Attention Cube Network for Image Restoration

Yucheng Hang  
Shenzhen International Graduate School, Tsinghua University  
Shenzhen, China  
ychang20@163.com

Qingmin Liao  
Shenzhen International Graduate School & Department of Electronic Engineering, Tsinghua University  
Shenzhen, China  
liaoqm@tsinghua.edu.cn

Wenming Yang*  
Shenzhen International Graduate School & Department of Electronic Engineering, Tsinghua University  
Shenzhen, China  
yang.wenming@sz.tsinghua.edu.cn

Yupeng Chen  
Peng Cheng Laboratory  
Shenzhen, China  
chenyp01@pcl.ac.cn

Jie Zhou  
Department of Automation, Tsinghua University  
Beijing, China  
jzhou@tsinghua.edu.cn

Figure 1: Gray image denoising results with noise level 50 on BSD68: 223061 from BSD68. Our method obtains better visual quality and recovers more textural details compared with other state-of-the-art methods.

ABSTRACT

Recently, deep convolutional neural network (CNN) have been widely used in image restoration and obtained great success. However, most of existing methods are limited to local receptive field and equal treatment of different types of information. Besides, existing methods always use a multi-supervised method to aggregate different feature maps, which can not effectively aggregate hierarchical feature information. To address these issues, we propose an attention cube network (A-CubeNet) for image restoration for more powerful feature expression and feature correlation learning. Specifically, we design a novel attention mechanism from three dimensions, namely spatial dimension, channel-wise dimension and hierarchical dimension. The adaptive spatial attention branch (ASAB) and the adaptive channel attention branch (ACAB) constitute the adaptive dual attention module (ADAM), which can capture the long-range spatial and channel-wise contextual information to expand the receptive field and distinguish different types of information for more effective feature representations. Furthermore, the adaptive hierarchical attention module (AHAM) can capture the long-range hierarchical contextual information to flexibly aggregate different feature maps by weights depending on the global context. The ADAM and AHAM cooperate to form an "attention in attention" structure, which means AHAM’s inputs are enhanced by ASAB and ACAB. Experiments demonstrate the superiority of our method over state-of-the-art image restoration methods in both quantitative comparison and visual analysis.

CCS CONCEPTS

- Computing methodologies → Computational photography; Reconstruction; Image processing.

KEYWORDS

Image restoration; Attention cube; Contextual information

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM ’20, October 12–16, 2020, Seattle, WA, USA  
© 2020 Association for Computing Machinery.  
ACM ISBN 978-1-4503-7988-5/20/10 ... $15.00  
https://doi.org/10.1145/3394171.3413564
1 INTRODUCTION

Image restoration is a classic computer vision task that aims to recover high-quality images from low-quality images corrupted by various kinds of degradations. Due to the irreversible nature of the image degradation process, it is an ill-posed problem. It can be categorized into different tasks such as image super-resolution, image denoising, JPEG image deblocking, etc.

Recently, methods based on deep convolutional neural network (CNN) have been widely used in image restoration due to their strong non-linear representational power. [9, 12, 16, 20, 22, 23, 46, 48] focused on designing a deeper, wider or lighter network structure, aiming at improving the performance of image super-resolution. Dong et al. [10] proposed ARCNN with several stacked convolutional layers for JPEG image deblocking. By taking a tunable noise level map as input, FFDNet can deal with noise on different levels [44]. Guo et al. [15] proposed CBDNet for blind denoising of real images. However, these methods are specific for image restoration tasks. Different from them, [29, 30, 42, 43] developed a couple of valuable methods that can be generalized to different image restoration tasks.

Although above CNN-based image restoration methods have achieved gratifying results, they still have some problems: (1) The receptive fields of these methods are relatively small. Most of these methods extract features in a local way through convolution operations, which cannot capture the long-range dependencies between pixels in the whole image. A larger receptive field can make better use of training images and more contextual information, which is very helpful for image restoration, especially when the images suffer from heavy corruptions. (2) Most of these methods treat all types of information (e.g., low and high frequency information) equally, which may result in over-smoothed reconstructed images and fail to recover some textural details. In other words, the power of discrimination of these methods is limited. (3) In response to problem one and two, although there have been recent methods that use attention mechanism (e.g., SENet [17] and non-local neural network [39]) for image super-resolution, most methods directly introduce the attention mechanism in high-level computer vision tasks and ignore the difference between high-level and low-level computer vision tasks [9, 46]. (4) Most of these methods only use the feature map outputted from the last layer for image restoration. In fact, in deep convolutional neural networks, feature map information at different levels can complement each other. If only the feature map outputted from the last layer is used for image restoration, part of the information is bound to be lost. Although there are some methods that take this problem into account, they just cascade them together [48] or use a multi-supervised method [23, 37], and the feature map information of each level is not fully and effectively used.

In order to solve above problems, this paper proposes an attention cube network (A-CubeNet) for image restoration based on adaptive dual attention module (ADAM) and adaptive hierarchical attention module (AHAM). Specifically, this paper proposes an adaptive dual attention module for above mentioned problem one, two, and three. This module can capture the long-range dependencies between pixels and channels to expand the receptive field and distinguish different types of information for more effective feature representations, which is very helpful for the restoration of textural details. In addition, inspired by the non-local neural network, this paper designs an adaptive hierarchical attention module for above mentioned problem four. This module first performs squeeze operations on the spatial and channel-wise dimensions of each feature map for global context modeling. Then this module captures the long-range dependencies between different feature maps. Finally, this module fuses each feature map based on the long-range dependencies between different feature maps. Full experiments show that compared with other state-of-the-art methods, our A-CubeNet achieves the best results in all tasks.

In summary, the main contributions of this paper are listed as follows:

- We propose an adaptive dual attention module (ADAM), including an adaptive spatial attention branch (ASAB) and an adaptive channel attention branch (ACAB). ADAM can capture the long-range spatial and channel-wise contextual information to expand the receptive field and distinguish different types of information for more effective feature representations. Therefore our A-CubeNet can obtain high-quality image restoration results, as shown in Figure 1.

- Inspired by the non-local neural network, we design an adaptive hierarchical attention module (AHAM), which flexibly aggregates all output feature maps together by the hierarchical attention weights depending on global context. To the best of our knowledge, this is the first time to consider aggregating output feature maps in a hierarchical attention method with global context.

- Through sufficient experiments, we prove that our A-CubeNet is powerful for various image restoration tasks. Compared with other state-of-the-art methods for image denoising, JPEG image deblocking and image super-resolution, our A-CubeNet obtains superior results in both quantitative metrics and visual quality.

2 RELATED WORK

2.1 Image Restoration

CNN-based image restoration methods cast image restoration as an image-to-image regression problem, and learn an end-to-end mapping from low-quality (LQ) to high-quality (HQ) directly. Dong et al. [10, 11] proposed SRCNN for image super resolution and ARCNN for JPEG image deblocking. Both of them achieved superior performance against previous methods that are not based on CNN. Then [9, 16, 18, 22, 23, 46, 48] focused on designing a deeper and wider network structure or considering feature correlations to improve the performance of image super-resolution. Zhang et al. [42] proposed DnCNN for image denoising and JPEG image deblocking by introducing residual learning to train deeper network. [43] introduced the denoiser prior for fast image restoration. Tai et al. [37] lately designed a persistent memory network for image restoration and achieved promising results. However, most methods neglect to consider feature correlations in spatial or channel-wise dimensions and can not make full use of hierarchical feature maps.
2.2 Attention Mechanism

Attention mechanism is inspired by the cognitive process of human [33]. Human always focuses on more important information. Attention mechanism has been widely used in various computer vision tasks, such as image and video classification tasks [17, 39]. Wang et al. [39] proposed non-local neural network by incorporating non-local operations for spatial attention in video classification. [17] modelled channel-wise relationships to obtain significant performance gain for image classification. Woo et al. [40] developed CBAM to model both channel-wise and spatial relationships. In all, these works mainly aimed to concentrate on more useful information in features.

Recently, SENet was introduced to improve SR performance [46]. [9, 18, 24] focused on introducing both channel attention and spatial attention to image super-resolution. Dai et al. [9] designed a second-order attention mechanism based on SENet and introduced non-local neural network to further improve SR performance. Following the importance of self-similarity prior, [47] adapted non-local operations into their network. However, most methods directly introduce the attention mechanism in high-level computer vision tasks and ignore the difference between high-level and low-level computer vision tasks. For example, non-local neural network is computationally intensive and is actually not suitable for image restoration.

3 METHOD

3.1 Framework

As shown in Figure 2, the proposed A-CubeNet mainly consists of three modules: the shallow feature extraction module, the deep feature extraction module stacked with G residual dual attention groups (RDAGs) and one adaptive hierarchical attention module (AHAM), and the construction module. Given \( I_{LQ} \) and \( I_{HQ} \) as the low-quality (e.g., noisy, low resolution, or compressed images) and high-quality images. We apply only one convolutional layer to extract the shallow feature \( F_0 \) from the low-quality input:

\[
F_0 = H_{SF}(I_{LQ}), \tag{1}
\]

where \( H_{SF}() \) represents the shallow feature extraction module. Then the extracted shallow feature \( F_0 \) is used for RDAGs and AHAM based deep feature extraction, which thus produces the deep feature as:

\[
F_{DF} = H_{DF}(F_0) = F_0 + W_{LSC}H_{AHAM}(F_1, \cdots, F_g, \cdots, F_G), \tag{2}
\]

where \( H_{DF}() \), \( W_{LSC} \) and \( H_{AHAM}() \) represent the deep feature extraction module, the convolutional layer at the end of the deep feature extraction module and the adaptive hierarchical attention module (AHAM), respectively. To address the image super-resolution task, we add an extra upscale layer before the last convolutional layer. Specifically, we utilize sub-pixel convolutional operation (convolution + pixel shuffle) [35] to upscale feature maps:

\[
F_{UP} = H_{UP}(F_{DF}), \tag{3}
\]

where \( H_{UP}() \) represents the upscale module. Then we use a convolutional layer to get the reconstructed image:

\[
I_{REC} = H_{REC}(F_{DF}) \text{ or } I_{REC} = H_{REC}(F_{UP}), \tag{4}
\]

where \( H_{REC}() \) represents the reconstruction module. The overall reconstruction process can be expressed as:

\[
I_{REC} = H_{A-CubeNet}(I_{LQ}), \tag{5}
\]

where \( H_{A-CubeNet}() \) represents the function of our A-CubeNet.

Then A-CubeNet is optimized with a certain loss function. Some loss functions have been widely adopted, such as L2 [10, 37, 42, 43], L1 [9, 28, 48], perceptual and adversarial losses [26]. To verify the effectiveness of our A-CubeNet, we adopt the same loss functions as previous works (e.g., L1 loss function for image super-resolution, L2 loss function for image denoising and JPEG image deblocking).

Given a training set \( \{I_{LQ}, I_{HQ}\}_{i=1}^{N} \) with N LQ images and their HQ counterparts. The goal of training A-CubeNet is to optimize the loss function:

\[
L_1(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left\| H_{A-CubeNet}(I_{LQ}) - I_{HQ} \right\|_1, \tag{6}
\]

\[
L_2(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left\| H_{A-CubeNet}(I_{LQ}) - I_{HQ} \right\|_2,
\]

where \( \theta \) indicates the updateable parameters of our A-CubeNet.

3.2 Residual Dual Attention Group (RDAG)

As shown in Figure 2, the deep feature extraction module stacked with several RDAGs, while an RDAG consists of two parts: U stacked RDAUs and one convolutional layer. The g-th RDAG can be expressed as:

\[
F_g = F_{g-1} + W_{SSC}H_{g,u}(H_{g,u-1}(\cdots H_{g,1}(F_{g-1})\cdots)) \tag{7},
\]

where \( F_{g-1} \) and \( F_g \) represent the input and output of the g-th RDAG. \( W_{SSC} \) and \( H_{g,u}() \) represent the convolutional layer at the end of the g-th RDAG and the u-th RDAU of the g-th RDAG, respectively. Each RDAU consists of one residual block and one adaptive dual attention module (ADAM). Then we give more details to ADAM and AHAM.

3.3 Adaptive Dual Attention Module (ADAM)

As shown in Figure 3, our adaptive dual attention module (ADAM) consists of two branches: adaptive spatial attention branch (ASAB) and adaptive channel attention branch (ACAB). These two branches cooperate to draw global features rather than local features and endow the network the ability to treat different types of information differently. Thus our A-CubeNet obtains better feature representations for high-quality image restoration. Take the adaptive spatial attention branch in Figure 3 as an example. First, we apply a convolutional layer to squeeze channel-wise features and apply softmax function to obtain the attention weights. Second, we perform a matrix multiplication between the attention weights and the original features to obtain long-range spatial contextual information. Third, we perform feature transform and feature fusion with adaptive weight. Similarly, we obtain channel-wise long-range contextual information by the adaptive channel attention branch.

In general, each branch has three steps: (1) Squeeze features; (2) Extract long-range contextual information; (3) Perform feature fusion with adaptive weight. Our ADAM can be formulated as:

\[
Out_{ADAM} = X + Out_{ASAB} + Out_{ACAB}. \tag{8}
\]
Figure 2: Framework of the proposed attention cube network (A-CubeNet)

**3.3.1 Adaptive spatial attention branch (ASAB).** As shown in Figure 3, given an input feature map \( X = \{X_i\}_{i=1}^{N} \subset \mathbb{R}^{C \times H \times W} \), where \( N = H \times W \) is the number of pixels. First, we apply a convolutional layer \( W_k \) to squeeze channel-wise features: \( A_s = W_k X \). Then we reshape it to \( \mathbb{R}^{HW \times 1 \times 1} \) and apply a softmax function to obtain the spatial attention weights \( \omega_s \in \mathbb{R}^{HW \times 1 \times 1} \):

\[
\omega_s^j = \frac{\exp(A_s^j)}{\sum_{m=1}^{N} \exp(A_s^m)}.
\]

Then we reshape \( X \in \mathbb{R}^{C \times H \times W} \) to obtain \( B_s \in \mathbb{R}^{C \times H \times W} \). After that we perform long-range spatial contextual information modeling, which groups the features of all pixels together with the spatial attention weights to obtain the long-range spatial contextual features. Specifically, we perform a matrix multiplication between \( B_s \) and the spatial attention weights \( \omega_s^j \):

\[
D_s^j = \sum_{m=1}^{N} \omega_s^j B_s^m,
\]

where \( D_s^j \in \mathbb{R}^{C \times 1 \times 1} \) represents spatial long-range contextual features. Then we feed it into a bottleneck network (e.g., one \( 1 \times 1 \) convolutional layer \( W_s^1 \), one ReLU activation layer and one \( 1 \times 1 \) convolutional layer \( W_s^2 \)) to perform feature transform. Finally, we multiply it by an adaptive learning weight \( \alpha_s \):

\[
Out_{ASAB} = \alpha W_s^2 \text{ReLU} \left( W_s^1 \sum_{j=1}^{N} \omega_s^j B_s^j \right),
\]

where \( Out_{ASAB} \in \mathbb{R}^{C \times 1 \times 1} \) represents the output of ASAB.

We emphasize three points: (1) \( \alpha \) is initialized as 0 and gradually learns to assign larger weight. This adaptive learning weight helps our A-CubeNet fuse long-range spatial contextual features effectively. (2) We use a bottleneck network with the bottleneck ratio \( r \) instead of only one \( 1 \times 1 \) convolutional layer because: (a) Compared with using only one \( 1 \times 1 \) convolutional layer, the number of parameters reduces from \( C^2 \) to \( 2C^2 \) / \( r \). (b) Just like SENet [17], it can better fit the complex correlation between channels with more nonlinearity. However, noted that this correlation between channels is directed at long-range spatial contextual features instead of the input feature map. So it can only be regarded as a supplement to the spatial attention mechanism. (3) The Equation 11 shows that \( Out_{ASAB} \) is a weighted sum of the features across
all pixels. Therefore, it has a global receptive field and selectively aggregates context according to the spatial attention weights.

### 3.3.2 Adaptive channel attention branch (ACAB)
As shown in Figure 3, given an input feature map \( X = \{ X_i \}_{i=1}^C \), \( X \in \mathbb{R}^{C \times H \times W} \), where \( C \) is the number of channels. First, we apply an average pooling layer \( \text{Pool}_{\text{avg}} \) to squeeze spatial features: \( A^e = \text{Pool}_{\text{avg}} X \), \( A^e \in \mathbb{R}^{C \times 1 \times 1} \). Then we apply a softmax function to obtain the channel-wise attention weights \( \omega^e \in \mathbb{R}^{C \times 1 \times 1} \):

\[
\omega^e_j = \frac{\exp \left( A^e_j \right)}{\sum_{m=1}^{C} \exp \left( A^e_m \right)}, \quad (12)
\]

Then we reshape \( X \in \mathbb{R}^{C \times H \times W} \) to obtain \( B^e \in \mathbb{R}^{C \times H \times W} \). After that we perform long-range channel-wise contextual information modeling, which groups the features of all channels together with the channel-wise attention weights to obtain long-range channel-wise contextual features. Specifically, we perform a matrix multiplication between \( B^e \) and the channel-wise attention weights \( \omega^e \):

\[
D^e = \sum_{j=1}^{C} \omega^e_j B^e_j, \quad (13)
\]

where \( D^e \in \mathbb{R}^{C \times 1 \times 1} \) represents channel-wise long-range contextual features. Then we feed it into a network (e.g., one 3 \( \times \) 3 convolutional layer \( W_c \), one ReLU activation layer and one 3 \( \times \) 3 convolutional layer \( W_c \)) to perform feature transform. Finally, we multiply it by an adaptive learning weight \( \beta \):

\[
\text{Out}_{\text{ACAB}} = \beta W_c \text{ReLU} \left( W_c \sum_{j=1}^{C} \omega^e_j B^e_j \right) = \beta W_c \text{ReLU} \left( W_c \sum_{j=1}^{C} \frac{\exp \left( \text{Pool}_{\text{avg}} X_j \right)}{\sum_{m=1}^{C} \exp \left( \text{Pool}_{\text{avg}} X_m \right)} X_j \right), \quad (14)
\]

where \( \text{Out}_{\text{ACAB}} \in \mathbb{R}^{1 \times H \times W} \) represents the output of ACAB.

We emphasize three points: (1) \( \beta \) is initialized as 0 and gradually learns to assign larger weight. This adaptive learning weight helps our AGAM fuse long-range channel-wise contextual features effectively. (2) We use a network instead of only one 3 \( \times \) 3 convolutional layer because it can better fit the complex spatial correlation with more nonlinearity, just like CSAR block [18]. However, noted that this spatial correlation is directed at long-range channel-wise contextual features instead of the input feature map. So it can only be regarded as a supplement to the channel attention mechanism. (3) The Equation 14 shows that \( \text{Out}_{\text{ACAB}} \) is a weighted sum of the features across all channels. It makes our A-CubeNet focus on more informative features and improve the power of discrimination.

### 3.4 Adaptive Hierarchical Attention Module (AHAM)
As shown in Figure 4, given a set of each residual dual attention group’s output feature map \( F = \{ F_g \}_{g=1}^G \), \( F_g \in \mathbb{R}^{C \times H \times W} \), where \( G \) is the number of residual dual attention groups (RDAGs). First, we apply an average pooling layer \( \text{Pool}_{\text{avg}} \) and a convolutional layer \( W_g \) to squeeze global features of each feature map: \( A^g = \)

![Figure 4: Adaptive hierarchical attention module (AHAM)](image-url)

\[
W_g \text{Pool}_{\text{avg}} F_g, \quad A^g \in \mathbb{R}^{1 \times 1 \times 1}. \quad \text{Then we cascade them to obtain } A^h \in \mathbb{R}^{1 \times 1 \times G}. \quad \text{After that we apply a softmax function to obtain the hierarchical attention weights } \omega^h \in \mathbb{R}^{1 \times 1 \times G}:
\]

\[
\omega^h_j = \frac{\exp \left( A^h_j \right)}{\sum_{m=1}^{C} \exp \left( A^h_m \right)}, \quad (15)
\]

Then we cascade each \( F_g \) to obtain \( B^h \in \mathbb{R}^{C \times H \times W \times G} \). After that we perform global contextual information modeling, which groups all RDAG\(\text{AA}\)Zs output feature maps together with the self-adaptive weights to obtain global contextual features. Specifically, we perform a matrix multiplication between \( B^h \) and the hierarchical attention weights \( \omega^h \):

\[
D^h = \sum_{j=1}^{G} \omega^h_j B^h_j, \quad (16)
\]

where \( D^h \in \mathbb{R}^{C \times H \times W} \) represents global contextual features. Then, we multiply it by an adaptive learning weight \( \gamma \). Finally, we perform an element-wise sum operation with the last RDAG\(\text{AA}\)Zs output feature map \( F_G \) to obtain the final output \( \text{Out}_{\text{AHAM}} \in \mathbb{R}^{C \times H \times W} \):

\[
\text{Out}_{\text{AHAM}} = F_G + \gamma \sum_{j=1}^{G} \omega^h_j B^h_j = F_G + \gamma \sum_{j=1}^{G} \frac{\exp \left( W_g \text{Pool}_{\text{avg}} F_j \right)}{\sum_{m=1}^{G} \exp \left( W_m \text{Pool}_{\text{avg}} F_m \right)} F_j. \quad (17)
\]

We emphasize three points: (1) \( \gamma \) is initialized as 0 and gradually learns to assign larger weight. This adaptive learning weight helps our A-CubeNet fuse global contextual features effectively. (2) The Equation 17 shows that \( \text{Out}_{\text{AHAM}} \) is a weighted sum of all RDAG\(\text{AA}\)Zs output feature maps. The hierarchical attention weights depend on global context of all intermediate feature maps. (3) AHAM can learn how to combine different hierarchical feature maps that are most conducive to reconstruction.

### 4 EXPERIMENTS

#### 4.1 Datasets and Metrics
We apply our A-CubeNet to three classical image restoration tasks: image super-resolution, gray image denoising and JPEG image deblocking. DIV2K dataset [1] is used to train all of our models. For image super-resolution, we follow the same settings as EDSR [28].
Set5 [3], Set14 [41], BSD100 [31], Urban100 [19] and Manga109 [32] are adopted as the test datasets. For gray image denoising, we follow the same setting as IRCNN [43]. BSD68 [31] and Kodak24 are used as the test datasets. For JPEG image deblurring, we follow the same setting as ARCNN [10]. LIVE1 [34] and Classic5 [14] are applied as the test datasets. For each task, we adopt the mean PSNR and/or SSIM to evaluate the results.

4.2 Implementation Details
Our A-CubeNet contains 4 RDAGs (G = 4) and each RDAG contains 4 RDAUs (U = 4). In each ASAB, we use 1 × 1 convolutional filter with the bottleneck ratio r = 16. In each ACAB, we set the size and number of filters as 3 × 3 and 1. In AHAM, we set the size and number of filters as 1 × 1 and 1. For other convolutional layers, the size and number of filters are set as 3 × 3 and 64. During training, data augmentation is performed on the training images, which are randomly rotated by 90°, 180°, 270° and flipped horizontally. In each mini-batch, we randomly cropped 16 patches with size 48 × 48 from the low-quality (LQ) images. Our model is trained by ADAM optimizer with 10 − 9 and decreases to half every 10 5 iterations of back-propagation. We use PyTorch framework to implement our A-CubeNet with a GTX 1080Ti GPU.

4.3 Ablation Study
4.3.1 Adaptive dual attention module (ADAM). We follow the same ablation study setting as RAM [24] to compare different attention mechanisms conveniently. Specifically, we set a baseline network with 16 residual blocks. Then we implement the attention mechanisms for image super-resolution with scaling factor ×2. Red and blue colors indicate the best and second best performance, respectively.

Table 1: Performance of our ADAM and other attention mechanisms for image super-resolution with scaling factor ×2.

|                | Baseline | +ADAM | +RAM | +RCAM | +CBAM | +CSAR |
|----------------|----------|-------|------|-------|-------|-------|
| **Params.**    |          |       |      |       |       |       |
| Set5           | 1370K    | 1380K | 1389K| 1379K | 1381K | 1646K |
| Set14          | 37.90    | 38.10 | 37.98| 37.95 | 37.89 | 37.96 |
| BSD100         | 32.17    | 32.25 | 32.17| 32.17 | 32.11 | 32.16 |
| Urban100       | 32.13    | 32.35 | 32.28| 32.24 | 32.01 | 32.29 |
| Manga109       | 38.47    | 38.86 | 38.72| 38.60 | 38.20 | 38.62 |
| **Average**    | 34.40    | 34.63 | 34.53| 34.48 | 34.25 | 34.50 |

4.3.2 Adaptive hierarchical attention module (AHAM). As shown in Table 3, our AHAM obtains a significant improvement for image restoration with a little additional parameters. Therefore, it can be inserted into most CNN-based image restoration methods. Red color indicates the best performance.

Table 3: Performance of our AHAM for image super-resolution with scaling factor ×2.

|                | Baseline | +AHAM |
|----------------|----------|-------|
| **Params.**    |          |       |
| Set5           | 1370K    | 1371K |
| Set14          | 37.90    | 38.11 |
| BSD100         | 32.17    | 33.49 |
| Urban100       | 32.13    | 32.22 |
| Manga109       | 38.47    | 38.86 |
| **Average**    | 34.40    | 34.81 |

4.4 Image Super-Resolution
For image super-resolution, we compare our A-CubeNet with state-of-the-art image super-resolution methods: SRCNN [11], FSRCNN [13], VDSR [22], DRCN [23], LapSRN [25], DRRN [36], MemNet [37], CARN [2], FALSR [6], SelNet [5], MoreMNAS [7], SRMDNF [45], MSRN [27], IDN [21], SRRAM [24], IMDN [20], SRDenseNet [37]. All the quantitative results for various scaling factors (e.g., ×2, ×3, ×4) are reported in Table 4. As shown in Table 4, our A-CubeNet achieves the best PSNR and SSIM results for all scaling factors.

Table 2: Ablation study inside our ADAM. We report results for image super-resolution with scaling factor ×2.

|                | Baseline | +ADAM | +ADAM-C | +ADAM-S | +ADAM-NW |
|----------------|----------|-------|---------|---------|----------|
| **ACAB**       |          |       |         |         |          |
| **ASAB**       |          |       | √       | √       | √        |
| **α**          |          |       | √       |         |          |
| **β**          |          |       |         | √       |          |
| **PSNR**       | 34.40    | 34.63 | 34.48   | 34.51   | 34.54    |

It is worth noting that EDSR [28], D-DBPN [16], RDN [48], RCAN [46], and SAN [9] have higher performance than our A-CubeNet. However, we do not compare with these models because it is meaningless to compare two models with large differences in parameter and depth. Specifically, the maximum number of parameters of our A-CubeNet is only 1524K, which is much smaller than 43M in EDSR, 16M in RCAN and 15.7M in SAN. The network depth of our A-CubeNet (about 40 convolutional layers) is also much shallower than that of RCAN (about 400 convolutional layers) and SAN (about 400 convolutional layers).

We further show visual results of different methods in Figure 5. Visual results under scaling factor ×4 are provided. As shown in Figure 5, other methods fail to recover more image details and output heavy blurring artifacts. Compared to these methods, the SR images reconstructed by our A-CubeNet is closer to the HR image.
in details. These comparisons further demonstrate the effectiveness of our A-CubeNet with the usage of three-dimensional global attention.

Table 4: Quantitative results about image super-resolution. Red and blue colors indicate the best and second best performance, respectively.

| Scale | Model | Params | Set5 | Set4 | BSD100 | Urban100 | Manga109 |
|-------|-------|--------|------|------|--------|---------|---------|
| 2     | Bicubic | - | 33.61/0.9599 | 30.24/0.8688 | 28.56/0.8411 | 26.05/0.8404 | 28.69/0.9599 |
|       | SRCNN  | 37K | 36.61/0.9542 | 32.42/0.9063 | 31.36/0.8879 | 28.09/0.8946 | 31.74/0.9661 |
|       | FSRCNN | 495K | 37.33/0.9587 | 33.03/0.9128 | 31.00/0.8906 | 28.70/0.9540 | 32.72/0.9729 |
|       | VDSR  | 648K | 35.03/0.9500 | 30.38/0.8920 | 28.31/0.8689 | 26.08/0.8688 | 32.44/0.9576 |
|       | LapLRN | 812K | 37.02/0.9590 | 33.08/0.9120 | 31.00/0.8906 | 28.04/0.8940 | 31.72/0.9740 |
|       | DRRN  | 267K | 37.74/0.9591 | 33.20/0.9136 | 32.45/0.8973 | 31.29/0.9108 | 32.79/0.9750 |
|       | MemNet | 478K | 37.76/0.9595 | 32.80/0.9142 | 30.82/0.8787 | 29.21/0.8935 | 32.09/0.9697 |
| 4     | CARN-M | 412K | 37.53/0.9583 | 32.64/0.9141 | 31.92/0.8940 | 28.12/0.9193 | - |
|       | FALSR-B | 528K | 34.61/0.9542 | 30.59/0.8943 | 28.17/0.8697 | 26.00/0.8971 | - |
|       | FALSR-C | 408K | 37.66/0.9596 | 33.26/0.9140 | 31.00/0.8906 | 28.04/0.8940 | 31.74/0.9740 |
| 8     | SeNet  | 974K | 38.70/0.9596 | 33.61/0.9160 | 32.08/0.8944 | 31.42/0.9257 | - |
|       | MoeNet | 469K | 37.78/0.9595 | 32.80/0.9142 | 30.82/0.8787 | 29.21/0.8935 | 32.09/0.9697 |

4.5 Gray Image Denoising

For gray image denoising, we generate the degraded images by adding AWGN noise of different levels (e.g., 10, 30, 50, and 70) to clean images. We compare our A-CubeNet with state-of-the-art gray image denoising methods: BM3D [8], TNRD [4], DnCNN [42], MemNet [37], IRCNN [43], and FFDNet [44]. As shown in Table 5, our A-CubeNet achieves the best PSNR results with all noise levels.

Table 5: Quantitative results about gray image denoising. Red and blue colors indicate the best and second best performance, respectively.

| Method | Kodak24 | BSD68 |
|--------|---------|-------|
|        | 10  | 30  | 50  | 70  | 10  | 30  | 50  | 70  |
| Bicubic | -  | 32.84/0.7764 | 29.78/0.7764 | 26.53/0.7764 | 22.24/0.7764 | -  | -  | -  | -  |
| SRCNN  | 37K | 32.75/0.9542 | 30.49/0.9542 | 28.23/0.9542 | 26.00/0.9542 | -  | -  | -  | -  |
| FSRCNN | 495K | 37.33/0.9587 | 33.03/0.9128 | 31.00/0.8906 | 28.70/0.9540 | -  | -  | -  | -  |
| VDSR  | 648K | 35.03/0.9500 | 30.38/0.8920 | 28.31/0.8689 | 26.08/0.8688 | -  | -  | -  | -  |
| LapLRN | 812K | 37.02/0.9590 | 33.08/0.9120 | 31.00/0.8906 | 28.04/0.8940 | -  | -  | -  | -  |
| DRRN  | 267K | 37.74/0.9591 | 33.20/0.9136 | 32.45/0.8973 | 31.29/0.9108 | -  | -  | -  | -  |
| MemNet | 478K | 37.76/0.9595 | 32.80/0.9142 | 30.82/0.8787 | 29.21/0.8935 | -  | -  | -  | -  |
| CARN-M | 412K | 37.53/0.9583 | 32.64/0.9141 | 31.92/0.8940 | 28.12/0.9193 | -  | -  | -  | -  |
| FALSR-B | 528K | 34.61/0.9542 | 30.59/0.8943 | 28.17/0.8697 | 26.00/0.8971 | -  | -  | -  | -  |
| FALSR-C | 408K | 37.66/0.9596 | 33.26/0.9140 | 31.00/0.8906 | 28.04/0.8940 | -  | -  | -  | -  |
| SeNet  | 974K | 38.70/0.9596 | 33.61/0.9160 | 32.08/0.8944 | 31.42/0.9257 | -  | -  | -  | -  |
| MoeNet | 469K | 37.78/0.9595 | 32.80/0.9142 | 30.82/0.8787 | 29.21/0.8935 | 32.09/0.9697 | 32.09/0.9697 | 32.09/0.9697 | 32.09/0.9697 |

4.6 JPEG Image Deblocking

We also apply our A-CubeNet to reduce image compression artifacts. We use the MATLAB JPEG encoder to generate JPEG deblocking artifacts.
inputs with four JPEG quality settings $q = 10, 20, 30, 40$. For a fair comparison, we perform training and evaluating both on Y channel of the YCbCr color space. We compare our A-CubeNet with SA-DCT [14], ARCNN [10], TNRD [4], and DnCNN [42]. As shown in Table 6, our A-CubeNet achieves the best PSNR and SSIM results with all JPEG quality settings.

In Figure 7, we also show visual results of different methods. Visual results under very low JPEG quality ($q = 10$) are provided. As shown in Figure 7, SA-DCT, ARCNN, TNRD, and DnCNN can remove artifacts to some degree, but these methods also over-smooth some details. Compared with these methods, our A-CubeNet not only removes artifacts well, but also preserves more details because our A-CubeNet obtains more details with consistent structures by considering three-dimensional global attention.

## 5 CONCLUSION

In this paper, we propose a novel attention cube network based on the adaptive dual attention module (ADAM) and the adaptive hierarchical attention module (AHAM) for high-quality image restoration. These two modules constitute the attention cube from spatial, channel-wise and hierarchical dimensions. Our ADAM can capture the long-range contextual information between pixels and channels. As a result, this module successfully expands the receptive field and effectively distinguishes different types of information. Additionally, our AHAM can capture the long-range hierarchical contextual information to combine different feature maps by weights depending on global context. The ADAM and AHAM cooperate to form an “attention in attention” structure, which is very helpful for image restoration. Our method achieves state-of-the-art image restoration results. In the future, this method will be extended to other image restoration tasks.

## ACKNOWLEDGMENTS

This work was supported by the Natural Science Foundation of Guangdong Province (No. 2020A1515010711), the Natural Science Foundation of China (Nos. 61771276) and the Special Foundation for the Development of Strategic Emerging Industries of Shenzhen (No. JCYJ20170817161845824).
REFERENCES

[1] Eirikur Agustsson and Radu Timofte. 2017. Nire 2017 challenge on single image super-resolution: Dataset and study. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). 126–135.

[2] Namkwyuk Ahn, Byeongon Kang, and Kyung-Ah Sohn. 2018. Fast, accurate, and lightweight super-resolution with cascading residual network. In Proceedings of the European Conference on Computer Vision (ECCV). 252–268.

[3] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie-Line Alberi Morel. 2012. Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding. In Proceedings of the British Machine Vision Conference (BMVC). 135.1–135.10.

[4] 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.

[5] Jar-Seok Choi and Munchurl Kim. 2017. A deep convolutional neural network with selection units for super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). 154–160.

[6] Xiangxiang Chu, Bo Zhang, Hailong Ma, Ruijun Xu, Jixiang Li, and Qingyuan Li. 2019. Fast, accurate and lightweight super-resolution with neural architecture search. arXiv preprint arXiv:1901.07281 (2019).

[7] Xiangxiang Chu, Bo Zhang, Juiwun Xu, and Hailong Ma. 2019. Multi-objective reinforced evolution in mobile neural architecture search. arXiv preprint arXiv:1901.03004 (2019).

[8] Chao Dong, Chen Change Loy, and Kyoung Mu Lee. 2016. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 1646–1654.

[9] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. 2019. Second-order attention network for single image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 376–384.

[10] Kai Zhang, Wangmeng Zuo, and Lei Zhang. 2018. FFDNet: Toward a fast and accurate image super-resolution using deep convolutional networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 38, 2 (2015), 295–307.

[11] Chao Dong, Chen Change Loy, and Kyoung Mu Lee. 2016. Accelerating the super-resolution convolutional neural network. In Proceedings of the European Conference on Computer Vision (ECCV). 391–407.

[12] A. Foi, V. Katkovnik, and K. Egiazarian. 2007. Pointwise Shape-Adaptive DCT for image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Transactions on Image Processing 16, 8 (2007), 2080–2095.

[13] Shi Guo, Zifei Yan, Kai Zhang, and Lei Zhang. 2019. Toward reinforced evolution in mobile neural architecture search. arXiv preprint arXiv:1901.07261 (2019).

[14] D. Martin, C. Fowlkes, D. Tal, and J. Malik. 2002. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In Proceedings of the IEEE International Conference on Computer Vision (ICCV). 416–423.

[15] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Andrea Reynolds, Christian Yuan, and Rajen Bhosle. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 6681–6690.

[16] Junfeng Li, Faming Fang, Kangfu Mei, and Guizheng Zhang. 2018. Multi-scale residual network for image super-resolution. In Proceedings of the European Conference on Computer Vision (ECCV). 517–532.

[17] Christian Ledig, Lucas Theis, Ferenc Huszar, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 6681–6690.

[18] Ying Tai, Jian Yang, and Xiaoming Liu. 2017. Image super-resolution via deep recursive residual network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 3147–3155.

[19] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. 2017. Memnet: A persistent memory network for image restoration. In Proceedings of the IEEE International Conference on Computer Vision (ICCV). 4539–4547.

[20] Tong Tong, Gen Lin, Xiejie Liu, and QianPan Gao. 2017. Image super-resolution using dense skip connections. In Proceedings of the IEEE International Conference on Computer Vision (ICCV). 7499–7507.

[21] Kai Zhang, Wangmeng Zuo, and Lei Zhang. 2018. Enhanced deep residual networks for single image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 1664–1673.

[22] Tong Tong, Gen Lin, Xiejie Liu, and QianPan Gao. 2017. Image super-resolution using dense skip connections. In Proceedings of the IEEE International Conference on Computer Vision (ICCV). 391–407.

[23] Ying Tai, Jian Yang, and Xiaoming Liu. 2017. Image super-resolution via deep recursive residual network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 3147–3155.

[24] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. 2017. Memnet: A persistent memory network for image restoration. In Proceedings of the IEEE International Conference on Computer Vision (ICCV). 4539–4547.

[25] Roman Zeyde, Michael Elad, and Matan Protter. 2010. On Single Image Scale-Up Quality Assessment. IEEE Transactions on Image Processing 19, 10 (2010), 2653–2660.

[26] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Andrea Reynolds, Christian Yuan, and Rajen Bhosle. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 6681–6690.

[27] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Andrea Reynolds, Christian Yuan, and Rajen Bhosle. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 6681–6690.

[28] Roman Zeyde, Michael Elad, and Matan Protter. 2010. On Single Image Scale-Up Quality Assessment. IEEE Transactions on Image Processing 19, 10 (2010), 2653–2660.

[29] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Andrea Reynolds, Christian Yuan, and Rajen Bhosle. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 6681–6690.

[30] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Andrea Reynolds, Christian Yuan, and Rajen Bhosle. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 6681–6690.

[31] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Andrea Reynolds, Christian Yuan, and Rajen Bhosle. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 6681–6690.

[32] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Andrea Reynolds, Christian Yuan, and Rajen Bhosle. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 6681–6690.

[33] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Andrea Reynolds, Christian Yuan, and Rajen Bhosle. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 6681–6690.

[34] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Andrea Reynolds, Christian Yuan, and Rajen Bhosle. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 6681–6690.

[35] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Andrea Reynolds, Christian Yuan, and Rajen Bhosle. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 6681–6690.

[36] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Andrea Reynolds, Christian Yuan, and Rajen Bhosle. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 6681–6690.

[37] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Andrea Reynolds, Christian Yuan, and Rajen Bhosle. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 6681–6690.

[38] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Andrea Reynolds, Christian Yuan, and Rajen Bhosle. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 6681–6690.

[39] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Andrea Reynolds, Christian Yuan, and Rajen Bhosle. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 6681–6690.