

# Spatial-Adaptive Network for Single Image Denoising

Meng Chang<sup>1</sup>, Qi Li<sup>1</sup>, HuaJun Feng<sup>1</sup>, and Zhihai Xu<sup>1</sup>

State Key Lab for Modern Optical Instruments, Zhejiang University, China  
{changm, liqi, fenghj, xuzh}@zju.edu.cn

**Abstract.** Previous works have shown that convolutional neural networks can achieve good performance in image denoising tasks. However, limited by the local rigid convolutional operation, these methods lead to oversmoothing artifacts. A deeper network structure could alleviate these problems, but at the cost of additional computational overhead. In this paper, we propose a novel spatial-adaptive denoising network (SAD-Net) for efficient single image blind noise removal. To adapt to changes in spatial textures and edges, we design a residual spatial-adaptive block. Deformable convolution is introduced to sample the spatially related features for weighting. An encoder-decoder structure with a context block is introduced to capture multiscale information. By conducting noise removal from coarse to fine, a high-quality noise-free image is obtained. We apply our method to both synthetic and real noisy image datasets. The experimental results demonstrate that our method outperforms the state-of-the-art denoising methods both quantitatively and visually.

**Keywords:** image denoising, image restoration, image processing

## 1 Introduction

Image denoising is an important task in computer vision. During image acquisition, noise is often unavoidable due to imaging environment and equipment limitations. Therefore, noise removal is an essential step, not only for visual quality but also for other computer vision tasks. Image denoising has a long history, and many methods have been proposed. Many of the early model-based methods found natural image priors and then applied optimization algorithms to solve the model iteratively [23, 2, 30, 41]. However, these methods are time consuming and cannot effectively remove noise. With the rise of deep learning, convolutional neural networks (CNNs) have been applied to image denoising tasks and have achieved high-quality results.

On the other hand, the early works assumed that noise is independent and identically distributed. Additive white Gaussian noise (AWGN) is often adopted to create synthetic noisy images. People now realize that noise presents in more complicated forms that are spatially variant and channel dependent. Therefore, some recent works have made progress in real image denoising [26, 39, 12, 4].

However, despite numerous advances in image denoising, some issues remain to be resolved. A traditional CNN can use only the features in local fixed-location neighborhoods, but these may be irrelevant or even exclusive to the current location. Due to their inability to adapt to textures and edges, CNN-based methods result in oversmoothing artifacts and some details are lost. In addition, the receptive field of a traditional CNN is relatively small. Many methods deepen the network structure [27] or use a non-local module to expand the receptive field [18, 37]. However, these methods lead to high computational memory and time consumption, hence they cannot be applied in practice.

In this paper, we propose a spatial-adaptive denoising network (SADNet) to address the above issues. A residual spatial-adaptive block (RSAB) is designed to adapt to changes in spatial textures and edges. We introduce the modulated deformable convolution in each RSAB to sample the spatially relevant features for weighting. Moreover, we incorporate the RSAB and residual blocks (Res-Block) in an encoder-decoder structure to remove noise from coarse to fine. To further enlarge the receptive field and capture multiscale information, a context block is applied to the coarsest scale. Compared to the state-of-the-art methods, our method can achieve good performance while maintaining a relatively small computational overhead.

In conclusion, the main contributions of our method are as follows:

- We propose a novel spatial-adaptive denoising network for efficient noise removal. The network can capture the relevant features from complex image content, and recover details and textures from heavy noise.
- We propose the residual spatial-adaptive block, which introduces deformable convolution to adapt to spatial textures and edges. In addition, using an encoder-decoder structure with a context block to capture multiscale information, we can estimate offsets and remove noise from coarse to fine.
- We conduct experiments on multiple synthetic image datasets and real noisy datasets. The results demonstrate that our model achieves state-of-the-art performances on both synthetic and real noisy images with a relatively small computational overhead.

## 2 Related works

In general, image denoising methods include model-based and learning-based methods. Model-based methods attempt to model the distribution of natural images or noise. Then, using the modeled distribution as the prior, they attempt to obtain clear images with optimization algorithms. The common priors include local smoothing [23, 30], sparsity [2, 20, 33], non-local self-similarity [5, 9, 8, 34, 11] and external statistical prior [41, 32]. Non-local self-similarity is the notable prior in the image denoising task. This prior assumes that the image information is redundant and that similar structures exist within a single image. Then, self-similar patches are found in the image to remove noise. Many methods have been proposed based on the non-local self-similarity prior including NLM [5], BM3D [9, 8], and WNNM [11, 34], all of which are currently widely used.

With the popularity of deep neural networks, learning-based denoising methods have developed rapidly. Some works combine natural priors with deep neural networks. TRND [7] introduced the field-of-experts prior into a deep neural network. NLNet [17] combined the non-local self-similarity prior with a CNN. Limited by the designed priors, their performance is often inferior compared to end-to-end CNN methods. DnCNN [35] introduced residual learning and batch normalization to implement end-to-end denoising. FFDNet [36] introduced the noise level map as the input and enhanced the flexibility of the network for non-uniform noise. MemNet [27] proposed a very deep end-to-end persistent memory network for image restoration, which fuses both short-term and long-term memories to capture different levels of information. Inspired by the non-local self-similarity prior, a non-local module [28] was designed for neural networks. NLRN [18] attempted to incorporate non-local modules into a recurrent neural network (RNN) for image restoration. N3Net [26] proposed neural nearest neighbors block to achieve non-local operation. RNAN [37] designed non-local attention blocks to capture global information and pay more attention to the challenging parts. However, non-local operations lead to high memory usage and time consumption.

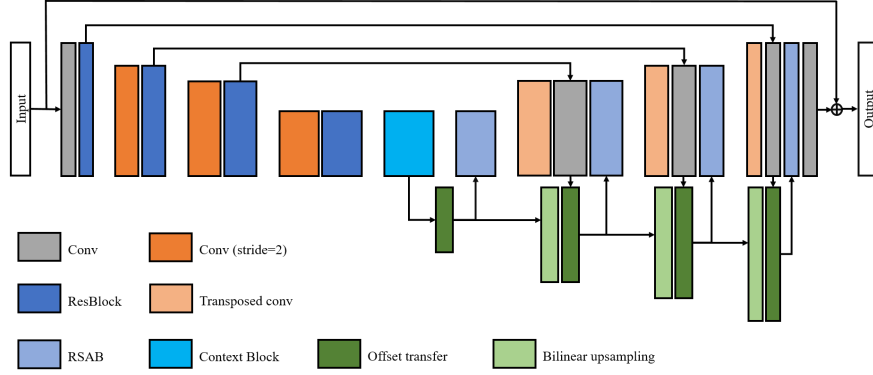
Recently, the focus of researchers has shifted from AWGN to more realistic noise. Some recent works have made progress on real noisy images. Several real noisy datasets have been established by capturing real noisy scenes [25, 3, 1], which promotes research into real-image denoising. N3Net [26] demonstrated the significance on real noisy dataset. CBDNet [12] trained two subnets to sequentially estimate noise and perform non-blind denoising. PD [39] applied the pixel-shuffle downsampling strategy to approximate the real noise to AWGN, which can adapt the trained model to real noises. RIDNet [4] proposed a one-stage denoising network with feature attention for real image denoising. However, these methods lack adaptability to image content and result in oversmoothing artifacts.

### 3 Framework

The architecture of our proposed spatial-adaptive denoising network (SADNet) is shown in Fig. 1. Let  $x$  denotes a noisy input image and  $\hat{y}$  denotes the corresponding output denoised image. Then our model can be described as follows:

$$\hat{y} = \text{SADNet}(x). \quad (1)$$

We use one convolutional layer to extract the initial features from the noisy input; then those features are input into a multiscale encoder-decoder architecture. In the encoder component, we use ResBlocks [14] to extract features of different scales. However, unlike the original ResBlock, we remove the batch normalization and use leaky ReLU [19] as the activation function. To avoid damaging the image structures, we limit the number of downsampling operations and implement a context block to further enlarge the receptive field and capture multiscale information. Then, in the decoder component, we design residual spatial-adaptive



**Fig. 1.** The framework of our proposed spatial-adaptive denoising network.

blocks (RSABs) to sample and weight the related features to remove noise and reconstruct the textures. In addition, we estimate the offsets and transfer them from coarse to fine, which is beneficial for obtaining more accurate feature locations. Finally the reconstructed features are fed to the last convolutional layer to restore the denoised image. By using the long residual connection, our network learns only the noise component.

In addition to the network architecture, the loss function is crucial to the performance. Several loss functions, such as  $L_2$  [35–37],  $L_1$  [4], perceptual loss [15], and asymmetric loss [12], have been used in denoising tasks. In general,  $L_1$  and  $L_2$  are the two losses used most commonly in previous works. The  $L_2$  loss has good confidence for Gaussian noise, whereas the  $L_1$  loss has better tolerance for outliers. In our experiment, we use the  $L_2$  loss for training on synthetic image datasets and the  $L_1$  loss for training on real-image noise datasets.

The following subsections focus on the RSAB and context block to provide more detailed explanations.

### 3.1 Residual spatial-adaptive block

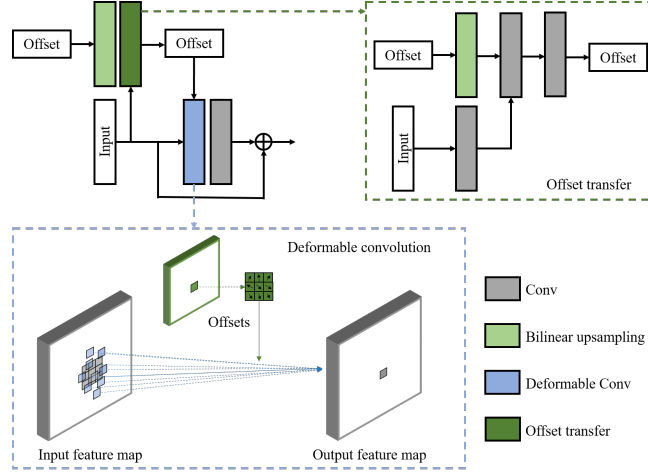
In this section, we first introduce the deformable convolution [10, 40] and then propose our RSAB in detail.

Let  $x(p)$  denote the features at location  $p$  from the input feature map  $x$ . Then, for a traditional convolution operation, the corresponding output features  $y(p)$  can be obtained by

$$y(p) = \sum_{p_i \in N(p)} w_i \cdot x(p_i), \quad (2)$$

where  $N(p)$  denotes the neighborhood of location  $p$ , whose size is equal to the size of the convolutional kernel.  $w_i$  denotes the weight of location  $p$  in the convolutional kernel, and  $p_i$  denotes the location in  $N(p)$ . The traditional convolution





**Fig. 2. The architecture of the residual spatial-adaptive block (RSAB).** The offset transfer component is shown in the green dashed box. The deformable convolution architecture is shown in the blue dashed box.

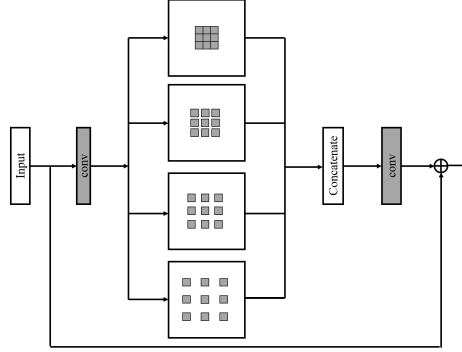
operation strictly takes the feature of the fixed location around  $p$  when calculating the output feature. Thus, some unwanted or unrelated features can interfere with the output calculation. For example, when the current location is near the edge, the distinct features located outside the object are introduced for weighting, which may smooth the edges and destroy the texture. For the denoising task, we would prefer that only the related or similar features are used for noise removal, similar to the self-similarity weighted denoising methods [5, 8, 9].

Therefore, we introduce deformable convolution [10, 40] to adapt to spatial texture changes. In contrast to traditional convolutional layers, deformable convolution can change the shapes of convolutional kernels. It first learns an offset map for every location and applies the resulting offset map to the feature map, which resamples the corresponding features for weighting. Here, we use modulated deformable convolution [40], which provides another dimension of freedom to adjust its spatial support regions,

$$y(p) = \sum_{p_i \in N(p)} w_i \cdot x(p_i + \Delta p_i) \cdot \Delta m_i, \quad (3)$$

where  $\Delta p_i$  is the learnable offset for location  $p_i$ , and  $\Delta m_i$  is the learnable modulation scalar, which lies in the range  $[0, 1]$ . It reflects the degree of correlation between the sampled features  $x(p_i)$  and the features in the current location. Thus, the modulated deformable convolution can modulate the input feature amplitudes to further adjust the spatial support regions. Both  $\Delta p$  and  $\Delta m$  are obtained from the previous features.

In each RSAB, we first fuse the extracted features and the reconstructed features from the previous scale as the input. The RSAB is constructed by a



**Fig. 3. The architecture of the context block.** Instead of downsampling operations, multisize dilated convolutions are implemented to extract different receptive-field features.

modulated deformable convolution followed by a traditional convolution with a short skip connection. Similar to ResBlock, we implement local residual learning to enhance the information flow and improve representation ability of the network. However, unlike ResBlock, we replace the first convolution with modulated deformable convolution and use leaky ReLU as our activation function. Hence, the RSAB can be formulated as

$$F_{RSAB}(x) = F_{cn}(F_{act}(F_{dcn}(x))) + x, \quad (4)$$

where  $F_{dcn}$  and  $F_{cn}$  denote the modulated deformable convolution and traditional convolution respectively.  $F_{act}$  is the activation function (leaky ReLU here). The architecture of RSAB is shown in Fig. 2.

Furthermore, to better estimate the offsets from coarse to fine, we transfer the last-scale offsets  $\Delta p^{s-1}$  and modulation scalars  $\Delta m^{s-1}$  to the current scale  $s$ , and then use both  $\{\Delta p^{s-1}, \Delta m^{s-1}\}$  and the input features  $x^s$  to estimate  $\{\Delta p^s, \Delta m^s\}$ . Given the small-scale offsets as the initial reference, the related features can be located more accurately on the large scale. The offset transfer can be formulated as follows:

$$\{\Delta p^s, \Delta m^s\} = F_{offset}(x, F_{up}(\{\Delta p^{s-1}, \Delta m^{s-1}\})), \quad (5)$$

where  $F_{offset}$  and  $F_{up}$  denote the offset transfer and upsampling functions, separately, as shown in Fig. 2. The offset transfer function involves several convolutions, and it extracts features from input and fuses them with the previous offsets to estimate the offsets in the current scale. The upsampling function magnifies both the size and value of the previous offset maps. In our experiment, bilinear interpolation is adopted to upsample the offsets and modulation scalars.

### 3.2 Context block

Multiscale information is important for image denoising tasks; therefore, the downsampling operation is often adopted in networks. However, when the spatial

resolution is too small, the image structures are destroyed, and information is lost, which is not conducive to reconstructing the features.

To increase the receptive field and capture multiscale information without further reducing the spatial resolution, we introduce a context block into the minimum scale between the encoder and decoder. Context blocks have been successfully used in image segments [6] and deblurring tasks [38]. In contrast to spatial pyramid pooling [13], the context block uses several dilated convolutions with different dilation rates rather than downsampling. It can expand the receptive field without increasing the number of parameters or damaging the structures. Then, the features extracted from the different receptive fields are fused to estimate the output (as shown in Fig. 3). It is beneficial to estimate offsets from a larger receptive field.

In our experiment, we remove the batch normalization layer and only use four dilation rates which are set to 1, 2, 3, and 4. To further simplify the operation and reduce the running time, we first use a  $1 \times 1$  convolution to compress the feature channels. The compression ratio is set to 4 in our experiments. In the fusion setup, we use a  $1 \times 1$  convolution to output the fusion features whose channels are equal to the original input features. Similarly, a local skip connection between the input and output features is applied to prevent information blocking.

### 3.3 Implementation

In the proposed model, we use four scales for the encoder-decoder architecture, and the number of channels for each scale is set to 32, 64, 128, and 256. The kernel size of the first and last convolutional layers is set to  $1 \times 1$ , and the final output is set to 1 or 3 channels depending on the input. Moreover, we use  $2 \times 2$  filters for up/down-convolutional layers, and all the other convolutional layers have a kernel size of  $3 \times 3$ .

## 4 Experiments

In this section, we demonstrate the effectiveness of our model on both synthetic datasets and real noisy datasets. We adopt DIV2K [21] which contains 800 images with 2K resolution, and add different levels of noise to synthetic noise datasets. For real noisy images, we use the SIDD [1], RENOIR [3] and Poly [31] datasets. We randomly rotate and flip the images horizontally and vertically for data augmentation. In each training batch, we use 16 patches with size of  $128 \times 128$  as inputs. We train our model using the ADAM [16] optimizer with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , and  $\epsilon = 10^{-8}$ . The initial learning rate is set to  $10^{-4}$  and then halved after  $3 \times 10^5$  iterations. Our model is implemented in the PyTorch framework [24] with an Nvidia GeForce RTX 1080Ti. In addition, we employ PSNR and SSIM [29] to evaluate the results.

#### 4.1 Ablation study

We perform ablation study on the Kodak24 dataset with a noise sigma of 50. The results are shown in Table 1.

**Table 1.** Ablation study of different components. PSNR values are based on Kodak24 ( $\sigma = 50$ )

RSAB	×	✓	×	✓	✓	✓
offset transfer	×	×	×	×	✓	✓
Context block	×	×	✓	✓	×	✓
PSNR	29.05	29.55	29.12	29.60	29.59	29.64

**Ablation on RSAB** RSAB is the crucial block in our network. Without it, the network will lose its ability to adapt to image content. When we replace RSAB with an original ResBlock, the performance decreases substantially, which demonstrates its effect.

**Ablation on the context block** The context block complements the down-sampling operations to capture larger field information. We can observe that the performance improves when the context block is introduced.

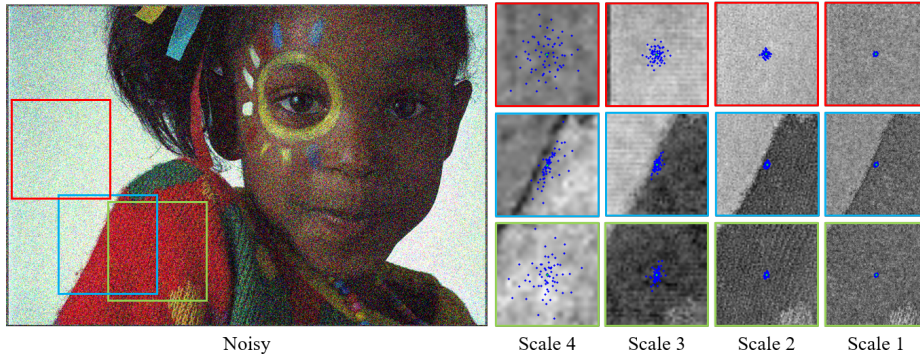
**Ablation on the offset transfer** We remove the offset transfer from coarse to fine and use only the features on the current scale to estimate the offsets for RSAB. This comparison validates the effectiveness of offset transfer.

#### 4.2 Analyses of the spatial adaptability

As discussed above, our network introduces the adaptability to spatial textures and edges. The RSABs can extract related features by change the sampling locations based on the image content. We visualize the learned kernel locations of the RSABs in Fig. 4. The visualization results show that in the smooth regions or the homogeneous textured regions, the convolution kernels are approximately uniformly distributed, while in the regions close to the edge, the shapes of the convolution kernels extend along the edge. Most of sampling points fall on the similar texture regions inside the object, which demonstrates that our network has indeed learned spatial adaptability. Moreover, as shown in Fig. 4, the RSAB can extract features from a larger receptive field at the coarse scale, while at the fine scale, the sampled features are located in the neighborhood of the current point. The multiscale structure enables the network to obtain the information of different receptive fields for image reconstruction.

#### 4.3 Comparisons

In this subsection, we compare our algorithm with the state-of-the-art denoising methods. For a fair comparison, all the compared methods employ the default



**Fig. 4. Visualization of the learned kernels.** The scales from 4 to 1 are in order from coarse to fine.

settings provided by the corresponding authors. We first make a comparison on the synthetic noise datasets, since many methods provide only Gaussian noise removal results. Then, we report the denoising results on the real noisy datasets using the state-of-the-art real noise removal methods.

**Synthetic noisy images** In the comparisons of synthetic noisy images, we use BSD68 and Kodak24 as our test datasets. These datasets include both color and grayscale images for testing. We add AWGN at different noise levels to the clean images. We choose BM3D [9] and CBM3D [8] as representatives of the classical traditional methods as well as some CNN-based methods, including DnCNN [35], MemNet [27], FFDNet [36], RNAN [37], and RIDNet [4], for the comparisons.

Tables 2 shows the average results of PSNR on grayscale images with three different noise levels. Our SADNet achieves the highest values on most of the datasets and tested noise levels. Note that although RNAN can achieve comparable evaluations to our method on partial low noise levels, it requires more parameters and a larger computational overhead. Next, Table 3 reports the quantitative results on color images. We replace the input and output channels from one to three as the other methods. Our SADNet outperforms the state-of-the-art methods on all the datasets with all tested noise levels. In addition, we can observe that our method shows more improvement at higher noise levels, which demonstrates its effectiveness for heavy noise removal.

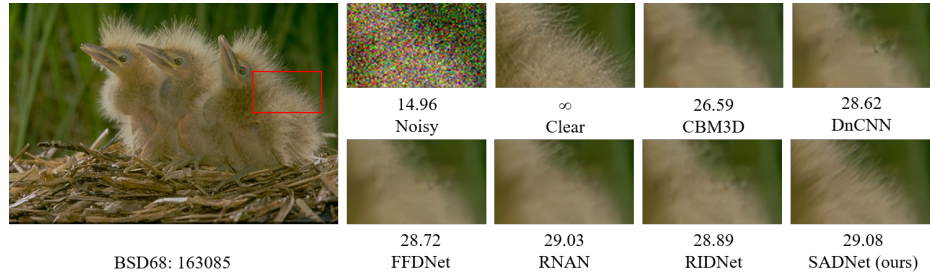
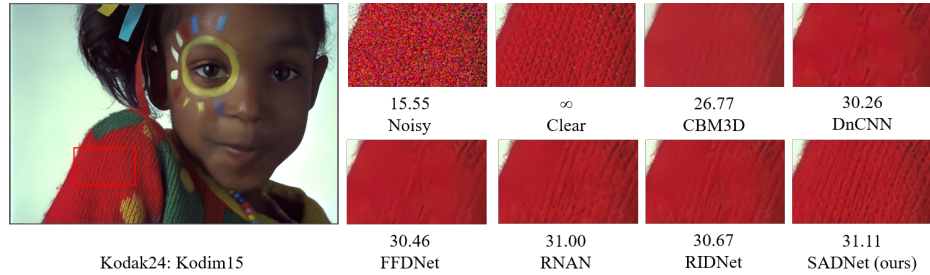
The visual comparisons are shown in Fig. 5 and Fig. 6. We present some challenging examples from BSD68 and Kodak24. In particular, the birds' feathers and the clothing textures are difficult to separate from heavy noise. The compared methods tend to remove the details along with the noise, resulting in oversmoothing artifacts. Many of the textured areas are heavily smeared in the denoising results. Due to its adaptivity to the image content, our method can restore the vivid textures from noisy images without introducing other artifacts.

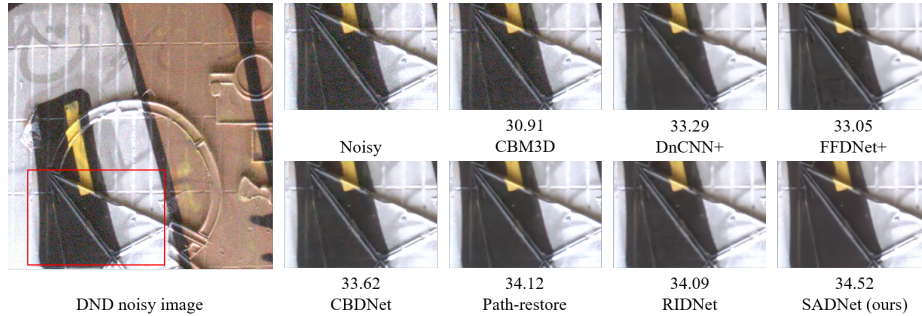
**Table 2.** Average PSNR(dB) results on synthetic **grayscale** noisy images

Dataset	$\sigma$	BM3D	DnCNN	MemNet	FFDNet	RNAN	RIDNet	SADNet (ours)
BSD68	30	27.76	28.36	28.43	28.39	28.61	28.54	<b>28.61</b>
	50	25.62	26.23	26.35	26.29	26.48	26.40	<b>26.51</b>
	70	24.44	24.90	25.09	25.04	25.18	25.12	<b>25.24</b>
Kodak24	30	29.13	29.62	29.72	29.70	<b>30.04</b>	29.90	30.00
	50	26.99	27.51	27.68	27.63	27.93	27.79	<b>27.96</b>
	70	25.73	26.08	26.42	26.34	26.60	26.51	<b>26.72</b>

**Table 3.** Average PSNR(dB) results on synthetic **color** noisy images

Dataset	$\sigma$	CBM3D	DnCNN	MemNet	FFDNet	RNAN	RIDNet	SADNet (ours)
BSD68	30	29.73	30.40	28.39	30.31	30.63	30.47	<b>30.64</b>
	50	27.38	28.01	26.33	27.96	28.27	28.12	<b>28.32</b>
	70	26.00	26.56	25.08	26.53	26.83	26.69	<b>26.93</b>
Kodak24	30	30.89	31.39	29.67	31.39	31.86	31.64	<b>31.86</b>
	50	28.63	29.16	27.65	29.10	29.58	29.25	<b>29.64</b>
	70	27.27	27.64	26.40	27.68	28.16	27.94	<b>28.28</b>

**Fig. 5.** Synthetic image denoising results on BSD68 with noise level  $\sigma = 50$ .**Fig. 6.** Synthetic image denoising results on Kodak24 with noise level  $\sigma = 50$ .



**Fig. 7.** Real image denoising results from the DnD dataset.

**Real noisy images** To conduct comparisons on real noisy images, we choose DnD [25], SIDD [1] and Nam [22] as test datasets. **DnD** contains 50 real noisy images and their corresponding clear images. One thousand patches with a size of  $512 \times 512$  are extracted from the dataset by the providers for testing and comparison purposes. Since the ground truth images are not publicly available, we can obtain only the PSNR/SSIM results through the online submission system introduced by [25]. The validation dataset of **SIDD** is introduced for our evaluation, which contains 1280  $256 \times 256$  noisy-clean image pairs. **Nam** includes 15 large image pairs with JPEG compression for 11 scenes. We cropped the images into  $512 \times 512$  patches and selected 25 patches picked by CBDNet [12] for testing.

We train our model on the SIDD medium dataset and RENOIR for evaluation on the DnD and SIDD validation datasets. Then, we finetune our model on the Poly [31] for Nam, which improves the performance on the noisy images with JPEG compression. Furthermore, as comparisons, we choose the state-of-the-art methods whose validity has previously been demonstrated on real noisy images, including CBM3D [8], DnCNN [35], CBDNet [12], PD [39], and RIDNet [4].

**DnD** The quantitative results are listed in Table 4, which are obtained from the public DnD benchmark website. FFDNet+ is the improved version of FFDNet with a uniform noise level map manually selected by the providers. CDnCNN-B is the original DnCNN model for blind color denoising. DnCNN+ is finetuned on CDnCNN-B with the results of FFDNet+. SADNet (1248) is the modified version of our SADNet with 1, 2, 4, 8 dilation rates in the context block. Both non-blind and blind denoising methods are included for comparisons. CDnCNN-B cannot effectively generalize to real noisy images. The performances of non-blind denoising methods are limited due to the different distributions between AWGN and real-world noise. In contrast, our SADNet outperforms the state-of-the-art methods with respect to both PSNR and SSIM values. We further perform a visual comparison on denoised images from the DnD dataset, as shown in Fig. 7. The other methods corrode the edges with residual noise, while our method can effectively remove the noise from the smooth region and maintain clear edges.

**Table 4.** Quantitative results on DnD sRGB images

Method	Blind/Non-blind	PSNR	SSIM
CDnCNN-B	Blind	32.43	0.7900
TNRD	Non-blind	33.65	0.8306
BM3D	Non-blind	34.51	0.8507
WNNM	Non-blind	34.67	0.8646
MCWNNM	Non-blind	37.38	0.9294
FFDNet+	Non-blind	37.61	0.9415
DnCNN+	Non-blind	37.90	0.9430
CBDNet	Blind	38.06	0.9421
N3Net	Blind	38.32	0.9384
PD	Blind	38.40	0.9452
Path-Restore	Blind	39.00	0.9542
RIDNet	Blind	39.26	0.9528
SADNet (1248)	Blind	39.37	<b>0.9544</b>
SADNet (ours)	Blind	<b>39.59</b>	0.9523

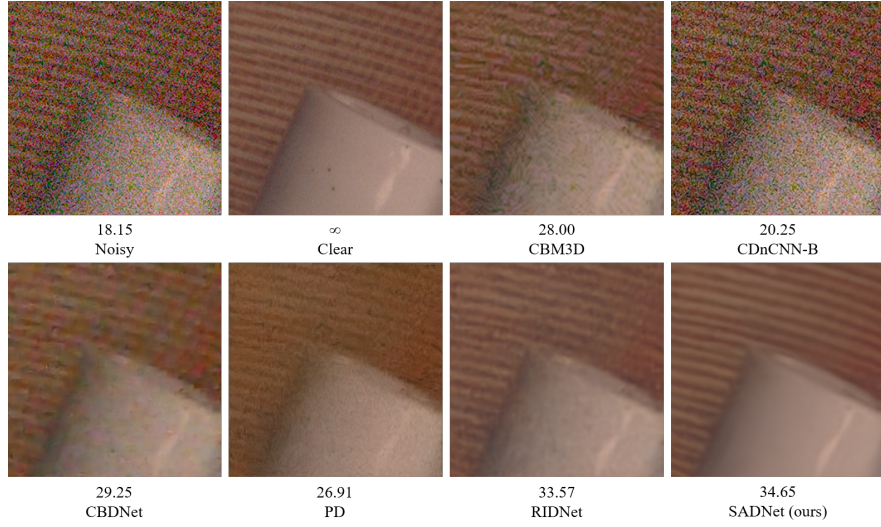
**Table 5.** Quantitative results on SIDD sRGB validation dataset

Method	CBM3D	CDnCNN-B	CBDNet	PD	RIDNET	SADNet (ours)
Blind/Non-blind	Non-blind	Blind	Blind	Blind	Blind	Blind
PSNR	30.88	26.21	30.78	32.94	38.71	<b>39.46</b>

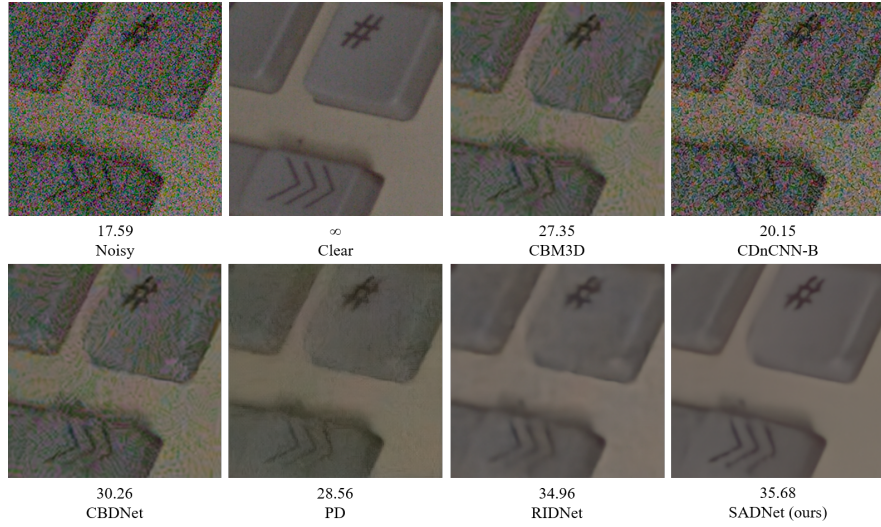
**SIDD** The images in the SIDD dataset are captured by smartphones, and some noisy images have high noise levels. We employ 1,280 validation images for quantitative comparisons as listed in Table 5. The results demonstrates that our method achieves significant improvements over the other tested methods. For visual comparisons, we choose two challenging examples from the denoised results. The first scene has rich textures, while the second scene has prominent structures. As shown in Fig. 8 and Fig. 9, CDnCNN-B and CBDNet fail at noise removal. CBM3D results in pseudo artifacts, and PD and RIDNet destroy the textures. In contrast, our network recovers textures and structures that are closer to the ground truth.

**Nam** The JPEG compression makes the noise more stubborn on the Nam dataset. For a fair comparison, we use the patches chosen by CBDNet [12] for evaluation. Furthermore, CBDNet\* [12] is introduced for comparison, which was retrained on JPEG compressed datasets by its providers. We report the average PSNR and SSIM values for Nam in Table 6. With respect to PSNR, Our SADNet achieves 1.88, 1.83 and 1.61 dB gains over RIDNet, PD, and CBDNet\*. Similarly, our SSIM values exceed those of all the other methods in the comparison. In the visual comparison shown in Fig. 10, our method again obtains the best result for texture restoration and noise removal.





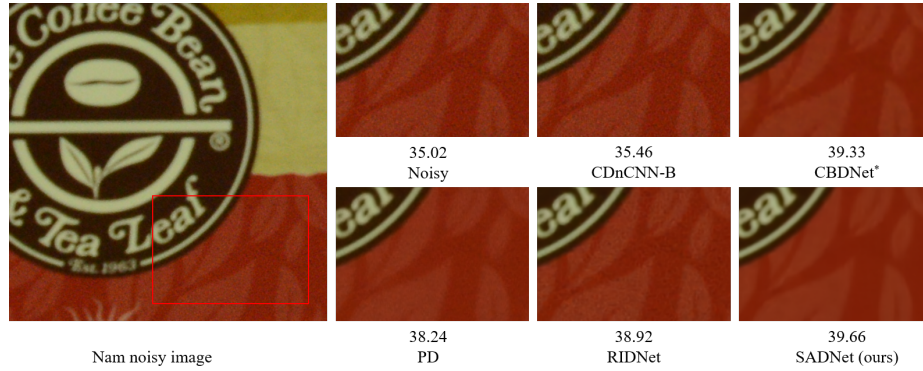
**Fig. 8.** A real image denoising example from the SIDD dataset.



**Fig. 9.** Another real image denoising example from the SIDD dataset.

**Table 6.** Quantitative results on Nam dataset with JPEG compression

Method	CBM3D	CDnCNN-B	CBDNet*	PD	RIDNET	SADNet (ours)
Blind/Non-blind	Non-blind	Blind	Blind	Blind	Blind	Blind
PSNR	39.84	37.49	41.31	41.09	41.04	<b>42.92</b>
SSIM	0.9657	0.9272	0.9784	0.9780	0.9814	<b>0.9839</b>



**Fig. 10.** Real image denoising results from the Nam dataset with JPEG compression.

**Table 7.** Parameters and time comparisons on  $480 \times 320$  color images

Method	DnCNN	MemNet	RNAN	RIDNet	SADNET (ours)
Parameters	558k	2,908k	8,960k	1,499k	4,321k
FLOPs	86.1G	449.2G	1163.5G	230.0G	50.1G
times (ms)	21.3	154.2	1072.2	84.4	26.7

**Parameters and running times** To compare the running times, we test different methods when denoising  $480 \times 320$  color images. Note that the running time may depend on the test platform and code; thus, we also provide the number of floating point operations (FLOPs). All the methods are implemented in PyTorch. As shown in Table 7, although SADNet has high parameter numbers, its FLOPs are minimal, and its running time is short due to the multiple down-sampling operations. Because most of the operations run on smaller-scale feature maps, our model performs faster than many others with fewer parameters.

## 5 Conclusion

In this paper, we propose a spatial-adaptive denoising network for effective noise removal. The network is built by multiscale residual spatial-adaptive blocks, which sample relevant features for weighting based on the content and textures of images. We further introduce a context block to capture multiscale information and implement offset transfer to more accurately estimate the sampling locations. We find that the introduction of spatially adaptive capability can restore richer details in complex scenes under heavy noise. The proposed SADNet achieves state-of-the-art performances on both synthetic and real noisy images and has a moderate running time.

## References

1. Abdelhamed, A., Lin, S., Brown, M.S.: A high-quality denoising dataset for smart-phone cameras. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1692–1700 (2018)
2. Aharon, M., Elad, M., Bruckstein, A.: K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing* **54**(11), 4311–4322 (2006)
3. Anaya, J., Barbu, A.: Renoir—a dataset for real low-light image noise reduction. *Journal of Visual Communication and Image Representation* **51**, 144–154 (2018)
4. Anwar, S., Barnes, N.: Real image denoising with feature attention. arXiv preprint arXiv:1904.07396 (2019)
5. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05). vol. 2, pp. 60–65. IEEE (2005)
6. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
7. Chen, Y., Pock, T.: Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE transactions on pattern analysis and machine intelligence* **39**(6), 1256–1272 (2016)
8. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: 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, pp. I–313. IEEE (2007)
9. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing* **16**(8), 2080–2095 (2007)
10. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 764–773 (2017)
11. Gu, S., Zhang, L., Zuo, W., Feng, X.: Weighted nuclear norm minimization with application to image denoising. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2862–2869 (2014)
12. Guo, S., Yan, Z., Zhang, K., Zuo, W., Zhang, L.: Toward convolutional blind denoising of real photographs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1712–1722 (2019)
13. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence* **37**(9), 1904–1916 (2015)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
15. Jiao, J., Tu, W.C., He, S., Lau, R.W.H.: Formresnet: Formatted residual learning for image restoration. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2017)
16. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
17. Lefkimmiatis, S.: Non-local color image denoising with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3587–3596 (2017)

18. Liu, D., Wen, B., Fan, Y., Loy, C.C., Huang, T.S.: Non-local recurrent network for image restoration. In: *Advances in Neural Information Processing Systems*. pp. 1673–1682 (2018)
19. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: *Proc. icml*. vol. 30, p. 3 (2013)
20. Mairal, J., Bach, F.R., Ponce, J., Sapiro, G., Zisserman, A.: Non-local sparse models for image restoration. In: *ICCV*. vol. 29, pp. 54–62. Citeseer (2009)
21. Martin, D., Fowlkes, C., Tal, D., Malik, J., et al.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *Iccv Vancouver*: (2001)
22. Nam, S., Hwang, Y., Matsushita, Y., Joo Kim, S.: A holistic approach to cross-channel image noise modeling and its application to image denoising. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 1683–1691 (2016)
23. Osher, S., Burger, M., Goldfarb, D., Xu, J., Yin, W.: An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation* **4**(2), 460–489 (2005)
24. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing Systems*. pp. 8024–8035 (2019)
25. Plotz, T., Roth, S.: Benchmarking denoising algorithms with real photographs. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 1586–1595 (2017)
26. Plötz, T., Roth, S.: Neural nearest neighbors networks. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2018)
27. Tai, Y., Yang, J., Liu, X., Xu, C.: Memnet: A persistent memory network for image restoration. In: *Proceedings of the IEEE international conference on computer vision*. pp. 4539–4547 (2017)
28. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 7794–7803 (2018)
29. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* **13**(4), 600–612 (2004)
30. Xu, J., Osher, S.: Iterative regularization and nonlinear inverse scale space applied to wavelet-based denoising. *IEEE Transactions on Image Processing* **16**(2), 534–544 (2007)
31. Xu, J., Li, H., Liang, Z., Zhang, D., Zhang, L.: Real-world noisy image denoising: A new benchmark. *arXiv preprint arXiv:1804.02603* (2018)
32. Xu, J., Zhang, L., Zhang, D.: External prior guided internal prior learning for real-world noisy image denoising. *IEEE Transactions on Image Processing* **27**(6), 2996–3010 (2018)
33. Xu, J., Zhang, L., Zhang, D.: A trilateral weighted sparse coding scheme for real-world image denoising. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. pp. 20–36 (2018)
34. Xu, J., Zhang, L., Zhang, D., Feng, X.: Multi-channel weighted nuclear norm minimization for real color image denoising. In: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 1096–1104 (2017)

35. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing* **26**(7), 3142–3155 (2017)
36. Zhang, K., Zuo, W., Zhang, L.: Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing* **27**(9), 4608–4622 (2018)
37. Zhang, Y., Li, K., Li, K., Zhong, B., Fu, Y.: Residual non-local attention networks for image restoration. *arXiv preprint arXiv:1903.10082* (2019)
38. Zhou, S., Zhang, J., Zuo, W., Xie, H., Pan, J., Ren, J.S.: Davanet: Stereo deblurring with view aggregation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 10996–11005 (2019)
39. Zhou, Y., Jiao, J., Huang, H., Wang, Y., Wang, J., Shi, H., Huang, T.: When awgn-based denoiser meets real noises. *arXiv preprint arXiv:1904.03485* (2019)
40. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 9308–9316 (2019)
41. Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: *2011 International Conference on Computer Vision*. pp. 479–486. IEEE (2011)