CUR Transformer: A Convolutional Unbiased Regional Transformer for Image Denoising

KANG XU[#], Beijing University of Posts and Telecommunications, P. R. China
WEIXIN LI[#], [1]Beihang University, P. R. China [2]Zhongguancun Laboratory, P. R. China
XIA WANG, Beijing University of Posts and Telecommunications, P. R. China
XIAOJIE WANG, Beijing University of Posts and Telecommunications, P. R. China
KE YAN, Alibaba DAMO Academy, P. R. China
XIAOYAN HU, Beijing University of Posts and Telecommunications, P. R. China
XUAN DONG*, Beijing University of Posts and Telecommunications, P. R. China

Image denoising is a fundamental problem in computer vision and multimedia computation. Non-local filters are effective for image denoising. But existing deep learning methods that use non-local computation structures are mostly designed for high-level tasks, and global self-attention is usually adopted. For the task of image denoising, they have high computational complexity, and have a lot of redundant computation of uncorrelated pixels. To solve this problem and combine the marvelous advantages of non-local filter and deep learning, we propose a Convolutional Unbiased Regional (CUR) transformer. Based on the prior that, for each pixel, its similar pixels are usually spatially close, our insights are that 1) we partition the image into non-overlapped windows and perform regional self-attention to reduce the search range of each pixel, and 2) we encourage pixels across different windows to communicate with each other. Based on our insights, the CUR transformer is cascaded by a series of convolutional regional self-attention (CRSA) blocks with U-style short connections. In each CRSA block, we use convolutional layers to extract the query, key, and value features, namely **Q**, **K**, and **V**, of the input feature. Then, we partition the **Q**, **K**, and **V** features into local non-overlapped windows, and perform regional self-attention within each window to obtain the output feature of this CRSA block. Among different CRSA blocks, we perform the unbiased window partition by changing the partition positions of the windows. Experimental results show that the CUR transformer outperforms the state-of-the-art methods significantly on four low-level vision tasks, including real and synthetic image denoising, JPEG compression artifact reduction, and low-light image enhancement.

CCS Concepts: • **Computing methodologies → Reconstruction**; *Supervised learning*; *Image processing*.

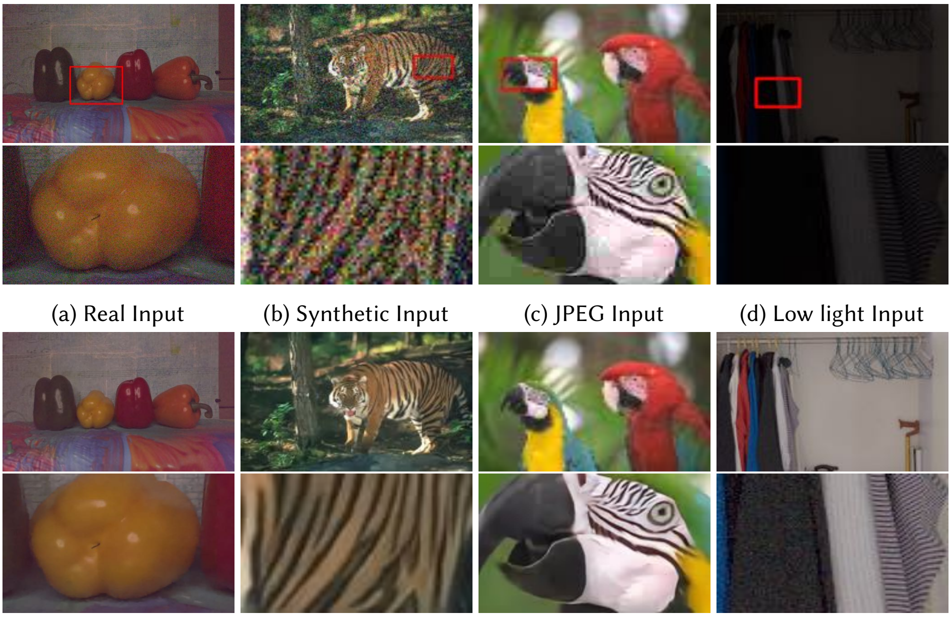
Additional Key Words and Phrases: image denoising, non-local filters, regional self-attention, computer vision, local non-overlapped windows

[#] denotes equal contribution. * denotes corresponding author: Xuan Dong.

Authors' addresses: Kang Xu[#], Beijing University of Posts and Telecommunications, Beijing, P. R. China, xukang981@foxmail.com; Weixin Li[#], [1]Beihang University, Beijing, P. R. China, [2]Zhongguancun Laboratory, Beijing, P. R. China, weixinli@buaa.edu.cn; Xia Wang, Beijing University of Posts and Telecommunications, Beijing, P. R. China, wangxia17863982566@163.com; Xiaojie Wang, Beijing University of Posts and Telecommunications, Beijing, P. R. China, xjwang@bupt.edu.cn; Ke Yan, Alibaba DAMO Academy, Beijing, P. R. China, yankethu@gmail.com; Xiaoyan Hu, Beijing University of Posts and Telecommunications, Beijing, P. R. China, hxy@bupt.edu.cn; Xuan Dong*, Beijing University of Posts and Telecommunications, Beijing, P. R. China, dongxuan8811@bupt.edu.cn..

1551-6857/2022/0-ART0 $15.00
https://doi.org/XX.XXXX/XXXXXXX.XXXXXXX

ACM Reference Format:
Kang Xu[#], Weixin Li[#], Xia Wang, Xiaojie Wang, Ke Yan, Xiaoyan Hu, and Xuan Dong[*]. 2022. CUR Transformer: A Convolutional Unbiased Regional Transformer for Image Denoising. *ACM Trans. Multimedia Comput. Commun. Appl.* 0, 0, Article 0 (2022), 22 pages. https://doi.org/XX.XXXX/XXXXXXX.XXXXXXX



(a) Real Input (b) Synthetic Input (c) JPEG Input (d) Low light Input

(e) Our result (f) Our result (g) Our result (h) Our result

Fig. 1. Example results in the tasks of real and synthetic image denoising, JPEG compression artifact reduction, and low-light image enhancement.

1 INTRODUCTION

As shown in Fig. 1, different kinds of noises, e.g. spot noise, compression noise and structure noise, widely exist in our daily used images. Image denoising is not only important for enhancing the human visual perception of the images, but also beneficial for improving the accuracy of the following high-level computer vision algorithms, e.g. face recognition and image classification.

Due to the success of deep learning in various fields, the study of deep learning-based filters for image denoising has attracted much attention. Most existing deep learning-based filters are CNN-based, e.g. DnCNN [65], and have achieved great performance improvements for image denoising. On the other hand, recently, the transformer network provides a different deep learning architecture and has obtained tremendous success in different vision tasks. The self-attention mechanism in transformer provides the non-local computation structure between pixels. Since non-local filters, e.g. BM3D [9], have achieved competitive performance among traditional hand-crafted filters, in this paper, we seek to expand the applicability of transformer and propose a deep learning-based non-local filter.

The limitation of most existing deep learning methods that use non-local computation structures, e.g. non-local neural network [52], is that they usually follow the basic global self-attention form of non-local computation and thus, for the image denoising task, every pixel must be compared with every other pixel in the whole image. As a result, 1) the computational complexity $O((HW)^2)$ is quite high, where H and W are the height and width of the image. In addition, 2) most of the compared pixels are uncorrelated and have little contribution to the image denoising task.

Based on the commonly mentioned prior [4], that, for each pixel, its best fitting pixels are usually spatially close, our insights are that 1) we reduce the search range of each pixel by performing regional self-attention, i.e. we partition the image into windows, and perform self-attention within each window. Thus, given the window size as $w \times w$, the computational complexity is reduced to $O(w^2(HW))$. 2) Pixels across different windows can communicate with each other so that the partition positions of windows will not affect the collection of similar pixels for image denoising.

Based on our insight, as shown in Fig. 2, 1) we propose the convolutional regional self-attention (CRSA) block. For the input feature, we extract the query, key, and value features, namely $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$, by multiple convolutional layers instead of commonly used embeddings of partitioned patches in previous transformer-based works [7, 49]. This is because the convolution operation can increase the receptive field of each pixel, i.e. increasing the number of communicated pixels for each pixel, across different windows when more CRSA blocks are cascaded. Then, the $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ maps are partitioned into non-overlapped windows. And regional self-attention is performed between features of pixels within each window to get the output feature of this CRSA block. 2) Among different CRSA blocks, as shown in Fig. 3, we perform an unbiased window partition, i.e. the partition positions of windows change among different blocks. This can avoid that each partitioned window always contains the same set of pixels, and help different pixels in local neighborhood communicate with each other. 3) To build the convolutional unbiased regional (CUR) transformer, we cascade a series of CRSA blocks with the popular U-style short connections [44] between them without downsampling and upsampling of the feature.

Experimental results on four tasks, i.e. real image denoising, synthetic image denoising, JPEG compression artifact reduction, and low-light image enhancement, show that the proposed CUR transformer outperforms all the comparison methods largely.

The contributions of this paper are the following:

- We develop a deep learning-based non-local filter that reduces the search range of pixels by regional self-attention.
- We integrate convolution into the regional self-attention block to encourage communications of neighboring pixels across different windows in each regional self-attention block.
- We propose an unbiased window partition mechanism to let neighboring pixels have unbiased opportunity to communicate in different regional self-attention blocks.

2 RELATED WORK

Hand-crafted image filters are one of the main solutions for the task of image denoising. According to the computation of the filtering weight [2], image filters can be classified into domain filters, bilateral filters, and non-local filters. Domain filters, e.g. Gaussian filter, have poor performance for edge-preserving and will introduce reversed gradients in edge regions. Bilateral filters, e.g. the original bilateral filter [47] and guided image filter [25], have better performance for edge-preserving. But, in edge regions, the edge pixels may have few pixels around them with similar colors, leading to reversed gradients too. Non-local filters, e.g. non-local means [5] and BM3D [9], have better performance due to no smoothness assumptions about neighboring pixels. The advantage is also analyzed by [31]. However, traditional hand-crafted image filters usually have many parameters that need to be adjusted manually, and thus have several limitations, e.g. weak generalization ability and low denoising accuracy, especially in comparison with the deep learning based denoising methods.

Existing deep learning based filters are mostly CNN-based, like SRCNN [12] and RCAN [66] (which combines CNN with channel attention) for super resolution, DnCNN [65], TNRD [8] and HRCN [29] for image denoising, and RDN [68] for various low-level tasks. With the help of CNN, these filters are powerful and have obtained impressive performance improvements in the entire field. However, the basic structure of CNN is usually the 3 × 3 convolution. Under this structure, pixels with very small distance (less than 3 × 3) can communicate with each other directly, and most pixels in the local neighborhood have to communicate with each other indirectly by multiple layers of convolutions. A straight-forward solution to enable more direct communications of CNN-based methods is to enlarge the convolution kernel, but this increases the number of model parameters dramatically and makes the model difficult to be trained. Since deep learning based filters are not only import for image enhancement problems [13–17], but also beneficial to high-level vision applications [20, 28, 33, 34, 53–55], we propose a new deep learning model in this paper.

Immigrated from Bert [10] in NLP fields, the self-attention based deep learning framework uses non-local computation structures and can enable the communications of pixels in non-local regions without increasing the model parameters a lot. And a series of self-attention based models have been proposed in the field of computer vision. The work in [52] inserts several non-local neural network layers to CNN backbones to filter the high-level visual features. The pioneer visual transformer models, e.g. ViT [19] and DeiT [48], treat the image as a sequence of visual words and the processing is similar to the previous works in NLP fields, e.g. Bert. CVT [58] and Ceit [62] propose to combine convolution with self-attention in the transformer structure. However, most of them are designed for high-level vision tasks, e.g. facial expression recognition and image classification. Some recent works of RNAN [67], IPT [7] and Resformer [63] also show the marvelous power of the transformer for low-level tasks, which is closely related to the problem in this paper. However, most existing self-attention based methods perform global self-attention, where each pixel must be compared with all the other pixels over the image. This is not a big problem for high-level tasks because the feature size has usually been reduced to be small by pooling. But, when using global self-attention for image denoising, 1) the computational complexity is very high, due to the use of full resolution of pixels of the images. In addition, 2) most of the compared pixels are unrelated and have little contribution for image denoising.

Several recent works, e.g. Swin transformer [38], Twins [59], and HaloNets [3], propose to use regional self-attention backbones for high-level tasks, e.g. image classification. In comparison with global self-attention, regional self-attention can reduce the computational complexity a lot. But, directly using them for image denoising is not suitable and the denoising quality is not competitive, because during the straight-forward regional self-attention computation, the communications of different pixels in local neighborhood are usually biased. SwinIR [35] cascades the computation block from Swin transformer for the low-level tasks, but it also shares similar limitations of biased communications of different local neighboring pixels. In comparison, in this paper, 1) we use multiple convolutional layers to extract the **Q**, **K**, **V** features in the CRSA block. And 2) we use the unbiased window partition between different CRSA blocks. Lastly, 3) we cascade a series of CRSA blocks with U-style short connections. The differences make our method be able to encourage pixels to have more and unbiased communications with neighboring pixels across partitioned windows, and the framework is more suitable for generating full-resolution output for the image denoising task.

3 METHOD

An overview of the model is shown in Fig. 2. The CUR transformer is built by cascading multiple convolutional regional self-attention (CRSA) blocks (14 blocks in this paper) in a U-style connection way [44] without downsampling or upsampling.

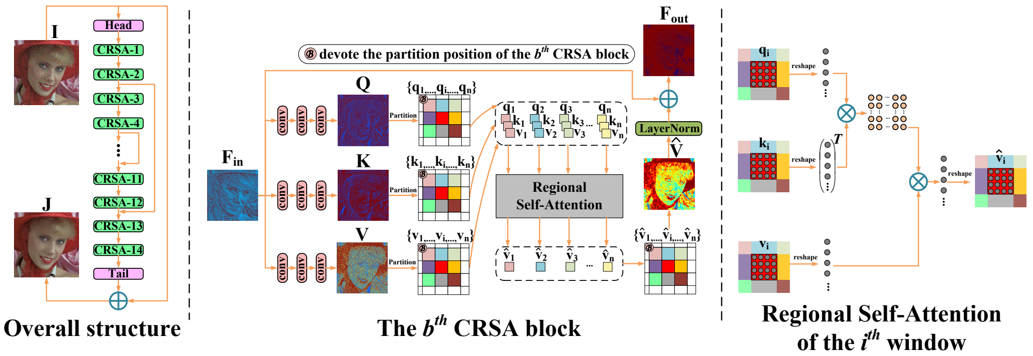

I
J
Head
CRSA-1
CRSA-2
CRSA-3
CRSA-4
⋮
CRSA-11
CRSA-12
CRSA-13
CRSA-14
Tail
Overall structure
ℬ devote the partition position of the b^{th} CRSA block
F_{in}
conv
Q
K
V
Partition
$\{q_1,...,q_i,...,q_n\}$
$\{k_1,...,k_i,...,k_n\}$
$\{v_1,...,v_i,...,v_n\}$
q_1 k_1 v_1 q_2 k_2 v_2 q_3 k_3 v_3 ... q_n k_n v_n
Regional Self-Attention
$\hat{v}_1$ $\hat{v}_2$ $\hat{v}_3$ ... $\hat{v}_n$
$\{\hat{v}_1,...,\hat{v}_i,...,\hat{v}_n\}$
$\hat{V}$
LayerNorm
F_{out}
The b^{th} CRSA block
q_i
k_i
v_i
reshape
T
$\hat{v}_i$
Regional Self-Attention of the i^{th} window

Fig. 2. The overall structure of our model. The CRSA blocks are cascaded with U-style short connections to build the CUR transformer. In each CRSA block, the features **Q**, **K**, **V** are extracted from the input feature $\mathbf{F}_{in}$ by multiple convolutional layers, which are then partitioned into non-overlapped windows. Regional self-attention for the partitioned features in each window i, i.e. $\mathbf{q}_i$, $\mathbf{k}_i$, $\mathbf{v}_i$, is performed to get the processed feature $\widehat{\mathbf{v}}_i$. The features $\widehat{\mathbf{v}}_i$ over all windows are concatenated to get $\widehat{\mathbf{V}}$, which is then normalized by LayerNorm to get the output feature $\mathbf{F}_{out}$ with the residual connection with $\mathbf{F}_{in}$.

In each of the proposed CRSA blocks, the query, key, and value feature maps, i.e. $\mathbf{Q} \in R^{C\times H\times W}$, $\mathbf{K} \in R^{C\times H\times W}$, $\mathbf{V} \in R^{C\times H\times W}$, are extracted by multiple convolutional layers instead of the commonly used patch embedding in transformer-based works [7, 38], because convolution can encourage the communications of different pixels across the boundary of partitioned windows, where C is the channel number (32 in this paper), H and W are the height and width of the image respectively. Then, given the partition position $\mathscr{B}$, the $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ features are partitioned into non-overlapped windows with the window size of $w \times w$. For each window i, the partitioned features from **Q**, **K**, **V** are named $\mathbf{q}_i \in R^{C\times w\times w}$, $\mathbf{k}_i \in R^{C\times w\times w}$, $\mathbf{v}_i \in R^{C\times w\times w}$, respectively. Next, we perform the regional self-attention, i.e. conducting the self-attention between $\mathbf{q}_i, \mathbf{k}_i, \mathbf{v}_i$ within each window i to obtain the output feature $\widehat{\mathbf{v}}_i$ of this window, and collect the $\widehat{\mathbf{v}}_i$ over all windows to obtain the final output feature $\hat{\mathbf{V}}$. In comparison with the global self-attention in traditional vision transformers, e.g. RNAN[67], whose computational complexity is $O((HW)^2)$, the regional self-attention only lets pixels within the same window perform self-attention with each other. In each window, the computational complexity is $O(w^4)$ and there are $\frac{HW}{w^2}$ windows in total, so the computational complexity of the regional self-attention is $O(w^2(HW))$, which is much smaller than $O((HW)^2)$ of the global self-attention.

In each partitioned window of the CRSA block, the center pixels can have enough communications with their neighboring pixels, but the corner pixels can only have communications with just a part of their neighboring pixels, e.g. the top-left pixel can only communicate with a quarter of its neighboring pixels due to the partition. This leads to biased sampling of neighboring pixels for different pixels. So, as shown in Fig. 3, we propose an unbiased window partition strategy to change the partition positions $\mathscr{B}$ in different CRSA blocks. In detail, in neighboring CRSA blocks, we change the window partition positions along the diagonal direction, so as to avoid that some pixels always locate in the corner of the partitioned windows in different CRSA blocks and thus avoid biased treatment of different pixels.

The head module consists of three convolutional layers, and each layer has a 3 × 3 convolution, Batch Normalization and ReLu, with the channel number of C. The tail module consists of a 1 × 1 convolutional layer without Batch Normalization or ReLu, with the channel number of 3.

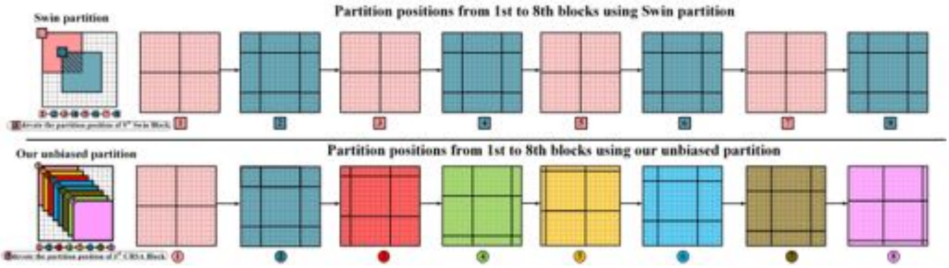

Swin partition
Partition positions from 1st to 8th blocks using Swin partition
1
2
3
4
5
6
7
8
Our unbiased partition
Partition positions from 1st to 8th blocks using our unbiased partition

Fig. 3. The partition positions of windows in different Swin blocks using the Swin-style partition and in different CRSA blocks using our unbiased window partition. Here, for simplification, we show the example of 8 blocks with the window size of 8 × 8. The region marked by slashes shows the set of pixels that are always partitioned into the same window according to the Swin-style partition, leading to biased sampling of neighboring pixels.

3.1 Convolutional Regional Self-Attention (CRSA)

In each CRSA block, as shown in Fig. 2, the convolutional part consists of three convolutional layers to extract the feature maps $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ from the input feature $\mathbf{F}_{in}$. Each convolutional layer has a 3 × 3 convolution with Batch Normalization and ReLu.

Then, we partition the feature maps into non-overlapped windows, and get the features $\mathbf{q}_i$, $\mathbf{k}_i$, $\mathbf{v}_i$ of each window i. We follow [38] to add the relative position bias $\mathbf{B} \in R^{w^2 \times w^2}$ and dimension C into the softmax, and the self-attention within each window is performed by

$$\widehat{\mathbf{v}}_i = Softmax(\mathbf{q}_i \mathbf{k}_i^T / \sqrt{C} + \mathbf{B})\mathbf{v}_i. \tag{1}$$

Then we concatenate $\widehat{\mathbf{v}}_i$ over all windows to get $\hat{\mathbf{V}}$, and get the output feature $\mathbf{F_{out}}$ of this CRSA block by

$$\mathbf{F}_{out} = LN(\hat{\mathbf{V}}) + \mathbf{F}_{in}, \tag{2}$$

where LN denotes a LayerNorm layer.

3.2 The Unbiased Window Partition

The work of Swin Transformer in [38] also changes partition positions of windows in successive blocks. But it only has two different partition positions among all CRSA blocks and repeats the partition positions in every two blocks. This will lead to the same set of pixels always being partitioned into the same window, and thus, for each pixel, its communication with the neighboring pixels is biased.

To overcome the limitation of the Swin-style partition, we propose the unbiased window partition in this paper. Fig. 3 provides the details about the partitioned windows from the 1st to the 8th blocks using the Swin-style partition and our unbiased partition mechanism to introduce the proposed method and explain the differences between Swin-style partition and ours (this example uses 8 blocks for simplification). Similar with Swin Transformer, we change the partition positions of windows in different CRSA blocks. And different from Swin Transformer that only has two different partition positions among all blocks and repeats the partition positions in every two blocks, we propose the unbiased partition strategy. As shown in the figure where the blocks with different partition positions are marked with different colors, for the settings of 8 CRSA blocks and the window size of 8×8, we select eight different partition positions.

To further explain why Swin is biased while ours is unbiased, we provide the Fig. 4. In the figure, we select three pixels A, B and C, and the distance between A and B is equal to the distance

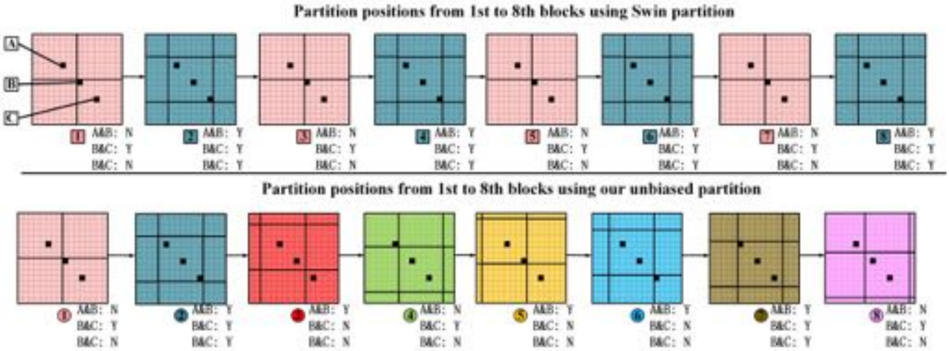


Fig. 4. An example to show the communication times between the pixels A, B, and C from the 1st to the 8th blocks ('Y' means the pixels have communications while 'N' means they do not have communication) using the Swin-style partition and our unbiased partition, and thus to explain why our method is 'unbiased'.

between B and C. According to the Swin-style partition (the top line of this figure), A and B have communications with 4 times, i.e. blocks 2,4,6,8. And, B and C have communications with 8 times, i.e. blocks 1,2,3,4,5,6,7,8. Although B has the same distance with A and C, the communication times of B&A and B&C are different, so we call the Swin-style partition biased. According to our method (the bottom line of this figure), A and B have communications with 5 times, i.e. blocks 2,3,5,6,7. And, B and C have communications with 5 times too, i.e. blocks 1,2,4,7,8. This shows that, in comparison with the Swin-style partition, the communication times of pixels are more unbiased using our method. In addition, for pixels with larger distance, for example pixels A and C, according to the Swin-style partition, they have communications with 4 times, i.e. blocks 2,4,6,8. And, according to our method, they have communications with 2 times, i.e. blocks 2,7, which are less than the communication times of A&B and B&C. In our opinion, our method is more reasonable because pixels with larger distances are usually less correlated for the image denoising task, and thus should have less communications. In short, as explained above, for pixels with the same distance, the communication times of our method are more unbiased than the Swin-style partition. In addition, for pixels with different distances, the communication times of our method are more reasonable than the Swin-style partition.

In our unbiased window partition method, the partition positions $\mathscr{B}$ of windows change like binary search along the diagonal direction in different CRSA blocks. In the traditional binary search, for an 1D array of 8 elements, the search order is that: The first search position is 5. The second search position is 3 or 7 (depending on the first search result). The third search position is 2 or 4 or 6 or 8. Following this binary search style, the order of the partition positions along the diagonal direction in our method is 1-5-3-7-2-4-6-8. The first and second CRSA blocks in our method have the same partition positions with the Swin-style partition, while the following 6 blocks have different partition positions. The advantage of the binary-search-style arrangement of the eight partition positions is that we can avoid that any two pixels always have communications in neighboring CRSA blocks and thus encourage each pixel to communicate with different pixels in neighboring CRSA blocks.

In the image boundary regions, the partitioned windows may not be with the size of $w \times w$. For example, as Fig. 5 shows, there are 16 partitioned windows, and except windows 6,7,10,11, the sizes of the other windows are not $w \times w$. In our processing, whatever the window size is, we let the pixels within each window perform the regional self-attention to obtain their output

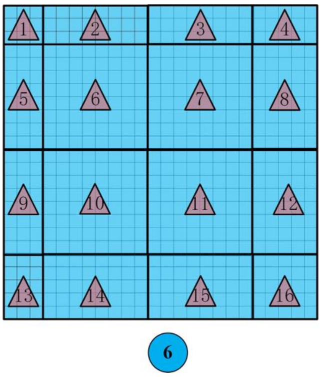


Fig. 5. An example to show the partitioned windows using the partition position of the 6th block according to our unbiased window partition mechanism (as introduced in Fig. 3), and to explain the processing in the image boundary regions.

features. When performing the regional self-attention within each window, as the subfigure of the Regional Self-Attention of the i^{th} window in Fig. 2 shows, the query, key, value features are firstly reshaped to 1D vectors and then the self-attention between these 1D vectors is performed. So, the irregular windows do not change the computation a lot. The only difference is that the sizes of the reshaped1D vectors are smaller than $1 * w^2$, but the computation can still be performed.

4 EXPERIMENTS

4.1 Datasets

The datasets for the four tasks of real image denoising, synthetic image denoising, JPEG compression artifact reduction, and low-light image enhancement are described below, respectively.

Real image denoising. The popular SIDD dataset [1] is used, which contains 320 pairs of noisy images and the noise-free ground-truth images for training and 1280 pairs of images for validation.

Synthetic image denoising. The well-known COCO dataset [37], which consists of over 160,000 color images of high diversity, is used as the source of the training dataset and the validation dataset. We select 1,500 images from COCO randomly, and divide them into 1000 images as the training dataset and 500 images as the validation dataset. And we use Urban100 [26] and BSD68 [39] datasets for the testing. To perform a fair comparison, for images in the training and validation datasets, we follow [7] to process the original images from COCO by cropping them to the resolution of 224×224 at random positions. Then, we generate the distorted images by adding zero-mean Gaussian noises at four different noise levels, i.e. σ^2 is set as 10, 30, 50, and 70, respectively.

JPEG compression artifact reduction. Same with the image denoising task, we follow [7] to generate the training and validation datasets for fair comparison. We randomly select 1,500 images from COCO dataset, divide them randomly into the training dataset with 1,000 images and the validation dataset with 500 images, and crop them to the resolution of 224×224. The Classic5 [21] and LIVE [45] datasets are used as the testing dataset. To generate the distorted images by JPEG compression, we use Matlab JPEG encoder [27] to generate compressed images with four different compress quality levels (CQL), i.e. 10, 20, 30, and 40, respectively.

Low-light image enhancement. We use the well-known LOL [57] dataset, which contains 500 low/normal-light image pairs. For fair comparison, we follow the method of Wei et al. [57] to select 485 image pairs of the dataset into the training dataset and the left 15 image pairs into the testing dataset.

4.2 Training

We implement our model using MindSpore [41]. We use one Nvidia 3090 GPU to train the deep model using the conventional Adam optimizer [30] with b1 = 0.9, and b2 = 0.999 for 200 epochs on each dataset. The initial learning rate is set as 0.0001 and decayed by half every 20 epochs.The batch size is set as 1. Training our model roughly takes 1 day for 200 epochs. For image denoising and image compression artifact reduction, we use end-to-end learning. But for the low-light image enhancement, we perform a two-step enhancement: first we use the state-of-the-art global tone mapping method of 3D LUT [64] to brighten the input low light images (noises are also amplified at the same time), and the intermediate results are then denoised to generate the final result by our method. The reason of using the two-step enhancement is that low-light image enhancement has two problems that need to be solved, i.e. tonal adjustment and denoising. Pushing a single network to solve the two problems is not a good strategy, and the work in [36] also has similar findings in related problems. Because in this paper, we focus more on the image denoising performance, the tonal adjustment task is finished by the existing method of 3D LUT. For all the comparison methods in the low-light image enhancement task, we also use 3D-LUT to perform the tonal adjustment.

4.3 Loss

For image denoising and JPEG compression artifact reduction, the loss function is

$$L_1 = ||\mathbf{J} - \mathbf{J}_{gt}||_1 \tag{3}$$

where $\mathbf{J}$ and $\mathbf{J}_{gt}$ are the output and ground-truth images, respectively. For the low-light image enhancement, SSIM loss function [56], which originally consists of luminance, contrast, and structure components, is utilized in our experiment. As mentioned above, we separate the enhancement into 2 steps and the first step, i.e. global tone mapping, is finished by 3D-LUT. Since the result of 3D-LUT may not have exactly the same luminance level with the ground-truth image, using all the three components in SSIM as the loss will have negative effect on the training. So, we remove the luminance component and the loss is

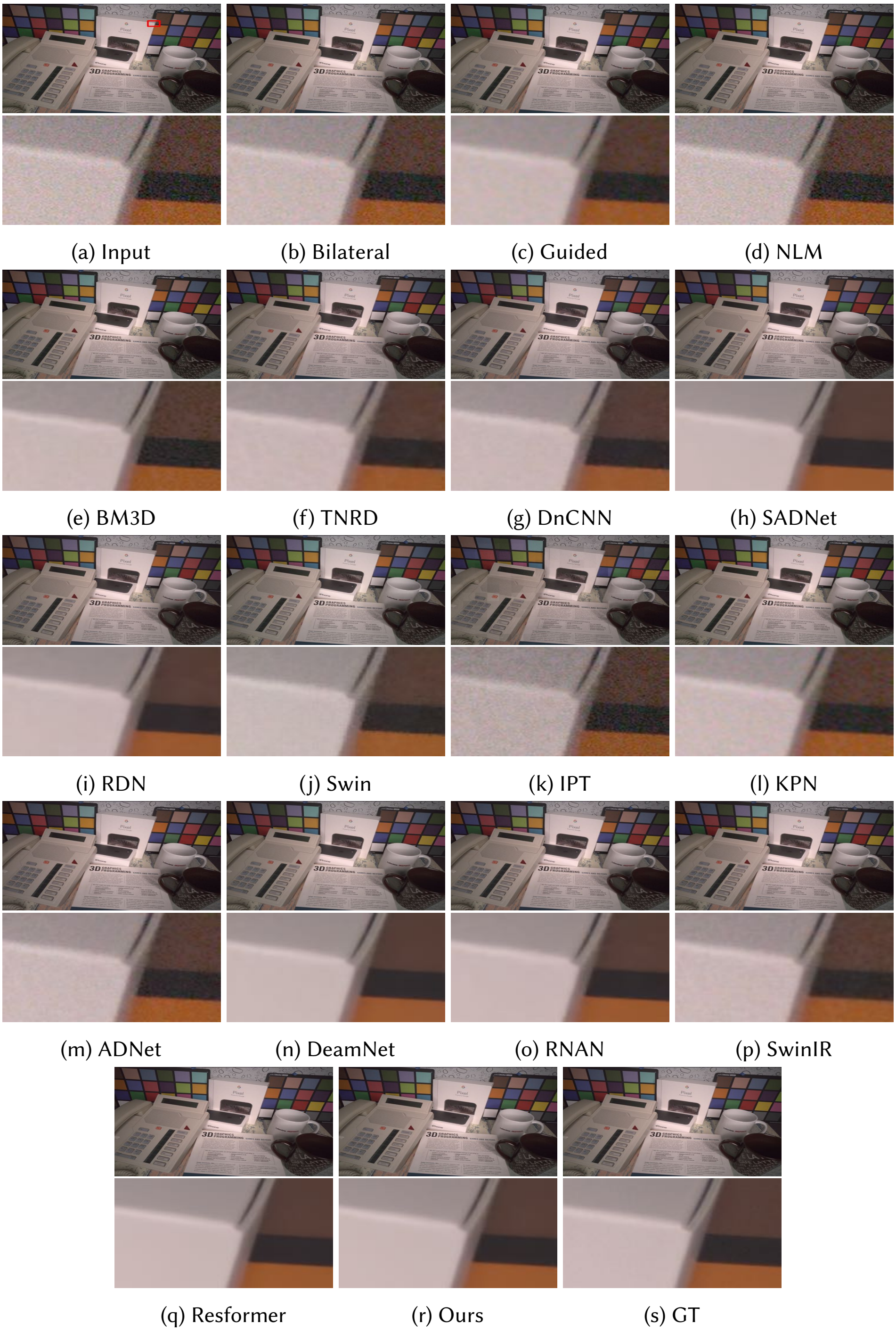
$$L_2 = \frac{2\sigma_{\mathbf{J},\mathbf{J}_{gt}} + C_1}{\sigma_{\mathbf{J}}^2 + \sigma_{\mathbf{J}_{gt}}^2 + C_1} \tag{4}$$

where $\sigma_{\mathbf{J},\mathbf{J}_{gt}}$ and σ^2 are the covariance and variance respectively and C_1 is a constant to prevent division by 0.

4.4 Comparison Methods

We compare our method with several state-of-the-art traditional hand-crafted filters, including gaussian filter, bilateral filter [47], guided image filter [25], Non-Local Means filter [5] and BM3D [9]. We also compare our method with several state-of-the-art CNN-based filters, including DnCNN [65], TNRD [8], RDN [68], SADNet [6], KPN [40], ADNet[46] and DeamNet[42]. Finally, we compare our method with the state-of-the-art transformer-based filters, including IPT [7], Swin transformer [38], RNAN [67], SwinIR [35] and Resformer [63]. To the best of our knowledge, IPT, RNAN and SwinIR are the vision transformers that are designed for low-level vision tasks. Swin transformer is not originally designed for low-level vision task, but it is a latest backbone based on the regional self-attention, and is related to our architecture. So, we adapt it by removing its pooling layers for comparison.

In the task of JPEG compression artifact reduction, we compare our method with the state-of-the-art compression artifact reduction methods, including SA-DCT [21] and ARCNN [11].



(a) Input (b) Bilateral (c) Guided (d) NLM

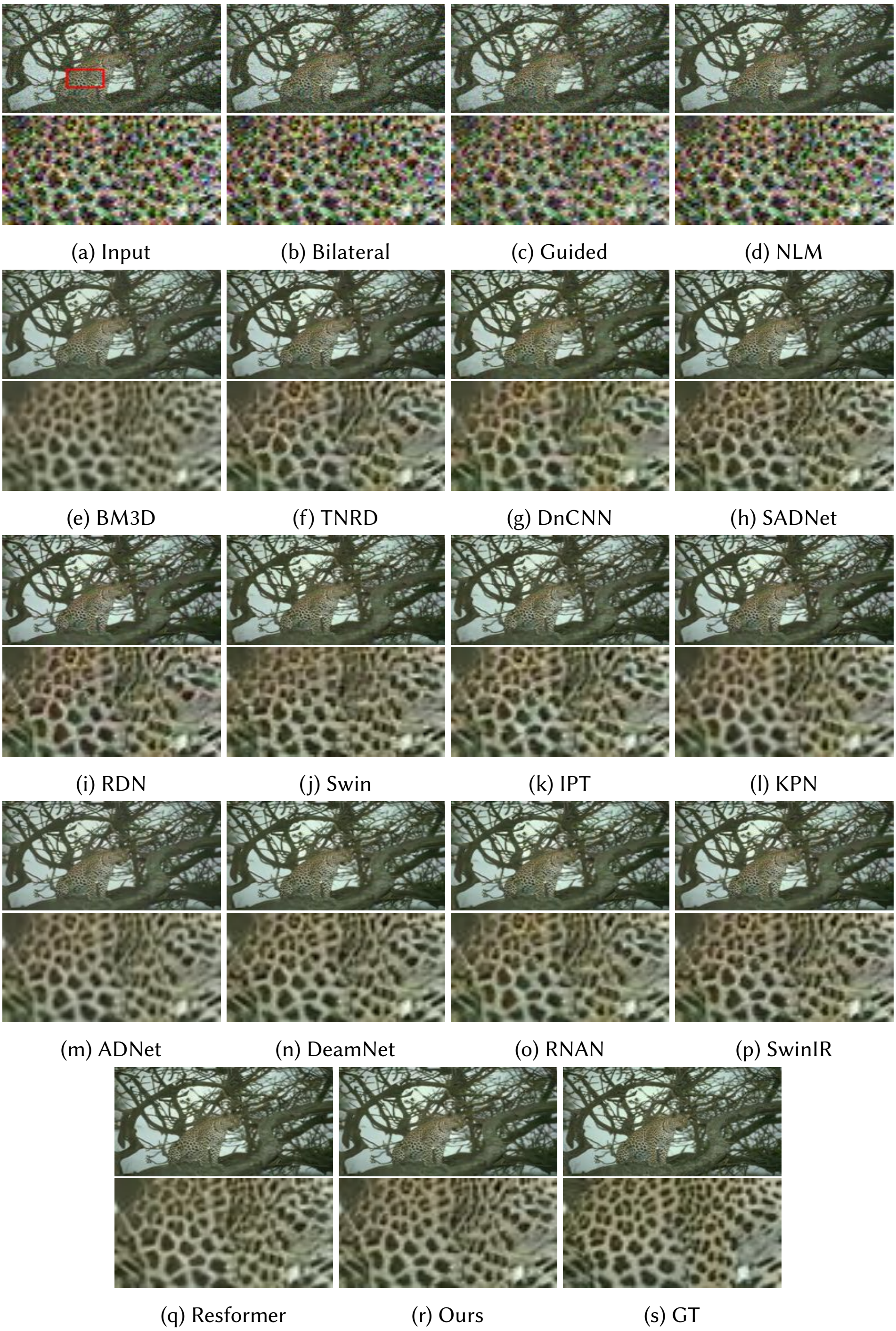
(e) BM3D (f) TNRD (g) DnCNN (h) SADNet

(i) RDN (j) Swin (k) IPT (l) KPN

(m) ADNet (n) DeamNet (o) RNAN (p) SwinIR

(q) Resformer (r) Ours (s) GT

Fig. 6. Example results in the task of real image denoising. The region marked in red is enlarged and shown in the following row.



(a) Input (b) Bilateral (c) Guided (d) NLM

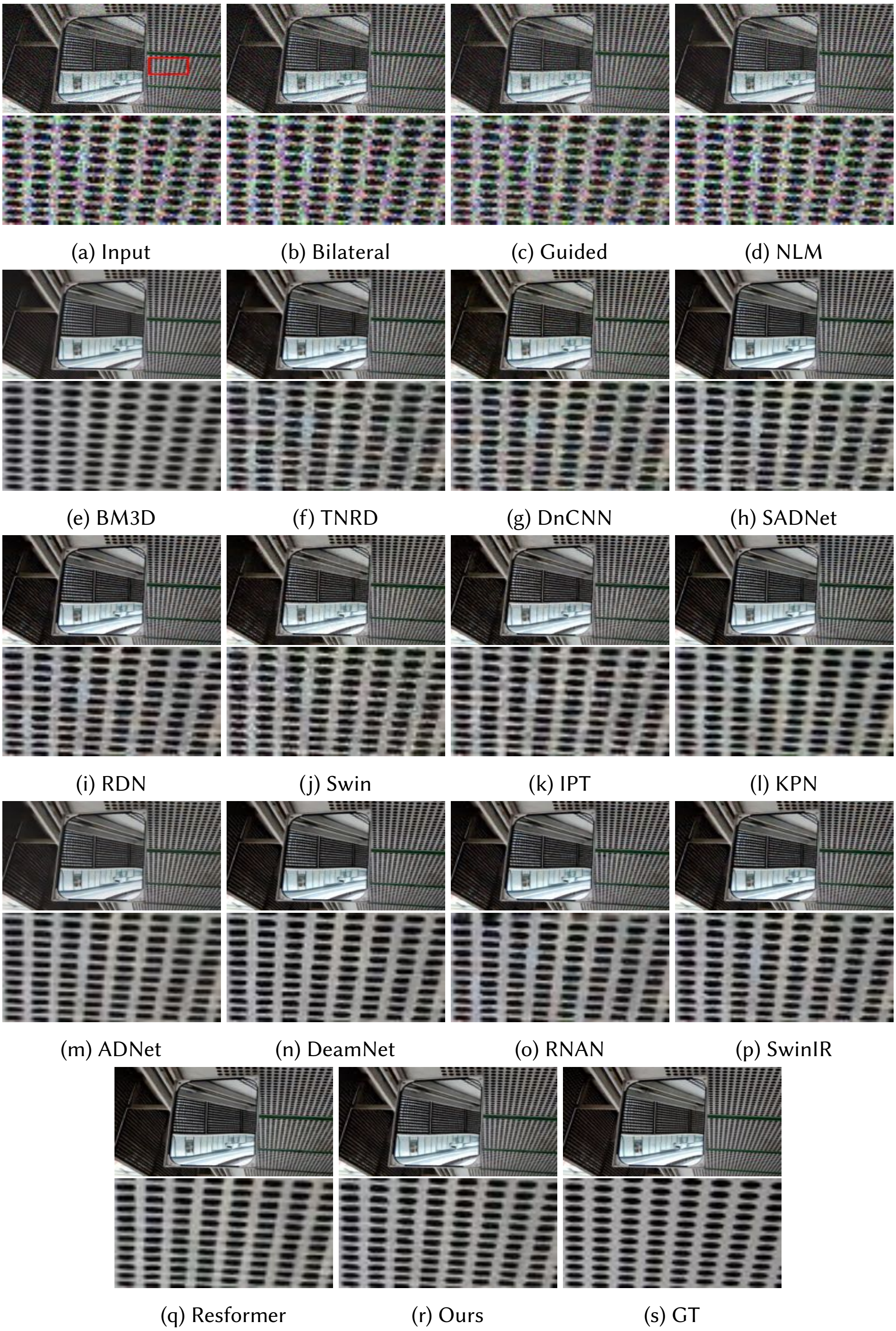
(e) BM3D (f) TNRD (g) DnCNN (h) SADNet

(i) RDN (j) Swin (k) IPT (l) KPN

(m) ADNet (n) DeamNet (o) RNAN (p) SwinIR

(q) Resformer (r) Ours (s) GT

Fig. 7. Example results in the task of synthetic image denoising. The region marked in red is enlarged and shown in the following row.



(a) Input (b) Bilateral (c) Guided (d) NLM

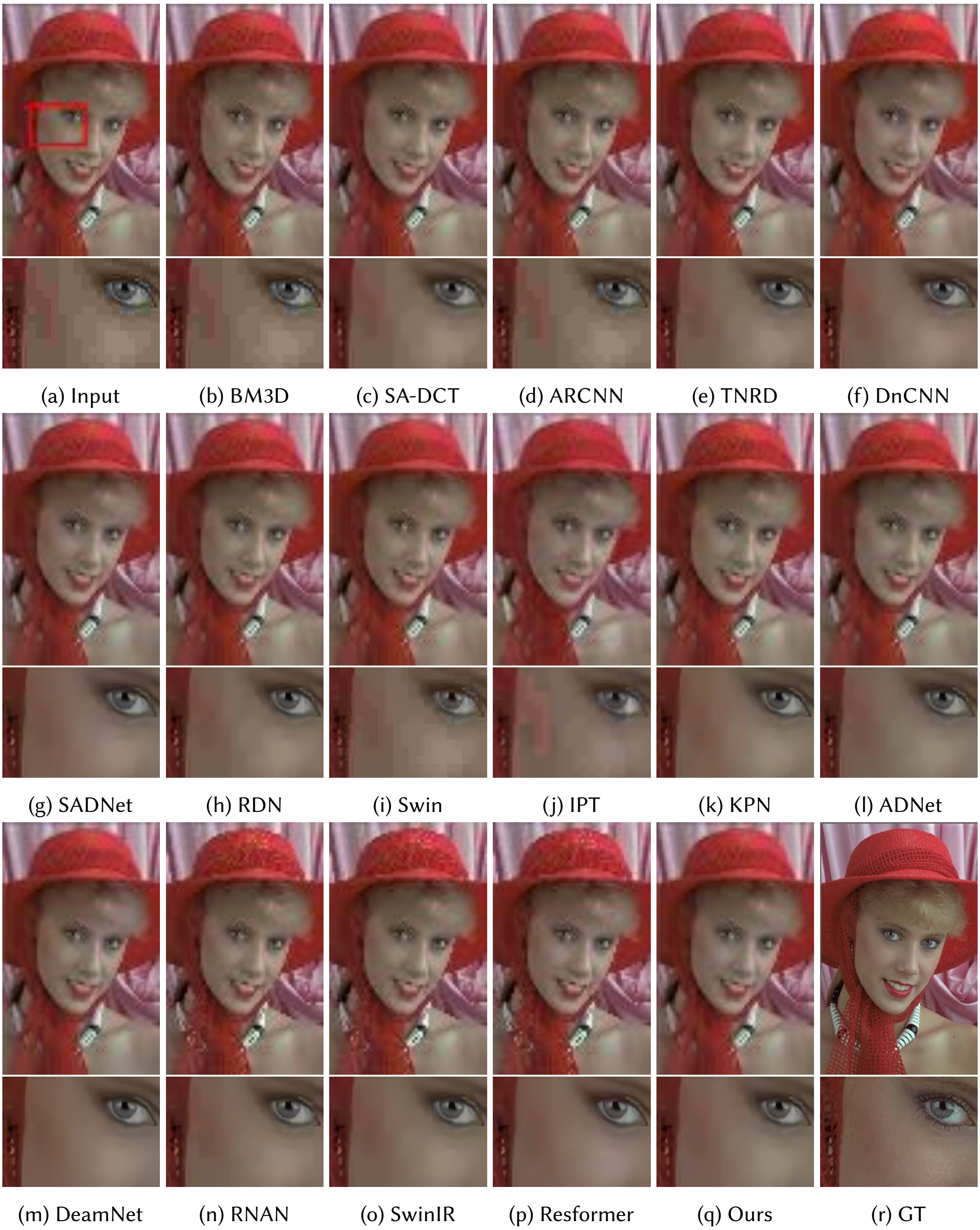
(e) BM3D (f) TNRD (g) DnCNN (h) SADNet

(i) RDN (j) Swin (k) IPT (l) KPN

(m) ADNet (n) DeamNet (o) RNAN (p) SwinIR

(q) Resformer (r) Ours (s) GT

Fig. 8. Example results in the task of synthetic image denoising. The region marked in red is enlarged and shown in the following row.

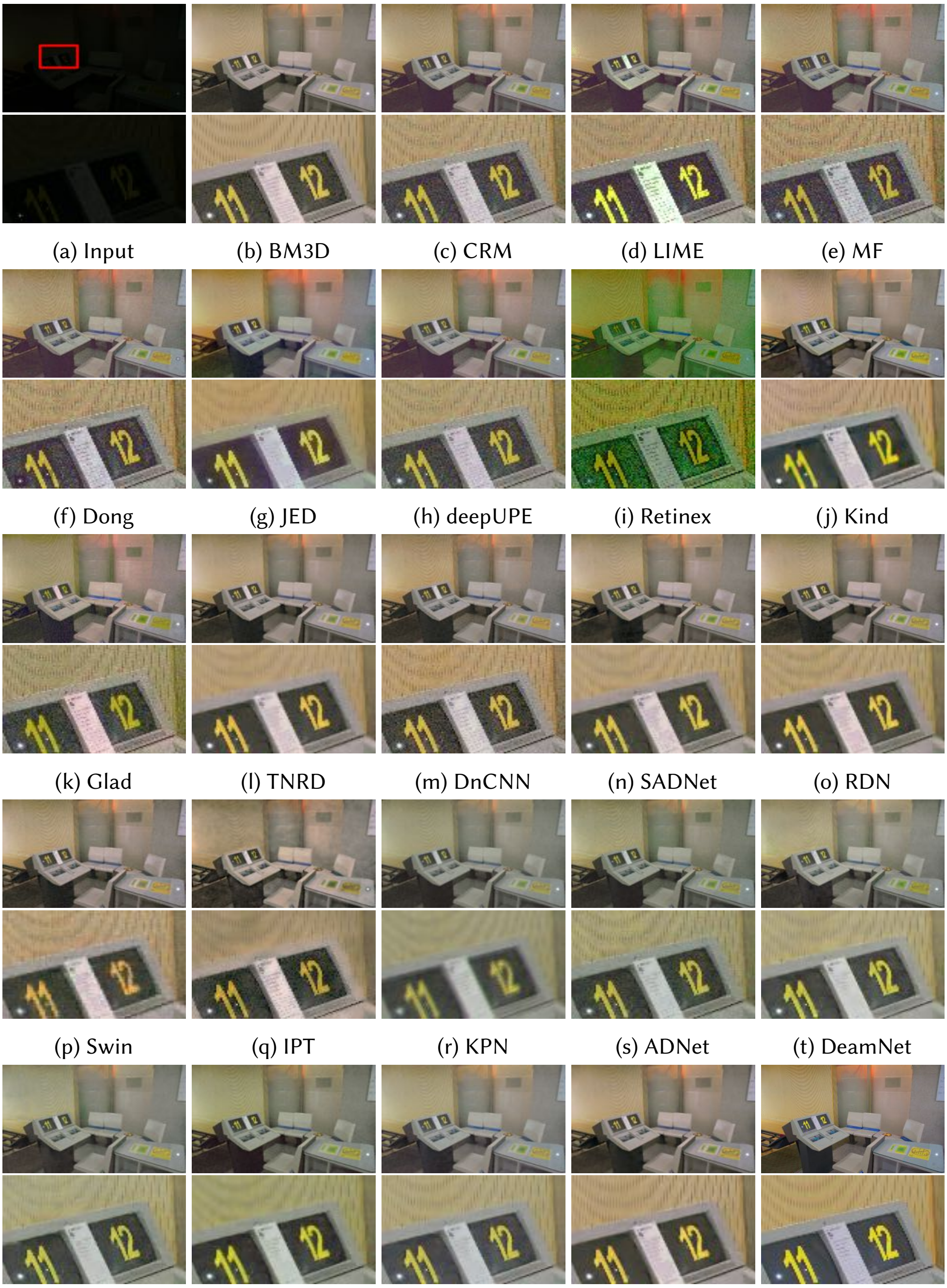


(a) Input (b) BM3D (c) SA-DCT (d) ARCNN (e) TNRD (f) DnCNN

(g) SADNet (h) RDN (i) Swin (j) IPT (k) KPN (l) ADNet

(m) DeamNet (n) RNAN (o) SwinIR (p) Resformer (q) Ours (r) GT

Fig. 9. Example results in the task of JPEG compression artifact reduction. The region marked in red is enlarged and shown in the following row.



(a) Input (b) BM3D (c) CRM (d) LIME (e) MF

(f) Dong (g) JED (h) deepUPE (i) Retinex (j) Kind

(k) Glad (l) TNRD (m) DnCNN (n) SADNet (o) RDN

(p) Swin (q) IPT (r) KPN (s) ADNet (t) DeamNet

(u) RNAN (v) SwinIR (w) Resformer (x) Ours (y) GT

Fig. 10. Example results in the task of low-light image enhancement. The region marked in red is enlarged and shown in the following row.

Table 1. Average PSNR (dB) and SSIM values of different methods on the SIDD dataset for real image denoising.

Method	Gaussian	Bilateral	Guided	NLM	BM3D	TNRD	DnCNN	SADNet	RDN
PSNR	29.27	30.53	33.41	27.55	35.56	36.72	37.18	39.53	39.05
SSIM	0.603	0.679	0.797	0.517	0.843	0.912	0.899	0.936	0.932
Method	IPT	Swin	KPN	ADNet	DeamNet	RNAN	SwinIR	Restormer	Ours
PSNR	26.47	36.31	32.06	34.01	38.84	38.54	38.97	39.00	**40.08**
SSIM	0.798	0.890	0.761	0.796	0.935	0.926	0.930	0.938	**0.938**

We highlight the best-performing model in each metrics.

Table 2. Average PSNR (dB) and SSIM values of different methods on the Urban100 and CBSD68 datasets for synthetic image denoising with four different noise level (σ^2).

	Urban100				**CBSD68**			
Method	σ^2=10	σ^2=30	σ^2=50	σ^2=70	σ^2=10	σ^2=30	σ^2=50	σ^2=70
Gaussian	24.68/0.793	23.24/0.669	21.42/0.557	19.71/0.469	27.75/0.809	25.31/0.653	22.85/0.513	20.79/0.413
Bilateral	32.50/0.912	20.04/0.495	15.32/0.318	12.75/0.229	32.89/0.894	19.99/0.386	15.21/0.216	12.64/0.145
Guided	29.40/0.881	22.04/0.565	16.89/0.358	13.87/0.252	30.55/0.862	22.52/0.479	16.96/0.258	13.85/0.166
NLM	30.07/0.882	21.15/0.575	17.61/0.421	15.68/0.329	29.94/0.833	21.76/0.498	18.44/0.349	16.57/0.263
BM3D	35.65/0.958	29.00/0.881	25.41/0.803	22.72/0.729	35.66/0.951	29.07/0.835	25.87/0.736	23.50/0.661
TNRD	34.46/0.964	28.17/0.879	25.51/0.804	23.86/0.742	35.12/0.950	29.07/0.841	26.64/0.762	25.24/0.706
DnCNN	35.07/0.961	28.60/0.876	25.22/0.768	23.38/0.695	35.47/0.949	29.19/0.832	26.45/0.729	24.90/0.656
SADNet	35.10/0.963	29.22/0.892	26.53/0.829	24.72/0.772	35.66/0.952	29.80/0.855	27.50/0.785	25.95/0.729
RDN	35.75/0.967	29.66/0.899	26.85/0.834	25.01/0.774	35.96/0.954	30.02/0.855	27.53/0.778	26.04/0.719
Swin	31.56/0.924	26.61/0.844	24.26/0.765	22.72/0.699	33.02/0.907	28.20/0.816	25.99/0.738	24.63/0.681
IPT	30.26/0.937	28.39/0.879	26.17/0.817	24.61/0.761	31.60/0.926	29.13/0.840	27.10/0.765	25.78/0.707
KPN	35.01/0.951	29.05/0.873	26.42/0.828	24.58/0.759	35.39/0.945	29.57/0.842	27.17/0.762	25.81/0.716
ADNet	34.76/0.921	28.85/0.809	25.69/0.808	22.54/0.678	34.59/0.892	29.07/0.820	26.33/0.756	23.43/0.693
DeamNet	35.49/0.946	29.58/0.865	27.38/0.853	25.24/0.798	35.79/0.934	29.12/0.832	27.59/0.758	26.24/0.739
RNAN	35.84/0.961	30.08/0.909	27.30/0.850	25.23/0.786	36.05/0.954	30.26/0.864	27.82/0.793	26.25/0.733
SwinIR	35.82/0.967	29.87/0.904	27.05/0.843	25.25/0.788	35.99/0.954	30.18/0.862	27.73/0.789	26.30/0.735
Resformer	35.91/0.969	29.98/0.908	27.24/0.853	25.47/0.805	36.04/0.955	30.13/0.863	27.65/0.792	26.21/0.740
Ours	**35.98/0.969**	**30.11/0.910**	**27.64/0.862**	**25.86/0.811**	**36.11/0.956**	**30.29/0.865**	**27.94/0.799**	**26.49/0.745**

We highlight the best-performing model in each column.

In the low light image enhancement task, we compare our method with the state-of-the-art low-light image enhancement methods, including CRM [61], LIME [24], BIMEF [60], SRIE [23], MF [22], RRM [32], Dong [18], JED [43], DeepUPE [50], RetinexNet [57], KinD [69] and GLAD [51].

For fair comparison, we re-train all the deep learning-based comparison methods and ours with the same training strategy. And in the objective evaluation, we use PSNR and SSIM [56] as the metrics.

4.5 Results

Objective results of the four tasks, i.e. real image denoising, synthetic image denoising, JPEG compression artifact reduction, and low-light image enhancement, are shown in Tables 1, 2, 3, and 4. Example subjective results are shown in Figs. 6, 7, 8, 9, and 10. From the results, we can notice that our method unsurprisingly outperforms the traditional hand-crafted methods a lot, i.e. the Gaussian filter, the bilateral filter, the guided filter, non-local means, and BM3D. This verifies the effectiveness of deep learning techniques for image denoising. In addition, our method also achieves better accuracy than the CNN-based methods, i.e. TNRD, DnCNN, SADNet, RDN, in all the four tasks. This shows that the regional non-local computation within the CSRA block of our method can provide more powerful ability for image denoising than the network structures of

Table 3. Average PSNR (dB) and SSIM values of different methods on the LIVE and Classic5 datasets for JPEG compression artifact reduction with four different compress quality levels (CQL).

	LIVE1				Classic5			
Method	CQL=10	CQL=20	CQL=30	CQL=40	CQL=10	CQL=20	CQL=30	CQL=40
Gaussian	25.24/0.735	26.44/0.785	26.84/0.803	27.05/0.812	27.69/0.757	28.63/0.796	28.95/0.811	29.09/0.818
Bilateral	26.07/0.766	28.52/0.842	29.83/0.872	30.73/0.889	28.48/0.785	30.60/0.843	31.75/0.868	32.45/0.881
Guided	25.96/0.742	27.67/0.790	28.45/0.809	28.95/0.820	28.08/0.762	29.41/0.801	30.06/0.818	30.42/0.827
NLM	25.69/0.749	28.06/0.830	29.37/0.865	30.29/0.886	27.82/0.767	30.14/0.841	31.51/0.872	32.46/0.889
BM3D	26.06/0.768	28.61/0.848	29.97/0.880	30.91/0.897	28.72/0.795	31.05/0.854	32.30/0.875	33.09/0.886
SA-DCT	26.63/0.778	28.83/0.848	30.03/0.879	30.88/0.897	28.88/0.795	30.91/0.853	32.13/0.879	32.99/0.894
ARCNN	26.50/0.774	28.75/0.846	30.01/0.877	30.83/0.894	28.59/0.784	30.75/0.848	32.00/0.877	32.87/0.894
TNRD	26.89/0.783	29.19/0.853	30.48/0.885	31.39/0.902	28.99/0.792	31.15/0.853	32.42/0.880	33.34/0.897
DnCNN	26.78/0.786	29.13/0.856	30.48/0.887	31.38/0.903	28.98/0.799	31.27/0.859	32.58/0.885	33.40/0.900
SADNet	27.12/0.792	29.29/0.859	30.67/0.890	31.44/0.905	29.12/0.802	31.32/0.861	32.62/0.885	33.37/0.898
RDN	27.13/0.793	29.48/0.861	30.80/0.891	31.70/0.908	29.28/0.804	31.42/0.861	32.71/0.887	33.60/0.901
Swin	26.37/0.768	28.58/0.840	29.79/0.873	30.64/0.891	28.39/0.777	30.52/0.842	31.82/0.873	32.71/0.891
IPT	22.04/0.721	19.64/0.751	17.18/0.735	16.69/0.738	20.87/0.675	19.07/0.691	16.76/0.663	15.13/0.669
KPN	27.14/0.794	29.55/0.871	30.82/0.895	31.73/0.915	29.26/0.806	31.47/0.865	32.73/0.882	33.62/0.899
ADNet	26.95/0.787	29.23/0.841	30.52/0.892	31.40/0.911	29.12/0.799	31.25/0.855	32.48/0.885	33.47/0.900
DeamNet	27.20/0.791	29.53/0.856	30.89/0.895	31.79/0.909	29.34/0.809	31.46/0.869	32.63/0.871	33.52/0.887
RNAN	27.23/0.796	29.64/0.864	30.88/0.894	31.78/0.910	29.27/0.807	31.50/0.864	32.78/0.889	33.66/0.903
SwinIR	27.22/0.794	29.60/0.862	30.90/0.892	31.81/0.908	29.26/0.803	31.48/0.861	32.79/0.887	33.68/0.902
Restormer	27.22/0.796	29.53/0.864	30.87/0.893	31.79/0.910	29.24/0.808	31.52/0.864	32.79/0.888	33.64/0.902
Ours	**27.31/0.806**	**29.68/0.873**	**30.97/0.902**	**31.85/0.918**	**29.42/0.817**	**31.60/0.873**	**32.87/0.898**	**33.74/0.912**

We highlight the best-performing model in each column.

Table 4. Average PSNR (dB) and SSIM values of different methods on the LOL dataset for low-light image enhancement.

Method	Gaussian	Bilateral	Guided	NLM	BM3D	TNRD	DnCNN	SADNet	RDN	Swin
PSNR	21.98	22.39	22.23	22.14	22.41	22.85	22.49	22.97	23.03	22.31
SSIM	0.796	0.812	0.808	0.768	0.829	0.820	0.684	0.830	0.837	0.765
Method	IPT	KPN	ADNet	DeamNet	RNAN	SwinIR	Resformer	CRM	LIME	BIMEF
PSNR	19.08	21.71	22.86	22.19	23.00	23.01	23.04	17.20	16.76	13.88
SSIM	0.719	0.727	0.750	0.818	0.831	0.832	0.829	0.622	0.444	0.595
Method	SRIE	MF	RRM	Dong	JED	DeepUPE	RetinexNet	KinD	GLAD	Ours
PSNR	13.03	16.97	15.36	16.72	13.69	13.36	16.77	19.66	19.72	**23.11**
SSIM	0.607	0.505	0.654	0.479	0.658	0.465	0.429	0.821	0.685	**0.849**

We highlight the best-performing model in each metrics.

only using convolutional layers. Also, our method has much better accuracy than the transformer-based methods, i.e. IPT, Swin transformer, RNAN, SwinIR and Resformer on all the four tasks. In comparison with IPT, RNAN and Resformer, the combination of convolution and regional self-attention structure of our CUR transform can reduce the redundant computation of unrelated pixels during the image denoising. In comparison with Swin transformer and SwinIR, extracting features by convolution, the unbiased window partition, and the U-style short connections of different CRSA blocks contribute to a more suitable non-local filter structure in our method for solving the image denoising problem. In conclusion, the results in all the four tasks demonstrate the effectiveness of the proposed model for removing various kinds of image noises.

We analyze the computational complexity of different types of self-attention methods in Table. 5, including global self-attention, global self-attention with patch embedding, and regional self-attention that we use in this paper. For the global self-attention, the computational complexity is $O((HW)^2)$. ViT [19] and IPT [7] change the traditional global self-attention by adopting the patch embedding within the global self-attention framework, and thus the time complexity is reduced to

Table 5. Computational complexity of different types of self-attention, i.e. global self-attention, global self-attention with patch embedding, and regional self-attention.

Type	Example Methods	Complexity
Global self-attention	RNAN[67], Non-local Neural Network[52]	$O(2(HW)^2C) = O((HW)^2)$
Global self-attention with patch embedding	ViT[19], IPT[7]	$O(2\frac{(HW)^2C}{p^4}) = O(\frac{(HW)^2}{p^4})$
Regional self-attention	Ours	$O(2HWw^2C) = O(HWw^2)$

Table 6. Computational costs of different methods for processing images with the size of 224 × 224.

Method	Flops/G	params/M	memory/MB	time/s/img
TNRD	2.832	0.056	2961	0.1816
DnCNN	28.02	0.558	2175	0.0223
SADNet	14.52	3.451	1363	0.0082
RDN	278.6	5.552	13761	0.0651
Swin	77.72	1.557	9435	0.1131
IPT	573.2	176.7	13346	0.2565
KPN	40.61	27.65	2399	0.0217
ADNet	26.16	0.521	2501	0.0270
DeamNet	111.9	1.876	6077	0.0707
SwinIR	43.99	0.866	14058	0.0935
Resformer	107.9	26.12	20381	0.3003
Ours	121.7	2.425	4105	0.0898

$O(\frac{(HW)^2}{p^4})$, where p is the patch size. For the regional self-attention, the computational complexity is $O(w^4)$ in each window with the window size of w, and there are $\frac{HW}{w^2}$ windows in total, so the computational complexity of the regional self-attention is $O(HWw^2)$, which is much smaller than the other two self-attention methods. Given a real example of $H = W = 1024$, $p = 4$, and the channel $C = 96$, the costs of global self-attention, global self-attention with patch embedding, and regional self-attention are $2.11 * 10^{14}$, $8.24 * 10^{11}$ and $3.95 * 10^{10}$, respectively. These show our method is much efficient than the global self-attention with or without patch embedding. Please note that in ViT and IPT, the channel C contains the feature of the set of pixels in each patch, while in global self-attention and our regional self-attention, the channel C only contains the feature of a single pixel. So, the value C in ViT and IPT is usually larger than the value C in global self-attention and our regional self-attention. In our method, the value C is set to be 32, and under this setting the cost of this real example is reduced to $1.32 * 10^{10}$.

We also report the computational costs of the comparison methods and ours in Table 6. In comparison with transformer-based methods, we can notice that the computational cost of our method is lower than IPT and the memory and time costs of our method are lower than Swin transformer and SwinIR. And the accuracy of our method, which is reported in Tables 1, 2, 3, and 4, is higher than them on all the four tasks. This verifies the effectiveness of the combination of regional self-attention, convolution, and the unbiased window partition in our CUR transformer for image denoising. In addition, the computational cost of our method is higher than most of the CNN-based methods, due to the self-attention computation architecture. But, as reported in Tables 1, 2, 3, and 4, the accuracy of the CUR transformer is higher than them for the four tasks. We believe the future study will be conducted to reduce the computational consumption, and the CUR transformer provides a good start for the field of image denoising to make use of transformer techniques for designing deep learning-based regional non-local filter.

4.6 Ablation Study

Table 7. Ablation study of different variants of our model. Average PSNR(dB) and SSIM values are reported.

Task	**Real**	**Synthetic**	**JPEG**	**Low-Light**
Dataset	SIDD	Urban100 (σ^2=50)	classic5 (CQL=10)	LOL
No conv.	39.01/0.881	26.60/0.829	29.06/0.797	23.04/0.824
Global	39.12/0.892	25.93/0.809	28.67/0.789	22.49/0.822
No shift	39.57/0.911	27.35/0.850	29.24/0.804	22.95/0.823
Swin-shift	39.78/0.929	27.49/0.855	29.21/0.803	22.97/0.824
No U-style	39.31/0.902	27.02/0.841	29.18/0.801	23.01/0.831
Ours	**40.08/0.938**	**27.64/0.862**	**29.42/0.817**	**23.11/0.849**

Table 8. Ablation study of our model with different numbers of training images. “Base” and “Large” devote using 1,000 and 10,000 images from COCO dataset for training, respectively. Average PSNR(dB) and SSIM values are reported.

	Base(σ^2=50)		**Large(σ^2=50)**	
Method	CBSD68	Urban100	CBSD68	Urban100
TNRD	26.64/0.762	25.51/0.804	27.25/0.765	26.30/0.814
DnCNN	26.45/0.729	25.22/0.768	27.46/0.776	26.63/0.826
SADNet	27.50/0.785	26.53/0.829	27.96/0.801	27.57/0.861
RDN	27.53/0.778	26.85/0.834	27.78/0.789	27.38/0.850
Swin	25.99/0.738	24.26/0.765	27.66/0.790	26.71/0.840
IPT	27.10/0.765	26.17/0.817	27.83/0.806	27.55/0.860
KPN	27.17/0.762	26.42/0.828	27.69/0.797	27.32/0.848
ADNet	26.33/0.756	25.69/0.808	27.32/0.762	26.53/0.828
DeamNet	27.59/0.758	27.38/0.853	27.79/0.798	27.68/0.863
RNAN	27.82/0.793	27.30/0.850	28.01/0.801	27.95/0.872
SwinIR	27.73/0.789	27.05/0.843	27.90/0.794	27.71/0.868
Resformer	27.65/0.792	27.24/0.853	27.94/0.801	27.89/0.871
Ours	**27.94/0.799**	**27.64/0.864**	**28.15/0.806**	**28.29/0.876**

We compare a number of different model variants at the key parts of our CUR transformer. The key ideas of our method include the combination of convolution and regional self-attention in the CRSA block, and the unbiased window partition among different CRSA blocks. So we 1) remove the convolutional layers in the CRSA block and use traditional embeddings of patches to extract the **Q**, **K**, **V** features, 2) remove the regional self-attention computation and use traditional global self-attention computation, which is very similar to the structure of the non-local neural network

[52], 3) replace the unbiased window partition by partitioning without changing the positions, 4) replace the unbiased window partition by the shifted partition of Swin transformer, and 5) remove the U-style short connections. Table 7 shows the performance of different model variants. The results show that any of these variants will degrade the image enhancement quality. This verifies the contributions of different parts in our model.

To show the performance differences with different number of training images, we also show the results of different comparison methods by training them with 1,000 and 10,000 images respectively. Table 8 shows the performance of different number of training images. The results show that, more training data will generally help different methods to obtain higher accuracy. And with different training data, the CUR transformer has stable improvements in comparison with the other methods.

Table 9. Computational costs of the CUR transformer with different window size w and the corresponding denoising accuracy on the datasets of CBSD68 and Urban100 (σ^2=50).

w	params/M	Flops/G	CBSD68	Urban100
7	2.416	115.1	27.75/0.789	27.18/0.847
14	2.425	121.7	27.94/0.799	27.64/0.862
28	2.478	148.3	28.02/0.802	27.69/0.866
56	2.601	253.8	28.05/0.809	27.71/0.867

To test the performance differences with different window sizes of our CUR transformer, we also show the computational costs of the CUR transformer under different window size w and the corresponding denoising results with σ^2=50 in Table 9. As shown, increasing the window size in a certain range will improve the performance of the model, but when the window size continues to increase, the performance will reach an upper limit, and the computational costs of the model will increase significantly. To balance the computational costs and accuracy of the CUR transformer, we use the window size of 14 in this paper.

5 CONCLUSIONS

We propose a Convolutional Unbiased Regional (CUR) transformer for solving the problem of image denoising in this paper. To overcome the limitations of high computational complexity and redundant computation of uncorrelated pixels of existing deep non-local filters, we propose a convolutional regional self-attention (CRSA) block to combine convolution and regional self-attention. We also cascade a series of CRSA blocks with U-style short connections, and among different CRSA blocks, we perform the unbiased window partition by changing the partition positions of windows. Experimental results on four tasks, i.e. real image denoising, synthetic image denoising, JPEG compression artifact reduction, and low-light image enhancement, show the effectiveness of the proposed method for removing different kinds of image noises.

The limitations of the proposed CUR transformer is that it is only suitable for the tasks where the high-frequency components of the images need smoothing, e.g. image denoising, while not suitable for some other tasks where the high-frequency components of the images need reconstructing like image super-resolution. It is because the regional self-attention operations within the CUR transformer actually perform weighted average of neighboring pixels to generate the output values. This process is good at removing wrong high-frequency components but is not good at recovering missing high-frequency components.

ACKNOWLEDGMENTS

This work is sponsored by the National Nature Science Foundation of China (No. 61802026, 61806016, 62076032 and 62276018) and CAAI-Huawei MindSpore Open Fund. We gratefully acknowledge the support of MindSpore, CANN (Compute Architecture for Neural Networks) and Ascend AI Processor used for this research.

REFERENCES

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. 2018. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1692–1700.

[2] Andrew Adams, Natasha Gelfand, Jennifer Dolson, and Marc Levoy. 2009. Gaussian kd-trees for fast high-dimensional filtering. In *ACM SIGGRAPH 2009 papers*. 1–12.

[3] Vaswani Ashish, Ramachandran Prajit, Srinivas Aravind, Parmar Niki, Hechtman Blake, and Shlens Jonathon. 2021. Scaling local self-attention for parameter efficient visual backbones. *CVPR* (2021).

[4] Thomas Brox, Oliver Kleinschmidt, and Daniel Cremers. 2008. Efficient nonlocal means for denoising of textural patterns. *IEEE Transactions on Image Processing* 17, 7 (2008), 1083–1092.

[5] Antoni Buades, Bartomeu Coll, and J-M Morel. 2005. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 2. IEEE, 60–65.

[6] Meng Chang, Qi Li, Huajun Feng, and Zhihai Xu. 2020. Spatial-adaptive network for single image denoising. In *European Conference on Computer Vision*. Springer, 171–187.

[7] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. 2021. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12299–12310.

[8] Yunjin Chen and Thomas Pock. 2016. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE transactions on pattern analysis and machine intelligence* 39, 6 (2016), 1256–1272.

[9] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. 2007. Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space. In *2007 IEEE International Conference on Image Processing*, Vol. 1. IEEE, I–313.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[11] Chao Dong, Yubin Deng, Chen Change Loy, and Xiaoou Tang. 2015. Compression artifacts reduction by a deep convolutional network. In *Proceedings of the IEEE International Conference on Computer Vision*. 576–584.

[12] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2015. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence* 38, 2 (2015), 295–307.

[13] Xuan Dong, Boyan Bonev, Yu Zhu, and Alan L Yuille. 2015. Region-based temporally consistent video post-processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 714–722.

[14] Xuan Dong, Weixin Li, Xiaoyan Hu, Xiaojie Wang, and Yunhong Wang. 2020. A colorization framework for monochrome-color dual-lens systems using a deep convolutional network. *IEEE Transactions on Visualization and Computer Graphics* (2020).

[15] Xuan Dong, Weixin Li, Xiaojie Wang, and Yunhong Wang. 2019. Learning a deep convolutional network for colorization in monochrome-color dual-lens system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 8255–8262.

[16] Xuan Dong, Weixin Li, Xiaojie Wang, and Yunhong Wang. 2020. Cycle-CNN for colorization towards real monochrome-color camera systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 10721–10728.

[17] Xuan Dong, Chang Liu, Weixin Li, Xiaoyan Hu, Xiaojie Wang, and Yunhong Wang. 2021. Self-supervised colorization towards monochrome-color camera systems using cycle CNN. *IEEE Transactions on Image Processing* 30 (2021), 6609–6622.

[18] Xuan Dong, Guan Wang, Yi Pang, Weixin Li, Jiangtao Wen, Wei Meng, and Yao Lu. 2011. Fast efficient algorithm for enhancement of low lighting video. In *2011 IEEE International Conference on Multimedia and Expo*. IEEE, 1–6.

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[20] Zhengyin Du, Suowei Wu, Di Huang, Weixin Li, and Yunhong Wang. 2019. Spatio-temporal encoder-decoder fully convolutional network for video-based dimensional emotion recognition. *IEEE Transactions on Affective Computing* 12, 3 (2019), 565–578.

[21] Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. 2007. Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. *IEEE transactions on image processing* 16, 5 (2007), 1395–1411.

[22] Xueyang Fu, Delu Zeng, Yue Huang, Yinghao Liao, Xinghao Ding, and John Paisley. 2016. A fusion-based enhancing method for weakly illuminated images. *Signal Processing* 129 (2016), 82–96.

[23] Xueyang Fu, Delu Zeng, Yue Huang, Xiao-Ping Zhang, and Xinghao Ding. 2016. A weighted variational model for simultaneous reflectance and illumination estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2782–2790.

[24] Xiaojie Guo, Yu Li, and Haibin Ling. 2016. LIME: Low-light image enhancement via illumination map estimation. *IEEE Transactions on image processing* 26, 2 (2016), 982–993.

[25] Kaiming He, Jian Sun, and Xiaoou Tang. 2012. Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence* 35, 6 (2012), 1397–1409.

[26] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. 2015. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5197–5206.

[27] Jeremy Jancsary, Sebastian Nowozin, and Carsten Rother. 2012. Loss-specific training of non-parametric image restoration models: A new state of the art. In *European Conference on Computer Vision*. Springer, 112–125.

[28] Jungseock Joo, Weixin Li, Francis F Steen, and Song-Chun Zhu. 2014. Visual persuasion: Inferring communicative intents of images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 216–223.

[29] Hui Ying Khaw, Foo Chong Soon, Joon Huang Chuah, and Chee-Onn Chow. 2019. High-density impulse noise detection and removal using deep convolutional neural network with particle swarm optimisation. *Iet image processing* 13, 2 (2019), 365–374.

[30] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[31] Anat Levin and Boaz Nadler. 2011. Natural image denoising: Optimality and inherent bounds. In *CVPR 2011*. IEEE, 2833–2840.

[32] Mading Li, Jiaying Liu, Wenhan Yang, Xiaoyan Sun, and Zongming Guo. 2018. Structure-revealing low-light image enhancement via robust retinex model. *IEEE Transactions on Image Processing* 27, 6 (2018), 2828–2841.

[33] Weixin Li, Xuan Dong, and Yunhong Wang. 2021. Human emotion recognition with relational region-level analysis. *IEEE Transactions on Affective Computing* (2021).

[34] Weixin Li, Jungseock Joo, Hang Qi, and Song-Chun Zhu. 2016. Joint image-text news topic detection and tracking by multimodal topic and-or graph. *IEEE Transactions on Multimedia* 19, 2 (2016), 367–381.

[35] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. 2021. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1833–1844.

[36] Zhetong Liang, Jianrui Cai, Zisheng Cao, and Lei Zhang. 2021. Cameranet: A two-stage framework for effective camera isp learning. *IEEE Transactions on Image Processing* 30 (2021), 2248–2262.

[37] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.

[38] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030* (2021).

[39] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. 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*, Vol. 2. IEEE, 416–423.

[40] Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. 2018. Burst denoising with kernel prediction networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2502–2510.

[41] MindSpore. [n.d.]. https://www.mindspore.cn/.

[42] Chao Ren, Xiaohai He, Chuncheng Wang, and Zhibo Zhao. 2021. Adaptive Consistency Prior Based Deep Network for Image Denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8596–8606.

[43] Xutong Ren, Mading Li, Wen-Huang Cheng, and Jiaying Liu. 2018. Joint enhancement and denoising method via sequential decomposition. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 1–5.

[44] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.

[45] HR Sheikh. 2005. LIVE image quality assessment database release 2. *http://live. ece. utexas. edu/research/quality* (2005).

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

[47] Carlo Tomasi and Roberto Manduchi. 1998. Bilateral filtering for gray and color images. In *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*. IEEE, 839–846.

[48] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2021. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*. PMLR, 10347–10357.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[50] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. 2019. Underexposed photo enhancement using deep illumination estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6849–6857.

[51] Wenjing Wang, Chen Wei, Wenhan Yang, and Jiaying Liu. 2018. GLADNet: Low-light enhancement network with global awareness. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 751–755.

[52] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7794–7803.

[53] Xueping Wang, Yunhong Wang, and Weixin Li. 2019. U-Net conditional GANs for photo-realistic and identity-preserving facial expression synthesis. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 15, 3s (2019), 1–23.

[54] Xueping Wang, Yunhong Wang, Weixin Li, Zhengyin Du, and Di Huang. 2021. Facial Expression Animation by Landmark Guided Residual Module. *IEEE Transactions on Affective Computing* (2021).

[55] Yunhong Wang, Zhaoxiang Zhang, Weixin Li, and Fangyuan Jiang. 2012. Combining tensor space analysis and active appearance models for aging effect simulation on face images. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42, 4 (2012), 1107–1118.

[56] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

[57] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. 2018. Deep retinex decomposition for low-light enhancement. *arXiv preprint arXiv:1808.04560* (2018).

[58] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. 2021. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808* (2021).

[59] Chu Xiangxiang, Tian Zhi, Wang Yuqing, Zhang Bo, Ren Haibing, Wei Xiaolin, and Xia Huaxia. 2021. Twins: Revisiting spatial attention design in vision transformers. *Arxiv* (2021).

[60] Zhenqiang Ying, Ge Li, and Wen Gao. 2017. A bio-inspired multi-exposure fusion framework for low-light image enhancement. *arXiv preprint arXiv:1711.00591* (2017).

[61] Zhenqiang Ying, Ge Li, Yurui Ren, Ronggang Wang, and Wenmin Wang. 2017. A new low-light image enhancement algorithm using camera response model. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 3015–3022.

[62] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. 2021. Incorporating convolution designs into visual transformers. *arXiv preprint arXiv:2103.11816* (2021).

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

[64] Hui Zeng, Jianrui Cai, Lida Li, Zisheng Cao, and Lei Zhang. 2020. Learning image-adaptive 3D lookup tables for high performance photo enhancement in real-time. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[65] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. 2017. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing* 26, 7 (2017), 3142–3155.

[66] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. 2018. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*. 286–301.

[67] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. 2019. Residual non-local attention networks for image restoration. *arXiv preprint arXiv:1903.10082* (2019).

[68] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. 2020. Residual dense network for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 7 (2020), 2480–2495.

[69] Yonghua Zhang, Jiawan Zhang, and Xiaojie Guo. 2019. Kindling the darkness: A practical low-light image enhancer. In *Proceedings of the 27th ACM international conference on multimedia*. 1632–1640.