symmetry

MDPI

Article

Hybrid Domain Attention Network for Efficient Super-Resolution

Qian Zhang, Linxia Feng *, Hong Liang and Ying Yang

School of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China; 20060075@upc.edu.cn (Q.Z.); liangh@upc.edu.cn (H.L.); z19070045@s.upc.edu.cn (Y.Y.)
* Correspondence: z20070087@s.upc.edu.cn

Abstract: Image SR reconstruction methods focus on recovering the lost details in the image, that is, high-frequency information, which exists in the region of edges and textures. Consequently, the low-frequency information of an image often requires few computational resources. At present, most of the recent CNN-based image SR reconstruction methods allocate computational resources uniformly and treat all features equally, which inevitably results in wasted computational resources and increased computational effort. However, the limited computational resources of mobile devices can hardly afford the expensive computational cost. This paper proposes a symmetric CNN (HDANet), which is based on the Transformer's self-attention mechanism and uses symmetric convolution to capture the dependencies of image features in two dimensions, spatial and channel, respectively. Specifically, the spatial self-attention module identifies important regions in the image, and the channel self-attention module adaptively emphasizes important channels. The output of the two symmetric modules can be summed to further enhance the feature representation and selectively emphasize important feature information, which can enable the network architecture to precisely locate and bypass low-frequency information and reduce computational cost. Extensive experimental results on Set5, Set14, B100, and Urban100 datasets show that HDANet achieves advanced SR reconstruction performance while reducing computational complexity. HDANet reduces FLOPs by nearly 40% compared to the original model. ×2 SR reconstruction of images on the Set5 test set achieves a PSNR value of 37.94 dB.

Keywords: super-resolution; attention mechanism; deep learning; symmetry

Citation: Zhang, Q.; Feng, L.; Liang, H.; Yang, Y. Hybrid Domain Attention Network for Efficient Super-Resolution. *Symmetry* **2022**, *14*, 697. https://doi.org/10.3390/sym14040697

Academic Editor: Lorentz Jäntschi

Received: 22 February 2022
Accepted: 24 March 2022
Published: 28 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Image super-resolution (SR) reconstruction aims to generate a high-resolution image by certain technical means, which improves the image quality while magnifying the image. Single image super-resolution (SISR) reconstruction is an asymmetric problem that aims to reconstruct a high-resolution (HR) image by recovering the high-frequency information lost during image acquisition using a low-resolution (LR) observed image, with a difficult forward process and a simple reverse process [1,2]. Deep neural networks, with their powerful nonlinear fitting, feature extraction, and fault tolerance capabilities with high-dimensional data processing, have been deeply applied in various fields, especially in the field of image processing. The technical problems in the field of computer vision have been solved one by one with the increasing maturity of neural network technology, and SISR based on the convolutional neural network has also emerged, including depth-blind super-resolution [3,4]and accelerated super-resolution [5,6]. Smart devices such as tablets and cell phones are becoming increasingly popular, but their computational resources are still limited, which urgently requires the implementation of efficient SR techniques.

Dong et al. [7] first applied CNN networks to single-image super-resolution reconstruction, proposing a three-layer convolutional neural network, SRCNN, and since then, more and more studies have tried to use CNN to implement SISR. The shallow layer neural network has a low level of feature abstraction, while the deeper the layer, the higher the level of feature abstraction [8], and more studies have focused on exploring the effect of "depth" on the expressiveness and performance of the model. Kim et al. [9] proposed VDSR, which increased the number of layers to 20 and improved the hyper-segmentation performance. Lim et al. [8], proposed a wider and deeper network architecture, EDSR, which increased the depth of the network to more than 60 layers and improved the hyper-segmentation performance. RCAN [10] borrows the idea of residuals from ResNet and directly increases the depth of the network to 400 layers through the global residual structure and the local residual structure. The huge improvement in the performance of the RCAN network proves that the network depth is crucial for the SR technique. However, increasing the number of network layers can lead to higher computational costs while improving the SR performance. The widespread popularity of mobile terminals, such as smartphones and tablets, has led to increasing demand for high-resolution images from users, but the hardware resources of these mobile devices are limited to bear the excessive computational cost [7,10]. To address this problem, knowledge distillation migrates knowledge learned from a complex model or multiple models to another lightweight model, making the model lighter without losing performance [11]. Furthermore, pruning methods have shown to be effective at reducing the size of deep neural networks while keeping accuracy almost intact [12]. However, these schemes still involve redundant computations. Image SR reconstruction focuses on recovering lost details in the image, i.e., high-frequency information present in edge and texture regions. Therefore, a smooth area (smoothed area) requires less computational resources [1]. However, these CNN-based SR methods extract features from the original LR input and treat all locations equally, and such a process leads to the redundant computation of low-frequency features.

To address the above problems, a novel framework is proposed in this paper, called a Hybrid Domain Attention Network (HDANet), for single-image super-resolution, which is illustrated in Figure 1. It introduces the self-attention mechanism of the Transformer [13] to improve the inference efficiency of the network. Specifically, in this paper, two parallel self-attention modules are added to the feature extraction process, namely, the spatial self-attention module and the channel self-attention module. The spatial self-attention module captures key image information (e.g., edge and texture regions) and adaptively finds the regions in the image information that need to be attended to. The channel self-attention module captures the interdependencies between different feature channels and adaptively reinforces the important channels to suppress the non-important ones. The two modules work in concert to further refine the redundancy calculation. Such a hybrid domain self-attention mechanism enables the network to bypass low-frequency information and focus on more useful information to effectively accomplish the task of image SR reconstruction [14].

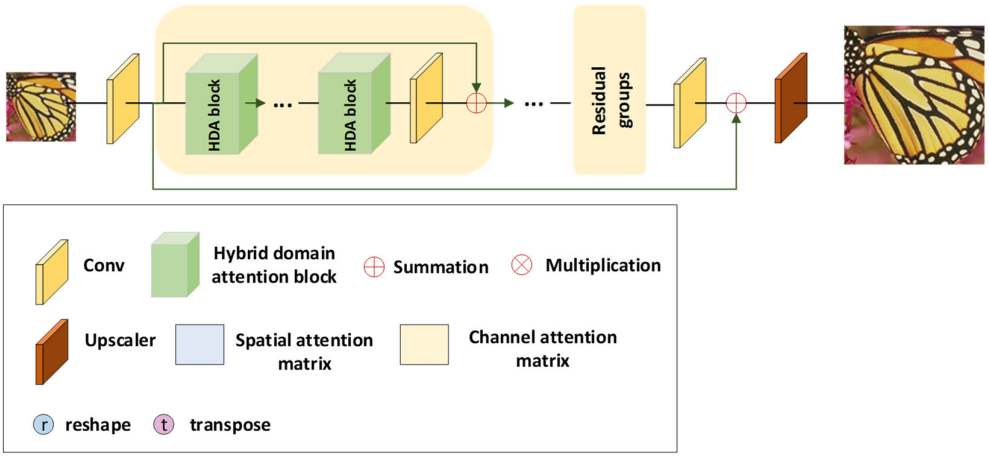


Figure 1. An overview of HDANet.

The main contributions of this paper are twofold: (1) A hybrid domain network with a self-attention mechanism is proposed to dynamically skip redundant computations for efficient image SR reconstruction. Different from existing methods that focus on lightweight networks, this paper improves the efficiency of SR reconstruction by reducing redundant computations. (2) This paper proposes a spatial self-attention module to learn the region that needs to be paid attention to in the image and design a channel self-attention module to allocate the resources among the individual convolutional channels. These two modules work together to locate redundant computations.

2. Related Work

With the development of computer vision and deep learning, image super-resolution has ushered in a wave of development. From traditional image super-resolution methods to deep learning-based super-resolution methods, image super-resolution techniques have been significantly improved. After the attention mechanism is proposed, deep neural networks no longer treat all features equally, but highlight key features and suppress useless features with the help of an attention mechanism to improve the efficiency and accuracy of information processing. This section focuses on the methods related to image SR reconstruction.

2.1. Traditional Methods

Traditional methods to improve image resolution are based on certain rules to calculate the value of the inserted pixels, such as interpolation, reconstruction, and learning [15]. Interpolation refers to the use of known data to predict unknown data, while image interpolation involves predicting the value of a pixel given a pixel point based on the information of the pixel points around it [16]. The common interpolation methods are bicubic interpolation, nearest-neighbor interpolation, and bilinear interpolation. Among them, bicubic interpolation is the most complex and produces the best results to interpolate the object accurately. By the same token, it is also the slowest due to its computational complexity. The interpolation-based method is simple and easy to implement, does not consider the semantic information of the whole image, and only uses the information between the pixels of the low-resolution image to improve the resolution. The interpolation method reconstructs high-resolution images faster and enables real-time super-resolution reconstruction of images. Although the pixel points of images are increased, the quality of super-resolution reconstructed images is lower and often has problems such as mosaic, jaggedness, and blurred edges. Reconstruction methods include maximum a posteriori estimation and iterative back-projection algorithms, which have better reconstruction results than interpolation methods, but the shortcomings are that the models are inefficient

and are influenced by the magnification factor [17]. Local embedding [18] and sparse coding [19] are both learning-based hyper-segmentation algorithms, and the reconstruction quality of these algorithms is the best compared to the first two methods, and they are the mainstream direction of current research.

2.2. Methods Based on Deep Learning

Deep learning is widely used in various fields with its powerful feature extraction and model fitting capabilities, especially in the field of image processing and computer vision [10]. Convolutional neural networks have made a big splash, which has led a large number of researchers to apply deep learning to the field of super-resolution reconstruction. Deep learning-based image super-resolution methods address the shortcoming that traditional methods are difficult to learn deep features of images and achieve the current optimal reconstruction performance and results on several publicly available datasets. According to the different network models, the deep learning-based super-resolution reconstruction methods can be divided into two categories, one is the SR method based on CNN networks and the other is the SR method based on GAN networks. The former is most widely used in the field of SISR reconstruction.

Among them, the SR methods based on CNN network models are further divided into direct-connected models, residual models, dense models, and attention models. The SRCNN algorithm and FSRCNN algorithm are simple direct-connected model structures, which are very easy to implement, but the network training is difficult, and the SR reconstruction effect is not good. The algorithms, such as VDSR, EDSR, DRCN, and DRRN, improve the SR reconstruction effect by increasing the number of network layers with the help of the residual idea of the ResNet network, and these algorithms belong to the residual model structure. To solve the problem of gradient disappearance caused by increasing network depth, SRDenseNet [20] was inspired by residual connectivity and applied densely connected networks to SR reconstruction for the first time, and the image reconstruction quality was greatly improved. The above network model treats all features equivalently and involves the redundant computation of useless information. Inspired by the human visual attention mechanism, RCAN introduces channel attention mechanism into image super-resolution reconstruction, dynamically assigns channel weights to strengthen useful channels while suppressing useless channels, and fully utilizes computational resources [10].

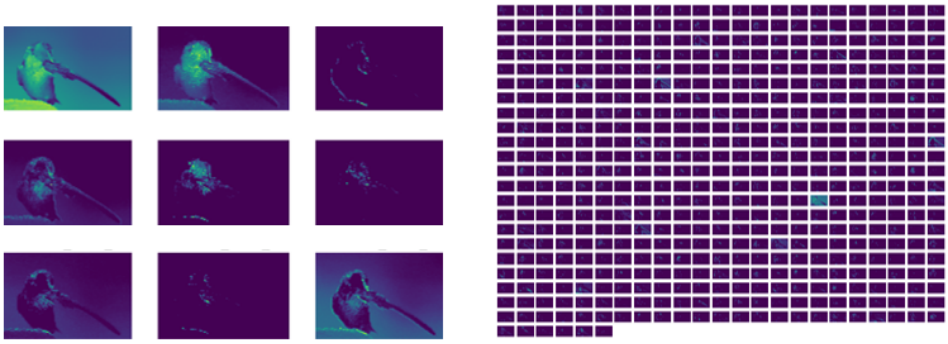
If CNN achieves a new breakthrough in image super-resolution reconstruction, then generative adversarial networks push the quality of image super-resolution reconstruction to a new level, making the recovered images more realistic and natural, such as SRGAN and ESRGAN algorithms. Although GAN-based super-resolution reconstruction methods are more capable of generating high-quality images, their complex network structure and slow learning speed will lead to greater training difficulty.

3. Proposed Method

3.1. Overview

To generate high-quality high-resolution images, the symmetric network shown in Figure 1 is designed in this paper. HDANet consists of three main parts, namely, the shallow feature extraction part, the deep feature extraction part, and the up-sampling part. The shallow feature extraction part consists of a convolutional layer, which is used to extract the edge, shape, and other information of the image. The deep feature extraction part is used to extract more abstract semantic information. Figure 2 shows the results of shallow feature and deep feature visualization. The deep feature extraction part borrows the idea of residuals from the ResNet network and uses multiple serial residual groups, where each residual group contains multiple building blocks, namely HDA blocks and short skip connections. The short skip connection, also known as residual connection, solves the problem of gradient disappearance caused by network depth increase while increasing

the network depth and achieving high-performance image SR [10]. HDA block, as a feature extraction module, mainly consists of two parallel and symmetric attention modules, which are spatial domain self-attention module and channel domain self-attention module, and its specific implementation process is shown in Figure 3. The HAD block enables the network to focus on the focal region, focusing on recovering the high-frequency information of the image and skipping the redundant computation. Meanwhile, the long skip connection is used to pass detailed information from the bottom to the top layer to improve the up-sampling results by fusing shallow features and deep features. The combination of short-skip connections and long-skip connections further improves the performance of image SR. The up-sampling methods are deconvolution, inverse pooling, and interpolation.



(**a**) (**b**)

Figure 2. Visualization of the feature map. (**a**) Shallow feature map; (**b**) Deep feature map.

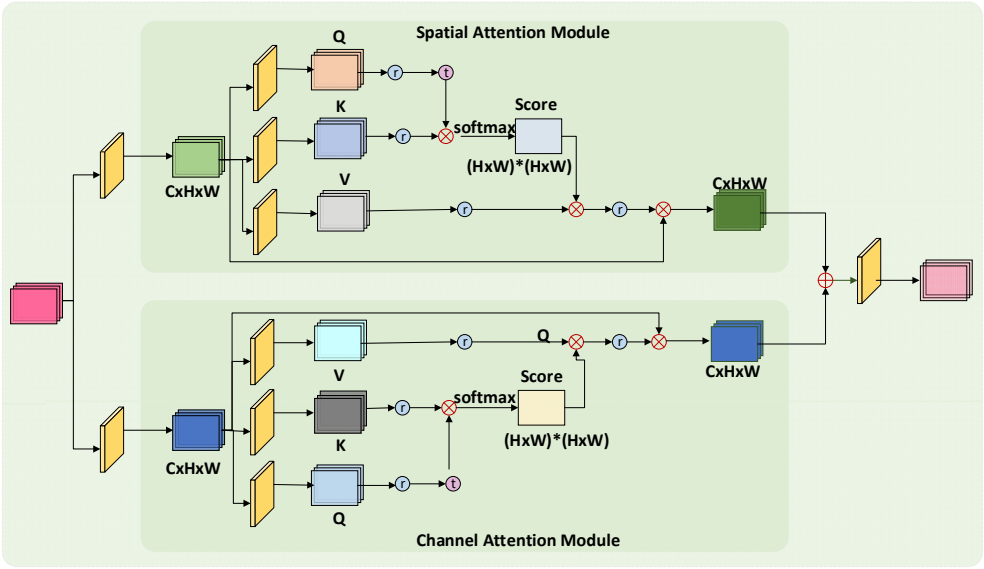


Figure 3. HDANet. An illustration of Spatial Attention Module and Channel Attention Module.

3.2. Hybrid Domain Attention Network

The low-level features have high resolution and contain more location and detail information, but their semantic information is less and more noisy. The high layer features have low resolution and are less perceptive of details, but their semantic information is richer. As shown in Figure 1, a convolutional layer is used to perform shallow feature extraction on the input low-resolution image. The operation can be expressed as follows:

$$Y_0 = F\left(I_{LR}\right) \tag{1}$$

where I_{LR} denotes the low-resolution image and $F(\cdot)$ denotes the convolution operation. To obtain a more abstract feature representation, the obtained low-level features $Y_0 \in R^{C\times H\times W}$ are input to multiple residual groups for deep feature extraction. The operation can be expressed as follows:

$$Y_{RG} = F_{RG}\left(Y_0\right) \tag{2}$$

where $F_{RG}(\cdot)$ indicates multiple residual groups for further deep feature extraction of the input shallow features. The obtained high-level features Y_{RG} are added to the bottom-level features Y_0 in a pixel-by-pixel phase after a convolution layer to achieve feature fusion, and then the fused deep-level features are input to the up-sampling module to complete the image super-resolution reconstruction. The operation can be expressed as follows:

$$I_{HR} = F_{UP}\left(Y_0 + F\left(F_{RG}\right)\right) \tag{3}$$

Most super-resolution reconstruction networks use pixel-based loss function to train the network, but the pixel loss function does not in consideration of the perceptual, texture quality of the image, and the network often outputs perceptually unsatisfactory results, for example, the output image lacks high-frequency details. Moreover, this paper adopts the perceptual loss function proposed by Bruna et al. [8], which can recover richer high-frequency details, and the feature reconstruction loss function proposed by the pre-trained VGG19 network to extract the hyper-resolution reconstructed image and the original high-resolution image in the feature space for the feature mapping, whose expressions are:

$$\mathrm{L}_{ij} = \frac{1}{W_j H_j}\sum_{h=1}^{H_j}\sum_{w=1}^{W_j}\left(\varphi_j\left(I_{HR}\right)_{x,y} - \varphi_j\left(G_{\theta_G}\left(I_{LR}\right)\right)_{x,y}\right) \tag{4}$$

where $G_{\theta_G}(X)$ is the generated high-resolution image and φ_j is the feature map obtained by the *j*-th convolution of the VGG19 network.

3.3. Spatial Self-Attention Module

The computational process of the self-attentive mechanism can be summarized into two processes, the first process is to calculate the weight coefficients based on Query and Key, and the second process is to weight the summation of value based on the weight coefficients. The spatial self-attention module treats each channel feature equally and ignores the information interaction between channels, while the channel self-attention module pools the information within a channel directly globally on average and ignores the local information within each channel [21]. Therefore, in this paper, the self-attention mechanism is applied to both the spatial and channel domains. As shown in Figure 2, the shallow feature extraction is first performed on the images input to the HAD block, and then the obtained feature maps are input to the spatial self-attention module and the channel self-attention module, respectively.

Not all regions that contribute to the image super-resolution reconstruction are equally important, and only the regions related to SR reconstruction are required to be related. The spatial self-attention module is to find the important parts of the network for processing, which is essential to locate the target and perform some transformations or

obtain weights. In this paper, the spatial attention module locates the important regions of the image and skips the redundant computation. It is implemented as follows:

Suppose the feature map input to the spatial attention module is $A \in R^{C\times H\times W}$, and after convolution, the feature map $A_i \in R^{C\times H\times W}\left(i=1,2,3\right)$ is obtained, which is denoted as Query, Key and Score, respectively. The operation can be expressed as follows:

$$A_i = F_i\left(A\right),\left(i=1,2,3\right) \tag{5}$$

where $F_i\left(\cdot\right)$ denotes the convolution operation, and then the reshape operation is performed on $A_i\left(i=1,2,3\right)$, that is, a C-dimensional matrix down to two dimensions, the specific implementation of the C two-dimensional matrix stitching into a two-dimensional matrix. The resulting feature map is $A_i^{reshape} \in R^{C\times N}\left(i=1,2,3\right)$, where, $N=H\times W$. The operation can be expressed as follows:

$$A_i^{reshape} = R\left(A^i\right),\left(i=1,2,3\right) \tag{6}$$

where $R\left(\cdot\right)$ denotes the reshape function, which is used to expand multiple two-dimensional vectors into a single two-dimensional vector. Subsequently, $A_1^{reshape}$ is transposed to obtain $\left(A_1^{reshape}\right)^T$. This operation can be expressed as:

$$\left(A_1^{reshape}\right)^T = T\left(A_1^{reshape}\right) \tag{7}$$

where $T\left(\cdot\right)$ denotes the transpose function that converts the size of the matrix from $C\times N$ to $N\times C$. The resulting matrix after transposition is multiplied by $A_2^{reshape}$ to obtain the weight coefficient score, and the result is denoted as $S \in R^{N\times N}$. This operation can be expressed as:

$$S = \left(A_1^{reshape}\right)^T \times A_2^{reshape} \tag{8}$$

$$S = \begin{pmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{NN} \end{pmatrix} \tag{9}$$

Then, *softmax* is performed on *S*, and the resulting feature map is $S_{softmax}$. This operation can be expressed as:

$$a_{ij} = \frac{\exp\left(x_{ij}\right)}{\sum_{i=1,j=1}^{N}\exp\left(x_{ij}\right)} \tag{10}$$

$$S_{soft\max} = \begin{pmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{pmatrix} \tag{11}$$

After obtaining the feature map $S_{softmax}$, then transpose it to find $\left(S_{softmax}\right)^T$, and multiply the transposed $\left(S_{softmax}\right)^T$ with $A_3^{reshape}$. The result of the multiplication is transposed and then added with the original features *A* input to the HDA block to obtain the spatial attention feature map $E \in R^{C\times H\times W}$. The operation can be expressed as:

$$E = A + \alpha \times T\left(T\left(S_{soft\max}\right)\times A_3^{reshape}\right) \tag{12}$$

where α is the scale factor, representing the degree of integration of local features with global features, initialized to 0 and gradually learned to assign larger weights. From Equation (12), it can be inferred that each position in the obtained spatial feature *E* is a weighted sum of the features of all positions and the original features. Therefore, it has global semantic information.

3.4. Channel Self-Attention Module

Each layer of CNN has multiple convolutional kernels, and each convolutional kernel is for a featured channel, and each channel is related to the important information to a different degree. If the weight represents the relevance of the channel to the important information, then the higher the weight, the higher the relevance. Conversely, the lower the degree of relevance. In this paper, the channel attention module adaptively allocates the resources among each convolutional channel according to the calculated weights, which is mainly implemented as follows:

Suppose the feature map input to the spatial attention module is $B \in R^{C\times H\times W}$, and after convolution, the feature map $B_i \in R^{C\times H\times W}\left(i=1,2,3\right)$ is obtained, which is denoted as Query, Key and Score, respectively. The operation can be expressed as follows:

$$B_i = F_i\left(B\right),\left(i=1,2,3\right) \tag{13}$$

where $F_i\left(\cdot\right)$ denotes the convolution operation, and then the reshape operation is performed on $B_i\left(i=1,2,3\right)$, that is, a C-dimensional matrix down to two dimensions, the specific implementation of the C two-dimensional matrix stitching into a two-dimensional matrix. The resulting feature map is $B_i^{reshape} \in R^{C\times N}\left(i=1,2,3\right)$, where, $N = H\times W$. The operation can be expressed as follows:

$$B_i^{reshape} = R\left(B^i\right),\left(i=1,2,3\right) \tag{14}$$

where $R\left(\cdot\right)$ denotes the reshape function, which is used to expand multiple two-dimensional vectors into a single two-dimensional vector. Subsequently, $B_1^{reshape}$ is transposed to obtain $\left(B_1^{reshape}\right)^T$. This operation can be expressed as:

$$\left(B_1^{reshape}\right)^T = T\left(B_1^{reshape}\right) \tag{15}$$

where $T\left(\cdot\right)$ denotes the transpose function that converts the size of the matrix from $C\times N$ to $N\times C$. The resulting matrix after transposition is multiplied by $B_2^{reshape}$ to obtain the weight coefficient Score, and the result is denoted as $S \in R^{N\times N}$. This operation can be expressed as:

$$S = \left(B_1^{reshape}\right)^T \times B_2^{reshape} \tag{16}$$

$$S = \begin{pmatrix} y_{11} & \cdots & y_{1N} \\ \vdots & \ddots & \vdots \\ y_{N1} & \cdots & y_{NN} \end{pmatrix} \tag{17}$$

Then, *softmax* is performed on *S*, and the resulting feature map is $S_{softmax}$. This operation can be expressed as:

$$b_{ij} = \frac{\exp\left(y_{ij}\right)}{\sum_{i=1, j=1}^{N} \exp\left(y_{ij}\right)} \tag{18}$$

$$S_{soft\max} = \begin{pmatrix} b_{11} & \cdots & b_{1N} \\ \vdots & \ddots & \vdots \\ b_{N1} & \cdots & b_{NN} \end{pmatrix} \tag{19}$$

After obtaining the feature map $S_{softmax}$, then transpose it to find $\left(S_{softmax}\right)^T$, and multiply the transposed $\left(S_{softmax}\right)^T$ with $B_3^{reshape}$. The result of the multiplication is transposed and then added with the original features *B* input to the HDA block to obtain the spatial attention feature map $E \in R^{C \times H \times W}$. The operation can be expressed as:

$$E = B + \beta \times T\left(T\left(S_{soft\max}\right) \times B_3^{reshape}\right) \tag{20}$$

where β is the scale parameter, initialized to a value of 0, and then the weights are gradually increased by learning. From Equation (20), it can be inferred that each channel in the obtained channel features *E* is the sum of the additive persuasion of the features of all channels and the original features.

3.5. Hybrid Domain Attention Module

Most existing methods use compressed neural networks to achieve efficient image SR reconstruction. Commonly used methods for compressing neural networks are knowledge distillation and model pruning. These methods can reduce the complexity of neural network models and improve the inference speed of the models, but still involve redundant computations. For this reason, this paper introduces a hybrid domain attention module (HDAM) in the network to reduce the redundant computations in both dimensions, space, and channel, to obtain efficient image SR reconstruction. Specifically, in each HDAM, a spatial self-attention module and a channel self-attention module are introduced in a symmetric manner, and these two modules generate spatial and channel scores by computing spatial correlation and channel correlation among features, respectively, as shown in Figure 2. The obtained scores are multiplied with the convolved features, thus skipping the redundant calculations in both spatial and channel dimensions. The outputs of the two modules are then summed pixel-by-pixel to achieve feature fusion and further enhance the feature representation. Compared with cascading, the cascaded approach can reduce the GPU footprint and speed up the model operation.

4. Experiment

4.1. Datasets and Implementation Details

The DIV2K [22] dataset, which is widely used in learning-based image SR reconstruction methods, was selected to train the network, which consists of 800 training images and 100 validation images. The performance of this paper's method is evaluated on the B100,

Set5 [23], Set14 [24], and Urban100 datasets, and the SR results are evaluated using the PSNR and SSIM on the Y channel (i.e., luminance) of the transformed YCbCr space. Meanwhile, the method of this paper is evaluated on the Urban100 dataset by comparing the visual results of SISR reconstruction.

We implement our method based on Pytorch [25] with an NVIDIA Titan Xp GPU. For training, eight patches of a size randomly cropped from LR images and the corresponding HR patch are used as input, and then to drive the depth model to the best performance, random cropping and horizontal flipping are used to expand the dataset. We train our model with the ADAM optimizer [26]. β_1, β_2, and ϵ are set to 0.9, 0.99 and 10^{-8}, respectively. The initial learning rate is set to 10^{-4} and then reduced to half after every 200 epochs.

4.2. Effect of Data Preprocessing

To explore the effect of data preprocessing on the performance of image SR reconstruction, this paper conducts ablation experiments on the public dataset Set14. First, two different sets of experiments, denoted as G1 and G2, are designed. Group G1 experiments do not perform any preprocessing on the data, and group G2 experiments perform data enhancement (cropping, horizontal flipping) on the data. Then, to verify the validity of the experiments, the public data set Set14 is chosen as the validation set and the images are processed with 2-fold magnification, while PSNR, SSIM, and FLOPs are used as experimental evaluation metrics to represent the SR reconstruction performance, and the experimental results are shown in Table 1. It can be seen that the G2 group experiments did not perform data amplification, so its computation is slightly lower than that of G2, but its image SR reconstruction quality is inferior to that of the G2 group. These comparisons prove that the model computation will slightly increase after the data is preprocessed, but the image reconstruction quality will be improved. Thus, it can be seen that data preprocessing can improve the generalization ability of the model at a lower computational cost.

Table 1. Ablation study with ×2 SR on Set14.

Group	Data Preprocessing	FLOPs	PSNR	SSIM
G1	×	112.3 G	33.51	0.9169
G2	√	130.4 G	33.59	0.9175

4.3. Effect of SAM and CAM

To explore the effects of the spatial self-attentive module (SAM) and the channel self-attentive module (CAM) on the model, ablation experiments are conducted on the public dataset Set5. First, four different models are designed. Model 1 is the model without either SAM or CAM added and is denoted as model 1. Model 2 and model 3 are the models with only the CAM module added and SAM module added, respectively. Model 4 is the model with both the SAM and CAM modules added and is also the model used in this paper. Then, to verify the validity of the experiment, the public data set Set4 is chosen as the validation set, and the images are processed with 2x magnification, while PSNR, SSIM, Parameters, and FLOPs are used as the experimental evaluation indexes to represent the image reconstruction performance.

The comparison results are shown in Table 2. It can be seen that model 1 without adding SAM and CAM modules has the highest computational cost, and its FLOPs value is almost twice as high as the other models. This is because model 1 processes all features equally when extracting deep features. Model 2 adds only the CAM module to trim redundant channels on all spatial information, so model 2 has the least number of parameters and FLOPs. However, its image SR performance is severely degraded compared to other models (37.83 vs. 37.86), which is because the CAM module filters out the

unimportant channel while filtering out all the spatial information of that channel, which contains important features. Model 3 adds only the SAM module, which filters only the non-important features, so its SR reconstruction performance is higher than that of model 2, but its FLOPs decrease as the performance increases. Our network with both SAM and CAM modules reduces the computational effort to 40% of the original while improving the SR reconstruction performance (37.94 vs. 37.86). This is because the SAM module treats each channel feature equally, ignoring the information interaction between channels, and the CAM module pools the information within a channel directly and globally, ignoring the local information within each channel. Model 4 combines the outputs of the CAM and SAM modules, improving the shortcomings of both modules, further bypassing low-frequency information and focusing on more useful information. The SR reconstruction performance is improved while reducing the computational effort. These comparisons strongly demonstrate the effectiveness of SAM and CAM, and that the best results are achieved when the two work together on the network.

Table 2. Ablation study with ×2 SR on Set5.

Model	SAM	CAM	Params	FLOPs	PSNR	SSIM
model 1	×	×	1.00 M	213.8 G	37.86	0.9582
model 2	×	√	0.67 M	128.3 G	37.83	0.9583
model 3	√	×	1.25 M	139.1 G	37.90	0.9591
model 4	√	√	1.06 M	130.4 G	37.94	0.9598

4.4. Comparison with State-of-the-Arts

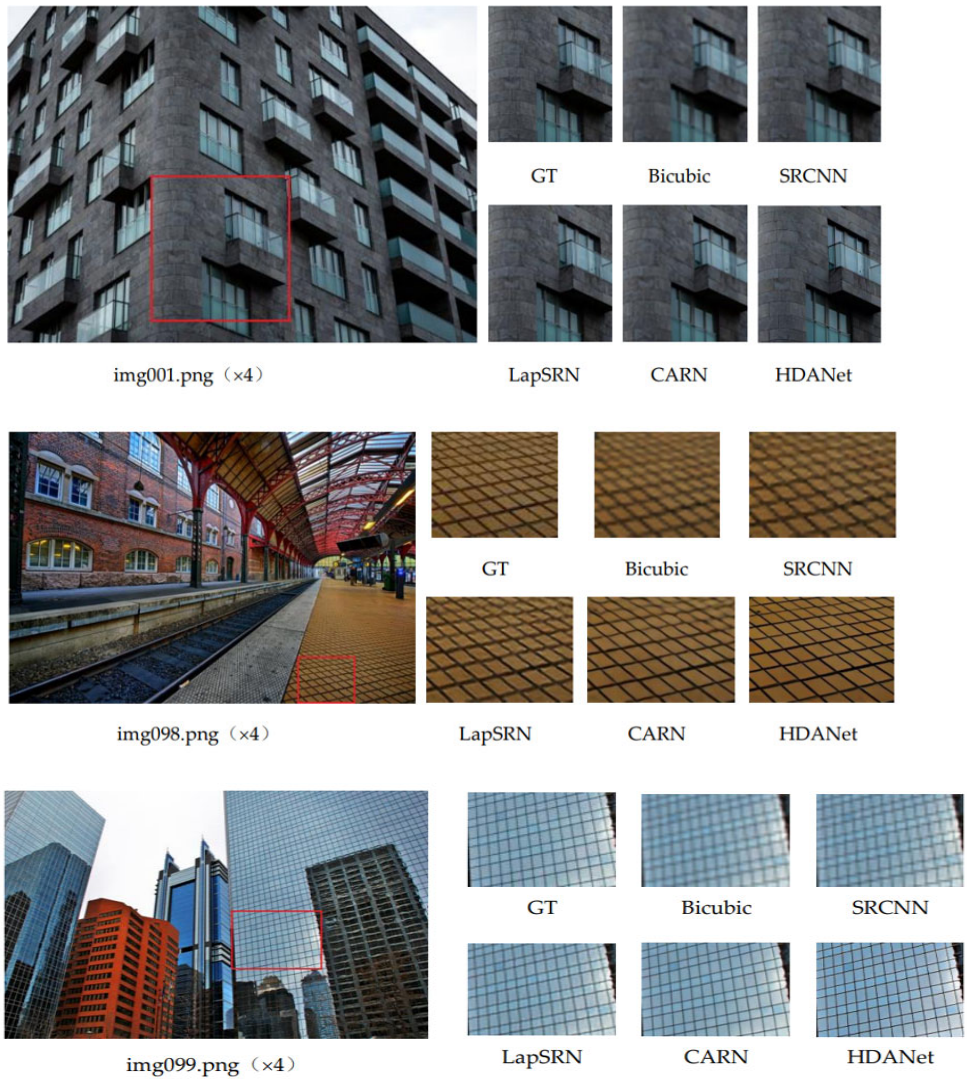
To verify the sophistication and validity of the model, HDANet is compared with Bicubic, RDN-DAQ, SRCNN, VDSR, LapSRN, and CARN in this paper. Quantitative results are shown in Table 2 and visualization results are provided in Figure 3. The implementations of these methods are based on the officially released source code and their experimental results are evaluated using the trained weights. As shown in Table 3, the proposed HDANet achieves the highest PSNR and SSIM on most of the datasets for the current state-of-the-art methods. For example, for ×2 SR, HDANet achieves better performance than CARN with 4a 1% and 33% reduction in FLOPs and parameters, respectively. With comparable model sizes our HDANet achieves better thrust efficiency in terms of FLOPs (66.7 G vs. 118.8 G). With comparable computational volume (40.5 G vs. 52.7 G), HDANet has 30% fewer parameters. For ×2/3/4 SR, our HDANet achieves much higher PSNR and SSIM values than VDSR, but its FLOPs values are much smaller than VDSR. These results show that our approach overcomes the dilemma caused by the performance improvement well, achieving high PSNR performance and low computational cost.

Table 3. Performance evaluation of each network model on Set5, Set14, B100, and Urban100 test sets (×2, ×3, ×4). Best results are shown in bold.

Model	Scale	FLOPs	Params	Set5 PSNR/SSIM	Set14 PSNR/SSIM	B100 PSNR/SSIM	Urban100 PSNR/SSIM
Bicubic	×2			33.66/0.9299	30.24/0.8688	29.56/0.8431	26.88/0.8403
SRCNN [7]		52.7 G	1.55 M	36.66/0.9545	32.42/0.9063	31.36/0.8879	29.50/0.8946
VDSR [9]		612.6 G	0.67 M	37.53/0.9587	33.03/0.9124	31.90/0.8960	30.76/0.9140
LapSRN [27]		29.9 G	0.81 M	37.52/0.9591	33.08/0.9130	31.08/0.8950	30.41/0.9101
CARN [28]		222.8 G	1.59 M	37.76/0.9590	33.52/0.9166	32.09/0.8978	31.92/0.9256
HDANet		130.4 G	1.06 M	**37.94/0.9598**	**33.59/0.9175**	**32.13/0.8988**	**32.17/0.9283**
Bicubic	×3			30.39/0.8682	27.55/0.7742	27.21/0.7385	24.46/0.7349
SRCNN [7]		52.7 G	1.55 M	32.75/0.9090	29.28/0.8209	28.41/0.7863	26.24/0.7989
VDSR [9]		612.6 G	0.67 M	33.66/0.9213	29.77/0.8314	28.82/0.7976	27.14/0.8279
LapSRN [27]		29.9 G	0.81 M	33.82/0.9227	29.87/0.8320	28.82/0.7980	27.07/0.8280

CARN [28]		118.8 G	1.59 M	34.29/0.9255	**30.29/0.8407**	29.06/0.8034	28.06/0.8493
HDANet		66.7 G	1.06 M	**34.35/0.9210**	30.28/0.8405	**29.11/0.8053**	**28.23/0.8531**
Bicubic				28.42/0.8104	26.00/0.7027	25.96/0.6675	23.14/0.6577
SRCNN [7]		52.7 G	1.55 M	30.48/0.8628	27.49/0.7503	26.90/0.7101	24.52/0.7221
VDSR [9]		612.6 G	0.67 M	31.35/0.8838	28.01/0.7674	27.29/0.7251	25.18/0.7524
LapSRN [27]	×4	149.4 G	0.81 M	31.54/0.8850	28.19/0.7720	27.32/0.7270	25.21/0.7560
CARN [28]		90.9 G	1.59 M	32.13/0.8937	28.60/0.7806	**27.58/0.7349**	26.07/0.7837
RDN-DAQ [29]			6.9 M	31.61/.	28.21/.	27.31/.	25.52/.
HDANet		40.5 G	1.08 M	**32.15/0.8941**	**28.61/0.7810**	27.56/0.7338	**26.12/0.7871**

The visual results of image SR reconstruction are shown in Figure 4. The images used for testing are selected from the public dataset Urban100, and the comparison shows that the HDANet method generates clearer images with better details and higher contrast, and HDANet has significant improvements over Bicubic, SRCNN, VDSR, LapSRN, and CARN. For example, in terms of the 4-magnification effect of image img001, it is obvious that the stripes of the buildings in the image generated by HDANet are clearer, while the stripes of the buildings in the image generated by other methods are very blurred and have obvious distortion compared to the original image.



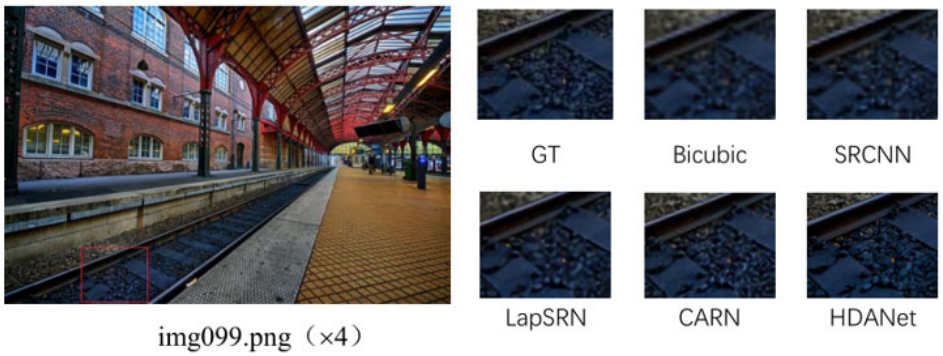


Figure 4. Visual comparison for ×4 SR on Urban100 datasets.

5. Conclusions

In this paper, a symmetric hybrid domain attention network (HDANet) for image SR reconstruction is presented, which utilizes Transformer's self-attention mechanism to suppress unimportant information and skip redundant computations. Specifically, the spatial self-attention module captures the important features of the image, the channel self-attention module suppresses the non-important channels, and then the outputs of the two symmetric modules are summed to further enhance the feature representation, thus locating the redundant computations. The experimental results show that our self-attentive network achieves the highest PSNR and SSIM on most of the datasets. Moreover, HDANet reduces the computational effort by nearly 40% compared to the original model first. This shows that HDANet effectively accomplishes the task of image SR reconstruction and achieves excellent performance while reducing the computational cost. In the next work, we will further investigate how to reduce the computational complexity of the model and how to enhance the robustness of the model to obtain effective improvement in image quality and efficiency.

Author Contributions: Conceptualization, Q.Z. and H.L.; methodology, L.F.; software, L.F.; validation, L.F.; formal analysis, L.F.; investigation, L.F.; resources, L.F.; data curation, L.F.; writing—original draft preparation, L.F.; writing—review and editing, Q.Z.; visualization, Y.Y.; supervision, Q.Z., H.L. and Y.Y.; project administration, Q.Z. and H.L.; funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by The Science Foundation of Shandong Province, grant number ZR2020MF005.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The part of dataset presented in this study are openly available at https://data.vision.ee.ethz.ch/cvl/DIV2K/, accessed on 21 January 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, L.; Dong, X.; Wang, Y.; Ying, X.; Lin, Z.; An, W.; Guo, Y. Exploring sparsity in image super-resolution for efficient inference. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 4917–4926.
2. Liu, H.; Han, J.; Hou, S.; Shao, L.; Ruan, Y. Single image super-resolution using a deep encoder–decoder symmetrical network with iterative back projection. *Neurocomputing* **2018**, *282*, 52–59.
3. Wang, L.; Wang, Y.; Dong, X.; Xu, Q.; Yang, J.; An, W.; Guo, Y. Unsupervised degradation representation learning for blind super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 10581–10590.

4. Zhang, K.; Liang, J.; Van Gool, L.; Timofte, R. Designing a practical degradation model for deep blind image super-resolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 4791–4800.
5. Kong, X.; Zhao, H.; Qiao, Y.; Dong, C. Classsr: A general framework to accelerate super-resolution networks by data characteristic. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 12016–12025.
6. Song, D.; Wang, Y.; Chen, H.; Xu, C.; Xu, C.; Tao, D. Addersr: Towards energy efficient image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 15648–15657.
7. Xie, W.; Song, D.; Xu, C.; Xu, C.; Zhang, H.; Wang, Y. Learning Frequency-aware Dynamic Network for Efficient Super-Resolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 4308–4317.
8. Lim, B.; Son, S.; Kim, H.; Nah, S.; Mu Lee, K. Enhanced deep residual networks for single image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 136–144.
9. Kim, J.; Lee, J.K.; Lee, K.M. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–20 June 2016; pp. 1646–1654.
10. Zhang, Y.; Li, K.; Li, K.; Wang, L.; Zhong, B.; Fu, Y. Image super-resolution using very deep residual channel attention networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 286–301.
11. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
12. Lemaire, C.; Achkar, A.; Jodoin, P.M. Structured pruning of neural networks with budget-aware regularization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9108–9116.
13. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
14. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
15. Dong, C.; Loy, C.C.; He, K.; Tang, X. Learning a deep convolutional network for image super-resolution. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Cham, Switzerland, 2014; pp. 184–199.
16. Hou, H.; Andrews, H. Cubic splines for image interpolation and digital filtering. *IEEE Trans. Acoust. Speech Signal Process.* **1978**, *26*, 508–517.
17. Zhong, M.; Lin, J. A Review of Super-Resolution Image Reconstruction Algorithms. *J. Front. Comput. Sci. Technol.* **2022**, 1–24. https://doi.org/10.3778/j.issn.1673-9418.2111126.
18. Chang, H.; Yeung, D.Y.; Xiong, Y. Super-resolution through neighbor embedding. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), Washington, DC, USA, 27 June–2 July 2004; Volume 1, p. I.
19. Wang, Z.; Liu, D.; Yang, J.; Han, W.; Huang, T. Deep networks for image super-resolution with sparse prior. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 370–378.
20. Tong, T.; Li, G.; Liu, X.; Gao, Q. Image super-resolution using dense skip connections. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4799–4807.
21. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
22. Agustsson, E.; Timofte, R. Ntire 2017 challenge on single image super-resolution: Dataset and study. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 126–135.
23. Bevilacqua, M.; Roumy, A.; Guillemot, C.; Alberi-Morel, M.L. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In Proceedings of the 23rd British Machine Vision Conference (BMVC), Surrey, UK, 3–7 September 2012.
24. Yang, J.; Wright, J.; Huang, T.S.; Ma, Y. Image super-resolution via sparse representation. *IEEE Trans. Image Process.* **2010**, *19*, 2861–2873.
25. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic differentiation in pytorch. In Proceedings of the NIPS 2017 Workshop Autodiff, Long Beach, CA, USA, 9 December 2017.
26. Da, K. A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.

27. Lai, W.S.; Huang, J.B.; Ahuja, N.; Yang, M.H. Deep laplacian pyramid networks for fast and accurate super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 624–632.
28. Ahn, N.; Kang, B.; Sohn, K.A. Fast, accurate, and lightweight super-resolution with cascading residual network. In Proceedings of the European conference on computer vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 252–268.
29. Hong, C.; Kim, H.; Baik, S.; Oh, J.; Lee, K.M. DAQ: Channel-Wise Distribution-Aware Quantization for Deep Image Super-Resolution Networks. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2022; pp. 2675–2684.