

XFORMER: HYBRID X-SHAPED TRANSFORMER FOR IMAGE DENOISING

Jiale Zhang¹, Yulun Zhang^{1*†}, Jinjin Gu^{2,3}, Jiahua Dong⁴, Linghe Kong^{1*}, Xiaokang Yang¹

¹Shanghai Jiao Tong University, ²Shanghai AI Laboratory, ³University of Sydney,

⁴Shenyang Institute of Automation, Chinese Academy of Sciences

ABSTRACT

In this paper, we present a hybrid X-shaped vision Transformer, named Xformer, which performs notably on image denoising tasks. We explore strengthening the global representation of tokens from different scopes. In detail, we adopt two types of Transformer blocks. The spatial-wise Transformer block performs fine-grained local patches interactions across tokens defined by spatial dimension. The channel-wise Transformer block performs direct global context interactions across tokens defined by channel dimension. Based on the concurrent network structure, we design two branches to conduct these two interaction fashions. Within each branch, we employ an encoder-decoder architecture to capture multi-scale features. Besides, we propose the Bidirectional Connection Unit (BCU) to couple the learned representations from these two branches while providing enhanced information fusion. The joint designs make our Xformer powerful to conduct global information modeling in both spatial and channel dimensions. Extensive experiments show that Xformer, under the comparable model complexity, achieves state-of-the-art performance on the synthetic and real-world image denoising tasks. We also provide code and models at <https://github.com/gladzhang/Xformer>.

1 INTRODUCTION

As a fundamental vision task, image denoising aims to recover the high-quality image from its noisy counterpart. It has been a very challenging problem as the denoising process is hard to distinguish the tiny textures and details from the noise. Recently, deep convolutional neural networks (CNNs) have shown great power to solve this inverse problem (Zhang et al., 2017a; 2020; Tian et al., 2020; Zhang et al., 2021a; Hu et al., 2021). With the help of convolution operations, deep features can be extracted to provide powerful image representations. However, the disadvantages of convolution are also obvious. Due to the poor receptive field scaling, CNN has limited ability to capture long-range dependencies among visual elements. Moreover, the convolution filters are parameter-dependent and content-independent and thus experience difficulty to show flexibility for the dynamic inputs. To address the above shortcomings, several recent works investigate the self-attention (SA) mechanism to replace the convolution and build the Transformer-based networks (Liang et al., 2021; Chen et al., 2021; Zamir et al., 2022; Wang et al., 2022; Lee et al., 2022).

Transformer has shown state-of-the-art performance on high-level vision tasks (Xu et al., 2021; Ali et al., 2021; Zhang & Yang, 2021; Liu et al., 2021; Chu et al., 2021; Wang et al., 2021; Xie et al., 2021). The SA mechanism has great power to capture content-dependent global representations while modeling long-distance relationships. Despite of the growing computational cost, researchers are investigating the employment of Transformer in solving low-level vision problems (Liang et al., 2021; Chen et al., 2021). Liang et al. (2021) proposed SwinIR based on Swin Transformer (Liu et al., 2021) to utilize spatial-wise window-based SA blocks. The tokens are extracted from a square location. Zamir et al. (2022) proposed Restormer to apply SA across channel dimension rather than the spatial dimension. It is demonstrated that the channel-wise SA is able to model global connectivity. For further analysis, these two types of SA mechanisms are considered to focus on different respects of global information modeling. In detail, the spatial-wise SA is good at capturing local patch-level information and modeling fine-detailed spatial features. On the other hand, the channel-wise SA is capable of capturing

*Corresponding authors: Yulun Zhang, yulun100@gmail.com; Linghe Kong, linghe.kong@sjtu.edu.cn.

†The work was mainly done when Yulun Zhang was at ETH Zürich.

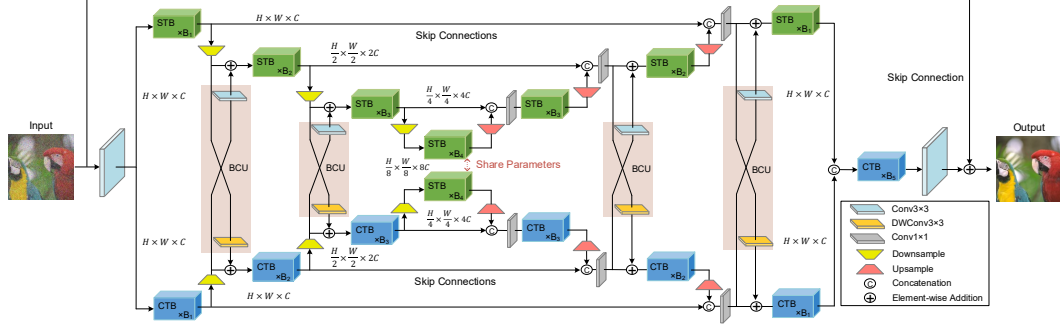


Figure 1: Architecture of our proposed Xformer. The modules include spatial-wise Transformer block (STB), channel-wise Transformer block (CTB), and bidirectional connection unit (BCU).

global channel-level information and modeling specific channel features. Especially, both types of information modeling are important for enhancing representation learning in Transformer.

Therefore, we explore adopting spatial-wise SA and channel-wise SA together in this paper. It remains challenging as there are gaps between these two types of SA mechanisms. We consider designing a concurrent network with dual branches. Similar parallel ideas have also been investigated for other visual tasks in recent years (Chen et al., 2022; Pan et al., 2022; Peng et al., 2021). In special, the concurrent structure is beneficial for the network to build direct interactions between different branches. Furthermore, we apply spatial-wise Transformer block and channel-wise Transformer block in respective branch. Following previous works (Yue et al., 2020; Cheng et al., 2021; Zamir et al., 2021; 2022), we employ an encoder-decoder structure within each branch to obtain multi-scale features. In short, the spatial-wise branch can perform fine-grained local patches interactions across spatial-dimension tokens. The channel-wise branch can perform direct global context interactions across channel-dimension tokens.

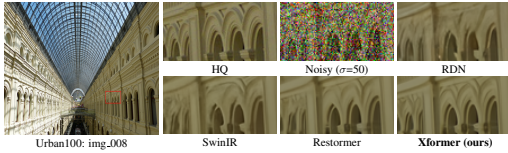


Figure 2: Visual examples for Gaussian color image denoising with noise level $\sigma=50$ on Urban100.

For further investigation, the concurrent network enables dual branches to model patch-level and channel-level information respectively. However, there are still some limitations. The direct concatenating operation in the end fails to effectively use these two types of features. In this way, each branch cannot capture information from different levels. Therefore, we propose the Bidirectional Connection Unit (BCU) as the bridge between two branches, which provides information fusion in an interactive manner. With BCU, the network can couple two styles of deep features. In detail, we utilize convolution layers with a 3×3 kernel to refine the learned deep features in corresponding branches. Then, we add the refined features to respective branches. Such a fusion operation can greatly enhance the global representation of tokens from different dimensions.

Based on the designs above, we present a hybrid X-shaped Transformer for image denoising, named Xformer, as shown in Fig. 1. We design a concurrent network with two branches. Specifically, we separately utilize spatial-wise SA blocks and channel-wise SA blocks in respective branch. Besides, we employ the proposed BCU to bridge these two branches for information fusion. The joint designs enable our network to obtain stronger global representations in Transformer. More details can be found in Sec. 3. Our Xformer can achieve superior results against recent leading image denoising methods. As shown in Fig. 2, our proposed method obtains visually pleasing results while others suffer from the loss of details. Overall, we summarize our main contributions as follows:

- We propose Xformer, an X-shaped Transformer with hybrid implementations of spatial-wise and channel-wise Transformer blocks, thereby exploiting the stronger global representation of tokens in Transformer-based neural network.
- We propose the Bidirectional Connection Unit (BCU) that is able to effectively couple the learned representations from two branches of Xformer. This simple design significantly enhances the global information modeling of our method.
- We employ Xformer to train an efficient and effective Transformer-based network for image denoising. We conduct extensive experiments on the synthetic and real-world noise removal tasks. Our method can achieve state-of-the-art performance.

2 RELATED WORK

Image Denoising. Due to the powerful generalizing ability from large-scale data, CNN-based methods (Zhang et al., 2020; Tian et al., 2020; Zhang et al., 2021a; Hu et al., 2021) have achieved superior performance over the traditional denoising algorithms (Perona & Malik, 1990; Mairal et al., 2009; Elad & Aharon, 2006). Zhang et al. (2017a) proposed DnCNN as a representative image denoising method, which trained mappings from noisy images to noises. For further improvements, many subsequent works used more elaborate network architecture designs, including encoder-decoder structure (Yue et al., 2020; Cheng et al., 2021; Zamir et al., 2021), non-local modules (Liu et al., 2018a; Zhang et al., 2019), attention mechanism (Zhang et al., 2021b), and dynamic convolution (Jiang et al., 2022). Unfortunately, most CNNs suffer from the limited ability to model long-range dependencies, which is crucial for recovering clean images. Very recently, researchers have started to utilize self-attention strategy to replace the single convolution operation. (Liang et al., 2021; Chen et al., 2021; Zamir et al., 2022; Wang et al., 2022; Lee et al., 2022).

Vision Transformer. In recent years, Transformer has achieved impressive success in machine translation tasks (Vaswani et al., 2017). It also performs outstandingly to solve numerous high-level vision problems (Zhang & Yang, 2021; Chu et al., 2021; Wang et al., 2021; Xie et al., 2021) due to the content-dependent global receptive field of the network. Dosovitskiy et al. (2021) firstly proposed ViT to introduce Transformer into image recognition. Inspired by these, more and more works started to apply Transformer to solve low-level vision tasks (Liang et al., 2021; Chen et al., 2021; Wang et al., 2022; Zamir et al., 2022). Wang et al. (2022) proposed a general U-shaped Transformer named Uformer based on U-Net (Ronneberger et al., 2015) for image restoration. Zamir et al. (2022) proposed a strong baseline model named Restormer and achieved state-of-the-art performance in several image restoration tasks. Chen et al. (2021) proposed IPT to apply standard Transformer blocks while using pre-training on additional datasets. To sum up, these works have not explored enhancing the global representation of tokens from different dimensions. In contrast, we design a general X-shaped Transformer to bridge this gap.

Concurrent Network. Compared to the commonly-used serial network, the concurrent network has parallel branches in the whole network architecture. Therefore, it has the natural advantage of simultaneously conducting different types of representation learning and building direct interactions between dual branches. Recently, few efforts have been made to explore this field. Some works only focused on parallel blocks such as Chen et al. (2022) and Pan et al. (2022). Peng et al. (2021) defined a representative concurrent network named Conformer to solve some high-level vision problems. They designed two branches to respectively leverage convolution operators and self-attention mechanisms. Specifically, the CNN branch preserves fine-detailed local features. The Transformer branch captures long-range dependencies. In this paper, we also design a concurrent network. Different from it, our proposed Xformer applies different Transformer blocks in two branches and we focus on low-level vision tasks. Besides, we propose the Bidirectional Connection Unit (BCU) to greatly enhance the information fusion within two branches.

3 METHOD

3.1 OVERALL PIPELINE

As shown in Fig. 1, our Xformer is a hybrid X-shaped Transformer-based network with two branches. Following the design of U-Net structures (Cheng et al., 2021; Zamir et al., 2022; Wang et al., 2022), each branch is treated as a separate U-shaped network with skip connections between encoders and decoders. We utilize the spatial-wise window-based Transformer blocks (STBs) to construct the spatial-wise branch. The channel-wise cross-covariance Transformer blocks (CTBs) are used to construct the channel-wise branch. Then, we design the Bidirectional Connection Unit (BCU) to bridge the dual branches for feature complementarity. It can bring information fusion for different branches. Besides, we provide two additional designs. Firstly, we make the last encoder involving STBs of two branches share parameters for the purpose of computational efficiency. Secondly, we concatenate the output features from two branches and send them to a new refinement module involving CTBs. The overall pipeline is as follows.

Given a degraded image $\mathbf{I} \in \mathbb{R}^{H \times W \times C_{in}}$ where H , W , and C_{in} are the height, width, and input channels, our proposed Xformer first uses a 3×3 convolutional layer (Conv) to obtain the shallow feature $\mathbf{F}_0 \in \mathbb{R}^{H \times W \times C}$, where C is the size of new feature dimension. Next, the feature \mathbf{F}_0 is

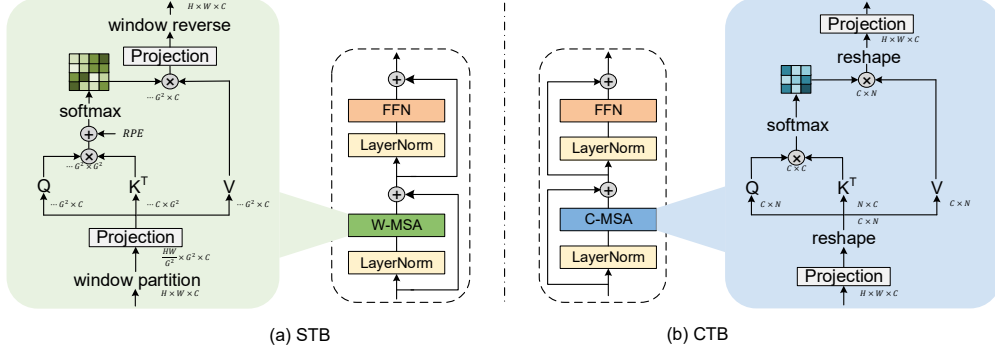


Figure 3: Module architectures of spatial-wise and channel-wise Transformer blocks (STB | CTB). concurrently sent to two 4-level symmetric encoder-decoder branches. Through these two branches, it is transformed into two new deep features $\mathbf{F}_s, \mathbf{F}_c \in \mathbb{R}^{H \times W \times C}$. In detail, each encoder or decoder contains cascaded Transformer blocks. Encoders take \mathbf{F}_0 as input and reduce half of the spatial resolution while doubling the number of feature channels as the stage grows. Decoders take the low-resolution features as input and reduce half of the feature channels while doubling the size of feature maps. By counting, input features experience three times of downsampling and upsampling. Besides, the up-sampled features are concatenated with the corresponding features from encoders via skip connections for the recovery improvement. The concatenated features further pass through a 1×1 Conv to reduce channels by half. For information fusion, the down-sampled features after the first and second encoders pass through the BCU and then integrate to the opposite branch through element-wise add. Similarly, the reduced features after the first and second decoders pass through the BCU and are added to the opposite branch. In the end, the new features \mathbf{F}_s and \mathbf{F}_c are concatenated and then flow to the refinement module. The output is further transmitted into a 3×3 Conv to obtain a residual image $\mathbf{I}_r \in \mathbb{R}^{H \times W \times C_{in}}$. Finally, the restored image is generated by $\hat{\mathbf{I}} = \mathbf{I} + \mathbf{I}_r$.

3.2 DUAL BRANCHES

Spatial-wise Branch. As shown in the upper part of Fig. 1, the spatial-wise branch adopts the encoder-decoder structure with skip connections. The components of encoders and decoders are cascaded spatial-wise window-based Transformer blocks (STBs). We provide the details of STB in Fig. 3(a). We make \mathbf{x}_l as the output at the l_{th} block. We formulate the calculation process of STB as

$$\begin{aligned} \mathbf{x}'_l &= \text{W-MSA}(\text{LN}(\mathbf{x}_{l-1})) + \mathbf{x}_{l-1}, \\ \mathbf{x}_l &= \text{FFN}(\text{LN}(\mathbf{x}'_l)) + \mathbf{x}'_l, \end{aligned} \quad (1)$$

where W-MSA means the window-based multi-head self-attention. Here we assume that the number of heads is 1 to transfer MSA to single-head mode. Given the feature $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ generated by layer normalization (LN) (Ba et al., 2016), W-MSA first split it into non-overlapping $G \times G$ windows to get features $\mathbf{X}^i \in \mathbb{R}^{G^2 \times C}$ for i_{th} window. Next, it performs linear projecting to generate query (\mathbf{Q}^i), key (\mathbf{K}^i), and value (\mathbf{V}^i), yielding $\mathbf{Q}^i = \mathbf{X}^i \mathbf{W}^Q$, $\mathbf{K}^i = \mathbf{X}^i \mathbf{W}^K$, and $\mathbf{V}^i = \mathbf{X}^i \mathbf{W}^V$, where $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{C \times C}$ are learnable parameters. We formulate the calculation in i_{th} window as

$$\hat{\mathbf{X}}^i = \text{Softmax}\left(\frac{\mathbf{Q}^i \mathbf{K}^{iT}}{\sqrt{C}} + \mathbf{B}\right) \mathbf{V}^i, \quad (2)$$

where $\hat{\mathbf{X}}^i$ is the output feature map in i_{th} window and \mathbf{B} is the relative position encoding (RPE) (Shaw et al., 2018; Raffel et al., 2020). Furthermore, the features from all windows are projected together and reshaped to the new feature map of size $\mathbb{R}^{H \times W \times C}$, as the last output of W-SMA. For the Feed-forward network (FFN), we use the basic multi-layer perception (MLP) used in recent works (Liu et al., 2021; Liang et al., 2021) to deal with the input features. In short, the STB utilizes non-overlapping windows to generate shorter token sequences for the self-attention computation, which can enable the network to obtain fine-grained local patches interactions.

Channel-wise Branch. Similarly, this channel-wise branch contains a 4-level encoder-decoder structure. In special, the encoders and decoders are constructed by cascaded channel-wise Transformer blocks (CTBs). The details of CTB are shown in Fig. 3(b). Assuming that \mathbf{x}_k is the output at the k_{th} block, the calculation process can be formulated as

$$\begin{aligned} \mathbf{x}'_k &= \text{C-MSA}(\text{LN}(\mathbf{x}_{k-1})) + \mathbf{x}_{k-1}, \\ \mathbf{x}_k &= \text{FFN}(\text{LN}(\mathbf{x}'_k)) + \mathbf{x}'_k, \end{aligned} \quad (3)$$

where C-MSA means the channel-wise multi-head self-attention. We also assume that the number of heads is 1 and transfer MSA to a single-head fashion. Given the normalized feature $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$, C-MSA first utilizes the projecting module to get prepared query, key and value. In order to introduce contextualized information into self-attention computation, we choose to use 3×3 depth-wise convolution (Conv) following 1×1 Conv to generate query (\mathbf{Q}), key (\mathbf{K}), and value (\mathbf{V}). We make $\mathbf{Q} = W_d^Q W_p^Q X$, $\mathbf{K} = W_d^K W_p^K X$, and $\mathbf{V} = W_d^V W_p^V X$, where $W_p^{(\cdot)}$ means parameters of 1×1 point-wise Conv and $W_d^{(\cdot)}$ means parameters of 3×3 depth-wise Conv. Then, the obtained \mathbf{Q} , \mathbf{K} , \mathbf{V} are reshaped into new feature maps of size $\mathbb{R}^{C \times N}$, where $N = H \times W$. The query and key are further normalized to prepare for cross-covariance attention. The new transposed attention map is calculated by \mathbf{Q} and \mathbf{K}^T with size of $\mathbb{R}^{C \times C}$. The calculation process of the C-MSA is formulated as

$$\hat{\mathbf{X}} = \text{Softmax}(\mathbf{Q}\mathbf{K}^T / \tau)\mathbf{V}, \quad (4)$$

where τ is a learnable temperature parameter and $\hat{\mathbf{X}} \in \mathbb{R}^{C \times N}$ is the output. Then $\hat{\mathbf{X}}$ is reshaped to the original feature size of $\mathbb{R}^{H \times W \times C}$. The output further passes through a linear projecting layer. Added by the shortcut \mathbf{x}_{k-1} , new features \mathbf{x}'_k is transmitted to the following part. For the FFN, we introduce the gating mechanism and depth-wise convolutions proposed in the recent work (Zamir et al., 2022) to enrich the feature transferring, which is validated to be effective. Equipped with the used C-MSA and FFN, the CTB enjoys strong ability to capture direct global context interactions.

3.3 BIDIRECTIONAL CONNECTION UNIT

Motivation. As the dual branches enable the network to capture both patch-level and channel-level information, the information fusion is treated as an important step to enhance global information modeling. Simple concatenating operation is not able to effectively utilize information from different branches. The direct connection of dual branches is not the best choice. Therefore, we propose the Bidirectional Connection Unit (BCU) to couple the deep features from their respective feed-forward processes for feature complementarity. We demonstrate that the proposed BCU plays an important role to provide enhanced information fusion.

Specific Design. We design the BCU to bridge the two branches in an interactive manner. We carry out the specific feature complementarity like the form of absolute position encoding (Vaswani et al., 2017). On one hand, we add the global context information brought by channel-wise self-attention to the feature maps of the spatial-wise branch. On the other hand, we add the fine-grained patch-level information brought by local patches interactions to the feature maps of the channel-wise branch. In detail, the BCU contains two simple convolution layers. Specifically, we use a 3×3 depth-wise convolution layer to refine the deep features from the spatial-wise branch for the purpose of saving computational consumption. We use a common 3×3 convolution layer to refine features from the channel-wise branch to provide more channel-dimension interactions. With the 3×3 kernel size, the feature refinement process can provide extra contextualized information. The specific implementation is shown in Fig. 1.

3.4 IMPLEMENTATION DETAILS

Specific Settings. Firstly, we set the layer numbers of both branches the same, which are [2, 4, 4, 6, 4, 4, 2]. The number of CTBs in the refinement stage is set to 4. Secondly, we set the number of heads in corresponding layers to [1, 2, 4, 8, 4, 2, 1]. The head number of CTBs in the refinement stage is set to 1. Meanwhile, the channels number of shallow features generated by the first convolution layer is set to 48. The expansion size of hidden layers in FFN is set to 2.66. Thirdly, the window size in spatial-wise Transformer blocks is set to 16. Note that we also utilize the shifted-window strategy (Liu et al., 2021). Besides, we use pixel-unshuffle and pixel-shuffle operations (Shi et al., 2016) for downsampling and upsampling. Lastly, following the recent work (Zamir et al., 2022), we use the progressive training strategy for fair comparisons.

Loss Function. Following most recent works (Lai et al., 2017; Zhang et al., 2020; Zamir et al., 2022), we use L_1 loss function to optimize our proposed Xformer. For image denoising, the goal of training Xformer is to minimize the L_1 loss, which is formulated as

$$\mathcal{L} = \|\hat{I}_{HQ} - I_{HQ}\|_1, \quad (5)$$

where \hat{I}_{HQ} is the output of our Xformer and I_{HQ} is the corresponding ground-truth image.

Method	All STB	All CTB	STB+CTB	Method	w/o BCU	BCU-1	BCU-2	Complete BCU	Method	w/o Shift	w/ Shift
Params (M)	26.03	28.81	25.23	Params (M)	24.70	24.71	25.22	25.23	Params (M)	25.23	25.23
FLOPs (G)	38.1	42.3	42.2	FLOPs (G)	40.9	40.9	42.2	42.2	FLOPs (G)	42.2	42.2
PSNR (dB)	29.87	29.67	29.94	PSNR (dB)	29.82	29.84	29.92	29.94	PSNR (dB)	29.88	29.94
SSIM	0.8851	0.8830	0.8865	SSIM	0.8842	0.8848	0.8859	0.8865	SSIM	0.8852	0.8865

(a) Ablation study of block setting.

(b) Ablation study of BCU settings.

(c) Whether to use shift.

ID	STB	CTB	BCU	Structure	Params (M)	FLOPs (G)	PSNR (dB)	SSIM	Method	SwinIR	Restormer	Xformer (ours)
1	✓			single-branch	26.48	40.6	29.84	0.8853	Params (M)	11.50	26.11	25.23
2		✓		single-branch	26.11	38.7	29.68	0.8829	FLOPs (G)	201.2	38.7	42.2
3	✓	✓		two-branches	24.70	40.9	29.82	0.8842	McMaster	30.22	30.30	30.38
4	✓	✓	✓	two-branches	25.23	42.2	29.94	0.8865	Urban100	29.82	30.02	30.36

(d) Ablation study of designed models with different branches.

(e) Params-FLOPs-PSNR comparisons.

Table 1: Ablation experiments (a-d) and model complexity comparisons (e). For ablation, we train models on Gaussian color image denoising task with $\sigma=50$ for 100k iterations and test on Urban100.

Figure 4: Visualization of feature maps from the last STB and CTB in the encoder-decoder module of dual branches. We compare different situations about whether to use BCU.

4 EXPERIMENTAL RESULTS

4.1 EXPERIMENTAL SETTINGS

Data and Evaluation. We conduct experiments on two denoising tasks, including synthetic image denoising using noisy images generated with additive white Gaussian noise and real image denoising using real-world noisy images. For Gaussian denoising, following the previous work (Liang et al., 2021), we use DIV2K (Timofte et al., 2017), Flickr2K (Lim et al., 2017), BSD500 (Arbelaez et al., 2010), and WED (Ma et al., 2016) as training data. Set12 (Zhang et al., 2017a), BSD68 (Martin et al., 2001), Kodak24 (Franzen, 1999), McMaster (Zhang et al., 2011), and Urban100 (Huang et al., 2015) are the testing data. For real image denoising, same with Restormer (Zamir et al., 2022), we use SIDD (Abdelhamed et al., 2018) to train our model. The evaluation is performed on 1,280 patches of the SIDD validation set (Abdelhamed et al., 2018) and 50 pairs of images from the DND (Plotz & Roth, 2017). Note that we evaluate the performance with the commonly-used PSNR and SSIM (Wang et al., 2004) metrics. Besides, we also provide comparisons of FLOPs and model size. We set the input image size to $3 \times 128 \times 128$ when calculating FLOPs.

Training Settings. We perform data augmentation on the training data through random horizontal or vertical flips and rotation of 90° , 180° , and 270° . Using progressive training strategy proposed by Restormer (Zamir et al., 2022), we set the batch size and patch size pairs to $[(64, 128^2), (40, 160^2), (32, 192^2), (16, 256^2), (8, 320^2), (8, 384^2)]$ at training iterations [0k, 92k, 156k, 204k, 240k, 276k]. AdamW (Loshchilov & Hutter, 2019) is used to optimize our model with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay 10^{-4} . We train our model for total 300k iterations and the initial learning rate is set to 3×10^{-4} and gradually reduced to 10^{-6} through the cosine annealing (Loshchilov & Hutter, 2017). Our Xformer is implemented on PyTorch (Paszke et al., 2017) using 4 Nvidia A100 GPUs.

4.2 ABLATION STUDY

For ablation experiments, we train all the models on Gaussian color image denoising task with noise level $\sigma=50$. We train these models for 100k iterations. The evaluations are performed on Urban100 dataset. We also report the model size and FLOPs. The results are shown in Tab. 1.

Impact of STB and CTB. We design the ablation study to support using both STB and CTB.

Ablation design. As shown in Tab. 1a, we present three different network designs. Specifically, using all STB or CTB means that we replace all the Transformer blocks in Xformer with STB or CTB. For fair comparisons, we design these models with comparable complexity.

Analyses and conclusion. As we can see, the model using all CTB gets poor performance since it pays less attention to patch-level information. Furthermore, the model using all STB obtains suboptimal performance. However, using STB and CTB together in dual branches can achieve the best performance gains. The joint application of hybrid Transformer blocks can simultaneously obtain patch-level and channel-level information, which is very important.

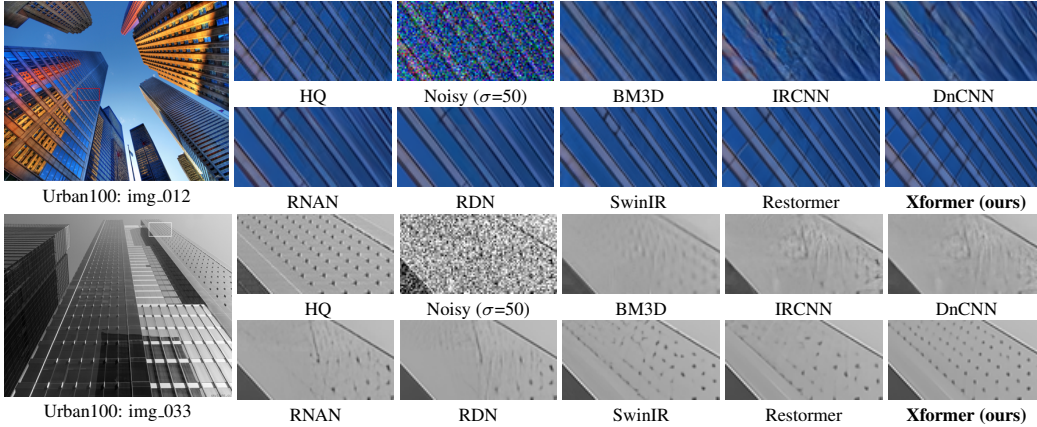


Figure 5: Visual comparisons on Gaussian color and gray image denoising ($\sigma=50$).

Impact of BCU. We further discuss the impact of the BCU by the ablation study.

Ablation design. We set four different experimental conditions. As shown in Tab. 1b, we compare the results of models without BCU, using single-direction BCU, and using complete BCU. Note that using single-direction BCU means that we only use the DWConv or the Conv to provide the information transmission from a single direction. Furthermore, BCU-1 denotes the model using DWConv and BCU-2 denotes the model using Conv.

Analyze the importance of using BCU. As we can see, the model with complete BCU achieves PSNR gain of 0.12 dB over that without BCU, which indicates that the BCU is an important component.

Analyze the necessity of the interactive manner. The models using single-direction BCU have sub-optimal performance. Moreover, we find that the model with BCU-2 yields 0.08 dB gain over that with BCU-1. It reveals that the information flow from the channel-wise branch has a bigger impact.

Visual analyses and conclusion. We provide visual results in Fig. 4. We visualize the deep features from the last STB or CTB in corresponding branches. We can see that the STB and CTB in the network with BCU can capture more extra information and thus show better visual results. We conclude that our proposed BCU can provide effective information fusion for the dual branches. With BCU, our proposed network can achieve promising performance.

Impact of Different Branches. We also discuss the importance of different branches.

Ablation design. We design the networks with comparable model size and FLOPs. As shown in Tab. 1d, we present four different networks, including using STB-based branch, using CTB-based branch, using dual branches without BCU, and using dual branches with BCU. The model using single branch is degraded to a complete U-shaped network like the architecture of Restormer.

Analyze the results of using single branch. The model using STB-based branch performs better than that using CTB-based one. It demonstrates that the patch-level information deserves more attention.

Analyze the results of using dual branches without BCU. We find that the model using dual branches without BCU gets limited performance. It confirms the statement we discuss in Sec. 3.3. The simple concatenating operation fails to effectively utilize the obtained information. Thus, the direct connection of dual branches even brings unsatisfied results.

Analyze the results of using dual branches with BCU. As we can see, with BCU, our model using dual branches can achieve greatly enhanced performance. It further validates that the BCU plays an important role in fusing patch-level and channel-level information in our concurrent network.

Conclusion. Thanks to these two joint designs, our proposed method can explore enhancing global information modeling in Transformer and thus outperform previous promising methods.

Impact of Shift. Table 1c shows the comparisons about whether to use shift. As we can see, the model using shift operation obtains better performance gains. It is because shift operation can bring more global receptive fields for window-based self-attention. Therefore, we choose to use shift.

4.3 GAUSSIAN IMAGE DENOISING RESULTS

We provide the comparisons of our Xformer with recent representative image denoising methods on both Gaussian color and grayscale image denoising tasks. As shown in Tabs. 2 and Tab. 3,

Dataset	σ	BM3D	DnCNN	IRCNN	FFDNet	NLRN	MWCNN	RNAN	RDN	DRUNet	SwinIR	Restormer	Xformer (ours)
Set12	15	32.37	32.86	32.76	32.75	33.16	33.15	-	-	33.25	33.36	<u>33.42</u>	33.46
	25	29.97	30.44	30.37	30.43	30.80	30.79	-	-	30.94	31.01	<u>31.08</u>	31.16
	50	26.72	27.18	27.12	27.32	27.64	27.74	27.70	27.60	27.90	27.91	<u>28.00</u>	28.10
BSD68	15	31.08	31.73	31.63	31.63	31.88	31.86	-	-	31.91	<u>31.97</u>	31.96	31.98
	25	28.57	29.23	29.15	29.19	29.41	29.41	-	-	29.48	29.50	<u>29.52</u>	29.55
	50	25.60	26.23	26.19	26.29	26.47	26.53	26.48	26.41	26.59	26.58	<u>26.62</u>	26.65
Urban100	15	32.35	32.64	32.46	32.40	33.45	33.17	-	-	33.44	33.70	<u>33.79</u>	33.98
	25	29.70	29.95	29.80	29.90	30.94	30.66	-	-	31.11	31.30	<u>31.46</u>	31.78
	50	25.95	26.26	26.22	26.50	27.49	27.42	27.65	27.40	27.96	27.98	<u>28.29</u>	28.71

Table 2: PSNR (dB) comparisons for **Gaussian grayscale image denoising** on three benchmark datasets. The underlined and **bold** numbers indicate the second best and the best results.

Dataset	σ	BM3D	DnCNN	IRCNN	FFDNet	RNAN	RDN	DRUNet	P3AN	IPT	SwinIR	Restormer	Xformer (ours)
CBSD68	15	33.52	33.90	33.86	33.87	-	-	34.30	-	-	<u>34.42</u>	34.40	34.43
	25	30.71	31.24	31.16	31.21	-	-	31.69	-	-	31.78	<u>31.79</u>	31.82
	50	27.38	27.95	27.86	27.96	28.27	28.31	28.51	28.37	28.39	28.56	<u>28.60</u>	28.63
Kodak24	15	34.28	34.60	34.69	34.63	-	-	35.31	-	-	35.34	* <u>35.35</u>	35.39
	25	32.15	32.14	32.18	32.13	-	-	32.89	-	-	32.89	* <u>32.93</u>	32.99
	50	28.46	28.95	28.93	28.98	29.58	29.66	29.86	29.69	29.64	29.79	* <u>29.87</u>	29.94
McMaster	15	34.06	33.45	34.58	34.66	-	-	35.40	-	-	35.61	<u>35.61</u>	35.68
	25	31.66	31.52	32.18	32.35	-	-	33.14	-	-	33.20	<u>33.34</u>	33.44
	50	28.51	28.62	28.91	29.18	29.72	-	30.08	-	29.98	30.22	<u>30.30</u>	30.38
Urban100	15	33.93	32.98	33.78	33.83	-	-	34.81	-	-	35.13	<u>35.13</u>	35.29
	25	31.36	30.81	31.20	31.40	-	-	32.60	-	-	32.90	<u>32.96</u>	33.21
	50	27.93	27.59	27.70	28.05	29.08	29.38	29.61	29.51	29.71	29.82	<u>30.02</u>	30.36

Table 3: PSNR (dB) comparisons for **Gaussian color image denoising** on four benchmark datasets. The underlined and **bold** numbers indicate the second best and the best results. * denotes results which are obtained by testing with officially provided pre-trained models.

BM3D (Dabov et al., 2007) is the classical denoising method. The CNN-based methods include DnCNN (Zhang et al., 2017a), IRCNN (Zhang et al., 2017b), FFDNet (Zhang et al., 2018), NLRN (Liu et al., 2018a), MWCNN (Liu et al., 2018b), RNAN (Zhang et al., 2019), RDN (Zhang et al., 2020), DRUNet (Zhang et al., 2021a), and P3AN (Hu et al., 2021). The Transformer-based methods include SwinIR (Liang et al., 2021), IPT (Chen et al., 2021), and Restormer (Zamir et al., 2022). All the results are obtained by open available data. We get some results by using officially provided pre-trained models. Following most recent works (Liang et al., 2021; Zamir et al., 2022), we set the additional noise level to 15, 25, and 50. We also provide the visual comparisons in Fig. 5. The results of model parameters and FLOPs comparisons are shown in Tab. 1e.

Quantitative Comparisons. We present the PSNR results of all compared approaches in Tab. 2 and Tab. 3. The corresponding scores are obtained by testing on several benchmark datasets for Gaussian grayscale and color image denoising. As we can see, our Xformer achieves the best PSNR performance across all evaluation datasets. Specifically, for the evaluation on high-resolution Urban100 dataset (Huang et al., 2015) under the challenging noise level 50, Xformer obtains 0.42 dB performance gain over the previous best Transformer-based network Restormer (Zamir et al., 2022), as shown in Tab. 2. Similarly, Table 3 also shows that the best performance is achieved by Xformer for Gaussian color image denoising. Compared to the results of SwinIR (Liang et al., 2021), Xformer has better performance gains while maintaining $4.76\times$ fewer FLOPs. It is also worth mentioning that our proposed method has the comparable model size and FLOPs with Restormer. In short, the experimental results demonstrate that our proposed Xformer becomes a new promising Transformer-based network for the Gaussian image denoising.

Visual Comparisons. The visual comparisons for Gaussian color and grayscale image denoising on some challenging examples are shown in Figs. 5. The noise level is set to 50 and results are obtained by testing on Urban100. We can see that our Xformer is able to remove heavy noise corruption for color image denoising. Compared to some previous denoising methods, our method obtains visually pleasing results. Besides, for the gray image denoising, the detailed textures and high-frequency components of the original images are reserved by using our method. However, others suffer from the heavy blurring and missing details. It demonstrates that our Xformer performs excellently for both color and grayscale image denoising for visual results.

Dataset	Method	BM3D	DnCNN	CBDNet	RIDNet	AINDNet	VDN	SADNet	DANet	CycleISP	MIRNet	DeamNet	DAGL	MAXIM	Uformer	Restormer	Xformer
SIDD	PSNR	25.65	23.66	30.78	38.71	39.08	39.28	39.46	39.47	39.52	39.72	39.47	38.94	39.96	39.89	40.02	<u>39.98</u>
	SSIM	0.685	0.583	0.801	0.951	0.954	0.956	0.957	0.957	0.957	0.959	0.957	0.953	0.960	0.960	<u>0.960</u>	0.960
DND	PSNR	34.51	32.43	38.06	39.26	39.37	39.38	39.59	39.58	39.56	39.88	39.63	39.77	39.84	<u>40.04</u>	40.03	40.19
	SSIM	0.851	0.790	0.942	0.953	0.951	0.952	0.952	0.955	0.956	0.956	0.953	0.956	0.954	0.956	<u>0.956</u>	0.957

Table 4: PSNR (dB) and SSIM comparisons for **real image denoising** on two benchmark datasets. The underlined and **bold** numbers indicate the second best and the best results.

Model Size Comparisons. Table 1e provides comparisons of parameters number and FLOPs with existing state-of-the-art methods. We calculate the FLOPs assuming that the input size is $3 \times 128 \times 128$. The PSNR scores are reported on the popular benchmark datasets McMaster (Zhang et al., 2011) and Urban100 (Huang et al., 2015) under Gaussian color image denoising with noise level $\sigma=50$. We mainly compare our proposed method to recent Transformer-based networks, including SwinIR and Restormer. We find that our Xformer enjoys very low FLOPs when compared to SwinIR. Meanwhile, it is seen that our method has comparable model size and FLOPs with Restormer. However, our Xformer can achieve the best performance among them. In special, it obtains 0.34 dB higher PSNR score over Restormer on Urban100. It indicates that our method has an acceptable computational and memory cost while performing promising image denoising.

4.4 REAL IMAGE DENOISING RESULTS

As shown in Tab. 4, We show the quantitative comparisons of our Xformer with other state-of-the-art methods on real-world image denoising task. In detail, we report the evaluation results from the classic denoising method BM3D (Dabov et al., 2007), CNN-based methods DnCNN (Zhang et al., 2017a), CBDNet (Guo et al., 2019), RIDNet (Anwar & Barnes, 2019), AINDNet (Kim et al., 2020), VDN (Yue et al., 2019), SADNet (Chang et al., 2020), DANet (Yue et al., 2020), CycleISP (Zamir et al., 2020a), MIRNet (Zamir et al., 2020b), DeamNet (Ren et al., 2021), DAGL (Mou et al., 2021), MAXIM (Tu et al., 2022), and Transformer-based methods Uformer (Wang et al., 2022) and Restormer (Zamir et al., 2022). Following the recent work (Zamir et al., 2022), we only use SIDD (Abdelhamed et al., 2018) dataset to train our models. Then, the trained models are directly used to perform evaluations on the DND (Plotz & Roth, 2017) benchmark. As DND does not provide ground-truth labels, the corresponding results are obtained by uploading images to the online server. Note that all the results are obtained from the open-source data.

Quantitative Comparisons. Table 4 shows the PSNR and SSIM scores of recent approaches on real-world image denoising. As we can see, our proposed Xformer outperforms all the state-of-the-art methods on the DND dataset (Plotz & Roth, 2017) and achieves comparable performance on the SIDD dataset (Abdelhamed et al., 2018). Compared to all the CNN-based methods, our Xformer has the obvious performance improvement. Furthermore, compared to Restormer (Zamir et al., 2022), our method performs better with comparable mode complexity. Besides, our Xformer achieves higher performance gains than Uformer (Wang et al., 2022) while maintaining $2.01 \times$ fewer model parameters. In special, Restormer only utilized channel-wise self-attention and paid less attention to local patches interactions. On the contrary, Uformer focused on spatial-wise self-attention and neglected the channel-dimension token interactions. In contrast, our proposed Xformer explore building interactions among tokens from both spatial and channel dimensions. Through fusing both patch-level and channel-level information, our method is able to obtain better performance.

5 CONCLUSION

In this work, we propose a hybrid X-shaped vision Transformer, named Xformer, for image denoising task. We design a concurrent network structure to utilize spatial-wise window-based Transformer blocks and channel-wise Transformer blocks respectively in two branches. Our proposed Xformer can enable each branch to proceed representation learning from the corresponding dimension, spatial or channel. Besides, we propose the Bidirectional Connection Unit (BCU) to bridge the separate branches. Specifically, the BCU provides information fusion in an interactive manner and greatly enhances the global information modeling ability of both branches. We conduct extensive experiments on the synthetic and real-world image denoising tasks. Experimental results demonstrate that our Xformer can outperform recent state-of-the-art methods both quantitatively and visually with comparable model size and computational cost.

ACKNOWLEDGMENTS

This work was supported in part by NSFC grant (62141220, U19B2035) and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102).

REFERENCES

- Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018.
- Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. In *NeurIPS*, 2021.
- Saeed Anwar and Nick Barnes. Real image denoising with feature attention. In *ICCV*, 2019.
- Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 2010.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- Meng Chang, Qi Li, Huajun Feng, and Zhihai Xu. Spatial-adaptive network for single image denoising. In *ECCV*, 2020.
- Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chun-jing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *CVPR*, 2021.
- Qiang Chen, Qiman Wu, Jian Wang, Qinghao Hu, Tao Hu, Errui Ding, Jian Cheng, and Jingdong Wang. Mixformer: Mixing features across windows and dimensions. In *CVPR*, 2022.
- Shen Cheng, Yuzhi Wang, Haibin Huang, Donghao Liu, Haoqiang Fan, and Shuaicheng Liu. Nbnnet: Noise basis learning for image denoising with subspace projection. In *CVPR*, 2021.
- Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. In *NeurIPS*, 2021.
- Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *TIP*, 2007.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.
- Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *TIP*, 2006.
- Rich Franzen. Kodak lossless true color image suite. *source: <http://r0k.us/graphics/kodak>*, 1999.
- Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *CVPR*, 2019.
- Xiaowan Hu, Ruijun Ma, Zhihong Liu, Yuanhao Cai, Xiaole Zhao, Yulun Zhang, and Haoqian Wang. Pseudo 3d auto-correlation network for real image denoising. In *CVPR*, 2021.
- Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, 2015.
- Yifan Jiang, Bart Wronski, Ben Mildenhall, Jon Barron, Zhangyang Wang, and Tianfan Xue. Fast and high-quality image denoising via malleable convolutions. In *ECCV*, 2022.
- Yoonsik Kim, Jae Woong Soh, Gu Yong Park, and Nam Ik Cho. Transfer learning from synthetic to real-noise denoising with adaptive instance normalization. In *CVPR*, 2020.

- Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *CVPR*, 2017.
- Hunsang Lee, Hyesong Choi, Kwanghoon Sohn, and Dongbo Min. Knn local attention for image restoration. In *CVPR*, 2022.
- Jingyun Liang, Jiezhong Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *ICCV*, 2021.
- Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, 2017.
- Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. In *NeurIPS*, 2018a.
- Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level wavelet-cnn for image restoration. In *CVPRW*, 2018b.
- Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.
- Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.
- Kede Ma, Zhengfang Duanmu, Qingbo Wu, Zhou Wang, Hongwei Yong, Hongliang Li, and Lei Zhang. Waterloo exploration database: New challenges for image quality assessment models. *TIP*, 2016.
- Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *ICCV*, 2009.
- David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001.
- Chong Mou, Jian Zhang, and Zhuoyuan Wu. Dynamic attentive graph learning for image restoration. In *ICCV*, 2021.
- Xuran Pan, Chunjiang Ge, Rui Lu, Shiji Song, Guanfu Chen, Zeyi Huang, and Gao Huang. On the integration of self-attention and convolution. In *CVPR*, 2022.
- Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.
- Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling global representations for visual recognition. In *ICCV*, 2021.
- Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *TPAMI*, 1990.
- Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 2020.
- Chao Ren, Xiaohai He, Chuncheng Wang, and Zhibo Zhao. Adaptive consistency prior based deep network for image denoising. In *CVPR*, 2021.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

- Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.
- Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016.
- Chunwei Tian, Yong Xu, and Wangmeng Zuo. Image denoising using deep cnn with batch renormalization. *Neural Networks*, 2020.
- Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, Lei Zhang, Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, Kyoung Mu Lee, et al. Ntire 2017 challenge on single image super-resolution: Methods and results. In *CVPRW*, 2017.
- Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxim: Multi-axis mlp for image processing. In *CVPR*, 2022.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
- Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, 2021.
- Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *CVPR*, 2022.
- Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004.
- Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021.
- Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. In *ICCV*, 2021.
- Zongsheng Yue, Hongwei Yong, Qian Zhao, Deyu Meng, and Lei Zhang. Variational denoising network: Toward blind noise modeling and removal. In *NeurIPS*, 2019.
- Zongsheng Yue, Qian Zhao, Lei Zhang, and Deyu Meng. Dual adversarial network: Toward real-world noise removal and noise generation. In *ECCV*, 2020.
- Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Cycleisp: Real image restoration via improved data synthesis. In *CVPR*, 2020a.
- Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Learning enriched features for real image restoration and enhancement. In *ECCV*, 2020b.
- Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *CVPR*, 2021.
- Syed Waqas Zamir, Aditya Arora, Salman H. Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022.
- Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *TIP*, 2017a.
- Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *CVPR*, 2017b.

- Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *TIP*, 2018.
- Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *TPAMI*, 2021a.
- Lei Zhang, Xiaolin Wu, Antoni Buades, and Xin Li. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *JEI*, 2011.
- Qinglong Zhang and Yu-Bin Yang. Rest: An efficient transformer for visual recognition. In *NeurIPS*, 2021.
- Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. In *ICLR*, 2019.
- Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image restoration. *TPAMI*, 2020.
- Yulun Zhang, Kunpeng Li, Kai Li, Gan Sun, Yu Kong, and Yun Fu. Accurate and fast image denoising via attention guided scaling. *TIP*, 2021b.