the feature distribution in channel and space domains. Although the above paradigm shows a feasible solution, it is far from the final goal since the obtained dependency actually works at the superpixel level. Hence, we need to transfer such a superpixel-wise global dependency to a pixel-wise global dependency. As a result, pixel-wise restoration can depend on the global information from superpixels.

Based on the above motivations, we propose an efficient Transformer for COmprehensive and DElicate image restoration (CODE). To be specific, CODE consists of a condensed attention neural block (CA) and a dual adaptive neural block (DA). In brief, CA captures the global dependency at the superpixel level with acceptable computations thanks to the aforementioned paradigm. To obtain the pixel-wise global dependency, DA extracts the globality from the CA output and then dynamically encapsulates it into each pixel. Thanks to the two complementary neural blocks, CODE could capture the pixel-wise global dependency, while embracing high computational efficiency.

The contributions and novelty of this work could be summarized as below.

- Unlike existing efficient Transformers only capture the pixel-wise local dependency, our solution could obtain the pixel-wise global dependency through superpixel-wise dependency computation and transformation.
- The proposed image restoration Transformer (CODE) consists of two neural blocks. In brief, CA employs feature aggregation, attention computation, and feature recovery to efficiently capture the superpixel-wise global dependency. To obtain pixel-wise global dependency, DA takes a novel dual-way structure and a dynamic weighting fashion to distribute the superpixel-wise globality into each pixel.
- Extensive experiments are conducted on four image restoration tasks to demonstrate the efficiency and effectiveness of CODE, *e.g.*, it achieves comparable performance while taking only $\sim$6% FLOPs compared with SwinIR [22].

2. Related Work

In this section, we first introduce the related works in image restoration and vision Transformers, and then elaborate on the differences between CODE and existing methods.

2.1. Image Restoration

At present, deep image restoration methods could be divided into two categories according to the architectures, *i.e.*, CNN- and Transformer-based methods. Here, we introduce the former while detailing the latter in the Sec. 2.2.

In the past decades, CNN-based methods have achieved promising performance thanks to the introduction of the inductive biases, *e.g.*, weight sharing and spatial locality. As a result, plentiful CNN-based methods have been proposed [10, 11, 19, 27, 32, 39, 48, 57, 60, 69], and advanced the image restoration to a new stage in both efficiency and effectiveness. For example, BSRN [21] proposed an efficient and effective neural network by introducing blueprint separable convolutions and spatial-channel attention modules. MPRNet [57] proposed an effective multi-stage architecture for image restoration by progressively recovering the degraded images. Although CNNs have shown impressive performance, they still suffer from the limitations of i) static (non-dynamic) weights for every instance, and ii) sparse connection for every output. As a result, CNNs are short in instance adaptation ability and global dependency capturing. To alleviate these problems, some ingenious designs are introduced. For example, CBAM [51] inferred attention maps along channel and spatial dimensions for adaptive refinement. IRCNN [63] used dilated convolutions to enlarge the receptive field, thus capturing more contextual information. RDN [67] designed a very deep network to capture rich hierarchical features. Although these methods have shown effectiveness, more studies are expected so that the aforementioned limitations could be further overcome.

2.2. Vision Transformers

Vision Transformers [5, 20, 28] have achieved remarkable performance due to the attention mechanism, which naturally embraces the powerful dynamic weights and global dependency capturing. However, as each coin has two sides, the intensive computations in attention limit the application to vision tasks, in which the images are generally with a high dimensionality. Therefore, some vision Transformers decompose the images into small patches and take the sequence of patches as input, thus alleviating computational costs. IPT [3] could be the first work that introduces this strategy into image restoration. Afterward, some methods employed window-based attention, which performs attention on a local window instead of the global feature map. For example, Uformer [49] introduced the locally-enhanced window Transformer block that performs non-overlapping

window-based attention for image restoration. SwinIR [22] used the local and shifted window scheme to sequentially perform the within- and cross-window attention for image restoration. In addition, to avoid intensive computations in the space domain, Restormer [56] employed the channel attention to substitute the original spatial attention, so that the attention is performed in a low dimension. Although these methods have achieved both effectiveness and efficiency, they restrict the attention within a local range, which may not fully explore the potential of Transformers in capturing global dependency. Meanwhile, their computations still are non-trivial, especially in mobile scenarios, *e.g.*, SwinIR [22] involves ∼373G FLOPs on 128×128 images.

Overall, the differences between our CODE and existing methods could be summarized as follows. On the one hand, we design an efficient image restoration Transformer CODE to obtain the pixel-wise global dependency, through superpixel-wise dependency computation and transformation. On the other hand, our CODE consists of two neural blocks, *i.e.*, CA and DA, which capture the superpixel-wise global dependency and distribute the globality in superpixels into pixels in an efficient way, respectively.

3. Method

In this section, we first introduce the overall architecture and then elaborate on the two neural blocks, *i.e.*, condensed attention(CA) and dual adaptive neural block (DA).

3.1. Overall Architecture

As shown in the Fig. 2, our network is the hierarchical multi-scale architecture, which enjoys the advantage of fewer computation costs than non-hierarchical or single-scale ones. For a given degraded image, we first use a 3×3 convolution to extract the shallow features F_0 and then employ an encoder-decoder with four scales to extract the deep feature F_0'. Each scale in the encoder-decoder consists of multiple Transformer blocks and each Transformer block is the sequential combination of CA and DA.

At the beginning of each scale in the encoder (except for the first scale), we reduce the feature resolution to half while expanding the feature channel to double (denoted as $\downarrow$) and then extract the deep feature through multiple Transformer blocks $T_i(\cdot)$, *i.e.*,

$$F_i = \begin{cases} T_i(F_0), & i = 1, \\ T_i(F_{i-1}\downarrow), & i = 2, 3, 4, \end{cases} \tag{1}$$

where F_{i-1} and F_i are the input and output features of the i-th scale in the encoder. As for the decoder, we double the feature resolution while halving the feature channel (denoted as $\uparrow$) at the beginning of each scale (except for the last scale). Meanwhile, a skip connection of the corresponding scale from the encoder is introduced to fuse the hierarchical multi-scale features, and follows multiple Transformer blocks to refine them, *i.e.*,

$$F'_{i-1} = \begin{cases} T'_i(F_i), & i = 4, \\ T'_i(F'_i\uparrow + F_i), & i = 3, 2, 1, \end{cases} \tag{2}$$

where F'_i and F'_{i-1} are the input and output features of i-th scale in the decoder, and F_i is the feature from the encoder, which is the same scale as F'_i.

After the encoder-decoder, we fuse the deep feature F'_0 with the shallow feature F_0, and refine them through several Transformer blocks T''_r to obtain the final feature F_r, *i.e.*,

$$F_r = T''_r(F'_0 + F_0). \tag{3}$$

Finally, we use a 3×3 convolution to fuse F_r as a residual image, which would be added to the degraded image for obtaining the restored image.

3.2. Condensed Attention Neural Block

To capture the global dependency at the superpixel level, we propose a three-step paradigm of feature aggregation, attention computation, and feature recovery, which is implemented by our CA. As shown in Fig. 2, CA first aggregates the channel and spatial features into the condensed ones, *i.e.*, superpixel features. Then, as the features involve two dimensions of channel and space, CA sequentially conducts the channel and spatial attentions on them, so that the global dependency could be fully captured along the two dimensions. Finally, CA recovers the space and channel dimensions so that the output superpixel features' resolution and channels are consistent with the input pixel features. In this section, we first introduce the feature aggregation and recovery, and then detail the channel and spatial attentions.

Feature Aggregation and Recovery. Motivated by the observation that there exists a lot of redundant features in both the channel and space domains, we reduce them to obtain the superpixel features before attention computation, while recovering them after that to achieve better efficiency. However, the biggest challenge of this paradigm is how to properly reduce the redundant features while retaining the informative ones as far as possible and recover the feature distribution in the channel and space domains after attention computation. To achieve this end, CA performs the feature aggregation and recovery in an adaptive fashion, *i.e.*, adaptively learning to aggregate the informative features, and recover the feature distribution through the network.

Given the input features $F \in \mathbb{R}^{H \times W \times C}$, where $H \times W$ denotes the spatial resolution and C is the number of channels, CA aggregates the features along the channel dimension to obtain the channel-condensed features $\widetilde{F} \in \mathbb{R}^{H \times W \times C'}$, *i.e.*,

$$\widetilde{F} = \Phi^{CA}(F), \tag{4}$$

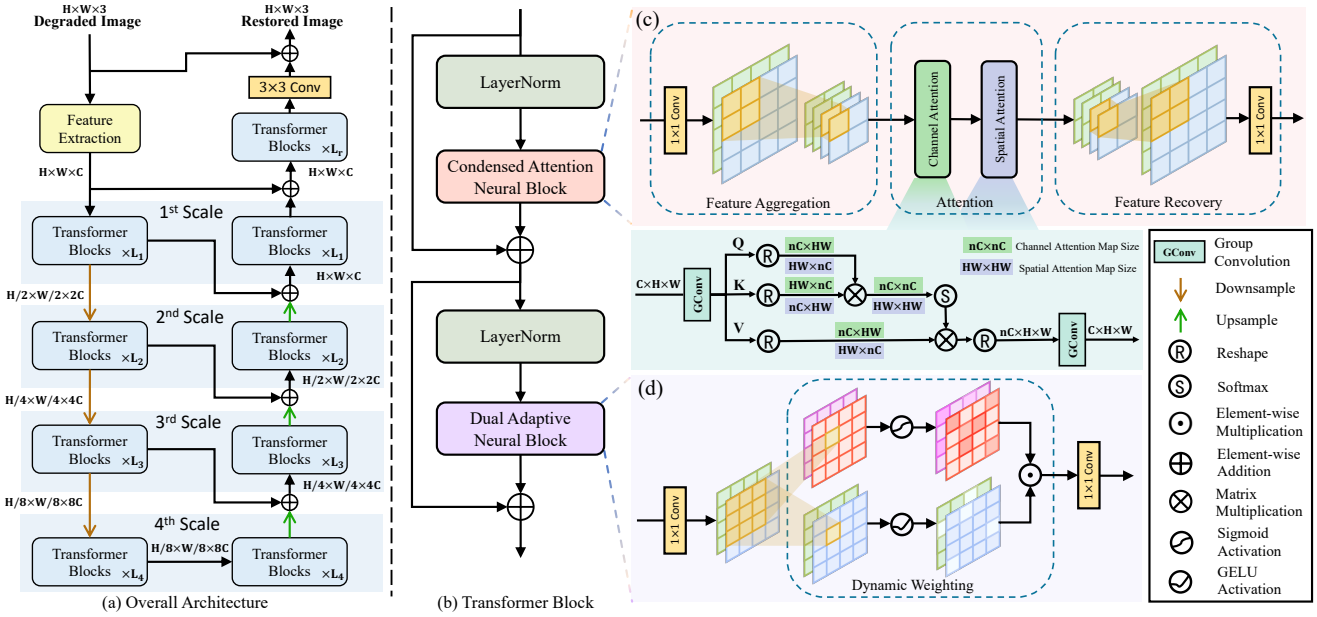

Figure 2. The overall architecture of our proposed efficient Transformer for image restoration. The key components are shown in (b) Transformer block which consists of (c) condensed attention neural block (CA) and (d) dual adaptive neural block (DA).

where $C' = C/r_c$ and r_c is the aggregation factor. For generality, CA introduces a point-wise convolution as $\Phi^{CA}(\cdot)$ to adaptively aggregate the informative features in channel domain. Afterward, CA aggregates the features along the spatial dimensions to obtain the space-condensed features $\hat{F} \in \mathbb{R}^{H' \times W' \times C''}$, *i.e.*,

$$\hat{F} = \Psi^{SA}(\widetilde{F}), \tag{5}$$

where $H' = H/S$, $W' = W/S$ and $C'' = C' r_s$. In brief, $\Psi^{SA}(\cdot)$ aggregates the spatial features with the patch size of $S \times S \times 1$ into the condensed features with that of $1 \times 1 \times r_s$, thus obtaining the $\hat{F}$ of size $H/S \times W/S \times C' r_s$. For generality, CA introduces a channel-wise group convolution, whose input and output channels are C' and C'', kennel size is $S \times S$, and stride is S, to implement it. In this way, the spatial features could be aggregated while being sufficiently preserved in the expanded channels. As a result, the superpixel features in the channel and space domains could be obtained, and used to capture the superpixel-wise dependency in a lower dimensional space.

To recover the feature distribution in channel and space domains after attention, CA employs the reverse process of the aggregation. To be specific, it recovers the spatial features first and then the channel features, *i.e.*,

$$\overline{F} = \Phi^{CR}(\Psi^{SR}(\Theta(\hat{F}))), \tag{6}$$

where $\Theta(\hat{F})$ is the features after attention calculation, $\Psi^{SR}(\cdot)$ and $\Phi^{CR}(\cdot)$ are the recovery functions of spatial and channel features, respectively. To recover the spatial features from size $H' \times W' \times C''$ to $H \times W \times C'$, $\Psi^{SR}(\cdot)$ employs a 1×1 channel-wise group convolution, whose input and output channels are C'' and $C'S^2$, followed by a pixel-shuffle [42] to recover the spatial resolution of $H \times W$. To recover the feature channels, $\Phi^{CR}(\cdot)$ uses a point-wise convolution with the input and output channel of C' and C. As a result, the final features $\overline{F}$ could be obtained while keeping the spatial resolution and the channel number identical to the input features F.

Channel and Spatial Attention. To fully capture the superpixel-wise global dependency along the two dimensions, we sequentially perform the channel and spatial attentions on the superpixel features $\hat{F} \in \mathbb{R}^{H' \times W' \times C''}$. In brief, channel attention captures the dependency along the channel dimension, which is followed by the spatial attention to capture the dependency along the space dimension. As a result, the global dependency could be fully captured in both the channel and space domains.

To improve the efficiency of our attention block, we introduce a novel channel-wise slice and merge mechanism before and after the multi-head attention calculation, respectively, and the overall pipeline could be formulated as,

$$\{X_i^j\}_{j=1}^{3n} = \Phi_i^{proj}(\hat{F}_i), \quad i = 1, \cdots, C'', \tag{7}$$

$$\{\{Y_i^j\}_{j=1}^{n}\}_{i=1}^{C''} = \Theta(\{\{X_i^j\}_{j=1}^{3n}\}_{i=1}^{C''}), \tag{8}$$

$$Z_i = \Psi_i^{fuse}(\{Y_i^j\}_{j=1}^{n}), \quad i = 1, \cdots, C'', \tag{9}$$

where $\Phi_i^{proj}(\cdot)$ projects the ith channel feature $\hat{F}_i$ into $3n$ channel slices $\{X_i^j\}_{j=1}^{3n}$, and thus obtaining the over-

all channel features $\{\{X_i^j\}_{j=1}^{3n}\}_{i=1}^{C''}$, where the number of channels is $3n \times C''$. $\Psi_i^{fuse}(\cdot)$ merges the channel slices into one channel after attention computation. Both $\Phi_i^{proj}(\cdot)$ and $\Psi_i^{fuse}(\cdot)$ are implemented by group convolutions. $\Theta(\cdot)$ is the multi-head attention calculation, which first uniformly divides the sliced channels into the query (Q), key (K), value (V), *i.e.*,

$$\{\{Q_i^j\}_{j=1}^{n}, \{K_i^j\}_{j=1}^{n}, \{V_i^j\}_{j=1}^{n}\}_{i=1}^{C''} = \{\{X_i^j\}_{j=1}^{3n}\}_{i=1}^{C''}, \quad (10)$$

and then respectively shuffles them, so that the sliced channels from the same channel could be divided into different attention heads, *i.e.*,

$$\{\{Q_i^j\}_{i=1}^{C''}, \{K_i^j\}_{i=1}^{C''}, \{V_i^j\}_{i=1}^{C''}\}_{j=1}^{n} = \{\{Q_i^j\}_{j=1}^{n}, \{K_i^j\}_{j=1}^{n}, \{V_i^j\}_{j=1}^{n}\}_{i=1}^{C''}. \quad (11)$$

Next, we calculate the attention with the head number greater than or equal to n. At each head, we calculate the attention through the following formulation,

$$F = \text{Softmax}(QK^T/\alpha)V, \quad (12)$$

where α is the learnable scaling factor, and here we ignore the superscripts and subscripts for simplicity. Finally, we concatenate the results from multiple heads and unshuffle them to obtain the attention features $\{\{Y_i^j\}_{j=1}^{n}\}_{i=1}^{C''}$.

Both the channel and spatial attentions follow the above pipeline with the only difference in Eq. (12), where the former calculates attention along the channel dimension while the latter does that along the spatial dimension. Benefiting from the pipeline, the efficiency and effectiveness of our method could be further enhanced, as it expands more channel slices for attention calculation while introducing fewer parameters and computations.

3.3. Dual Adaptive Neural Block

Thanks to the three-step paradigm, CA has captured the superpixel-wise global dependency and each superpixel feature embraces rich global information. However, obtaining such a global dependency at the superpixel level is far from our final goal of the pixel-wise global dependency. As a result, pixel-wise restoration can depend on the global information from superpixels. To solve this problem, we introduce a dual adaptive neural block (DA) to encapsulate the superpixel-wise globality into the pixel-wise global dependency, through a novel dual-way structure and a dynamic weighting fashion. To be specific, DA first employs a point-wise convolution $P_{mix}(\cdot)$ to mix the superpixel features, *i.e.*,

$$\tilde{F} = P_{mix}(\overline{F}). \quad (13)$$

Then, as each superpixel contains global information, DA introduces a dual-way structure to capture the dependency of each pixel on superpixels from a short range, while extracting the pixel's features from superpixels' ones in a local region. After that, we let the former act on the latter in a dynamic weighting fashion, *i.e.*,

$$\{\tilde{X}_i^1, \tilde{X}_i^2\} = \Phi_i^{DW}(\tilde{F}_i), \quad (14)$$

$$\tilde{Y}_i = \sigma(\tilde{X}_i^1) \odot \phi(\tilde{X}_i^2) \quad (15)$$

where $i = 1, \cdots, C$ is the index of channels. In brief, Φ_i^{DW} extracts two channel slices $\{\tilde{X}_i^1, \tilde{X}_i^2\}$ from each channel $\tilde{F}_i$, which is implemented by a group convolution with the kernel size of 7×7. $\sigma(\cdot)$ and $\phi(\cdot)$ are Sigmoid and GELU activations, which convert the dependency to the weights and filter the extracted pixel features, respectively. Finally, CA employs a point-wise convolution $P_{fus}(\cdot)$ to refine the features with the globality, *i.e.*,

$$\tilde{Z} = P_{fus}(\tilde{Y}). \quad (16)$$

With the above designs, DA could capture the pixel-wise global dependency by transferring the global dependency as well as global features from the superpixels into each pixel, while only introducing a few computations and parameters.

4. Experiment

In this section, we evaluate CODE on four image restoration tasks, *i.e.*, grayscale and color image denoising, JPEG compression artifact reduction, and motion deblurring. In the following, we first introduce the experimental settings and then show the quantitative and qualitative results. Finally, we conduct analysis experiments to demonstrate the efficient and effective designs in CODE. Due to the space limitations, more experiments and analyses are presented in the supplementary material.

4.1. Experiment Settings

Implementation Details. We use the same settings for all experiments. To be specific, the channel numbers and Transformer blocks are [64, 128, 256, 512] and [4, 6, 6, 2] from the first scale to the fourth scale in the encoder-decoder, respectively. After that, four Transformer blocks are used to fuse and refine the deep and shallow features. In the aggregation of CA, r_c is 4, r_s and S respectively are [32, 16, 8, 4] and [16, 8, 4, 2] from the first to fourth scales. Besides, n is 4 in the attention calculations of CA.

Training details: The experiments are conducted in PyTorch [37] framework with NVIDIA GeForce RTX 3090 GPUs. For training, we use the Adam optimizer [16] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is initialized to $2e^{-4}$ and gradually decreased to $1e^{-6}$ through the cosine annealing strategy [29]. We train our model with the batch size of 16 and the patch size of 128 for 1000K iterations. Meanwhile, we adopt horizontal and vertical flips, and 90°, 180°, and 270° rotations for augmentation. In experiments, all FLOPs are computed on 128×128 images.

Table 1. Quantitative results of **grayscale image denoising** on benchmark datasets. The best and second-best results are colored by red and blue, respectively, and "Ours*" denotes our method with self-ensemble. σ refers to the noise level, of which a larger value denotes a higher noise level.

Method		DnCNN [62]	IRCNN [63]	FFDNet [64]	NLRN [24]	FOCNet [14]	RNAN [66]	MWCNN [26]	DRUNet [60]	SwinIR [22]	Ours	Ours*
#Params		0.56M	0.19M	0.49M	0.34M	-	8.96M	16.15M	32.64M	11.43M	12.18M	12.18M
FLOPs		18.22G	6.09G	3.98G	1382.08G	-	248.13G	28.95G	71.71G(**+319%**)	373.02G(**+1662%**)	22.44G	179.52G
Set12	$\sigma = 15$	32.86	32.76	32.75	33.16	33.07	-	33.15	33.25	33.36	33.33	33.36
	$\sigma = 25$	30.44	30.37	30.43	30.80	30.73	-	30.79	30.94	31.01	31.01	31.05
	$\sigma = 50$	27.18	27.12	27.32	27.64	27.68	27.70	27.74	27.90	27.91	27.93	27.97
BSD68	$\sigma = 15$	31.73	31.63	31.63	31.88	31.83	-	31.86	31.91	31.97	31.96	31.98
	$\sigma = 25$	29.23	29.15	29.19	29.41	29.38	-	29.41	29.48	29.50	29.51	29.53
	$\sigma = 50$	26.23	26.19	26.29	26.47	26.50	26.48	26.53	26.59	26.58	26.58	26.60

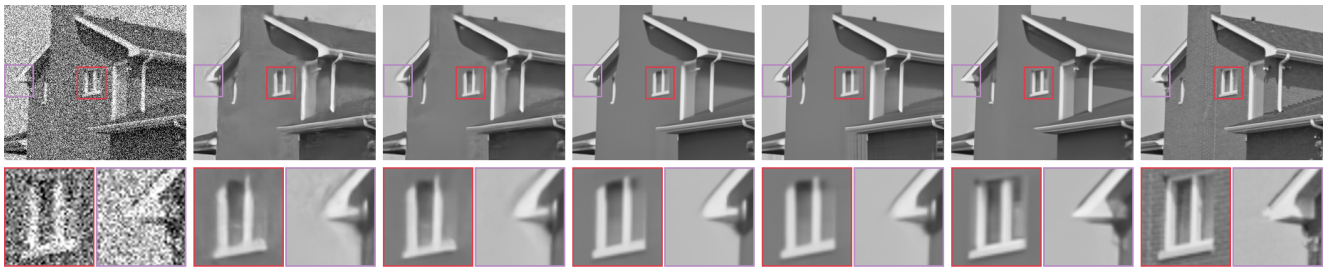

Figure 3. Qualitative comparisons of **grayscale image denoising** (noise level 50) on the image "*House*" from Set12.

4.2. Grayscale and Color Image Denoising

In this section, we evaluate our CODE on both the grayscale and color image denoising w.r.t. the additive white Gaussian noise level of $\sigma = 15, 25, 50$. To be specific, for color image denoising, we train CODE on the combination of DIV2K [1], Flickr2k [46], BSD400 [2], and WED [31], which respectively contain 800, 2650, 400 and 4774 images, and test it on CBSD68 [33], Kodak24 [8], and McMaster [65]. For grayscale image denoising, we train CODE on the grayscale version of the above training combination and test it on Set12 [62] and BSD68 [33].

We compare our CODE with 9 representative grayscale image denoising methods and 10 color image denoising methods, where six methods could be simultaneously evaluated on grayscale and color denoising, *i.e.*, DnCNN [62], IRCNN [63], FFDNet [64], RNAN [66], DRUNet [60] and SwinIR [22]. Besides, three methods for grayscale image denoising are NLRN [24], FOCNet [14] and MWCNN [26], and four methods for color image denoising are DSNet [38], BRDNet [45], RDN [67] and IPT [3].

Tab. 1 shows the quantitative results of grayscale image denoising. In addition to PSNR, we also show the parameters and FLOPs of our and compared methods for evaluating the complexities. From the Tab. 1, one could observe that our method achieves the best balance between efficiency and effectiveness. To be specific, our method embraces significantly fewer FLOPs or parameters while obtaining comparable performance compared with other state-of-the-art methods. For example, our method obtains competitive performance compared to SwinIR which has $\sim$17$\times$ FLOPs than ours, and better performance compared to DRUNet which has $\sim$3$\times$ parameters than ours. Moreover, our method is 0.02 and 0.01 higher in PSNR than SwinIR on Set12 and BSD68 when $\sigma = 50, 25$, respectively. To further investigate the potential of our method, we increase its FLOPs by introducing the self-ensemble strategy [23], since it increases 8$\times$ FLOPs without modifying our method. With this strategy, our method could obtain the best performance. Specifically, our method outperforms SwinIR with 0.04dB in PSNR, while still being less than half of its FLOPs.

Tab. 2 shows the quantitative results of color image denoising. As shown in the table, our method obtains comparable even better performance while embracing much higher efficiency. For example, our method with self-ensemble outperforms SwinIR with 0.01dB$\sim$0.07dB in PSNR on Kodak24 while being less than half of its FLOPs, and outperforms DRUNet with at most 0.05dB, 0.04dB, and 0.03dB in PSNR on CBSD68, Kodak24, and McMaster, respectively, while having nearly a third of its parameters. Note that although our method has inferior performance than SwinIR on CBSD68 and McMaster, it acquires the second-best performance with the highest efficiency.

Fig. 3 and Fig. 4 respectively show the qualitative results on grayscale and color image denoising. From the figures, one could observe that DnCNN and FFDNet have residual noises and artifacts, DRUNet and SwinIR obtain the over-smoothed and distorted results. In contrast, our method recovers the structures better and details finer, thus obtaining clearer restorations.

4.3. Motion Deblurring

To evaluate our method on motion deblurring, we train it on GoPro dataset [34] which consists of 2103 clean and

Table 2. Quantitative results of **color image denoising** on benchmark datasets. The best and second-best results are colored by red and blue, respectively, and "Ours*" denotes our method with self-ensemble. σ refers to the noise level, of which a larger value denotes a higher noise level.

Method		DnCNN [62]	IRCNN [63]	FFDNet [64]	DSNet [38]	BRDNet [45]	RNAN [66]	RDN [67]	IPT [3]	DRUNet [60]	SwinIR [22]	Ours	Ours*
#Params		0.56M	0.19M	0.85M	-	-	8.96M	22.12M	67.17M	32.64M	11.43M	12.18M	12.18M
FLOPs		18.30G	6.17G	7.00G	-	-	248.21G	725.11G	248.40G	71.79G(**+319%**)	373.02G(**+1656%**)	22.52G	180.16G
CBSD68	$\sigma = 15$	33.90	33.86	33.87	33.91	34.10	-	-	-	34.30	34.42	34.33	34.35
	$\sigma = 25$	31.24	31.16	31.21	31.28	31.43	-	-	-	31.69	31.78	31.69	31.71
	$\sigma = 50$	27.95	27.86	27.96	28.05	28.16	28.27	28.31	28.39	28.51	28.56	28.47	28.50
Kodak24	$\sigma = 15$	34.60	34.69	34.63	34.63	34.88	-	-	-	35.31	35.34	35.32	35.35
	$\sigma = 25$	32.14	32.18	32.13	32.16	32.41	-	-	-	32.89	32.89	32.88	32.91
	$\sigma = 50$	28.95	28.93	28.98	29.05	29.22	29.58	29.66	29.64	29.86	29.79	29.82	29.86
McMaster	$\sigma = 15$	33.45	34.58	34.66	34.67	35.08	-	-	-	35.40	35.61	35.38	35.43
	$\sigma = 25$	31.52	32.18	32.35	32.40	32.75	-	-	-	33.14	33.20	33.11	33.16
	$\sigma = 50$	28.62	28.91	29.18	29.28	29.52	29.72	-	29.98	30.08	30.22	30.03	30.08

Figure 4. Qualitative comparisons of **color image denoising** (noise level 50) on the image "*351093*" from CBSD68.

Table 3. Quantitative results of **motion deblurring** on the benchmark datasets of GoPro and HIDE. The best and second-best results are colored by red and blue, respectively.

Method	#Params	FLOPs	GoPro	HIDE
DGAN [17]	-	16.96G	28.70	24.51
DeepDeblur [34]	303.60M	44.00G	29.08	25.73
RNNDeblur [59]	-	-	29.19	-
DGANv2 [18]	7.83M	10.28G	29.55	26.61
SRN [44]	3.76M	35.87G	30.26	28.36
HAMD [41]	-	-	-	28.89
DSD [9]	2.84M	-	30.90	29.11
DBGAN [61]	11.59M	379.92G	31.10	28.94
MT-RNN [36]	2.64M	13.72G	31.15	29.15
DMPHN [58]	86.80M	-	31.20	29.09
EBMD [15]	-	-	31.79	-
SAPHNet [43]	23.00M	-	31.85	29.98
Ours	12.18M	22.52G	31.94	29.67

blurry image pairs. For evaluation, we employ two widely used datasets, *i.e.*, GoPro testset and HIDE [41], which consist of 1111 and 2025 clean and blurry image pairs with the size of 1280 $\times$ 720, respectively. For comparisons, we choose 12 representative methods that have comparable complexity as our method, *i.e.*, DGAN [17], DeepDeblur [34], RNNDeblur [59], DGANv2 [18], SRN [44], HAMD [41], DSD [9], DBGAN [61], MT-RNN [36], DMPHN [58], EBMD [15], and SAPHNet [43].

Tab. 3 shows the quantitative results, from which one could observe that our method achieves comparable or even better performance while having fewer parameters and/or FLOPs. For instance, our network obtains the PSNR of 31.94dB and 29.67dB on GoPro and HIDE, which respectively are 0.74dB and 0.58dB higher in PSNR than DMPHN that has more than 7$\times$ ours parameters. Although our PSNR is lower than SAPHNet on HIDE, our method obtains better results on GoPro with only nearly half of its parameters, which could also demonstrate the efficiency and effectiveness of our method.

4.4. JPEG Compression Artifact Reduction

To evaluate our method on JPEG compression artifact reduction, we train it on the same training datasets as color image denoising, by applying JPEG compression algorithm to the images with the quality factor of 10, 20, 30, and 40. For evaluations, we employ two widely used datasets, *i.e.*, Classic5 [7] and LIVE1 [40], which consist of five classic grayscale images and 29 natural color images, respectively. For comparisons, seven representative methods are introduced including ARCNN [4], DnCNN-3 [62], QGAC [6], RNAN [66], RDN [67], DRUNet [60], and SwinIR [22]. Following conventions, we calculate the quantitative results of these methods on the Y channel of YCbCr color space.

The results are shown in Tab. 4, from which one could observe that our method obtains the second-best results in most cases, and achieves the best tradeoff between efficiency and effectiveness. For example, our method outperforms DRUNet with at most 0.02dB/0.0003 and 0dB/0.0003 in PSNR/SSIM on Classic5 and LIVE1, respectively, while having its $\sim$37% parameters and $\sim$31% FLOPs. Note that although our method obtains inferior results than SwinIR, our method achieves the second-best results through much fewer FLOPs and/or parameters than SOTAs, *i.e.*, SwinIR, DRUNet, and RDN.

Table 4. Quantitative results of **JPEG compression artifact reduction** on benchmark datasets. The best and second-best results are colored by red and blue, respectively. q refers to the compression level, of which a smaller value denotes a higher compression level.

Method		ARCNN [4]	DnCNN-3 [62]	QGAC [6]	RNAN [66]	RDN [67]	DRUNet [60]	SwinIR [22]	Ours
#Params		0.11M	0.56M	-	8.96M	22.12M	32.64M	11.43M	12.18M
FLOPs		3.49G	18.22G	-	248.13G	724.92G	71.71G(**+319%**)	373.02G(**+1656%**)	22.44G
Classic5	$q=10$	29.03/0.7929	29.40/0.8026	29.84/0.8370	29.96/0.8178	30.00/0.8188	30.16/0.8234	30.27/0.8249	30.13/0.8225
	$q=20$	31.15/0.8517	31.63/0.8610	31.98/0.8850	32.11/0.8693	32.15/0.8699	32.39/0.8734	32.52/0.8748	32.36/0.8731
	$q=30$	32.51/0.8806	32.91/0.8861	33.22/0.9070	33.38/0.8924	33.43/0.8930	33.59/0.8949	33.73/0.8961	33.61/0.8951
	$q=40$	33.32/0.8953	33.77/0.9003	-	34.27/0.9061	34.27/0.9061	34.41/0.9075	34.52/0.9082	34.43/0.9078
LIVE1	$q=10$	28.96/0.8076	29.19/0.8123	29.53/0.8400	29.63/0.8239	29.67/0.8247	29.79/0.8278	29.86/0.8287	29.79/0.8281
	$q=20$	31.29/0.8733	31.59/0.8802	31.86/0.9010	32.03/0.8877	32.07/0.8882	32.17/0.8899	32.25/0.8909	32.16/0.8901
	$q=30$	32.67/0.9043	32.98/0.9090	33.23/0.9250	33.45/0.9149	33.51/0.9153	33.59/0.9166	33.69/0.9174	33.59/0.9168
	$q=40$	33.63/0.9198	33.96/0.9247	-	34.47/0.9299	34.51/0.9302	34.58/0.9312	34.67/0.9317	34.58/0.9313

4.5. Analysis Experiments

To investigate the efficiency and effectiveness of our CODE, we conduct analysis experiments on the two neural blocks, *i.e.*, CA and DA, and the factor n in the channel-wise slice and merge mechanism. Due to space limitations, additional experiments would be presented in the supplementary material.

Table 5. Analysis experiments on Set12 with the noise level of 50. Note that Swin Attention has the same parameters and FLOPs as Local Attention, because the shift operation in Swin Attention consumes almost no parameters and computations.

Ablation	#Params	FLOPs	PSNR	SSIM
Local Attention [49]	14.05M	40.48G	27.90	0.8077
Swin Attention [22]	14.05M	40.48G	27.90	0.8079
Channel Attention [56]	14.05M	50.58G	27.91	0.8060
CA (ours)	12.18M	22.44G	27.93	0.8083
Vanilla FFN [22]	11.44M	17.93G	27.74	0.8001
DA (ours)	12.18M	22.44G	27.93	0.8083

Table 6. Ablation study for the factor n in the channel-wise slice and merge mechanism of CA.

Factor	#Params	FLOPs	PSNR	SSIM
$n=2$	11.61M	22.37G	27.90	0.8072
$n=4$	12.18M	22.44G	27.93	0.8083
$n=8$	13.33M	22.59G	27.92	0.8079

For CA, we replace it with three existing effective attention mechanisms, *i.e.*, local attention [49], shift window attention [22], and channel attention [56]. The results are shown in Tab. 5. From the table, one could see that CA obtains the best PSNR and SSIM values with fewer parameters and FLOPs compared with the existing attentions, which demonstrates not only its efficiency and effectiveness, but also the significance of our proposed paradigms for capturing the superpixel-wise global dependency. To be specific, compared with the channel attention, CA obtains 0.02dB and 0.0023 higher values in PSNR and SSIM while 1.87M and 28.14G lower values in parameters and FLOPs, respectively. Compared to the swin and the local attentions, CA obtains 0.03dB/0.03dB and 0.0004/0.0006 gains in PSNR and SSIM, respectively, with 1.87M and 18.04G fewer parameters and FLOPs.

For DA, we compare it with the vanilla FFN [22] in an MLP fashion and show the results in Tab. 5. From the table, one could see that DA performs significantly better than the vanilla FFN with slightly more parameters and FLOPs, because the vanilla FFN cannot effectively distribute the superpixel-wise globality into pixels for better pixel-wise restoration. Namely, DA could encapsulate the globality from superpixels into pixels in an efficient and effective way, and thus better cooperating with our CA.

As for the influences of the factor n in the channel-wise slice and merge mechanism, we change it from 2, 4, to 8, and show the results in Tab. 6. According to the results, a larger n could enhance the performance, as the channel slices increase the flexibility of features at each channel. Meanwhile, an overlarge n cannot obtain the best results, as too many slices cause lots of feature redundancies, making the attention hard to attend informative features. In our method, we experimentally find $n=4$ is a suitable value.

5. Conclusions

In this paper, we propose a novel efficient image restoration Transformer that obtains the pixel-wise global dependency by first capturing the global dependency at the superpixel level, and then transferring the globality from superpixels to pixels. To achieve this end, two neural blocks are proposed. In brief, CA implements a three-step paradigm to efficiently capture the global dependency at the superpixel level. DA takes a novel dual-way structure to adaptively encapsulate the globality from superpixels into pixels. Thanks to the two complementary neural blocks, our CODE enjoys the advantage of capturing the pixel-wise global dependency, while embracing high computational efficiency.

6. Acknowledgement

This work was supported in part by NSFC under Grant U21B2040, 62176171, and U19A2078; in part by Sichuan Science and Technology Planning Project under Grant 2022YFQ0014.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 6

[2] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010. 6

[3] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12299–12310, 2021. 1, 2, 6, 7

[4] Chao Dong, Yubin Deng, Chen Change Loy, and Xiaoou Tang. Compression artifacts reduction by a deep convolutional network. In *Proceedings of the IEEE international conference on computer vision*, pages 576–584, 2015. 7, 8

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2

[6] Max Ehrlich, Larry Davis, Ser-Nam Lim, and Abhinav Shrivastava. Quantization guided jpeg artifact correction. In *European Conference on Computer Vision*, pages 293–309. Springer, 2020. 7, 8

[7] Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images. *IEEE transactions on image processing*, 16(5):1395–1411, 2007. 7

[8] Rich Franzen. Kodak lossless true color image suite. *source: http://r0k. us/graphics/kodak*, 4(2), 1999. 6

[9] Hongyun Gao, Xin Tao, Xiaoyong Shen, and Jiaya Jia. Dynamic scene deblurring with parameter selective sharing and nested skip connections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3848–3856, 2019. 7

[10] Yuanbiao Gou, Peng Hu, Jiancheng Lv, and Xi Peng. Multi-scale adaptive network for single image denoising. *arXiv preprint arXiv:2203.04313*, 2022. 2

[11] Yuanbiao Gou, Boyun Li, Zitao Liu, Songfan Yang, and Xi Peng. Clearer: Multi-scale neural architecture search for image restoration. *Advances in Neural Information Processing Systems*, 33:17129–17140, 2020. 2

[12] Qi Han, Zejia Fan, Qi Dai, Lei Sun, Ming-Ming Cheng, Jiaying Liu, and Jingdong Wang. On the connection between local attention and dynamic depth-wise convolution. In *International Conference on Learning Representations*, 2021. 1

[13] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1389–1397, 2017. 2

[14] Xixi Jia, Sanyang Liu, Xiangchu Feng, and Lei Zhang. Focnet: A fractional optimal control network for image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6054–6063, 2019. 6

[15] Zhe Jiang, Yu Zhang, Dongqing Zou, Jimmy Ren, Jiancheng Lv, and Yebin Liu. Learning event-based motion deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3320–3329, 2020. 7

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[17] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8183–8192, 2018. 7

[18] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8878–8887, 2019. 7

[19] Boyun Li, Xiao Liu, Peng Hu, Zhongqin Wu, Jiancheng Lv, and Xi Peng. All-In-One Image Restoration for Unknown Corruption. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 17431–17441, New Orleans, LA, June 2022. 2

[20] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021. 2

[21] Zheyuan Li, Yingqi Liu, Xiangyu Chen, Haoming Cai, Jinjin Gu, Yu Qiao, and Chao Dong. Blueprint separable residual network for efficient image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 833–843, June 2022. 2

[22] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. *arXiv preprint arXiv:2108.10257*, 2021. 1, 2, 3, 6, 7, 8

[23] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 6

[24] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. *Advances in neural information processing systems*, 31, 2018. 6

[25] Jiaying Liu, Dong Liu, Wenhan Yang, Sifeng Xia, Xiaoshuai Zhang, and Yuanying Dai. A comprehensive benchmark for single image compression artifact reduction. *IEEE Transactions on image processing*, 29:7845–7860, 2020. 1

[26] Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level wavelet-cnn for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 773–782, 2018. 6

[27] Yang Liu, Jinshan Pan, Jimmy Ren, and Zhixun Su. Learning deep priors for image dehazing. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2492–2500, 2019. 2

[28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1, 2

[29] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5

[30] Xiaotong Luo, Yanyun Qu, Yuan Xie, Yulun Zhang, Cuihua Li, and Yun Fu. Lattice network for lightweight image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1

[31] Kede Ma, Zhengfang Duanmu, Qingbo Wu, Zhou Wang, Hongwei Yong, Hongliang Li, and Lei Zhang. Waterloo exploration database: New challenges for image quality assessment models. *IEEE Transactions on Image Processing*, 26(2):1004–1016, 2016. 6

[32] Long Ma, Tengyu Ma, Risheng Liu, Xin Fan, and Zhongxuan Luo. Toward fast, flexible, and robust low-light image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5637–5646, 2022. 2

[33] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423. IEEE, 2001. 6

[34] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3883–3891, 2017. 6, 7

[35] Jinshan Pan, Jiangxin Dong, Yang Liu, Jiawei Zhang, Jimmy Ren, Jinhui Tang, Yu-Wing Tai, and Ming-Hsuan Yang. Physics-based generative adversarial models for image restoration and beyond. *IEEE transactions on pattern analysis and machine intelligence*, 43(7):2449–2462, 2020. 1

[36] Dongwon Park, Dong Un Kang, Jisoo Kim, and Se Young Chun. Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In *European Conference on Computer Vision*, pages 327–343. Springer, 2020. 7

[37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[38] Yali Peng, Lu Zhang, Shigang Liu, Xiaojun Wu, Yu Zhang, and Xili Wang. Dilated residual networks with symmetric skip connection for image denoising. *Neurocomputing*, 345:67–76, 2019. 6, 7

[39] Wenqi Ren, Xiaochun Cao, Jinshan Pan, Xiaojie Guo, Wangmeng Zuo, and Ming-Hsuan Yang. Image deblurring via enhanced low-rank prior. *IEEE Transactions on Image Processing*, 25(7):3426–3437, 2016. 2

[40] HR Sheikh. Live image quality assessment database release 2. *http://live. ece. utexas. edu/research/quality*, 2005. 7

[41] Ziyi Shen, Wenguan Wang, Xiankai Lu, Jianbing Shen, Haibin Ling, Tingfa Xu, and Ling Shao. Human-aware motion deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5572–5581, 2019. 7

[42] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 4

[43] Maitreya Suin, Kuldeep Purohit, and AN Rajagopalan. Spatially-attentive patch-hierarchical network for adaptive motion deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3606–3615, 2020. 7

[44] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8174–8182, 2018. 7

[45] Chunwei Tian, Yong Xu, and Wangmeng Zuo. Image denoising using deep cnn with batch renormalization. *Neural Networks*, 121:461–473, 2020. 6, 7

[46] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 114–125, 2017. 6

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[48] Wenxin Wang, Boyun Li, Yuanbiao Gou, Peng Hu, and Xi Peng. Relationship quantification of image degradations. *arXiv preprint arXiv:2212.04148*, 2022. 2

[49] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17683–17693, 2022. 1, 2, 8

[50] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. 2

[51] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In

Proceedings of the European conference on computer vision (ECCV), pages 3–19, 2018. 1, 2

[52] Yanyang Yan, Wenqi Ren, Yuanfang Guo, Rui Wang, and Xiaochun Cao. Image deblurring via extreme channels prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4003–4011, 2017. 1

[53] Zizheng Yang, Mingde Yao, Jie Huang, Man Zhou, and Feng Zhao. Sir-former: Stereo image restoration using transformer. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6377–6385, 2022. 1

[54] Mingde Yao, Dongliang He, Xin Li, Fu Li, and Zhiwei Xiong. Towards interactive self-supervised denoising. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023. 1

[55] Mingde Yao, Dongliang He, Xin Li, Zhihong Pan, and Zhiwei Xiong. Bidirectional translation between uhd-hdr and hd-sdr videos. *IEEE Transactions on Multimedia*, 2023. 1

[56] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5728–5739, 2022. 1, 3, 8

[57] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14821–14831, 2021. 2

[58] Hongguang Zhang, Yuchao Dai, Hongdong Li, and Piotr Koniusz. Deep stacked hierarchical multi-patch network for image deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5978–5986, 2019. 7

[59] Jiawei Zhang, Jinshan Pan, Jimmy Ren, Yibing Song, Linchao Bao, Rynson WH Lau, and Ming-Hsuan Yang. Dynamic scene deblurring using spatially variant recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2521–2529, 2018. 7

[60] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1, 2, 6, 7, 8

[61] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, and Hongdong Li. Deblurring by realistic blurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2737–2746, 2020. 7

[62] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017. 6, 7, 8

[63] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3929–3938, 2017. 2, 6, 7

[64] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018. 6, 7

[65] Lei Zhang, Xiaolin Wu, Antoni Buades, and Xin Li. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *Journal of Electronic imaging*, 20(2):023016, 2011. 6

[66] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. *arXiv preprint arXiv:1903.10082*, 2019. 6, 7, 8

[67] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(7):2480–2495, 2020. 1, 2, 6, 7, 8

[68] Hengyuan Zhao, Xiangtao Kong, Jingwen He, Yu Qiao, and Chao Dong. Efficient image super-resolution using pixel attention. In *European Conference on Computer Vision*, pages 56–72, 2020. 1

[69] Guijing Zhu, Long Ma, Xin Fan, and Risheng Liu. Hierarchical bilevel learning with architecture and loss search for hadamard-based image restoration. In *IJCAI*, pages 1732–1739, 07 2022. 2