

DnSwin: Toward Real-World Denoising via a Continuous Wavelet Sliding Transformer

Hao Li^a, Zhijing Yang^a, Xiaobin Hong^{b,*}, Ziyang Zhao^c, Junyang Chen^a,
Yukai Shi^a, Jinshan Pan^d

^a*School of Information Engineering, Guangdong University of Technology, Guangzhou, 510006, China*

^b*School of Mechanical Automotive Engineering, South China University of Technology, Guangzhou, 510641, China*

^c*School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China*

^d*School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China*

Abstract

Real-world image denoising is a practical image restoration problem that aims to obtain clean images from in-the-wild noisy inputs. Recently, the Vision Transformer (ViT) has exhibited a strong ability to capture long-range dependencies, and many researchers have attempted to apply the ViT to image denoising tasks. However, a real-world image is an isolated frame that makes the ViT build long-range dependencies based on the internal patches, which divides images into patches, disarranges noise patterns and damages gradient continuity. In this article, we propose to resolve this issue by using a continuous Wavelet Sliding-Transformer that builds frequency correspondences under real-world scenes, called DnSwin. Specifically, we first extract the bottom features from noisy input images by using a convolutional neural network (CNN) encoder. The key to DnSwin is to extract high-frequency and low-frequency information from the observed features and build frequency dependencies. To this end, we propose a Wavelet Sliding-Window Transformer (WSWT) that utilizes the discrete wavelet transform (DWT), self-attention and the inverse DWT (IDWT) to extract deep features. Finally, we reconstruct the deep features into denoised images using a CNN decoder. Both quantitative and qualitative evaluations conducted

*Corresponding author

on real-world denoising benchmarks demonstrate that the proposed DnSwin performs favorably against the state-of-the-art methods.

Keywords: real-world image denoising, vision transformer, frequency correspondence, wavelet sliding window

1. Introduction

Image denoising is a fundamental topic in low-level vision fields and is widely applied in image restoration [1], super-resolution [2, 3], optical flow [4], *etc.* It also plays an important role in the preprocessing steps of high-level computer vision tasks, such as object detection and image segmentation [5, 6]. The purpose of image denoising is to restore a latent clean image x from its noisy observation y . Traditional image denoising can be formulated as follows:

$$y = x + n. \quad (1)$$

where n is the additive noise that is generally modeled as zero-mean white Gaussian noise with a standard deviation of σ . To address image degradation problems, many traditional image recovery models and learning-based methods have been developed in recent decades.

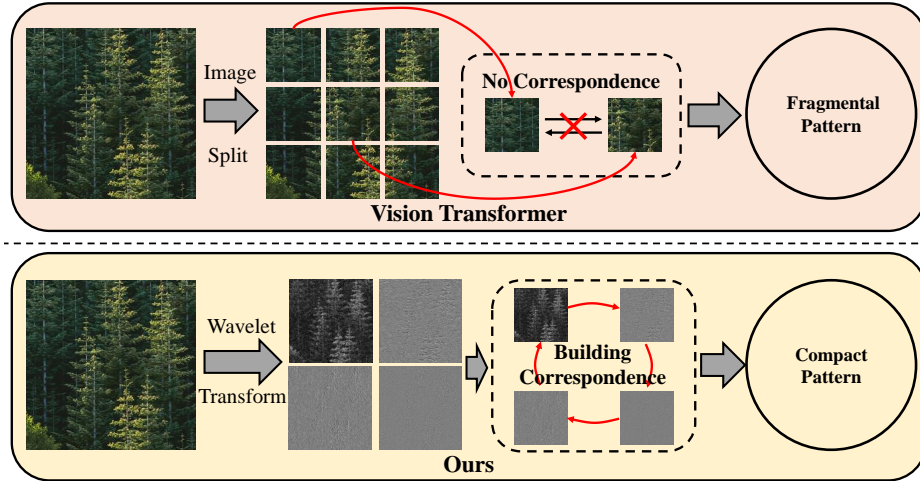


Figure 1: The tokenization strategy in the ViT roughly divides the input image into patches and breaks the image continuity. We address the **long-range correspondence** on a compact yet intact representation via a wavelet sliding-transformer.

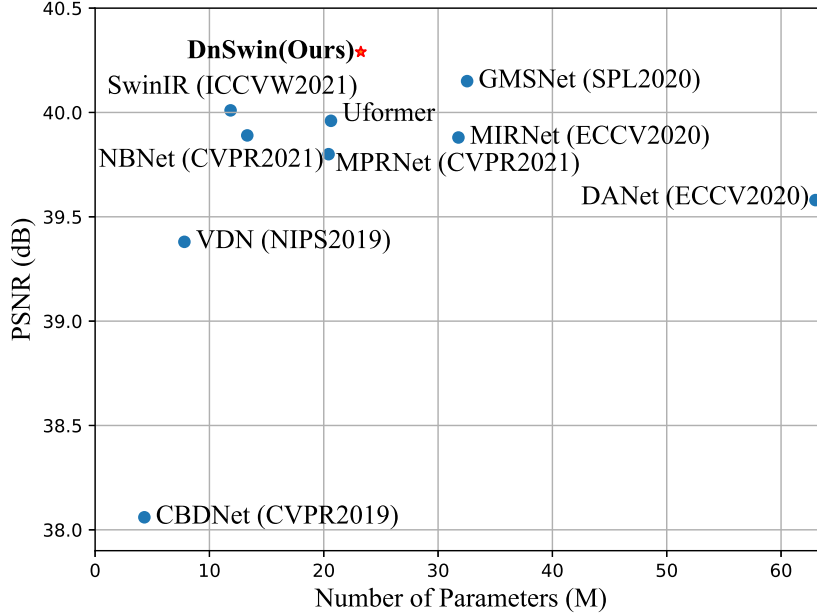


Figure 2: PSNR results vs. the total number of parameters for different methods in real-world image denoising tasks conducted on the DND [7] benchmark.

Traditional model-based solutions try to explore the prior image knowledge to regularize this essentially ill-posed degradation processing problem from a Bayesian viewpoint. In recent decades, various methods have been proposed by learning prior knowledge from nonlocal self-similarity [8], sparse representation [9], and total variation [10] models. Although the above methods have produced satisfactory results, they usually suffer from utilizing too many manually selected features to constrain images, which makes their optimization process time-consuming and limits their practicality.

Compared with model-based denoising methods, the existing learning-based restoration methods [11–14] address the above problem more efficiently and obtain charming visual quality when handling images with synthetic noise, such as additive white Gaussian noise (AWGN). However, real-world image denoising remains a challenging task, as the processing steps within camera systems, such as demosaicing, gamma correction and compression, are diversely constructed. In recent years, researchers have tried to build convolutional neural network (CNN)-based denoisers to handle real-world noisy photographs by modeling real noise [11, 15], obtaining noisy and noise-free image pairs [7, 16], and applying unsupervised/self-supervised

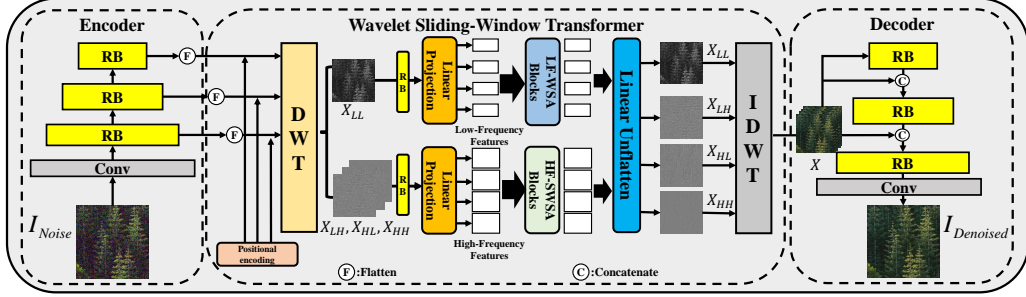


Figure 3: The overall architecture of DnSwin consists of an encoder module, a Wavelet Sliding-Window Transformer (WSWT) and a decoder module.

learning strategies [17, 18]. Typically, CNN-based real-world denoising approaches [19, 20] choose ‘UNet’ [21] as their backbone architecture. Nevertheless, the reduplicative downsampling operations in UNet cause serious damage to high-frequency information, which is consequential for image content preservation.

To address this problem, several recent works [22, 23] have attempted to utilize the vision transformer (ViT) for image restoration tasks. However, two challenges are involved in using the ViT for image restoration. First, the standard ViT structure [24] applies a global attention mechanism between all tokens. As shown in Figure 1, the ViT exhibits a strong ability to capture long-range dependencies. However, in a real-world denoising task, the real-world image is an isolated frame for which the ViT must build dependencies on the fragmental patches. A ViT-based denoiser realizes tokenization by cutting a large-size image into patches, which breaks the noise pattern. Second, previous works [25, 26] showed that the ViT has a limitation in capturing local dependencies, which is essential for image restoration tasks.

To address these issues, we propose a novel hybrid framework for image denoising, which is depicted in Figure 3. DnSwin consists of three encoder stages, three decoder stages and a wavelet sliding-window transformer (WSWT). In our model, we first utilize a CNN encoder to encode bottom features derived from noisy input images. As the discrete wavelet transform (DWT) can preserve the information contained in different frequency bands, which is important when recovering images, we apply the DWT to achieve the separation of low-frequency features and high-frequency features. Furthermore, we introduce low-frequency window-based self-attention (LF-SWSA) and high-frequency sliding window-based self-attention (HF-SWSA)

to build frequency correspondences on compact yet intact representations. Subsequently, we integrate the multifrequency subbands into features using the inverse DWT (IDWT). Finally, we reconstruct the features into denoised images using a CNN decoder.

Extensive experiments are conducted to evaluate DnSwin on real-world denoising tasks. The results obtained on four real-world denoising benchmarks show that DnSwin achieves state-of-the-art denoising performance on real-world scenes. Especially on the Darmstadt Noise Dataset (DND) [7] benchmark, as shown in Figure 2, our proposed DnSwin achieves obvious improvements over other state-of-the-art methods while utilizing suitable parameters. Generally, our contributions can be summarized as follows.

- We propose a wavelet ViT architecture for real-world image denoising. The proposed model captures long-range correspondences along the frequency subbands on compact yet intact image content. This strategy disentangles structural information from high- and low-frequency correspondences and provides a better interpretation for the ViT-based denoiser.
- We propose an LF-WSA block and an HF-SWSA block to make DnSwin intently exact features with different frequencies, especially the noise features in high frequency.
- Experimental results demonstrate that our model achieves state-of-the-art performance on real-world benchmarks in terms of peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM).

The remaining parts of this paper are organized as follows. Section 2 briefly surveys the advances achieved in deep denoising models dealing with AWGN and real-world noise and the progress made regarding the ViT. Section 3 presents the network architecture and detailed explanations of the LF-WSA and HF-SWSA. In Section 4, we perform experiments to verify the effectiveness and efficiency of DnSwin and compare it with existing state-of-the-art methods. Finally, we conclude our work and propose some future research directions in Section 5.

2. Related Work

Gaussian Denoising. As a fundamental low-level image processing challenge, image denoising has been intensively studied for decades. Most

traditional denoising methods can be interpreted within the maximum a posteriori (MAP) framework, e.g., total variation [10], sparse representation [9], wavelet coring [27] and nonlocal self-similarity [8]. These methods learn prior image model parameters via recognition and compact unrolled inference and introduce model-guided discriminatory learning. However, these methods tend to oversmooth image details. To this end, Mahdaoui et al. [28] proposed a compressed sensing (CS) reconstruction method that combines total variation regularization and the nonlocal self-similarity constraint to achieve significant improvements in terms of denoising efficiency and visual quality.

Benefiting from the success of deep learning, many researchers have focused on the exploration of CNN structures. Advanced architecture designs have achieved very competitive performance in comparison with traditional methods. For instance, Burger et al. [29] trained a plain multi-layer perceptron (MLP) with a large synthetic noise dataset, achieving comparable results to those of block matching and 3D filtering (BM3D). Chen et al. [30] proposed a trainable nonlinear reaction diffusion (TNRD) approach to perform truncated gradient descent inference. Zhang et al. [31] further proposed the denoising CNN (DnCNN), which forms a model by utilizing batch normalization (BN) [32] and residual connection [33]. Mao et al. [34] proposed the residual encoder-decoder network (REDNet), a deep fully convolutional encoding-decoding network with symmetric skip connections.

Driven by the advances in self-attention [35], many approaches try to extract the global features of images by using a self-attention mechanism to improve the resulting visual effects. Mei et al. [36] proposed a novel pyramid attention module for image restoration, which can capture long-range feature correspondences from a multi-scale feature pyramid. Zhang et al. [37] designed nonlocal attention blocks to capture global information and pay more attention to the challenging parts of images. However, this method leads to high memory occupation and time consumption levels.

Nevertheless, the above learning-based denoising approaches are all trained with synthetic noisy datasets, so they can only perform well when processing images with AWGN but significantly degrade when applied to real-world images [38]. Inspired by the aforementioned works, we propose a novel hybrid network architecture that inherits the superiority of a CNN and the ViT and demonstrates a stronger generalization ability.

Real-World Noisy Images. For real-world noisy images, the noise model is indeed distinct from AWGN. Actually, real noise is sophisticated, and the camera image signal processing (ISP) pipeline further increases its

complexity. As a remedy, existing deep denoisers for handling real-world noisy images are usually trained either by exploiting realistic noise models to synthesize noisy images or acquiring real paired noisy and noise-free images. The convolutional blind denoising network (CBDNet) [11] utilizes a realistic noise model including heterogeneous Gaussian and ISP pipelines; it is composed of two subnetworks, including a noise estimation module and a non-blind denoising module. Chen et al. proposed GCBD [39] to extract a smaller image with a clear background from a noisy image and then incorporated a generative adversarial network (GAN) to generate more fake noise samples for training the denoising CNN. RIDNet [40] uses a single-state denoising network with feature attention for real-world noise reduction. GRDN [41] was the champion of the sRGB track at NTIRE 2019 [42]; it incorporates groups of residual dense blocks (GRDBs) and convolutional block attention modules (CBAMs) for real-world image denoising.

However, the above approaches ignore the correspondence between multi-frequency features and fail to build local and global contexts; therefore, the complex noise information inside real-world images cannot be effectively removed. To this end, DnSwin utilizes a WSWT to decompose, extract, and integrate multi-frequency features, achieving state-of-the-art results on real-world denoising benchmarks.

Vision Transformer. Transformer [35] has achieved significant performance in natural language processing (NLP). Compared with a CNN, the multi-head self-attention (MSA) and feedforward MLP layer in a transformer are naturally good at capturing long-range dependencies between words via global self-attention. Due to the success of Transformer in NLP, many attempts have been made to explore the potential of transformers in computer vision tasks. The pioneering work on the ViT [43] obtained excellent results compared to those of state-of-the-art CNNs on image classification tasks by dividing images into 16×16 image patches and then stretching them into one-dimensional vectors that were fed into Transformer blocks. Chen et al. [44] proposed TransUnet, a hybrid framework using the ViT and UNet based on convolutional operations for medical image segmentation. Because of the limitation of the ViT in capturing local contexts, several works have tried to add convolutional layers to the MSA module [25, 45] and the feedforward network [26, 46]. Liu et al. [47] proposed the Swin Transformer, which significantly reduces the computational cost of the ViT by using shifted window-based multi-head attention mechanisms.

In addition to image classification and segmentation, Transformer has also

exhibited potential in image restoration tasks. Yang et al. [22] first proposed the texture Transformer network for image super-resolution (TTSR) and incorporated the ViT-based architecture into image restoration tasks. They used a cross-scale feature integration module to stack a texture transformer, thereby achieving a more powerful feature transformer. IPT [23] exploits the potential of pretraining and transfer learning with a shared ViT-based backbone. To address multiple image restoration tasks (e.g., super-resolution, denoising, and deraining), it employs each head and each tail separately for each image restoration task. However, the above approaches still need large-scale reference data for training. In contrast, DnSwin is able to achieve surprising performance with limited training samples by incorporating a wavelet CNN and the ViT into a combination-based architecture.

3. Methodology

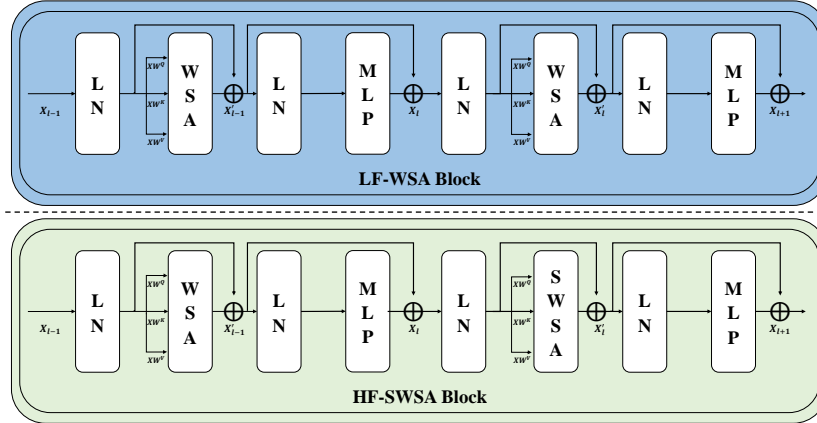


Figure 4: The details of the LF-WSA block and the HF-SWSA block.

In this section, we present our real-world image denoising network based on DnSwin. We first briefly describe the overall architecture details of DnSwin. Then, we provide more detailed explanations about bottom feature acquisition, the WSWT with three submodules (i.e., wavelet feature decomposition, frequency correspondence building and multi-frequency integration) and reconstruction.

3.1. Overall Architecture Details

As shown in Figure 3, DnSwin consists of three components: an encoder module, a WSWT and a decoder module. Specifically, the feature encoder module uses convolutional layers to extract the bottom features:

$$F_b^i = H_E(I_{noisy}), \quad (2)$$

where F_b^i denotes an extracted bottom feature, i is the index of the layer, H_E is the function of the feature encoder, and I_{noisy} refers to the noisy image. Then, we use the WSWT to separate information with different frequencies from the bottom feature and build the frequency dependencies:

$$F_d^i = H_W(F_b^i), \quad (3)$$

where H_W is the function of the WSWT and F_d^i denotes the deep feature extracted by the WSWT. The feature decoder uses convolution layers and a concatenation operator to fuse hierarchical features and reconstruct the image:

$$F_r^{i-1} = H_D(Concat(F_d^i \uparrow, F_d^{i-1})), \quad (4)$$

where F_r^i denotes the reconstruction feature obtained from the decoder (when $i = 0$, $F_r^0 = I_{Denoised}$ refers to the denoised image), H_D is the function of the feature decoder, and \uparrow is the upsampling operation.

3.2. Bottom Feature Acquisition

We utilize the encoder to execute bottom feature encoding for noisy input images. The feature encoder contains a convolutional layer and three stages of residual blocks (RBs). Each RB consists of three convolutional layers and two leaky ReLU layers. Specifically, given a noisy image $I_{noisy} \in \mathbb{R}^{3 \times H \times W}$ (H and W are the input image height and width, respectively), we first apply a 3×3 convolutional layer to extract features $\mathbf{X}^0 \in \mathbb{R}^{C \times H \times W}$, where C denotes the number of output channels. Next, the feature maps \mathbf{X}^l produced by the RBs [33] at different stages can be formally expressed as:

$$\begin{aligned} \mathbf{X}^0 &= \text{Conv}(I_{noisy}), \\ \mathbf{X}^i &= H_{RB}(\mathbf{X}^{i-1}) \downarrow. \end{aligned} \quad (5)$$

where Conv is the convolutional operation, $\{\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^i\} \in \mathbb{R}^{2^i C \times \frac{H}{2^i} \times \frac{W}{2^i}}$ denotes the output feature maps of the i -th stage, and $H_{RB}(\cdot)$ is the function of the RBs. In the downsampling layer \downarrow , we use a 4×4 convolution with a stride of 2, and the number of feature maps is doubled to reduce the information loss incurred during each downsampling step.

3.3. Wavelet Sliding-Window Transformer

Since the typical convolutional operator has a local receptive field, the feature encoder fails to capture the long-range dependencies of pixels. To this end, we propose the Wavelet Sliding-Window Transformer, which introduces a multi-frequency self-attention mechanism for long-range contextual extraction and correspondence modeling.

3.3.1. Wavelet Feature Decomposition

The DWT performs signal decomposition with attractive time-frequency localization properties that have the potential to decompose feature maps into different frequencies for further processing. More specifically, the DWT decomposes an image into a low-frequency subband and three high-frequency subbands. The low-frequency subband contains detailed information, while the high-frequency subbands contain structural information. Given the feature maps $\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^i\}$ obtained by feature acquisition, we first flatten them into patches, apply learnable positional encoding for spatial information modeling, and then use the DWT to decompose the patches into different frequency intervals by the following equations:

$$\begin{cases} \mathbf{X}_{LL} = (\mathbf{f}_{LL} * \mathbf{X}) \downarrow_2, \\ \mathbf{X}_{LH} = (\mathbf{f}_{LH} * \mathbf{X}) \downarrow_2, \\ \mathbf{X}_{HL} = (\mathbf{f}_{HL} * \mathbf{X}) \downarrow_2, \\ \mathbf{X}_{HH} = (\mathbf{f}_{HH} * \mathbf{X}) \downarrow_2, \end{cases} \quad (6)$$

where the subbands $\{\mathbf{X}_{LL}, \mathbf{X}_{LH}, \mathbf{X}_{HL}, \mathbf{X}_{HH}\}$ denote the approximation, horizontal detail, vertical detail and diagonal detail of the corresponding feature obtained using the Harr wavelet transform, respectively. $*$ is the convolution operator, and \downarrow_2 is the standard downsampling operator with a factor of 2. The low-pass filter \mathbf{f}_{LL} and three typical high-pass filters \mathbf{f}_{LH} , \mathbf{f}_{HL} , and \mathbf{f}_{HH} are defined as:

$$\begin{aligned} \mathbf{f}_{LL} &= \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, & \mathbf{f}_{LH} &= \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}, \\ \mathbf{f}_{HL} &= \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}, & \mathbf{f}_{HH} &= \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \end{aligned} \quad (7)$$

After we obtain $\{\mathbf{X}_{LL}, \mathbf{X}_{LH}, \mathbf{X}_{HL}, \mathbf{X}_{HH}\}$, we divide them into a low-frequency subband \mathbf{X}_{LL} and three high-frequency subbands $\{\mathbf{X}_{LH}, \mathbf{X}_{HL}, \mathbf{X}_{HH}\}$ and use RBs to extract the local features of the different frequency subbands.

To build correspondences between the different frequency subbands, we use LF-WSA and HF-SWSA to extract global features and capture long-range dependencies.

3.3.2. Building Frequency Correspondences via Sliding-Window

The details of LF-WSA and HF-SWSA are shown in Figure 4; they benefit from the self-attention mechanism in the transformer to capture long-range dependencies and build frequency correspondences between all subbands.

As shown in Figure 4, each HF-SWSA block consists of a layer normalization (LN) [48] layer, a multi-head windows-based self-attention (WSA) unit, a multi-head sliding window-based self-attention (SWSA) unit and the 2-layer MLP with the GELU non-linearity activation function; the LF-WSA block is modified from the HF-SWSA block by replacing an SWSA unit with a WSA unit. The computation of the HF-SWSA block is represented as:

$$\begin{aligned}\mathbf{X}'_l &= \text{WSA}(\text{LN}(\mathbf{X}_{l-1})) + \mathbf{X}_{l-1}, \\ \mathbf{X}_l &= \text{MLP}(\text{LN}(\mathbf{X}'_l)) + \mathbf{X}'_l, \\ \mathbf{X}'_{l+1} &= \text{SWSA}(\text{LN}(\mathbf{X}_l)) + \mathbf{X}_l, \\ \mathbf{X}_{l+1} &= \text{MLP}(\text{LN}(\mathbf{X}'_{l+1})) + \mathbf{X}'_{l+1}.\end{aligned}\tag{8}$$

where \mathbf{X}'_l and \mathbf{X}_l represent the outputs of the WSA and SWSA units and the MLP module in the l -th layer, respectively. Next, we introduce the details of the WSA and SWSA.

Windows based Self Attention. Given a feature map $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$, we first split the feature map into nonoverlapping windows with a window size of $M \times M$ and obtain the transposed features $\{\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^N\} \in \mathbb{R}^{M^2 \times C}$, $N = HW/M^2$ from each window. Then, we apply a self-attention mechanism on \mathbf{X}^N . Suppose that the number of heads is h and that the head dimensionality is $d = C/h$. The workflow of WSA can be formulated as:

$$\begin{aligned}\mathbf{X} &= \{\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^N\}, \quad N = HW/M^2, \\ \mathbf{Y}_k^n &= \text{Attention}(\mathbf{X}^n \mathbf{W}_k^Q, \mathbf{X}^n \mathbf{W}_k^K, \mathbf{X}^n \mathbf{W}_k^V), \quad n = 1, \dots, N, \\ \hat{\mathbf{X}}_k &= \{\mathbf{Y}_k^1, \mathbf{Y}_k^2, \dots, \mathbf{Y}_k^N\},\end{aligned}\tag{9}$$

where $\mathbf{W}_k^Q, \mathbf{W}_k^K, \mathbf{W}_k^V \in \mathbb{R}^{C \times d}$ denote the query, key and value window-based projection matrices for the k -th head, respectively. The outputs for all heads $\{\hat{\mathbf{X}}_1, \hat{\mathbf{X}}_2, \dots, \hat{\mathbf{X}}_h\}$ are concatenated and linearly projected to obtain the final result. Following previous works, the self-attention mechanism can be

formulated as:

$$\begin{aligned} & \text{Attention}(\mathbf{XW}^Q, \mathbf{XW}^K, \mathbf{XW}^V) \\ &= \text{SoftMax}\left(\frac{(\mathbf{XW}^Q)(\mathbf{XW}^K)^T}{\sqrt{d}} + B\right)(\mathbf{XW}^V), \end{aligned} \quad (10)$$

where B denotes the relative position bias, and the values in B are taken from the bias matrix $\hat{B} \in \mathbb{R}^{(2M-1) \times (2M-1)}$.

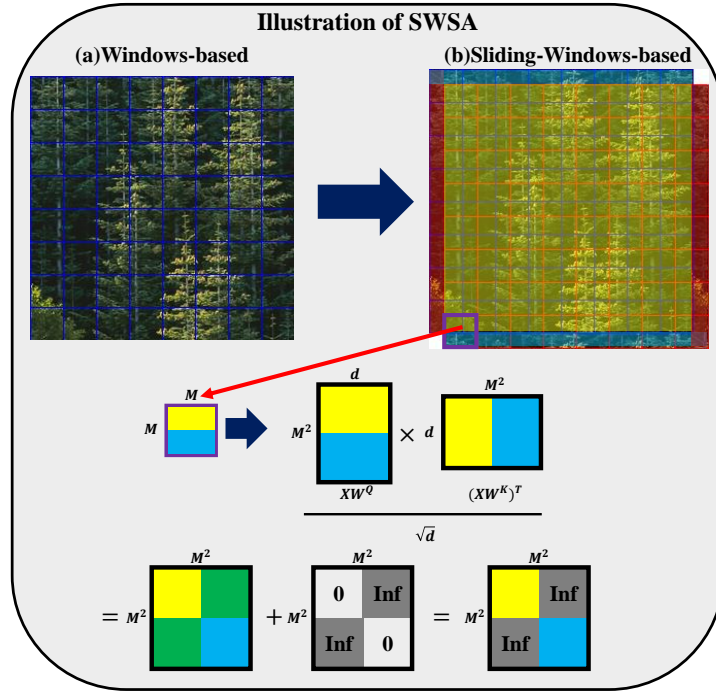


Figure 5: Illustration of SWSA.

As shown in Figure 3, for the low-frequency subband, we use LF-WSA blocks to extract features to reduce the computational cost. For high-frequency subbands, we replace the LF-WSA blocks with HF-SWSA blocks to further extract the noise features, as noise belongs to the high-frequency features.

Sliding-Windows based Self Attention. The WSA mechanism performs self-attention within each window and lacks cross-window connectivity, which limits its modeling capabilities. Based on WSA, we introduce a sliding mechanism to build connections across windows called the SWSA mechanism.

As shown in Figure 5(a), given a feature map $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$, we use the same window partitioning strategy as that of WSA. Then, we slide all windows to the bottom right by $(\lfloor \frac{M}{2} \rfloor, \lfloor \frac{M}{2} \rfloor)$ from the top-left pixel, through which the red part on the left side and the blue part on top in the feature move to the right and to the bottom, respectively, as shown in Figure 5(b). The SWSA process is divided into the following parts. 1) The sub-windows consisting entirely of yellow overlapping areas are similar to those in Eq. (10). 2) The sub-windows consisting of areas with different colors are illustrated at the bottom of Figure 5. Specifically, the color area (yellow, blue) in the feature remains the same color after applying the self-attention mechanism to the purple feature. In contrast, the green area obtained from the matrix calculation of the yellow and blue areas is meaningless, so we apply a mask mechanism to it to restrict the self-attention computation within each sub-window.

Sliding Mechanism. Benefiting from the sliding mechanism, three high-frequency subbands build correspondences that can effectively extract different noise features at high frequencies. More specifically, given the low-frequency feature map \mathbf{X}_{LL} and high-frequency feature maps $\{\mathbf{X}_{LH}, \mathbf{X}_{HL}, \mathbf{X}_{HH}\}$, we extract deep features from the LF-WSA and HF-SWSA mechanisms as follows:

$$\begin{aligned} \hat{\mathbf{X}}_{LL} &= H_{LF}(\mathbf{X}_{LL}), \\ \{\hat{\mathbf{X}}_{LH}, \hat{\mathbf{X}}_{HL}, \hat{\mathbf{X}}_{HH}\} &= H_{HF}(\{\mathbf{X}_{LH}, \mathbf{X}_{HL}, \mathbf{X}_{HH}\}). \end{aligned} \quad (11)$$

where $H_{LF}(\cdot)$ and $H_{HF}(\cdot)$ are the LF-WSA and HF-SWSA blocks, respectively.

Furthermore, WSA can significantly reduce the computational cost compared with that of global self-attention. Given the feature map $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$ and a window size of M , the computational complexity levels of the global MSA and WSA mechanisms are:

$$\begin{aligned} O(\text{MSA}) &= 4HWC^2 + 2(HW)^2C, \\ O(\text{WSA}) &= 4HWC^2 + 2M^2HWC. \end{aligned} \quad (12)$$

Difference from the Swin Transformer. In the Swin Transformer [47], tokenization is a necessary step, but it is inappropriate for large-size image denoising. If we simply apply the Swin Transformer to a real-world denoising task, the full-size image will be cut into fragmental patches, which undoubtedly breaks the image continuity and noise patterns. In the proposed DnSwin, we combine the wavelet operator with Transformer, which

preserves compact yet intact feature representations. Moreover, we investigate a sliding transformer to build long-range correspondences along the frequency subband on the unbroken image content. This strategy disentangles the structural information from the high- and low-frequency correspondences to better interpret real-world image denoising.

3.3.3. Multi-frequency Integration

As shown in Figure 3, since we design DnSwin as a hierarchical architecture, the LF-WSA blocks at the low-frequency feature map \mathbf{X}_{LL} work on larger receptive fields and are sufficient for learning long-range dependencies. The blocks at the high-frequency feature maps $\{\mathbf{X}_{LH}, \mathbf{X}_{HL}, \mathbf{X}_{HH}\}$ learn the different noise features and build connections between the three high-frequency subbands.

Nevertheless, we need to build correspondences between the low-frequency subband and the high-frequency subbands. Therefore, we propose applying the IDWT to the integration of multi-frequency subbands. Specifically, we first obtain the outputs \mathbf{X}_{LL} and $\{\mathbf{X}_{LH}, \mathbf{X}_{HL}, \mathbf{X}_{HH}\}$ of the LF-WSA blocks and HF-SWSA blocks, as described by Eq. (11); then, we unflatten and expand them to feature maps $\{\hat{\mathbf{X}}_{LL}, \hat{\mathbf{X}}_{LH}, \hat{\mathbf{X}}_{HL}, \hat{\mathbf{X}}_{HH}\}$. Finally, the multi-frequency feature maps $\{\hat{\mathbf{X}}_{LL}, \hat{\mathbf{X}}_{LH}, \hat{\mathbf{X}}_{HL}, \hat{\mathbf{X}}_{HH}\}$ are integrated by the IDWT, which can accurately reconstruct the original features from the wavelet subbands. The operation process of the IDWT is defined as follows:

$$\left\{ \begin{array}{l} \hat{\mathbf{X}}(2i-1, 2j-1) = (\hat{\mathbf{X}}_{LL}(i, j) - \hat{\mathbf{X}}_{LH}(i, j) \\ \quad - \hat{\mathbf{X}}_{HL}(i, j) + \hat{\mathbf{X}}_{HH}(i, j))/4, \\ \hat{\mathbf{X}}(2i-1, 2j) = (\hat{\mathbf{X}}_{LL}(i, j) - \hat{\mathbf{X}}_{LH}(i, j) \\ \quad + \hat{\mathbf{X}}_{HL}(i, j) - \hat{\mathbf{X}}_{HH}(i, j))/4, \\ \hat{\mathbf{X}}(2i, 2j-1) = (\hat{\mathbf{X}}_{LL}(i, j) + \hat{\mathbf{X}}_{LH}(i, j) \\ \quad - \hat{\mathbf{X}}_{HL}(i, j) - \hat{\mathbf{X}}_{HH}(i, j))/4, \\ \hat{\mathbf{X}}(2i, 2j) = (\hat{\mathbf{X}}_{LL}(i, j) + \hat{\mathbf{X}}_{LH}(i, j) \\ \quad + \hat{\mathbf{X}}_{HL}(i, j) + \hat{\mathbf{X}}_{HH}(i, j))/4, \end{array} \right. \quad (13)$$

where $\hat{\mathbf{X}}(i, j)$ is the (i, j) -th index of $\hat{\mathbf{X}}$. Feature maps at different stages $\hat{\mathbf{X}} = \{\hat{\mathbf{X}}^1, \hat{\mathbf{X}}^2, \dots, \hat{\mathbf{X}}^i\}$ are obtained by Eq. (13).

Benefiting from the IDWT, the multi-frequency subbands can be completely integrated into features $\hat{\mathbf{X}}^i$ that have the same dimensions as the input

features \mathbf{X}^i . Moreover, the **correspondences** between the low-frequency subband and the high-frequency subbands are established, thereby disentangling valuable structural information from the high- and low-frequency correspondences to achieve better real-world image denoising performance.

3.4. Reconstruction

The reconstruction phase employs three stages of RBs and a convolutional layer for feature reconstruction. The convolutional layers are adopted in the reconstruction phase for two reasons. 1) Although Transformer can be considered a specific instantiation of a spatially varying convolution [49, 50], convolutional layers with spatially invariant filters can enhance the translational equivalence of DnSwin. 2) The RBs and concatenation operator aggregate the features at different levels.

We use 2×2 transposed convolutions with strides of 2 for the upsampling layers. The upsampling layers double the sizes of the feature maps and reduce the number of channels by half. After completing the upsampling operation, we concatenate $\mathbf{X}^{\hat{i}-1}$ and $\mathbf{X}^{\hat{i}}$ together to reduce the impact of the spatial information caused by downsampling. The denoised image $I_{Denoised}$ can be reconstructed by the following equation:

$$\begin{cases} \hat{\mathbf{X}}^i = H_{RB}(\hat{\mathbf{X}}^i), \\ \mathbf{X}^{\hat{i}-1} = H_{RB}(\text{Concat}(\hat{\mathbf{X}}^i \uparrow, \mathbf{X}^{\hat{i}-1})), \\ \dots, \\ \hat{\mathbf{X}}^1 = H_{RB}(\text{Concat}(\hat{\mathbf{X}}^2 \uparrow, \hat{\mathbf{X}}^1)), \\ I_{Denoised} = \text{Conv}(\hat{\mathbf{X}}^1). \end{cases} \quad (14)$$

where \uparrow is the upsampling operation.

3.5. Loss Function

Instead of using the \mathcal{L}_1 loss, our model is optimized with the robust Charbonnier loss [51] to better handle outliers and achieve improved performance. The Charbonnier loss is defined as follows:

$$\mathcal{L}_{char} = \sqrt{\|I_{Denoised} - I_{GT}\|^2 + \varepsilon^2} \quad (15)$$

where I_{GT} represents the ground-truth image, and ε is an empirical value. Compared with the \mathcal{L}_1 loss, the \mathcal{L}_{char} loss makes the model more robust. In Section 4.4, we show that using the \mathcal{L}_{char} loss is superior to using the \mathcal{L}_1 loss.

4. Experiments

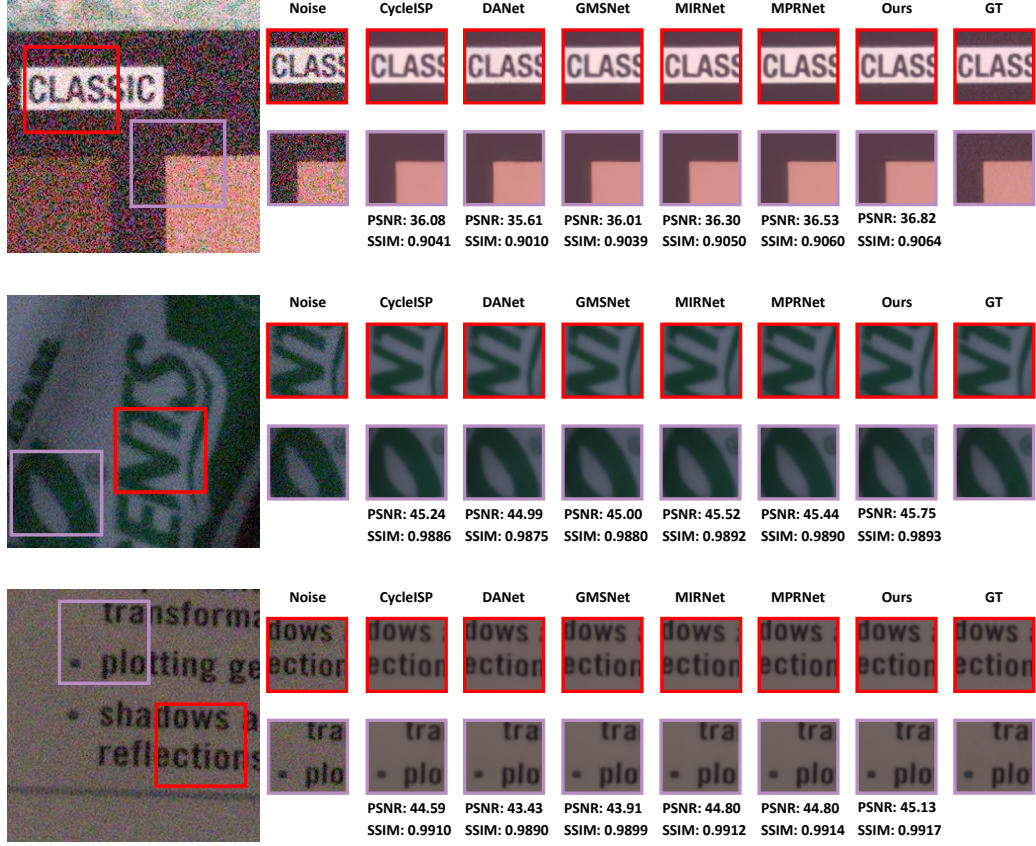


Figure 6: Comparison with state-of-the-art methods on real noisy images from the SIDD validation dataset [16] (**Zoom in for the best view**).

In this section, we first describe the experimental settings, including the training and testing settings and implementation details. Then, we compare our model with state-of-the-art methods on three real-world image denoising benchmarks (*i.e.*, the SIDD validation dataset, SIDD benchmark and DND benchmark). Finally, we evaluate the contributions of different DnSwin components and parameters through an ablation study and analyze the efficiency of our model.

4.1. Dataset and Evaluation Protocols

We use the following two real-world noisy-clean datasets for conducting comprehensive comparisons to validate our model:

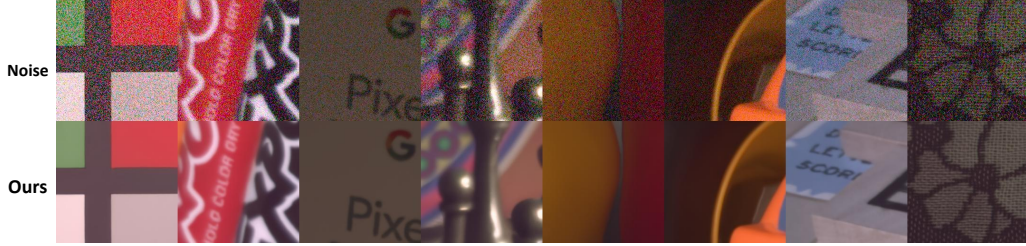


Figure 7: The visual results of DnSwin provided by SIDD benchmark online testing. Note that no ground-truth images or visual results of other state-of-the-art methods are included in the comparison (**Zoom in for the best view**).

- *SIDD* [16] consists of 30000 LR-HR image pairs, which were collected under different lighting conditions using five representative smartphone cameras, and their generated ground-truth images. For fast training and evaluation, we use the SIDD-Medium dataset (320 image pairs) for training and evaluate the approaches on the SIDD validation set and SIDD benchmark¹.
- *DND* [7] consists of 50 pairs of real noisy images and their corresponding ground-truth images, which were captured with consumer-grade cameras possessing different sensor sizes. However, the DND² does not provide any additional training data for fine-tuning denoising networks and can only be evaluated online.

In the evaluation protocols, we adopt the PSNR and SSIM to verify our model because they are the most commonly used evaluation metrics in image restoration, as they focus on pixel fidelity.

4.2. Implementation Details and Competing Methods

Following the common training of GMSNet [52], we use the Adam optimizer [53] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to train our framework. To improve the tradeoff between the size of the input patches and the available computing power, we set the batch size to 16 and the image patch size to 128×128 . The initial learning rate is 2×10^{-4} . We employ the cosine decay strategy to decrease the learning rate to 1×10^{-6} . For all experiments, we use flipping

¹<https://www.eecs.yorku.ca/~kamel/sidd/benchmark.php>

²https://noise.visinf.tu-darmstadt.de/benchmark/#results_srgb

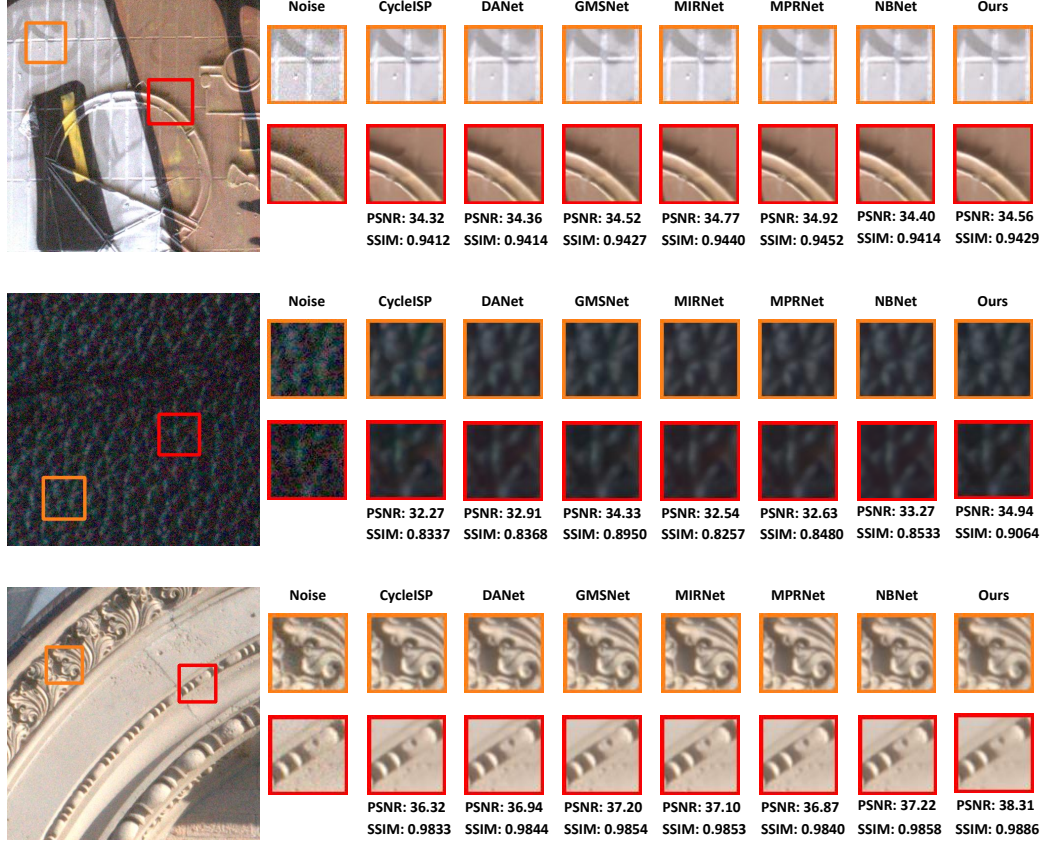


Figure 8: Comparison with state-of-the-art methods on real noisy images from the DND benchmark [7]. Notice that no ground-truth images are provided by the DND benchmark (**Zoom in for the best view**).

and random rotation with angles of 90° , 180° and 270° for data augmentation. For the LF-WSA and HF-SWSA blocks, we set the window size $M = 8$. All the experiments are carried out in a Linux environment with PyTorch (1.4.0) running on a server with 2 NVIDIA Tesla V100 GPUs. Nvidia CUDA 10.2 and cuDNN are utilized to accelerate the GPU computations. We utilize 6000 training epochs for DnSwin, which take approximately four days.

To prove the efficiency and superiority of DnSwin, we compare it with state-of-the-art approaches on a real-world image denoising task. In particular, we evaluate its real-world image denoising performance on three test sets, the SIDD validation dataset [16], SIDD benchmark and DND benchmark.

SIDD Validation Dataset and SIDD Benchmark. For the SIDD

Methods	PSNR \uparrow	SSIM \uparrow	Parameters
CBDNet [11]	30.78	0.801	4.31M
VDN [12]	39.28	0.956	7.81M
DANet [19]	39.47	0.957	63.01M
MIRNet [54]	39.72	0.959	31.18M
CycleISP [55]	39.52	0.957	-
MPRNet [56]	39.71	0.958	20.42M
NBNet [20]	39.75	<u>0.959</u>	13.31M
GMSNet [52]	39.63	0.956	32.54M
Uformer [57]	<u>39.77</u>	<u>0.959</u>	20.63M
SwinIR [58]	<u>39.77</u>	0.958	11.85M
Ours	39.80	0.960	23.24M

Table 1: Quantitative results on the SIDD validation dataset [16].

Methods	PSNR \uparrow	SSIM \uparrow	Parameters
CBDNet [11]	33.28	0.868	4.31M
VDN [12]	39.26	0.955	7.81M
DANet [19]	39.43	0.956	63.01M
GMSNet [52]	<u>39.51</u>	<u>0.958</u>	32.54M
Ours	39.65	0.959	23.24M

Table 2: Quantitative results on the SIDD benchmark [16].

validation dataset and SIDD benchmark, we follow the setting of DANet [19], which only uses the SIDD-Medium dataset for training. We compare our method with previously developed state-of-the-art methods, including CBDNet [11], VDN [12], DANet [19], MIRNet [54], CycleISP [55], MPNet [56], NBNet [20] and GMSNet [52]. It should be noted that no official test script is available for the corresponding test, so fewer baselines are used for comparison on the SIDD benchmark.

DND. For the DND benchmark, we train our model on the SIDD-Medium dataset and synthesize the noisy images provided by GMSNet [52] to conduct a fair comparison. The synthesized noisy images are generated by DIV2K [59]. The high-resolution images in DIV2K are cropped into nonover-

lapping image patches, and then noise is added to these patches. We randomly sample 320 image patches from the SIDD-Medium dataset and 100 image patches from the synthetic noisy dataset for each training iteration. We use the same baselines as those implemented on the SIDD validation dataset for comparison purposes.

Methods	PSNR \uparrow	SSIM \uparrow	Parameters
CBDNet [11]	38.06	0.942	4.31M
VDN [12]	39.38	0.952	7.81M
DANet [19]	39.58	0.955	63.01M
MIRNet [54]	39.88	0.956	31.18M
CycleISP [55]	39.56	0.956	-
MPRNet [56]	39.80	0.954	20.42M
NBNet [20]	39.89	0.955	13.31M
GMSNet [52]	<u>40.15</u>	<u>0.961</u>	32.54M
Uformer [57]	39.96	0.956	20.63M
SwinIR [58]	40.01	0.958	11.85M
Ours	40.29	0.962	23.24M

Table 3: Quantitative results on the DND benchmark [7].

Method	VDN	DANet	MIRNet	SwinIR	Ours
Time(frame/s)	0.1502	<u>0.0907</u>	0.1524	0.4726	0.0847
Parameters	7.81M	63.01M	31.78M	11.85M	23.24M

Table 4: Efficiency analysis for a 256×256 image from the SIDD validation dataset [16].

4.3. Quantitative and Qualitative Comparisons

SIDD Validation Dataset. As depicted in Table 1, the results indicate that our model achieves better performance than all the state-of-the-art methods. For instance, our method performs favorably against the recently developed state-of-the-art SwinIR method [58], where the PSNR and SSIM of the proposed results are 0.03 dB and 0.001 higher than those of SwinIR, respectively, justifying the effectiveness of DnSwin. Compared with

DANet [19], which uses a GAN [60], our model still achieves a 0.33 dB improvement in the PSNR index. Additionally, we present the visual comparison in Figure 6. It can be observed that our model recovers sharper and clearer images than those of the other methods and is therefore more faithful to the ground truths, which verifies the effectiveness of the local dependencies captured by the CNN and the long-range pixel dependencies captured by the WSWT.

SIDD Benchmark. In Table 2, we present the quantitative comparison results obtained on the SIDD benchmark. This shows that our model still achieves first place for all indices. Our model surpasses the prior state-of-the-art GMSNet method [52] by 0.14 dB and 0.001 in terms of the PSNR and SSIM, respectively. Some visual results obtained by DnSwin during online server testing are shown in Figure 7. The visual results show that DnSwin can not only successfully remove the noise but also preserve the texture details of images. It should be noted that the online server does not provide the ground truths and visual results of other state-of-the-art methods for comparison.

DND Benchmark. As depicted in Table 3, our model achieves a significant improvement over other state-of-the-art methods. Compared with MPRNet [56], our model demonstrates performance gains of 0.49 dB and 0.008 in terms of the PSNR and SSIM, respectively, justifying the notion that building frequency correspondences via a sliding window makes our model more robust when handling information with different frequencies. Although DANet utilizes many more parameters, our model achieves a 0.71 dB gain in terms of the PSNR index. Compared with the ViT-based methods, Uformer [57] and SwinIR [58], our model outperforms them by PSNRs margin of up to 0.30 dB. The qualitative results depicted in Figure 8 verify that our model achieves a significant visual quality improvement over CycleISP with a realistic texture and clear structure.

Efficiency Analysis. Despite achieving competitive results in quantitative comparisons, our model still exhibits outstanding running efficiency. The comparison in Table 4 shows that our model achieves a competitive efficiency level. Although VDN [12] has the fewest parameters, the noisy test image must be fed into two networks separately to obtain the final denoised image, which seriously reduces the inference speed of this method. Compared with MIRNet [54], our model has nearly $2\times$ the running efficiency and a significant restoration quality improvement. Compared with DANet [19], our model achieves an obvious quantitative improvement with a similar time consumption. Although SwinIR [58] has fewer parameters, it has the worst

running efficiency due to the computational complexity of the self-attention mechanism.

4.4. Ablation Study

In this section, we first conduct an ablation study on the ‘sliding mechanism’, ‘Convolution + Transformer’ and ‘Wavelet + Transformer’ to justify the effectiveness of our proposed components. Moreover, we conduct additional ablation studies on the number of filters, window size, and loss function using the auxiliary training dataset. All the ablation experiments are evaluated on the DND benchmark.

Test Set	DND [7]		
Metric	PSNR \uparrow	SSIM \uparrow	Parameters
Ours w/o LF-WSA	40.08	0.960	22.8M
Ours w/o HF-SWSA	40.03	0.958	22.7M
Ours Full	40.29	0.962	23.2M

Table 5: Ablation study on the effectiveness of the sliding mechanism.

Test Set	DND [7]		
Metric	PSNR \uparrow	SSIM \uparrow	Parameters
Standard UNet ($C = 32$)	39.48	0.956	9.5M
Ours ($C = 16$)	40.15	0.957	10.62M
Ours ($C = 32$)	40.29	0.962	23.2M

Table 6: Ablation study on the effectiveness of Convolution + Transformer.

Effects of the Sliding Mechanism. We conduct an ablation study on the sliding mechanism and present the experimental results. As depicted in Table 5, ‘Ours w/o LF-WSA’ achieves a 0.05 dB improvement, a 0.002 SSIM gain and a similar number of parameters in a comparison with ‘Ours w/o HF-SWSA’, which justifies the effectiveness of the sliding mechanism in real-world denoising scenarios. Moreover, compared with ‘Ours w/o HF-SWSA’, ‘Ours Full’ demonstrates better performance with 0.26 dB and 0.004 SSIM

Test Set	DND [7]		
Metric	PSNR \uparrow	SSIM \uparrow	Parameters
Standard UNet	39.48	0.956	9.5M
Ours w/o LF-WSA and HF-SWSA	39.97	0.958	22.3M
Ours w/o DWT and IDWT	40.15	0.960	19.5M
Ours Full	40.29	0.962	23.2M

Table 7: Ablation study on the effectiveness of Wavelet + Transformer.

Test Set	DND [7]		
Metric	PSNR \uparrow	SSIM \uparrow	Parameters
$C = 16$	40.15	0.957	10.62M
$C = 32$	40.29	0.962	23.24M
$C = 44$	40.28	0.962	30.56M
$C = 64$	40.18	0.960	48.52M

Table 8: Ablation study on the number of filters.

improvements, further justifying the effectiveness of the sliding mechanism and the effectiveness of the correspondences between the multi-frequency features obtained by incorporating the LF-WSA and HF-SWSA.

Convolution + Transformer. We conduct an analysis to verify the effectiveness of the convolutional operator and Transformer. Specifically, we replace all the LF-WSA and HF-SWSA blocks with ResBlocks [33] and remove the DWT and IDWT operations, resulting in ‘Standard UNet ($C = 32$)’, as shown in Table 6. We observe that ‘Ours ($C = 16$)’ achieves a 40.15 dB PSNR and outperforms ‘Standard UNet ($C = 32$)’ by 0.67 dB and 0.001 in terms of the PSNR and SSIM, respectively, with a closed number of parameters, which verifies that the architecture with ‘Convolution + Transformer’ possesses a better local and global context modeling ability than the pure CNN-based model. In addition, ‘Ours ($C = 32$)’ surpasses ‘Standard UNet ($C = 32$)’ by 0.81 dB and 0.006 in terms of the PSNR and SSIM, respectively, with more parameters, which indicates the effectiveness of ‘Convolution + Transformer’ in real-world denoising. We define C as the number of filters in the convolution layer in each model.

Test Set	DND [7]	
Metric	PSNR↑	Parameters
$M = 2$	39.52	8.81M
$M = 4$	39.96	14.52M
$M = 8$	40.29	23.24M
$M = 16$	40.31	40.46M

Table 9: Ablation study on the window size M .

Test Set	DND [7]	
Metric	PSNR↑	SSIM↑
$\mathcal{L} = \mathcal{L}_1$	40.06	0.958
$\mathcal{L} = \mathcal{L}_{char}$	40.29	0.962

Table 10: Ablation study on the loss function.

Effects of Wavelet + Transformer. To analyze the effectiveness of Wavelet + Transformer, we conduct an ablation study on different network architectures and observe their results obtained on the DND benchmark [7]. As shown in Table 7, compared with ‘Standard UNet’, ‘Ours w/o LF-WSA and HF-SWSA’ achieves limited a performance improvement with nearly $2.5\times$ the parameters, which certainly justifies the notion that simply decomposing features using the DWT and integrating them using the IDWT is useless for real-world image denoising yet brings a significant parameter overhead. Additionally, ‘Ours Full’ obviously surpasses ‘Ours w/o DWT and IDWT’ by 0.24 dB and 0.002 in terms of the SSIM, which justifies the effectiveness of Wavelet + Transformer. Moreover, ‘Ours Full’ exhibits the best PSNR and SSIM results, benefitting from the combination of multi-frequency feature decomposition, integration and between-feature correspondence building. Therefore, we apply the Wavelet + Transformer architecture in the proposed DnSwin.

Number of Filters. We conduct an experiment on the number of filters C . Table 8 shows that ‘ $C = 16$ ’ achieves a slightly worse performance than that obtained with ‘ $C = 32$ ’ but at a lower computational cost. In addition, ‘ $C = 44$ ’ yields a similar performance to that of ‘ $C = 32$ ’ with an obvious

Test Set	DND [7]	
Metric	PSNR \uparrow	SSIM \uparrow
\mathcal{D}_1 : denoising dataset synthesized by DIV2K	39.42	0.951
\mathcal{D}_2 : real denoising dataset SIDD-Medium	39.54	0.952
$\mathcal{D}_1 + \mathcal{D}_2$	40.29	0.962

Table 11: Ablation study on the auxiliary training dataset.

parameter increment. However, when ‘ $C = 64$ ’, the performance becomes significantly worse because the large amount of redundant information introduced by the unnecessary filters may hamper the network performance. Although the performance can be improved with a larger number of filters C , the computational complexity also increases, so we choose a compromise number of filters $C = 32$ to keep the computational complexity suitable while maintaining better performance.

Window Size. We perform an ablation study on the window size M . Due to patch boundary artifacts, the patch size (128) should be an integer multiple of the window size. As shown in Table 9, we choose $M = \{2, 4, 8, 16\}$ to verify the effects of different window size settings. When ‘ $M = 2$ ’ or ‘ $M = 4$ ’, the window is too small to capture the long-range dependencies of pixels, which leads to a significant drop in the model performance. Compared with ‘ $M = 8$ ’, ‘ $M = 16$ ’ only achieves only a marginal performance improvement while significantly increasing the number of model parameters. Based on the above analysis, we choose $M = 8$ as the default window size.

Loss Function. We compare different loss settings on the DND benchmark, as shown in Table 10. ‘ $\mathcal{L} = \mathcal{L}_{char}$ ’ achieves 40.29 dB, and ‘ $\mathcal{L} = \mathcal{L}_1$ ’ obtains 40.06 dB, justifying the effectiveness of the L_{char} loss in real-world image denoising. Moreover, L_{char} also yields a 0.004 SSIM gain over that of L_1 . To this end, we replace L_1 with L_{char} as the default loss function in the model optimization process.

Auxiliary Training Dataset. We also explore the impact of the auxiliary training dataset on the model performance. As depicted in Table 11, the results show that using the synthesized dataset (i.e., DIV2K \mathcal{D}_1) or the real-world denoising dataset (SIDD-Medium \mathcal{D}_2) leads to lower model performance. Synthetic denoising datasets are more important than real denoising datasets. Abdelhamed et al. [16] only collected image pairs in a few indoor

scenarios, which makes the trained model overfit these scenarios. However, our model achieves the best score when it is trained on $\mathcal{D}_1 + \mathcal{D}_2$, which verifies that auxiliary training datasets can effectively improve the real-world image denoising performance of the model.

5. Conclusion and Future Work

In this paper, we propose an effective real-world denoising ViT called DnSwin by incorporating a WSWT. We integrate the advantages of both a CNN and the ViT, not only taking advantage of the CNN to process images with large sizes due to its local receptive field but also utilizing a self-attention mechanism to capture long-range dependencies. Moreover, with wavelet feature decomposition, by building frequency correspondences between multi-frequency subbands and multi-frequency integration, our model can handle features with different frequencies, especially high-frequency noise features, and produces promising real-world image denoising results. Experimental results on real-world denoising benchmarks demonstrate that our model greatly surpasses the previous state-of-the-art methods with a suitable number of parameters and faster inference times on Euclidean-based evaluation protocols.

We aim to extend our work in the following directions. First, since real-world images have more unknown degradation, we would like to change our model to a GAN [60] model so that we can obtain results with better perceptual quality. Second, we are considering extending DnSwin to handle more general restoration tasks, such as image deblurring, deraining, and super-resolution.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (no. 62002069), Science and Technology Project of Guangdong Province (no. 2021A1515011341), Guangzhou Science and Technology Plan Project (no. 202002030386), and Guangdong Provincial Key Laboratory of Human Digital Twin (2022B1212010004).

References

- [1] Y. Huang, X. Hou, Y. Dun, J. Qin, L. Liu, X. Qian, L. Shao, Learning deformable and attentive network for image restoration, Knowledge-

- Based Systems 231 (2021) 107384. doi:<https://doi.org/10.1016/j.knosys.2021.107384>.
- [2] Z. Sun, J. Zhao, Z. Zhou, Q. Gao, L1 model-driven recursive multi-scale denoising network for image super-resolution, Knowledge-Based Systems 225 (2021) 107115. doi:<https://doi.org/10.1016/j.knosys.2021.107115>.
 - [3] H. Li, J. Qin, Z. Yang, P. Wei, J. Pan, L. Lin, Y. Shi, Real-world image super-resolution by exclusionary dual-learning, IEEE Transactions on Multimedia (2022).
 - [4] C. Bailer, B. Taetz, D. Stricker, Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 4015–4023.
 - [5] D. Liu, B. Wen, J. Jiao, X. Liu, Z. Wang, T. S. Huang, Connecting image denoising and high-level vision tasks via deep learning, IEEE Transactions on Image Processing 29 (2020) 3695–3706.
 - [6] M. Singh, M. C. Govil, E. S. Pilli, S. K. Vipparthi, Sod-ced: salient object detection for noisy images using convolution encoder–decoder, IET Computer Vision 13 (6) (2019) 578–587.
 - [7] T. Plötz, S. Roth, Benchmarking denoising algorithms with real photographs, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2750–2759. doi:[10.1109/CVPR.2017.294](https://doi.org/10.1109/CVPR.2017.294).
 - [8] M. Lebrun, A. Buades, J. M. Morel, A nonlocal bayesian image denoising algorithm, SIAM Journal on Imaging Sciences 6 (3) (2013) 1665–1688. doi:[10.1137/120874989](https://doi.org/10.1137/120874989).
 - [9] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Transactions on Image Processing 15 (12) (2006) 3736–3745. doi:[10.1109/TIP.2006.881969](https://doi.org/10.1109/TIP.2006.881969).
 - [10] L. I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D: Nonlinear Phenomena 60 (1) (1992) 259–268. doi:[https://doi.org/10.1016/0167-2789\(92\)90242-F](https://doi.org/10.1016/0167-2789(92)90242-F).

- [11] S. Guo, Z. Yan, K. Zhang, W. Zuo, L. Zhang, Toward convolutional blind denoising of real photographs, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 1712–1722. doi:10.1109/CVPR.2019.00181.
- [12] Z. Yue, H. Yong, Q. Zhao, D. Meng, L. Zhang, Variational denoising network: Toward blind noise modeling and removal, in: Advances in Neural Information Processing Systems, 2019, pp. 1690–1701.
- [13] C. Tian, Y. Xu, W. Zuo, B. Du, C.-W. Lin, D. Zhang, Designing and training of a dual cnn for image denoising, Knowledge-Based Systems 226 (2021) 106949.
- [14] X. Li, F. Zhou, H. Tan, Joint image fusion and denoising via three-layer decomposition and sparse representation, Knowledge-Based Systems 224 (2021) 107087.
- [15] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, J. T. Barron, Unprocessing images for learned raw denoising, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 11036–11045.
- [16] A. Abdelhamed, S. Lin, M. S. Brown, A high-quality denoising dataset for smartphone cameras, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 1692–1700.
- [17] X. Wu, M. Liu, Y. Cao, D. Ren, W. Zuo, Unpaired learning of deep image denoising, in: Proceedings of the European Conference on Computer Vision (ECCV), 2020, pp. 352–368.
- [18] T. Huang, S. Li, X. Jia, H. Lu, J. Liu, Neighbor2neighbor: Self-supervised denoising from single noisy images, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 14781–14790.
- [19] Z. Yue, Q. Zhao, L. Zhang, D. Meng, Dual adversarial network: Toward real-world noise removal and noise generation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2020, pp. 41–58.

- [20] S. Cheng, Y. Wang, H. Huang, D. Liu, H. Fan, S. Liu, Nbnnet: Noise basis learning for image denoising with subspace projection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 4896–4906.
- [21] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, 2015, pp. 234–241.
- [22] F. Yang, H. Yang, J. Fu, H. Lu, B. Guo, Learning texture transformer network for image super-resolution, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 5791–5800.
- [23] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, W. Gao, Pre-trained image processing transformer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 12299–12310.
- [24] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations, 2021.
- [25] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, L. Zhang, Cvt: Introducing convolutions to vision transformers, arXiv preprint arXiv:2103.15808 (2021).
- [26] Y. Li, K. Zhang, J. Cao, R. Timofte, L. Van Gool, Localvit: Bringing locality to vision transformers, arXiv preprint arXiv:2104.05707 (2021).
- [27] E. P. Simoncelli, E. H. Adelson, Noise removal via bayesian wavelet coring, in: Proceedings of 3rd IEEE International Conference on Image Processing, 1996, pp. 379–382. doi:10.1109/ICIP.1996.559512.
- [28] A. E. Mahdaoui, A. Ouahabi, M. S. Moulay, Image denoising using a compressive sensing approach based on regularization constraints, Sensors 22 (6) (2022).

- [29] H. C. Burger, C. J. Schuler, S. Harmeling, Image denoising: Can plain neural networks compete with bm3d?, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 2392–2399. doi:10.1109/CVPR.2012.6247952.
- [30] Y. Chen, T. Pock, Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (6) (2017) 1256–1272. doi:10.1109/TPAMI.2016.2596743.
- [31] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising, IEEE Transactions on Image Processing 26 (7) (2017) 3142–3155.
- [32] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, 2015, pp. 448–456.
- [33] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
- [34] X.-J. Mao, C. Shen, Y.-B. Yang, Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections, in: Advances in Neural Information Processing Systems, 2016, pp. 2810–2818.
- [35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
- [36] Y. Mei, Y. Fan, Y. Zhang, J. Yu, Y. Zhou, D. Liu, Y. Fu, T. S. Huang, H. Shi, Pyramid attention networks for image restoration, arXiv preprint arXiv:2004.13824 (2020).
- [37] Y. Zhang, K. Li, K. Li, B. Zhong, Y. Fu, Residual non-local attention networks for image restoration, in: International Conference on Learning Representations, 2019.

- [38] Y. Shi, H. Li, S. Zhang, Z. Yang, X. Wang, Criteria comparative learning for real-scene image super-resolution, *IEEE Transactions on Circuits and Systems for Video Technology* (2022).
- [39] J. Chen, J. Chen, H. Chao, M. Yang, Image blind denoising with generative adversarial network based noise modeling, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3155–3164.
- [40] S. Anwar, N. Barnes, Real image denoising with feature attention, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 3155–3164. doi:10.1109/ICCV.2019.00325.
- [41] D. Kim, J. R. Chung, S. Jung, GRDN: grouped residual dense network for real image denoising and gan-based real-world noise modeling, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 2086–2094.
- [42] A. Abdelhamed, R. Timofte, M. S. Brown, Ntire 2019 challenge on real image denoising: Methods and results, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 2197–2210.
- [43] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, D. Tran, Image transformer, in: *International Conference on Machine Learning*, 2018, pp. 4055–4064.
- [44] Chen, Jieneng and Lu, Yongyi and Yu, Qihang and Luo, Xiangde and Adeli, Ehsan and Wang, Yan and Lu, Le and Yuille, Alan L., and Zhou, Yuyin, Transunet: Transformers make strong encoders for medical image segmentation, *arXiv preprint arXiv:2102.04306* (2021).
- [45] S. d’Ascoli, H. Touvron, M. Leavitt, A. Morcos, G. Biroli, L. Sagun, Convit: Improving vision transformers with soft convolutional inductive biases, *arXiv preprint arXiv:2103.10697* (2021).
- [46] K. Yuan, S. Guo, Z. Liu, A. Zhou, F. Yu, W. Wu, Incorporating convolution designs into visual transformers, *arXiv preprint arXiv:2103.11816* (2021).

- [47] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, arXiv preprint arXiv:2103.14030 (2021).
- [48] J. L. Ba, J. R. Kiros, G. E. Hinton, Layer normalization (2016).
- [49] G. Elsayed, P. Ramachandran, J. Shlens, S. Kornblith, Revisiting spatial invariance with low-rank local connectivity, in: International Conference on Machine Learning, 2020, pp. 2868–2879.
- [50] A. Vaswani, P. Ramachandran, A. Srinivas, N. Parmar, B. Hechtman, J. Shlens, Scaling local self-attention for parameter efficient visual backbones, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 12894–12904.
- [51] P. Charbonnier, L. Blanc-Feraud, G. Aubert, M. Barlaud, Two deterministic half-quadratic regularization algorithms for computed imaging, in: Proceedings of 1st International Conference on Image Processing, 1994, pp. 168–172. doi:10.1109/ICIP.1994.413553.
- [52] Y. Song, Y. Zhu, X. Du, Grouped multi-scale network for real-world image denoising, IEEE Signal Processing Letters (2020) 2124–2128doi:10.1109/LSP.2020.3039726.
- [53] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
- [54] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, L. Shao, Learning enriched features for real image restoration and enhancement, in: Proceedings of the European Conference on Computer Vision (ECCV), 2020, pp. 492–511.
- [55] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, L. Shao, Cycleisp: Real image restoration via improved data synthesis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 2696–2705.
- [56] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, L. Shao, Multi-stage progressive image restoration, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 14821–14831.

- [57] Z. Wang, X. Cun, J. Bao, J. Liu, Uformer: A general u-shaped transformer for image restoration, arXiv preprint 2106.03106 (2021).
- [58] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, R. Timofte, Swinir: Image restoration using swin transformer, in: IEEE International Conference on Computer Vision Workshops, 2021.
- [59] E. Agustsson, R. Timofte, Ntire 2017 challenge on single image super-resolution: Dataset and study, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 126–135.
- [60] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.