GCSTormer: Gated Swin Transformer with Channel weights for Image Denoising

GCSTormer: Gated Swin Transformer with Channel weights for Image Denoising

Minling zhu[a,*], Zhixin Xu[a], Qi Zhang[b,c], Yonglin Lu[d], Dongbing Gu[e], Sendren Shengdong Xu[f]

[a]*School of Computer Science,Beijing Information Science and Technology University,Beijing,100101,China*
[b]*State Key Laboratory for Novel Software Technology,Nanjing University,Nanjing,Jiangsu,China*
[c]*school of Economics and Management,Harbin Institute of Technology,Weihai,Shandong,China*
[d]*School of Public Management and Media,Beijing Information Science and Technology University,Beijing,100192,China*
[e]*School of Computer Science and Electronic Engineering, University of Essex, CO4 3SQ Colchester, U.K.*
[f]*Department of Power Mechanical Engineering, National Tsing Hua University, Hsinchu 300044, Taiwan,China*

Abstract

In recent years, deep learning models typified by CNN and Transformer have achieved remarkable success in computer vision. However, CNN struggles to capture global information and lacks effective cross-region interaction. Transformer excels in global feature modeling but comes with high computational complexity and a large number of parameters.In this paper, we propose GCSTormer, a novel image denoising model that integrates shallow and deep feature extraction. For shallow feature extraction, we employ a three-layer atrous convolution with dense skip connections. This design enhances CNN's ability to capture global information. In the main architecture of the model, we utilize the U-Net structure to learn multi-scale image information and adopt the improved Swin Transformer block as the basic layer. This block has been improved in two aspects. First, we use gated convolution to optimize information flow. Second,

*Corresponding author.
Email addresses: zhuminling@bistu.edu.cn (Minling zhu), 2512139637@qq.com (Zhixin Xu), hit_zq910057@163.com (Qi Zhang), 13810806131@139.com (Yonglin Lu), dgu@essex.ac.uk (Dongbing Gu), sdxu@pme.nthu.edu.tw (Sendren Shengdong Xu)

we apply transposed attention to compute attention in the channel domain. To evaluate the model performance, we conduct denoising experiments on grayscale and color images with synthetic Gaussian noise. The experimental results show our model has good results in both quantitative metrics and visual quality while maintaining a certain number of parameters and complexity.

Keywords: atrous convolution, U-Net, Swin Transformer, gate mechanism, channel weights, image denoising

PACS: 0000, 1111

2000 MSC: 0000, 1111

---

1. Introduction

With the popularization of digital camera technology and the continuous advancement of image acquisition devices, the problem of image noise has become increasingly prominentZhang et al. (2023a); Tian et al. (2022a). The noise can be caused by various factors. Sensor noise, signal transmission loss, and environmental conditions during image capture are some of themTian et al. (2020a); Fan et al. (2019); Tian et al. (2024c). These noise not only affects the visual quality of images but also the subsequent high-level computer vision tasks. To address these challenges, a variety of image denoising methods have been proposed. These methods range from traditional methods such as filter-based and model-based to deep learning-based methods.

Traditional image denoising methods are based on signal processing and statistical modeling principles. They involve techniques such as filtering, waveform matching, and model optimizationFan et al. (2019). Common methods include Gaussian smoothing and median filtering. There are also more complex algorithms such as BM3DDabov et al. (2007) and WNNMGu et al. (2014). These methods are widely recognized and used. They have a solid mathematical basis and low computational requirements. However, they struggle with complex noise patterns and diverse image structures. Their reliance on manually designed features and parameters limits the adaptability. As a result, they

generalize poorly to real-world scenarios with varying noise characteristics.

In contrast, deep learning-based methods show great potential in overcoming these limitationsTian et al. (2020a), especially those based on CNN and TransformerVaswani et al. (2017); Tian et al. (2024c). CNN can learn complex image representations and denoising patterns directly from large datasets. This enables it to adapt to various types of noise and image structures. CNN models, such as DnCNNZhang et al. (2017a), IRCNNZhang et al. (2017b), and FFDNetZhang et al. (2018), have set a high standard in the domain of image denoising. The transformer, on the other hand, learns the features of the image using a global self-attention mechanism. This global processing capability makes it suitable for image denoising. Understanding the broader context of the values of the pixels improves the precision of the reconstruction. The works Zhang et al. (2023a); Zamir et al. (2022); Liang et al. (2021); Wang et al. (2022); Fan et al. (2022); Tian et al. (2024d) are prominent examples of its application in image denoising. However, CNN struggles to capture long-range pixel dependencies due to its convolutional operations Zamir et al. (2022). This limitation results in the loss of global contextual information. As for Transformer, the global information modeling provides a huge number of parameters. As a result, it requires more computational resources than CNN during both training and inference.

In this paper, we propose GCSTormer for image denoising. The model mainly has two parts: a shallow feature extraction module(SFEM) and a deep feature extraction module(DFEM). The SFEM uses atrous convolution and dense skip connections. Each basic block includes three atrous convolution layers with progressively increasing receptive fields to capture global information. Skip connections are used to facilitate feature interaction. The DEEM employs a U-Net structure for multi-scale feature extraction. Each U-Net layer incorporates an improved block based on Swin Transformer. This improvement addresses two key issues: First, Transformer relies on global information often incorporate noise. To mitigate this, we integrate a gating mechanism at the end of each block to filter information flow. Second, Transformer's pixel

correlation computation in the spatial dimension is computationally intensive. We introduce transposed attention to Swin Transformer. This operation shifts computation to the channel dimension, which further reducing computational cost while enhancing channel processing capability.

The proposed method can be summarized as the following contributions.

- We propose a stacked block utilizing atrous convolution and dense connections. This design enhances CNN's ability to process global information.
- We implement a gate unit within the Swin Transformer block. This modification optimizes information flow and facilitate effective image feature learning while suppressing noise.
- We introduce tranposed attention to Swin Transformer to further reduce the computational complexity. It compute the attention weights from the image channel dimension instead of the spatial dimension.

Remaining parts of this paper can be organized as follows. Section 2 shows related work. Section 3 illustrates the proposed method. Section 4 gives experimental results. Section 5 discussed the limitation of our work and looks ahead to future. Section 6 points out conclusion.

2. Related Work

2.1. Deep CNNs for image denoising

Traditional image restoration methodsDabov et al. (2007); Gu et al. (2014); Tian et al. (2022c) have primarily relied on signal processing and statistical modeling principles. These methods utilize techniques such as filtering, wavelet transforms, and model-based optimizations. They often depend on manually designed features and parameters, which are tailored to specific noise models. This approach limits their adaptability to complex and heterogeneous noise distributions or intricate image structures. Furthermore, these methods are typically computationally intensive, posing challenges for real-time or resource-constrained applications. In contrast, deep learning-based models, such as CNN

and Transformer, have shown superior performance in image denoising tasks. These models learn intricate image representations and denoising patterns directly from the image. Early CNN-based models, including DnCNNZhang et al. (2017a), IRCNNZhang et al. (2017b) and FFDNetZhang et al. (2018), achieved significant improvements over traditional methods despite their relatively simple architectures. These models effectively capture spatial dependencies in images, leading to better generalization and adaptability to diverse noise types.

Overtime, The denoising models based on CNN have been enhanced to improve image denoising performance. The integration of U-Net [19] has become a common strategy to facilitate hierarchical multi-scale feature learning. The U-Net architecture allows for the preservation of spatial resolution during the learning process, enabling the network to better retain and reconstruct fine details in the imagecitePark et al. (2019); Gurrola-Ramos et al. (2021); Zamir et al. (2021); Zhang et al. (2021). Skip connections, as proposed in ResNetHe et al. (2016), mitigate the vanishing gradient problem and enhance information flow between shallow and deep layers, improving the network's ability to handle complex noiseHuang et al. (2017); Park et al. (2019); Liu et al. (2019); Ilesanmi & Ilesanmi (2021). Other works have also seen the incorporation of traditional image processing techniques into deep learning models. For instance, some researchers have integrated traditional filtering methods, such as wavelet transform, into CNN frameworks to enhance the extraction of noise characteristics and improve denoising resultsTian et al. (2023); Liu et al. (2018a); Huang & Dragotti (2022). Additionally, gate mechanismZhu & Li (2023) and attention mechanismTian et al. (2020b) have been employed to selectively focus on more relevant regions of the image. The evolution of image denoising models has also been marked by the development of more sophisticated architectures, including multi-branch and multi-stage networks that progressively refine the denoised image at different scales. These hierarchical methodsZamir et al. (2021); Tian et al. (2023, 2021a); Mei et al. (2023); Tian et al. (2021b, 2022b,d); Zhang et al. (2023b), often coupled with self-supervised learningTian et al. (2024a,b), have significantly enhanced the ability of CNN to generalize across different types of

noise and imaging conditions.

2.2. Transformer methods for image denoising

Transformer, initially designed for NLP, has seen significant adaptation and success in computer vision tasks, such as image classificationDosovitskiy et al. (2020); Touvron et al. (2021); Yuan et al. (2021), object detectionHuang et al. (2023); Liu et al. (2021); Carion et al. (2020); Zhu et al. (2020) and image segmentationXie et al. (2021); Zheng et al. (2021). The core idea of the model is to utilize the self-attention mechanism to capture the global dependencies within the input sequences thereby enabling sequence modeling and information extraction. Vision Transformer(ViT)Dosovitskiy et al. (2020) adapted this concept to images by decomposing an image into a series of patches, similar to how words are treated in a sequence. This approach allows images to be processed similarly to language. However, fundamental differences between images and word sequences still remain, as the computational complexity of Transformer's self-attention grows quadratically with the size of the image patches. This makes the model computationally intensive and complex, especially for high-resolution and pixel-dense image processing tasks, such as image restoration.

Several advancements have been made to optimize the application of Transformer in image denoising. For instance, the SwinIRLiang et al. (2021) leverages the Swin Transformer, which uses a hierarchical approach and computes self-attention within non-overlapping windows rather than across the entire image. This reduces computational complexity while still capturing local and global dependencies effectively. Moreover, SUNetFan et al. (2022) integrates the Swin Transformer with a U-Net structure, further enhancing multi-scale image feature learning. This progressive and multi-level approach allows for a more efficient and powerful feature extraction process, crucial for high-quality image restoration. Another noteworthy advancement is UformerWang et al. (2022), which introduces the Locally-enhanced Window (LeWin) Transformer block. This block combines window-based multi-head self-attention with convolutional operations, allowing the model to capture both long-range dependencies and

essential local context. This design significantly reduces the computational burden compared to global self-attention mechanisms and enhances the model's ability to restore fine image details. Restormer further innovates by introducing Multi-Dconv Head Transposed Attention (MDTA), which computes self-attention across channels rather than spatial dimensions. This approach, along with a hierarchical encoder-decoder Unet structure, allows Restormer to handle high-resolution images efficiently. The Gated-Dconv feed-forward network (GDFN) used in Restormer also improves representation learning by focusing on both local and global contexts. In addition, inspired by the success of the CNN model in image restoration, other worksXue & Ma (2023); Chen et al. (2021); Yao et al. (2022); Wang et al. (2024); Zhao et al. (2022); Li et al. (2024a); Luthra et al. (2021); Xu et al. (2023); Li et al. (2024b); Tian et al. (2024d) have improved the performan of Transformer for image denoising by introducing local information enhancement, combining CNN and traditional filtering algorithms. On one hand, these methods compensate for the shortcomings of Transformer in handling local information. On the other hand, integrating CNN and traditional filtering algorithms alleviate some of the limitations inherent in pure Transformer to some extent.

3. Method

3.1. Network Architecture

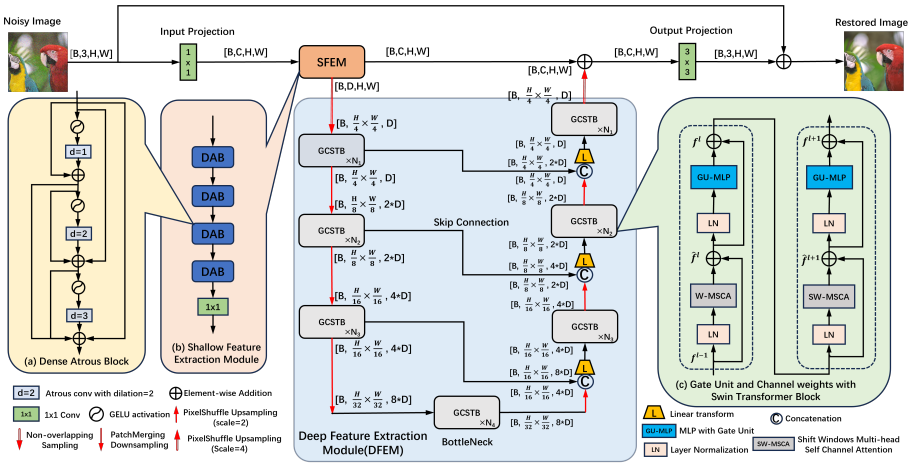

Figure 1: Architecture of the image-denoising model GCSTormer. Our model for image denoising consists of a hierarchical design incorporating key modules. The core modules are: **(i)**SFEM uses DABs to extract initial and shallow features; **(ii)**DFEM uses multiple GCSTB as basic layer, which further processes features for effective denoising. Subfigures (a), (b) and (c) respectively show the structures of DAB, SFEM and GCSTB.

As illustrated in Figure 1, our GCSTormer is mainly composed of two key modules: the SFEM and DFEM. Given an image $X_n \in \mathbb{R}^{H\times W\times 3}$(here we ignore the batch dimension), first we transform it into the structure of $\mathbb{R}^{H\times W\times C}$ through an input projection of a 1 $\times$1 convolution. This will serve as the input of the SFEM. The SFEM consists of four stacked dense atrous blocks(DABs). Each DAB uses cascaded atrous convolutions with dilation rates $r_1 = 1$, $r_2 = 2$ and $r_3 = 3$. Meanwhile, we incorporate a dense skip-connection mechanism, which enhances feature interaction between the lower and higher layers of the block. After that, the feature information processed by the SFEM will be divided into two branches. One branch will be used as the input to the DFEM for deep feature extraction. The other branch is used for feature fusion with subsequent processing results to reconstruct the denoised image. In the DFEM, we use the U-Net for backbone network design. We improve upon the Swin

Transformer block to create our basic block for feature extraction. Finally, we perform feature fusion on the results from the SFEM and DFEM. And then, the denoised image is restored through the output projection. The entire process can be defined as:

$$X_d = GCSTormer(X_n) = SFEM(X_n) \oplus DFEM(X_n) \tag{1}$$

where X_n represents the noisy image, X_d represents the denoised image, and $\oplus$ represents feature fusion.

3.2. Shallow Feature Extraction Module

We design SFEM to effectively capture and extract shallow information from images, as illustrated in subfigure (b) in Figure 1. The SFEM is composed of multiple DABs to enhance feature extraction capabilities. CNN excels at extracting fine details in shallow layers but struggles with global information due to limited receptive fields. To address this, we integrate atrous convolution, which expands the receptive field without increasing parameters, to improve global context capture. Additionally, we implement a dense skip connection mechanism within the DAB to promote interaction and fusion of image information across different layers. This ensures that features extracted at lower levels are retained and effectively integrated into subsequent layers. The entire process is shown as:

$$X^{out}_{SFEM} = SFEM(X^{in}_{SFEM}) = DAB(DAB(DAB(DAB(X^{in}_{SFEM})))) \tag{2}$$

For each DAB, it is composed of a cascaded atrous convolutions with dilation rates $r_1 = 1$, $r_2 = 2$ and $r_3 = 3$ that increase successively. These convolutions are interconnected through dense skip connections to enable efficient information transfer across layers. Given an input X^{in}_{DAB} of DAB, the process of DAB is

represented as:

$$
\begin{aligned}
X_{DAB}^{out} &= \sum_{i=1}^{3} l_{DAB}^{i} + X_{DAB}^{in} \\
l_{DAB}^{i} &= atr_{r=i}(GELU(\sum_{j=1}^{i-1} l_{DAB}^{j} + X_{DAB}^{in}))
\end{aligned}
\tag{3}
$$

Here, X_{DAB}^{out} represents the output of the DAB, l_{DAB}^{i} denotes the output of the i-th layer in a single DAB block. The $atr_{r=i}$ denotes the atrous convolution with dilation rate r= i. GELU($\cdot$)Hendrycks & Gimpel (2016) is the non-linear activation process.

3.3. Deep Feature Extraction Module

Since CNN excels at capturing local structures and low-level details, whereas Transformer leverages global attention mechanisms to model long-range dependencies. It is promising to combine their strengths, and this is also one of the key motivations behind our network design. In the SFEM, we employ a CNN-based algorithm to design a dense cascaded atrous convolutional block for preliminary image feature extraction. In the DFEM, we design a multi-scale deep feature extraction network based on the Transformer. This hierarchical integration enables an effective balance between local precision and global contextual understanding.

As we can see from Figure 1, DFEM follows a U-Net structure, progressively downsampling the feature maps while increasing the number of channels. In the encoding stage of DFEM, we follow the basic principles of U-Net design. At each corresponding layer, the spatial dimensions of the features are downsampled to half, until they reach 1/32 of the original size. Meanwhile, the channel dimension is doubled to facilitate the extraction of deeper image information. Considering that we have already conducted preliminary shallow feature extraction in SFEM and inspired by SUNet, we apply a 4$\times$ non-overlapping downsampling before the first corresponding layer to reduce the computational load in the deep network.

The core functional block in DFEM is GCSTB, which is a improved Swin Transformer block with a gate unit and channel weights. In the field of image

denoising, the Swin Transformer has achieved great achievementsLiang et al. (2021); Fan et al. (2022). As shown in Figure 1(c), there are mainly two improvements in GCSTB: First, gated convolution is added to the feedforward neural network layer to enhance information flow. We call it GU-MLP, the MLP with gate unit. Second, transposed attention is introduced to compute the attention weights from the channel dimension(H×W → C×C). These new blocks are named with W-MSCA and SW-MSCA. In addition, to facilitate layer normalization, we reshape the feature tensor from $\mathbb{R}^{H\times W\times C}$ to $\mathbb{R}^{H*W\times C}$. The entire process of GCSTB is represented as:

$$\begin{aligned}
\hat{f}^{l} &= W-MSCA(LN(f^{l-1})) + f^{l-1} \\
f^{l} &= GU-MLP(LN(\hat{f}^{l})) + \hat{f}^{l} \\
\hat{f}^{l+1} &= SW-MSCA(LN(f^{l})) + f^{l} \\
f^{l+1} &= GU-MLP(LN(\hat{f}^{l+1})) + \hat{f}^{l+1}
\end{aligned} \tag{4}$$

Here, f^{l-1} is the input of GCSTB and f^{l+1} is the output of GCSTB. $\hat{f}$, f^{l} and $\hat{f}^{l+1}$ are the intermediate outputs.

3.3.1. ***GU-MLP***

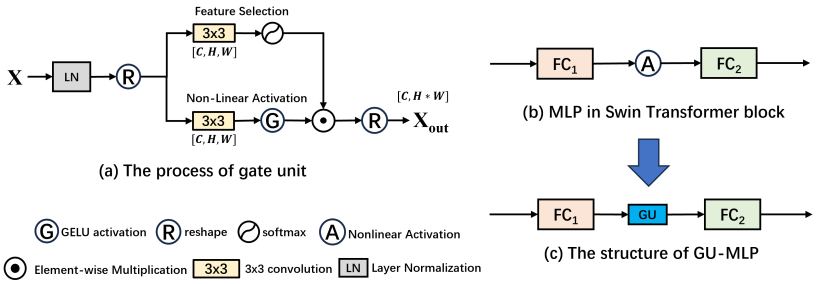

Figure 2: The principle and process of GU-MLP. GU-MLP incorporates gated convolution into the MLP layer of Swin Transformer, replacing the original nonlinear activation process. This gated activation unit adaptively learns image and noise feature information to enhance information flow within the network. Subfigures: (a) The processing procedure of the gate unit; (b) The original MLP structure in Swin Transformer; (c) The structure of GU-MLP.

For models like Transformer with global modeling capabilities, it is inevitable

to learn noise features during the feature processing. It is necessary for the model to learn to distinguish between image features and noise features. Faced with this problem, we use gated convolutionYu et al. (2019) to improve the MLP layer of Swin Transformer block to enhance the model's adaptive learning ability and improve the information flow. In conventional convolution, all pixels are treated equally as valid pixels without any distinction. This indiscriminate processing method may cause noise and less relevant features to propagate in the network. To address this issue, gated convolution introduces a dynamic feature selection mechanism that adjusts the information flow for each spatial location and each channel respectively. By selectively controlling which features are allowed to pass through, it effectively enhances the network's ability to suppress noise and focus on more meaningful image information during the processing. Therefore, we apply this technique to the model design.

As Figure 2 (b) shows, the MLP in Swin Transformer block is composed of two fully connected layers, FC_1 and FC_2. The first fully connected layer, FC_1, expands the number of feature channels(typically with an expansion factor of $\gamma = 4$) to increase the network's capacity for learning complex representations. After being processed by FC_1, the data will go through an non-linear activation function to enhance the network's non-linear representation ability. The second layer, FC_2, then reduces the channels back to their original dimensionality, ensuring that the expanded features are compressed to fit the subsequent layers' requirements. We add gated convolution during nonlinear activation to form a gating unit. It replaces the original simple nonlinear activation processing. In the gate unit, the input feature map splits into two branches. One branch is for nonlinear activation. The other is for feature selection. In each branch, we use a 3×3 convolution to learn local feature information. The entire process is represented as:

$$
\begin{aligned}
X' &= R(LN(X)) \\
X_g &= \phi(conv(X')) \\
X_f &= GELU(conv(X')) \\
X_{out} &= R(X_g \odot X_f)
\end{aligned} \tag{5}
$$

where X and $X_{out} \in \mathbb{R}^{H*W \times C}$ are the input and output of the gate unit respectively. X_g and $X_f \in \mathbb{R}^{H \times W \times C}$ respectively represent the gating weight and the non-linear activation feature. LN represents layer normalization. $R(\cdot)$ denotes the reshaping operation. $\phi(\cdot)$ is the softmax function. $conv(\cdot)$ is the convolution operation. $GELU(\cdot)$ represents the GELU non-linear activation processing. $\odot$ represents element-wise multiplication.

3.3.2. ***W-MSCA and SW-MSCA***

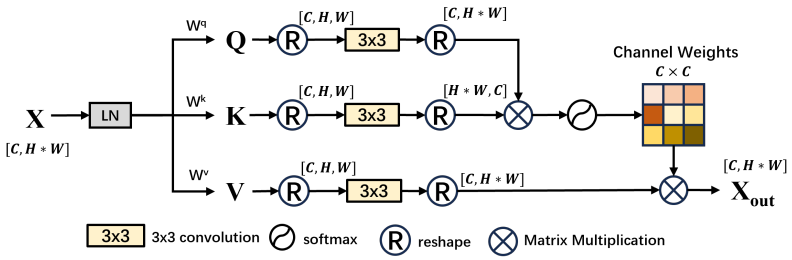

Figure 3: The process of tranposed attention in W-MSCA and SW-MSCA. Transposed Attention computes attention from the channel dimension rather than the spatial dimension. Channel weights are derived from Q and K to achieve both computational reduction and feature selection. A 3×3 convolutional block is employed for extracting local feature information.

In the Swin Transformer block, we introduce transposed attention into its window multi-head self-attention(W-MSA) and shifted window multi-head self-attention (SW-MSA), thereby forming W-MSCA and SW-MSCA. In the original Transformer, the multiplication of the three matrices Q, K, and V is calculated from the spatial perspective, i.e., H×W. Transposed attention converts the spatial dimension computation into channel dimensions, i.e., H×W→C×C. This approach effectively reduces the computational load of the Transformer. Ad-

ditionally, this processing establishes connections between different channels, enabling feature interaction. It selectively emphasizes or suppresses features of certain channels, thereby better capturing important information. This helps filter out more valuable features for the task within complex feature spaces.

Figure 3 shows the operation process of this attention. As shown in the figure, for the input feature map X, after normalization, it is mapped to the Q, K, and V projection matrices through three learnable weights W^q, W^k, and W^v. Subsequently, we apply a 3×3 convolution to each of these feature maps to extract local features. The attention converts the Q and K matrices into the forms of $\mathbb{R}^{H*W\times C}$ and $\mathbb{R}^{C\times H*W}$ respectively. This is used for matrix multiplication and then utilize softmax function to scale the values to the range of 0-1 . Finally, these obtained channel weights $CW \in \mathbb{R}^{H*W\times C}$ are multiplied with V to produce the output. The entire process is represented as:

$$\begin{aligned} Q, K, V &= W^q X, W^k X, W^v X \\ Q', K', V' &= conv(Q, K, V) \\ X_{out} &= V' \cdot softmax(\frac{Q' \cdot K'}{\alpha}) \end{aligned} \tag{6}$$

Here, α is a learnable scaling parameter used to control the magnitude of the dot product of K and Q before applying the $softmax$ function. Q', K' and V' are intermediate output results.

4. Experiment

4.1. Experimental Datasets

The training datasets are conducted on the image super-resolution datasets DIV2KAgustsson & Timofte (2017) and Flickr2KTimofte et al. (2017). The DIV2K contains a total of 1000 images with an average resolution of 1920×1080. Among these, 800 images are used for the training set, 100 for validation and 100 for testing. The Flickr2K dataset includes a total of 2650 images with an average resolution of 2040×1350. To preserve the original image distribution and avoid distortions caused by scaling, we employed random cropping during

training. Each image for training was cropped into 16 smaller patches with size 128×128. We add synthetic gaussian noise with a standard deviation σ ranging from 10 to 50 to these patches.

To test the performance of the model, we conduct denoising experiments on grayscale and color datasets. For grayscale images, we use datasets Set12Zhang et al. (2017a) and BSD68Martin et al. (2001). Set12 is a standard benchmark composed of 12 widely used images. BSD68 consists of 68 grayscale images from the Berkeley Segmentation Dataset. For color image denoising, we test on three well-known datasets: CBSD68Martin et al. (2001), Kodak24Franzen (1999), and McMasterZhang et al. (2011). The CBSD68 includes 68 color images, each with a resolution of 768×512, serving as a color counterpart to the grayscale BSD68 dataset. The Kodak24 consists of 24 images with a resolution of 500×500, widely recognized for its natural image content. The McMaster contains 18 high-quality color images with resolution of 500×500. Figure 4 presents visual samples from both grayscale and color testing datasets.

Figure 4: The testing samples from Set12 and Kodak24. Set12 is a standard benchmark composed of 12 widely used images. Kodak24 consists of 24 images with a resolution of 500×500, widely recognized for its natural image content.

4.2. Experimental Setup

4.2.1. Implementation Details

We conduct our experiments in a GPU-accelerated environment using PyTorch 1.10.0+cu118 and utilize a single NVIDIA GTX 3090 GPU for accelerated

training. The total number of epochs was set to 200 with a batch size of 8. We employ ADAMWKingma & Ba (2014); Loshchilov & Hutter (2017) optimizer with the $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$. The initial learning rate was 2e-4, and the minimum learning rate was 1e-6.

We conduct denoising experiments and test the model inference time on a personal laptop. The configuration of the personal laptop is as follows: 12th Gen Intel(R) Core(TM) i7 - 12650H with a clock speed of 2.30 GHz, 32.0 GB of RAM, and it is equipped with a GPU of RTX4060.

4.2.2. Loss Function and Evaluation metrics

We use the conventional L_1 loss to optimize the model. The formula is as follows:

$$\mathcal{L}_{denoise} = \frac{1}{N}\sum_{i=1}^{N}|f(x_i) - y_i| \tag{7}$$

where N represents the total number of training samples, $f(x_i)$ is the predicted output for the i-th input image x_i, and y_i is the corresponding ground truth image. The absolute difference between $f(x_i)$ and y_i is computed, summed over all samples, and then averaged.

Peak Signal-to-Noise Ratio (PSNR) is used as one of evaluation metrics for denoising performanceT ts formula is shown as:

$$PSNR = 10log_{10}(\frac{MAX^2}{MSE}) \tag{8}$$

where MAX is the maximum pixel value of the image. MSE (Mean Squared Error) represents the difference between the reference image and the reconstructed image, and is calculated as:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[\mathbf{I}(i,j) - \mathbf{K}(i,j)]^2 \tag{9}$$

where $\mathbf{I}(i,j)$ is the original image pixel, $\mathbf{K}(i,j)$ is the denoised image pixel. M and N are the spatial dimensions of the image.

SSIM(Structural Similarity Index) is an another image quality assessment metric based on structural information. Compared with PSNR, it is more in

line with the visual perception of the human eye. SSIM is calculated from three aspects: luminance, contrast, and structure. The higher the SSIM value, the more similar the processed image is to the original image. The formula for SSIM is shown as:

$$SSIM = \frac{(2\mu_a\mu_b + C_1)(2\sigma_{ab} + C_2)}{(\mu_a^2 + \mu_b^2 + C_1)(\sigma_a^2 + \sigma_b^2 + C_2)} \tag{10}$$

where a and b represent the original image and the denoised image respectively. μ_a and μ_b denote the mean intensity of a and b, while σ_a and σ_b represent their intensity variances. σ_{ab} represents the covariance between a and b. $C_1 = (K_1L)^2$ and $C_2 = (K_2L)^2$ are constants used to avoid instability when the denominator approaches zero. K_1=0.01, K_2=0.03, and L is the range of pixel values. In this paper, images are represented with 8 bits, so L=255.

4.3. Performance of Image Denoising

To fully test denoising effect of the proposed GCSTormer, we apply quantitative and qualitative metrics to evaluate it. Quantitative evaluation mainly uses some state-of-the-arts, i.e., BM3DDabov et al. (2007), WNNMGu et al. (2014), DnCNNZhang et al. (2017a), image restoration CNN (IRCNN)Zhang et al. (2017b), attention-guided denoising convolutional neural network (ADNet)Tian et al. (2020b), FFDNetZhang et al. (2018), multi-level wavelet CNN (MWCNN)Liu et al. (2018b), multi-scale adaptive network (MSANet)Gou et al. (2022), densely connected hierarchical image denoising network(DHDN)Park et al. (2019), residual dense U-Net neural network (RDUnet)Gurrola-Ramos et al. (2021), efficient transformer for high-resolution image restoration (Restormer) (Zamir et al. (2022)), swin transformer UNet (SUNet)Fan et al. (2022), image restoration using swin transformer (SwinIR)Liang et al. (2021), efficient wavelet transformer (EWT)Li et al. (2024a) and a cross Transformer for image denoising (CTNet)Tian et al. (2024d) on noisy images for image denoising. Specifically, synthetic noisy images contain gray and color synthetic noisy images.

4.3.1. Gray image denoising

For gray synthetic noisy image denoising, we choose Set12Zhang et al. (2017a), BSD68Martin et al. (2001) to evaluate denoising performance of our GCSTormer as shown in table 1. According to the table, GCSTormer demonstrates notable performance. In the dataset of Set12 , when σ=15, GCSTormer achieves the PSNR value of 33.22dB, which is relatively high compared to models like DnCNN, FFDNet and MWCNN. This indicates its effectiveness in handling low-level noise. At high noise level of σ =50, the value of 27.77dB also shows a competitive result compared with EWT and CTNet. However, when compared to SwinIR, GCSTormer shows a slight performance gap, particularly at lower noise levels. For example, SwinIR achieves the highest PSNR of 33.42 at σ =15. This suggests that while GCSTormer is highly effective, there is room for further optimization to match the state-of-the-art performance of SwinIR. As for BSD68, the performance of GCSTormer is also commendable. At noise levels of σ =15, σ =25, and σ =50, GCSTormer achieves PSNR values of 31.92 dB, 29.41 dB, and 26.55 dB respectively. Compared to traditional methods like BM3D and WNNM, it shows significant improvements with almost 1dB gap. Even with the state-of-the-art models like EWT, CTNet and SwinIR, the the gap remains around 0.05 dB. This demonstrates the great advantages of our model in grayscale image denoising.

For qualitative evaluation, we use several popular denoising methods as comparative methods to conduct visual figures for testing denoising effects of the proposed GCSTormer. That is, we choose an area of denoising images from different methods as an observation area, the observation area is clearer, and its corresponding method is more effective for image denoising. We take the "starfish" image from the Set12 and the "test039" image from the BSD68 dataset as examples for visual demonstration. As shown in Figs. 5 and 6, these two groups of images respectively display the original images, noisy images, and the denoising results of methods such as BM3D, DnCNN, FFDNet, and SwinIR under the two datasets. According to the PSNR evaluation results,

Table 1: Average PSN(dB) of different methods on Set12 and BSD68 with σ =15, 25 and 50. The highest PSNR is highlighted in red font and the second highest one is highlighted in blue font.

Method	Set12			BSD68		
	σ =15	σ =25	σ =50	σ =15	σ =25	σ =50
BM3D	32.37	29.97	26.72	31.08	28.57	25.60
WNNM	32.70	30.28	27.05	31.37	28.83	25.87
DnCNN	32.86	30.44	27.18	31.73	29.23	26.23
IRCNN	32.76	30.37	27.12	31.63	29.15	26.19
FFDNet	32.75	30.43	27.32	31.63	29.19	26.29
FOCNet	33.07	30.73	27.68	31.83	29.38	26.50
MWCNN	33.15	30.79	27.74	31.86	29.41	26.53
ADNet	32.98	30.58	27.37	31.76	29.35	26.32
MSANet	33.07	30.71	27.59	31.79	29.35	26.25
EWT	33.25	30.89	27.83	31.90	29.43	26.55
CTNet	33.31	30.94	27.79	31.94	29.46	26.49
SwinIR	33.42	31.01	27.91	31.97	29.50	26.58
GCSTormer	33.22	30.81	27.77	31.92	29.41	26.55

the proposed GCSFormer has high values in both groups of images. This indicates its excellent performance in noise removal. From the locally magnified areas, it can be seen that the model performs remarkably well in restoring image details and is comparable to the best-performing model SwinIR. For example, as shown in Fig. 5, the result processed by GCSFormer shows a clearer hexagonal grid structure in the surface structure of the starfish; As for the fish from the BSD68, GCSFormer is likely to be able to restore the contour of the fish's eye and the tiny details around it more clearly. In terms of visual effects, compared with the methods like BM3D, DnCNN, and IRCNN, the images denoised by our model are smooth and rich in details, without obvious artifacts or blurring

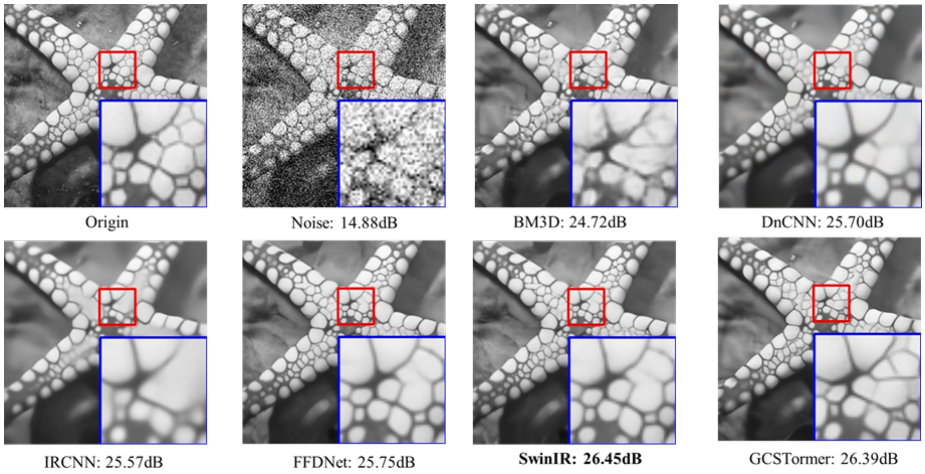

Figure 5: Visual comparisons for gray image denoising on image '04.png' from Set12 dataset corrupted by AWGN with $\sigma = 50$. The PSNR value below the subfigures are calculated by patches.

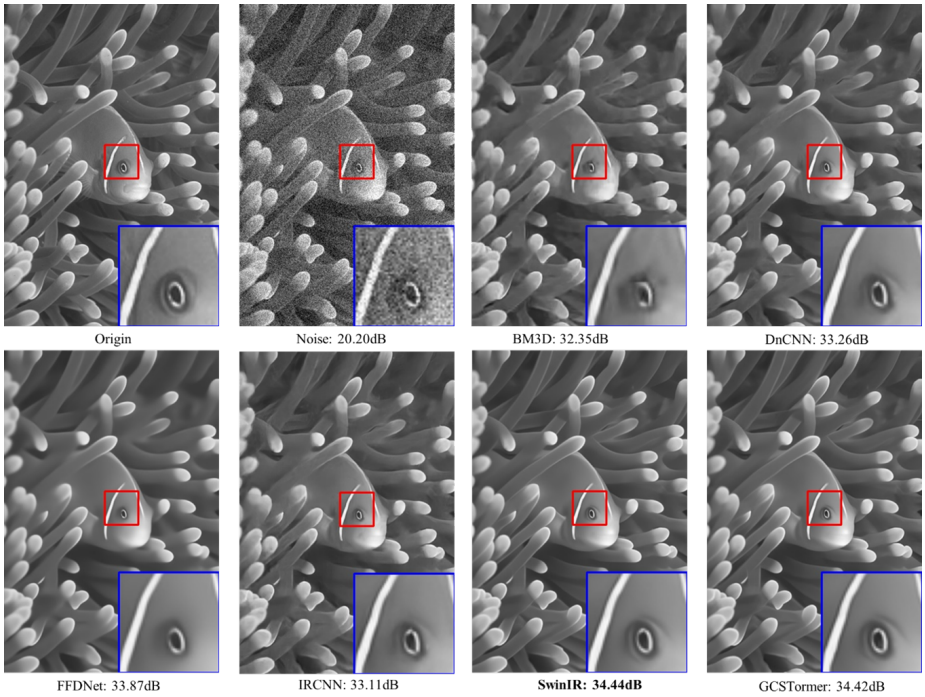

Figure 6: Visual comparisons for gray image denoising on image 'test036.png' from BSD68 dataset corrupted by AWGN with $\sigma = 50$. The PSNR value below the subfigures are calculated by patches.

4.3.2. Color image denoising

For evaluating color image denoising, we tested our model using three prominent datasets: CBSD68, Kodak24, and McMaster. Tables 2- 4 show the image denoising results on these datasets with PSNR and SSIM. The denoising test is conducted at noise levels σ =10, 30 and 50. According to the results, GCSTormer demonstrates remarkable performance advantages. On the CBSD68, GCSTormer achieved high PSNR and SSIM values in denoising at different noise levels. These results are very close to those of the best-performing methods, such as RDUNet and EWT. In terms of SSIM, GCSTormer even has slight advantages. Regarding Kodak24, GCSTormer had obvious advantages at a low noise level(σ =10),its value of SSIM was significantly improved compared with classic CNN models like DnCNN, FFDNet. At a high noise level of σ =50,it also maintains comparable performance to the EWT and Restormer. As to McMas-

ter, GCSTormer also performed well. Especially at σ =50, its PSNR reaches 30.05dB, exceeding other methods and second only to Restormer.

Table 2: Image denoising results on dataset CBSD68. The best result and the second best are respectively highlighted in red and blue.

Methods	CBSD68					
	σ =10		σ =30		σ =50	
	PSNR(dB)	SSIM	PSNR(dB)	SSIM	PSNR(dB)	SSIM
Noise	28.28	0.8253	19.03	0.4698	15.00	0.3060
CBM3D	35.89	0.9510	29.71	0.8430	27.36	0.7633
DnCNN	36.12	0.9510	30.34	0.8620	27.9	0.7897
IRCNN	36.06	0.9530	30.23	0.8612	27.88	0.7898
FFDNet	36.14	0.9540	30.32	0.8608	27.97	0.7887
U-Net	35.39	0.9481	29.74	0.8490	27.35	0.7710
DHDN	36.05	0.9533	30.12	0.8580	27.71	0.7870
RDUNet	36.48	0.9512	30.72	0.8720	28.38	0.8070
Restormer	-	-	-	-	28.59	-
SUNet	35.94	0.9581	30.28	0.8700	27.85	0.7990
EWT	36.52	-	30.72	-	28.39	-
GCSTormer	36.49	0.9589	30.58	0.8726	28.50	0.8077

Table 3: Image denoising results on dataset Kodak24. The best result and the second best are respectively highlighted in red and blue. All of scores are the average values of the whole dataset.

Methods	Kodak24					
	σ =10		σ =30		σ =50	
	PSNR(dB)	SSIM	PSNR(dB)	SSIM	PSNR(dB)	SSIM
Noise	28.22	0.7971	18.926	0.4146	14.8676	0.2596
CBM3D	33.32	0.9430	27.75	0.7730	25.6	0.6860
DnCNN	36.58	0.9450	31.17	0.8569	28.83	0.7909
IRCNN	36.70	0.9450	31.12	0.8570	28.8	0.7929
FFDNet	36.80	0.9460	31.27	0.8584	28.98	0.7942
U-Net	35.89	0.9390	30.55	0.8450	28.11	0.7740
DHDN	37.30	0.9510	31.98	0.8740	29.72	0.8170
RDUNet	37.29	0.9010	31.97	0.8740	29.72	0.8180
Restormer	-	-	-	-	30.00	-
SUNet	36.79	0.9530	31.82	0.8990	29.54	0.8100
EWT	37.31	-	31.96	-	29.67	-
GCSTormer	37.30	0.9543	31.91	0.8746	29.81	0.8175

Table 4: Image denoising results on dataset McMaster. The best result and the second best are respectively highlighted in red and blue. All of scores are the average values of the whole dataset.

Methods	McMaster					
	σ =10		σ =30		σ =50	
	PSNR(dB)	SSIM	PSNR(dB)	SSIM	PSNR(dB)	SSIM
Noise	25.13	0.6683	19.418	0.4183	15.3826	0.2601
DnCNN	33.45	0.9035	30.79	0.8539	28.62	0.7984
IRCNN	34.58	0.9195	31.31	0.8643	28.93	0.8070
FFDNet	34.66	0.9216	31.53	0.8701	29.19	0.8150
Restormer	35.55	-	-	-	30.29	-
CTNet	-	-	-	-	30.02	-
GCSTormer	35.43	0.9436	32.26	0.8980	30.05	0.8462

For qualitative evaluation, we select "seagull" image from CBSD68, "hat" image from Kodak24, and "cartoon" image from the McMaster. Fig. 7 - 9 show the original images, noisy images, and the denoised results with PSNR and SSIM values annotated. In the seagull image, the noisy image has significant noise points. Although CBM3D improves the image, it remains blurry. DnCNN and IRCNN yield similar PSNR and SSIM results, while RDUNet and GCSTormer show notable improvements. From the enlarged view of the seagull's neck, it is evident that the proposed model retains more feather details during denoising. The denoised image is smooth and free of artifacts, achieving comparable denoising effects to RDUNet. When denoising the hat from Kodak24, GCSFormer has get the highest PSNR of 33.38 dB and the SSIM of 0.8870. From the visualization results, GCSTormer performs well in retaining the stroke details of the text on the hat. The denoised image is bright and clear, with obvious advantages in restoring the text structure and texture. Fig. 9 presents a vivid cartoon image. From both the overall denoised image and the enlarged view of the kitten, GCSTormer produces a brighter and clearer image with more distinct contours. In contrast, although DnCNN and IRCNN remove noise to a certain extent, the overall image quality is low, with a large amount of blurring, and the object details are also damaged. In terms of the quantitative results of PSNR and SSIM, GCSTormer also shows significant advantages.

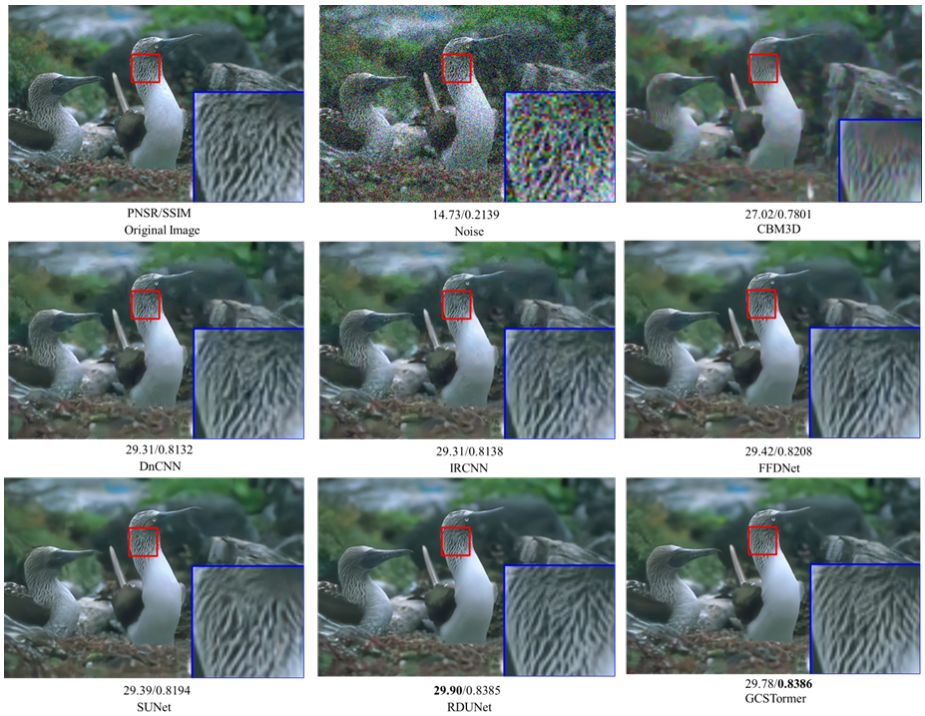

Figure 7: Visual comparisons for image denoising on image '103070' from CBSD68 dataset corrupted by AWGN with $\sigma = 50$. The PSNR and SSIM values below the subfigures are calculated by patches.

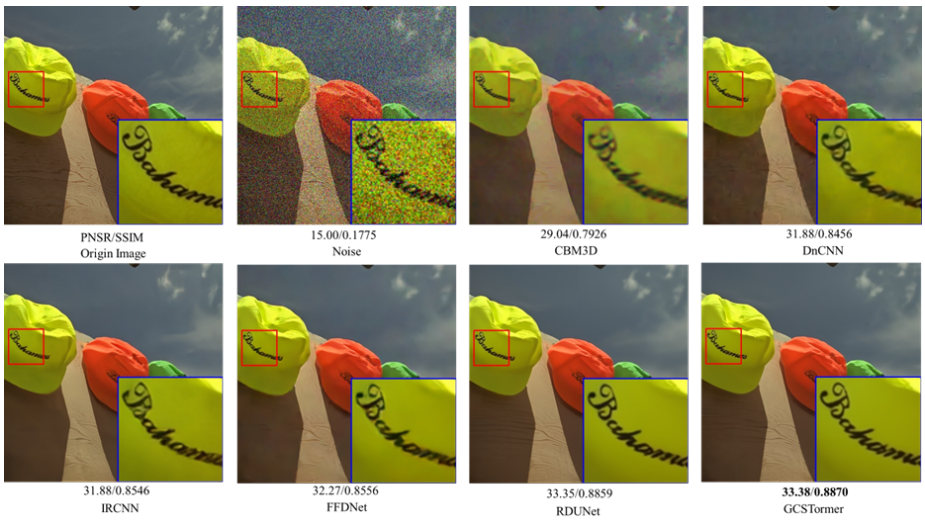

Figure 8: Visual comparisons for image denoising on image 'kodim01' from Kodak24 dataset corrupted by AWGN with $\sigma = 50$.

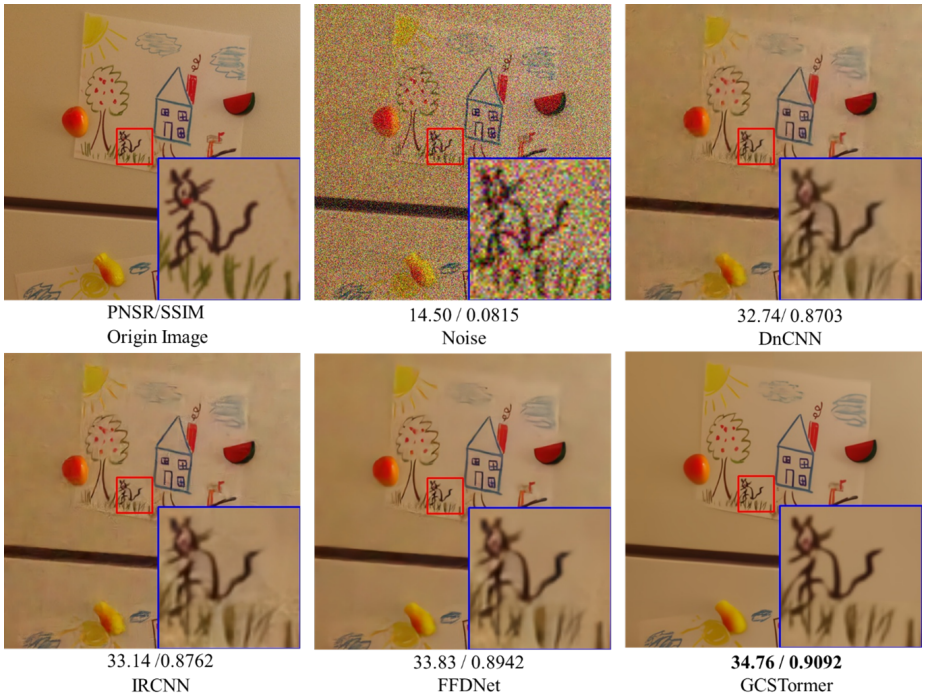

Figure 9: Visual comparisons for image denoising on image '05.png' from McMaster dataset corrupted by AWGN with $\sigma = 50$.

4.4. Data Analysis

4.4.1. Performance of the Model

In the above text, we use PSNR and SSIM for quantitative evaluation and do qualitative analysis by visualizing denoised images. However, for deep-learning model, the number of parameters and computational complexity are also important. We conduct comparative experiments on parameters and floating point operations(Flops.) to analyze performance of our proposed model, as the table 5 shows. Regarding the number of parameters, DnCNN, IRCNN, and FFDNet have relatively few. This clearly shows the advantages of CNN methods. In contrast, denoising methods using the U-Net structure, like DHDN and RDUNet, have a much larger number of parameters. GCSTormer has a smaller parameters of 52 million bit, that means it has more advantages in terms of model storage and computational resource requirements. In terms of Flops. , GCSTormer is also at a low level comparable to models like DnCNN and SUNet.

This suggests that GCSTormer can complete computational tasks more quickly, reduce inference time, and improve operational efficiency.

Table 5: The comparision of parameters and complexity . The experiment is conducted on 256×256 clolor images.

Metrics \ Models	DnCNN	IRCNN	FFDNet	U-Net	DHDN	RDUNet	SUNet	**GCSTormer**
Param.(M)	0.6	0.4	0.8	17	168	166	99	52
Flops.(G)	36	27	18	40	1019	807	30	32
PSNR(dB)	31.17	31.12	31.27	30.55	31.98	31.97	31.82	31.91

In addition to analyzing the number of model parameters and computational complexity, we also evaluate the denoising time (i.e., inference time). Table 6 presents the results. We compare our model with state-of-the-art methods, including DnCNN, IRCNN, FFDNet, SwinIR, and Restormer. Compared to CNN-based denoisers such as DnCNN, GCSTormer has significantly more parameters. However, it also achieves notable improvements in denoising performance. When compared to SwinIR and Restormer, our model has slightly lower performance and a larger number of parameters. Nevertheless, it requires less inference time, consumes less GPU memory, and has lower computational complexity. As a whole, GCSTormer demonstrates its advantages over both CNN-based and Transformer-based denoising models.

Table 6: The comparision of the inference time of the models in the dataset Kodak24 with σ =30. We conduct on five metrics: the parameters(Param.(M)), the complexity(Flops.(G)), the usage of GPU memory when denoising image(GPU(G)), the time of denoising image(Time(s)), and the PSNR(dB). The experiment is conducted on a personal laptop equipped with a GPU of RTX4060.

Models \ Metrics	Param.(M)	Flops.(G)	GPU(G)	Time(s)	PSNR(dB)
DnCNN	0.60	36.00	1.4	4.68	31.17
IRCNN	0.40	27.00	1.1	3.32	31.12
FFDNet	0.80	18.00	1.1	3.15	31.27
SwinIR	0.88	59.03	4.1	105.17	31.99
Restormer	26.10	140.99	4.3	30.07	32.01
GCSTormer	51.97	31.67	2.0	19.30	31.91

4.4.2. Ablation Studies on Hyperparameters and Construction

To investigate the impact of our model's internal structure and hyperparameters on performance, we has conducted extensive experiments. The results are shown in Tab. 7 and 8. As Tab.7 shows, the mainly ablation studies on the construction focus on the contributions of SEFM, transposed attention (TA), and gate unit (GU). In Case 4, where SFEM, TA, and GU are all included, the model achieves a PSNR of 31.91 dB. When comparing with Case 1, which only has TA and GU enabled, we can observe that the presence of SFEM contributes to an improvement in the PSNR. Similarly, comparing case 2 and case 3 to the baseline cases without certain components, it's clear that both SFEM and GU play important roles in enhancing the model's denoising performance, especially the SFEM. AS to the inference time, in Case 1, where TA is present, the inference time is 16.73 seconds. In contrast, in Case 2, where TA is absent, the inference time increases to 22.52 seconds. This significant difference in inference time shows that the introduction of TA has a notable effect on reducing the model's denoising inference time. Even when comparing with other cases, the presence of TA is associated with relatively shorter inference times, suggesting

that it helps in making the model's denoising process more efficient.

Table 7: The ablation studies on the construction of the model in the dataset Kodak24 with σ =30. We focus on SEFM, tranposed attention(TA), gate unit(GU) to make the inner construction more clearly. The experiment is conducted on personal laptop with a GPU of RTX4060.

Case	SFEM	TA	GU	Param.(M)	Flop.(G)	GPU(G)	Time(s)	PSNR(dB)
1		√	√	51.74	16.83	1.8	16.73	31.71
2	√		√	51.71	31.51	2.2	22.52	31.88
3	√	√		51.27	30.02	1.9	17.25	31.79
4	√	√	√	51.97	31.67	2.0	19.30	31.91

The hyperparameters studied include the dilation rate(DR), the number of stacked GCSTBs(×NGs), the training epochs, and the activation function(AF), as Tab. 8 shows. In terms of DR, when the values are [1,2,3], the PSNR is 31.87 dB. As the values change to [1,2,5] and [1,3,5], the PSNR drops slightly to 31.84 dB and 31.80 dB respectively. This suggests that a larger dilation rate may cause discontinuous sampling of information, resulting in the model missing some key local information related to noise. Also, it can lead to sparser feature maps, which might make the model lose important details and increase the difficulty of learning appropriate feature combinations for denoising. Regarding the ×NGs, when it is set to 1, the PSNR gets the value of 31.66dB, that shows our designed block has a strong abliliy of extracting feature. As the number increase, the model's performance improves steadily. When it comes to the training epochs, the PSNR has already got a value of 31.68dB. This shows that the model's performance improves with the increase of epochs initially. And that's also the reason why we only conduct 100 epochs in other ablation experiments. As for the AF, when using ReLU, GELU, and PReLU, the differences in PSNR among these different activation functions are relatively small. Overall, the changes in PSNR values are relatively minor across different settings of these hyperparameters. This indicates that the model is not overly sensitive to changes in hyperparameters, which in turn reflects the strong

robustness of the model. It can maintain relatively stable performance under various hyperparameter configurations, making it more reliable and easier to optimize.

Table 8: The ablation studies on the hyperparameters in the dataset Kodak24 with σ =30. We focus on the dilation rate(DR), the number of stacked GCSTBs(×NGs), the training epochs, and the activation function(AF) to study the sensitivity of the hyperparameters. The experiment is conducted on personal laptop with a GPU RTX4060

Hyperparameter	Value	DR	×NGs	Epochs	AF	Param.(M)	Flop.(G)	GPU(G)	Time(s)	PSNR(dB)
Dilation Rate(DR)	[1,2,3]	-	×4	100	GELU	51.97	31.67	-	-	31.87
	[1,2,5]									31.84
	[1,3,5]									31.80
×N GCSTBs (×NGs)	×1	[1,2,3]	-	100	GELU	16.24	24.15	1.69	11.78	31.66
	×2					28.23	27.49	1.88	14.42	31.80
	×4					51.97	31.67	2.02	19.30	31.87
	×8					99.90	45.03	2.34	28.28	31.90
Epochs	50	[1,2,3]	×4	-	GELU	51.97	31.67	-	-	31.68
	100									31.87
	150									31.91
	200									31.91
Activiation Function(AF)	ReLU	[1,2,3]	×4	100	-	51.97	31.67	-	-	31.83
	GELU									31.87
	PReLU									31.85

5. Limitation and Future Work

This work is centered around the research on synthetic Gaussian noise. Its inception lies in the attempt to conduct image denoising research by integrating the advantages of CNN and Transformer. In the aspect of model design, we have devised a shallow network, where CNN plays a supplementary role, and a deep network mainly based on Transformer for feature extraction. This is aimed at achieving the integration of shallow and deep layers. To validate the effectiveness of this design, we have carried out numerous experiments. However, this design may not be the optimal solution. In the future, we plan to explore more combination strategies. For instance, we will study Heterogeneous dual-end denoising network based on distillation and parallel dual-encoder denoising networks. Additionally, while this work currently focuses on the theoretical study of Gaussian denoising, in the days to come, we will apply the model to

practical scenarios of image denoising, such as medical imaging, satellite image enhancement, or low-light photography. Moreover, we will also extend its application to other image restoration tasks, such as image deraining, deblurring, and super-resolution.

6. Conclusion

In this paper, we present GCSTormer for image denoising. Specifically, this model mainly consists of two core feature extraction modules: a shallow feature extraction moudule and a deep backbone feature extraction module. To address the CNN's limitations in extracting global information, we use atrous convolution to increase the receptive field and employ dense skip connection mechanism. In the deep feature extraction module, we combined the Swin Transformer with a U-Net structure for multi-scale global modeling of image features. Additionally, we introduced gate mechanism to further enhance the extraction of effective information and suppress the propagation of noise features. To further minimize the computating cost of the Transformer, we compute it from the channel dimension instead of the spatial dimension, adding the channel weights to the feature map. Experiments on the color image datasets CBSD68, Kodak24 , McMaster and gray image datasets set12 and BSD68 demonstrate the effectiveness of our model. Currently, the application of Transformers to low-level vision tasks is still in its early stages and there is significant room for improvement in terms of reducing computational complexity, parameter count and data dependency. Integrating the local information extraction capabilities of CNN into Transformer is a promising direction for improvement. In the future, we will study other combination methods and apply the model to other image restoration tasks and practical applications.

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.

Acknowledgements

This research was supported by the national nature science foundation of China 92354207, the Fundamental Research Funds for the Central Universities 2023RC09 , national social science foundation of China(BAA240034) and State Key Laboratory for Novel Software Technology, 348 Nanjing University under Grant KFKT2024B08. This research was funded by the Qiyuan Innovation Foundation (S20210201067) and sub-themes (No. 9072323404).

References

Agustsson, E., & Timofte, R. (2017). Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 126–135).

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European conference on computer vision* (pp. 213–229). Springer.

Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., & Gao, W. (2021). Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12299–12310).

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*, 2080–2095.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, .

Fan, C.-M., Liu, T.-J., & Liu, K.-H. (2022). Sunet: swin transformer unet for image denoising. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 2333–2337). IEEE.

Fan, L., Zhang, F., Fan, H., & Zhang, C. (2019). Brief review of image denoising techniques. *Visual Computing for Industry, Biomedicine, and Art*, *2*, 7.

Franzen, R. (1999). Kodak lossless true color image suite. *source: http://r0k. us/graphics/kodak*, *4*, 9.

Gou, Y., Hu, P., Lv, J., Zhou, J. T., & Peng, X. (2022). Multi-scale adaptive network for single image denoising. *Advances in Neural Information Processing Systems*, *35*, 14099–14112.

Gu, S., Zhang, L., Zuo, W., & Feng, X. (2014). Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2862–2869).

Gurrola-Ramos, J., Dalmau, O., & Alarcón, T. E. (2021). A residual dense u-net neural network for image denoising. *IEEE Access*, *9*, 31742–31754.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, .

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700–4708).

Huang, J.-J., & Dragotti, P. L. (2022). Winnet: Wavelet-inspired invertible network for image denoising. *IEEE Transactions on Image Processing*, *31*, 4377–4392.

Huang, K., Tian, C., Xu, Z., Li, N., & Lin, J. C.-W. (2023). Motion context guided edge-preserving network for video salient object detection. *Expert Systems with Applications*, *233*, 120739.

Ilesanmi, A. E., & Ilesanmi, T. O. (2021). Methods for image denoising using convolutional neural network: a review. *Complex & Intelligent Systems*, *7*, 2179–2198.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, .

Li, J., Cheng, B., Chen, Y., Gao, G., Shi, J., & Zeng, T. (2024a). Ewt: Efficient wavelet-transformer for single image denoising. *Neural Networks*, (p. 106378).

Li, J., Cheng, B., Chen, Y., Gao, G., Shi, J., & Zeng, T. (2024b). Ewt: Efficient wavelet-transformer for single image denoising. *Neural Networks*, (p. 106378).

Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., & Timofte, R. (2021). Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1833–1844).

Liu, P., Zhang, H., Zhang, K., Lin, L., & Zuo, W. (2018a). Multi-level wavelet-cnn for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 773–782).

Liu, P., Zhang, H., Zhang, K., Lin, L., & Zuo, W. (2018b). Multi-level wavelet-cnn for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 773–782).

Liu, X., Suganuma, M., Sun, Z., & Okatani, T. (2019). Dual residual networks leveraging the potential of paired operations for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7007–7016).

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10012–10022).

Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, .

Luthra, A., Sulakhe, H., Mittal, T., Iyer, A., & Yadav, S. (2021). Eformer: Edge enhancement based transformer for medical image denoising. *arXiv preprint arXiv:2109.08044*, .

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 *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001* (pp. 416–423). IEEE volume 2.

Mei, Y., Fan, Y., Zhang, Y., Yu, J., Zhou, Y., Liu, D., Fu, Y., Huang, T. S., & Shi, H. (2023). Pyramid attention network for image restoration. *International Journal of Computer Vision*, *131*, 3207–3225.

Park, B., Yu, S., & Jeong, J. (2019). Densely connected hierarchical network for image denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (pp. 0–0).

Tian, C., Fei, L., Zheng, W., Xu, Y., Zuo, W., & Lin, C.-W. (2020a). Deep learning on image denoising: An overview. *Neural Networks*, *131*, 251–275.

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

Tian, C., Xu, Y., Zuo, W., Du, B., Lin, C.-W., & Zhang, D. (2021a). Designing and training of a dual cnn for image denoising. *Knowledge-Based Systems*, *226*, 106949.

Tian, C., Xu, Y., Zuo, W., Lin, C.-W., & Zhang, D. (2021b). Asymmetric cnn for image superresolution. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *52*, 3718–3730.

Tian, C., Yuan, Y., Zhang, S., Lin, C.-W., Zuo, W., & Zhang, D. (2022a). Image super-resolution with an enhanced group convolutional neural network. *Neural Networks*, *153*, 373–385.

Tian, C., Yuan, Y., Zhang, S., Lin, C.-W., Zuo, W., & Zhang, D. (2022b). Image super-resolution with an enhanced group convolutional neural network. *Neural Networks*, *153*, 373–385.

Tian, C., Zhang, X., Lin, J. C.-W., Zuo, W., Zhang, Y., & Lin, C.-W. (2022c). Generative adversarial networks for image super-resolution: A survey. *arXiv preprint arXiv:2204.13620*, .

Tian, C., Zhang, Y., Zuo, W., Lin, C.-W., Zhang, D., & Yuan, Y. (2022d). A heterogeneous group cnn for image super-resolution. *IEEE transactions on neural networks and learning systems*, .

Tian, C., Zheng, M., Jiao, T., Zuo, W., Zhang, Y., & Lin, C.-W. (2024a). A self-supervised cnn for image watermark removal. *IEEE Transactions on Circuits and Systems for Video Technology*, .

Tian, C., Zheng, M., Li, B., Zhang, Y., Zhang, S., & Zhang, D. (2024b). Perceptive self-supervised learning network for noisy image watermark removal. *IEEE Transactions on Circuits and Systems for Video Technology*, .

Tian, C., Zheng, M., Zuo, W., Zhang, B., Zhang, Y., & Zhang, D. (2023). Multi-stage image denoising with the wavelet transform. *Pattern Recognition*, *134*, 109050.

Tian, C., Zheng, M., Zuo, W., Zhang, S., Zhang, Y., & Lin, C.-W. (2024c). A cross transformer for image denoising. *Information Fusion*, *102*, 102043.

Tian, C., Zheng, M., Zuo, W., Zhang, S., Zhang, Y., & Lin, C.-W. (2024d). A cross transformer for image denoising. *Information Fusion*, *102*, 102043.

Timofte, R., Agustsson, E., Van Gool, L., Yang, M.-H., & Zhang, L. (2017). Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 114–125).

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. In *International conference on machine learning* (pp. 10347–10357). PMLR.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Wang, Q., Li, Z., Zhang, S., Chi, N., & Dai, Q. (2024). A versatile wavelet-enhanced cnn-transformer for improved fluorescence microscopy image restoration. *Neural Networks*, *170*, 227–241.

Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J., & Li, H. (2022). Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 17683–17693).

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (2021). Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, *34*, 12077–12090.

Xu, K., Li, W., Wang, X., Hu, X., Yan, K., Wang, X., & Dong, X. (2023). Cur transformer: A convolutional unbiased regional transformer for image denoising. *ACM Transactions on Multimedia Computing, Communications and Applications*, *19*, 1–22.

Xue, T., & Ma, P. (2023). Tc-net: transformer combined with cnn for image denoising. *Applied Intelligence*, *53*, 6753–6762.

Yao, C., Jin, S., Liu, M., & Ban, X. (2022). Dense residual transformer for image denoising. *Electronics*, *11*, 418.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., & Huang, T. S. (2019). Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 4471–4480).

Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z.-H., Tay, F. E., Feng, J., & Yan, S. (2021). Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 558–567).

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5728–5739).

Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., Yang, M.-H., & Shao, L. (2021). Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 14821–14831).

Zhang, K., Li, Y., Zuo, W., Zhang, L., Van Gool, L., & Timofte, R. (2021). Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*, 6360–6376.

Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2017a). Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, *26*, 3142–3155.

Zhang, K., Zuo, W., Gu, S., & Zhang, L. (2017b). Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3929–3938).

Zhang, K., Zuo, W., & Zhang, L. (2018). Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, *27*, 4608–4622.

Zhang, L., Wu, X., Buades, A., & Li, X. (2011). Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *Journal of Electronic imaging*, *20*, 023016–023016.

Zhang, Q., Xiao, J., Tian, C., Xu, J., Zhang, S., & Lin, C.-W. (2023a). A parallel and serial denoising network. *Expert Systems with Applications*, *231*, 120628.

Zhang, Q., Xiao, J., Tian, C., Xu, J., Zhang, S., & Lin, C.-W. (2023b). A parallel and serial denoising network. *Expert Systems with Applications*, *231*, 120628.

Zhao, M., Cao, G., Huang, X., & Yang, L. (2022). Hybrid transformer-cnn for real image denoising. *IEEE Signal Processing Letters*, *29*, 1252–1256.

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P. H. et al. (2021). Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6881–6890).

Zhu, M., & Li, Z. (2023). Ngdcnet: Noise gating dynamic convolutional network for image denoising. *Electronics*, *12*, 5019.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., & Dai, J. (2020). Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, .