Pseudo 3D Auto-Correlation Network for Real Image Denoising

Xiaowan Hu[1,2], Ruijun Ma[3], Zhihong Liu[1], Yuanhao Cai[1], Xiaole Zhao[4], Yulun Zhang[5], Haoqian Wang[1,2,*]
[1] The Shenzhen International Graduate School, Tsinghua University, China
[2] The Shenzhen Institute of Future Media Technology, Shenzhen 518071, China
[3] University of Macau, China [4] Southwest Jiaotong University, China [5] Northeastern University, US

Abstract

The extraction of auto-correlation in images has shown great potential in deep learning networks, such as the self-attention mechanism in the channel domain and the self-similarity mechanism in the spatial domain. However, the realization of the above mechanisms mostly requires complicated module stacking and a large number of convolution calculations, which inevitably increases model complexity and memory cost. Therefore, we propose a pseudo 3D auto-correlation network (P3AN) to explore a more efficient way of capturing contextual information in image denoising. On the one hand, P3AN uses fast 1D convolution instead of dense connections to realize criss-cross interaction, which requires less computational resources. On the other hand, the operation does not change the feature size and makes it easy to expand. It means that only a simple adaptive fusion is needed to obtain contextual information that includes both the channel domain and the spatial domain. Our method built a pseudo 3D auto-correlation attention block through 1D convolutions and a lightweight 2D structure for more discriminative features. Extensive experiments have been conducted on three synthetic and four real noisy datasets. According to quantitative metrics and visual quality evaluation, the P3AN shows great superiority and surpasses state-of-the-art image denoising methods.

1. Introduction

As a low-level vision task, image denoising aims to recover the underlying clean image from an observed noisy one, which is a fundamental step for various high-level vision and image analysis applications [42]. In recent years, many advanced methods have achieved remarkable progress in removing synthesized additive white Gaussian noise. However, the noise in real images often has a complicated generation process in CCD or CMOS camera systems. Affected by different devices and image signal processing (ISP) pipelines within the camera, the denoising algorithms based on synthetic data are inherently difficult to simulate and remove irregular real noise accurately [1, 36]. For blind image denoising, the low-quality noise images without specific noise statistical priors become the only source of guidance. So it is particularly important to capture auto-correlation prior information from the input.

The image auto-correlation prior has been widely explored and played an essential role in many traditional noise reduction algorithms. In order to obtain more robust learning and presentation abilities, recent methods try to use deep learning methods to accumulate more useful and comprehensive prior knowledge. In recent years, auto-correlation features extracted by convolutional neural networks (CNN) can be divided into channel-wise and spatial-wise features, which are calculated by the channel-based model and space-based model, respectively. Similar to other high-level vision tasks, channel-based model can capture non-linear cross-channel information about channels of interest, which is defined as a self-attention mechanism [3, 12, 22, 29, 34, 55]. Through features are aggregated and recalibrated in different ways, self-attention can capture cross-channel interaction. Lightweight attention aggregation is conducive to assign sophisticated channel-wise dependencies efficiently.

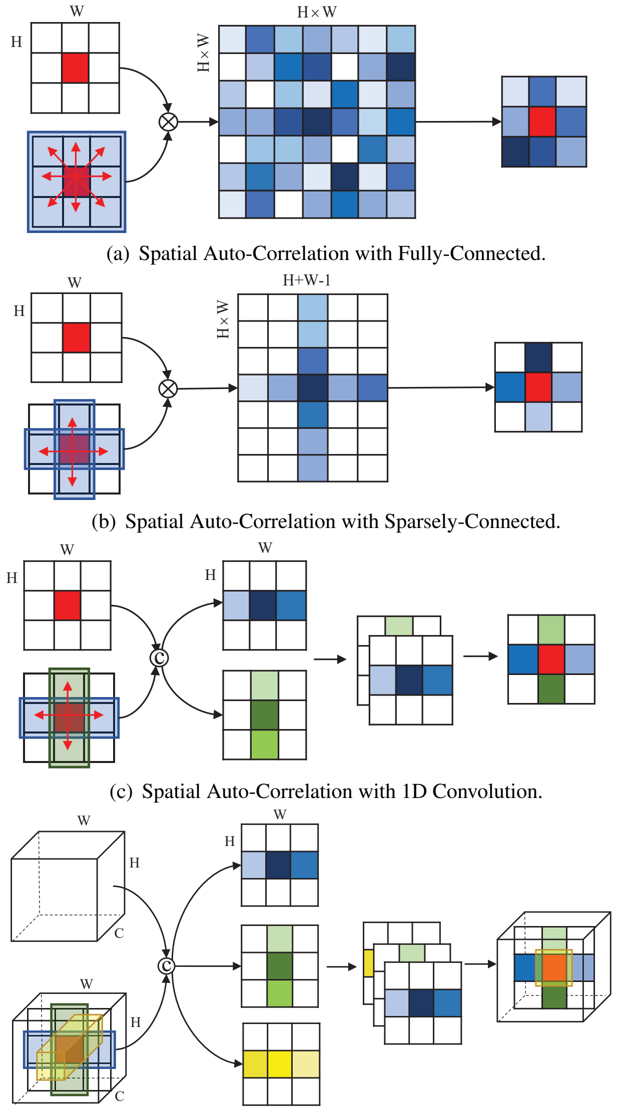
The space-based model revolves around the spatial self-similarity that has been proven as a powerful feature of natural images [30, 35, 44]. A common way to obtain the global auto-correlation in the spatial domain is using a non-local block [56]. As shown in Fig. 1(a), it calculates the response as the weighted sum of all pixels. The spatial auto-correlation is hidden in the pixel-wise long-range contextual information. The dense connection generates huge attention maps with high complexity. Huang et al. proposed criss-cross attention with sparsely-connected graphs, as shown in Fig. 1(b) [24]. They only extract contextual information in its horizontal and vertical direction to narrow the search. But they still need to traverse each pixel to obtain full-image auto-correlation for each position, which does not fully play the role of sparse connections. Although the idea of utilizing auto-correlation prior of images has achieved great successes in various restoration tasks, most CNNs built with this still suffer from a heavier computational burden.

[*] Haoqian Wang is the corresponding author.

Therefore, the exploration of image auto-correlation encounters a bottleneck in balancing model performance and computational complexity. On the one hand, most existing methods are still dedicated to building more sophisticated auto-correlation modules to achieve better performance, which further increases application difficulty in the real world. On the other hand, the feature maps are usually 3-dimensional, so a complete auto-correlation feature should have contextual information from both channel and spatial directions. It is not feasible to introduce 3D convolution accompanying the explosive growth of parameters. There are currently some methods that combine the channel self-attention and the spatial self-similarity through a series or parallel structure [15, 46]. This step-by-step operation increases the complexity of the model and destroys the continuous correlation in the local area.

To address the above problems efficiently, we proposed a pseudo 3D auto-correlation network (P3AN) to simulate 3D convolution with a 2D structure and integrate channel and spatial auto-correlation into a unified module. As shown in Fig. 1(c), to avoid the high computational complexity caused by huge attention maps, we use fast 1D convolution instead of a non-local densely-connected layer. The feature correlations in a specific horizontal or vertical direction are captured and merged to minimize the correlation map. At the same time, parameter sharing is introduced into convolution operations to save GPU memory. Our operation does not change the size of the feature map, which makes cross-directional fusion possible. It can be seen from Fig. 1(d) that only simple channel integration and adaptive fusion are needed to integrate the three directions of horizontal, vertical, and channel. Through consecutive pseudo 3D auto-correlation blocks (P3AB) stack and skip connection, each location can collect contextual information by considering all local pixels in 3D space. Our method realizes a lightweight pseudo 3D interaction network that can be learned end-to-end with low time and space complexity. The main innovative contributions are as follows:

- We propose a novel spatial auto-correlation module with fast 1D convolution. The strategy of direction independence and parameter sharing can effectively reduce time and space complexity while capturing contextual information from full image dependencies.
- The operation in one-dimensional space avoids dimensionality reduction, which is easy to expand. We design a lightweight 2D structure to adaptively fuse the correlation features in the three directions of horizontal, vertical, and channel, and get more discriminative features for real image denoising.
- We evaluate quantitative indicators and visual quality on synthetic and real noise datasets. Our proposed network has lower model complexity and higher performance than state-of-the-art image denoising methods.



(a) Spatial Auto-Correlation with Fully-Connected.

(b) Spatial Auto-Correlation with Sparsely-Connected.

(c) Spatial Auto-Correlation with 1D Convolution.

(d) Pseudo 3D Auto-Correlation with 1D Convolution.

Figure 1. Diagram of four methods to aggregate auto-correlation. Each position (e.g., red) can collect information from other pixels

2. Related Works

2.1. Image Denoising

Most traditional denoising methods [6, 7, 11, 14] aimed to remove synthetic additive white Gaussian noise. Because it conforms to a specific prior distribution, the removal of synthetic noise is usually modeled by sparsity or self-similarity. The use of deep learning for image denoising has been extensively researched recently. CNN applies a powerful learning model to eliminate noise and has obtained a significant performance improvement [4, 7, 20, 28, 32, 52, 53, 57]. However, the real camera noise is heavily transformed by the camera ISP, the flexible network structures to extract discriminative features needs to be further improved. CBDNet [19] trains a blind denoising model through two steps of noise estimation and non-blind denoising. RIDNet [3] se-

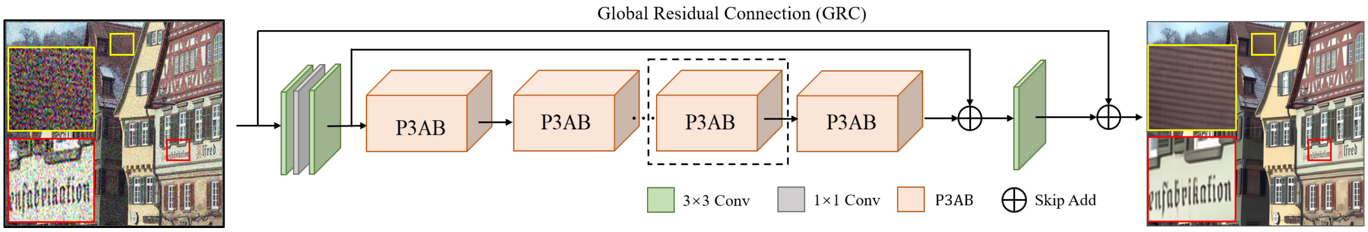


Figure 2. The architecture of the proposed P3AN. The P3ABs captures the pseudo 3D auto-correlation by consecutive feature interaction.

lectively learns distinctive distinguishing features through an attention mechanism. The advanced VDN proposed by Yue et al. [50] uses variational inference techniques to estimate the noise distribution within the Bayesian framework. But the sensor noise model ignores the signal auto-correlation, which makes it hard to estimate the real noise in a spatio-chromatically correlated form. AINDNet [26] uses a migration learning strategy to propose a denoising structure with a strong generalization. MIRNet [51] learns rich spatial context features through parallel multi-resolution convolution streams and maintains high-resolution details. Although such deep networks have achieved good performance, recent explorations focus on building more complex structures and deeper networks. The hard training process increases the risk of saturation of denoising performance.

2.2. Auto-Correlation Aggregation

Auto-correlation features usually consist of self-attention and self-similarity, representing the contextual information of channel direction and spatial direction, respectively. SE-Net [22] proposed by Hu et al. designed an effective mechanism to learn channel attention. Subsequently, GE [21]used deep convolution [9] to aggregate features to explore spatial expansion. GSoP [17] introduced second-order pooling, and SAN [13] used covariance pooling to obtain second-order attention, both of which achieved more effective feature aggregation through high-order statistics. The self-similar prior provides a powerful self-prediction ability for natural image restoration [5, 18, 58]. Wang et al. [44] proposed a non-local attention module for deep CNN, which calculates long-range semantic relevance by assigning weights to elements in all locations. Recent methods including NLRN [31], RNAN [56], and SAN [13] incorporate non-local operations in their networks for image restoration. For reducing the computational complexity caused by dense global convolution, Huang et al. proposed criss-cross attention [25], which adopts a criss-cross approach to obtain context information. To obtain more complete correlation features, CBAM [46] and scSE [40] use 2D convolution to calculate spatial attention and then combine it with a independent channel attention. Dual Attention Network (DANet) [16] also considers non-local channel and spatial attention. Due to their high model complexity, most auto-correlation modules are only used in single or several convolutional blocks. It is important to learn the cross-direction correlation with low model complexity.

3. Proposed Method

3.1. Network Architecture

As shown in Fig. 2, for the noise image $\mathbf{x}$, we use three convolution layers to extract shallow features. The size of the convolution kernel is 3×3, 1×1 and 3×3 respectively. Define $F_S(\cdot)$ as the corresponding function, the shallow features $\mathbf{x}_0$ extracted in the first stage can be expressed as:

$$\mathbf{x}_0 = F_S(\mathbf{x}). \tag{1}$$

Next,the auto-correlation feature learning process consists of several stacked P3ABs and a skip connection. We define the function corresponding to the P3AB as $F_P(\cdot)$. Assume the number of P3AB in the entire network is n, then the output of the i-th block of the network is

$$\mathbf{x}_i = F_P^i(\mathbf{x}_{i-1}),\ i = 1,2,...,n, \tag{2}$$

where $F_P^i(\cdot)$ corresponds to the i-th P3AB. The implementation details of $F_P^i(\cdot)$ will be explained in Section 3.2. The input of the first P3AB is $\mathbf{x}_0$. This process is executed iteratively, and the output of the last P3AB is as follows:

$$\mathbf{x}_n = F_P^n(\mathbf{x}_{n-1}) = F_P^n(F_P^{n-1}(\cdots(F_P^1(\mathbf{x}_0))\cdots)). \tag{3}$$

Then we establish a skip connection between the shallow and the deep features to facilitate the cross-layer flow:

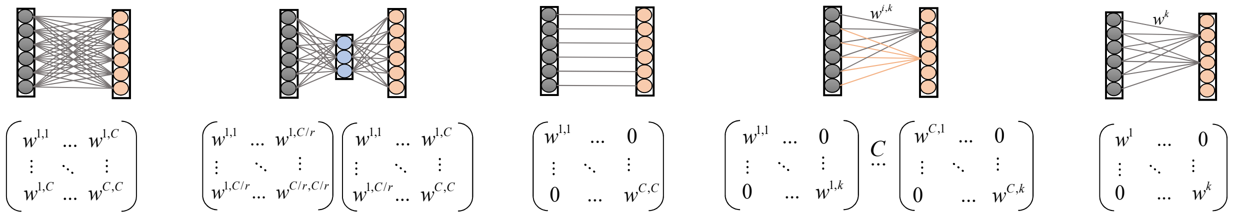
$$\mathbf{x}_f = \mathbf{x}_n + \mathbf{x}_0. \tag{4}$$

In image reconstruction, the original image is undoubtedly highly similar to the high-quality image, where there is no noise. The fact indicates that the two images have a lot of shared information, so we introduced the global residual connection (GRC) as a shortcut map to learn the residual information between the original input x and output denoised image y. Here we use a 3×3 convolutional layer defined as F_{C3} to adjust the fused features adaptively. Finally, the reconstructed image can be obtained as follows:

$$y = F_{C3}(\mathbf{x}_f) + \mathbf{x}. \tag{5}$$

Multiple skip connections form a multi-level residual mechanism. Cross-layer information exchange between layers that are far apart can help the model to retain more prior information in the noisy image. The multi-level residual learning can stabilize training and improve performance.

We choose the same $L1$ loss function used in the previous network training methods for a fair comparison. The optimization goal is making the denoised images as close as possible to the corresponding real clean images. Given a training set $\{x_{LQ}^j, y_{HQ}^j\}$ which contains N pairs of images, we minimize the $L1$ reconstruction loss as:



(a) Fully Connected (b) Channel Reduction (c) Channel Mapping (d) Locally Connected Convolution (e) 1D Fast Convolution

Figure 3. Different feature correlation remapping structures and their parameter matrices. The C is the number of elements and w is learnable weights. (a) The fully connected convolution layer with densely-connected weights. (b) The channel reduction structure with two fully connected layers. The r means the reduction factor. (c) The channel mapping with independent weights. (d) The locally connected layer with C convolutions and k is the size of kernels. (e) The 1D fast convolution with all elements sharing the same learning parameters.

Methods	Auto-Correlation	Parameters	PSNR (dB)
Baseline	N/A	0	28.01
Fully Connected	$\sigma(\mathbf{We})$	C^2	30.34
Channel Reduction	$\sigma(f_{\{\mathbf{W_1},\mathbf{W_2}\}}(\mathbf{e}))$	$2\times C^2/r$	29.86
Channel Mapping	$\sigma(\mathbf{w}\odot\mathbf{e})$	C	29.64
Locally Connected	$\sigma(\sum_{j=1}^{k} w_i^j e_i^j)$	$k\times C$	31.58
1D Fast Conv (Ours)	$\sigma(\sum_{j=1}^{k} w^j e_i^j)$	k	**31.72**

Table 1. Comparison of various feature auto-correlation modules on Urban100 [23] with noise level $\sigma = 50$. Parameters indicates number of learnable weights; σ is a Sigmoid function; $\odot$ indicates element-wise product; k is kernel size of the local convolution.

$$\mathscr{L}_1(\Theta) = \frac{1}{N}\sum_{j=1}^{N} \|F_{P3AN}(x_{LQ}^j) - y_{HQ}^j\|, \tag{6}$$

where $F_{P3AN}(\cdot)$ represents our network, and Θ is a set of all the parameters that to be optimized. We provide more details about the P3AB in the next section.

3.2. Pseudo 3D Auto-Correlation Extraction

3.2.1 1D Fast Convolution

We compare the current popular structure of extracting feature auto-correlation in Fig. 3. Then we explained the effectiveness and efficiency of the 1D fast convolution used in terms of model complexity and parameter amount.

For the fully connected layer, as shown in Fig. 3(a), each node is connected to all the nodes in the previous layer and integrates the features extracted from the front. Due to its dense connection characteristics, the parameters of the fully connected layer are really large. In the process of extracting spatial self-similarity, every node is fully connected, so a lot of calculations are required. Given the aggregated feature $\mathbf{e} \in \mathbb{R}^{\mathrm{C}}$, auto-correlation weights can be learned by

$$\omega = \sigma(\mathbf{We}), \tag{7}$$

where σ is a Sigmoid function and $\mathbf{W}$ is a matrix with C^2 parameters. The long-range dependence calculate all pixels, so a computational burden is required for the global spatial self-similarity. Specifically, the size of the intermediate feature map $f \in \mathbb{R}^{\mathrm{H}\times\mathrm{W}}$ in Fig. 1(a) and Fig. 1(b) is $(\mathrm{H}\times\mathrm{W})\times(\mathrm{H}\times\mathrm{W})$ in non-local operations and $(\mathrm{H}\times\mathrm{W})\times(\mathrm{H}+\mathrm{W}-1)$ in the criss-cross attention operations, respectively.

Channel reduction in Fig. 3(b) has been widely adopted by currently popular attention mechanism. It usually uses two non-linear fully connected layers and a Sigmoid function to generate channel weights. To control the complexity of the model, the two fully connected layers reduce the dimension and capture non-linear cross-channel interaction. The weights of auto-correlation can be computed as

$$\omega = \sigma(f_{\{\mathbf{W_1},\mathbf{W_2}\}}(\mathbf{e}))). \tag{8}$$

After channel reduction, the sizes of $\mathbf{W_1}$ and $\mathbf{W_2}$ are set to $C\times(\frac{C}{r})$ and $(\frac{C}{r})\times C$. The number of parameters of the correlation matrix can be calculated as $2\times C^2/r$, which is less than full connection but still a large computational burden.

Channel mapping in Fig. 3(c) shows that the optimization of the weight of each channel is independent, so the amount of its parameter number is only C. But it just contains directly corresponding weight and does not learn the correlation between neighbors. Therefore, to ensure both efficiency and effectiveness, our work explores another method that can capture local unidirectional interaction, as shown in Fig. 3(d). Specifically, we use single-direction locally connected convolution to learn auto-correlation. This local connection does not use the global receptive field, so the feature map will keep the original size in dimension.

Compared with the methods of extracting spatial self-similarity and channel self-attention mentioned above, this local auto-correlation is only calculated by considering the interaction between e_i and its k neighbors. Each feature vector can be calculated separately. The formula is as follows:

$$\omega_i = \sigma(\sum_{j=1}^{k} w_i^j e_i^j), \tag{9}$$

where i depends on the size of the feature map. When the feature is a C-dimensional vector, the amount of parameters in the auto-correlation is $k\times C$, which is already less than all the above methods that consider contextual information.

In this paper, we explored a more efficient way to capture local contextual information. By sharing the same parameters to be learned, the Eq. 9 can be simplified to:

$$\omega_i = \sigma(\sum_{j=1}^{k} w^j e_i^j). \tag{10}$$

As shown in the Fig. 3(e), the cross-channel local connection is replaced with a lightweight structure based on 1D fast convolution. The number of sharing parameters of the lightweight local connection is incredibly small, and only k weights are required for one operation. We reduce the computational complexity from a quadratic to a linear scale in the input length and then to a constant. This is a very significant improvement in the efficiency of calculating the auto-correlation information in a single direction.

We compare various feature auto-correlation modules on Urban100 [23] with noise level $\sigma = 50$ in Tab. 1. We use the ResNet-50 [20] as the baseline and then add corresponding structures to compare model complexity and denoising performance. We can see that our method achieves better performance with much lower model complexity and fewer parameters and the local connection is the second best. It proves that efficiency and effectiveness can be guaranteed by capturing local interactions properly.

3.2.2 Pseudo 3D Auto-Correlation Block

For the elements in each direction of features, 1D fast convolution can collect information from all other locations without changing the feature size. Therefore, this aggregation method can be directly extended to 3D space to capture all-around context information. As shown in Fig. 4, we build a lightweight pseudo-3D auto-correlation block based on 1D fast convolution, which is defined as P3AB.

We use $\mathbf{x}_t \in \mathbb{R}^{\mathrm{H}\times\mathrm{W}\times\mathrm{C}}$ ($t \in [0, n]$) to denote the input feature map. For each position, we extract the relevance of all elements in the horizontal, vertical and channel directions, and the lengths are w, h, and c, respectively. We define the 1D convolution in Eq. 11 as the function $F_{1DC}(\cdot)$. The $[f_i^{\hat{h}}, f_j^{\hat{v}}, f_k^{\hat{c}}]$ represent the original feature vectors in different directions. Then the auto-correlations are extracted by

$$[f_i^{\hat{H}}, f_j^{\hat{V}}, f_m^{\hat{C}}] = F_{1DC}([f_i^{\hat{h}}, f_j^{\hat{v}}, f_m^{\hat{c}}]),\ f_{i,j,m}, \in x_t, \tag{11}$$

$[f_i^{\hat{H}}, f_j^{\hat{V}}, f_k^{\hat{C}}]$ capture the correlations without length changing. Feature vectors in different directions share convolution parameters independently. We traverse all positions:

$$\begin{aligned} f^{\hat{\mathbf{H}}} &= \{f_0^{\hat{H}}, f_1^{\hat{H}}, f_2^{\hat{H}}, \dots, f_{c\times h}^{\hat{H}}\}, \\ f^{\hat{\mathbf{V}}} &= \{f_0^{\hat{V}}, f_1^{\hat{V}}, f_2^{\hat{V}}, \dots, f_{c\times w}^{\hat{V}}\}, \\ f^{\hat{\mathbf{C}}} &= \{f_0^{\hat{C}}, f_1^{\hat{C}}, f_2^{\hat{C}}, \dots, f_{w\times h}^{\hat{C}}\}. \end{aligned} \tag{12}$$

As shown in Fig. 4, thanks to the good usability of 1D fast convolution, $f^{\hat{\mathbf{H}}}$, $f^{\hat{\mathbf{V}}}$, and $f^{\hat{\mathbf{C}}}$ have the same size, so it can be concatenated in the channel, and a feature map with channel 3c can be obtained. Then the adaptive feature fusion (AFF) is performed, which is defined as:

$$f_a = F_C^2(F_C^1([f^{\hat{\mathbf{H}}}, f^{\hat{\mathbf{V}}}, f^{\hat{\mathbf{C}}}])), \tag{13}$$

where F_C^2 and F_C^1 represent the 1×1 convolutional layer with the kernels of 3c and c, respectively. f_a is the output of pseudo 3D auto-correlation feature map, and its shape is consistent with the input size. Therefore, residual learning is added directly to reserve better cross-layer information. When input feature map $\mathbf{x}_t$ to the P3AB module, the output feature map $\mathbf{x}_{t+1}$ can be obtained as follow:

$$x_{t+1} = x_t + f_a. \tag{14}$$

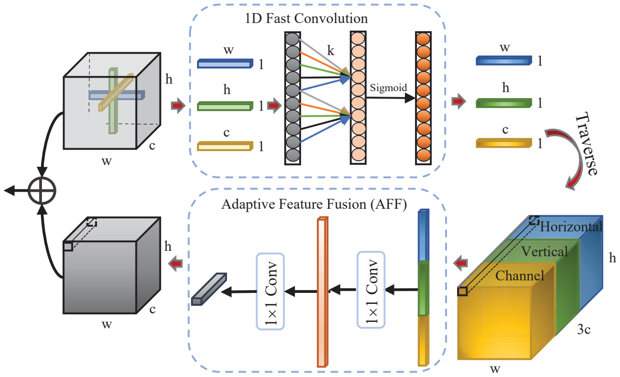


Figure 4. The internal implementation details of P3AB. After 1D fast convolution and adaptively feature fusion (AFF), the output contains the auto-correlation from three directions: horizontal, vertical, and channel. *Red arrows* indicate the flow of operations.

It should be noted that when $t = 0$, the input of t-th PCAB does not carry any auto-correlation information. If $t > 0$, the P3AB can further explore information from the output of the previous block. Through stacking P3ABs continuously, each position can collect auto-correlation from all pixels in a given image. Based on forwarding multiple transformations and the AFF, feature receptive field can be expanded to global reception only with local 1D convolutions. This decomposition strategy will reduce the time and space complexity efficiently. Besides, the cross-layer connection between layers of different depth proved to be more conducive to transfer the prior information in the noise image. We design the skip connections at different network locations to form a multi-level residual mechanism, which can stabilize model training and improve model performance.

4. Experiments

4.1. Datasets

We train our network on three synthetic noisy images datasets and four real noisy images datasets respectively. For ensuring fairness of comparison, we train all competition methods on the same training set.

Synthetic Noisy Images: We use training sets in DIV2K as high-quality clean images. Then four different white Gaussian noise levels with $\sigma = 10, 30, 50, 70$ are added to clean images respectively to generate noise image pairs. The color image denoising performance of P3AN is evaluated on the benchmarks: BSD68 [39], Kodak24 [41], and Urban100 [23], all are publicly available for downloading.

Real Noisy Images: We use the Smartphone Image Denoising Dataset (SIDD) [1] to train and evaluate the performance of our model on real-world image denoising. The images in SIDD are captured by five smartphone cameras in 10 static scenes. They have different lighting conditions and camera settings, and a total of 30,000 noisy images are included. We separated 24,000 images for model training and 1280 images for validation. We used two open-source

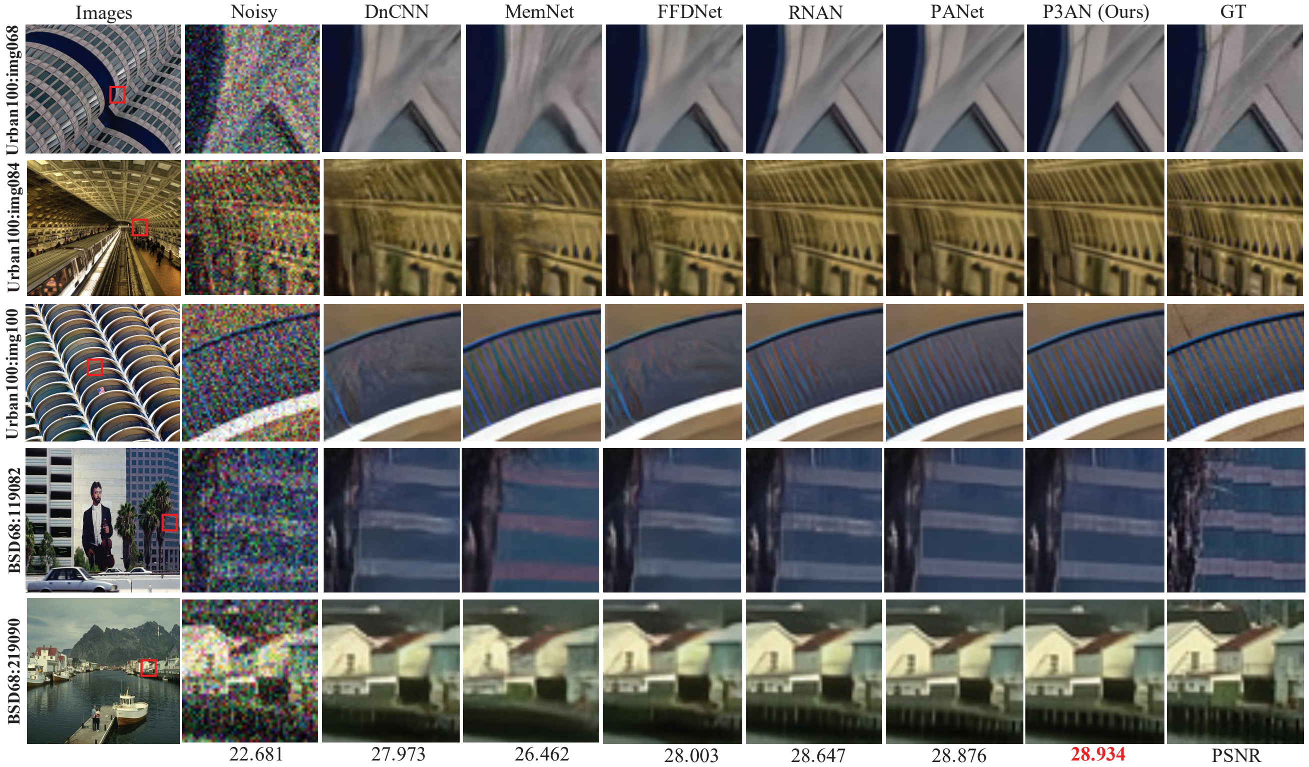


Figure 5. Visual comparison for color image denoising with noise level $\sigma = 50$ on Urban100 [23] and BSD68 [39].

Method	Kodak24				BSD68				Urban100			
	10	30	50	70	10	30	50	70	10	30	50	70
CBM3D [10]	36.57	30.89	28.63	27.27	35.91	29.73	27.38	26.00	36.00	30.36	27.94	26.31
TNRD [8]	34.33	28.83	27.17	24.94	33.36	27.64	25.96	23.83	33.60	27.40	25.52	22.63
RED [33]	34.91	29.71	27.62	26.36	33.89	28.46	26.35	25.09	34.59	29.02	26.40	24.74
DnCNN [52]	36.98	31.39	29.16	27.64	36.31	30.40	28.01	26.56	36.21	30.28	28.16	26.17
MemNet [43]	N/A	29.67	27.65	26.40	N/A	28.39	26.33	25.08	N/A	28.93	26.53	24.93
IRCNN [53]	36.70	31.24	28.93	N/A	36.06	30.22	27.86	N/A	35.81	30.28	27.69	N/A
FFDNet [54]	36.81	31.39	29.10	27.68	36.14	30.31	27.96	26.53	35.77	30.53	28.05	26.39
RNAN [56]	37.24	31.86	29.58	28.16	36.43	30.63	28.27	26.83	36.59	31.50	29.08	27.45
PANet [34]	37.35	31.96	29.65	28.20	36.50	30.70	28.33	26.89	36.80	31.87	29.47	27.87
P3AN (Ours)	**37.38**	**31.99**	**29.69**	**28.25**	**36.54**	**30.72**	**28.37**	**26.94**	**36.84**	**31.90**	**29.51**	**27.96**

Table 2. Quantitative results about **color** image denoising. Best results are **highlighted**.

real noise datasets during the test: Darmstadt Noise Dataset (DND) [38] and Nam [36]. Specifically, DND contains 50 pairs of images from four consumer cameras. Nam contains paired images of 11 static scenes. Besides, was also adopted the datasets from PolyU [47] to train models and further evaluate the denoising performance.

4.2. Implementation Details

In the experiment, we insert 20 P3ABs in the main network (i.e., $n = 20$). And we set $k = 5$ in the 1D fast convolution of P3AB. Each training data performs the same data augmentation, including random rotations of 90, 180, 270, and horizontal flipping. When training models, we select the best configuration according to the property of the datasets. For the synthetic noise training set DIV2K, there are 16 cropped 96×96 noise patches in each training batch, and the training epochs is taken as 1000. The number of input and output feature channels of each P3AB is 64. We use ADAM optimizer with $\{\beta_1 = 0.9, \beta_2 = 0.999,\ epsilon = 10^{-8}\}$. The learning rate is set as 1×10^{-4} and is halved after every 200 epochs. For the real noise training set SSID with higher resolution, 32 cropped 128×128 noise patches are each training batch and epochs are taker as 200. The parameters of ADAM optimizer are set to $\{\beta_1 = 0.9, \beta_2 = 0.999,\ epsilon = 10^{-8}\}$. The learning rate is initialized to 2×10^{-4} and linearly decreases to half every 20 epochs until 1e-6. We use PyTorch [37] to implement all our models and train them on NVIDIA GeForce RTX 2080 Ti GPU.

4.3. Comparisons with Other Methods

We report the comparing results on standard benchmarks to show the different performance of the proposed method and state-of-the-art denoising methods. We use peak signal-to-noise ratio (PSNR) and structural similarity index metric (SSIM) [45] as the quantitative criteria.

4.3.1 Quantitative Comparison:

Synthetic Noise: We compare the following state-of-the-art algorithms: CBM3D [10], TNRD [8], RED [33], DnCNN [52], MemNet [43], IRCNN [53], FFDNet [54], RNAN [56], and PANet [34]. Tab. 2 shows quantitative re-

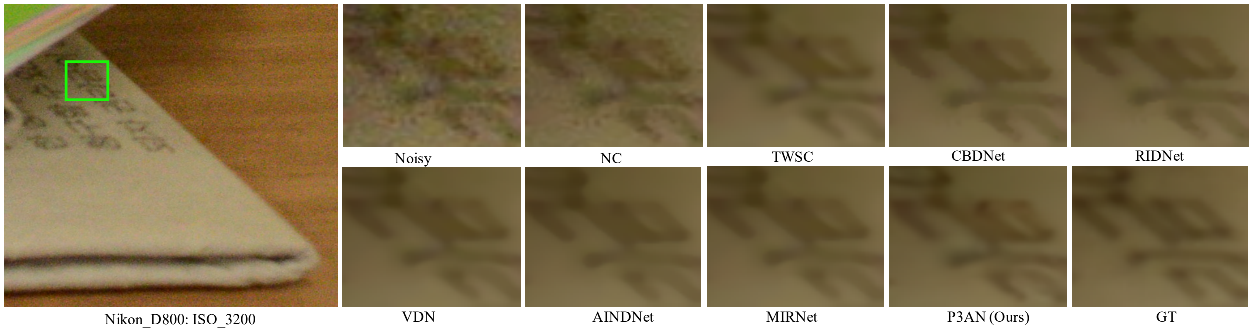


Figure 6. Visual comparisons between P3AN and other state-of-the-art denoising methods on the on the SIDD [1] benchmark. Our method eliminates complex noise effectively while retaining more structural content and texture, leading to artifact-free results.

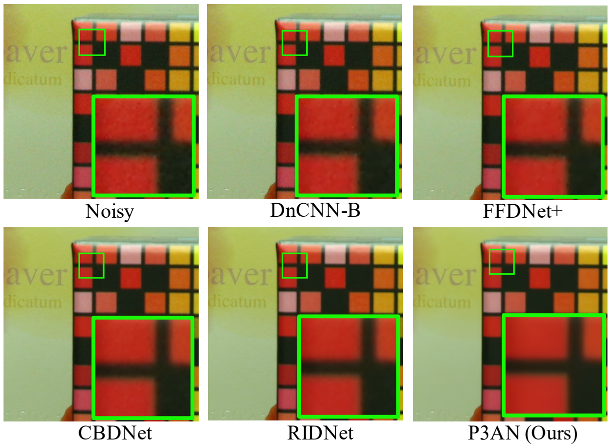


Figure 7. Denoising example from Canon_EOS_ISO_3200.

Method	SIDD	DND
DnCNN-B [52]	23.66 / 0.583	32.43 / 0.790
FFDNet+ [54]	- / -	37.61 / 0.942
CBDNet [19]	33.28 / 0.868	38.06 / 0.942
RIDNet [3]	38.71 / 0.914	39.26 / 0.953
VDN [50]	39.23 / 0.955	39.38 / 0.952
AINDNet [26]	39.15 / 0.955	39.53 / 0.956
MIRNet [51]	39.72 / 0.959	**39.88**/0.956
P3AN (Ours)	**39.85 / 0.971**	39.68 / **0.960**

Table 3. Quantitative comparison on SSID and DND.

sults of color image denoising. Compared with all previous methods, our proposed P3AN performs the best results on all the datasets with all noise levels. In particular, our network performs significantly well on the noise level $\sigma = 70$. It shows that the proposed method has better resilience on heavily polluted images. Our network can extract complete auto-correlation from the cross-directional 3D space simultaneously to guide recovery, which is more superior than self-similarity state-of-the-art PANet.

Real Noise: We compare our approach with **12** state-of-the-art real noise removal algorithms: DnCNN-B [52], Neat Image (NI) [2], Noise Clinic (NC) [27], MCWNNM [49], RDN [57], FFDNet+ [54], TWSC [48], CBDNet [19], RIDNet [3], VDN [50], AINDNet [26], and MIRNet [51]. As shown in Tab. 3 and Tab. 4, four real photographs benchmark datasets are used to evaluate models quantitatively. Obviously, compared to other methods, P3AN has achieved a significant improvement in PSNR and SSIM. In the open-source testsets PolyU and Nam, the denoising results of our method are 1.61 dB and 1.10 dB better than VDN on PSNR, respectively. The test results of SIDD and DND need to be accessed on the official benchmark website. We report the obtained official evaluation results in Tab. 3. On the SIDD, it achieves a gain of 0.13 dB compared with the state-of-the-art. Although our method does not obtain the highest PSNR on DND compared with the latest MIRNet, we have the highest SSIM, which further indicates that the denoised images from our model are closer to the ground truth.

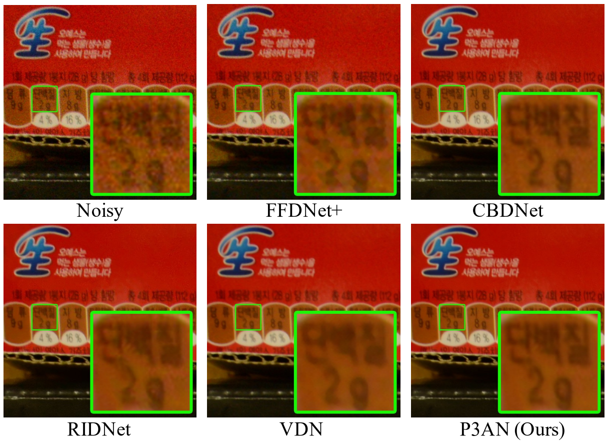


Figure 8. Denoising example from Nam [36] testing set.

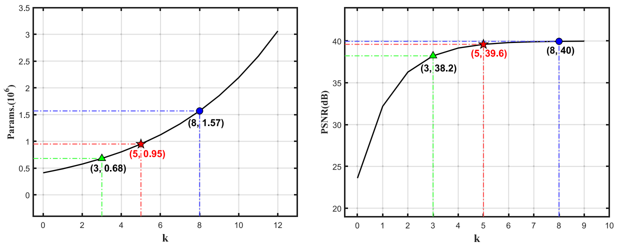
Method	PolyU	Nam
DnCNN-B [52]	34.68 / 0.874	34.95 / 0.885
NI	35.91 / 0.921	36.61 / 0.926
NC [27]	36.84 / 0.936	37.69 / 0.952
MCWNNM [49]	37.72 / 0.945	37.84 / 0.956
RDN [57]	37.94 / 0.946	38.16 / 0.956
FFDNet+ [54]	38.17 / 0.951	38.81 / 0.957
TWSC [48]	38.68 / 0.958	38.96 / 0.962
CBDNet [19]	38.74 / 0.961	39.08 / 0.969
RIDNet [3]	38.86 / 0.962	39.20 / 0.973
VDN [50]	39.04 / 0.965	39.68 / 0.976
P3AN (Ours)	**40.65 / 0.976**	**40.78 / 0.982**

Table 4. Quantitative comparison on PolyU and Nam.

Visual Comparison: The visual evaluation of synthetic noise removal was performed on Urban100. As shown in Fig. 5, for some more complex textures, DnCNN, IRCNN, FFDNet, and RNAN exhibit excessive smoothness. RED and MemNet produce color difference, for example, the blue line in img100 becomes green. PANet has slightly better results, but it also brings some distortion and blur in the straight lines. It can be seen that only our method removes

Method	Synthetic Noise						Real Noise						
	RED	DnCNN	MemNet	RNAN	PANet	Ours	DnCNN-B	FFDNet+	RDN	CBDNet	RIDNet	VDN	Ours
Params. (10^6)	4.2	**0.68**	0.69	7.59	6.10	1.53	0.56	**0.49**	21.97	4.34	1.50	4.36	0.95
Flops (10^9)	135.17	148.53	149.63	276.84	249.62	**122.87**	146.94	107.29	717.82	144.36	392.53	158.49	**105.76**

Table 5. Analysis of the complexity and inference speed of different models for synthetic noise removal and real noise removal.



(a) Paramaters. (b) Performance.

Figure 9. Variation curve of model complexity and performance with the different local receptive fields of 1D fast convolution.

noise while preserving details of the original structure as much as possible, and get results closest to the ground truth.

For real noisy images, we visualize the denoising results on the SSID benchmark and Nam test sets. In Fig. 6 we compare popular methods, including the latest denoising network AIDNet and MIRNet this year. Obviously, the NC, CBDNet, and RIDNet methods have some residual noise and artifacts. The results of TWSC and VDN have missed and deformed lines and lost detailed information. Although the AIDNet and MIRNet methods show better denoising performance, they excessively smooth the structural lines. Our method effectively recovers a more accurate image from the challenging irregular noise image, as close to the original image as possible in color and texture details. The same comparison can be seen in Figs. 7 and 8. It is difficult for other networks to remove noise and blur while retaining accurate textures. The proposed P3AN better describes the complete image information and produces visually-pleasing results. It achieves the best overall visual quality among the latest competitive algorithms.

4.4. Model Analysis

Local Receptive Field: We analyzed the influence of the receptive field in 1D fast convolution on the model complexity and performance. The relation curves are drawn in Fig 9. The number of parameters will increase along with the increase of the receptive field. It can be seen in Fig 9(a) that the model parameters show a non-linear exponential increase when k becomes larger. We selected the coordinate points of $k = 3, 5, 8$ and marked them in green, red, and blue. The relation curve between k and the PSNR is ploted in Fig. 9(b), which shows that in the previous increase of k, the PSNR value will increase accordingly. Then the PSNR only improves slightly when $k > 6$. The model performance changes hardly when $k > 8$, but the amount of parameters has an explosive increase. The corresponding coordinates are marked for better indication. Therefore, for the trade-off between model performance and complexity and calculation burden, we set k as 5. The quantitative results show that when $k = 5$, the parameter quantity is 0.95×10^6, and the PSNR evaluated on the DND testset is 39.6. This performance is sufficient to defeat other state-of-the-art methods while ensuring the lightweight of the model, which is beneficial for further expansion and practical applications.

Model Complexity: We analyze the model parameters and floating-point operations (FLOP) in the experiment to evaluate the model complexity. The comparison results between different methods based on synthetic noise and real noise images are reported in Tab. 5, respectively. To ensure a fair comparison, we use PolyU and BSD68 to evaluate different types of denoising networks. Each result is an average value obtained after ten repeated experiments. Compared with the DnCNN with only 17 layers, it can be seen our method achieves the best computational efficiency with a deeper network on synthetic data. As for the real noise removal task, the P3AN utilizes moderate-scale parameters and supports fast model inference, achieving a trade-off between model complexity and performance. These results show that our P3AN not only achieves high precision in denoising tasks but also is a more practical lightweight network. It is worth mentioning that due to the superiority of 1D convolution, the computational memory of our network will not rise sharply as the input image size increases. This progress dramatically reduces the dependence of deep networks on high-performance hardware devices.

5. Conclusion

In this paper, we focus on modifying the popular basic convolution structure used for feature auto-correlation extraction. A lightweight pseudo 3D auto-correlation network (P3AN) is designed to avoid dense connections and high-dimensional operations. Based on local 1D fast convolution, P3AN can extract auto-correlation information from the three directions of horizontal, vertical, and channel simultaneously. Consecutive blocks can obtain features with varying receptive fields, which means that effective local attention can also cover global interactions of the entire input. This novel method of aggregating contextual features unifies channel self-attention and spatial self-similarity capture into the same framework. Experiments show that our method has much fewer parameters and low computational cost while achieving very competitive performance.

Acknowledgments: This work is partially supported by the NSFC fund (61831014), the Guangdong Provincial Science and Technology Project (2017B010110005), the Shenzhen Science and Technology Project under Grant (ZDYBH201900000002, JCYJ20180508152042002, GGFW2017040714161462).

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018. 1, 5, 7

[2] Neatlab ABSoft. Neat image. *https://ni.neatvideo.com/home*. 7

[3] Saeed Anwar and Nick Barnes. Real image denoising with feature attention. In *ICCV*, 2019. 1, 2, 7

[4] Saeed Anwar, Cong Phuoc Huynh, and Fatih Porikli. Identity enhanced residual image denoising. In *CVPR Workshops*, 2020. 2

[5] Yuval Bahat and Michal Irani. Blind dehazing using internal patch recurrence. In *ICCP*, 2016. 3

[6] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *CVPR*, 2005. 2

[7] Harold C Burger, Christian J Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *CVPR*, 2012. 2

[8] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *TPAMI*, 2017. 6

[9] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017. 3

[10] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space. In *ICIP*, 2007. 6

[11] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *TIP*, 2007. 2

[12] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *CVPR*, 2019. 1

[13] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *CVPR*, 2019. 3

[14] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *TIP*, 2006. 2

[15] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, 2019. 2

[16] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, 2020. 3

[17] Zilin Gao, Jiangtao Xie, Qilong Wang, and Peihua Li. Global second-order pooling convolutional networks. In *CVPR*, 2019. 3

[18] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *ICCV*, 2009. 3

[19] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *CVPR*, 2019. 2, 7

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 5

[21] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. In *NeurIPS*, 2018. 3

[22] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 1, 3

[23] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, 2015. 4, 5, 6

[24] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, 2019. 1

[25] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. *TPAMI*, 2020. 3

[26] Yoonsik Kim, Jae Woong Soh, Gu Yong Park, and Nam Ik Cho. Transfer learning from synthetic to real-noise denoising with adaptive instance normalization. In *CVPR*, 2020. 3, 7

[27] Marc Lebrun, Miguel Colom, and Jean-Michel Morel. Multiscale image blind denoising. *TIP*, 2015. 7

[28] Stamatios Lefkimmiatis. Non-local color image denoising with convolutional neural networks. In *CVPR*, 2017. 2

[29] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *CVPR*, 2019. 1

[30] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. In *NeurIPS*, 2018. 1

[31] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. In *NeurIPS*, 2018. 3

[32] Y. Liu, S. Anwar, L. Zheng, and Q. Tian. Gradnet image denoising. In *CVPR Workshops*, 2020. 2

[33] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *NeurIPS*, 2016. 6

[34] Yiqun Mei, Yuchen Fan, Yulun Zhang, Jiahui Yu, Yuqian Zhou, Ding Liu, Yun Fu, Thomas S Huang, and Honghui Shi. Pyramid attention networks for image restoration. 2020. 1, 6

[35] Tomer Michaeli and Michal Irani. Blind deblurring using internal patch recurrence. In *ECCV*. Springer, 2014. 1

[36] Seonghyeon Nam, Youngbae Hwang, Yasuyuki Matsushita, and Seon Joo Kim. A holistic approach to cross-channel image noise modeling and its application to image denoising. In *CVPR*, 2016. 1, 6, 7

[37] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 6

[38] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017. 6

[39] Stefan Roth and Michael J Black. Fields of experts: A framework for learning image priors. In *CVPR*, 2005. 5, 6

[40] Abhijit Guha Roy, Nassir Navab, and Christian Wachinger. Recalibrating fully convolutional networks with spatial and channel "squeeze and excitation" blocks. *TMI*, 2018. 3

[41] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 5

[42] Robert J Schalkoff. *Digital image processing and computer vision*. 1989. 1

[43] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *ICCV*, 2017. 6

[44] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 1, 3

[45] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004. 6

[46] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, 2018. 2, 3

[47] Jun Xu, Hui Li, Zhetong Liang, David Zhang, and Lei Zhang. Real-world noisy image denoising: A new benchmark. *arXiv preprint arXiv:1804.02603*, 2018. 6

[48] Jun Xu, Lei Zhang, and David Zhang. A trilateral weighted sparse coding scheme for real-world image denoising. In *ECCV*, 2018. 7

[49] Jun Xu, Lei Zhang, David Zhang, and Xiangchu Feng. Multi-channel weighted nuclear norm minimization for real color image denoising. In *ICCV*, 2017. 7

[50] Zongsheng Yue, Hongwei Yong, Qian Zhao, Deyu Meng, and Lei Zhang. Variational denoising network: Toward blind noise modeling and removal. In *NeurIPS*, 2019. 3, 7

[51] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Learning enriched features for real image restoration and enhancement. 2020. 3, 7

[52] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *TIP*, 2017. 2, 6, 7

[53] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *CVPR*, 2017. 2, 6

[54] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn based image denoising. *arXiv preprint arXiv:1710.04026*, 2017. 6, 7

[55] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018. 1

[56] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. In *ICLR*, 2019. 1, 3, 6

[57] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image restoration. *TPAMI*, 2020. 2, 7

[58] Maria Zontak and Michal Irani. Internal statistics of a single natural image. In *CVPR*, 2011. 3