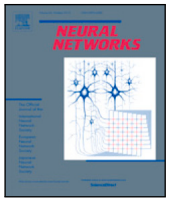
Neural Networks 177 (2024) 106378

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



Full Length Article

EWT: Efficient Wavelet-Transformer for single image denoising

Juncheng Li [a,b], Bodong Cheng [c,*], Ying Chen [d], Guangwei Gao [e,g], Jun Shi [a], Tieyong Zeng [f,*]

[a] School of Communication and Information Engineering, Shanghai University, Shanghai, 200444, China
[b] Shanghai Key Laboratory of Multidimensional Information Processing, East China Normal University, Shanghai, 200444, China
[c] School of Computer Science and Technology, East China Normal University, Shanghai, 200444, China
[d] Department of Cyberspace Security, Beijing Electronic Science and Technology Institute, Beijing, 100070, China
[e] Institute of Advanced Technology, Nanjing University of Posts and Telecommunications, Nanjing, 210049, China
[f] Department of Mathematics, The Chinese University of Hong Kong, New Territories, 999077, Hong Kong, China
[g] Key Laboratory of Artificial Intelligence, Ministry of Education, 200240, Shanghai, China

ARTICLE INFO

Keywords:
Image denoising
Wavelet transform
Dual-stream network

ABSTRACT

Transformer-based image denoising methods have shown remarkable potential but suffer from high computational cost and large memory footprint due to their linear operations for capturing long-range dependencies. In this work, we aim to develop a more resource-efficient Transformer-based image denoising method that maintains high performance. To this end, we propose an Efficient Wavelet Transformer (EWT), which incorporates a Frequency-domain Conversion Pipeline (FCP) to reduce image resolution without losing critical features, and a Multi-level Feature Aggregation Module (MFAM) with a Dual-stream Feature Extraction Block (DFEB) to harness hierarchical features effectively. EWT achieves a faster processing speed by over **80%** and reduces GPU memory usage by more than **60%** compared to the original Transformer, while still delivering denoising performance on par with state-of-the-art methods. Extensive experiments show that EWT significantly improves the efficiency of Transformer-based image denoising, providing a more balanced approach between performance and resource consumption.

1. Introduction

Image denoising stands as a prominent subject within the domain of image restoration, seeking to reconstruct a pristine image from one afflicted by noise. As a pivotal step in numerous practical applications, the quality of denoised images markedly influences the efficacy of subsequent tasks, including but not limited to image classification (Liu, Jiao, & Tang, 2019; Wu et al., 2023), image segmentation (Shim, Yu, Kong, & Kang, 2023; Wei & Ye, 2022), and target detection (Shih, Chiu, Lin, & Bu, 2019; Zhang, Liu, & Lu, 2022). Nevertheless, it remains a formidable challenge owing to its nature as an ill-posed inverse problem.

Over the past few decades, researchers have undertaken extensive explorations and endeavors in the realm of single image denoising (SID). SID methodologies can be broadly categorized into traditional denoising methods (Im, Apley, & Runger, 2012; Jorgensen & Hansen, 2011) and learning-based methods. Traditional methods typically employ iterative processes, rendering them inefficient and beset with poor generalization performance. Conversely, learning-based approaches aim to directly acquire the mapping between noisy and clean images, endowing the model with inherent denoising capabilities. Notably, the advent of deep learning across diverse domains, coupled with the remarkable performance of convolutional neural networks (CNN) in computer vision, has led to the emergence of several CNN-based SID techniques (Ma, Li, Zhang, & Li, 2022; Park, Yu, & Jeong, 2019; Shen, Zhao, & Zhang, 2023; Xu et al., 2021; Zhang, Tian, Kong, Zhong, & Fu, 2021). These methods harness the potent feature extraction capabilities of CNNs, employing specially crafted strategies for learning and training, ultimately yielding promising results.

The performance of CNN-based methods is indeed better than traditional methods. However, the working mechanism of CNN still has some limitations. Specifically, CNN uses shared convolutional layers to extract image features, which can only extract some local features, so the relationship between pixel-level features cannot be extracted. For SID, fine-grained features are important for high-quality image reconstruction. To solve these problems, the most widely used method is to increase the depth of the network and adjust the connection method. However, this will lead to an explosion in the number of parameters, affect the inference speed, increase GPU memory consumption, and is not conducive to model deployment. Therefore, it is necessary to build a model that can effectively capture global information.

* Corresponding authors.
E-mail addresses: bdcheng@stu.xidian.edu.cn (B. Cheng), zeng@math.cuhk.edu.hk (T. Zeng).

https://doi.org/10.1016/j.neunet.2024.106378
Received 18 December 2023; Received in revised form 20 April 2024; Accepted 6 May 2024
Available online 8 May 2024

Table 1
Investigation on the relationship between various indicators of traditional Transformer and input image resolution.

Case	Patchsize	Time	GPU	FLOPs
1	256	13.77 s	8429 MiB	18.24 G
2	512	29.60 s	12301 MiB	30.06 G
3	1024	45.93 s	23814 MiB	58.47 G

Recently, with the outstanding performance of Transformer in Natural Language Processing (NLP), some works began to introduce Transformer to computer vision and achieved good performance. However, these models are mostly designed for high-level vision tasks, such as image recognition (Dosovitskiy et al., 2020), target detection (Liu et al., 2021), and image segmentation (Liang et al., 2020). These tasks only require the model to infer a certain probability distribution or generate some degraded images. Therefore, the downsampling operation can be performed on image features before it is sent to the Transformer to compress feature information and reduce computational and storage overhead. However, this is not suitable for image restoration tasks since downsampling will lose a lot of information and affect the model performance. Although some studies (Chen et al., 2021; Liang et al., 2021) have incorporated Transformers into their models with promising results, the use of matrix operations on each pixel in an image leads to significant time and space overhead. Meanwhile, this phenomenon will worsen rapidly as the input image resolution increases (Table 1). To mitigate this, many Transformer-based image restoration approaches use the patch processing method, dividing the image into smaller patches for operation. Although this method can reduce the computational load, it still requires substantial GPU memory, prolonging inference times. Consequently, achieving a balance between performance and resource efficiency remains a challenge for these methods.

To address the bottleneck of Transformer in image restoration, especially in SID, we propose an Efficient Wavelet-Transformer (EWT). Specifically, we introduce an image compression strategy with a Frequency-domain Conversion Pipeline (FCP). FCP is designed based on Discrete Wavelet Transform (DWT) and Inverse Wavelet Transform (IWT), which take advantage of the reversible characteristics of wavelet as the sampling unit for model input and output. This pipeline can preserve image features while reducing the image resolution, thereby effectively increase the inference speed of the model and reduce a large number of GPU memory occupations. In the network backbone, we propose an efficient multi-level feature aggregation module (MFAM). MFAM consists of a series of Dual-stream Feature Extraction Block (DFEB), which combine Transformer and CNN to realize the extraction and fusion of local and global features. The contributions of this work are as follows:

- We 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), which consist of specially designed Dual-stream Feature Extraction Blocks (DFEB) that combines the advantages of CNN and Transformer to help the model extract different levels of image features.
- We 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.

The rest of this paper is organized as follows. Related works are reviewed in Section 2. A detailed explanation of the proposed EWT is given in Section 3. The experimental results and ablation analysis are presented in Section 4 and 5, respectively. The limitation and feature works are provided in Section 6. Finally, we draw a conclusion in Section 7.

2. Related works

Recently, several Transformer methods for image denoising have been proposed and the effectiveness of Transformer in this task has been demonstrated. Although these methods have achieved gratifying performance, they do not consider the carrying capacity of the equipment, which is not conducive to 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.

2.1. CNN-based SID methods

As deep learning has progressed, CNN-based Single Image Denoising (SID) methods have demonstrated advanced outcomes. The triumph of these approaches can be ascribed to their robust feature extraction capabilities and intricately designed network structures, enabling the extraction of both coarse and fine-grained features through diverse receptive fields. For instance, Zhang, Zuo, Chen, Meng and and Zhang (2017) introduced DnCNN, a method tailored for Gaussian noise removal, achieving competitive results by leveraging batch normalization and residual learning. Yang and Sun (2017) proposed BM3D-Net, a nonlocal-based network that incorporated BM3D into CNN through wavelet shrinkage. Zhang, Zuo, and Zhang (2018) devised a flexible FFDNet, employing the noise level map and the noisy image as inputs for image denoising. Fang, Li, Yuan, Zeng, and Zhang (2021) put forth a multi-level edge features guided MLEFGN, optimizing the use of edge features for reconstructing noise-free images. Bai, Liu, Yao, Lin, and Zhao (2023) proposed a Multi-Stage Progressive Denoising Network (MSPNet), which decomposes the image denoising task into sub-tasks and progressively removes noise through a series of stages. Wu, Liu, Xia, and Zhang (2024) proposed a Dual-branch Residual Attention Network (DRANet) for image denoising, which has both the merits of a wide model architecture and the attention-guided feature learning. The proposed DRANet includes two different parallel branches, which can capture complementary features to enhance the learning ability of the model. Jiang, Lu, Chen, Lu, and Lu (2023) proposed a Graph Attention in Attention Network (GAiA-Net) for image denoising, which construct a graph from noisy image patches and utilize k-nearest neighbors to initialize edges, enabling the capture of both pixel-level and structure-level features through iterative learning.

Most of the methods discussed above have focused on developing efficient modules for capturing local features to reconstruct noise-free images. Additionally, to recover finer details, several studies (Cui & Knoll, 2023; Park et al., 2019; Yu, Park, & Jeong, 2019; Zhuge, Wang, Xu, & Xu, 2023) have chosen to deepen the network architecture, leading to a significant increase in the number of model parameters. However, this enlargement in parameter count tends to result in larger model sizes and slower execution times, which will hinder their practical application in real-world scenarios. Therefore, our research aim to investigate a more powerful model that is capable of more effectively encoding the global information within images without compromising on computational efficiency.

2.2. Transformer-based IR methods

To capture the intricate dependencies among pixel-level features, recent works have begun to integrate Transformers into image restoration tasks, such as IPT (Chen et al., 2021) and SwinIR (Liang et al., 2021). Among them, IPT draws on the network structure of DERT (Carion et al., 2020), and uses a convolutional layer with a step size of 3 to reduce the dimensionality of the image. Although this approach can alleviate the dimensionality problem, it is very demanding on GPU storage, training dataset, and inference time. SwinIR directly migrated Swin Transformer (Liu et al., 2021) to the image restoration task, and achieved outstanding results. However, SwinIR stacks a

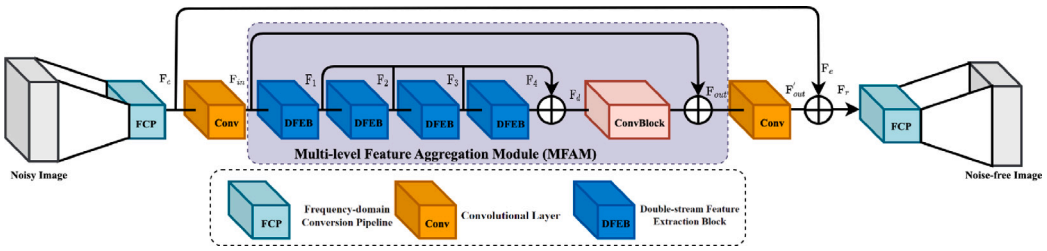


Fig. 1. The complete architecture of the proposed Efficient Wavelet-Transformer (EWT).

large number of Transformers, which consumes a lot of inference time and GPU memory. On this basis, a series of new Transformer image restoration models are proposed, such as Uformer (Wang et al., 2022) and Restormer (Zamir et al., 2022). Among them, Uformer constructs a layered Transformer codec network, which uses a window-based non-overlapping self-attention mechanism to reduce the amount of calculation. Restormer uses a new encode-decode Transformer for global and local representation learning on high-resolution images, without the need for local window splitting, which makes it significantly improved in performance. In addition, some works combine Transformer with graph convolutional networks to achieve better global modeling. For example, Jiang, Lu, Zhang and Lu (2023) proposed an AGP-Net, which employ a novel graph construction method to capture long-range dependencies at both pixel and patch levels. Meanwhile, AGP-Net incorporates graph supplementary prior and graph noise prior to generate supplementary features and regularization noise for improving the performance and generalization of image denoising. After that, Jiang, Li, et al. (2023) also proposed an EFF-Net for SID. EFF-Ne incorporates a Dynamic Hash Attention (DHA) module to mitigate the negative impact of low-weight tokens on denoising performance and an Enhanced Frequency Fusion (EFF) module to separate and fuse noisy image content in the frequency domain, enabling the reconstruction of different frequency components at various locations.

Despite these advancements, however, methods based on Transformers for image restoration still face several challenges. One significant issue is the high computational cost associated with Transformer-based models, which can make real-time processing on resource-constrained devices difficult. Moreover, these models are unfriendly when processing high-resolution images since high-resolution images will take up a large amount of GPU memory. Therefore, directly applying related methods to SID tasks is not the best choice. This work aims to explore a new Transformer architecture that can effectively reduce GPU memory.

2.3. 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 good results. For example, Bae, Yoo, and Chul Ye (2017) found that learning on wavelet sub-bands is more effective, and proposed a Wavelet Residual Network (WavResNet) for image restoration. After that, Guo, Seyed Mousavi, Huu Vu, and Monga (2017) proposed a deep wavelet super-resolution network to recover the lost details on the wavelet sub-bands. Zhong, Shen, Yang, Lin, and Zhang (2018) jointed the sub-bands learning with CliqueNet (Yang, Zhong, Shen, & Lin, 2018) structures for wavelet domain super-resolution. Liu, Zhang, Zhang, Lin, and Zuo (2018) proposed a Multi-level Wavelet-CNN (MWCNN) for image restoration, which use multi-level wavelet to deal with 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.

3. Efficient Wavelet-Transformer (EWT)

As shown in Fig. 1, we first use the FCP to downsample the image, which can effectively separate high-frequency and low-frequency features while reducing the resolution of the image. After that, 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 FCP reverse sampling to restore the image and reconstruct the corresponding noise-free image. Define $I_{noisy} \in H \times W \times C$ as the input noisy image, the FCP down-sampling layer f_{FCP} will convert I_{noisy} into 4 sub-images

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

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. In this work, we concatenate them as the shallow features $F_e \in \frac{H}{2} \times \frac{W}{2} \times 4C$ of EWT, and 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 initial features. And these features are sent to MFAM for further processing. After that, a 3×3 convolutional layer is applied on the output F_{out} to obtain 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, FCP reverse sampling operation is used to transform the features to the original resolution and reconstruct the noise-free image

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

where $f_{FCPr}(\cdot)$ denotes FCP reverse sampling 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.

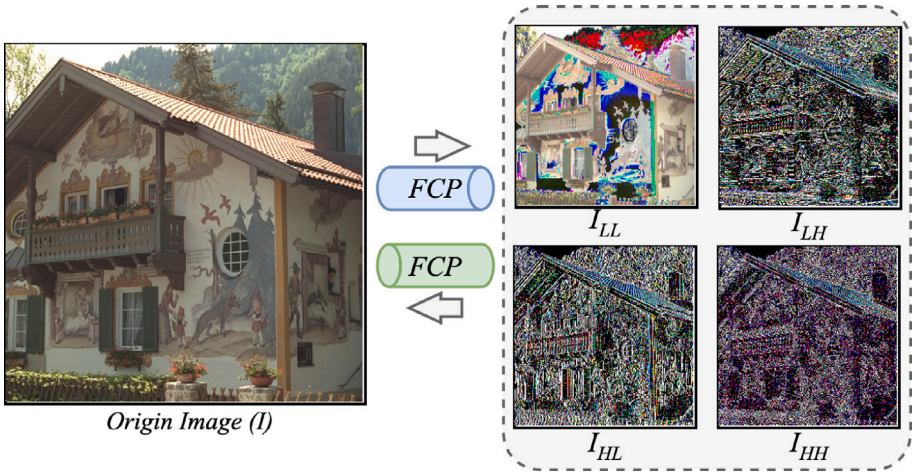


Fig. 2. Schematic diagram of Frequency-domain Conversion Pipeline (FCP).

3.1. Frequency-domain Conversion Pipeline (FCP)

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. Initially, many strategies have been proposed, such as the 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 solve this issue, we proposed a Frequency-domain Conversion Pipeline (FCP) to effectively compress images to reduce image resolution.

As shown in Fig. 2, FCP decomposes the image into four sub-images. They have different frequencies and mainly reflect the color of filled areas and object edges. Specifically, I_{LL} is the low-frequency information of the image, which is an approximation of the original image. I_{LH} and I_{HL} are the frequency information of the horizontal and vertical directions of the image, reflecting the edge characteristics of these two directions. I_{HH} is the diagonal subband of the image, reflecting the diagonal edge features. These sub-images can serve as priors to guide the model to focus on frequency information and help recover texture details. In FCP, we apply the Discrete Wavelet Transform (DWT) as the downsampling module and the Inverse Wavelet Transform (IWT) as the upsampling module. With the help of FCP, all information of the image can be preserved since wavelet transform are reversible. Meanwhile, it can capture frequency and position information, which is beneficial for image restoration. Most importantly, wavelet transform can effectively reduce the image resolution, thus reduce GPU memory consumption.

3.2. Multi-level Feature Aggregation Module (MFAM)

As the core component of the entire model, Multilevel Feature Aggregation Module (MFAM) is specially designed for feature extraction and aggregation in wavelet domain. As shown in Fig. 1, 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 current methods simply stacking Transformer layers, we carefully design a double-stream structural unit (DFEB), and adopt the dense connection to combine the outputs of each DFEB. In this way, hierarchical features of the model can be better aggregated to enhance the feature representation. At the end of the module, a ConvBlock is applied to fuse these different levels of 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 ith 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 for large tensors in Transformer, which consume a lot of GPU memory and computing 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. In contrast, 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. It cannot be compared with the ability of CNN to automatically learn location information. 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 levels. Therefore, we designed the DFEB, 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. In this work, we use Transformer as an alternative to multiple receptive fields. Specifically, Transformer and CNN are used as two branches to extract different features in images.

As shown in Fig. 3, DFEB contains two branches: surface information extraction branch and fine-gained information branch. The features 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. 3), which is a simple module composed of two convolutional layers and a ReLU activation function. This structure benefit for image surface information extraction. The other one, fine-gained information branch

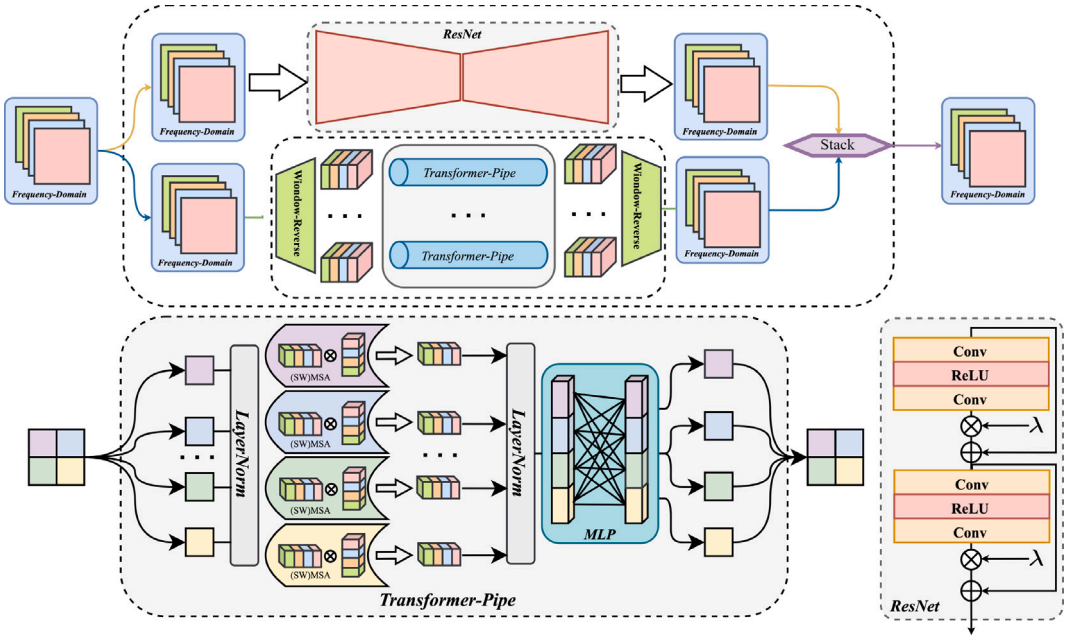


Fig. 3. Schematic diagram of the detailed internal structure of Dual-stream Feature Extraction Block (DFEB).

introduces 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 (Liu et al., 2021) to decompose the feature map into smaller windows. Meanwhile, the window displacement mechanism is applied to enhance the information flow and interaction between windows. As shown in Fig. 3, (SW) MAS denotes the (Shift Window) Multi-Head Self-Attention mechanism proposed by Swin Transformer. Finally, the outputs of CNN and Transformer are concatenated, and convolutional layers are used to weight and fuse different features to guide the module to learn useful features adaptively.

4. Experiments

4.1. Datasets

In this paper, we use 800 training images in DIV2K (Agustsson & Timofte, 2017) as the training set. For evaluation, we choose six benchmark test sets, including Set12 (Zeyde, Elad, & Protter, 2010), BSD68 (Roth & Black, 2005), Kodak24 (Franzen, 1999), CBSD68 (Martin, Fowlkes, Tal, & Malik, 2001), and Urban100 (Huang, Singh, & Ahuja, 2015). 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 CUrban100 to evaluate the denoising effect of model on color images.

4.2. Evaluation metrics

In this work, we use Peak Signal-to-Noise Ratio (PSNR) to evaluate the quality of reconstructed images. PSNR is a metric commonly used to assess the quality of reconstructed images, particularly in the context of image compression and denoising algorithms. It is defined as the ratio of the maximum possible value of a signal to the noise level added when the signal is compressed or otherwise altered. The higher the PSNR value, the closer the reconstructed image is to the original, with a lower level of perceived noise.

4.3. 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 $\sigma = 15$, 25, and 50 for grayscale images and set $\sigma = 10$, 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 DFEBs in MFAM, and each DFEB contains 1 ConvBlock and 6 Transformer blocks. In the Transformer, the window size is set to 8, the number of attention heads is set to 6, and the MLP dimension is as twice as the embedding dimension.

4.4. Comparisons with state-of-the-art methods

Gray-scale Image Denoising: In Table 2, we report PSNR results of different SID methods on three benchmark test sets. Obviously, EWT achieves competitive results on these test sets with different noise levels. It is worth noting that MWCNN is also a wavelet-based SID model, which achieves close results to EWT on BSD68 ($\sigma = 25$ and 50). However, it cannot be ignored that the results of MWCNN under other test sets are worse than EWT, including the average result. 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. 4, we provide the visual comparison of the denoised images with noise levels $\sigma = 50$. According to the figure, we can clearly observe that the images reconstructed by other methods contain a lot of noise and artifacts. In contrast, our EWT can reconstruct high-quality images with more clear and accurate texture details and edges. This illustrates the performance of our EWT.

Table 2
PSNR (dB) comparison with other classic SID methods (BM3D (Dabov, Foi, Katkovnik, & Egiazarian, 2007), WNNM (Gu, Zhang, Zuo, & Feng, 2014), IRCNN (Zhang, Zuo, Gu & Zhang, 2017), DnCNN (Zhang, Zuo, Chen, et al., 2017), FFDNet (Zhang et al., 2018), RED30 (Mao, Shen, & Yang, 2016), SCN (Fan, Yu, Liu, & Huang, 2020), ADNet (Tian, Xu, Li, et al., 2020), MLEFGN (Fang et al., 2021), MSANet (Gou, Hu, Lv, Zhou, & Peng, 2022), and MWCNN (Liu et al., 2018)) on grayscale image test datasets. The best results are **highlighted**.

Method	Set12			BSD68			Urban100		
Noise level	15	25	50	15	25	50	15	25	50
BM3D	32.37	29.97	26.72	31.08	28.57	25.60	32.35	29.70	25.95
WNNM	32.70	30.28	27.05	31.37	28.83	25.87	32.97	30.39	26.83
IRCNN	32.76	30.37	27.12	31.63	29.15	26.19	32.46	29.80	26.22
DnCNN	32.86	30.44	27.18	31.73	29.23	26.23	32.64	29.95	26.26
FFDNet	32.75	30.43	27.32	31.63	29.19	26.29	32.40	29.90	26.50
RED30	32.83	30.48	27.34	31.72	29.26	26.35	32.75	30.21	26.48
SCN	31.99	30.64	27.43	31.80	29.31	26.34	32.99	30.39	26.84
ADNet	32.98	30.58	27.37	31.76	29.35	26.32	33.09	30.41	26.82
MLEFGN	33.04	30.66	27.54	31.81	29.34	26.39	33.21	30.64	27.22
MSANet	33.07	30.71	27.59	31.79	29.35	26.25	32.81	30.41	27.33
MWCNN	33.15	30.79	27.74	31.86	29.41	26.53	33.17	30.66	27.42
EWT (Ours)	**33.25**	**30.89**	**27.83**	**31.90**	**29.43**	**26.55**	**33.57**	**31.10**	**27.72**

Table 3
PSNR (dB) comparison with other classic SID methods (CBM3D (Dabov et al., 2007), IRCNN (Zhang, Zuo, Gu & Zhang, 2017), DnCNN (Zhang, Zuo, Chen, et al., 2017), FFDNet (Zhang et al., 2018), MLEFGN (Fang et al., 2021), RNAN (Zhang, Li, Li, Zhong, & Fu, 2019), RDN (Zhang et al., 2021), MSANet (Gou et al., 2022)) on color image test datasets. The best results are **highlighted**.

Method	Kodak24			CBSD68			CUrban100		
Noise level	10	30	50	10	30	50	10	30	50
CBM3D	36.57	30.89	28.63	35.91	29.73	27.38	36.00	30.36	27.94
IRCNN	36.70	31.24	28.93	36.06	30.22	27.86	35.81	30.28	27.69
DnCNN	36.98	31.39	29.16	36.31	30.40	28.01	36.21	30.28	28.16
FFDNet	36.81	31.39	29.10	36.14	30.31	27.96	35.77	30.53	28.05
MLEFGN	37.04	31.67	29.38	36.37	30.56	28.21	36.42	31.32	28.92
RNAN	37.24	31.86	29.58	36.43	30.63	28.27	36.59	31.50	29.08
RDN	37.31	31.94	29.66	36.47	30.67	28.31	36.69	31.69	29.29
MSANet	37.16	31.78	29.57	36.40	30.67	28.36	36.60	31.62	28.94
EWT (Ours)	**37.31**	**31.96**	**29.67**	**36.52**	**30.72**	**28.39**	**36.73**	**31.86**	**29.57**

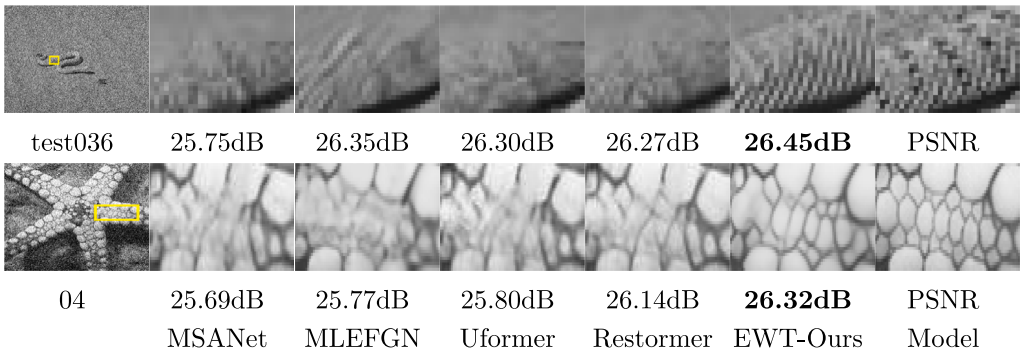


Fig. 4. Visual comparison on grayscale images with $\sigma = 50$. Obviously, our EWT can reconstruct high-quality noise-free images with clear edges.

Color Image Denoising: As for color image denoising, we use Kodak24, CBSD68, and CUrban100 to verify its performance. According to Table 3, we can clearly observe that our EWT still achieves excellent results on color images, especially on CUrban100. 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 parameters (EWT: 11M vs RDN: 22M). These results fully demonstrate the denoising ability of our proposed EWT on color images.

In Fig. 5, we provide the visual comparisons of the denoised images with $\sigma = 50$ on CBSD68. In this part, we also choose four most representative CNN-based image denoising methods for comparison, including MSANet, MLEFGN, Uformer, and Restormer. Obviously, our EWT can reconstruct high-quality noise-free images with sharper and more accurate edges. This is due to the fact that the Transformer introduced in EWT can capture the global information of the image, thereby reconstructing high-quality image. All these results further illustrate the performance of EWT.

4.5. Restoration on 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, Table 4 compares EWT with three classic image restoration Transformer methods. The results show that our EWT also performs well in other noisy images. This further validates the effectiveness of the proposed EWT, and also reflects the generality of EWT on different noisy images.

Fig. 5. Visual comparison on color images with $\sigma = 50$. Obviously, our EWT can reconstruct high-quality noise-free images.

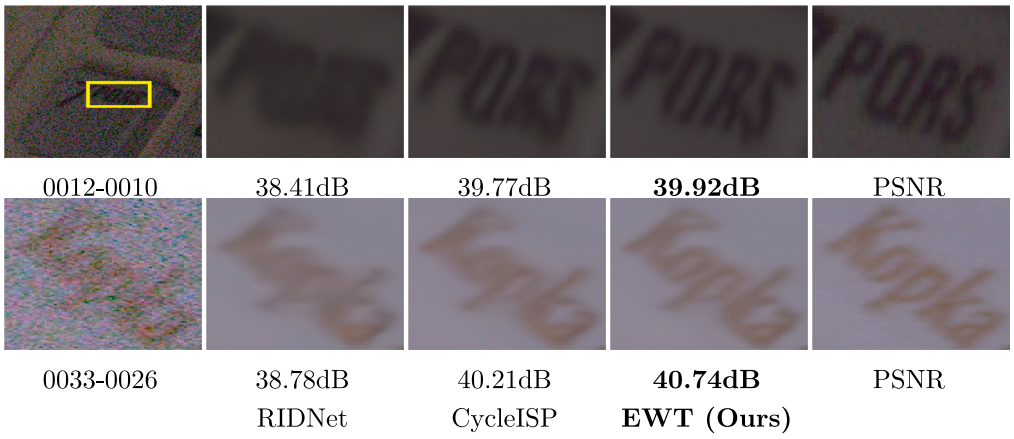


Fig. 6. Visual comparison on real-noise images (SIDD Abdelhamed & Lin, 2018). Obviously, EWT can reconstruct high-quality noise-free images. It is worth noting that SIDD provides ground-truth as label.

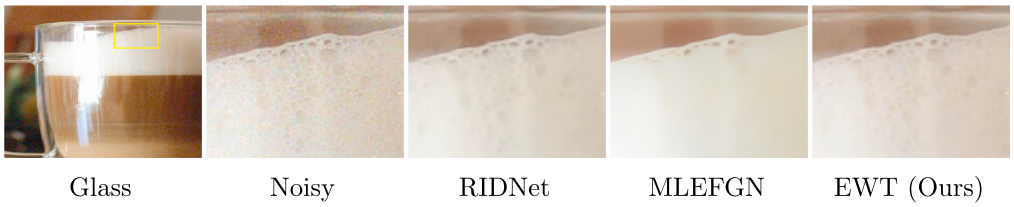


Fig. 7. Visual comparison on real-noise images (RNI15 Lebrun, Colom, & Morel, 2015). Obviously, EWT can reconstruct high-quality noise-free images. It is worth noting that RNI15 do not provide ground-truth as label.

Table 4
Quantitative comparison with other Transformer methods on Poisson and Speckle noise. Best results are **highlight**.

Noise level	Poisson			Speckle		
Method	Kodak24	CBSD68	Urban100	Kodak24	CBSD68	Urban100
SwinIR	37.09 dB	36.44 dB	36.58 dB	31.07 dB	29.87 dB	29.91 dB
Uformer	36.80 dB	36.08 dB	36.20 dB	30.71 dB	29.42 dB	29.72 dB
Restormer	37.14 dB	36.51 dB	36.61 dB	31.01 dB	29.85 dB	29.90 dB
EWT (Ours)	**37.20 dB**	**36.52 dB**	**36.64 dB**	**31.24 dB**	**29.98 dB**	**29.91 dB**

4.6. Restoration on real images

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 5, 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 result 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 (Abdelhamed & Lin, 2018) and RNI15 (Lebrun et al., 2015) sets in Fig. 6 and 7,

Table 5
Real image denoising comparison with DnCNN (Zhang, Zuo, Chen, et al., 2017), BM3D (Dabov et al., 2007), CBDNet (Guo, Yan, Zhang, Zuo, & Zhang, 2019), RIDNet (Anwar & Barnes, 2019), AINDNet (Kim, Soh, Park, & Cho, 2020), VDN (Yue, Yong, Zhao, Meng, & Zhang, 2019), SADNet (Chang, Li, Feng, & Xu, 2020), DANet+ (Yue, Zhao, Zhang, & Meng, 2020), CycleISR (Zamir et al., 2020), DeamNet (Ren, He, Wang, & Zhao, 2021) on SIDD (Abdelhamed & Lin, 2018). Best results are **highlight**.

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

[a] Denote the model using additional training sets to train the model.

Table 6
Study of Multi-level Wavelet on color images (Kodak24, noise level $\sigma = 30$).

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

Table 7
Study of the combination strategy of CNN and Transformer on color images (Kodak24, noise level $\sigma = 30$).

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

respectively. Obviously, EWT still can reconstruct high-quality noise-free images. This demonstrates that EWT also performs well on the real image denoising task.

5. Ablation studies

5.1. Wavelet investigation

In this section, we designed a series of studies in Table 6 to further verify the influence of multi-level wavelet on the model performance. Among them, cases 1, 2, and 3 denote the different levels of wavelet with fixed patch size. According to these results, we can find that when the number of wavelet transforms used increases, the required execution time and GPU memory consumption will be greatly reduced. However, 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 will greatly reduced. However, low-resolution will lead to the loss of local information, 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.

5.2. Research on combination strategies in DFEB

As the most important component of EWT, Dual-branch Feature Extraction Block (DFEB) is designed for feature extraction while reducing the model size and shortening the running time. This is benefit 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 7. 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. It is worth noting that the model using our proposed strategy achieves the best PSNR result and intermediate results on multiple metrics. Therefore, we can conclude that the combination of CNN and Transformer is necessary and effective.

Table 8
PSNR (dB) and parameter quantity comparison with DHDN (Yu et al., 2019) and DIDN (Park et al., 2019) on color image test datasets.

Method	Noise level	DHDN	DIDN	EWT (Ours)
Kodak24	$\sigma = 10$	**37.33**	37.32	37.31
	$\sigma = 30$	31.95	**31.97**	31.96
	$\sigma = 50$	29.67	**29.72**	29.67
CBSD68	$\sigma = 10$	36.45	36.48	**36.52**
	$\sigma = 30$	30.41	30.70	**30.72**
	$\sigma = 50$	28.02	28.35	**28.39**
Parameters		168 M	165 M	**11.8 M**

5.3. 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 (Park et al., 2019; Yu et al., 2019; Zhang et al., 2021) also accompanied by a large number of parameters. In Fig. 8, we provide the performance and parameter comparisons of EWT with other SID models, including IRCNN (Zhang, Zuo, Gu & Zhang, 2017), DnCNN (Zhang, Zuo, Chen, et al., 2017), FFDNet (Zhang et al., 2018), ADNet (Tian, Xu, Li, et al., 2020), BRDNet (Tian, Xu, & Zuo, 2020), MLEFGN (Fang et al., 2021), RNAN (Zhang et al., 2019), RDN (Zhang et al., 2021), DIDN (Park et al., 2019), and IPT (Chen et al., 2021). Among them, the red star represents our 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 (Yu et al., 2019) and DIDN (Park et al., 2019) in Table 8. **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.

5.4. Comparison with SwinIR

In the submitted paper, we compared EWT with SwinIR (Liang et al., 2021) to verify the positive effect of wavelet on the model. Here, we provide more datasets to further verify the effectiveness of EWT. As we mentioned before, SwinIR uses additional training sets and the GPU memory required for it exceeds the maximum limit of our device. Therefore, for a fair comparison, we build a simplified version called SwinIR⋆. The embedding dimension of MFAM in SinwIR⋆ and our EWT⋆ are both reduced from 180 to 120, and these two models are retrained under the same data set and settings. In Table 9, we provide the parameters amount of the two models, the GPU memory occupied during training (patch size: 56×56), the PSNR results and the average execution time on different test sets. According to these results, we can clearly observe that our EWT⋆ achieves close PSNR results (0.04dB–0.06 dB worse than SiwnIR⋆) to SwinIR⋆ with only **1/6** running time and **1/3** GPU memory consumption. This huge breakthrough fully demonstrated the advancement and effectiveness of the proposed EWT.

In Fig. 9 and 10, we provide the visual comparisons with SwinIR (Liang et al., 2021) 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 (Agustsson & Timofte, 2017) (800 training images),

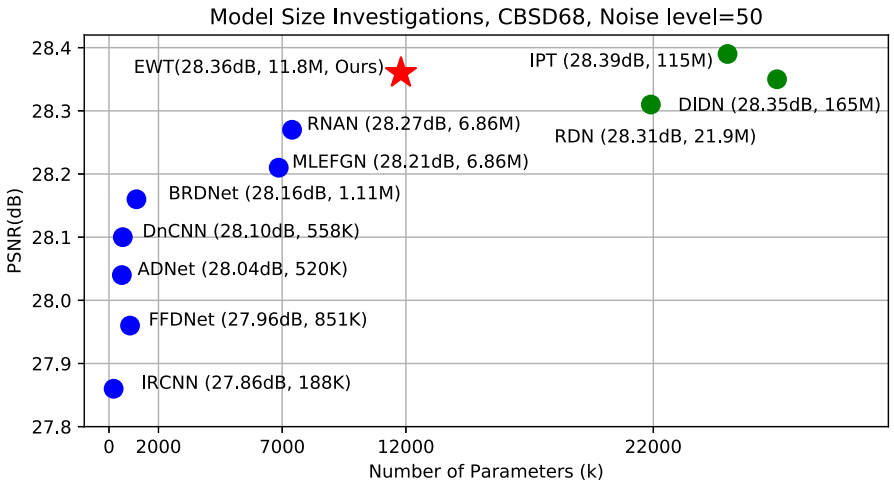


Fig. 8. Model performance and size comparison with other SID methods on CBSD68.

Table 9
Comparison with SwinIR⋆ on Kodak24 (Noise level $\sigma = 30$, color).

Method	Patchsize	GPU	Time	Params	PSNR
SwinIR⋆	56	18432 MiB	53.29 s	5.17 M	**31.79 dB**
EWT⋆	56	**6347 MiB**	**9.14 s**	5.18 M	31.73 dB
$EWT^{\star}_{w/oFCP}$	56	**14032 MiB**	**38.14 s**	5.12 M	31.70 dB
$EWT^{\star}_{Channel-atttion}$	56	**5279 MiB**	**4.62 s**	5.17 M	31.62 dB
$EWT^{\star}_{Pixal-atttion}$	56	**12274 MiB**	**25.38 s**	5.23 M	31.69 dB

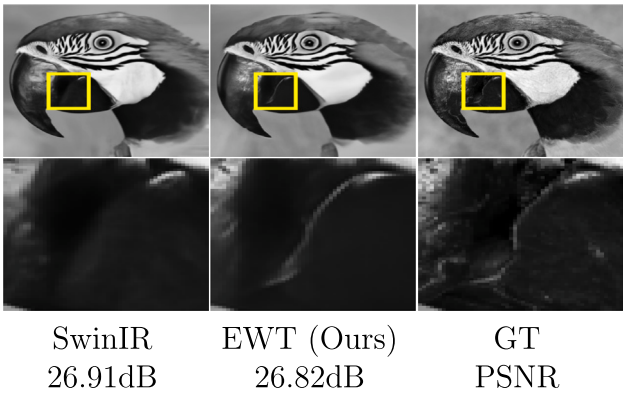


Fig. 9. Visual comparison with SwinIR (Liang et al., 2021) on grayscale image. Obviously, EWT can reconstruct more accurate and clear edges. (Set12 Zeyde et al., 2010, $\sigma = 50$).

Flickr2K (Lugmayr et al., 2019) (2650 images), BSD500 (Martin et al., 2001) (400 training and testing images) and Waterloo Exploration Database (Ma et al., 2016) (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.

5.5. Comparison with MWCNN

In this paper, we proposed a novel EWT for SID. This is the first attempt of Transformer in wavelet domain. As we mentioned in the previous section, EWT was proposed inspired by MWCNN (Liu et al., 2018). Therefore, we give a detailed comparison with MWCNN in Table 10. 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.

5.6. DFEB quantity investigations

We also study the impact of the number of Dual-stream Feature Extraction Blocks (DFEBs) on model performance, execution time, and GPU usage in Table 11. 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 0.17 dB. 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.

5.7. Efficiency investigation

In this Section, we compare with SwinIR (Liang et al., 2021), Uformer (Wang et al., 2022), and Restormer (Zamir et al., 2022) to further verify the efficiency of EWT. All models are retrained under the same dataset and settings. As can be seen from Table 12, EWT achieved better results than Uformer and Restorer with less GPU memory and execution time. It is worth noting that Uformer does improve efficiency through multiple 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. **Compared with SwinIR, the performance of EWT is slightly worse, but the speed is increased by more than 80% and the GPU memory usage is reduced by more than 60%.** In summary, our EWT is a very potential method for SID and provide a new solution for image restoration.

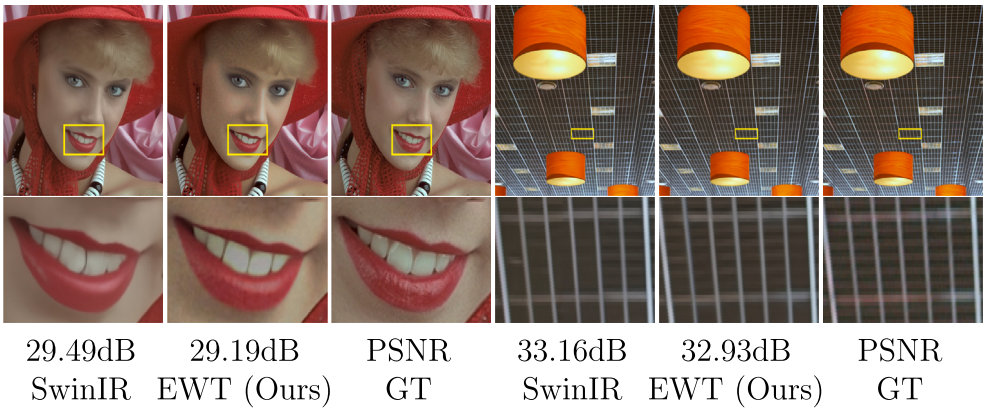


Fig. 10. Visual comparison with SwinIR (Liang et al., 2021) on color image. Obviously, EWT can reconstruct more accurate and clear lines. (σ = 50).

Table 10
Comparison with MWCNN (Liu et al., 2018) on grayscale images. The best results are **highlighted**.

Method	Param	Set12			BSD68			Urban100		
Noise level		15	25	50	15	25	50	15	25	50
MWCNN	19.2 M	33.15	30.79	27.74	31.86	29.41	26.53	33.17	30.66	27.42
EWT (Ours)	11.8 M	**33.25**	**30.89**	**27.83**	**31.90**	**29.43**	**26.55**	**33.57**	**31.10**	**27.72**

Table 11
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	×1	3.34 M	3.01 s	2656 MiB	5.44 G	31.55
2	×2	6.12 M	6.21 s	4934 MiB	9.52 G	31.72
3	×3	8.84 M	9.84 s	6324 MiB	13.69 G	31.75
4	×4	11.8 M	11.96 s	7636 MiB	17.82 G	**31.78**

Table 12
Detailed comparison with Transformer-based method under Gaussian noise condition (Noise level σ = 30).

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

6. Limitation and feature works

Utilizing self-attention mechanisms in the Transformer model is pivotal for capturing global dependencies. However, when this model is employed in low-level visual tasks, it excessively concentrates on semantic interactions among pixels. Consequently, its efficacy in managing images with intricate contextual information or long-term dependencies is diminished, potentially compromising the quality of the final image. Although attempts have been made to mitigate this drawback by integrating CNN branches into EWT, there remains a necessity for further deliberation on how to align these two types of information effectively. Therefore, future research must prioritize addressing the alignment issues between different types of information to enhance the performance and applicability of the model.

7. Conclusion

In this work, a novel Efficient Wavelet-Transformer (EWT) is proposed for single image denoising (SID). This is a new attempt of Transformer in the wavelet domain. Specifically, we introduced a Frequency-domain Conversion Pipeline (FCP). FCP can preserve image features while reducing the image resolution with the help of DWT and IWT. Meanwhile, an efficient Multi-level Feature Aggregation Module (MFAM) is proposed to make full use of hierarchical features. 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 the proposed EWT model significantly improves the efficiency of the original Transformer, with an increase in speed of over 80% and a reduction in GPU memory usage of over 60%. This efficiency makes our model more suitable for real-world applications.

CRediT authorship contribution statement

Juncheng Li: Investigation. **Bodong Cheng:** Methodology. **Ying Chen:** Supervision. **Guangwei Gao:** Validation. **Jun Shi:** Supervision. **Tieyong Zeng:** Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grants 62301306, 62301601, in part by the Science and Technology Commission of Shanghai Municipality under Grant 23ZR1422200, 23YF1412800, 22DZ2229004, in part by the foundation of Key Laboratory of Artificial Intelligence of Ministry of Education under Grant AI202404, and in part by the Fundamental Research Funds for the Central Universities.

References

Abdelhamed, A., & Lin, S. (2018). A high-quality denoising dataset for smartphone cameras. In *CVPR*.

Agustsson, E., & Timofte, R. (2017). Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPRW*.

Anwar, S., & Barnes, N. (2019). Real image denoising with feature attention. In *ICCV*.

Bae, W., Yoo, J., & Chul Ye, J. (2017). Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification. In *CVPR workshops*.

Bai, Y., Liu, M., Yao, C., Lin, C., & Zhao, Y. (2023). MSPNet: Multi-stage progressive network for image denoising. *Neurocomputing*, *517*, 71–80.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *ECCV*.

Chang, M., Li, Q., Feng, H., & Xu, Z. (2020). Spatial-adaptive network for single image denoising. In *ECCV*.

Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., et al. (2021). Pre-trained image processing transformer. In *CVPR*.

Cui, Y., & Knoll, A. (2023). Dual-domain strip attention for image restoration. *Neural Networks*.

Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2007). Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, *16*(8), 2080–2095.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16 × 16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Fan, Y., Yu, J., Liu, D., & Huang, T. S. (2020). Scale-wise convolution for image restoration. In *AAAI*.

Fang, F., Li, J., Yuan, Y., Zeng, T., & Zhang, G. (2021). Multilevel edge features guided network for image denoising. *IEEE Transactions on Neural Networks and Learning Systems*.

Franzen, R. (1999). Kodak lossless true color image suite., source.

Gou, Y., Hu, P., Lv, J., Zhou, J. T., & Peng, X. (2022). Multi-scale adaptive network for single image denoising. In *NeurIPS*.

Gu, S., Zhang, L., Zuo, W., & Feng, X. (2014). Weighted nuclear norm minimization with application to image denoising. In *CVPR*.

Guo, T., Seyed Mousavi, H., Huu Vu, T., & Monga, V. (2017). Deep wavelet prediction for image super-resolution. In *CVPR workshops*.

Guo, S., Yan, Z., Zhang, K., Zuo, W., & Zhang, L. (2019). Toward convolutional blind denoising of real photographs. In *CVPR*.

Huang, J.-B., Singh, A., & Ahuja, N. (2015). Single image super-resolution from transformed self-exemplars. In *CVPR*.

Im, J.-K., Apley, D. W., & Runger, G. C. (2012). Tangent hyperplane kernel principal component analysis for denoising. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(4), 644–656.

Jiang, B., Li, J., Li, H., Li, R., Zhang, D., & Lu, G. (2023). Enhanced frequency fusion network with dynamic hash attention for image denoising. *Information Fusion*, *92*, 420–434.

Jiang, B., Lu, Y., Chen, X., Lu, X., & Lu, G. (2023). Graph attention in attention network for image denoising. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

Jiang, B., Lu, Y., Zhang, B., & Lu, G. (2023). AGP-Net: Adaptive graph prior network for image denoising. *IEEE Transactions on Industrial Informatics*.

Jorgensen, K. W., & Hansen, L. K. (2011). Model selection for Gaussian kernel PCA denoising. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(1), 163–168.

Kim, Y., Soh, J. W., Park, G. Y., & Cho, N. I. (2020). Transfer learning from synthetic to real-noise denoising with adaptive instance normalization. In *CVPR*.

Lebrun, M., Colom, M., & Morel, J.-M. (2015). The noise clinic: A blind image denoising algorithm. *Image Processing on Line*, *5*, 1–54.

Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., & Timofte, R. (2021). SwinIR: Image restoration using swin transformer. In *ICCV*.

Liang, J., Homayounfar, N., Ma, W.-C., Xiong, Y., Hu, R., & Urtasun, R. (2020). Polytransform: Deep polygon transformer for instance segmentation. In *CVPR*.

Liu, F., Jiao, L., & Tang, X. (2019). Task-oriented GAN for PolSAR image classification and clustering. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(9), 2707–2719.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*.

Liu, P., Zhang, H., Zhang, K., Lin, L., & Zuo, W. (2018). Multi-level wavelet-CNN for image restoration. In *CVPR workshops*.

Lugmayr, A., Danelljan, M., Timofte, R., Fritsche, M., Gu, S., Purohit, K., et al. (2019). Aim 2019 challenge on real-world image super-resolution: Methods and results. In *ICCV workshop*.

Ma, K., Duanmu, Z., Wu, Q., Wang, Z., Yong, H., Li, H., et al. (2016). Waterloo exploration database: New challenges for image quality assessment models. *IEEE Transactions on Image Processing*, *26*(2), 1004–1016.

Ma, R., Li, S., Zhang, B., & Li, Z. (2022). Generative adaptive convolutions for real-world noisy image denoising. In *AAAI*.

Mao, X., Shen, C., & Yang, Y.-B. (2016). Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *NeurIPS*.

Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*.

Park, B., Yu, S., & Jeong, J. (2019). Densely connected hierarchical network for image denoising. In *CVPR workshops*.

Ren, C., He, X., Wang, C., & Zhao, Z. (2021). Adaptive consistency prior based deep network for image denoising. In *CVPR*.

Roth, S., & Black, M. J. (2005). Fields of experts: A framework for learning image priors. In *CVPR*.

Shen, H., Zhao, Z.-Q., & Zhang, W. (2023). Adaptive dynamic filtering network for image denoising. In *AAAI*.

Shih, K.-H., Chiu, C.-T., Lin, J.-A., & Bu, Y.-Y. (2019). Real-time object detection with reduced region proposal network via multi-feature concatenation. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(6), 2164–2173.

Shim, J.-h., Yu, H., Kong, K., & Kang, S.-J. (2023). FeedFormer: Revisiting transformer decoder for efficient semantic segmentation. In *AAAI*.

Tian, C., Xu, Y., Li, Z., Zuo, W., Fei, L., & Liu, H. (2020). Attention-guided CNN for image denoising. *Neural Networks*, *124*, 117–129.

Tian, C., Xu, Y., & Zuo, W. (2020). Image denoising using deep CNN with batch renormalization. *Neural Networks*, *121*, 461–473.

Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J., & Li, H. (2022). Uformer: A general u-shaped transformer for image restoration. In *CVPR*.

Wei, L., & Ye, Y. (2022). Fine-grained action segmentation based on complementary frame-level classification model and action-wise regression model. *Displays*, *74*, Article 102212.

Wu, J., Chang, D., Sain, A., Li, X., Ma, Z., Cao, J., et al. (2023). Bi-directional feature reconstruction network for fine-grained few-shot image classification. In *AAAI*.

Wu, W., Liu, S., Xia, Y., & Zhang, Y. (2024). Dual residual attention network for image denoising. *Pattern Recognition*, *149*, Article 110291.

Xu, L., Zhang, J., Cheng, X., Zhang, F., Wei, X., & Ren, J. (2021). Efficient deep image denoising via class specific convolution. In *AAAI*.

Yang, D., & Sun, J. (2017). Bm3d-net: A convolutional neural network for transform-domain collaborative filtering. *IEEE Signal Processing Letters*, *25*(1), 55–59.

Yang, Y., Zhong, Z., Shen, T., & Lin, Z. (2018). Convolutional neural networks with alternately updated clique. In *CVPR*.

Yu, S., Park, B., & Jeong, J. (2019). Deep iterative down-up CNN for image denoising. In *CVPR workshops*.

Yue, Z., Yong, H., Zhao, Q., Meng, D., & Zhang, L. (2019). Variational denoising network: Toward blind noise modeling and removal. In *NeurIPS*.

Yue, Z., Zhao, Q., Zhang, L., & Meng, D. (2020). Dual adversarial network: Toward real-world noise removal and noise generation. In *ECCV*.

Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., & Yang, M.-H. (2022). Restormer: Efficient transformer for high-resolution image restoration. In *CVPR* (pp. 5728–5739).

Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., Yang, M.-H., et al. (2020). Cycleisp: Real image restoration via improved data synthesis. In *CVPR*.

Zeyde, R., Elad, M., & Protter, M. (2010). On single image scale-up using sparse-representations. In *ICCS*.

Zhang, Y., Li, K., Li, K., Zhong, B., & Fu, Y. (2019). Residual non-local attention networks for image restoration. In *ICLR*.

Zhang, J., Liu, H., & Lu, J. (2022). A semi-supervised 3D object detection method for autonomous driving. *Displays*, *71*, Article 102117.

Zhang, Y., Tian, Y., Kong, Y., Zhong, B., & Fu, Y. (2021). Residual dense network for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(7), 2480–2495.

Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2017). Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, *26*(7), 3142–3155.

Zhang, K., Zuo, W., Gu, S., & Zhang, L. (2017). Learning deep CNN denoiser prior for image restoration. In *CVPR*.

Zhang, K., Zuo, W., & Zhang, L. (2018). FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE Transactions on Image Processing*.

Zhong, Z., Shen, T., Yang, Y., Lin, Z., & Zhang, C. (2018). Joint sub-bands learning with clique structures for wavelet domain super-resolution. In *NeurIPS*.

Zhuge, R., Wang, J., Xu, Z., & Xu, Y. (2023). Single image denoising with a feature-enhanced network. *Neural Networks*, *168*, 313–325.