EWT: Efficient Wavelet-Transformer for Single Image Denoising

Juncheng Li, Bodong Cheng, Ying Chen, Guangwei Gao, and Tieyong Zeng

***Abstract*—Transformer-based image denoising methods have achieved encouraging results in the past year. However, it must uses linear operations to model long-range dependencies, which greatly increases model inference time and consumes GPU storage space. Compared with convolutional neural network-based methods, current Transformer-based image denoising methods cannot achieve a balance between performance improvement and resource consumption. In this paper, we propose an Efficient Wavelet Transformer (EWT) for image denoising. Specifically, we use Discrete Wavelet Transform (DWT) and Inverse Wavelet Transform (IWT) for downsampling and upsampling, respectively. This method can fully preserve the image features while reducing the image resolution, thereby greatly reducing the device resource consumption of the Transformer model. Furthermore, we propose a novel Dual-stream Feature Extraction Block (DFEB) to extract image features at different levels, which can further reduce model inference time and GPU memory usage. Experiments show that our method speeds up the original Transformer by more than 80%, reduces GPU memory usage by more than 60%, and achieves excellent denoising results. All code will be public.**

***Index Terms*—Image denoising, vision Transformer, wavelet transform, dual-stream network, efficient model.**

I. Introduction

IMAGE denoising is a popular topic in image restoration (IR), which aims to reconstruct a clean image from the noisy one. As the key step in many practical applications, the quality of denoised images will significantly affect the performance of downstream tasks, such as image classification [1], [2], image segmentation [3], [4], target detection [5], [6]. However, due to the complex noise environment, image denoising is still a challenging inverse problem.

In the past few decades, researchers have made many explorations and attempts on single image denoising (SID).

This work was supported in part by the National Natural Science Foundation of China under Grants 61972212 and 61833011, the Shanghai Sailing Program under Grant 23YF1412800, and the Natural Science Foundation of Shanghai under Grant 23ZR1422200. (Juncheng Li and Bodong Cheng contributed equally to this work.)

Juncheng Li is with the School of Communication & Information Engineering, Shanghai University, Shanghai, China, also with Jiangsu Key Laboratory of Image and Video Understanding for Social Safety, Nanjing University of Science and Technology, Nanjing, China (E-mail: cvjunchengli@gmail.com).

Bodong Cheng is with the School of Computer Science and Technology, Xidian University, Xi'an, China. (E-mail: bdcheng@stu.xidian.edu.cn)

Y. Chen is with the Department of Cyberspace Security, Beijing Electronic Science Technology Institute, Beijing, China. (E-mail:ychen@besti.edu.cn)

Guangwei Gao is with the Institute of Advanced Technology, Nanjing University of Posts and Telecommunications, Nanjing, China, and also with the Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou, China (E-mail: csggao@gmail.com.

Tieyong Zeng is with the Department of Mathematics, The Chinese University of Hong Kong, New Territories, Hong Kong. (E-mail: zeng@math.cuhk.edu.hk)

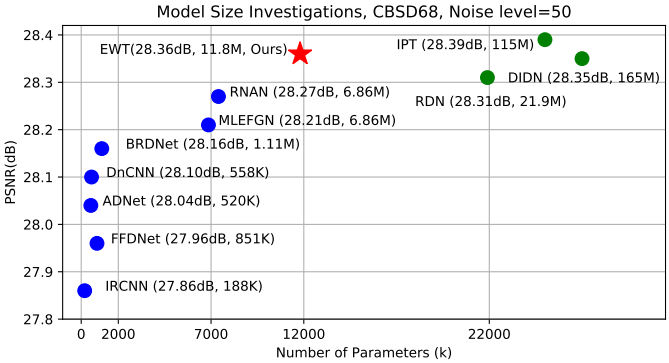

Fig. 1. Model performance and size comparison with classic single image denoising methods on CBSD68 (σ = 50).

The method of SID can be divided into traditional denoising methods [7]–[11] and learning-based methods. Among them, traditional methods are usually implemented in an iterative manner, which is inefficient. In addition, manual design is required and the generalization performance is poor. For learning-based methods, the purpose is to learn the mapping between noisy and clean images, thus making the model has denoising ability. Recently, with the wide application of deep learning in various fields and the excellent performance of convolutional neural networks (CNN) in computer vision, many CNN-based methods [12]–[17] have been proposed for SID. Most of them use the powerful feature extraction abilities of CNN to extract image features and use various strategies for modeling, which have achieved gratifying results. Recently, with the proposal and wide application of the visual Transformer model, a new research domain has been provided for SID. Facts have proved that the ability of Transformer to extract long-range dependencies of images makes it have better denoising performance than CNN model. Therefore, some representative image restoration Transformer models [18]–[21] have been proposed.

However, since the mechanism of Transformer is to use matrix operations to operate on the features of each pixel in the image, which will cause excessive consumption of time and space. Although the current Transformer-based image restoration method uses the patch processing method, dividing the image into multiple patches for operation, it still occupy a large amount of GPU memory space, resulting in longer inference time. Therefore, it is difficult to balance the performance and resource consumption of the model. The above problems make it difficult for the Transformer model to run on server devices with low GPU performance, which greatly limits the research

of Transformer on SID tasks.

To overcome the Transformer's bottleneck in image denoising, we propose a novel Efficient Wavelet Transformer (EWT). Although both DWT and Transformer are common technologies, as far as we know, this is the first model that introduced the wavelet transform into Transformer and applies it to image restoration task. It is worth mentioning that we do not forcefully combine them but elegantly integrate them according to their own advantages and disadvantages. EWT uses the reversible nature of wavelets as the sampling unit for model input and output, which can effectively improve the inference speed of the Transformer model and reduce a large amount of GPU memory usage. In the network backbone, we refer to the shift-windows self-attention mechanism in Swin Transformer, and combine the local feature extraction and aggregation capabilities of CNN to construct a dual-stream feature extraction block (DFEB) that combines the respective advantages of Transformer and CNN. In summary, the main contributions of this work are as follows:

- We consider the limitations of Transformer in image restoration tasks and propose a novel Efficient Wavelet-Transformer (EWT) for SID. This is the first attempt of Transformer in wavelet domain, which increases the speed of the original Transformer by more than **80%** and reduces GPU memory consumption by more than **60%**.
- We propose an efficient Multi-level Feature Aggregation Module (MFAM). MFAM is a lightweight feature aggregation module that can make full use of hierarchical features by using local and global residual learning. We also propose an elegant Dual-stream Feature Extraction Block (DFEB), which combines the advantages of CNN and Transformer that can take into account the information of different levels to better extract image features.
- We fully demonstrate the effectiveness of wavelets in Transformer models. Solve the drawbacks of the slow inference speed and high GPU memory usage of Transformer in image restoration tasks. In other words, EWT is a new attempt to balance model performance and resource consumption, which is helpful for more work in the future.

The rest of this paper is organized as follows. Related works are reviewed in Section II. A detailed explanation of the proposed EWT is given in Section III. The experimental results, ablation analysis, and discussion are presented in Section IV, V, and VI respectively. Finally, we draw a conclusion in Section VII.

II. Related Works

Recently, several Transformer methods for image denoising have been proposed to demonstrate the effectiveness of the Transformer architecture in this task. Although these methods have achieved good performance, they will occupy a large amount of GPU memory space and prolong the inference time of the network, which is extremely unfavorable for the promotion and application of Transformer in image restoration. In this paper, we aim to explore an efficient Transformer model for image denoising that considers both model performance and resource consumption.

A. CNN-based SID Methods

With the development of deep learning, CNN-based image restoration methods have achieved advanced results and greatly promoted the development of SID. The success of these methods is attributed to its powerful feature extraction ability and well-designed network structure, which can extract coarse and fine-grained features through different receptive fields. For example, Zhang et al. [12] proposed a DnCNN for the Gaussian noise removal, which achieved competitive results by took advantage of batch normalization and residual learning. Yang et al. [13] proposed a BM3D-Net, which is a nonlocal-based network that introduced BM3D into CNN by using wavelet shrinkage. Zhang et al. [14] proposed a flexible FFDNet, which took the noise level map and the noisy image as the inputs for image denoising. Fang et al. [22] proposed a multi-level edge features guided MLEFGN, which can make full use of edge features to reconstruct noise-free images. Zhang et al. [15] proposed an efficient Residual Dense Network (RDN) to extract abundant local features via densely connected convolutional layers. Most of the aforementioned methods committed to budding efficient modules to extract local features to reconstruct noise-free images. In addition, in order to restore more detailed features, many methods [16], [23] directly increase the depth of the network, which results in a substantial increase in the parameters of the model. To better encode image global information, the goal of current research is to explore more powerful deep learning models.

B. Transformer-based IR Methods

In order to model the dependency of pixel-level features, researchers began to pay attention to Transformer in NLP. The self-attention unit in Transformer can well model the long-distance dependencies in the sequence. However, due to the particularity of the image, directly expanding it into a sequence as the input of the Transformer will cause excessive computational overhead. In order to solve this problem, ViT [24] uses the idea of dividing an image into multiple sub-images of the same size. Later, in order to better promote the flow of information between sub-images, Swin Transformer [25] introduced the idea of window displacement to indirectly model the entire image and demonstrated its excellent performance in high-level vision tasks such as image classification and target detection. Recently, some works also apply Transformer to image restoration tasks, such as IPT [18] and SwinIR [19]. Among them, IPT draws on the network structure of DERT [26], which use 3×3 convolution with a step size of 3 to reduce the dimensionality of the image. This method can alleviate the dimensionality problem to a certain extent. However, the demanding requirements for GPU memory, training datasets, and reasoning time are unacceptable. SwinIR directly migrated the Swin Transformer to IR task and achieved outstanding results. However, SwinIR stacks a large number of Transformers, the execution time and GPU memory consumption are still very high. Although Transformer can improve the performance of the model, its own mechanism will bring a lot of GPU memory consumption and time overhead. In addition, Transformer cannot encode

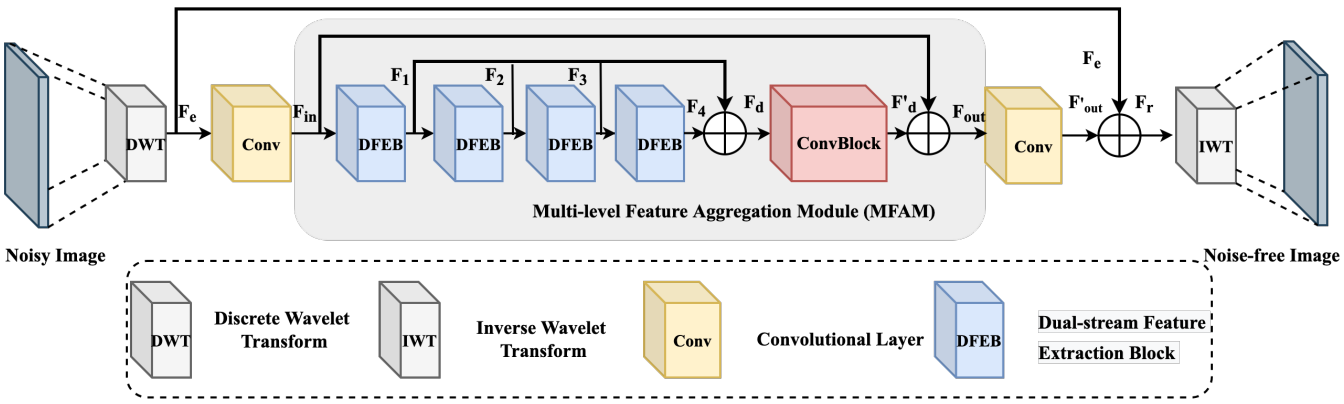

Fig. 2. The complete architecture of the proposed Efficient Wavelet-Transformer (EWT), Among them, MFAM is used for feature processing.

the two-dimensional position information of the image, and needs to embed relative position or absolute position encoding. In this regard, CNN inherently has the ability to encode the position of the image. Therefore, our goal is to incorporate CNNs and explore a more elegant and efficient Transformer for image restoration.

C. Wavelet-based IR Methods

Wavelet is widely used in image processing tasks. With the rise of deep learning, some studies combine wavelet with CNN and achieved excellent results. For example, Bae et al. [27] found that learning on wavelet sub-bands is more effective, and proposed a Wavelet Residual Network (WavResNet) for image restoration. After that, Bae et al. [28] also proposed a deep wavelet super-resolution network to recover the lost details on the wavelet sub-bands. Zhong et al. [29] jointed the sub-bands learning with CliqueNet [30] structures for wavelet domain super-resolution. Liu et al. [31] proposed a Multi-level Wavelet-CNN (MWCNN) for image restoration, which use multi-level wavelet to complete related tasks. Inspired by these methods, we intend to explore the performance of Transformer in the wavelet domain and build a more lightweight Transformer model with wavelet.

III. Efficient Wavelet-Transformer (EWT)

A. Network Architecture

As shown in Fig. 2, EWT mainly consists of three parts: Discrete Wavelet Transform (DWT), feature processing, and Inverse Wavelet Transform (IWT). Specifically, at the top of the model, we first use the DWT to downsample the image, which can effectively extract the high and low-frequency information of the image while reducing the resolution of the image. In the middle part of the model, a Multi-level Feature Aggregation Module (MFAM) is introduced for feature processing. This module can significantly improve the model inference speed while ensuring effective feature extraction. Finally, we use the IWT to restore the image and reconstruct its corresponding noise-free image. Define $I_{noisy} \in H \times W \times C$ as the original input noisy image, the DWT down-sampling layer f_{DWT} will convert I_{noisy} into 4 wavelet sub-images:

$$I_{LL}, I_{LH}, I_{HL}, I_{HH} = f_{DWT}(I_{noisy}), \tag{1}$$

where $I_{LL}, I_{LH}, I_{HL}, I_{HH} \in \frac{H}{2} \times \frac{W}{2} \times C$ are 4 sub-images with different frequencies. We concatenate them as the shallow features $F_e \in \frac{H}{2} \times \frac{W}{2} \times 4C$ of EWT, and then use them for feature extraction:

$$F_{in} = f_{conv}(F_e), \tag{2}$$

$$F_{out} = f_{MFAM}(F_{in}), \tag{3}$$

where $f_{conv}(\cdot)$ is a 3×3 convolutional layer used to extract the basic information of the image as the initial features. And these features are sent to MFAM to further extract more effective features. After that, a 3×3 convolutional layer also applied on the output F_{out} to obtain the merged features $F^{'}_{out}$:

$$F^{'}_{out} = f_{conv}(F_{out}), \tag{4}$$

and the global residual learning strategy is used to aggregate F_e and $F^{'}_{out}$ as the finally reconstructed feature

$$F_r = F_e + F^{'}_{out}. \tag{5}$$

Finally, the IWT operation is used to transform the features to the original resolution and reconstruct the noise-free image

$$I^{'}_{clean} = f_{IWT}(F_r), \tag{6}$$

where $f_{IWT}(\cdot)$ denotes inverse wavelet and $I^{'}_{clear}$ is the reconstruct clean image.

During training, EWT is optimized with $L1$ loss function. Given a training dataset $\left\{I^i_{noisy}, I^i_{clean}\right\}_{i=1}^{S}$, we solve

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{S} \sum_{i=1}^{S} \left\| F_\theta(I^i_{noisy}) - I^i_{clean} \right\|_1, \tag{7}$$

where θ denotes the parameter set of our EWT, $F(I_{noisy}) = I^{'}_{clean}$ is the reconstruct noise-free image.

B. Wavelet-based Image Sampling

Effective sampling of an image is a necessary considered problem in image restoration tasks since the resolution of the input image is usually very large. This means that it will take a lot of calculation costs to deal with them. Although the image size can be reduced by cropping, it will result in the inability to capture the global information of the image. To solve this

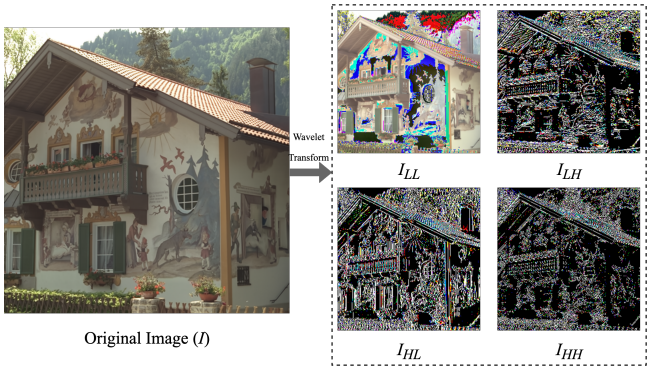

Fig. 3. The schematic diagram of Discrete Wavelet Transform (DWT).

problem, many methods have been proposed to reduce the image resolution, such as pooling or convolution operations. For image restoration tasks, the final output needs to be restored to the original image size. However, aforementioned operations will cause irreversible loss of information. To address this issue, we introduced wavelet to replace the down-sampling operation thus reduce the image resolution.

As shown in Fig. 3, when the Discrete Wavelet Transform (DWT) is applied to the image, the original image will be decomposed into four sub-images. Plenty of previous works have pointed out that these sub-bands have different frequencies, which mainly reflect the color of the filled area and the edge of the object. Specifically, I_{LL} is the low-frequency information sub-band of the image, which is an approximation of the original image. I_{LH} and I_{HL} are the horizontal and vertical sub-bands of the image, reflecting the edge characteristics of these two directions. I_{HH} is the diagonal sub-band of the image, reflecting the diagonal edge feature. Taking these sub-images as inputs of the model can guide the model to pay attention to frequency information and help to restore the texture details. Meanwhile, the connection between each sub-image can be established by the deep neural network, so that the model can extract deeper information. **Moreover, the wavelet is reversible and will not cause any loss of information, which is conducive to image restoration.** Therefore, we use Discrete Wavelet Transform (DWT) as the down-sampling module and use Inverse Wavelet Transform (IWT) as the up-sampling module in our EWT. In summary, the advantages of this method are: (1). The wavelet is reversible, so all information can be preserved through this sampling method; (2). Wavelet can capture the frequency and position information of the image, which is beneficial to restore the detailed features of the image; (3). Using wavelet can reduce the image resolution thus reducing the GPU memory consumption. Meanwhile, this process will not produce redundant parameters and can speed up the inference speed of the model, which benefit for efficient model building. (4). Wavelet will relatively increase the receptive area of the receptive field, so that the model can obtain richer features, which is benefit for image restoration.

C. Multi-level Feature Aggregation Module

As the core component of the entire model, Multi-level Feature Aggregation Module (MFAM) is specially designed for feature extraction and aggregation in the wavelet domain. As shown in Fig. 2, MFAM consists of a series of DFEBs and a ConvBlock, which are responsible for the extraction and aggregation of features at different levels of the image, respectively. Different from the current methods simply stacking Transformer layers, we carefully design a double-branched structural unit (DFEB), and adopt the dense connection to combine the outputs of each DFEB. In this way, the hierarchical features of the model can be better aggregated to enhance the feature representation. Then, a ConvBlock is applied to incorporate these features:

$$F_d = \sum_{i=1}^{N} F_i, \tag{8}$$

$$F_d^{'} = f_{ConvB}(F_d), \tag{9}$$

where F_i represents the output of the i-th DFEB, f_{ConvB} denotes the ConvBlock, and $F_d^{'}$ denotes the aggregated features. Finally, the global residual learning strategy is applied

$$F_{out} = F_{in} + F_d^{'}. \tag{10}$$

Dual-stream Feature Extraction Block (DFEB): Most Transformer-based methods limit the use of convolutional layers and only use it for feature aggregation or downsampling. However, we found that if the proportion of Transformer is too high, the model performance and resource consumption will be seriously unbalanced. This is because there are matrix operations on large tensors in Transformer, which will consume a huge of GPU computing and storage resources

$$Attention(Q, K, V) = Softmax(Norm(QK^T))V. \tag{11}$$

Our experiments also show that staking a large number of Transformers will not significantly improve the model performance. On the contrary, it will greatly increase the calculation time and GPU memory consumption of the model. Meanwhile, we find that the CNN-based method is significantly faster than the Transformer-based method. Moreover, as the most widely used neural network in computer vision, CNN has been well proven to have the natural ability to capture image information. In particular, CNNs can extract the positional information of images without the need for additional positional encoding embeddings while Transformer does not have the ability to encode location information. Although most visual Transformers have embedded the position-coding operation, most of these operations are designed by human intuition. Compared to the ability of CNN to automatically learn location information, this is far from enough. Therefore, directly replacing CNN with Transformer is a sub-optimal solution. In this work, we focus on elegantly combining CNN and Transformer to find a better solution.

Inspired by the idea of multi-scale feature extraction, we find that the multi-branch structure can better guide the model to learn information at different scales. In addition, the parallelism of the multi-branch structure allows each branch to extract different features without interfering with each other, reducing the information dilution problem caused by excessive stacking of neural network modules. In a multi-scale

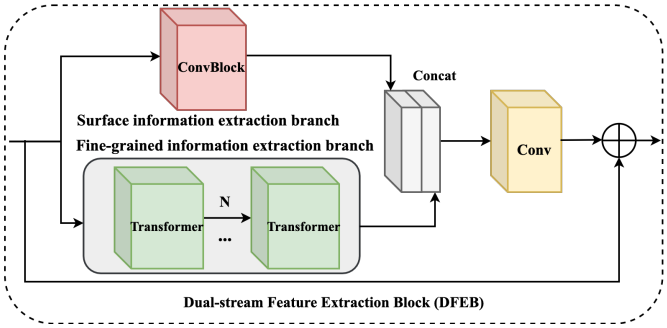

Fig. 4. The complete architecture of Dual-stream Feature Extraction Block.

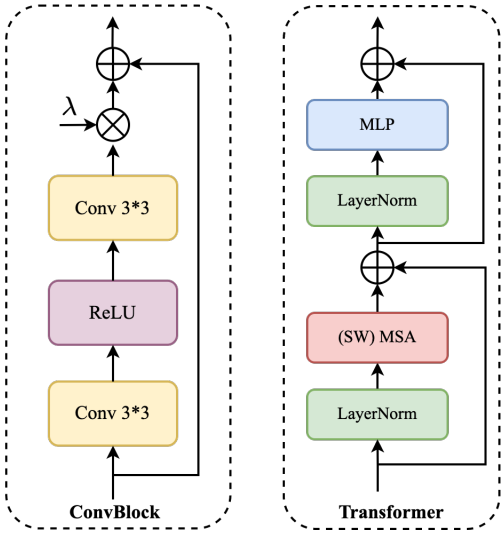

Fig. 5. The complete architecture of ConvBlock and Transformer.

CNN network, each branch is usually assigned a convolution kernel with different size to obtain different features under multiple receptive fields. In this work, we use Transformer as an alternative to multiple receptive fields. Specifically, we use Transformer and CNN as two branches to extract different features respectively, because CNN has strong local feature extraction ability and Transformer has better global encoding ability. Based on the above ideas, we designed a Dual-stream Feature Extraction Block (DFEB). DFEB is the most important component of MFAM, which is a dual-branch feature extraction module. The purpose of DFEB is to extract different levels of information and aggregate them to improve the expressive ability of the model. As shown in Fig. 4, DFEB contains two branches: surface information extraction branch and fine-gained information branch. When the features are sent to the DFEB, it will be divided into two groups, one group is used to extract rough features, and the other one is used to model the relationship among pixels and to learn the global information. Specifically, the surface information extraction branch only contains a ConvBlock (Fig. 5), which is a simple module composed of two convolutional layers and a ReLU activation function. This structure benefit for image restoration, especially for image surface information extraction. In the output, the weighted result of the convolution part is added to the input part to enhance the expression of shallow information. In the fine-gained information branch, we introduce the visual Transformer to extract the fine-grained information. Many methods have proved that Transformer can better model the pixel-level features of the image. However, since the image belongs to two-dimensional data, processing it in a serialized manner will destroy the location information of the image. Meanwhile, due to the huge overhead of the Transformer, it is unsuitable to directly model an entire feature map. Therefore, we borrowed the idea of Swin Transformer [25] to decompose the feature map into smaller windows. Meanwhile, the window displacement mechanism is also be applied to enhance the information flow and interaction between windows. As shown in Fig. 5, (SW) MAS denotes the (Shift Window) Multi-Head Self-Attention mechanism proposed by Swin Transformer. Considering that the working mechanism of CNN and Transformer is different, adding the output features of these two branches directly will lead to information confusion. Therefore, we concatenate the output of CNN and Transformer to get rich features, and then a convolutional layer is used to weight and fuse different features to guide the module to learn useful features adaptively.

IV. Experiments

A. Datasets

In this paper, we use 800 training images in DIV2K [32] as the training set. For evaluation, we choose six benchmark test sets, including Set12 [33], BSD68 [34], Kodak24 [35], CBSD68 [36], and Urban100 [37]. In addition, we choose additive white Gaussian noise (AWGN) as our research object since AWGN is the best approximation of the real mixture noise, which can simulate the disturbance of real noise to the image. Following previous works, we use Set12, BSD68, and Urban100 to evaluate the performance of EWT in grayscale images, and use Kodak24, CBSD68, and Urban100 to evaluate the denoising effect of model on color images. Meanwhile, to further verify the effectiveness and robustness of EWT, we utilize SIDD [38] and RNI15 [39] to evaluate the denoising performance of the model in the real image denoising task.

B. Implementation Details

Before training, we generate noisy images by adding AWGN with different noise levels. To verify the effectiveness of the model, we set the noise level σ = 15, 25, and 50 for grayscale images and set σ = 10, 30, and 50 for color images. During training, we randomly choose 16 noisy patches as inputs and these patches are randomly rotated and flipped to enhance the data. In addition, EWT is implemented with PyTorch framework and updated with the Adam optimizer.

In the final model, we use a single-scale wavelet to sample the image. The size of all convolution kernels in the model is 3×3, the λ in the ConvBlock is set to 0.1, and the embedding dimension of MFAM is set to 180. In addition, we use 4 DFEB in MFAM, and each DFEB contains 1 ConvBlock and 6 Transformer blocks. In the Transformer, the window size is 8, the number of attention heads is 6, and the MLP dimension is as twice as the embedding dimension.

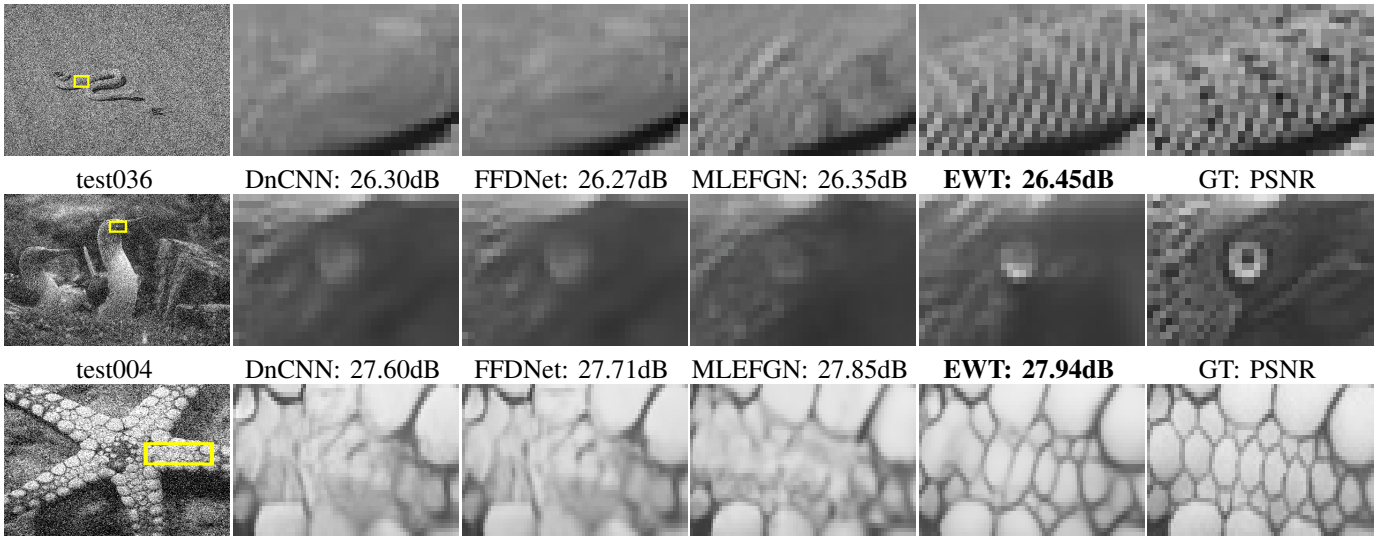

Fig. 6. Visual comparison on grayscale images with $\sigma = 50$. Obviously, our EWT can reconstruct high-quality noise-free images with clear edges.

TABLE I
PSNR (DB) COMPARISON WITH OTHER CLASSIC SID METHODS ON GRAYSCALE IMAGE TEST DATASETS. THE BEST RESULTS ARE **HIGHLIGHTED**.

Method	Set12			BSD68			Urban100			Average
Noise Level	σ = 15	σ = 25	σ = 50	σ = 15	σ = 25	σ = 50	σ = 15	σ = 25	σ = 50	
BM3D [40]	32.37dB	29.97dB	26.72dB	31.08dB	28.57dB	25.60dB	32.35dB	29.70dB	25.95dB	29.15dB
WNNM [41]	32.70dB	30.28dB	27.05dB	31.37dB	28.83dB	25.87dB	32.97dB	30.39dB	26.83dB	29.59dB
IRCNN [42]	32.76dB	30.37dB	27.12dB	31.63dB	29.15dB	26.19dB	32.46dB	29.80dB	26.22dB	29.52dB
DnCNN [12]	32.86dB	30.44dB	27.18dB	31.73dB	29.23dB	26.23dB	32.64dB	29.95dB	26.26dB	29.61dB
FFDNet [14]	32.75dB	30.43dB	27.32dB	31.63dB	29.19dB	26.29dB	32.40dB	29.90dB	26.50dB	29.60dB
RED30 [43]	32.83dB	30.48dB	27.34dB	31.72dB	29.26dB	26.35dB	32.75dB	30.21dB	26.48dB	29.71dB
MLEFGN [22]	33.04dB	30.66dB	27.54dB	31.81dB	29.34dB	26.39dB	33.21dB	30.64dB	27.22dB	29.98dB
MWCNN [31]	33.15dB	30.79dB	27.74dB	31.86dB	**29.41dB**	**26.53dB**	33.17dB	30.66dB	27.42dB	30.08dB
EWT (Ours)	**33.23dB**	**30.86dB**	**27.80dB**	**31.87dB**	29.40dB	26.47dB	**33.54dB**	**31.08dB**	**27.70dB**	**30.22dB**

TABLE II
PSNR (DB) COMPARISON WITH OTHER CLASSIC SID METHODS ON COLOR IMAGE TEST DATASETS. THE BEST RESULTS ARE **HIGHLIGHTED**.

Method	Kodak24			CBSD68			CUrban100			Average
Noise Level	σ = 10	σ = 30	σ = 50	σ = 10	σ = 30	σ = 50	σ = 10	σ = 30	σ = 50	
CBM3D [40]	36.57dB	30.89dB	28.63dB	35.91dB	29.73dB	27.38dB	36.00dB	30.36dB	27.94dB	31.49dB
IRCNN [42]	36.70dB	31.24dB	28.93dB	36.06dB	30.22dB	27.86dB	35.81dB	30.28dB	27.69dB	31.64dB
DnCNN [12]	36.98dB	31.39dB	29.16dB	36.31dB	30.40dB	28.01dB	36.21dB	30.28dB	28.16dB	31.87dB
FFDNet [14]	36.81dB	31.39dB	29.10dB	36.14dB	30.31dB	27.96dB	35.77dB	30.53dB	28.05dB	31.78dB
MLEFGN [22]	37.04dB	31.67dB	29.38dB	36.37dB	30.56dB	28.21dB	36.42dB	31.32dB	28.92dB	32.21dB
RNAN [44]	37.24dB	31.86dB	29.58dB	36.43dB	30.63dB	28.27dB	36.59dB	31.50dB	29.08dB	32.35dB
RDN [15]	**37.31dB**	**31.94dB**	**29.66dB**	36.47dB	30.67dB	28.31dB	**36.69dB**	31.69dB	29.29dB	32.45dB
EWT (Ours)	37.24dB	31.90dB	29.63dB	**36.49dB**	**30.71dB**	**28.36dB**	**36.69dB**	**31.81dB**	**29.52dB**	**32.48dB**

C. Comparisons with State-of-the-art Methods

Gray-scale Image Denoising: In Tabel I, we report the PSNR results of different SID methods on three benchmark test sets. Obviously, EWT achieves competitive results and the best average results on these test sets with different noise levels. It is worth noting that MWCNN is also a wavelet-based SID model, which achieved slightly better results than EWT on BSD68 (σ = 25 and 50). However, it cannot be ignored that the results of MWCNN under other test sets are all worse than our EWT, and the average result is 0.14dB worse than EWT. Meanwhile, MWCNN uses multiple training sets to train the model, which contains 5744 images (7 times of our training images). Under this disparity, EWT still achieves close or better results, which fully demonstrates its effectiveness.

In Fig. 6, we provide the visual comparison of the denoised images with noise levels σ = 50. In this part, we choose three most representative CNN-based image denoising methods for comparison, including DnCNN [12], FFDNet [14], and MLEFGN [22]. Among them, DnCNN and FFDNet are the two most classic CNN-based image denoising models. According to the figure, we can clearly observe that the images reconstructed by DnCNN and FDDNet are too smooth, and they have lost texture details and edge information. As for

Fig. 7. Visual comparison on color images with $\sigma = 50$. Obviously, our EWT can reconstruct high-quality noise-free images with clear edges.

MLEFGN, it can reconstruct more clear noise-free images, but the edges of the image are not accurate and complete enough. In contrast, our EWT can reconstruct high-quality images with clear and accurate texture details and edges. This further illustrates the effectiveness and excellence of EWT. **Color Image Denoising:** As for color image denoising, we use Kodak24, CBSD68, and CUrban100 to verify its performance. In this part, we choose three most representative CNN-based image denoising methods for comparison, including DnCNN [12], ADNet [14], and MLEFGN [22]. According to TABLE II, we can clearly observe that our EWT still achieves excellent results on color images, especially on Urban100. Among them, RDN is recognized as one of the most advanced SID models, which is specially designed for color image denoising. Compared with it, our EWT achieved close results on Kodak24 and better results on CBSD68 and CUrban100. It is worth noting that our EWT achieves better average result than RDN with only half of the parameters (EWT: 11M vs RDN: 22M). These results fully demonstrate the denoising ability of EWT on color images, further validating the effectiveness of EWT.

In Fig. 7, we provide the visual comparisons of the denoised images with σ = 50 on CBSD68. In this part, we also choose three most representative CNN-based image denoising methods for comparison, including DnCNN [12], ADNet [45], and MLEFGN [22]. Obviously, our EWT can reconstruct high-quality noise-free images with sharper and more accurate edges. Taking the human face as an example, our EWT can reconstruct clearer and more accurate contours. This is due to the fact that the Transformer introduced in EWT can capture the global information of the face, thereby reconstructing high-quality face. All these results further illustrate the effectiveness of the proposed EWT.

Restoration of Other Synthetic Noise: The noise used in practical applications is usually more than Gaussian noise, and other noises are also very common, such as Poisson noise and Speckle noise. Since it has a more complex distribution, it also needs to be considered emphatically. In order to verify the general applicability of the method in this paper, TABLE III compares EWT with three classic image restoration Transformer methods. The results show that EWT also performs well in other noisy images. This is due to the idea of combining wavelet transform in this paper, which ensures that the Transformer always maintains attention to the image texture details during the feature extraction process. This further validates the effectiveness of our proposed EWT, and also reflects the generality of EWT on different noisy images.

Real Image Denoising: Real image denoising is a more difficult task since real image noise comes from multiple sources. In this part, real noisy images are used to further assess the practicability of the proposed EWT. In TABLE IV, we provide PSNR comparisons of EWT with other models specially designed for real image denoising. Among them, * denote the model using additional training sets to train the model. Obviously, our model still achieves the best results even without using additional training sets. This further validates the effectiveness and versatility of our EWT. In addition, we also provide the visual comparison on SIDD [38] and RNI15 [39] sets in Figs. 8 and 9, respectively. Obviously, our EWT still can reconstruct high-quality noise-free images. This shows that EWT also performs well on the real image denoising task.

V. Ablation Studies

A. Wavelet Investigations

In our method, the wavelet plays a vital role in shortening the execution time and GPU memory consumption. To verify

TABLE III
QUANTITATIVE COMPARISON WITH OTHER TRANSFORMER METHODS ON POISSON NOISE AND SPECKLE NOISE.

Noise Level	Poisson			Speckle		
Method	Kodak24	CBSD68	Urban100	Kodak24	CBSD68	Urban100
SwinIR [19]	37.09dB	36.44dB	36.58dB	31.07dB	29.87dB	29.94dB
Uformer [20]	36.80dB	36.08dB	36.20dB	30.71dB	29.42dB	29.72dB
Restormer [21]	37.14dB	36.52dB	36.66dB	31.01dB	29.85dB	29.90dB
EWT	**37.20dB**	**36.52dB**	**36.61dB**	**31.24dB**	**29.98dB**	**29.90dB**

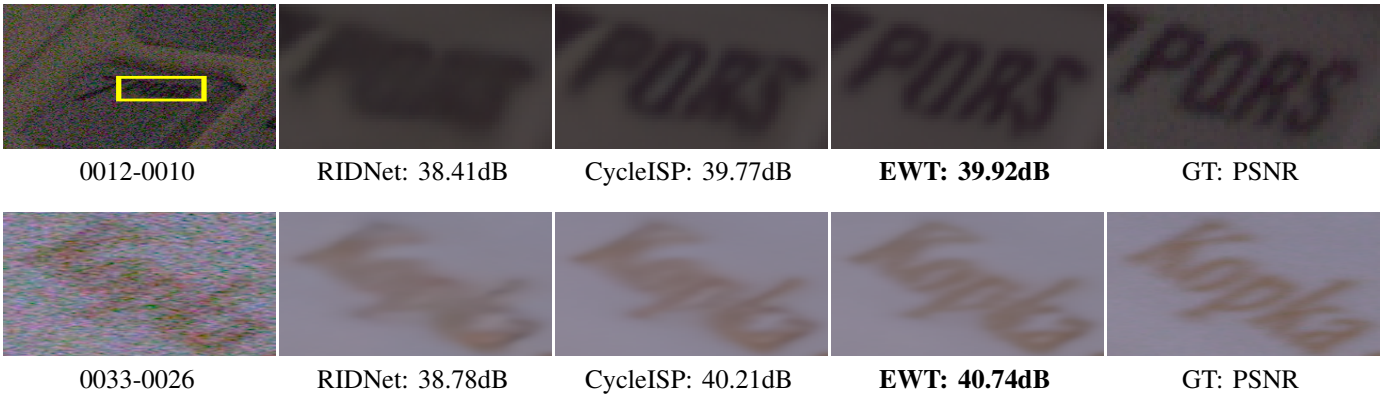

Fig. 8. Visual comparison on real-noise images (SIDD [38]). Obviously, EWT can reconstruct high-quality noise-free images.

Fig. 9. Visual comparison on real-noise images (RNI15 [39]). Obviously, EWT can reconstruct high-quality noise-free images.

TABLE IV
REAL IMAGE DENOISING COMPARISON WITH DNCNN [12], BM3D [40], CBDNET [46], RIDNET [47], AINDNET [48], VDN [49], SADNET [50], DANET+ [51], CYCLEISR [52], DEAMNET [53] ON SIDD [38] ($\star$ DENOTE THE MODEL USING ADDITIONAL TRAINING SETS TO TRAIN THE MODEL).

Method	DnCNN	BM3D	CBDNet$\star$	RIDNet$\star$	AINDNet$\star$	VDN	SADNet$\star$	DANet+$\star$	CycleISR$\star$	DeamNet$\star$	EWT (Ours)
PSNR	23.66dB	25.65dB	30.78dB	38.71dB	38.95dB	39.28dB	39.46dB	39.47dB	39.52dB	39.35dB	**39.52dB**

TABLE V
COMPARISON WITH SWINIR* ON KODAK24 (σ = 30, COLOR).

Method	Patchsize	GPU	Time	Params	PSNR
SwinIR*	56	18432MiB	53.29s	5.17M	**31.79dB**
DWT+SwinIR*	56	8276MiB	12.52s	5.20M	**31.57dB**
EWT*	56	**6347MiB**	**9.14s**	5.18M	31.73dB

this statement, we compare it with SwinIR [19]. SwinIR is a famous Transformer-based image restoration model, which does not use wavelet or other operations to change the image resolution. It is worth noting that SwinIR uses additional training sets and the GPU memory required for it exceeds the maximum limit of our device. For a fair comparison, the embedding dimension of MFAM in SinwIR and EWT are both reduced from 180 to 120, and these two models are retrained under the same data set and settings. In addition, we also consider the combination of DWT and SwinIR to further illustrate the effectiveness of EWT and the rationality of its structure. Meanwhile, we label these two modified models as SwinIR* and EWT*, respectively. According to TABLE V, we can clearly observe that EWT* and SwinIR* have a similar number of parameters, and EWT achieves close PSNR results to SwinIR with only 1/6 running time and 1/3 GPU memory. In addition, we also noticed that directly combine DWT with SwinIR will degrade the performance of the model since it does not optimize the structural design of the network. Contrastly, our EWT achieves better results due to its well-designed network structure and effective DFEB. This huge breakthrough fully demonstrated the advantages of wavelet and further verified the advancement and effectiveness of EWT.

In order to further verify the influence of multi-level wavelet on the model performance, we designed a series of studies in TABLE VI. Among them, cases 1, 2, and 3 denote the different

TABLE VI
STUDY OF MULTI-LEVEL WAVELET (KODAK24, σ = 30, COLOR).

Case	Multi-Level	Time	Patchsize	GPU	FLOPs	PSNR
1	$\times$1	11.96s	64	7636MiB	17.82G	**31.78dB**
2	$\times$2	3.09s	64	3658MiB	4.50G	31.62dB
3	$\times$3	1.91s	64	2758MiB	1.18G	27.94dB

TABLE VII
STUDY OF DFEB'S BRANCH STRATEGY (KODAK24, σ = 30, COLOR).

Case	Branch1	Branch2	Params	Time	GPU	FLOPs	PSNR
1	Conv	Conv	6.45M	2.44s	1459MiB	13.32G	31.12dB
2	Trans	Trans	6.08M	7.37s	6050MiB	8.29G	31.66dB
3	Conv	Trans	6.12M	6.21s	4934MiB	9.52G	**31.72dB**

levels of wavelet with fixed patch size. According to these results, we can find that when the level of wavelet increases, the required execution time and GPU memory consumption will be greatly reduced, but it cannot be ignored that the performance of the model will also decrease. This is because multiple downsampling operation makes the resolution of the image gradually decrease, so the GPU memory consumption is also greatly reduced. However, low-resolution will also cause the loss of local information of the image, making it difficult to reconstruct high-quality images. Therefore, multi-level wavelet-based models can be applied to mobile devices, which have strict restrictions on memory and execution time. In summary, the wavelet is effective to balance model performance and resource consumption. At the same time, multi-level wavelet can be considered according to actual needs.

B. DFEB Investigations

As the most important part of EWT, Dual-stream Feature Extraction Block (DFEB) is designed for feature extraction while reducing the model size and shortening the running time. This is benefits from the double-branch structure in DFEB, which can elegantly combine CNN and Transformer. In order to verify the effectiveness of this strategy, we designed a series of experiments in TABLE VII. Among them, all models only use two DFEBs and are trained with patchsize=64 for quick verification. According to the table, we can observe that the use of convolutional layers will lead to an increase in the number of parameters and FLOPs, and the use of Transformer will lead to more GPU memory consumption and longer execution time. Therefore, the model using our proposed strategy achieves intermediate results across multiple metrics. However, it is worth mentioning that our method achieves the best PSNR result and has a good balance between the performance, execution time, GPU memory consumption, FLOPs, and size of the model. All these results fully validate the necessity and effectiveness of the combination of CNN and Transformer.

In addition to this, we also study the impact of the number of DFEBs on model performance, execution time, and GPU usage in TABLE VIII. In this part, we set the patchsize to 64 to speed up training. Obviously, when the number of DFEBs is increased from 1 to 2, the model performance improves by 0.17dB. Continuing to increase the number of DFEBs can further improve the performance of the model, but the growth rate will gradually decrease. At the same time, it cannot be ignored that as the number of DFEBs increases, the GPU memory consumption and execution time of the model will greatly increase. Therefore, to ensure the efficiency of the model, we use 4 DFEBs in the final version of EWT.

TABLE VIII
STUDY OF THE DFEB NUMBER TO MODEL PERFORMANCE (KODAK24, σ = 30, COLOR). XN STANDS FOR N DFEBS.

Case	DFEBs	Params	Time	GPU	Flops	PSNR
1	$\times$1	3.34M	3.01s	2656MiB	5.44G	31.55
2	$\times$2	6.12M	6.21s	4934MiB	9.52G	31.72
3	$\times$3	8.84M	9.84s	6324MiB	13.69G	31.75
4	$\times$4	11.8M	11.96s	7636MiB	17.82G	**31.78**

TABLE IX
DETAILED COMPARISON STUDY WITH TRANSFORMER METHOD UNDER GAUSSIAN NOISE CONDITION (NOISE LEVEL σ = 30).

Method	GPU	Params	Dataset	PSNR	Time
SwinIR [19]	18432MiB	5.17M	Kodak24	**31.79dB**	53.29s
			CBSD68	**30.64dB**	85.91s
			CUrban100	**31.36dB**	232.46s
Uformer [20]	6875MiB	5.28M	Kodak24	31.57dB	9.46s
			CBSD68	30.07dB	16.21s
			CUrban100	30.82dB	44.50s
Restormer [21]	21894MiB	12.47M	Kodak24	31.62dB	42.86s
			CBSD68	30.51dB	82.53s
			CUrban100	31.16dB	215.02s
EWT	**6347MiB**	5.18M	Kodak24	31.73dB	**9.14s**
			CBSD68	30.60dB	**14.34s**
			CUrban100	31.35dB	**43.77s**

C. Comparision with SwinIR

In the previous subsection, we compared EWT with SwinIR [19] to verify the positive effect of wavelet on the model. Here we provide more datasets and methods (Uformer [20] and Restormer [21]) to further verify the effectiveness of EWT. All models are retrained under the same dataset and training settings. In TABLE IX we provide the number of parameters of the model, GPU memory used for training, PSNR results and average execution time on different test sets. As can be seen from the results in the table, EWT achieved better results than Uformer and Restorer with less GPU memory and execution time, maintaining a good balance between performance and operating efficiency. It is worth noting that Uformer does improve efficiency through multi-level downsampling but seriously affects the performance of the model. This is why we introduced the wavelet transform to replace the downsampling operation since the downsampling operation will cause a large number of features to be lost. On the whole, our EWT is a very potential method for image denoising and provide a new solution for image restoration.

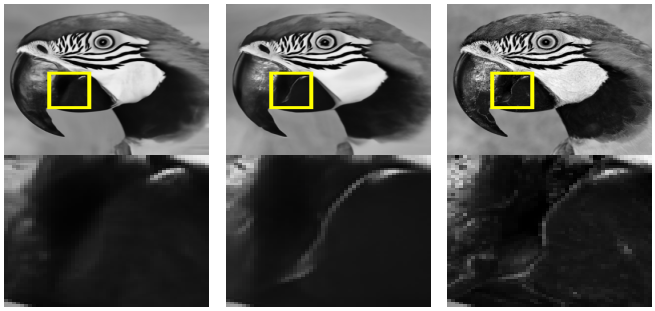

Fig. 10. Visual comparison with SwinIR [19] on grayscale image. Obviously, EWT can reconstruct more accurate and clear edges. (Set12 [33], σ = 50)

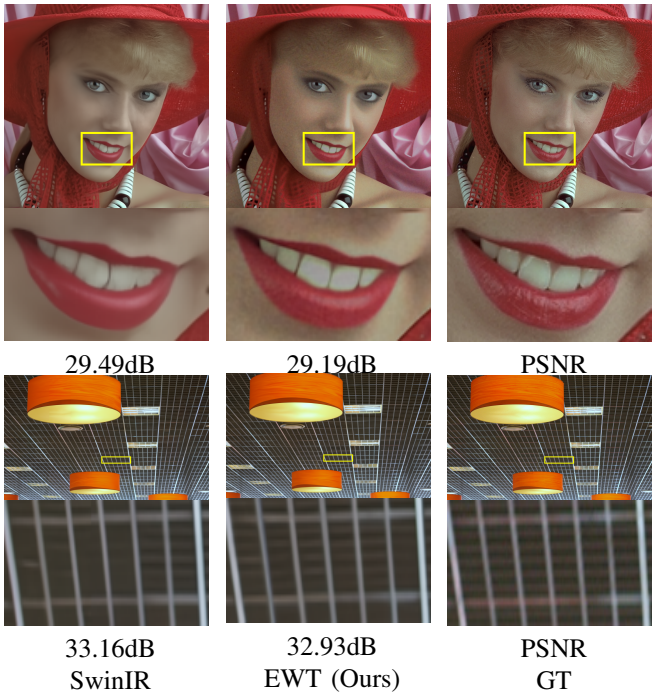

Fig. 11. Visual comparison with SwinIR [19] on color image. Obviously, EWT can reconstruct more accurate and clear lines. (σ = 50)

In Figs. 10 and 11, we provide the visual comparisons with SwinIR [19] on grayscale and color images, respectively. It is worth noting that the SwinIR results used here are the denoised image reconstructed by the original paper provided pre-trained model, which uses DIV2K [32] (800 training images), Flickr2K [54] (2650 images), BSD500 [36] (400 training and testing images) and Waterloo Exploration Database [55] (4744 images) for training. However, our EWT only use 800 training images from DIV2K, which is 1/10 of the SwinIR training set. According to the results, we can clearly observe that although SwinIR achieved slightly better PSNR results than our EWT, the reconstructed denoised images are also smoother and lack texture details. In contrast, our EWT can reconstruct sharper and more accurate image edges. This is because the introduced wavelet can capture the frequency and position information of the image, which is beneficial to restore the detailed features of the image. Therefore, we can draw the following conclusions: (1). Compared with SwinIR, our EWT can achieve close results with less GPU memory consumption and faster inference time; (2). Compared with SwinIR, our reconstructed denoised images have richer texture details and more accurate edges. All these results further validate the effectiveness of EWT. To sum up, our method has more advantages than previous Transformer-based models, which achieve a good balance between the performance and efficiency of the model.

D. Comparision with MWCNN

In this paper, we proposed a novel Efficient Wavelet-Transformer (EWT) for single image denoising. This is the first attempt of Transformer in wavelet domain. As we mentioned in the previous section, EWT was proposed inspired by MWCNN [31]. Therefore, we give a detailed comparison with MWCNN in TABLE XI. According to the table, we can clearly observe that our EWT achieves better results on the vast majority of datasets and noise levels with fewer parameters. This fully demonstrates the effectiveness of the proposed EWT. Meanwhile, it also means that it is meaningful and feasible to combine wavelet and Transformer, which further promoted the development of the wavelet in SID.

TABLE X
PSNR (DB) AND PARAMETER QUANTITY COMPARISON WITH DHDN [23] AND DIDN [16] ON COLOR IMAGE TEST DATASETS.

Method	Noise Level	DHDN	DIDN	EWT (Ours)
Kodak24	σ=10	**37.33dB**	37.32dB	37.24dB
	σ=30	31.95dB	**31.97dB**	31.90dB
	σ=50	29.67dB	**29.72dB**	29.63dB
CBSD68	σ=10	36.45dB	36.48dB	**36.49dB**
	σ=30	30.41dB	30.70dB	**30.71dB**
	σ=50	28.02dB	28.35dB	**28.36dB**
Parameters		168M	165M	**11.8M**

E. Model Size Investigations

Increasing the depth of the model is the easiest way to improve the model performance. However, it cannot be ignored that these models [15], [16], [23] also accompanied by a large number of parameters. In Fig. 1, we provide the performance and parameter comparisons of EWT with other SID models, including IRCNN [42], DnCNN [12], FFDNet [14], ADNet [45], BRDNet [56], MLEFGN [22], RNAN [44], RDN [15], DIDN [16], and IPT [18]. Among them, the red star represents EWT. Obviously, EWT achieves competitive results with few parameters, which strike a good balance between the performance and size of the model. Moreover, we provide a detailed comparison with DHDN [23] and DIDN [16] in TABLE X. **Obviously, EWT achieves best results on CBSD68 and close results on Kodak24 with only 1/14 parameters of DHDN and DIDN.** All these results validate that EWT is an efficient and accurate SID model.

VI. DISCUSSION

In this paper, we proposed an Efficient Wavelet-Transformer (EWT) and demonstrate its effectiveness on the SID task.

TABLE XI
COMPARISON WITH MWCNN ON GRAYSCALE IMAGES. THE BEST RESULTS ARE **HIGHLIGHTED**.

Method	Parameters	Set12			BSD68			Urban100		
Noise Level		σ =15	σ =25	σ =50	σ =15	σ =25	σ =50	σ =15	σ =25	σ =50
MWCNN [31]	19.2M	33.15dB	30.79dB	27.74dB	31.86dB	**29.41dB**	**26.53dB**	33.17dB	30.66dB	27.42dB
EWT (Ours)	11.8M	**33.23dB**	**30.86dB**	**27.80dB**	**31.87dB**	29.40dB	26.47dB	**33.54dB**	**31.08dB**	**27.70dB**

However, this does not mean that it is only suitable for SID. EWT is a general model that can be applied to other image restoration tasks, such as image super-resolution, image dehazing, and image deraining. In future works, we will further explore its effectiveness on other image restoration tasks, and optimize the model according to different tasks.

VII. CONCLUSION

In this paper, a novel Efficient Wavelet-Transformer (EWT) is proposed for single image denoising. Specifically, we introduced Discrete Wavelet Transform (DWT) and Inverse Wavelet Transform (IWT) for downsampling and upsampling operations, respectively. This method can greatly reduce the resolution of the image, thereby reducing GPU memory consumption, and will not cause any loss of information. Meanwhile, an efficient Multi-level Feature Aggregation Module (MFAM) is proposed to make full use of hierarchical features by using local and global residual learning. In addition, a novel Dual-stream Feature Extraction Block (DFEB) is specially designed for local and global features extraction, which combines the advantages of CNN and Transformer that can take into account the information of different levels. Extensive experiments show that our EWT achieves the best balance between the performance, size, execution time, and GPU memory consumption of the model.

REFERENCES

[1] Fang Liu, Licheng Jiao, and Xu Tang. Task-oriented gan for polsar image classification and clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 30(9):2707–2719, 2019.

[2] Wei Luo, Jun Li, Jian Yang, Wei Xu, and Jian Zhang. Convolutional sparse autoencoders for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 29(7):3289–3294, 2017.

[3] Jing Liu, Yuhang Wang, Yong Li, Jun Fu, Jiangyun Li, and Hanqing Lu. Collaborative deconvolutional neural networks for joint depth estimation and semantic segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 29(11):5655–5666, 2018.

[4] Xiaofeng Ding, Tieyong Zeng, Jian Tang, Zhengping Che, and Yaxin Peng. Srrnet: A semantic representation refinement network for image segmentation. *IEEE Transactions on Multimedia*, 2022.

[5] Feng Zhang, Xueying Wang, Shilin Zhou, and Yingqian Wang. Dardet: A dense anchor-free rotated object detector in aerial images. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2021.

[6] Tianhao Wu, Boyang Li, Yihang Luo, Yingqian Wang, Chao Xiao, Ting Liu, Jungang Yang, Wei An, and Yulan Guo. Mtu-net: Multi-level transunet for space-based infrared tiny ship detection. *IEEE Transactions on Geoscience and Remote Sensing*, 2023.

[7] Kasper Winther Jorgensen and Lars Kai Hansen. Model selection for gaussian kernel pca denoising. *IEEE Transactions on Neural Networks and Learning Systems*, 23(1):163–168, 2011.

[8] Joon-Ku Im, Daniel W Apley, and George C Runger. Tangent hyperplane kernel principal component analysis for denoising. *IEEE Transactions on Neural Networks and Learning Systems*, 23(4):644–656, 2012.

[9] Yu Sun, Jiaming Liu, and Ulugbek Kamilov. Block coordinate regularization by denoising. *Advances in Neural Information Processing Systems*, 32, 2019.

[10] Qianting Ma, Yang Wang, and Tieyong Zeng. Retinex-based variational framework for low-light image enhancement and denoising. *IEEE Transactions on Multimedia*, 2022.

[11] Minjie Chen, Mantao Xu, and Pasi Franti. Adaptive context-tree-based statistical filtering for raster map image denoising. *IEEE transactions on multimedia*, 13(6):1195–1207, 2011.

[12] 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.

[13] Dong Yang and Jian Sun. Bm3d-net: A convolutional neural network for transform-domain collaborative filtering. *IEEE SPL*, 25(1):55–59, 2017.

[14] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 2018.

[15] 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, 2021.

[16] Bumjun Park, Songhyun Yu, and Jechang Jeong. Densely connected hierarchical network for image denoising. In *CVPR Workshops*, 2019.

[17] Zehua Sheng, Xiongwei Liu, Si-Yuan Cao, Hui-Liang Shen, and Huaqi Zhang. Frequency-domain deep guided image denoising. *IEEE Transactions on Multimedia*, 2022.

[18] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *CVPR*, 2021.

[19] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *ICCV*, 2021.

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

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

[22] Faming Fang, Juncheng Li, Yiting Yuan, Tieyong Zeng, and Guixu Zhang. Multilevel edge features guided network for image denoising. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[23] Songhyun Yu, Bumjun Park, and Jechang Jeong. Deep iterative down-up cnn for image denoising. In *CVPR Workshops*, 2019.

[24] 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.

[25] 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.

[26] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020.

[27] Woong Bae, Jaejun Yoo, and Jong Chul Ye. Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification. In *CVPR Workshops*, 2017.

[28] Tiantong Guo, Hojjat Seyed Mousavi, Tiep Huu Vu, and Vishal Monga. Deep wavelet prediction for image super-resolution. In *CVPR Workshops*, 2017.

[29] Zhisheng Zhong, Tiancheng Shen, Yibo Yang, Zhouchen Lin, and Chao Zhang. Joint sub-bands learning with clique structures for wavelet domain super-resolution. *NeurIPS*, 2018.

[30] Yibo Yang, Zhisheng Zhong, Tiancheng Shen, and Zhouchen Lin. Convolutional neural networks with alternately updated clique. In *CVPR*, 2018.

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

[32] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPRW*, 2017.

[33] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *ICCS*, 2010.

[34] Stefan Roth and Michael J Black. Fields of experts: A framework for learning image priors. In *CVPR*, 2005.

[35] Rich Franzen. Kodak lossless true color image suite., source, 1999.

[36] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001.

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

[38] Abdelrahman Abdelhamed and Stephen Lin. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018.

[39] Marc Lebrun, Miguel Colom, and Jean-Michel Morel. The noise clinic: a blind image denoising algorithm. *Image Processing On Line*, 5:1–54, 2015.

[40] 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.

[41] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *CVPR*, 2014.

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

[43] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. *NeurIPS*, 2016.

[44] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. 2019.

[45] Chunwei Tian, Yong Xu, Zuoyong Li, Wangmeng Zuo, Lunke Fei, and Hong Liu. Attention-guided cnn for image denoising. *Neural Networks*, 124:117–129, 2020.

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

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

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

[49] Zongsheng Yue, Hongwei Yong, Qian Zhao, Lei Zhang, and Deyu Meng. Variational denoising network: Toward blind noise modeling and removal. *arXiv preprint arXiv:1908.11314*, 2019.

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

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

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

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

[54] Andreas Lugmayr, Martin Danelljan, Radu Timofte, Manuel Fritsche, Shuhang Gu, Kuldeep Purohit, Praveen Kandula, Maitreya Suin, AN Rajagoapalan, Nam Hyung Joon, et al. Aim 2019 challenge on real-world image super-resolution: Methods and results. In *ICCV Workshop*, 2019.

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

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