Restormer: Efficient Transformer for High-Resolution Image Restoration

Syed Waqas Zamir[1] Aditya Arora[1] Salman Khan[2] Munawar Hayat[2,3]
Fahad Shahbaz Khan[2,4] Ming-Hsuan Yang[5,6,7]
[1]Inception Institute of AI [2]Mohamed bin Zayed University of AI [3]Monash University
[4]Linköping Univeristy [5]University of California, Merced [6]Yonsei University [7]Google Research

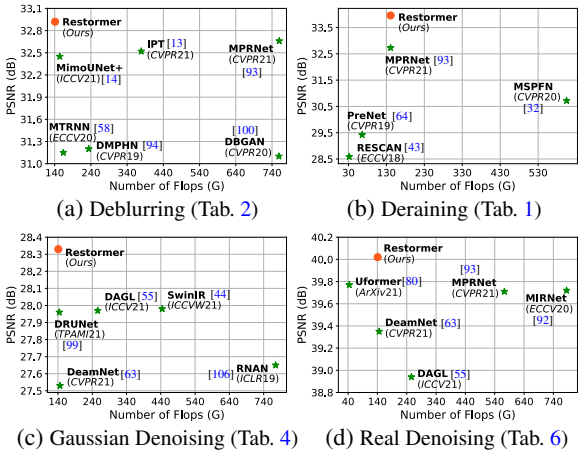
Abstract

Since convolutional neural networks (CNNs) perform well at learning generalizable image priors from large-scale data, these models have been extensively applied to image restoration and related tasks. Recently, another class of neural architectures, Transformers, have shown significant performance gains on natural language and high-level vision tasks. While the Transformer model mitigates the shortcomings of CNNs (i.e., *limited receptive field and inadaptability to input content), its computational complexity grows quadratically with the spatial resolution, therefore making it infeasible to apply to most image restoration tasks involving high-resolution images. In this work, we propose an efficient Transformer model by making several key designs in the building blocks (multi-head attention and feed-forward network) such that it can capture long-range pixel interactions, while still remaining applicable to large images. Our model, named Restoration Transformer (Restormer), achieves state-of-the-art results on several image restoration tasks, including image deraining, single-image motion deblurring, defocus deblurring (single-image and dual-pixel data), and image denoising (Gaussian grayscale/color denoising, and real image denoising). The source code and pre-trained models are available at* https://github.com/swz30/Restormer.

1. Introduction

Image restoration is the task of reconstructing a high-quality image by removing degradations (*e.g.*, noise, blur, rain drops) from a degraded input. Due to the ill-posed nature, it is a highly challenging problem that usually requires strong image priors for effective restoration. Since convolutional neural networks (CNNs) perform well at learning generalizable priors from large-scale data, they have emerged as a preferable choice compared to conventional restoration approaches.

The basic operation in CNNs is the '*convolution*' that provides local connectivity and translation equivariance. While these properties bring efficiency and generalization to CNNs, they also cause two main issues. **(a)** The convolution operator has a limited receptive field, thus preventing it from modeling long-range pixel dependencies. **(b)** The convolution filters have static weights at inference, and thereby cannot flexibly adapt to the input content. To deal with the above-mentioned shortcomings, a more powerful and dynamic alternative is the *self-attention (SA)* mechanism [17,77,79,95] that calculates response at a given pixel by a weighted sum of all other positions.



(a) Deblurring (Tab. 2) (b) Deraining (Tab. 1)

(c) Gaussian Denoising (Tab. 4) (d) Real Denoising (Tab. 6)

Figure 1. Our Restormer achieves the state-of-the-art performance on image restoration tasks while being computationally efficient.

Self-attention is a core component in Transformer models [34, 77] but with a unique implementation, *i.e.*, *multi-head SA* that is optimized for parallelization and effective representation learning. Transformers have shown state-of-the-art performance on natural language tasks [10,19,49,62] and on high-level vision problems [11,17,76,78]. Although SA is highly effective in capturing long-range pixel interactions, its complexity grows quadratically with the spatial resolution, therefore making it infeasible to apply to high-resolution images (a frequent case in image restoration). Recently, few efforts have been made to tailor Transformers for image restoration tasks [13,44,80]. To reduce the computational loads, these methods either apply SA on small spatial windows of size 8×8 around each pixel [44, 80], or

divide the input image into non-overlapping patches of size 48×48 and compute SA on each patch independently [13]. However, restricting the spatial extent of SA is contradictory to the goal of capturing the true long-range pixel relationships, especially on high-resolution images.

In this paper, we propose an efficient Transformer for image restoration that is capable of modeling global connectivity and is still applicable to large images. Specifically, we introduce a multi-Dconv head '*transposed*' attention (MDTA) block (Sec. 3.1) in place of vanilla multi-head SA [77], that has linear complexity. It applies SA across feature dimension rather than the spatial dimension, *i.e*., instead of explicitly modeling pairwise pixel interactions, MDTA computes cross-covariance across feature channels to obtain attention map from the (*key* and *query* projected) input features. An important feature of our MDTA block is the local context mixing before feature covariance computation. This is achieved via pixel-wise aggregation of cross-channel context using 1×1 convolution and channel-wise aggregation of local context using efficient depth-wise convolutions. This strategy provides two key advantages. First, it emphasizes on the spatially local context and brings in the complimentary strength of convolution operation within our pipeline. Second, it ensures that the contextualized global relationships between pixels are implicitly modeled while computing covariance-based attention maps.

A feed-forward network (FN) is the other building block of the Transformer model [77], which consists of two fully connected layers with a non-linearity in between. In this work, we reformulate the first linear transformation layer of the regular FN [77] with a gating mechanism [16] to improve the information flow through the network. This gating layer is designed as the element-wise product of two linear projection layers, one of which is activated with the GELU non-linearity [27]. Our gated-Dconv FN (GDFN) (Sec. 3.2) is also based on local content mixing similar to the MDTA module to equally emphasize on the spatial context. The gating mechanism in GDFN controls which complementary features should flow forward and allows subsequent layers in the network hierarchy to specifically focus on more refined image attributes, thus leading to high-quality outputs.

Apart from the above architectural novelties, we show the effectiveness of our progressive learning strategy for Restormer (Sec. 3.3). In this process, the network is trained on small patches and large batches in early epochs, and on gradually large image patches and small batches in later epochs. This training strategy helps Restormer to learn context from large images, and subsequently provides quality performance improvements at test time. We conduct comprehensive experiments and demonstrate state-of-the-art performance of our Restormer on 16 benchmark datasets for several image restoration tasks, including image deraining, single-image motion deblurring, defocus deblurring (on single-image and dual pixel data), and image denoising (on synthetic and real data); See Fig. 1. Furthermore, we provide extensive ablations to show the effectiveness of architectural designs and experimental choices.

The main contributions of this work are summarized below:

- We propose Restormer, an encoder-decoder Transformer for multi-scale *local-global* representation learning on high-resolution images without disintegrating them into local windows, thereby exploiting distant image context.
- We propose a multi-Dconv head transposed attention (MDTA) module that is capable of aggregating local and non-local pixel interactions, and is efficient enough to process high-resolution images.
- A new gated-Dconv feed-forward network (GDFN) that performs controlled feature transformation, *i.e*., suppressing less informative features, and allowing only the useful information to pass further through the network hierarchy.

2. Background

Image Restoration. In recent years, data-driven CNN architectures [7, 18, 92, 93, 105, 107] have been shown to outperform conventional restoration approaches [26,36,53,75]. Among convolutional designs, encoder-decoder based U-Net architectures [3, 14, 39, 80, 90, 93, 99] have been predominantly studied for restoration due to their hierarchical multi-scale representation while remaining computationally efficient. Similarly, skip connection based approaches have been shown to be effective for restoration due to specific focus on learning residual signals [24, 48, 92, 106]. Spatial and channel attention modules have also been incorporated to selectively attend to relevant information [43,92,93]. We refer the reader to NTIRE challenge reports [2,5,30,57] and recent literature reviews [8,42,73], which summarize major design choices for image restoration.

Vision Transformers. The Transformer model is first developed for sequence processing in natural language tasks [77]. It has been adapted in numerous vision tasks such as image recognition [17, 76, 88], segmentation [78, 83, 108], object detection [11, 50, 109]. The Vision Transformers [17, 76] decompose an image into a sequence of patches (local windows) and learn their mutual relationships. The distinguishing feature of these models is the strong capability to learn long-range dependencies between image patch sequences and adaptability to given input content [34]. Due to these characteristics, Transformer models have also been studied for the low-level vision problems such as super-resolution [44,85], image colorization [37], denoising [13, 80], and deraining [80]. However, the computational complexity of SA in Transformers can increase quadratically with the number of image patches, therby prohibiting its application to high-resolution images. Therefore, in low-level image processing applications, where high-resolution

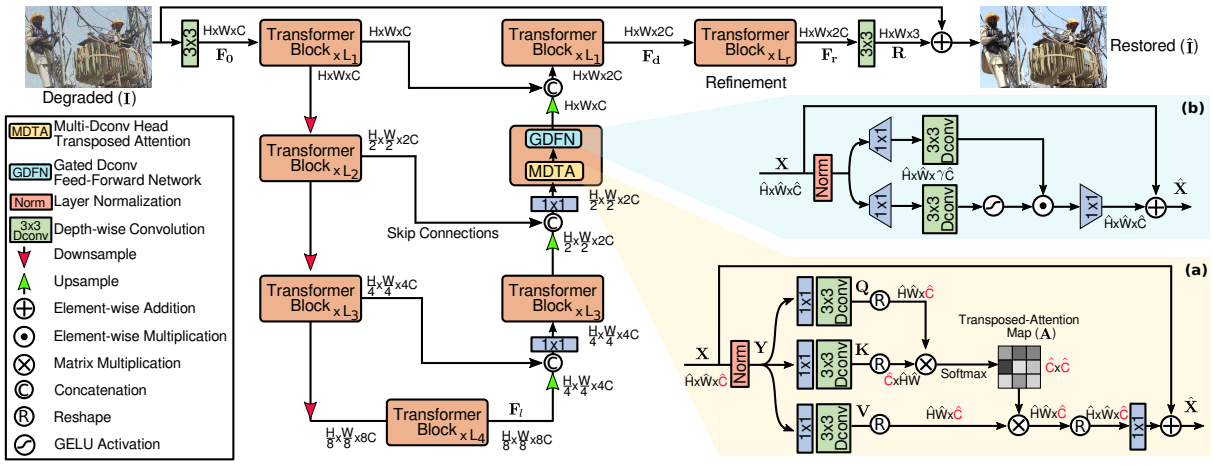


Figure 2. Architecture of Restormer for high-resolution image restoration. Our Restormer consists of multiscale hierarchical design incorporating efficient Transformer blocks. The core modules of Transformer block are: **(a)** multi-Dconv head transposed attention (MDTA) that performs (spatially enriched) *query-key* feature interaction across channels rather the spatial dimension, and **(b)** Gated-Dconv feed-forward network (GDFN) that performs controlled feature transformation, *i.e.*, to allow useful information to propagate further.

outputs need to be generated, recent methods generally employ different strategies to reduce complexity. One potential remedy is to apply self-attention within local image regions [44, 80] using the Swin Transformer design [44]. However, this design choice restricts the context aggregation within local neighbourhood, defying the main motivation of using self-attention over convolutions, thus not ideally suited for image-restoration tasks. In contrast, we present a Transformer model that can learn long-range dependencies while remaining computationally efficient.

3. Method

Our main goal is to develop an efficient Transformer model that can handle high-resolution images for restoration tasks. To alleviate the computational bottleneck, we introduce key designs to the multi-head SA layer and a multi-scale hierarchical module that has lesser computing requirements than a single-scale network [44]. We first present the overall pipeline of our Restormer architecture (see Fig. 2). Then we describe the core components of the proposed Transformer block: **(a)** multi-Dconv head transposed attention (MDTA) and **(b)** gated-Dconv feed-forward network (GDFN). Finally, we provide details on the progressive training scheme for effectively learning image statistics.

Overall Pipeline. Given a degraded image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, Restormer first applies a convolution to obtain low-level feature embeddings $\mathbf{F_0} \in \mathbb{R}^{H \times W \times C}$; where $H \times W$ denotes the spatial dimension and C is the number of channels. Next, these shallow features $\mathbf{F_0}$ pass through a 4-level symmetric encoder-decoder and transformed into deep features $\mathbf{F_d} \in \mathbb{R}^{H \times W \times 2C}$. Each level of encoder-decoder contains multiple Transformer blocks, where the number of blocks are gradually increased from the top to bottom levels to maintain efficiency. Starting from the high-resolution input, the encoder hierarchically reduces spatial size, while expanding channel capacity. The decoder takes low-resolution latent features $\mathbf{F}_l \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 8C}$ as input and progressively recovers the high-resolution representations. For feature downsampling and upsampling, we apply pixel-unshuffle and pixel-shuffle operations [69], respectively. To assist the recovery process, the encoder features are concatenated with the decoder features via skip connections [66]. The concatenation operation is followed by a 1×1 convolution to reduce channels (by half) at all levels, except the top one. At level-1, we let Transformer blocks to aggregate the low-level image features of the encoder with the high-level features of the decoder. It is beneficial in preserving the fine structural and textural details in the restored images. Next, the deep features $\mathbf{F_d}$ are further enriched in the refinement stage operating at high spatial resolution. These design choices yield quality improvements as we shall see in the experiment section (Sec. 4). Finally, a convolution layer is applied to the refined features to generate residual image $\mathbf{R} \in \mathbb{R}^{H \times W \times 3}$ to which degraded image is added to obtain the restored image: $\hat{\mathbf{I}} = \mathbf{I} + \mathbf{R}$. Next, we present the modules of the Transformer block.

3.1. Multi-Dconv Head Transposed Attention

The major computational overhead in Transformers comes from the self-attention layer. In conventional SA [17, 77], the time and memory complexity of the key-query dot-product interaction grows quadratically with the spatial resolution of input, *i.e.*, $\mathcal{O}(W^2H^2)$ for images of $W \times H$ pixels. Therefore, it is infeasible to apply SA on most image restoration tasks that often involve high-resolution im-

ages. To alleviate this issue, we propose MDTA, shown in Fig. 2(a), that has linear complexity. The key ingredient is to apply SA across channels rather than the spatial dimension, *i.e*., to compute cross-covariance across channels to generate an attention map encoding the global context implicitly. As another essential component in MDTA, we introduce depth-wise convolutions to emphasize on the local context before computing feature covariance to produce the global attention map.

From a layer normalized tensor $\mathbf{Y} \in \mathbb{R}^{\hat{H}\times\hat{W}\times\hat{C}}$, our MDTA first generates *query* ($\mathbf{Q}$), *key* ($\mathbf{K}$) and *value* ($\mathbf{V}$) projections, enriched with local context. It is achieved by applying 1×1 convolutions to aggregate pixel-wise cross-channel context followed by 3×3 depth-wise convolutions to encode channel-wise spatial context, yielding $\mathbf{Q}{=}W_d^Q W_p^Q \mathbf{Y}$, $\mathbf{K}{=}W_d^K W_p^K \mathbf{Y}$ and $\mathbf{V}{=}W_d^V W_p^V \mathbf{Y}$. Where $W_p^{(\cdot)}$ is the 1×1 point-wise convolution and $W_d^{(\cdot)}$ is the 3×3 depth-wise convolution. We use bias-free convolutional layers in the network. Next, we reshape query and key projections such that their dot-product interaction generates a transposed-attention map $\mathbf{A}$ of size $\mathbb{R}^{\hat{C}\times\hat{C}}$, instead of the huge regular attention map of size $\mathbb{R}^{\hat{H}\hat{W}\times\hat{H}\hat{W}}$ [17, 77]. Overall, the MDTA process is defined as:

$$\begin{aligned}\hat{\mathbf{X}} &= W_p\,\text{Attention}\left(\hat{\mathbf{Q}},\hat{\mathbf{K}},\hat{\mathbf{V}}\right)+\mathbf{X},\\ \text{Attention}\left(\hat{\mathbf{Q}},\hat{\mathbf{K}},\hat{\mathbf{V}}\right) &= \hat{\mathbf{V}}\cdot\text{Softmax}\left(\hat{\mathbf{K}}\cdot\hat{\mathbf{Q}}/\alpha\right),\end{aligned} \tag{1}$$

where $\mathbf{X}$ and $\hat{\mathbf{X}}$ are the input and output feature maps; $\hat{\mathbf{Q}} \in \mathbb{R}^{\hat{H}\hat{W}\times\hat{C}}$; $\hat{\mathbf{K}} \in \mathbb{R}^{\hat{C}\times\hat{H}\hat{W}}$; and $\hat{\mathbf{V}} \in \mathbb{R}^{\hat{H}\hat{W}\times\hat{C}}$ matrices are obtained after reshaping tensors from the original size $\mathbb{R}^{\hat{H}\times\hat{W}\times\hat{C}}$. Here, α is a learnable scaling parameter to control the magnitude of the dot product of $\hat{\mathbf{K}}$ and $\hat{\mathbf{Q}}$ before applying the softmax function. Similar to the conventional multi-head SA [17], we divide the number of channels into 'heads' and learn separate attention maps in parallel.

3.2. Gated-Dconv Feed-Forward Network

To transform features, the regular feed-forward network (FN) [17, 77] operates on each pixel location separately and identically. It uses two 1×1 convolutions, one to expand the feature channels (usually by factor $\gamma{=}4$) and second to reduce channels back to the original input dimension. A non-linearity is applied in the hidden layer. In this work, we propose two fundamental modifications in FN to improve representation learning: (1) gating mechanism, and (2) depthwise convolutions. The architecture of our GDFN is shown in Fig. 2(b). The gating mechanism is formulated as the element-wise product of two parallel paths of linear transformation layers, one of which is activated with the GELU non-linearity [27]. As in MDTA, we also include depth-wise convolutions in GDFN to encode information from spatially neighboring pixel positions, useful for learning local image structure for effective restoration. Given an input tensor $\mathbf{X} \in \mathbb{R}^{\hat{H}\times\hat{W}\times\hat{C}}$, GDFN is formulated as:

$$\begin{aligned}\hat{\mathbf{X}} &= W_p^0\,\text{Gating}\,(\mathbf{X})+\mathbf{X},\\ \text{Gating}(\mathbf{X}) &= \phi(W_d^1 W_p^1(\text{LN}(\mathbf{X})))\odot W_d^2 W_p^2(\text{LN}(\mathbf{X})),\end{aligned} \tag{2}$$

where $\odot$ denotes element-wise multiplication, ϕ represents the GELU non-linearity, and LN is the layer normalization [9]. Overall, the GDFN controls the information flow through the respective hierarchical levels in our pipeline, thereby allowing each level to focus on the fine details complimentary to the other levels. That is, GDFN offers a distinct role compared to MDTA (focused on enriching features with contextual information). Since the proposed GDFN performs more operations as compared to the regular FN [17], we reduce the expansion ratio γ so as to have similar parameters and compute burden.

3.3. Progressive Learning

CNN-based restoration models are usually trained on fixed-size image patches. However, training a Transformer model on small cropped patches may not encode the global image statistics, thereby providing suboptimal performance on full-resolution images at test time. To this end, we perform progressive learning where the network is trained on smaller image patches in the early epochs and on gradually larger patches in the later training epochs. The model trained on mixed-size patches via progressive learning shows enhanced performance at test time where images can be of different resolutions (a common case in image restoration). The progressive learning strategy behaves in a similar fashion to the curriculum learning process where the network starts with a simpler task and gradually moves to learning a more complex one (where the preservation of fine image structure/textures is required). Since training on large patches comes at the cost of longer time, we reduce the batch size as the patch size increases to maintain a similar time per optimization step as of the fixed patch training.

4. Experiments and Analysis

We evaluate the proposed Restormer on benchmark datasets and experimental settings for four image processing tasks: **(a)** image deraining, **(b)** single-image motion deblurring, **(c)** defocus deblurring (on single-image, and dual-pixel data), and **(d)** image denoising (on synthetic and real data). More details on datasets, training protocols, and additional visual results are presented in the supplementary material. In tables, the best and second-best quality scores of the evaluated methods are **highlighted** and underlined.

Implementation Details. We train separate models for different image restoration tasks. In all experiments, we use the following training parameters, unless mentioned otherwise. Our Restormer employs a 4-level encoder-decoder.

Table 1. **Image deraining** results. When averaged across all five datasets, our Restormer advances state-of-the-art by 1.05 dB.

Method	Test100 [97]		Rain100H [86]		Rain100L [86]		Test2800 [22]		Test1200 [96]		Average	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
DerainNet [21]	22.77	0.810	14.92	0.592	27.03	0.884	24.31	0.861	23.38	0.835	22.48	0.796
SEMI [81]	22.35	0.788	16.56	0.486	25.03	0.842	24.43	0.782	26.05	0.822	22.88	0.744
DIDMDN [96]	22.56	0.818	17.35	0.524	25.23	0.741	28.13	0.867	29.65	0.901	24.58	0.770
UMRL [87]	24.41	0.829	26.01	0.832	29.18	0.923	29.97	0.905	30.55	0.910	28.02	0.880
RESCAN [43]	25.00	0.835	26.36	0.786	29.80	0.881	31.29	0.904	30.51	0.882	28.59	0.857
PreNet [64]	24.81	0.851	26.77	0.858	32.44	0.950	31.75	0.916	31.36	0.911	29.42	0.897
MSPFN [32]	27.50	0.876	28.66	0.860	32.40	0.933	32.82	0.930	32.39	0.916	30.75	0.903
MPRNet [93]	30.27	0.897	30.41	0.890	36.40	0.965	33.64	0.938	32.91	0.916	32.73	0.921
SPAIR [61]	30.35	0.909	30.95	0.892	36.93	0.969	33.34	0.936	33.04	0.922	32.91	0.926
Restormer	**32.00**	**0.923**	**31.46**	**0.904**	**38.99**	**0.978**	**34.18**	**0.944**	**33.19**	**0.926**	**33.96**	**0.935**

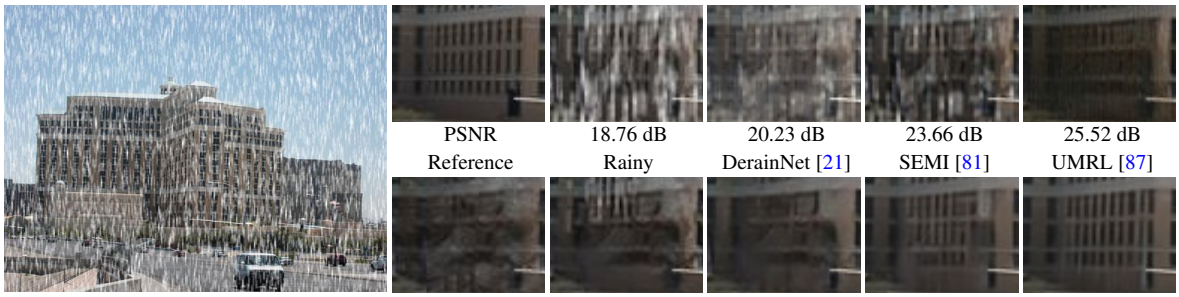


Figure 3. **Image deraining** example. Our Restormer generates rain-free image with structural fidelity and without artifacts.

Table 2. **Single-image motion deblurring** results. Our Restormer is trained only on the GoPro dataset [56] and directly applied to the HIDE [67] and RealBlur [65] benchmark datasets.

Method	GoPro [56]		HIDE [67]		RealBlur-R [65]		RealBlur-J [65]	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Xu *et al.* [84]	21.00	0.741	-		34.46	0.937	27.14	0.830
DeblurGAN [38]	28.70	0.858	24.51	0.871	33.79	0.903	27.97	0.834
Nah *et al.* [56]	29.08	0.914	25.73	0.874	32.51	0.841	27.87	0.827
Zhang *et al.* [98]	29.19	0.931	-		35.48	0.947	27.80	0.847
DeblurGAN-v2 [39]	29.55	0.934	26.61	0.875	35.26	0.944	28.70	0.866
SRN [72]	30.26	0.934	28.36	0.915	35.66	0.947	28.56	0.867
Shen *et al.* [67]	-		28.89	0.930	-		-	
Gao *et al.* [23]	30.90	0.935	29.11	0.913	-		-	
DBGAN [100]	31.10	0.942	28.94	0.915	33.78	0.909	24.93	0.745
MT-RNN [58]	31.15	0.945	29.15	0.918	35.79	0.951	28.44	0.862
DMPHN [94]	31.20	0.940	29.09	0.924	35.70	0.948	28.42	0.860
Suin *et al.* [71]	31.85	0.948	29.98	0.930	-		-	
SPAIR [61]	32.06	0.953	30.29	0.931	-		28.81	0.875
MIMO-UNet+ [14]	32.45	0.957	29.99	0.930	35.54	0.947	27.63	0.837
IPT [13]	32.52	-	-		-		-	
MPRNet [93]	32.66	0.959	30.96	0.939	35.99	0.952	28.70	0.873
Restormer	**32.92**	**0.961**	**31.22**	**0.942**	**36.19**	**0.957**	**28.96**	**0.879**

From level-1 to level-4, the number of Transformer blocks are $[4, 6, 6, 8]$, attention heads in MDTA are $[1, 2, 4, 8]$, and number of channels are $[48, 96, 192, 384]$. The refinement stage contains 4 blocks. The channel expansion factor in GDFN is γ=2.66. We train models with AdamW optimizer (β_1=0.9, β_2=0.999, weight decay $1e^{-4}$) and L_1 loss for 300K iterations with the initial learning rate $3e^{-4}$ gradually reduced to $1e^{-6}$ with the cosine annealing [51]. For progressive learning, we start training with patch size 128×128 and batch size 64. The patch size and batch size pairs are updated to $[(160^2,40), (192^2,32), (256^2,16), (320^2,8), (384^2,8)]$ at iterations [92K, 156K, 204K, 240K, 276K]. For data augmentation, we use horizontal and vertical flips.

4.1. Image Deraining Results

We compute PSNR/SSIM scores using the Y channel in YCbCr color space in a way similar to existing methods [32, 61, 93]. Table 1 shows that our Restormer achieves consistent and significant performance gains over existing approaches on all five datasets. Compared to the recent best method SPAIR [61], Restormer achieves 1.05 dB improvement when averaged across all datasets. On individual datasets, the gain can be as large as 2.06 dB, *e.g.*, Rain100L. Figure 3 shows a challenging visual example. Our Restormer reproduces a raindrop-free image while effectively preserving the structural content.

4.2. Single-image Motion Deblurring Results

We evaluate deblurring methods both on the synthetic datasets (GoPro [56], HIDE [67]) and the real-world datasets (RealBlur-R [65], RealBlur-J [65]). Table 2 shows that our Restormer outperforms other approaches on all four benchmark datasets. When averaged across all datasets, our method obtains a performance boost of 0.47 dB over the recent algorithm MIMO-UNet+ [14] and 0.26 dB over the previous best method MPRNet [93]. Compared to MPRNet [93], Restormer has 81% fewer FLOPs (See Fig. 1).

Figure 4. **Single image motion deblurring** on GoPro [56]. Restormer generates sharper and visually-faithful result.

Table 3. **Defocus deblurring** comparisons on the DPDD testset [3] (containing 37 indoor and 39 outdoor scenes). **S:** single-image defocus deblurring. **D:** dual-pixel defocus deblurring. Restormer sets new state-of-the-art for both single-image and dual pixel defocus deblurring.

Method	Indoor Scenes				Outdoor Scenes				Combined			
	PSNR ↑	SSIM ↑	MAE ↓	LPIPS ↓	PSNR ↑	SSIM ↑	MAE ↓	LPIPS ↓	PSNR ↑	SSIM ↑	MAE ↓	LPIPS ↓
EBDB$_S$ [33]	25.77	0.772	0.040	0.297	21.25	0.599	0.058	0.373	23.45	0.683	0.049	0.336
DMENet$_S$ [40]	25.50	0.788	0.038	0.298	21.43	0.644	0.063	0.397	23.41	0.714	0.051	0.349
JNB$_S$ [68]	26.73	0.828	0.031	0.273	21.10	0.608	0.064	0.355	23.84	0.715	0.048	0.315
DPDNet$_S$ [3]	26.54	0.816	0.031	0.239	22.25	0.682	0.056	0.313	24.34	0.747	0.044	0.277
KPAC$_S$ [70]	27.97	0.852	0.026	0.182	22.62	0.701	0.053	0.269	25.22	0.774	0.040	0.227
IFAN$_S$ [41]	28.11	0.861	0.026	0.179	22.76	0.720	0.052	0.254	25.37	0.789	0.039	0.217
Restormer$_S$	**28.87**	**0.882**	**0.025**	**0.145**	**23.24**	**0.743**	**0.050**	**0.209**	**25.98**	**0.811**	**0.038**	**0.178**
DPDNet$_D$ [3]	27.48	0.849	0.029	0.189	22.90	0.726	0.052	0.255	25.13	0.786	0.041	0.223
RDPD$_D$ [4]	28.10	0.843	0.027	0.210	22.82	0.704	0.053	0.298	25.39	0.772	0.040	0.255
Uformer$_D$ [80]	28.23	0.860	0.026	0.199	23.10	0.728	0.051	0.285	25.65	0.795	0.039	0.243
IFAN$_D$ [41]	28.66	0.868	0.025	0.172	23.46	0.743	0.049	0.240	25.99	0.804	0.037	0.207
Restormer$_D$	**29.48**	**0.895**	**0.023**	**0.134**	**23.97**	**0.773**	**0.047**	**0.175**	**26.66**	**0.833**	**0.035**	**0.155**

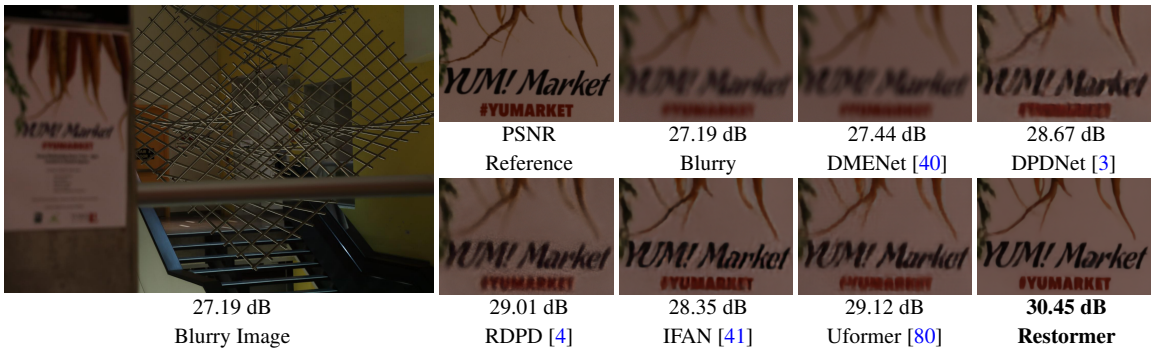


Figure 5. **Dual-pixel defocus deblurring** on DPDD [3]. Restormer effectively removes blur while preserving the fine image details.

Moreover, our method shows 0.4 dB improvement over the Transformer model IPT [13], while having 4.4× fewer parameters and runs 29× faster. Notably, our Restormer is trained only on the GoPro [56] dataset, yet it demonstrates strong generalization to other datasets by setting new state-of-the-art. Fig. 4 shows that the image produced by our method is more sharper and visually closer to the ground-truth than those of the other algorithms.

4.3. Defocus Deblurring Results

Table 3 shows image fidelity scores of the conventional defocus deblurring methods (EBDB [33] and JNB [68]) as well as learning based approaches on the DPDD dataset [3]. Our Restormer significantly outperforms the state-of-the-art schemes for the single-image and dual-pixel defocus deblurring tasks on all scene categories. Particularly on the combined scene category, Restormer yields ~ 0.6 dB improvements over the previous best method IFAN [41]. Compared to the Transformer model Uformer [80], our method provides a substantial gain of 1.01 dB PSNR. Figure 5 illustrates that our method is more effective in removing spatially varying defocus blur than other approaches.

4.4. Image Denoising Results

We perform denoising experiments on synthetic benchmark datasets generated with additive white Gaussian noise (Set12 [101], BSD68 [52], Urban100 [29], Kodak24 [20] and McMaster [104]) as well as on real-world datasets (SIDD [1] and DND [60]). Following [54, 93, 99], we use bias-free Restormer for denoising.

Table 4. **Gaussian grayscale image denoising** comparisons for two categories of methods. Top super row: learning a single model to handle various noise levels. Bottom super row: training a separate model for each noise level.

Method	Set12 [101]			BSD68 [52]			Urban100 [29]		
	σ=15	σ=25	σ=50	σ=15	σ=25	σ=50	σ=15	σ=25	σ=50
DnCNN [101]	32.67	30.35	27.18	31.62	29.16	26.23	32.28	29.80	26.35
FFDNet [103]	32.75	30.43	27.32	31.63	29.19	26.29	32.40	29.90	26.50
IRCNN [102]	32.76	30.37	27.12	31.63	29.15	26.19	32.46	29.80	26.22
DRUNet [99]	33.25	30.94	27.90	31.91	29.48	26.59	33.44	31.11	27.96
Restormer	**33.35**	**31.04**	**28.01**	**31.95**	**29.51**	**26.62**	**33.67**	**31.39**	**28.33**
FOCNet [31]	33.07	30.73	27.68	31.83	29.38	26.50	33.15	30.64	27.40
MWCNN [47]	33.15	30.79	27.74	31.86	29.41	26.53	33.17	30.66	27.42
NLRN [46]	33.16	30.80	27.64	31.88	29.41	26.47	33.45	30.94	27.49
RNAN [106]	-	-	27.70	-	-	26.48	-	-	27.65
DeamNet [63]	33.19	30.81	27.74	31.91	29.44	26.54	33.37	30.85	27.53
DAGL [55]	33.28	30.93	27.81	31.93	29.46	26.51	33.79	31.39	27.97
SwinIR [44]	33.36	31.01	27.91	**31.97**	29.50	26.58	33.70	31.30	27.98
Restormer	**33.42**	**31.08**	**28.00**	31.96	**29.52**	**26.62**	**33.79**	**31.46**	**28.29**

Table 5. **Gaussian color image denoising**. Our Restormer demonstrates favorable performance among both categories of methods. On Urban dataset [29] for noise level 50, Restormer yields 0.41 dB gain over CNN-based DRUNet [99], and 0.2 dB over Transformer model SwinIR [44].

Method	CBSD68 [52]			Kodak24 [20]			McMaster [104]			Urban100 [29]		
	σ=15	σ=25	σ=50	σ=15	σ=25	σ=50	σ=15	σ=25	σ=50	σ=15	σ=25	σ=50
IRCNN [102]	33.86	31.16	27.86	34.69	32.18	28.93	34.58	32.18	28.91	33.78	31.20	27.70
FFDNet [103]	33.87	31.21	27.96	34.63	32.13	28.98	34.66	32.35	29.18	33.83	31.40	28.05
DnCNN [101]	33.90	31.24	27.95	34.60	32.14	28.95	33.45	31.52	28.62	32.98	30.81	27.59
DSNet [59]	33.91	31.28	28.05	34.63	32.16	29.05	34.67	32.40	29.28	-	-	-
DRUNet [99]	34.30	31.69	28.51	35.31	32.89	29.86	35.40	33.14	30.08	34.81	32.60	29.61
Restormer	**34.39**	**31.78**	**28.59**	**35.44**	**33.02**	**30.00**	**35.55**	**33.31**	**30.29**	**35.06**	**32.91**	**30.02**
RPCNN [82]	-	31.24	28.06	-	32.34	29.25	-	32.33	29.33	-	31.81	28.62
BRDNet [74]	34.10	31.43	28.16	34.88	32.41	29.22	35.08	32.75	29.52	34.42	31.99	28.56
RNAN [106]	-	-	28.27	-	-	29.58	-	-	29.72	-	-	29.08
RDN [107]	-	-	28.31	-	-	29.66	-	-	-	-	-	29.38
IPT [13]	-	-	28.39	-	-	29.64	-	-	29.98	-	-	29.71
SwinIR [44]	**34.42**	31.78	28.56	35.34	32.89	29.79	35.61	33.20	30.22	35.13	32.90	29.82
Restormer	34.40	**31.79**	**28.60**	**35.47**	**33.04**	**30.01**	**35.61**	**33.34**	**30.30**	**35.13**	**32.96**	**30.02**

Table 6. **Real image denoising** on SIDD [1] and DND [60] datasets. $*$ denotes methods using additional training data. Our Restormer is trained only on the SIDD images and directly tested on DND. Among competing approaches, only Restormer surpasses 40 dB PSNR.

Dataset	Method	DnCNN [101]	BM3D [15]	CBDNet* [25]	RIDNet* [6]	AINDNet* [35]	VDN [89]	SADNet* [12]	DANet+* [90]	CycleISP* [91]	MIRNet [92]	DeamNet* [63]	MPRNet [93]	DAGL [55]	Uformer [80]	**Restormer** (Ours)
SIDD [1]	PSNR ↑	23.66	25.65	30.78	38.71	39.08	39.28	39.46	39.47	39.52	39.72	39.47	39.71	38.94	39.77	**40.02**
	SSIM ↑	0.583	0.685	0.801	0.951	0.954	0.956	0.957	0.957	0.957	0.959	0.957	0.958	0.953	0.959	**0.960**
DND [60]	PSNR ↑	32.43	34.51	38.06	39.26	39.37	39.38	39.59	39.58	39.56	39.88	39.63	39.80	39.77	39.96	**40.03**
	SSIM ↑	0.790	0.851	0.942	0.953	0.951	0.952	0.952	0.955	0.956	0.956	0.953	0.954	0.956	0.956	**0.956**

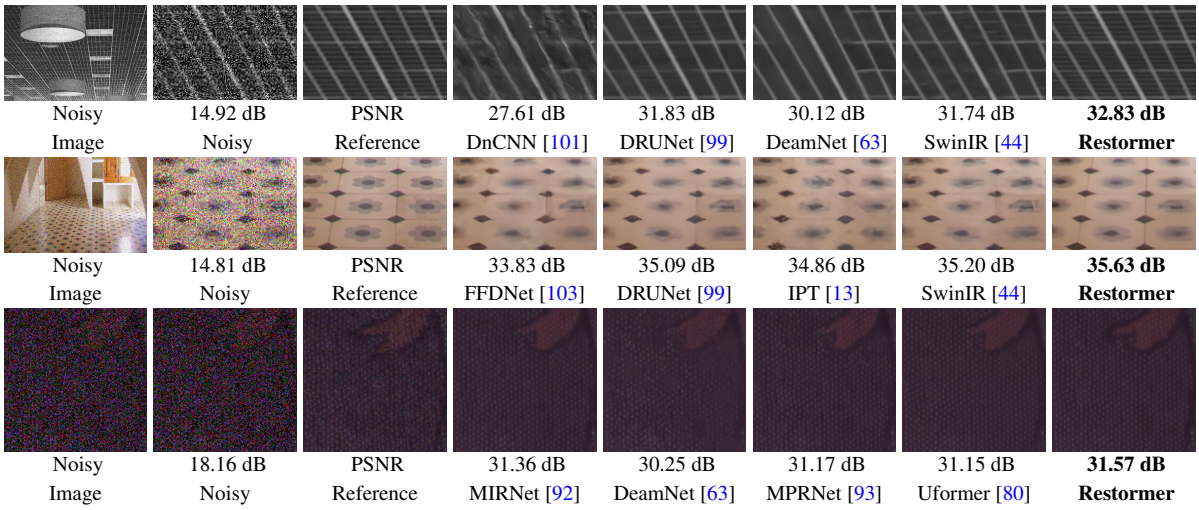


Figure 6. Visual results on **Image denoising**. Top row: Gaussian grayscale denoising. Middle row: Gaussian color denoising. Bottom row: real image denoising. The image reproduction quality of our Restormer is more faithful to the ground-truth than other methods.

Gaussian denoising. Table 4 and Table 5 show PSNR scores of different approaches on several benchmark datasets for grayscale and color image denoising, respectively. Consistent with existing methods [44, 99], we include noise levels 15, 25 and 50 in testing. The evaluated methods are divided into two experimental categories: (1) learning a single model to handle various noise levels, and (2) learning a separate model for each noise level. Our Restormer achieves state-of-the-art performance under both experimental settings on different datasets and noise levels. Specifically, for the challenging noise level 50 on high-resolution Urban100 dataset [29], Restormer achieves 0.37 dB gain over the previous best CNN-based method DRUNet [99], and 0.31 dB boost over the recent transformer-based network SwinIR [44], as shown in Table 4. Similar performance gains can be observed for the Gaussian color denoising in Table 5. It is worth mentioning that DRUNet [99] requires the noise level map as an addi-

Table 7. **Ablation experiments for the Transformer block**. PSNR is computed on a high-resolution Urban100 dataset [29].

Network		Component	FLOPs (B)	Params (M)	PSNR σ=15	PSNR σ=25	PSNR σ=50
Baseline	**(a)**	UNet with Resblocks [45]	83.4	24.53	34.42	32.18	29.11
Multi-head attention	**(b)**	MTA + FN [77]	83.7	24.84	34.66	32.39	29.28
	(c)	MDTA + FN [77]	85.3	25.02	34.72	32.48	29.43
Feed-forward network	**(d)**	MTA + GFN	83.5	24.79	34.70	32.43	32.40
	(e)	MTA + DFN	85.8	25.08	34.68	32.45	29.42
	(f)	MTA + GDFN	86.2	25.12	34.77	32.56	29.54
Overall	**(g)**	MDTA + GDFN	87.7	25.31	34.82	32.61	29.62

tional input, whereas our method only takes the noisy image. Furthermore, compared to SwinIR [44], our Restormer has 3.14× fewer FLOPs and runs 13× faster. Figure 6 presents denoised results by different methods for grayscale denoising (top row) and color denoising (middle row). Our Restormer restores clean and crisp images.

Real image denoising. Table 6 shows that our method is the only one surpassing 40 dB PSNR on both datasets. Notably, on the SIDD dataset our Restormer obtains PSNR gains of 0.3 dB and 0.25 dB over the previous best CNN method MIRNet [92] and Transformer model Uformer [80], respectively. Fig. 6 (bottom row) shows that our Restormer generates clean image without compromising fine texture.

4.5. Ablation Studies

For ablation experiments, we train Gaussian color denoising models on image patches of size 128×128 for 100K iterations only. Testing is performed on Urban100 [29], and analysis is provided for a challenging noise level σ=50. FLOPs and inference time are computed on image size 256×256. Table 7-10 show that our contributions yield quality performance improvements. Next, we describe the influence of each component individually.

Improvements in multi-head attention. Table 7c demonstrates that our MDTA provides favorable gain of 0.32 dB over the baseline (Table 7a). Furthermore, bringing locality to MDTA via depth-wise convolution improves robustness as removing it results in PSNR drop (see Table 7b).

Improvements in feed-forward network (FN). Table 7d shows that the gating mechanism in FN to control information flow yields 0.12 dB gain over the conventional FN [77]. As in multi-head attention, introducing local mechanism to FN also brings performance advantages (see Table 7e). We further strengthen the FN by incorporating gated depth-wise convolutions. Our GDFN (Table 7f) achieves PSNR gain of 0.26 dB over the standard FN [77] for the noise level 50. Overall, our Transformer block contributions lead to a significant gain of 0.51 dB over the baseline.

Design choices for decoder at level-1. To aggregate encoder features with the decoder at level-1, we do not employ 1×1 convolution (that reduces channels by half) after concatenation operation. It is helpful in preserving fine textural details coming from the encoder, as shown in Table 8. These results further demonstrate the effectiveness of adding Transformer blocks in the refinement stage.

Impact of progressive learning. Table 9 shows that the progressive learning provides better results than the fixed patch training, while having similar training time.

Deeper or wider Restormer? Table 10 shows that, under similar parameters/FLOPs budget, a deep-narrow model performs more accurately than its wide-shallow counterpart. However, the wider model runs faster due to parallelization. In this paper we use deep-narrow Restormer.

5. Conclusion

We present an image restoration Transformer model, Restormer, that is computationally efficient to handle high-resolution images. We introduce key designs to the core components of the Transformer block for improved feature aggregation and transformation. Specifically, our multi-Dconv head transposed attention (MDTA) module implicitly models global context by applying self-attention across channels rather than the spatial dimension, thus having linear complexity rather than quadratic. Furthermore, the proposed gated-Dconv feed-forward network (GDFN) introduces a gating mechanism to perform controlled feature transformation. To incorporate the strength of CNNs into the Transformer model, both MDTA and GDFN modules include depth-wise convolutions for encoding spatially local context. Extensive experiments on 16 benchmark datasets demonstrate that Restormer achieves the state-of-the-art performance for numerous image restoration tasks.

Acknowledgements. Ming-Hsuan Yang is supported by the NSF CAREER grant 1149783. Munawar Hayat is supported by the ARC DECRA Fellowship DE200101100.

Table 8. Influence of concat (w/o 1x1 conv) and refinement stage at **decoder (level-1)**. We add the components to experiment Table 7**(g)**.

	PSNR (σ=50)	FLOPs	Params
Table 7**(g)**	29.62	87.7	25.31
+ Concat	29.66	110	25.65
+ Refinement	29.71	141	26.12

Table 9. Results of training Restormer on fixed patch size and progressively large patch sizes. In **progressive learning** [28], we reduce batch size (as patch size increases) to have similar time per optimization step as of fixed patch training.

Patch Size	PSNR (σ=50)	Train Time (h)
Fixed (128^2)	29.71	22.5
Progressive (128^2 to 384^2)	29.78	23.1

Table 10. **Deeper vs wider** model. We adjust # of transformer blocks to keep flops and params constant. Deep narrow model is more accurate, while wide shallow model is faster.

Dim	PSNR (σ=50)	Train Time (h)	Test Time (s)
48	**29.71**	22.5	0.115
64	29.63	15.5	0.080
80	29.56	**13.5**	**0.069**

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018. 6, 7

[2] Abdelrahman Abdelhamed, Radu Timofte, and Michael S Brown. NTIRE 2019 challenge on real image denoising: Methods and results. In *CVPR Workshops*, 2019. 2

[3] Abdullah Abuolaim and Michael S Brown. Defocus deblurring using dual-pixel data. In *ECCV*, 2020. 2, 6

[4] Abdullah Abuolaim, Mauricio Delbracio, Damien Kelly, Michael S. Brown, and Peyman Milanfar. Learning to reduce defocus blur by realistically modeling dual-pixel data. In *ICCV*, 2021. 6

[5] Abdullah Abuolaim, Radu Timofte, and Michael S Brown. NTIRE 2021 challenge for defocus deblurring using dual-pixel images: Methods and results. In *CVPR Workshops*, 2021. 2

[6] Saeed Anwar and Nick Barnes. Real image denoising with feature attention. *ICCV*, 2019. 7

[7] Saeed Anwar and Nick Barnes. Densely residual laplacian super-resolution. *TPAMI*, 2020. 2

[8] Saeed Anwar, Salman Khan, and Nick Barnes. A deep journey into super-resolution: A survey. *ACM Computing Surveys*, 2019. 2

[9] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv:1607.06450*, 2016. 4

[10] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv:2005.14165*, 2020. 1

[11] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 2

[12] Meng Chang, Qi Li, Huajun Feng, and Zhihai Xu. Spatial-adaptive network for single image denoising. In *ECCV*, 2020. 7

[13] 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 *CVPR*, 2021. 1, 2, 5, 6, 7

[14] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *ICCV*, 2021. 1, 2, 5, 6

[15] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *TIP*, 2007. 7

[16] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *ICML*, 2017. 2

[17] 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. In *ICLR*, 2021. 1, 2, 3, 4

[18] Akshay Dudhane, Syed Waqas Zamir, Salman Khan, Fahad Khan, and Ming-Hsuan Yang. Burst image restoration and enhancement. In *CVPR*, 2022. 2

[19] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv:2101.03961*, 2021. 1

[20] Rich Franzen. Kodak lossless true color image suite. `http://r0k.us/graphics/kodak/`, 1999. Online accessed 24 Oct 2021. 6, 7

[21] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *TIP*, 2017. 5

[22] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *CVPR*, 2017. 5

[23] Hongyun Gao, Xin Tao, Xiaoyong Shen, and Jiaya Jia. Dynamic scene deblurring with parameter selective sharing and nested skip connections. In *CVPR*, 2019. 5, 6

[24] Shuhang Gu, Yawei Li, Luc Van Gool, and Radu Timofte. Self-guided network for fast image denoising. In *ICCV*, 2019. 2

[25] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *CVPR*, 2019. 7

[26] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *TPAMI*, 2010. 2

[27] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv:1606.08415*, 2016. 2, 4

[28] Elad Hoffer, Berry Weinstein, Itay Hubara, Tal Ben-Nun, Torsten Hoefler, and Daniel Soudry. Mix & match: training convnets with mixed image sizes for improved accuracy, speed and scale resiliency. *arXiv:1908.08986*, 2019. 8

[29] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, 2015. 6, 7, 8

[30] Andrey Ignatov and Radu Timofte. NTIRE 2019 challenge on image enhancement: Methods and results. In *CVPR Workshops*, 2019. 2

[31] Xixi Jia, Sanyang Liu, Xiangchu Feng, and Lei Zhang. Focnet: A fractional optimal control network for image denoising. In *CVPR*, 2019. 7

[32] Kui Jiang, Zhongyuan Wang, Peng Yi, Baojin Huang, Yimin Luo, Jiayi Ma, and Junjun Jiang. Multi-scale progressive fusion network for single image deraining. In *CVPR*, 2020. 1, 5

[33] Ali Karaali and Claudio Rosito Jung. Edge-based defocus blur estimation with adaptive scale selection. *TIP*, 2017. 6

[34] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Computing Surveys*, 2021. 1, 2

[35] Yoonsik Kim, Jae Woong Soh, Gu Yong Park, and Nam Ik Cho. Transfer learning from synthetic to real-noise denoising with adaptive instance normalization. In *CVPR*, 2020. 7

[36] Johannes Kopf, Boris Neubert, Billy Chen, Michael Cohen, Daniel Cohen-Or, Oliver Deussen, Matt Uyttendaele, and

Dani Lischinski. Deep photo: Model-based photograph enhancement and viewing. *ACM TOG*, 2008. 2

[37] Manoj Kumar, Dirk Weissenborn, and Nal Kalchbrenner. Colorization transformer. In *ICLR*, 2021. 2

[38] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. DeblurGAN: Blind motion deblurring using conditional adversarial networks. In *CVPR*, 2018. 5

[39] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better. In *ICCV*, 2019. 2, 5

[40] Junyong Lee, Sungkil Lee, Sunghyun Cho, and Seungyong Lee. Deep defocus map estimation using domain adaptation. In *CVPR*, 2019. 6

[41] Junyong Lee, Hyeongseok Son, Jaesung Rim, Sunghyun Cho, and Seungyong Lee. Iterative filter adaptive network for single image defocus deblurring. In *CVPR*, 2021. 6

[42] Siyuan Li, Iago Breno Araujo, Wenqi Ren, Zhangyang Wang, Eric K Tokuda, Roberto Hirata Junior, Roberto Cesar-Junior, Jiawan Zhang, Xiaojie Guo, and Xiaochun Cao. Single image deraining: A comprehensive benchmark analysis. In *CVPR*, 2019. 2

[43] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *ECCV*, 2018. 1, 2, 5

[44] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image restoration using swin transformer. In *ICCV Workshops*, 2021. 1, 2, 3, 7, 8

[45] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPR Workshops*, 2017. 8

[46] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. In *NeurIPS*, 2018. 7

[47] Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level wavelet-cnn for image restoration. In *CVPR Workshops*, 2018. 7

[48] Xing Liu, Masanori Suganuma, Zhun Sun, and Takayuki Okatani. Dual residual networks leveraging the potential of paired operations for image restoration. In *CVPR*, 2019. 2

[49] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized bert pretraining approach. *arXiv:1907.11692*, 2019. 1

[50] 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. *arXiv:2103.14030*, 2021. 2

[51] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 5

[52] 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 *ICCV*, 2001. 6, 7

[53] Tomer Michaeli and Michal Irani. Nonparametric blind super-resolution. In *ICCV*, 2013. 2

[54] Sreyas Mohan, Zahra Kadkhodaie, Eero P Simoncelli, and Carlos Fernandez-Granda. Robust and interpretable blind image denoising via bias-free convolutional neural networks. In *ICLR*, 2019. 6

[55] Chong Mou, Jian Zhang, and Zhuoyuan Wu. Dynamic attentive graph learning for image restoration. In *ICCV*, 2021. 1, 7

[56] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017. 5, 6

[57] Seungjun Nah, Sanghyun Son, Suyoung Lee, Radu Timofte, and Kyoung Mu Lee. Ntire 2021 challenge on image deblurring. In *CVPR Workshops*, 2021. 2

[58] 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 *ECCV*, 2020. 1, 5, 6

[59] Yali Peng, Lu Zhang, Shigang Liu, Xiaojun Wu, Yu Zhang, and Xili Wang. Dilated residual networks with symmetric skip connection for image denoising. *Neurocomputing*, 2019. 7

[60] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017. 6, 7

[61] Kuldeep Purohit, Maitreya Suin, AN Rajagopalan, and Vishnu Naresh Boddeti. Spatially-adaptive image restoration using distortion-guided networks. In *ICCV*, 2021. 5

[62] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. Technical report, OpenAI, 2018. 1

[63] Chao Ren, Xiaohai He, Chuncheng Wang, and Zhibo Zhao. Adaptive consistency prior based deep network for image denoising. In *CVPR*, 2021. 1, 7

[64] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *CVPR*, 2019. 1, 5

[65] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *ECCV*, 2020. 5

[66] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 3

[67] Ziyi Shen, Wenguan Wang, Xiankai Lu, Jianbing Shen, Haibin Ling, Tingfa Xu, and Ling Shao. Human-aware motion deblurring. In *ICCV*, 2019. 5

[68] Jianping Shi, Li Xu, and Jiaya Jia. Just noticeable defocus blur detection and estimation. In *CVPR*, 2015. 6

[69] 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 *CVPR*, 2016. 3

[70] Hyeongseok Son, Junyong Lee, Sunghyun Cho, and Seungyong Lee. Single image defocus deblurring using kernel-sharing parallel atrous convolutions. In *ICCV*, 2021. 6

[71] Maitreya Suin, Kuldeep Purohit, and A. N. Rajagopalan. Spatially-attentive patch-hierarchical network for adaptive motion deblurring. In *CVPR*, 2020. 5, 6

[72] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *CVPR*, 2018. 5

[73] Chunwei Tian, Lunke Fei, Wenxian Zheng, Yong Xu, Wangmeng Zuo, and Chia-Wen Lin. Deep learning on image denoising: An overview. *Neural Networks*, 2020. 2

[74] Chunwei Tian, Yong Xu, and Wangmeng Zuo. Image denoising using deep cnn with batch renormalization. *Neural Networks*, 2020. 7

[75] Radu Timofte, Vincent De Smet, and Luc Van Gool. Anchored neighborhood regression for fast example-based super-resolution. In *ICCV*, 2013. 2

[76] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 1, 2

[77] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1, 2, 3, 4, 8

[78] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, 2021. 1, 2

[79] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 1

[80] Zhendong Wang, Xiaodong Cun, Jianmin Bao, and Jianzhuang Liu. Uformer: A general u-shaped transformer for image restoration. *arXiv:2106.03106*, 2021. 1, 2, 3, 6, 7, 8

[81] Wei Wei, Deyu Meng, Qian Zhao, Zongben Xu, and Ying Wu. Semi-supervised transfer learning for image rain removal. In *CVPR*, 2019. 5

[82] Zhihao Xia and Ayan Chakrabarti. Identifying recurring patterns with deep neural networks for natural image denoising. In *WACV*, 2020. 7

[83] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *arXiv:2105.15203*, 2021. 2

[84] Li Xu, Shicheng Zheng, and Jiaya Jia. Unnatural l0 sparse representation for natural image deblurring. In *CVPR*, 2013. 5

[85] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. Learning texture transformer network for image super-resolution. In *CVPR*, 2020. 2

[86] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep joint rain detection and removal from a single image. In *CVPR*, 2017. 5

[87] Rajeev Yasarla and Vishal M Patel. Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image de-raining. In *CVPR*, 2019. 5

[88] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv:2101.11986*, 2021. 2

[89] Zongsheng Yue, Hongwei Yong, Qian Zhao, Deyu Meng, and Lei Zhang. Variational denoising network: Toward blind noise modeling and removal. In *NeurIPS*, 2019. 7

[90] Zongsheng Yue, Qian Zhao, Lei Zhang, and Deyu Meng. Dual adversarial network: Toward real-world noise removal and noise generation. In *ECCV*, 2020. 2, 7

[91] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. CycleISP: Real image restoration via improved data synthesis. In *CVPR*, 2020. 7

[92] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Learning enriched features for real image restoration and enhancement. In *ECCV*, 2020. 1, 2, 7, 8

[93] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *CVPR*, 2021. 1, 2, 5, 6, 7

[94] Hongguang Zhang, Yuchao Dai, Hongdong Li, and Piotr Koniusz. Deep stacked hierarchical multi-patch network for image deblurring. In *CVPR*, 2019. 1, 5, 6

[95] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, 2019. 1

[96] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. In *CVPR*, 2018. 5

[97] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *TCSVT*, 2019. 5

[98] 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 *CVPR*, 2018. 5

[99] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *TPAMI*, 2021. 1, 2, 6, 7

[100] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, and Hongdong Li. Deblurring by realistic blurring. In *CVPR*, 2020. 1, 5, 6

[101] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *TIP*, 2017. 6, 7

[102] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *CVPR*, 2017. 7

[103] Kai Zhang, Wangmeng Zuo, and Lei Zhang. FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *TIP*, 2018. 7

[104] Lei Zhang, Xiaolin Wu, Antoni Buades, and Xin Li. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *JEI*, 2011. 6, 7

[105] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018. 2

[106] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. In *ICLR*, 2019. 1, 2, 7

[107] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image restoration. *TPAMI*, 2020. 2, 7

[108] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021. 2

[109] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. *arXiv:2010.04159*, 2020. 2