

# SwinIR: Image Restoration Using Swin Transformer

Jingyun Liang<sup>1</sup> Jiezheng Cao<sup>1</sup> Guolei Sun<sup>1</sup> Kai Zhang<sup>1,\*</sup> Luc Van Gool<sup>1,2</sup> Radu Timofte<sup>1</sup>

<sup>1</sup>Computer Vision Lab, ETH Zurich, Switzerland <sup>2</sup>KU Leuven, Belgium

{jinliang, jiezcaoz, guosun, kai.zhang, vangool, timofter}@vision.ee.ethz.ch

<https://github.com/JingyunLiang/SwinIR>

## Abstract

*Image restoration is a long-standing low-level vision problem that aims to restore high-quality images from low-quality images (e.g., downscaled, noisy and compressed images). While state-of-the-art image restoration methods are based on convolutional neural networks, few attempts have been made with Transformers which show impressive performance on high-level vision tasks. In this paper, we propose a strong baseline model SwinIR for image restoration based on the Swin Transformer. SwinIR consists of three parts: shallow feature extraction, deep feature extraction and high-quality image reconstruction. In particular, the deep feature extraction module is composed of several residual Swin Transformer blocks (RSTB), each of which has several Swin Transformer layers together with a residual connection. We conduct experiments on three representative tasks: image super-resolution (including classical, lightweight and real-world image super-resolution), image denoising (including grayscale and color image denoising) and JPEG compression artifact reduction. Experimental results demonstrate that SwinIR outperforms state-of-the-art methods on different tasks by up to 0.14~0.45dB, while the total number of parameters can be reduced by up to 67%.*

## 1. Introduction

Image restoration, such as image super-resolution (SR), image denoising and JPEG compression artifact reduction, aims to reconstruct the high-quality clean image from its low-quality degraded counterpart. Since several revolutionary work [18, 40, 90, 91], convolutional neural networks (CNN) have become the primary workhorse for image restoration [43, 51, 43, 81, 92, 95, 93, 46, 89, 88].

Most CNN-based methods focus on elaborate architecture designs, such residual learning [43, 51] and dense connections [97, 81], although the performance is improved significantly compared with traditional model-based meth-

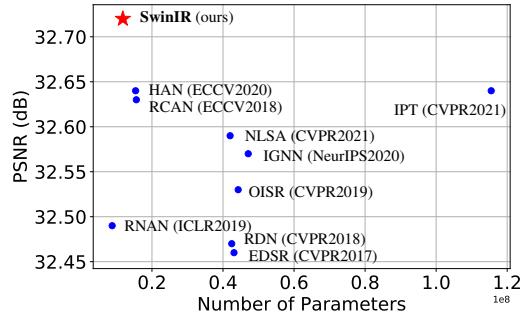


Figure 1: PSNR results v.s the total number of parameters of different methods for image SR ( $\times 4$ ) on Set5 [3].

ods [73, 14, 28], they generally suffer from two basic problems that stem from the basic building block, *i.e.*, the convolution layer. First, the interactions between images and convolution kernels are content-independent. Using the same convolution kernel to restore different image regions may not be a good choice. Second, with the principle of local processing, convolution is not effective for long-range dependency modelling.

As an alternative to CNN, Transformer [76] designs a self-attention mechanism to capture global interactions between contexts and has shown promising performance in several vision problems [6, 74, 19, 56]. However, vision Transformers for image restoration [9, 5] usually divide the input image into small patches with fixed size (*e.g.*,  $48 \times 48$ ) and process each patch independently. Such a strategy inevitably gives rise to two drawbacks. First, the restored image may introduce border artifacts around each small patch. Second, the border pixels of each patch lose information for better restoration. While this can be alleviated by patch overlapping, it would introduce extra computational burden.

Recently, Swin Transformer [56] has shown great promise as it integrates the advantages of both CNN and Transformer. On the one hand, it has the advantage of CNN to process image with large size due to the local attention mechanism. On the other hand, it has the advantage of Transformer to model long-range dependency with the shifted window scheme.

\*Corresponding author.

In this paper, we propose an image restoration model, namely SwinIR, based on Swin Transformer. More specifically, SwinIR consists of three modules: shallow feature extraction, deep feature extraction and high-quality image reconstruction modules. Shallow feature extraction module uses a convolution layer to extract shallow feature, which is directly transmitted to the reconstruction module so as to preserve low-frequency information. Deep feature extraction module is mainly composed of residual Swin Transformer blocks (RSTB), each of which utilizes several Swin Transformer layers for local attention and cross-window interaction. In addition, we add a convolution layer at the end of the block for feature enhancement and use a residual connection to provide shortcuts for feature aggregation. Finally, both shallow and deep features are fused in the reconstruction module for high-quality image reconstruction.

Compared with prevalent CNN-based image restoration models, Transformer-based SwinIR has several benefits: (1) content-based interactions between image content and attention weights, which can be interpreted as spatially varying convolution [13, 21, 75]. (2) long-range dependency modelling are enable by the shifted window mechanism. (3) better performance with less parameters. As shown in Fig. 1, SwinIR achieves better PSNR with less parameters compared with existing image SR methods.

## 2. Related Work

### 2.1. Image Restoration

Compared to traditional image restoration methods [28, 72, 73, 62, 32] which are generally model-based, learning-based methods, especially CNN-based methods, have become more popular due to their impressive performance. They often learn mappings between low-quality and high-quality images from large-scale paired datasets. Since pioneering work SRCNN [18] (for image SR), DnCNN [90] (for image denoising) and ARCNN [17] (for JPEG compression artifact reduction), a flurry of CNN-based models have been proposed to improve model representation ability by using larger and deeper neural network architecture designs, such as residual block [40, 7, 88], dense block [81, 97, 98] and others [10, 42, 93, 78, 77, 79, 50, 48, 49, 92, 70, 36, 83, 30, 11, 16, 96, 64, 38, 26, 41, 25]. Some of them have exploited the attention mechanism inside the CNN framework, such as channel attention [95, 15, 63], non-local attention [52, 61] and adaptive patch aggregation [100].

### 2.2. Vision Transformer

Recently, natural language processing model Transformer [76] has gained much popularity in the computer vision community. When used in vision problems such as image classification [66, 19, 84, 56, 45, 55, 75], ob-

ject detection [6, 53, 74, 56], segmentation [84, 99, 56, 4] and crowd counting [47, 69], it learns to attend to important image regions by exploring the global interactions between different regions. Due to its impressive performance, Transformer has also been introduced for image restoration [9, 5, 82]. Chen *et al.* [9] proposed a backbone model IPT for various restoration problems based on the standard Transformer. However, IPT relies on large number of parameters (over 115.5M parameters), large-scale datasets (over 1.1M images) and multi-task learning for good performance. Cao *et al.* [5] proposed VSR-Transformer that uses the self-attention mechanism for better feature fusion in video SR, but image features are still extracted from CNN. Besides, both IPT and VSR-Transformer are patch-wise attention, which may be improper for image restoration. A concurrent work [82] proposed a U-shaped architecture based on the Swin Transformer [56].

## 3. Method

### 3.1. Network Architecture

As shown in Fig. 2, SwinIR consists of three modules: shallow feature extraction, deep feature extraction and high-quality (HQ) image reconstruction modules. We employ the same feature extraction modules for all restoration tasks, but use different reconstruction modules for different tasks.

**Shallow and deep feature extraction.** Given a low-quality (LQ) input  $I_{LQ} \in \mathbb{R}^{H \times W \times C_{in}}$  ( $H$ ,  $W$  and  $C_{in}$  are the image height, width and input channel number, respectively), we use a  $3 \times 3$  convolutional layer  $H_{SF}(\cdot)$  to extract shallow feature  $F_0 \in \mathbb{R}^{H \times W \times C}$  as

$$F_0 = H_{SF}(I_{LQ}), \quad (1)$$

where  $C$  is the feature channel number. The convolution layer is good at early visual processing, leading to more stable optimization and better results [86]. It also provides a simple way to map the input image space to a higher dimensional feature space. Then, we extract deep feature  $F_{DF} \in \mathbb{R}^{H \times W \times C}$  from  $F_0$  as

$$F_{DF} = H_{DF}(F_0), \quad (2)$$

where  $H_{DF}(\cdot)$  is the deep feature extraction module and it contains  $K$  residual Swin Transformer blocks (RSTB) and a  $3 \times 3$  convolutional layer. More specifically, intermediate features  $F_1, F_2, \dots, F_K$  and the output deep feature  $F_{DF}$  are extracted block by block as

$$\begin{aligned} F_i &= H_{RSTB_i}(F_{i-1}), \quad i = 1, 2, \dots, K, \\ F_{DF} &= H_{CONV}(F_K), \end{aligned} \quad (3)$$

where  $H_{RSTB_i}(\cdot)$  denotes the  $i$ -th RSTB and  $H_{CONV}$  is the last convolutional layer. Using a convolutional layer at the

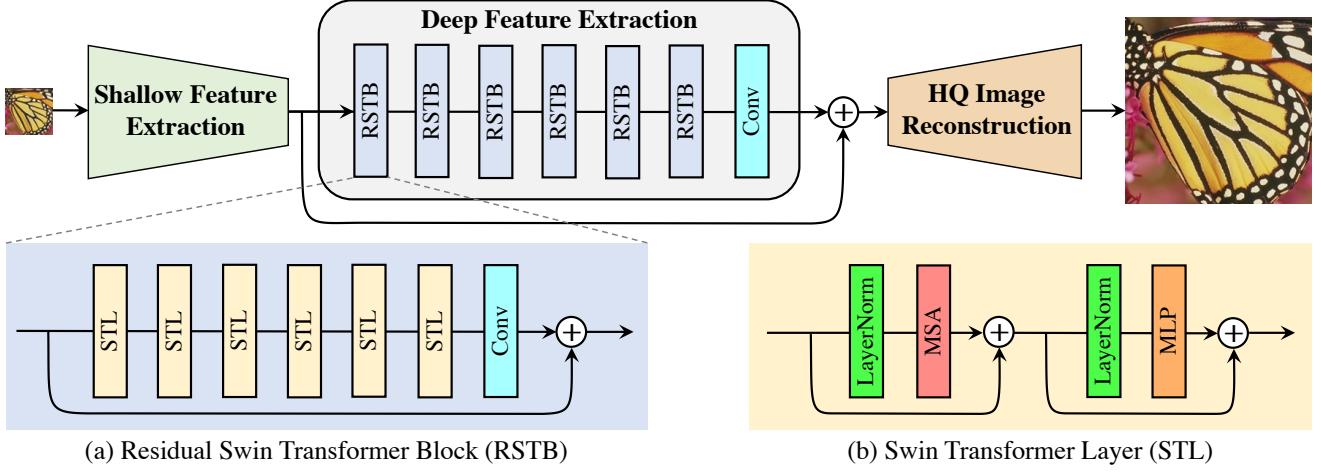


Figure 2: The architecture of the proposed SwinIR for image restoration.

end of feature extraction can bring the inductive bias of the convolution operation into the Transformer-based network, and lay a better foundation for the later aggregation of shallow and deep features.

**Image reconstruction.** Taking image SR as an example, we reconstruct the high-quality image  $I_{RHQ}$  by aggregating shallow and deep features as

$$I_{RHQ} = H_{REC}(F_0 + F_{DF}), \quad (4)$$

where  $H_{REC}(\cdot)$  is the function of the reconstruction module. Shallow feature mainly contain low-frequencies, while deep feature focus on recovering lost high-frequencies. With a long skip connection, SwinIR can transmit the low-frequency information directly to the reconstruction module, which can help deep feature extraction module focus on high-frequency information and stabilize training. For the implementation of reconstruction module, we use the sub-pixel convolution layer [68] to upsample the feature.

For tasks that do not need upsampling, such as image denoising and JPEG compression artifact reduction, a single convolution layer is used for reconstruction. Besides, we use residual learning to reconstruct the residual between the LQ and the HQ image instead of the HQ image. This is formulated as

$$I_{RHQ} = H_{SwinIR}(I_{LQ}) + I_{LQ}, \quad (5)$$

where  $H_{SwinIR}(\cdot)$  denotes the function of SwinIR.

**Loss function.** For image SR, we optimize the parameters of SwinIR by minimizing the  $L_1$  pixel loss

$$\mathcal{L} = \|I_{RHQ} - I_{HQ}\|_1, \quad (6)$$

where  $I_{RHQ}$  is obtained by taking  $I_{LQ}$  as the input of SwinIR, and  $I_{HQ}$  is the corresponding ground-truth HQ image. For

classical and lightweight image SR, we only use the naive  $L_1$  pixel loss as same as previous work to show the effectiveness of the proposed network. For real-world image SR, we use a combination of pixel loss, GAN loss and perceptual loss [81, 89, 80, 27, 39, 81] to improve visual quality.

For image denoising and JPEG compression artifact reduction, we use the Charbonnier loss [8]

$$\mathcal{L} = \sqrt{\|I_{RHQ} - I_{HQ}\|^2 + \epsilon^2}, \quad (7)$$

where  $\epsilon$  is a constant that is empirically set to  $10^{-3}$ .

### 3.2. Residual Swin Transformer Block

As shown in Fig. 2(a), the residual Swin Transformer block (RSTB) is a residual block with Swin Transformer layers and convolutional layers. Given the input feature  $F_{i,0}$  of the  $i$ -th RSTB, we first extract intermediate features  $F_{i,1}, F_{i,2}, \dots, F_{i,L}$  by  $L$  Swin Transformer layers as

$$F_{i,j} = H_{Swin_{i,j}}(F_{i,j-1}), \quad j = 1, 2, \dots, L, \quad (8)$$

where  $H_{Swin_{i,j}}(\cdot)$  is the  $j$ -th Swin Transformer layer in the  $i$ -th RSTB. Then, we add a convolutional layer before the residual connection. The output of RSTB is formulated as

$$F_{i,out} = H_{CONV_i}(F_{i,L}) + F_{i,0}, \quad (9)$$

where  $H_{CONV_i}(\cdot)$  is the convolutional layer in the  $i$ -th RSTB. This design has two benefits. First, although Transformer can be viewed as a specific instantiation of spatially varying convolution [21, 75], convolutional layers with spatially invariant filters can enhance the translational equivariance of SwinIR. Second, the residual connection provides a short identity-based connection from different blocks to the reconstruction module, allowing the aggregation of different levels of features.

**Swin Transformer layer.** Swin Transformer layer (STL) [56] is based on the standard multi-head self-attention of the original Transformer layer [76]. The main differences lie in local attention and the shifted window mechanism. As shown in Fig. 2(b), given an input of size  $H \times W \times C$ , Swin Transformer first reshapes the input to a  $\frac{HW}{M^2} \times M^2 \times C$  feature by partitioning the input into non-overlapping  $M \times M$  local windows, where  $\frac{HW}{M^2}$  is the total number of windows. Then, it computes the standard self-attention separately for each window (*i.e.*, local attention). For a local window feature  $X \in \mathbb{R}^{M^2 \times C}$ , the *query*, *key* and *value* matrices  $Q$ ,  $K$  and  $V$  are computed as

$$Q = X P_Q, \quad K = X P_K, \quad V = X P_V, \quad (10)$$

where  $P_Q$ ,  $P_K$  and  $P_V$  are projection matrices that are shared across different windows. Generally, we have  $Q, K, V \in \mathbb{R}^{M^2 \times d}$ . The attention matrix is thus computed by the self-attention mechanism in a local window as

$$\text{Attention}(Q, K, V) = \text{SoftMax}(QK^T / \sqrt{d} + B)V, \quad (11)$$

where  $B$  is the learnable relative positional encoding. In practice, following [76], we perform the attention function for  $h$  times in parallel and concatenate the results for multi-head self-attention (MSA).

Next, a multi-layer perceptron (MLP) that has two fully-connected layers with GELU non-linearity between them is used for further feature transformations. The LayerNorm (LN) layer is added before both MSA and MLP, and the residual connection is employed for both modules. The whole process is formulated as

$$\begin{aligned} X &= \text{MSA}(\text{LN}(X)) + X, \\ X &= \text{MLP}(\text{LN}(X)) + X. \end{aligned} \quad (12)$$

However, when the partition is fixed for different layers, there is no connection across local windows. Therefore, regular and shifted window partitioning are used alternately to enable cross-window connections [56], where shifted window partitioning means shifting the feature by  $(\lfloor \frac{M}{2} \rfloor, \lfloor \frac{M}{2} \rfloor)$  pixels before partitioning.

## 4. Experiments

### 4.1. Experimental Setup

For classical image SR, real-world image SR, image denoising and JPEG compression artifact reduction, the RSTB number, STL number, window size, channel number and attention head number are generally set to 6, 6, 8, 180 and 6, respectively. One exception is that the window size is set to 7 for JPEG compression artifact reduction, as we observe significant performance drop when using 8, possibly because JPEG encoding uses  $8 \times 8$  image

parts. For lightweight image SR, we decrease RSTB number and channel number to 4 and 60, respectively. Following [95, 63], when self-ensemble strategy [51] is used in testing, we mark the model with a symbol “+”, *e.g.*, SwinIR+. Training and evaluation details are provided in the supplementary.

### 4.2. Ablation Study and Discussion

For ablation study, we train SwinIR on DIV2K [1] for classical image SR ( $\times 2$ ) and test it on Manga109 [60].

**Impact of channel number, RSTB number and STL number.** We show the effects of channel number, RSTB number and STL number in a RSTB on model performance in Figs. 3(a), 3(b) and 3(c), respectively. It is observed that the PSNR is positively correlated with these three hyper-parameters. Although the performance keeps increasing for large channel numbers, the total number of parameters grows quadratically. To balance the performance and model size, we choose 180 as the channel number in rest experiments. As for RSTB number and layer number, the performance gain tends to be saturated gradually. We choose 6 for both of them to obtain a relatively small model.

**Impact of patch size and training image number; model convergence comparison.** We compare the proposed SwinIR with a representative CNN-based model RCAN to exploit the difference of Transformer-based and CNN-based models. From Fig. 3(d), one can see that SwinIR performs better than RCAN on different patch sizes, and the PSNR gain becomes larger when the patch size is larger. Fig. 3(e) shows the impact of the number of training images. Extra images from Flickr2K are used in training when the percentage is larger than 100% (800 images). There are two observations. First, as expected, the performance of SwinIR rises with the training image number. Second, different from the observation in IPT that Transformer-based models reply on large amount of training data, SwinIR achieves better results than CNN-based models using the same training data, even when the dataset is small (*i.e.*, 25%, 200 images). We also plot the PSNR during training for both SwinIR and RCAN in Fig. 3(f). It is clear that SwinIR converges faster and better than RCAN, which is contradictory to previous observations that Transformer-based models often suffer from slow model convergence.

**Impact of residual connection and convolution layer in RSTB.** Table 1 shows four residual connection variants in RSTB: no residual connection, using  $1 \times 1$  convolution layer, using  $3 \times 3$  convolution layer and using three  $3 \times 3$  convolution layers (channel number of the intermediate layer is set to one fourth of network channel number). From the table, we can have following observations. First, the residual connection in RSTB is important as it improves the PSNR by 0.16dB. Second, using  $1 \times 1$  convolution brings little improvement maybe because it cannot

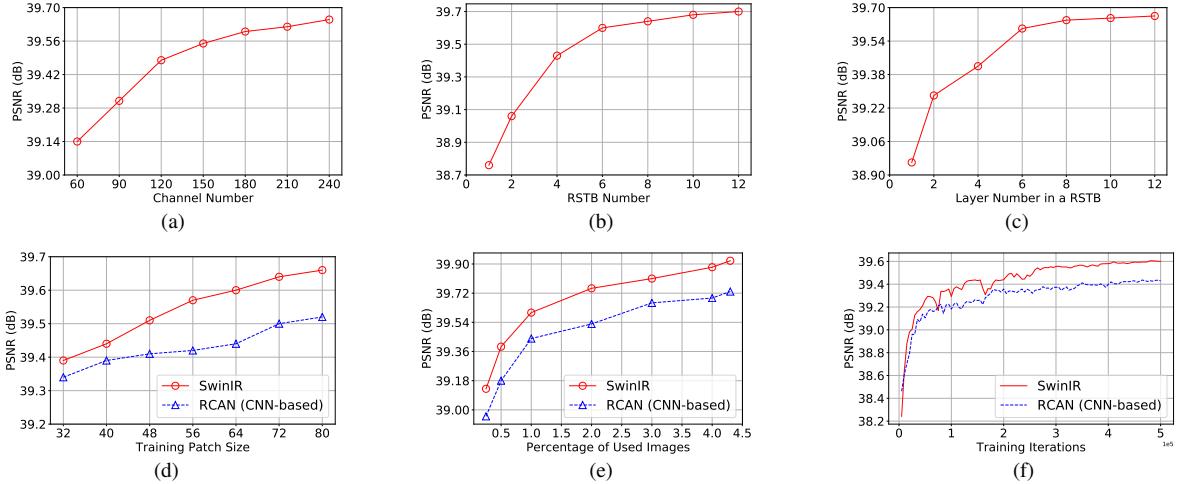


Figure 3: Ablation study on different settings of SwinIR. Results are tested on Manga109 [60] for image SR ( $\times 2$ ).

Table 1: Ablation study on RSTB design.

Design	No residual	$1 \times 1$ conv	$3 \times 3$ conv	Three $3 \times 3$ conv
PSNR	39.42	39.45	39.58	39.56

extract local neighbouring information as  $3 \times 3$  convolution does. Third, although using three  $3 \times 3$  convolution layers can reduce the number of parameters, the performance is slightly dropped.

### 4.3. Results on Image SR

**Classical image SR.** Table 2 shows the quantitative comparisons between SwinIR (middle size) and state-of-the-art methods: DBPN [31], RCAN [95], RRDB [81], SAN [15], IGNN [100], HAN [63], NLSA [61] and IPT [9]. As one can see, when trained on DIV2K, SwinIR achieves best performance on almost all five benchmark datasets for all scale factors. The maximum PSNR gain reaches 0.26dB on Manga109 for scale factor 4. Note that RCAN and HAN introduce channel and spatial attention, IGNN proposes adaptive patch feature aggregation, and NLSA is based on the non-local attention mechanism. However, all these CNN-based attention mechanisms perform worse than the proposed Transformer-based SwinIR, which indicates the effectiveness of the proposed model. When we train SwinIR on a larger dataset, the performance further increases by a large margin (up to 0.47dB), achieving better accuracy than the same Transformer-based model IPT, even though IPT utilizes ImageNet (more than 1.3M images) in training and has huge number of parameters (115.5M). In contrast, SwinIR has a small number of parameters (11.8M) even compared with state-of-the-art CNN-based models (15.4~44.3M). As for runtime, representative CNN-based model RCAN, IPT and SwinIR take about 0.2, 4.5s and 1.1s to test on a  $1,024 \times 1,024$  image, respectively. We show visual comparisons on scale factor 4 in Fig. 4. SwinIR can restore high-frequency details and alleviate the blurring arti-

facts, resulting in sharp and natural edges. In contrast, most CNN-based methods cannot restore the correct texture and produce blurry images or even different structures. IPT generates better images compared with CNN-based methods, but it suffers from image distortions and border artifact.

**Lightweight image SR.** We also provide comparison of SwinIR (small size) with state-of-the-art lightweight image SR methods: CARN [2], FALSR-A [12], IMDN [35], LAPAR-A [44] and LatticeNet [57]. In addition to PSNR and SSIM, we also report the total numbers of parameters and multiply-accumulate operations (evaluated on a  $1280 \times 720$  HQ image) to compare the model size and computational complexity of different models. As shown in Table 3, SwinIR outperforms competitive methods by a PSNR margin of up to 0.53dB on different benchmark datasets, with similar total numbers of parameters and multiply-accumulate operations. This indicates that the SwinIR architecture is highly efficient.

**Real-world image SR.** The ultimate goal of image SR is for real-world applications. Recently, Zhang *et al.* [89] proposed a practical degradation model BSRGAN for real-world image SR and achieved surprising results in real scenarios<sup>1</sup>. To test the performance of SwinIR for real-world SR, we re-train SwinIR by using the same degradation model as BSRGAN for low-quality image synthesis and test it on the real-world SR benchmark dataset Real-SRSet [89]. Since there is no ground-truth high-quality images, we only provide visual comparison with representative bicubic model ESRGAN [81] and state-of-the-art real-world image SR models FSSR [24], RealSR [37] and BSRGAN [89]. As shown in Fig. 5, SwinIR produces visually pleasing images with clear and sharp edges, whereas other compared methods may suffer from unsatisfactory artifacts. In addition, to exploit the full potential of SwinIR for real

<sup>1</sup><https://github.com/cszn/BSRGAN>

Table 2: Quantitative comparison (average PSNR/SSIM) with state-of-the-art methods for **classical image SR** on benchmark datasets. Best and second best performance are in red and blue colors, respectively. Results on  $\times 8$  are provided in supplementary.

Method	Scale	Training Dataset	Set5 [3]		Set14 [87]		BSD100 [58]		Urban100 [34]		Manga109 [60]	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
RCAN [95]	$\times 2$	DIV2K	38.27	0.9614	34.12	0.9216	32.41	0.9027	33.34	0.9384	39.44	0.9786
SAN [15]	$\times 2$	DIV2K	38.31	0.9620	34.07	0.9213	32.42	0.9028	33.10	0.9370	39.32	0.9792
IGNN [100]	$\times 2$	DIV2K	38.24	0.9613	34.07	0.9217	32.41	0.9025	33.23	0.9383	39.35	0.9786
HAN [63]	$\times 2$	DIV2K	38.27	0.9614	34.16	0.9217	32.41	0.9027	33.35	0.9385	39.46	0.9785
NLSA [61]	$\times 2$	DIV2K	38.34	0.9618	34.08	0.9231	32.43	0.9027	33.42	0.9394	39.59	0.9789
<b>SwinIR (Ours)</b>	$\times 2$	DIV2K	<b>38.35</b>	<b>0.9620</b>	34.14	0.9227	<b>32.44</b>	<b>0.9030</b>	33.40	0.9393	39.60	<b>0.9792</b>
<b>SwinIR+ (Ours)</b>	$\times 2$	DIV2K	<b>38.38</b>	<b>0.9621</b>	<b>34.24</b>	<b>0.9233</b>	<b>32.47</b>	<b>0.9032</b>	<b>33.51</b>	<b>0.9401</b>	<b>39.70</b>	<b>0.9794</b>
DBPN [31]	$\times 2$	DiF2K+Flickr2K	38.09	0.9600	33.85	0.9190	32.27	0.9000	32.55	0.9324	38.89	0.9775
IPT [9]	$\times 2$	ImageNet	38.37	-	34.43	-	32.48	-	33.76	-	-	-
<b>SwinIR (Ours)</b>	$\times 2$	DIV2K+Flickr2K	<b>38.42</b>	<b>0.9623</b>	<b>34.46</b>	<b>0.9250</b>	<b>32.53</b>	<b>0.9041</b>	<b>33.81</b>	<b>0.9427</b>	<b>39.92</b>	<b>0.9797</b>
<b>SwinIR+ (Ours)</b>	$\times 2$	DIV2K+Flickr2K	<b>38.46</b>	<b>0.9624</b>	<b>34.61</b>	<b>0.9260</b>	<b>32.55</b>	<b>0.9043</b>	<b>33.95</b>	<b>0.9433</b>	<b>40.02</b>	<b>0.9800</b>
RCAN [95]	$\times 3$	DIV2K	34.74	0.9299	30.65	0.8482	29.32	0.8111	29.09	0.8702	34.44	0.9499
SAN [15]	$\times 3$	DIV2K	34.75	0.9300	30.59	0.8476	29.33	0.8112	28.93	0.8671	34.30	0.9494
IGNN [100]	$\times 3$	DIV2K	34.72	0.9298	30.66	0.8484	29.31	0.8105	29.03	0.8696	34.39	0.9496
HAN [63]	$\times 3$	DIV2K	34.75	0.9299	30.67	0.8483	29.32	0.8110	29.10	0.8705	34.48	0.9500
NLSA [61]	$\times 3$	DIV2K	34.85	0.9306	30.70	0.8485	29.34	0.8117	29.25	0.8726	34.57	0.9508
<b>SwinIR (Ours)</b>	$\times 3$	DIV2K	<b>34.89</b>	<b>0.9312</b>	<b>30.77</b>	<b>0.8503</b>	<b>29.37</b>	<b>0.8124</b>	<b>29.29</b>	<b>0.8744</b>	<b>34.74</b>	<b>0.9518</b>
<b>SwinIR+ (Ours)</b>	$\times 3$	DIV2K	<b>34.95</b>	<b>0.9316</b>	<b>30.83</b>	<b>0.8511</b>	<b>29.41</b>	<b>0.8130</b>	<b>29.42</b>	<b>0.8761</b>	<b>34.92</b>	<b>0.9526</b>
IPT [9]	$\times 3$	ImageNet	34.81	-	30.85	-	29.38	-	29.49	-	-	-
<b>SwinIR (Ours)</b>	$\times 3$	DIV2K+Flickr2K	<b>34.97</b>	<b>0.9318</b>	<b>30.93</b>	<b>0.8534</b>	<b>29.46</b>	<b>0.8145</b>	<b>29.75</b>	<b>0.8826</b>	<b>35.12</b>	<b>0.9537</b>
<b>SwinIR+ (Ours)</b>	$\times 3$	DIV2K+Flickr2K	<b>35.04</b>	<b>0.9322</b>	<b>31.00</b>	<b>0.8542</b>	<b>29.49</b>	<b>0.8150</b>	<b>29.90</b>	<b>0.8841</b>	<b>35.28</b>	<b>0.9543</b>
RCAN [95]	$\times 4$	DIV2K	32.63	0.9002	28.87	0.7889	27.77	0.7436	26.82	0.8087	31.22	0.9173
SAN [15]	$\times 4$	DIV2K	32.64	0.9003	28.92	0.7888	27.78	0.7436	26.79	0.8068	31.18	0.9169
IGNN [100]	$\times 4$	DIV2K	32.57	0.8998	28.85	0.7891	27.77	0.7434	26.84	0.8090	31.28	0.9182
HAN [63]	$\times 4$	DIV2K	32.64	0.9002	28.90	0.7890	27.80	0.7442	26.85	0.8094	31.42	0.9177
NLSA [61]	$\times 4$	DIV2K	32.59	0.9000	28.87	0.7891	27.78	0.7444	26.96	0.8109	31.27	0.9184
<b>SwinIR (Ours)</b>	$\times 4$	DIV2K	<b>32.72</b>	<b>0.9021</b>	<b>28.94</b>	<b>0.7914</b>	<b>27.83</b>	<b>0.7459</b>	<b>27.07</b>	<b>0.8164</b>	<b>31.67</b>	<b>0.9226</b>
<b>SwinIR+ (Ours)</b>	$\times 4$	DIV2K	<b>32.81</b>	<b>0.9029</b>	<b>29.02</b>	<b>0.7928</b>	<b>27.87</b>	<b>0.7466</b>	<b>27.21</b>	<b>0.8187</b>	<b>31.88</b>	<b>0.9423</b>
DBPN [31]	$\times 4$	DiF2K+Flickr2K	32.47	0.8980	28.82	0.7860	27.72	0.7400	26.38	0.7946	30.91	0.9137
IPT [9]	$\times 4$	ImageNet	32.64	-	29.01	-	27.82	-	27.26	-	-	-
RRDB [81]	$\times 4$	DIV2K+Flickr2K	32.73	0.9011	28.99	0.7917	27.85	0.7455	27.03	0.8153	31.66	0.9196
<b>SwinIR (Ours)</b>	$\times 4$	DIV2K+Flickr2K	<b>32.92</b>	<b>0.9044</b>	<b>29.09</b>	<b>0.7950</b>	<b>27.92</b>	<b>0.7489</b>	<b>27.45</b>	<b>0.8254</b>	<b>32.03</b>	<b>0.9260</b>
<b>SwinIR+ (Ours)</b>	$\times 4$	DIV2K+Flickr2K	<b>32.93</b>	<b>0.9043</b>	<b>29.15</b>	<b>0.7958</b>	<b>27.95</b>	<b>0.7494</b>	<b>27.56</b>	<b>0.8273</b>	<b>32.22</b>	<b>0.9273</b>

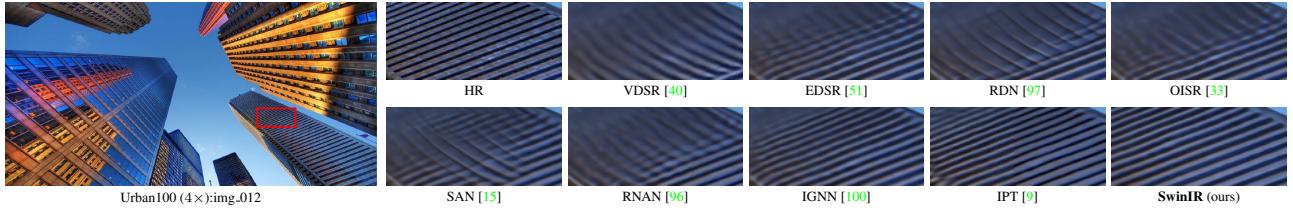


Figure 4: Visual comparison of **bicubic image SR** ( $\times 4$ ) methods. Compared images are derived from [9]. Best viewed by zooming.

Table 3: Quantitative comparison (average PSNR/SSIM) with state-of-the-art methods for **lightweight image SR** on benchmark datasets. Best and second best performance are in red and blue colors, respectively.

Method	Scale	#Params	#Mult-Add	Set5 [3]		Set14 [87]		BSD100 [58]		Urban100 [34]		Manga109 [60]	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
CARN [2]	$\times 2$	1,592K	222.8G	37.76	0.9590	33.52	0.9166	32.09	0.8978	31.92	0.9256	38.36	0.9765
FALSR-A [12]	$\times 2$	1,021K	234.7G	37.82	0.959	33.55	0.9168	32.1	0.8987	31.93	0.9256	-	-
IMDN [35]	$\times 2$	694K	158.8G	38.00	0.9605	33.63	0.9177	32.19	0.8996	32.17	0.9283	<b>38.88</b>	<b>0.9774</b>
LAPAR-A [44]	$\times 2$	548K	171.0G	38.01	0.9605	33.62	0.9183	32.19	0.8999	32.10	0.9283	38.67	0.9772
LatticeNet [57]	$\times 2$	756K	169.5G	<b>38.15</b>	<b>0.9610</b>	<b>33.78</b>	<b>0.9193</b>	<b>32.25</b>	<b>0.9005</b>	<b>32.43</b>	<b>0.9302</b>	-	-
<b>SwinIR (Ours)</b>	$\times 2$	878K	195.6G	<b>38.14</b>	<b>0.9611</b>	<b>33.86</b>	<b>0.9206</b>	<b>32.31</b>	<b>0.9012</b>	<b>32.76</b>	<b>0.9340</b>	<b>39.12</b>	<b>0.9783</b>
CARN [2]	$\times 3$	1,592K	118.8G	34.29	0.9255	30.29	0.8407	29.06	0.8034	28.06	0.8493	33.50	0.9440
IMDN [35]	$\times 3$	703K	71.5G	34.36	0.9270	30.32	0.8417	29.09	0.8046	28.17	0.8519	<b>33.61</b>	<b>0.9445</b>
LAPAR-A [44]	$\times 3$	544K	114.0G	34.36	0.9267	30.34	0.8421	29.11	0.8054	28.15	0.8523	33.51	0.9441
LatticeNet [57]	$\times 3$	765K	76.3G	<b>34.53</b>	<b>0.9281</b>	<b>30.39</b>	<b>0.8424</b>	<b>29.15</b>	<b>0.8059</b>	<b>28.33</b>	<b>0.8538</b>	-	-
<b>SwinIR (Ours)</b>	$\times 3$	886K	87.2G	<b>34.62</b>	<b>0.9289</b>	<b>30.54</b>	<b>0.8463</b>	<b>29.20</b>	<b>0.8082</b>	<b>28.66</b>	<b>0.8624</b>	<b>33.98</b>	<b>0.9478</b>
CARN [2]	$\times 4$	1,592K	90.9G	32.13	0.8937	28.60	0.7806	27.58	0.7349	26.07	0.7837	<b>30.47</b>	<b>0.9084</b>
IMDN [35]	$\times 4$	715K	40.9G	32.21	0.8948	28.58	0.7811	27.56	0.7353	26.04	0.7838	30.45	0.9075
LAPAR-A [44]	$\times 4$	659K	94.0G	32.15	0.8944	28.61	0.7818	27.61	0.7366	26.14	0.7871	30.42	0.9074
LatticeNet [57]	$\times 4$	777K	43.6G	<b>32.30</b>	<b>0.8962</b>	<b>28.68</b>	<b>0.7830</b>	<b>27.62</b>	<b>0.7367</b>	<b>26.25</b>	<b>0.7873</b>	-	-
<b>SwinIR (Ours)</b>	$\times 4$	897K	49.6G	<b>32.44</b>	<b>0.8976</b>	<b>28.77</b>	<b>0.7858</b>	<b>27.69</b>	<b>0.7406</b>	<b>26.47</b>	<b>0.7980</b>	<b>30.92</b>	<b>0.9151</b>



Figure 5: Visual comparison of **real-world image SR** ( $\times 4$ ) methods on RealSRSet [89]. Compared images are derived from [89].

Table 4: Quantitative comparison (average PSNR/SSIM/PSNR-B) with state-of-the-art methods for **JPEG compression artifact reduction** on benchmark datasets. Best and second best performance are in **red** and **blue** colors, respectively.

Dataset	$q$	ARCNN [17]	DnCNN-3 [90]	QGAC [20]	RNAN [96]	RDN [98]	DRUNet [88]	SwinIR (ours)
Classic5 [22]	10	29.03/0.7929/28.76	29.40/0.8026/29.13	29.84/0.8370/29.43	29.96/0.8178/29.62	30.00/0.8188/-	<b>30.16/0.8234/29.81</b>	<b>30.27/0.8249/29.95</b>
	20	31.15/0.8517/30.59	31.63/0.8610/31.19	31.98/0.8850/31.37	32.11/0.8693/31.57	32.15/0.8699/-	<b>32.39/0.8734/31.80</b>	<b>32.52/0.8748/31.99</b>
	30	32.51/0.8806/31.98	32.91/0.8861/32.38	33.22/0.9070/32.42	33.38/0.8924/32.68	33.43/0.8930/-	<b>33.59/0.8949/32.82</b>	<b>33.73/0.8961/33.03</b>
	40	33.32/0.8953/32.79	33.77/0.9003/33.20	-	34.27/0.9061/33.4	34.27/0.9061/-	<b>34.41/0.9075/33.51</b>	<b>34.52/0.9082/33.66</b>
LIVE1 [67]	10	28.96/0.8076/28.77	29.19/0.8123/28.90	29.53/0.8400/29.15	29.63/0.8239/29.25	29.67/0.8247/-	<b>29.79/0.8278/29.48</b>	<b>29.86/0.8287/29.50</b>
	20	31.29/0.8733/30.79	31.59/0.8802/31.07	31.86/0.9010/31.27	32.03/0.8877/31.44	32.07/0.8882/-	<b>32.17/0.8899/31.69</b>	<b>32.25/0.8909/31.70</b>
	30	32.67/0.9043/32.22	32.98/0.9090/32.34	33.23/0.9250/32.50	33.45/0.9149/32.71	33.51/0.9153/-	<b>33.59/0.9166/32.99</b>	<b>33.69/0.9174/33.01</b>
	40	33.63/0.9198/33.14	33.96/0.9247/33.28	-	34.47/0.9299/33.66	34.51/0.9302/-	<b>34.58/0.9312/33.93</b>	<b>34.67/0.9317/33.88</b>

applications, we further propose a large model and train it on much larger datasets. Experiments show that it can deal with more complex corruptions and achieves even better performance on real-world images than the current model. Due to page limit, the details are given in our project page <https://github.com/JingyunLiang/SwinIR>.

#### 4.4. Results on JPEG Compression Artifact Reduction

Table 4 shows the comparison of SwinIR with state-of-the-art JPEG compression artifact reduction methods: ARCNN [17], DnCNN-3 [90], QGAC [20], RNAN [96], RDN [98] and DRUNet [88]. All of compared methods are CNN-based models. Following [98, 88], we test different methods on two benchmark datasets (Classic5 [22] and LIVE1 [67]) for JPEG quality factors 10, 20, 30 and 40. As we can see, the proposed SwinIR has average PSNR gains of at least 0.11dB and 0.07dB on two testing datasets for different quality factors. Besides, compared with the previous best model DRUNet, SwinIR only has 11.5M parameters, while DRUNet is a large model that has 32.7M parameters.

#### 4.5. Results on Image Denoising

We show grayscale and color image denoising results in Table 5 and Table 6, respectively. Com-

pared methods include traditional models BM3D [14] and WNNM [29], CNN-based models DnCNN [90], IR-CNN [91], FFDNet [92], N3Net [65], NLRN [52], FOCNet [38], RNAN [96], MWCNN [54] and DRUNet [88]. Following [90, 88], the compared noise levels include 15, 25 and 50. As one can see, our model achieves better performance than all compared methods. In particular, it surpasses the state-of-the-art model DRUNet by up to 0.3dB on the large Urban100 dataset that has 100 high-resolution testing images. It is worth pointing out that SwinIR only has 12.0M parameters, whereas DRUNet has 32.7M parameters. This indicates that the SwinIR architecture is highly efficient in learning feature representations for restoration. The visual comparison for grayscale and color image denoising of different methods are shown in Figs. 6 and 7. As we can see, our method can remove heavy noise corruption and preserve high-frequency image details, resulting in sharper edges and more natural textures. By contrast, other methods suffer from either over-smoothness or over-sharpness, and cannot recover rich textures.

## 5. Conclusion

In this paper, we propose a Swin Transformer-based image restoration model SwinIR. The model is composed of three parts: shallow feature extraction, deep feature extrac-

Table 5: Quantitative comparison (average PSNR) with state-of-the-art methods for **grayscale image denoising** on benchmark datasets. Best and second best performance are in red and blue colors, respectively.

Dataset	$\sigma$	BM3D [14]	WNNM [29]	DnCNN [90]	IRCNN [91]	FFDNet [92]	N3Net [65]	NLRN [52]	FOCNet [38]	RNAN [96]	MWCNN [54]	DRUNet [88]	SwinIR (ours)
Set12 [90]	15	32.37	32.70	32.86	32.76	32.75	-	33.16	33.07	-	33.15	33.25	33.36
	25	29.97	30.28	30.44	30.37	30.43	30.55	30.80	30.73	-	30.79	30.94	31.01
	50	26.72	27.05	27.18	27.12	27.32	27.43	27.64	27.68	27.70	27.74	27.90	27.91
BSD68 [59]	15	31.08	31.37	31.73	31.63	31.63	-	31.88	31.83	-	31.86	31.91	31.97
	25	28.57	28.83	29.23	29.15	29.19	29.30	29.41	29.38	-	29.41	29.48	29.50
	50	25.60	25.87	26.23	26.19	26.29	26.39	26.47	26.50	26.48	26.53	26.59	26.58
Urban100 [34]	15	32.35	32.97	32.64	32.46	32.40	-	33.45	33.15	-	33.17	33.44	33.70
	25	29.70	30.39	29.95	29.80	29.90	30.19	30.94	30.64	-	30.66	31.11	31.30
	50	25.95	26.83	26.26	26.22	26.50	26.82	27.49	27.40	27.65	27.42	27.96	27.98

Table 6: Quantitative comparison (average PSNR) with state-of-the-art methods for **color image denoising** on benchmark datasets. Best and second best performance are in red and blue colors, respectively.

Dataset	$\sigma$	BM3D [14]	DnCNN [90]	IRCNN [91]	FFDNet [92]	DSNet [64]	RPCNN [85]	BRDNet [71]	RNAN [96]	RDN [98]	IPT [9]	DRUNet [88]	SwinIR (ours)
CBSD68 [59]	15	33.52	33.90	33.86	33.87	33.91	-	34.10	-	-	-	34.30	34.42
	25	30.71	31.24	31.16	31.21	31.28	31.24	31.43	-	-	-	31.69	31.78
	50	27.38	27.95	27.86	27.96	28.05	28.06	28.16	28.27	28.31	28.39	28.51	28.56
Kodak24 [23]	15	34.28	34.60	34.69	34.63	34.63	-	34.88	-	-	-	35.31	35.34
	25	32.15	32.14	32.18	32.13	32.16	32.34	32.41	-	-	-	32.89	32.89
	50	28.46	28.95	28.93	28.98	29.05	29.25	29.22	29.58	29.66	29.64	29.86	29.79
McMaster [94]	15	34.06	33.45	34.58	34.66	34.67	-	35.08	-	-	-	35.40	35.61
	25	31.66	31.52	32.18	32.35	32.40	32.33	32.75	-	-	-	33.14	33.20
	50	28.51	28.62	28.91	29.18	29.28	29.33	29.52	29.72	-	29.98	30.08	30.22
Urban100 [34]	15	33.93	32.98	33.78	33.83	-	-	34.42	-	-	-	34.81	35.13
	25	31.36	30.81	31.20	31.40	-	31.81	31.99	-	-	-	32.60	32.90
	50	27.93	27.59	27.70	28.05	-	28.62	28.56	29.08	29.38	29.71	29.61	29.82

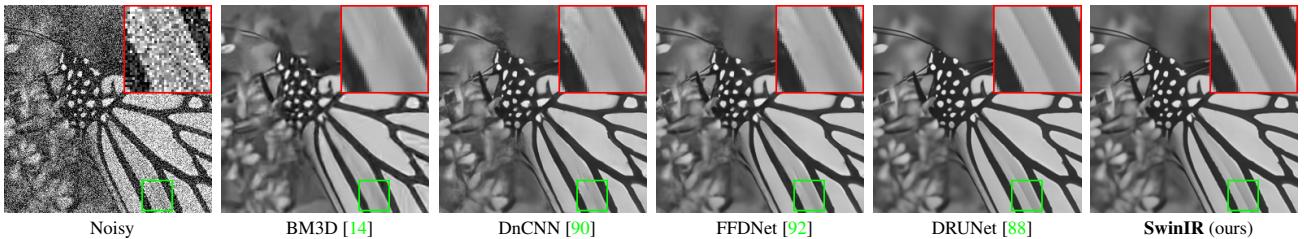


Figure 6: Visual comparison of **grayscale image denoising** (noise level 50) methods on image “Monarch” from Set12 [90]. Compared images are derived from [88].



Figure 7: Visual comparison of **color image denoising** (noise level 50) methods on image “163085” from CBSD68 [59]. Compared images are derived from [88].

tion and HR reconstruction modules. In particular, we use a stack of residual Swin Transformer blocks (RSTB) for deep feature extraction, and each RSTB is composed of Swin Transformer layers, convolution layer and a residual connection. Extensive experiments show that SwinIR achieves state-of-the-art performance on three representative image restoration tasks and six different settings: classic image SR, lightweight image SR, real-world image SR, grayscale image denoising, color image denoising and JPEG com-

pression artifact reduction, which demonstrates the effectiveness and generalizability of the proposed SwinIR. In the future, we will extend the model to other restoration tasks such as image deblurring and deraining.

**Acknowledgements** This paper was partially supported by the ETH Zurich Fund (OK), a Huawei Technologies Oy (Finland) project, the China Scholarship Council and an Amazon AWS grant. Special thanks goes to Yijue Chen.

## References

- [1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135, 2017. 4
- [2] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *European Conference on Computer Vision*, pages 252–268, 2018. 5, 6
- [3] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie line Alberi Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *British Machine Vision Conference*, pages 135.1–135.10, 2012. 1, 6
- [4] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. *arXiv preprint arXiv:2105.05537*, 2021. 2
- [5] Jiezhang Cao, Yawei Li, Kai Zhang, and Luc Van Gool. Video super-resolution transformer. *arXiv preprint arXiv:2106.06847*, 2021. 1, 2
- [6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 1, 2
- [7] Lukas Cavigelli, Pascal Hager, and Luca Benini. Cas-cnn: A deep convolutional neural network for image compression artifact suppression. In *2017 International Joint Conference on Neural Networks*, pages 752–759, 2017. 2
- [8] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *International Conference on Image Processing*, volume 2, pages 168–172. IEEE, 1994. 3
- [9] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12299–12310, 2021. 1, 2, 5, 6, 8
- [10] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1256–1272, 2016. 2
- [11] Wenlong Cheng, Mingbo Zhao, Zhiling Ye, and Shuhang Gu. Mfagan: A compression framework for memory-efficient on-device super-resolution gan. *arXiv preprint arXiv:2107.12679*, 2021. 2
- [12] Xiangxiang Chu, Bo Zhang, Hailong Ma, Ruijun Xu, and Qingyuan Li. Fast, accurate and lightweight super-resolution with neural architecture search. In *International Conference on Pattern Recognition*, pages 59–64. IEEE, 2020. 5, 6
- [13] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. *arXiv preprint arXiv:1911.03584*, 2019. 2
- [14] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007. 1, 7, 8
- [15] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11065–11074, 2019. 2, 5, 6
- [16] Xin Deng, Yutong Zhang, Mai Xu, Shuhang Gu, and Yiping Duan. Deep coupled feedback network for joint exposure fusion and image super-resolution. *IEEE Transactions on Image Processing*, 30:3098–3112, 2021. 2
- [17] Chao Dong, Yubin Deng, Chen Change Loy, and Xiaoou Tang. Compression artifacts reduction by a deep convolutional network. In *IEEE International Conference on Computer Vision*, pages 576–584, 2015. 2, 7
- [18] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision*, pages 184–199, 2014. 1, 2
- [19] 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
- [20] Max Ehrlich, Larry Davis, Ser-Nam Lim, and Abhinav Shrivastava. Quantization guided jpeg artifact correction. In *European Conference on Computer Vision*, pages 293–309, 2020. 7
- [21] Gamaleldin Elsayed, Prajit Ramachandran, Jonathon Shlens, and Simon Kornblith. Revisiting spatial invariance with low-rank local connectivity. In *International Conference on Machine Learning*, pages 2868–2879, 2020. 2, 3
- [22] 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
- [23] Rich Franzen. Kodak lossless true color image suite. source: <http://r0k.us/graphics/kodak>, 4(2), 1999. 8
- [24] Manuel Fritzsche, Shuhang Gu, and Radu Timofte. Frequency separation for real-world super-resolution. In *IEEE Conference on International Conference on Computer Vision Workshops*, pages 3599–3608, 2019. 5, 7
- [25] Xueyang Fu, Menglu Wang, Xiangyong Cao, Xinghao Ding, and Zheng-Jun Zha. A model-driven deep unfolding method for jpeg artifacts removal. *IEEE Transactions on Neural Networks and Learning Systems*, 2021. 2
- [26] Xueyang Fu, Zheng-Jun Zha, Feng Wu, Xinghao Ding, and John Paisley. Jpeg artifacts reduction via deep convolutional sparse coding. In *IEEE International Conference on Computer Vision*, pages 2501–2510, 2019. 2
- [27] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville,

- and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 3
- [28] Shuhang Gu, Nong Sang, and Fan Ma. Fast image super resolution via local regression. In *IEEE Conference on International Conference on Pattern Recognition*, pages 3128–3131, 2012. 1, 2
- [29] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *IEEE conference on computer vision and pattern recognition*, pages 2862–2869, 2014. 7, 8
- [30] Yong Guo, Jian Chen, Jingdong Wang, Qi Chen, Jiezhang Cao, Zeshuai Deng, Yanwu Xu, and Mingkui Tan. Closed-loop matters: Dual regression networks for single image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5407–5416, 2020. 2
- [31] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1664–1673, 2018. 5, 6
- [32] Kaiming He, Jian Sun, and Xiaou Tang. Single image haze removal using dark channel prior. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 33(12):2341–2353, 2010. 2
- [33] Xiangyu He, Zitao Mo, Peisong Wang, Yang Liu, Mingyuan Yang, and Jian Cheng. Ode-inspired network design for single image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1732–1741, 2019. 6
- [34] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2015. 6, 8
- [35] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *ACM International Conference on Multimedia*, pages 2024–2032, 2019. 5, 6
- [36] Takashi Isobe, Xu Jia, Shuhang Gu, Songjiang Li, Shengjin Wang, and Qi Tian. Video super-resolution with recurrent structure-detail network. In *European Conference on Computer Vision*, pages 645–660. Springer, 2020. 2
- [37] Xiaozhong Ji, Yun Cao, Ying Tai, Chengjie Wang, Jilin Li, and Feiyue Huang. Real-world super-resolution via kernel estimation and noise injection. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 466–467, 2020. 5, 7
- [38] Xixi Jia, Sanyang Liu, Xiangchu Feng, and Lei Zhang. Focnet: A fractional optimal control network for image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6054–6063, 2019. 2, 7, 8
- [39] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016. 3
- [40] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1646–1654, 2016. 1, 2, 6
- [41] Yoonsik Kim, Jae Woong Soh, Jaewoo Park, Byeongyong Ahn, Hyun-Seung Lee, Young-Su Moon, and Nam Ik Cho. A pseudo-blind convolutional neural network for the reduction of compression artifacts. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(4):1121–1135, 2019. 2
- [42] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 624–632, 2017. 2
- [43] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4681–4690, 2017. 1
- [44] Wenbo Li, Kun Zhou, Lu Qi, Nianjuan Jiang, Jiangbo Lu, and Jiaya Jia. Lapar: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond. *arXiv preprint arXiv:2105.10422*, 2021. 5, 6
- [45] 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
- [46] Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu. Feedback network for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3867–3876, 2019. 1
- [47] Dingkang Liang, Xiwu Chen, Wei Xu, Yu Zhou, and Xiang Bai. Transcrowd: Weakly-supervised crowd counting with transformer. *arXiv preprint arXiv:2104.09116*, 2021. 2
- [48] Jingyun Liang, Andreas Lugmayr, Kai Zhang, Martin Danelljan, Luc Van Gool, and Radu Timofte. Hierarchical conditional flow: A unified framework for image super-resolution and image rescaling. In *IEEE Conference on International Conference on Computer Vision*, 2021. 2
- [49] Jingyun Liang, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Mutual affine network for spatially variant kernel estimation in blind image super-resolution. In *IEEE Conference on International Conference on Computer Vision*, 2021. 2
- [50] Jingyun Liang, Kai Zhang, Shuhang Gu, Luc Van Gool, and Radu Timofte. Flow-based kernel prior with application to blind super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10601–10610, 2021. 2
- [51] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 136–144, 2017. 1, 4, 6
- [52] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. *arXiv preprint arXiv:1806.02919*, 2018. 2, 7, 8

- [53] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietik  inen. Deep learning for generic object detection: A survey. *International Journal of Computer Vision*, 128(2):261–318, 2020. 2
- [54] Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level wavelet-cnn for image restoration. In *IEEE conference on computer vision and pattern recognition workshops*, pages 773–782, 2018. 7, 8
- [55] Yun Liu, Guolei Sun, Yu Qiu, Le Zhang, Ajad Chhatkuli, and Luc Van Gool. Transformer in convolutional neural networks. *arXiv preprint arXiv:2106.03180*, 2021. 2
- [56] 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 preprint arXiv:2103.14030*, 2021. 1, 2, 4
- [57] Xiaotong Luo, Yuan Xie, Yulun Zhang, Yanyun Qu, Cuihua Li, and Yun Fu. Latticenet: Towards lightweight image super-resolution with lattice block. In *European Conference on Computer Vision*, pages 272–289, 2020. 5, 6
- [58] 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 *IEEE Conference on International Conference on Computer Vision*, pages 416–423, 2001. 6
- [59] 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 *IEEE International Conference on Computer Vision*, pages 416–423, 2001. 8
- [60] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76(20):21811–21838, 2017. 4, 5, 6
- [61] Yiqun Mei, Yuchen Fan, and Yuqian Zhou. Image super-resolution with non-local sparse attention. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3517–3526, 2021. 2, 5, 6
- [62] Tomer Michaeli and Michal Irani. Nonparametric blind super-resolution. In *IEEE Conference on International Conference on Computer Vision*, pages 945–952, 2013. 2
- [63] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Haifeng Shen. Single image super-resolution via a holistic attention network. In *European Conference on Computer Vision*, pages 191–207, 2020. 2, 4, 5, 6
- [64] 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. 2, 8
- [65] Tobias Pl  tz and Stefan Roth. Neural nearest neighbors networks. *arXiv preprint arXiv:1810.12575*, 2018. 7, 8
- [66] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. *arXiv preprint arXiv:1906.05909*, 2019. 2
- [67] HR Sheikh. Live image quality assessment database release 2. <http://live.ece.utexas.edu/research/quality>, 2005. 7
- [68] Wenzhe Shi, Jose Caballero, Ferenc Husz  r, 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1874–1883, 2016. 3
- [69] Guolei Sun, Yun Liu, Thomas Probst, Danda Pani Paudel, Nikola Popovic, and Luc Van Gool. Boosting crowd counting with transformers. *arXiv preprint arXiv:2105.10926*, 2021. 2
- [70] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *IEEE International Conference on Computer Vision*, pages 4539–4547, 2017. 2
- [71] Chunwei Tian, Yong Xu, and Wangmeng Zuo. Image denoising using deep cnn with batch renormalization. *Neural Networks*, 121:461–473, 2020. 8
- [72] Radu Timofte, Vincent De Smet, and Luc Van Gool. Anchored neighborhood regression for fast example-based super-resolution. In *IEEE Conference on International Conference on Computer Vision*, pages 1920–1927, 2013. 2
- [73] Radu Timofte, Vincent De Smet, and Luc Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Asian Conference on Computer Vision*, pages 111–126, 2014. 1, 2
- [74] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herv   J  gou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020. 1, 2
- [75] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. *arXiv preprint arXiv:2103.12731*, 2021. 2, 3
- [76] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 1, 2, 4
- [77] Longguang Wang, Yingqian Wang, Xiaoyu Dong, Qingyu Xu, Jungang Yang, Wei An, and Yulan Guo. Unsupervised degradation representation learning for blind super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10581–10590, 2021. 2
- [78] Longguang Wang, Yingqian Wang, Zhengfa Liang, Zaiping Lin, Jungang Yang, Wei An, and Yulan Guo. Learning parallax attention for stereo image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12250–12259, 2019. 2
- [79] Longguang Wang, Yingqian Wang, Zaiping Lin, Jungang Yang, Wei An, and Yulan Guo. Learning a single network for scale-arbitrary super-resolution. In *IEEE Conference on International Conference on Computer Vision*, pages 10581–10590, 2021. 2
- [80] Xiantao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution

- with pure synthetic data. *arXiv preprint arXiv:2107.10833*, 2021. 3
- [81] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *European Conference on Computer Vision Workshops*, pages 701–710, 2018. 1, 2, 3, 5, 6, 7
- [82] Zhendong Wang, Xiaodong Cun, Jianmin Bao, and Jianzhuang Liu. Uformer: A general u-shaped transformer for image restoration. *arXiv preprint arXiv:2106.03106*, 2021. 2
- [83] Yunxuan Wei, Shuhang Gu, Yawei Li, Radu Timofte, Longcun Jin, and Hengjie Song. Unsupervised real-world image super resolution via domain-distance aware training. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 13385–13394, 2021. 2
- [84] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision. *arXiv preprint arXiv:2006.03677*, 2020. 2
- [85] Zhihao Xia and Ayan Chakrabarti. Identifying recurring patterns with deep neural networks for natural image denoising. In *IEEE Winter Conference on Applications of Computer Vision*, pages 2426–2434, 2020. 8
- [86] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *arXiv preprint arXiv:2106.14881*, 2021. 2
- [87] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International Conference on Curves and Surfaces*, pages 711–730, 2010. 6
- [88] 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, 7, 8
- [89] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *IEEE Conference on International Conference on Computer Vision*, 2021. 1, 3, 5, 7
- [90] 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. 1, 2, 7, 8
- [91] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3929–3938, 2017. 1, 7, 8
- [92] 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. 1, 2, 7, 8
- [93] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3262–3271, 2018. 1, 2
- [94] Lei Zhang, Xiaolin Wu, Antoni Buades, and Xin Li. Color demosaicing by local directional interpolation and nonlocal adaptive thresholding. *Journal of Electronic imaging*, 20(2):023016, 2011. 8
- [95] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *European Conference on Computer Vision*, pages 286–301, 2018. 1, 2, 4, 5, 6
- [96] 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. 2, 6, 7, 8
- [97] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2472–2481, 2018. 1, 2, 6
- [98] 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. 2, 7, 8
- [99] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6881–6890, 2021. 2
- [100] Shangchen Zhou, Jiawei Zhang, Wangmeng Zuo, and Chen Change Loy. Cross-scale internal graph neural network for image super-resolution. *arXiv preprint arXiv:2006.16673*, 2020. 2, 5, 6