Single Image Super-Resolution Based on Directional Variance Attention Network

Parichehr Behjati[a,*], Pau Rodriguez[b], Carles Fernández[d], Isabelle Hupont[c], Armin Mehri[a], Jordi Gonzàlez[a]

[a] *Computer Vision Center, Univ. Autònoma de Barcelona, Bellaterra, Spain.*
[b] *Element AI, a ServiceNow company. Montreal, Canada.*
[c] *Joint Research Centre, European Commission. Seville, Spain.*
[d] *Oxolo GmbH, Hamburg, Germany*

Abstract

Recent advances in single image super-resolution (SISR) explore the power of deep convolutional neural networks (CNNs) to achieve better performance. However, most of the progress has been made by scaling CNN architectures, which usually raise computational demands and memory consumption. This makes modern architectures less applicable in practice. In addition, most CNN-based SR methods do not fully utilize the informative hierarchical features that are helpful for final image recovery. In order to address these issues, we propose a directional variance attention network (DiVANet), a computationally efficient yet accurate network for SISR. Specifically, we introduce a novel directional variance attention (DiVA) mechanism to capture long-range spatial dependencies and exploit inter-channel dependencies simultaneously for more discriminative representations. Furthermore, we propose a residual attention feature group (RAFG) for parallelizing attention and residual block computation. The output of each residual block is linearly fused at the RAFG output to provide access to the whole feature hierarchy. In parallel, DiVA extracts most relevant features from the network for improving the final output and preventing information loss along the successive operations inside the network. Experimental results demon-

*Fully documented templates are available in the elsarticle package on CTAN.

*Corresponding author
Email address: pbehjati@cvc.uab.cat (Parichehr Behjati)

August 24, 2022

strate the superiority of DiVANet over the state of the art in several datasets, while maintaining relatively low computation and memory footprint. The code is available at https://github.com/pbehjatii/DiVANet.

Keywords: single image super-resolution, efficient network, attention mechanism

1. Introduction

Single image super-resolution (SISR) refers to the process of reconstructing a high-resolution image (HR) from its low-resolution version (LR), which offers an opportunity for overcoming resolution limitations in various computer vision applications, such as medical imaging, security and surveillance. The problem of SISR is highly ill-posed procedure, since there are multiple different HR images that may correspond to an identical LR image. To address this problem, a great number of SR models have been proposed, ranging from interpolation methods, to recent learning-based methods [1, 2, 3].

Recently, Convolutional Neural Networks (CNNs) have become the main workhorse to tackle SISR. From SRCNN [4] (with only three convolutional layers) to MDSR [5] (with more than 160), network depth and overall performance have been dramatically growing over time. The increase of depth brings benefits in terms of representation power [6], but at the same time do not take into account the hierarchy of features and their interrelations across the whole architecture. Although SRDenseNet [7] and RDN [8] employ residual dense blocks to fuse different levels of features, the extreme connectivity pattern in their networks not only hinders their scalability when using large widths or depths but also increases computational demands and memory consumption dramatically, hence limiting the use of modern architectures in real-world scenarios. Therefore, it is of crucial importance to design a good lightweight network architecture which effectively computes multi-level feature representations for restoring high quality HR images within the network, yet this remains to be explored.

On the other hand, most existing CNN-based SR methods treat channel-wise

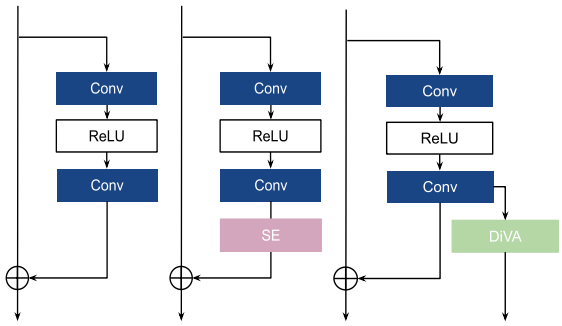


Figure 1: (a) Basic residual block without attention mechanisms. (b) Residual channel attention block proposed in previous works. (c) Our proposed directional variance attention (DiVA), which has its own dedicated computational path.

features equally and lack flexibility in dealing with different types of features across channels (e.g., low- and high-level features), which ends up reducing the efficiency of the network. To this end, researchers have devoted great efforts to expand the application of attention mechanisms to SISR. Taking efficiency into account, the most popular attention mechanism for SR networks is squeeze-and-excitation (SE) attention [9]. However, the SE attention encodes the whole feature map to a single value and hence ignores the spatial relationship between features, which is essential for capturing spatial structures in low-level vision tasks. Moreover, all the previous attention-based approaches performed *in-place* attention within the residual blocks, as in Figure 1 (b). To the best of our knowledge, this work is the first to identify that such *in-place* attention mechanisms may discard relevant details that will no longer be available at deeper levels of the architecture.

To confront these issues, we first present the concept of *directional variance pooling* which leverages both horizontal and vertical variance pooling operations from different spatial dimensions, thus enables the network to attend to larger regions and facilitates capturing longer-range dependencies. Based on the directional variance pooling, we introduce a novel and efficient directional variance attention mechanism (DiVA) specifically designed for low-level

vision tasks. DiVA leverages spatial relationships between features by exploiting higher-order feature statistics in order to enhance features in different channels and spatial regions without incurring significant computation overhead. Furthermore, we propose residual attention feature groups (RAFGs) to enhance representation capability by aggregating informative hierarchical features. The proposed RAFG is composed of two dedicated computational paths: (i) Residual path and (ii) attention path. The residual path consist of a collection of stacked residual blocks whose outputs are linearly fused at different stages of the feature hierarchy, to minimize the information loss during processing through the network and ease the gradient flow for optimization. The attention path is designed to alleviate the loss of information caused by commonly used in-place attention mechanisms. Figure 1(c) illustrates our approach, where the attention module has its own dedicated computational path.

To verify the effectiveness of the proposed approaches, we build a deep but lightweight architecture for SISR named directional variance attention network (DiVANet), illustrated in Figure 2. In summary, these are the main contributions of the paper:

- We propose a lightweight and efficient directional variance attention network (DiVANet) for high-quality image SR. Extensive experiments on a variety of public datasets demonstrate the superiority of the proposed architecture over state-of-the-art models, in terms of both quantitative and visual quality.

- We propose a directional variance attention mechanism (DiVA), specifically optimized for SR, to enhance features in different channels and spatial regions. Such a mechanism allows the network to focus on more informative features and improves discriminative capabilities.

- We introduce a novel procedure called residual attention feature group (RAFG), in which features and attention maps are processed simultaneously, following two independent but parallel computational paths. The

idea is to hierarchically aggregate their respective contributions across the network to facilitate the preservation of finer details.

2. Related Work

A number of SISR methods, different learning mechanisms, and various network architectures have been proposed in the literature. Here, we focus our discussion on the approaches that are most related to our work.

2.1. Evolution of Architectures for SISR

Dong et al. [4] pioneered the field of SR with neural networks, proposing SRCNN, a three layer CNN which outperformed traditional algorithms. Later, Kim et al. [10] first pushed the depth of SR network to 20 with residual learning, outperforming SRCNN by a large margin. At the same time, Kim et al. [11] presented a deeply-recursive convolutional network (DRCN), which applied recursive learning to the SR problem. Later, Lim et al. [5] employed residual blocks to construct a deeper network by removing unnecessary modules (e.g., batch normalization) from the residual blocks. By using effective building modules, image SR networks became deeper and yielded better performance. Furthermore, in order to employ hierarchical features from all the convolutional layers in deep networks, dense blocks started being employed in several SR architectures [8, 7]. Later, MSRN [12] proposed to explore the multi-scale information of LR images. Li et al. [13] incorporated the feedback mechanism into network designs for exploiting both LR and HR signals jointly. Although these existing deep learning-based approaches have made considerable progress to improve SR performance, they demand substantial memory and computational resources.

Numerous lightweight models have been proposed to alleviate the aforementioned computational burden. For example, Ahn et al. [14], Qin et al. [15] revisited DRCN by combining recursive structures and residual blocks so as to improve performance with fewer parameters. Likewise, Behjati et al. [16] and Jiang et al. [17] also joined residual connections and recursive layers to

reduce the computational cost. Later, Chu et al. [18] introduced Neural Architecture Search (NAS) strategies to automatically build an SR model given certain constraints. Meanwhile, Liu et al. [19] proposed an information multi-distillation block that extracted features at a granular level with the channel splitting strategy. Luo et al. [20] proposed lattice blocks that applied so-called butterfly structures to combine residual blocks. More recently, Lu et al. [21] proposed a high preserving block to reduce computational cost and refine high-frequency information. Wang et al. [22] proposed an attentive feature block to utilize auxiliary features of previous layers for facilitating features learning of the current layer. Li et al. [23] proposed a linearly-assembled pixel-adaptive regression network, which casts the direct LR to HR mapping learning into a linear coefficient regression task. Recently, to simplify the challenges of directly super-resolving details, some authors adopted the progressive structure to reconstruct HR images in a stage-by-stage upscaling manner [24, 25, 26]. Although all the aforementioned works demonstrate that lightweight SR networks are capable of providing good trade-offs between performance and number of parameters, there is still room for improvement in terms of performance.

2.2. Attention Mechanisms in SISR

The aim of introducing attention mechanisms to neural networks is to recalibrate the feature responses towards the most informative and important components of the inputs [9]. Attention mechanisms have been successfully applied to deep CNN-based image enhancement methods and, more particularly, to SISR. Zhang et al. [27] first incorporated an existing squeeze-and-excitation (SE) channel attention mechanism [9] into SR and pushed the state-of-the-art performance of SISR. Later, Hu et al. [6] combined the SE attention and a spatial attention mechanism. More recent works, such as [28, 29, 30], extend this idea by either adopting different spatial attention mechanisms or designing advanced attention blocks.

Non-local or self-attention modules are also popular due to their capability of building spatial or channel-wise attention. Mei et al. [31] proposed local and

non-local attention blocks to extract features that capture the long-range dependencies between pixels and pay attention to more challenging parts. Similarly NLSA[32] and NAAN [1] exploit non-local attention mechanisms to capture long-distance spatial contextual information. Nevertheless, these methods notoriously consume large amounts of memory to compute large affinity matrices at each spatial position, and are often adopted only in large models, thus not being suitable for real-world scenarios.

In contrast to previous approaches, the attention mechanism proposed in this paper considers a more efficient way of capturing spatial information and channel-wise relationships to augment the feature representation for SR networks, hence improving performance while still being lightweight.

3. Directional Variance Attention Network (DiVANet)

In this section, we first provide an overview of the proposed directional variance attention network (DiVANet) for SISR. Then, we present the detailed configuration of its two main components: the directional variance attention blocks (DiVA) and the residual attention feature groups (RAFGs).

3.1. Network Overview

As shown in Figure 2, the overall architecture of DiVANet consists of a non-linear mapping module and a final reconstruction module. Let's denote I_{LR} and I_{SR} the input and output of DiVANet, respectively. As recommended in [5], we apply only one 3×3 convolutional layer to extract the initial features F_0 from the LR input image:

$$F_0 = \text{Conv}_{3\times3}(I_{LR}). \tag{1}$$

Next, extracted features F_0 are sent to the non-linear mapping module (NLM) which computes useful representations of the LR patch in order to infer its HR version:

$$F = H_{NLM}(F_0), \tag{2}$$

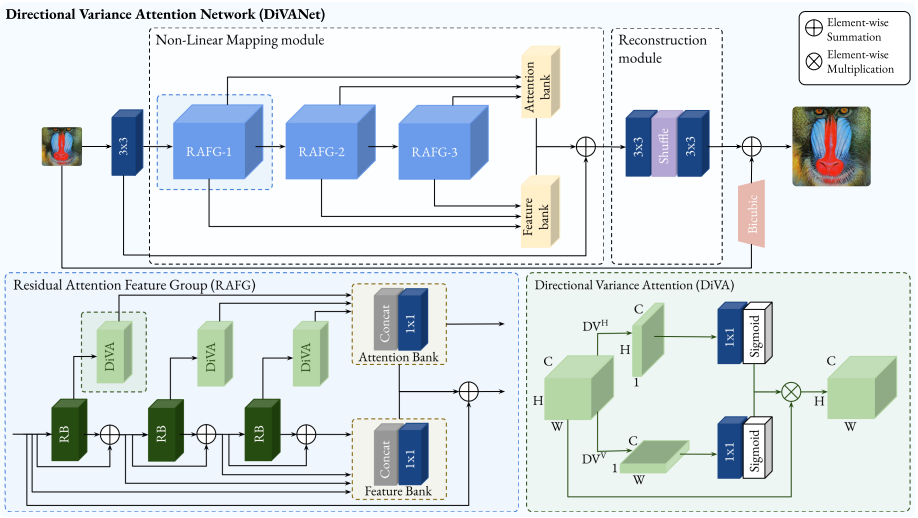


Figure 2: **Top**: Proposed directional variance attention network (DiVANet) architecture for SISR. **Bottom**: residual attention feature group (RAFG), containing residual blocks (RB) and the proposed directional variance attention (DiVA).

where F is the output of the non-linear mapping module H_{NLM} (further detailed in Section 3.3), containing high resolution features.

Finally, a reconstruction module with two convolutional layers and a pixel-shuffle layer upsamples the features to the HR size. In addition, we incorporate a global connection path H_{UP} to grant access to the original LR information and facilitate the back-propagation of the gradients, in which only a bicubic interpolation is applied to the input I_{LR}. Therefore, we obtain:

$$I_{SR} = H_{REC}(F) + H_{UP}(I_{LR}). \tag{3}$$

where $H_{REC}(.)$ is the reconstruction module, and I_{SR} is the final output of the network.

To optimize DiVANet, we adopt L_1 loss as a cost function for training. Given a training set with N pairs of LR images and HR counterparts, denoted by $\{I^i_{LR}, I^i_{HR}\}_{i=1}^N$, the network is optimized to minimize the L_1 loss function:

$$L_1(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} \left\| I_{SR} - I_{HR} \right\|_1, \tag{4}$$

where $\boldsymbol{\theta}$ denotes the parameter set.

3.2. Directional Variance Attention (DiVA)

To provide a clear description of the proposed DiVA mechanism, we first revisit the SE attention, which is widely used in SR networks.

3.2.1. Background: Squeeze-and-Excitation Attention

The well-known Squeeze-and-Excitation attention mechanism (SE) is employed in many image classification tasks. Structurally, an SE block is divided into two processes: *Squeeze* and *excitation*. Let $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_C}] \in \mathbb{R}^{C \times H \times W}$ be an input. Then, the squeeze step for the c-th channel can be formulated as follows:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i, j), \tag{5}$$

where $x_c(i, j)$ is the value at position (i, j) of the c-th channel, and z_c is the value obtained for channel c after average pooling. The main purpose of the squeeze operation is to extract and condense global information per channel.

The second step, excitation, aims to leverage inter-channel dependencies to enhance or decrease the response of individual channels. This operation can be formulated as:

$$\hat{\mathbf{X}} = \mathbf{X} \cdot \sigma(\hat{\mathbf{z}}), \tag{6}$$

where $\cdot$ refers to channel-wise multiplication, σ is the sigmoid function, and $\hat{z}$ is the result generated by a transformation function, which is formulated as follows:

$$\hat{\mathbf{z}} = \mathbf{W_2}(\delta(\mathbf{W_1}(\mathbf{z}))). \tag{7}$$

Here, δ denotes the ReLU function. $\mathbf{W_1}$ and $\mathbf{W_2}$ are fully connected layers that set the channel dimension of features to $\frac{C}{r}$ and C, respectively.

The SE block has been widely used in various SR networks [33, 27, 20], and proven to be a key component for achieving state-of-the-art performance. However, SE attention generally suffers from two basic problems, both stemming from the global average pooling operation. First, it re-weights the importance

of each channel by only modeling channel relationships, but neglects spatial information which would be advantageous to enhance image details. Second, it only exploits channel-wise statistics of features by global average pooling, while ignoring higher-order statistics of channels, thus hindering the discriminative ability of the network [34].

Inspired by the above observations, we propose the use of directional variance attention (DiVA) module that captures not only cross-channel but also spatial information, while considering higher-order feature statistics.

3.2.2. Directional Variance Attention Block

Figure 2 (bottom) depicts the proposed DiVA block. In order to encourage attention blocks to capture long-range interactions spatially while keeping a low computational footprint, we factorize two-dimensional global average pooling as formulated in Equation (5) into a pair of 1D feature encoding operations. We first feed $\mathbf{X}$ into two parallel pathways, to encode each channel along either the horizontal or the vertical dimension. We define the horizontal directional average pooling DA^h of the c-th channel at height h as:

$$z_c^h = DA^h(x_c) = \frac{1}{W} \sum_{0<i\leq W} x_c(i,h) \tag{8}$$

Similarly, the vertical directional average pooling DA^w of the c-th channel at width w is defined as:

$$z_c^w = DA^w(x_c) = \frac{1}{H} \sum_{0<j\leq H} x_c(w,j) \tag{9}$$

The described operations process features along two spatial directions individually. This is different from the squeeze operation in channel attention methods (Eq. 5), which produces a single feature vector and dismissing the spatial relationship between features. These two transformations also enable the attention block to build relationships among multiple spatial positions within the input feature. However, since image SR ultimately aims at restoring high-frequency components of images, it is important to extract statistics that can

effectively represent the characteristics of each channel. To this end, we replace directional average pooling with directional variance pooling, a higher-order feature statistic. Thus, we define the horizontal directional variance pooling DV^h of the c-th channel at height h as:

$$z_c^h = DV^h(x_c) = \frac{1}{W} \sum_{0 \le i < W} (x_c(i,h) - DA^h(x_c))^2 \tag{10}$$

Similarly, the vertical directional variance pooling DV^w of the c-th channel at width w is defined as:

$$z_c^w = DV^w(x_c) = \frac{1}{H} \sum_{0 \le j < H} (x_c(w,j) - DA^w(x_c))^2 \tag{11}$$

As described above, Equations 10 and 11 facilitate a global receptive field and encode spatial information by exploiting directional variance pooling. These two feature maps with spatial information are then separately encoded into two attention maps that can be complementarily applied to the input feature map to enhance features in different channels and spatial regions.

Specifically, given the aggregated feature maps produced by Equations 10 and 11, two 1×1 convolutional F_h and F_w are utilized to separately transform z^h and z^w, yielding:

$$\mathbf{a}^h = \sigma(F_h(\mathbf{z^h})), \tag{12}$$

$$\mathbf{a}^w = \sigma(F_w(\mathbf{z^w})). \tag{13}$$

Recall that σ is the sigmoid function. $\mathbf{a}^h \in \mathbb{R}^{C \times H}$ and $\mathbf{a}^w \in \mathbb{R}^{C \times W}$ are used as attention weights, respectively. Finally, the recalibrated output can be written as:

$$y_c(i,j) = x_c(i,j) \ \cdot \ a_c^h(i) \ \cdot \ a_c^w(j). \tag{14}$$

Compared to other works like [32, 1], which require a considerably large amount of computation to build relationships between each pair of locations, DiVA is substantially lightweight and can capture long-range spatial dependencies and exploit inter-channel dependencies simultaneously. Furthermore, unlike SE, which relies on global average pooling to exploit first-order statistics,

the proposed attention mechanism adaptively learns feature inter-dependencies by exploiting higher-order statistics that represent the characteristics of each channel. The DiVA mechanism helps to emphasize informative representations and improve discriminative learning ability. Sections 4.2.1 and 4.2.2 provide a more detailed analysis on the performance of our approach against existing attention-based methods.

3.3. Residual Attention Feature Group (RAFG)

The Residual Attention Feature Group (RAFG) is the core of the non-linear mapping module. It is designed to attend and preserve higher frequency details across the entire network. As shown in Figure 2, it is composed of two dedicated computational paths: (i) residual path and (ii) attention path. We detail each of these below.

Residual Path. It has been demonstrated that stacked residual blocks can be useful to construct deep CNNs [5]. However, in image SR, very deep networks built in such way would suffer from training difficulty and hardly gain performance [34]. This is because the residual features from initial blocks need to traverse a long path to propagate until the final blocks, as these features are repeatedly merged with identity features to form more complex ones during transmission. Therefore, highly representative features are mostly computed locally and lost in residuals during network propagation.

In this work, we address this issue from a different perspective. Instead of designing a complex architecture with various skip and dense connections, we propose to linearly combine the residual features at a feature bank which is built by aggregating all the features from previous blocks. Figure 2 (bottom) shows the details of the proposed RAFG. It contains three residual blocks, the output of which are respectively sent to the end of the RAFG, and then concatenated together. However, aggregating residual features from different residual blocks directly by systematic concatenation is problematic. Thus, we incorporate a 1×1 convolutional layer to project them into a common space after feature aggregation. In this way, information from preceding residual blocks can

be hierarchically propagated bottom-up without degradations or interference, leading to a more discriminative feature representation.

Using hierarchical feature banks enables us to exploit residual features non-locally, i.e. the information contained in them is not local in the sense that it does not belong to a single residual block. In other words, these feature banks capture detailed information from features across the whole architecture, thus reducing feature degradation and boosting the network's overall representational ability.

Attention Path. The features extracted by a deep neural network contain different types of information at each channel. If we are able to increase the network's sensitivity to specific channels that contain useful information for image reconstruction, the performance of the network will be improved.

Previous approaches performed channel attention *in-place* within the residual blocks to further boost the representational ability of the network [9]. This usually implied an element-wise product between the attention output and the residual block output. However, such in-place channel attention may discard relevant details which will no longer be available at deeper levels of the architecture, so we propose to keep a separate computational path to aggregate computations resulting of attention operations, independent from the aggregation of residual features, and parallel to it.

As shown in Figure 2 (bottom), the output of each residual block is directly sent to a DiVA block before element-wise addition. Specifically, the attention outputs are then aggregated to an attention bank followed by a 1×1 convolutional layer. Finally, the outputs of feature and attention banks are combined together by element-wise addition at the RAFG output, so that they are able to attend to relevant features while preserving higher frequency details across the whole network, further improving the representational ability.

4. Experimental Results

In this section, we first conduct an ablation study to validate the effectiveness of each proposed component. Then, we systematically compare DiVANet with state-of-the-art SISR algorithms on five commonly used benchmark datasets.

4.1. Settings

Datasets and Metrics. Following [3], we use 800 high-quality images from the DIV2K dataset [35] for training. We evaluate our models on several benchmark datasets: Set5 [36], Set14 [37], B100 [38], and Urban100 [39], and Manga109 [40], each with diverse characteristics. All results are evaluated with two commonly used metrics: PSNR (*peak-signal-to-noise-ratio*) and SSIM (*structural similarity index*). To keep the consistency with previous works, quantitative results are evaluated on the luminance channel (Y). Furthermore, we also adopt the Perceptual Index (PI) [41], which can avoid the situation where over-smoothed images may present a higher PSNR or SSIM when the performances of two methods are similar. The lower PI value denotes the better perceptual quality.

Degradation models. To fairly compare against existing works, we adopt bicubic downsampling (denoted as **BI**) as our standard degradation model for generating LR images from ground truth HR images at $\times 2$, $\times 3$ and $\times 4$ scales. Moreover, to comprehensively illustrate the efficacy of the proposed method, we further adopt two other multi-degradation models as in [8]. We define **BD** as a degradation model that performs bicubic downsampling on HR images at $\times 3$ scale, and then blurs them with a Gaussian kernel of size 7×7 and standard deviation 1.6. Additionally, we further produce LR images in a more challenging way: we first bicubic downsample HR images with scaling factor $\times 3$ and then add Gaussian noise with noise level 30 (denoted as **DN**).

Implementation details. During training, data augmentation is carried out by means of random horizontal flips and 90° rotation. At each training mini-batch, 64 LR RGB patches of size 64×64 are provided as inputs. We train our models using an ADAM optimizer with learning rate 10^{-3}. The learning rate is

decreased by half every 2×10^5 iterations. Our network has been implemented using PyTorch, and trained on a NVIDIA RTX 3090 GPU. We implement two lightweight models in this paper, namely DiVANet and DiVANet-S. DiVANet consists of 3 RAFGs, each with three residual blocks and three DiVA modules. In this implementation of DiVANet, all convolutional layers have 64 filters with kernel size 3×3, except for the 1×1 convolutional layers in the feature and attention banks. DiVANet-S has a similar structure as DiVANet, except the parameters of the residual blocks within each RAFG are shared.

4.2. Ablation Study

To further investigate the behavior of the proposed methods, we analyze their effect on model training via an ablation study. We first demonstrate the effectiveness of the proposed DiVA mechanism. Then, we conduct an ablation experiment to study the effect of the essential components of our architecture.

4.2.1. Comparing Pooling Methods

To demonstrate the advantages of the proposed directional variance pooling (*D-Var*) over other pooling methods, we attempt to replace it with: global average pooling (*Avg*), global variance pooling (*Var*), and directional average pooling (*D-Avg*). We do not employ maximum pooling in this experiment, since Mehri et al. [29] already demonstrated that it degrades SR performance. Additionally, we will also compare to a baseline implementation, identical to the proposed method except for the absence of the attention path (*Baseline*).

The results of this experiment are listed in Table 1. It can be seen that exploiting higher-order statistics (global variance pooling) is more effective than first-order ones (global average pooling). Furthermore, when we change from global average pooling to directional average pooling the performance increases by up to +0.14dB on average, with a negligible increase in the number of parameters. This is mainly because the proposed attention with directional average pooling simultaneously captures longer-range spatial interactions and exploits inter-channel dependencies, further improving the representational ability of the

Table 1: Effect of different pooling methods for DiVA. Average PSNR on five benchmark datasets with scale factor ×4 are shown.

Methods	Baseline	+ Avg	+ Variance	+ D-Avg	+ D-Var
Params	815K	902K	902K	939K	939K
Set5	32.18	32.33(+0.15dB)	32.35(+0.17dB)	32.37(+0.19dB)	**32.41**(**+0.23dB**)
Set14	28.59	28.62(+0.03dB)	28.64(+0.05dB)	28.66(+0.07dB)	**28.70**(**+0.11dB**)
B100	27.56	27.59(+0.03dB)	27.60(+0.04dB)	27.61(+0.05dB)	**27.65**(**+0.09dB**)
Urban100	26.09	26.30(+0.11dB)	26.34(+0.15dB)	26.35(+0.26dB)	**26.42**(**+0.33dB**)
Manga109	30.50	30.60(+0.10dB)	30.63(+0.13dB)	30.65(+0.15dB)	**30.73**(**+0.23dB**)

network. However, it only leverages first-order statistics of the features. Finally, when directional variance pooling is applied, the attention mechanism enhances features in different channel and spatial regions by exploiting higher-order feature statistics and attains the best performance in all datasets (PSNR: +0.20dB on average). This improvement is more prominent for the B100 and Urban100 datasets. Since B100 and Urban100 present contents with higher structural complexity, it can be interpreted that the attention with directional variance pooling can help the network to exploit more informative features and enhance its discriminative learning ability. These results demonstrate the superiority of using directional variance pooling over other pooling strategies.

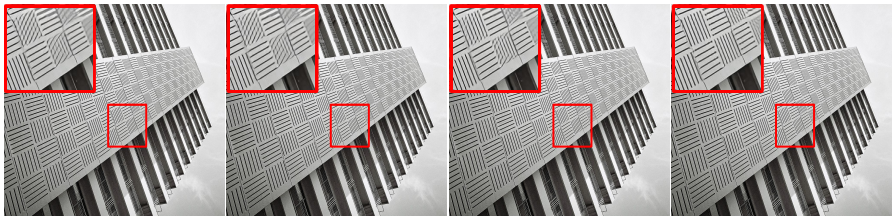
4.2.2. Comparing attention schemes

To demonstrate the effectiveness of our proposed attention mechanism, we use DiVANet as the basic network, and then replace our attention scheme with Squeeze-and-Excitation channel attention (SE) [9], spatial attention (SA) [6] and channel-wise spatial attention residual (CSAR) [6]. Note that we only compare the DiVA scheme with equally lightweight attention mechanisms. As shown in Figure 1 (c), the channel attention module feeds from the residual block but splits out from it through a dedicated computational path; we have trained all the aforementioned variations using this same architectural pattern.

Table 2 compares the performance of these attention methods in terms of PSNR. We see that all the methods with an attention mechanism obtain better performance than the one without it (*Baseline*). This indicates that attention contributes importantly in terms of performance. As reported in Table 2, inte-

Table 2: Average PSNR obtained with DiVANet when using different attention mechanisms on five benchmark datasets (scale factor ×4).

Name	Baseline	+ SE	+ SA	+ CSAR	+ DiVA (Ours)
Params	815K	902K	902K	940K	939K
Set5	32.18	32.33	32.29	32.35(+0.17dB)	**32.41**(**+0.23dB**)
Set14	28.59	28.63	38.60	28.66(+0.07dB)	**28.70**(**+0.11dB**)
B100	27.56	27.58	27.56	27.60(+0.04dB)	**27.63**(**+0.07dB**)
Urban100	26.09	26.31	26.30	26.33(+0.24dB)	**26.42**(**+0.33dB**)
Manga109	30.50	30.62	30.60	30.64(+0.14dB)	**30.73**(**+0.23dB**)



SE SA CSAR DiVA (Ours)

Figure 3: Visual comparison of SR results using DiVANet with different attention mechanisms (×4 scale factor).

grating SE or SA attention into DiVANet moderately improves the SR performance. Moreover, when CSAR [6] is utilized the performance is further boosted (+0.13dB on average), demonstrating the effectiveness of combining channel-wise and spatial attention. On the other hand, the model using the proposed DiVA yields the best performance (PSNR: +0.20dB on average). Compared to CSAR, DiVA efficiently encodes both cross-channel and spatial information, attaining better performance with fewer parameters. These experiments justify that with comparable learnable parameters, the proposed DiVA attention is more helpful for image SR. Figure 3 shows a visual comparison of networks with different attention mechanisms. It can be observed that the network with our proposed attention obtains better visual quality and restores more image details than other methods.

4.2.3. Influence of model size

We also investigate the effectiveness of DiVA attention in networks with different model sizes. For comparison, we select two state-of-the-art networks, SRDenseNet [7] and RCAN [27], whose number of parameters are 2,015K and

Table 3: The results of adding DiVA in different networks. Average PSNR on five benchmark datasets with scale factor ×4 are shown.

Methods	SRDensNet	SRDensNet+DiVA	RCAN	RCAN+DiVA
Multi-Adds	390G	392G	916.9G	964.1G
Params	2,015K	2,052K	15,592K	15,629K
Set5	32.02	**32.14**(**+0.12dB**)	32.68	**32.78**(**+0.10dB**)
Set14	28.50	**28.61**(**+0.11dB**)	28.95	**29.04**(**+0.09dB**)
B100	27.53	**27.63**(**+0.10dB**)	27.55	**27.64**(**+0.09dB**)
Urban100	26.05	**26.16**(**+0.11dB**)	27.05	**27.14**(**+0.09dB**)
Manga109	30.41	**30.55**(**+0.14dB**)	31.62	**31.73**(**+0.11dB**)

15,592K, respectively. Then, DiVA is performed *in-place*, either at the end of the SRDenseNet blocks (SRDenseNet+DiVA) or replacing RCAN's channel attention (RCAN+DiVA). For fair comparison, all networks are trained on their default settings. Table 3 shows the results of experiments conducted on five datasets at scale ×4. It can be observed that SRDenseNet+DiVA and RCAN+DiVA respectively achieve better performance than the original SRDenseNet and RCAN networks. These experimental results indicate that DiVA is also effective in heavier models, increasing the performance by 0.11dB on average.

4.2.4. Effect of the RAFG

This section discusses the effect of each of the two dedicated computational paths in the proposed RAFG: residual path and attention path.

Residual path. In this experiment, we use a ResNet architecture (*Baseline*) without the RAFG computational path, i.e., a regular architecture composed of several stacked residual blocks. Then, we add hierarchical feature banks to this baseline, denoting it as *Baseline+FB*. Table 4 shows the results of the experiments conducted on the five datasets with scale ×4. The small change in number of parameters between *Baseline* and *Baseline+FB* is due to adding feature banks, which contain 1×1 convolutions.

As reported in Table 4, the PSNR of *Baseline* is 25.73dB on Urban100, which is a strong baseline for lightweight SISR methods. When deploying our hierarchical bank of residuals, the PSNR increases to 26.09dB. In addition, we

Table 4: Average PSNR for a regular ResNet architecture (Baseline) vs one using the proposed feature banks on five benchmark dataset with $\times 4$ scale factor.

Methods	Baseline	Baseline+FB	HRFFN [15]	HDRN [17]	SRDenseNet [7]
Multi-Adds	50G	54G	61.64G	255.8G	390G
Params	749K	815K	871K	867K	2,015K
Set5	31.85	**32.26**(**+0.41dB**)	32.19	32.23	32.02
Set14	28.36	**28.60**(**+0.24dB**)	28.57	28.58	28.50
B100	27.30	**27.55**(**+0.22dB**)	27.53	27.53	27.53
Urban100	25.73	**26.13**(**+0.40dB**)	26.08	26.09	26.05
Manga109	30.29	**30.49**(**+0.20dB**)	30.36	30.43	30.41

compare our method with HRFFN [15], HDRN [17] and SRDenseNet [7]. For example, HDRN and SRDenseNet combines residual skip connections with dense connections to utilize all the hierarchical features from all the convolutional layers, hence being very computationally intensive due to this dense feature fusion strategy. In contrast, we preserve the local information progressively by placing a 1×1 convolution every three residual blocks. From Table 4, we find that our network achieves better performance with significantly lower computational cost and number of parameters. We attribute this considerable improvement to the effectiveness of the proposed connectivity pattern, where the features in each residual block can be better utilized by the network. **Attention path**. Previous SISR approaches perform channel attention in-place within the residual blocks, whereas this work takes the attention out of the main computational path, and computes it in parallel. To verify the effectiveness of this approach, in this experiment, we use a baseline which is identical to the proposed method except for the absence of the attention path (*Baseline*). We then place the DiVA attention mechanism both inside (*Baseline_in*) and outside (*Baseline_out*) of the residual blocks, comparing their performance in Table 5. As it can be observed, *Baseline_out* leads to performance improvement, having just a few more parameters due to the aggregation operation inside the attention feature bank. These results prove that moving attention operations outside of the residual blocks is beneficial to prevent the loss of information caused by commonly used in-place attention. This justify our choice for keeping a separate computational path to aggregate computations coming from attention operations.

Table 5: Average PSNR obtained on the ResNet baseline network, when placing the DiVA attention mechanism within (*Baseline_in*) or outside (*Baseline_out*) the residual blocks. Results are shown on five benchmark datasets and with a $\times 4$ scale factor.

Methods	Baseline	Baseline_in	Baseline_out
Params	815K	890K	939K
Set5	32.18	32.32(+0.14dB)	**32.41**(**+0.23dB**)
Set14	28.59	28.64(+0.05dB)	**28.70**(**+0.11dB**)
B100	27.55	27.58(+0.03dB)	**27.65**(**+0.10dB**)
Urban100	26.09	26.31(+0.02dB)	**26.41**(**+0.12dB**)
Manga109	30.49	30.65(+0.16dB)	**30.73**(**+0.24dB**)

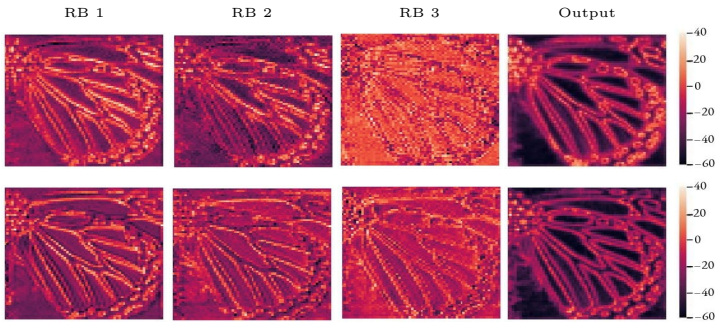


Figure 4: Average feature maps of residual blocks (RBs). **Top**: Attention is applied within the residual (classic approach). **Bottom**: Attention is applied outside the residual (our approach).

Figure 4 additionally shows average feature maps in residual blocks, when attention mechanisms are applied inside (in-place, top row) or outside (as in our RAFG, bottom row). This visualization shows how RAFGs are able to learn sharper representations than those obtained with in-place attention. In essence, each RAFG directs computations towards edges and details, thus obtaining a more defined representation at the output. In contrast, when using in-place attention, feature maps vary significantly from the first residual block to the last. As a result, edges and contours are outlined at the first layer, and smooth areas within the original image become suppressed at subsequent blocks.

4.3. Comparison with State-of-the-art Lightweight Methods

In this section, DiVANet and DiVANet-S are compared to other lightweight state-of-the-art SR methods. A self-ensemble method [42] is also used to further improve the performance of the DiVANet (denoted as DiVANet+).

*4.3.1. Results with **BI** Degradation Model*

Simulating LR images with the **BI** degradation model is widely used in the context of image SR. For the **BI** degradation model, we compare our proposed DiVANet-S, DiVANet and DiVANet+ with state-of-the-art SR frameworks, including VDSR [10], DRCN [11], SRDenseNet [7], CARN [14], SRFBN-S [13], CBPN [24], FALSR-A[18], SRMDNF [43], LAPAR-A [23], MAFFSRN [28], LatticeNet [20], MPRNet [29], RFDN-L [19], MADNet [33], HDRN [17], DPN [26], and A2F-L [22].

Table 6 shows quantitative results when evaluating PSNR and SSIM on five benchmark datasets with different algorithms. For a more informative comparison, the number of parameters and the number of multiplications and additions (Multi-Adds) are also given. It can be observed that the proposed DiVANet-S has only less than 500K parameters, but its performance is superior to many state-of-the-art methods. For example, in comparison with CARN and CBPN, DiVANet-S attains significantly better performance while only needing 30% and 40% of their parameters, respectively. Furthermore, DiVANet is the best performing one, at all scales and in all datasets. Especially on the challenging dataset Urban100, which contains rich structural contents, the proposed DiVANet advances the state-of-the-art with improvement margins of 0.14dB, 0.18dB and 0.10dB for scale factors $\times 2$, $\times 3$ and $\times 4$, respectively. In addition, more significant improvements are shown in the Manga109 dataset, where the proposed DiVANet model outperforms A^2F-L (with the highest performance amongst the aforementioned methods), by PSNR gains of 0.13dB and 0.11dB for $\times 2$ and $\times 3$ enlargement. The advantage of our method can be also verified via SSIM scores. The SSIM score focuses on the visible structures in the image. The proposed DiVANet also achieves the best SSIM score, which indicates that DiVANet can better recover visible structures. These results validate the superiority of the proposed method, particularly on super-resolving the images with fine structures such as those in Urban100 and Manga109. Furthermore, it can be seen that DiVANet+ achieves further improvements through the use of

Table 6: Average PSNR/SSIM values for models with the same order of magnitude of parameters. Performance is shown for scale factors ×2, ×3 and ×4 with **BI** degradation model. The Multi-Adds (MAC) is calculated corresponding to a 1280 × 720 HR image. The best and second best results are highlighted in red and blue respectively.

Scale	Method	Params	MAC	Set5		Set14		B100		Urban100		Manga109	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
×2	VDSR [10]	665K	613G	37.53	0.9587	33.03	0.9124	31.90	0.8960	30.76	0.9140	37.22	0.9750
	DRCN [11]	1,774K	17,974G	37.53	0.9587	33.03	0.9124	31.90	0.8960	30.76	0.9140	37.22	0.9750
	CARN [14]	1,592K	223G	37.76	0.9590	33.52	0.9166	32.09	0.8978	31.92	0.9256	38.36	0.9765
	SRFBN-S [13]	282K	680G	37.78	0.9597	33.35	0.9156	32.00	0.8970	31.41	0.9207	38.06	0.9757
	CBPN [24]	1,036K	240.7G	37.90	0.9590	33.60	0.9171	32.17	0.8989	32.14	0.9279	–	–
	FALSR-A[18]	1,021K	234.7G	37.82	0.9595	33.55	0.9168	32.12	0.8987	31.93	0.9256	–	–
	SRMDNF [43]	1,513K	348G	37.79	0.9600	33.32	0.9150	32.05	0.8980	31.33	0.9200	–	–
	LAPAR-A [23]	548K	171G	38.01	0.9605	33.62	0.9183	32.19	0.8999	32.10	0.9283	38.67	0.9772
	MAFFSRN [28]	790K	154.4G	38.07	0.9607	33.59	0.9177	32.23	0.9005	32.38	0.9308	–	–
	LatticeNet [20]	756K	169.5G	38.15	0.9610	33.78	0.9193	32.25	0.9005	32.24	0.9302	–	–
	MPRNet [29]	538K	163.3G	38.08	0.9608	33.79	0.9196	32.25	0.9004	32.25	0.9317	–	–
	RFDN-L [19]	626K	38G	38.08	0.9606	33.67	0.9190	32.18	0.8996	32.24	0.9290	38.95	0.9773
	MADNet [33]	878K	187.1G	37.94	0.9604	33.46	0.9167	32.10	0.8988	31.74	0.9246	–	–
	HDRN [17]	878K	316.2G	37.75	0.9590	33.49	0.9150	32.03	0.8980	31.87	0.9250	38.07	0.9770
	DPN [26]	832K	140G	37.52	0.9586	33.08	0.9129	31.89	0.8958	30.82	0.9144	–	–
	A^2F-L [22]	1,363K	306.1G	38.09	0.9607	33.78	0.9192	32.23	0.9002	32.46	0.9313	38.95	0.9772
	DRSAN [30]	1,190K	274.6G	38.14	0.9611	33.75	0.9188	32.25	0.9010	32.46	0.9317	–	–
	DiVANet-S	405K	75G	38.10	0.9605	33.76	0.9189	32.22	0.8999	32.40	0.9305	38.88	0.9771
	DiVANet	902K	189G	38.16	0.9612	33.80	0.9195	32.29	0.9012	32.60	0.9325	39.08	0.9775
	DiVANet+	902K	189G	38.23	0.9618	33.88	0.9201	32.36	0.9018	32.67	0.9330	39.15	0.9780
×3	VDSR [10]	665K	613G	33.66	0.9213	29.77	0.8314	28.82	0.7976	27.14	0.8279	37.22	0.9750
	DRCN [11]	1,774K	17,974G	33.82	0.9226	29.76	0.8311	28.80	0.7963	27.15	0.8276	32.24	0.9343
	CARN [14]	1,592K	119G	34.29	0.9255	30.29	0.8407	29.06	0.8034	28.06	0.8493	33.50	0.9440
	SRFBN-S [13]	376K	832G	34.20	0.9255	30.10	0.8372	28.96	0.8010	27.66	0.8415	33.02	0.9404
	SRMDNF [43]	1,530K	156G	34.12	0.9250	30.04	0.8370	28.97	0.8030	27.57	0.8400	–	–
	LAPAR-A [23]	544K	114G	34.36	0.9267	30.34	0.8421	29.11	0.8054	28.15	0.8523	33.51	0.9441
	MAFFSRN [28]	807K	68.5G	34.45	0.9277	30.40	0.8432	29.13	0.8061	28.26	0.8552	–	–
	LatticeNet [20]	765K	76.3G	34.53	0.9281	30.39	0.8424	29.15	0.8059	28.33	0.8538	–	–
	MPRNet [29]	538K	63.1G	34.57	0.9285	30.42	0.8441	29.17	0.8073	28.42	0.8578	–	–
	RFDN-L [19]	633K	38G	34.47	0.9280	30.35	0.8421	29.11	0.8053	28.32	0.8547	33.78	0.9458
	MADNet [33]	930K	88.4G	34.26	0.9262	30.29	0.8410	29.04	0.8033	27.91	0.8464	–	–
	HDRN [17]	878K	187.1G	34.24	0.9240	30.23	0.8400	28.96	0.8040	27.93	0.8490	33.17	0.9420
	DPN [26]	832K	114.2G	33.71	0.9222	29.80	0.8320	28.84	0.7981	27.17	0.8282	–	–
	A^2F-L [22]	1,367K	136.1G	34.54	0.9283	30.41	0.8436	29.14	0.8062	28.40	0.8574	33.83	0.9463
	DRSAN [30]	1,290K	133.4G	34.59	0.9282	30.42	0.8443	29.18	0.8079	28.52	0.8593	–	–
	DiVANet-S	451K	38G	34.48	0.9275	30.43	0.8431	29.13	0.8055	28.42	0.8568	33.80	0.9455
	DiVANet	949K	89G	34.60	0.9285	30.47	0.8447	29.19	0.8073	28.58	0.8603	33.94	0.9468
	DiVANet+	949K	89G	34.66	0.9289	30.53	0.8452	29.26	0.8077	28.66	0.8610	34.02	0.9473
×4	VDSR [10]	665K	613G	31.35	0.8838	28.01	0.7674	27.29	0.7251	25.18	0.7524	28.83	0.8809
	DRCN [11]	1,774K	17,974G	31.54	0.8850	29.19	0.7720	27.32	0.7280	25.12	0.7560	29.09	0.8845
	SRDenseNet [7]	2,015K	390G	32.00	0.8931	28.50	0.7782	27.53	0.7337	26.05	0.7819	30.41	0.9071
	CARN [14]	1,592K	91G	32.13	0.8937	28.60	0.7806	27.58	0.7349	26.07	0.7837	30.47	0.9084
	SRFBN-S [13]	483K	1,037G	31.98	0.8923	28.45	0.7779	27.44	0.7313	25.71	0.7719	29.91	0.9008
	CBPN [24]	1,197K	97.9G	32.21	0.8944	28.63	0.7813	27.58	0.7356	26.14	0.7869	–	–
	SRMDNF [43]	1,555K	89G	31.96	0.8930	28.35	0.7770	27.49	0.7340	25.68	0.7730	–	–
	LAPAR-A [23]	659K	94G	32.15	0.8944	28.61	0.7818	27.61	0.7366	26.14	0.7871	30.42	0.9074
	MAFFSRN [28]	830K	38.6G	32.20	0.8953	28.62	0.7822	27.59	0.7370	26.16	0.7887	–	–
	LatticeNet [20]	777K	43.6G	32.30	0.8962	28.68	0.7830	27.62	0.7367	26.25	0.7873	–	–
	MPRNet [29]	538K	31.3G	32.38	0.8969	28.69	0.7841	27.63	0.7385	26.31	0.7921	–	–
	RFDN-L [19]	643K	38G	32.28	0.8957	28.61	0.7818	27.58	0.7363	26.20	0.7883	30.61	0.9096
	MADNet [33]	1,002K	54.1G	32.11	0.8939	28.52	0.7799	27.52	0.7340	25.89	0.7782	–	–
	HDRN [17]	867K	316.2G	32.23	0.8960	28.58	0.7810	27.53	0.7370	26.09	0.7870	30.43	0.9080
	DPN [26]	832K	140G	31.42	0.8849	28.07	0.7688	27.30	0.7256	25.25	0.7546	–	–
	A^2F-L [22]	1,374K	77.2G	32.32	0.8964	28.67	0.7839	27.62	0.7379	26.32	0.7931	30.72	0.9115
	DRSAN [30]	1,270K	88.7G	32.34	0.8960	28.65	0.7841	27.63	0.7390	26.33	0.7936	30.72	0.9115
	DiVANet-S	442K	28G	32.32	0.8958	28.63	0.7827	27.61	0.7377	26.35	0.7926	30.68	0.9105
	DiVANet	939K	57G	32.41	0.8973	28.70	0.7844	27.65	0.7391	26.42	0.7958	30.73	0.9119
	DiVANet+	939K	57G	32.48	0.8978	28.78	0.7848	27.73	0.7395	26.49	0.7963	30.78	0.9124

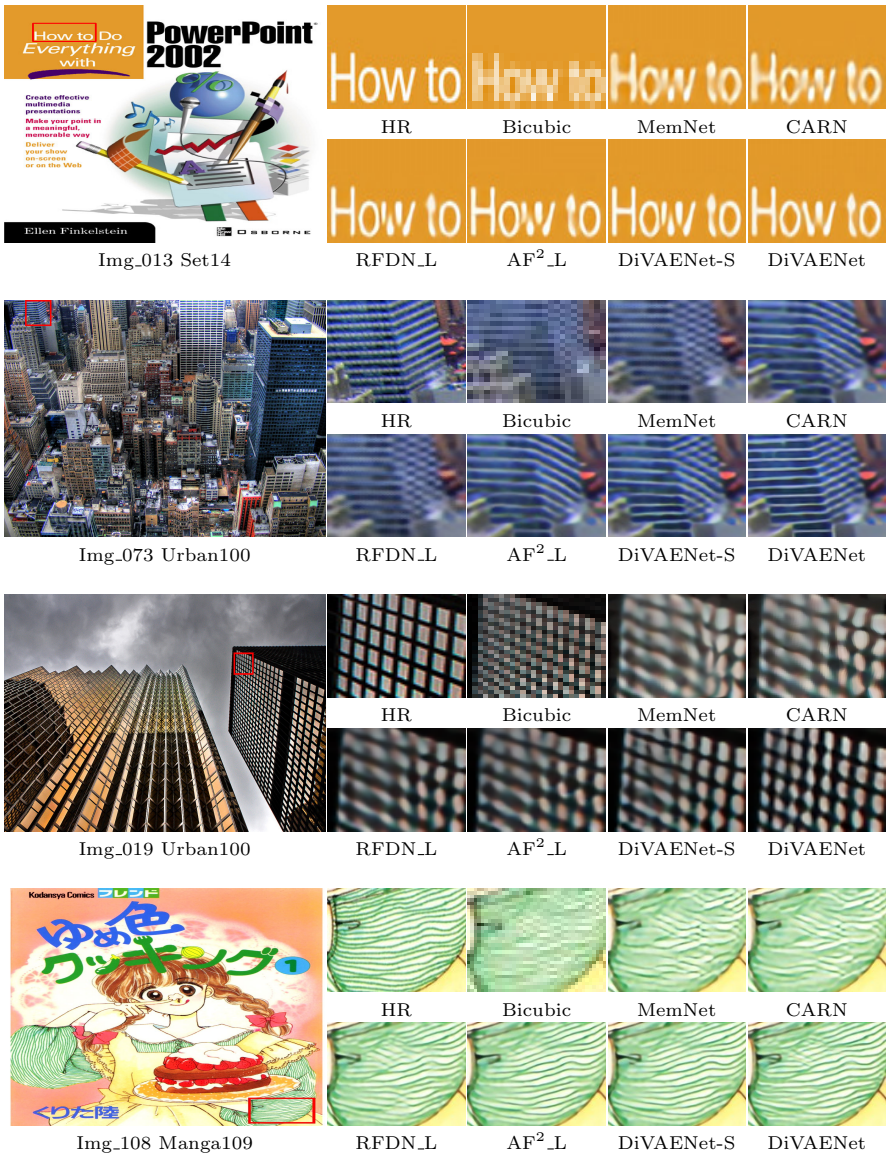


Figure 5: Visual results of **BI** degradation model for $\times 4$ scale factor.

self-ensembles [42].

In Figure 5, we present some qualitative visual comparisons for the $\times 4$ scale factor. It can be observed that DiVANet-S and DiVANet prevent distortions, suppress artifacts and generate more faithful results. This visual comparisons

Table 7: Quantitative results with **BD** and **DN** degradation models. Performance is shown for scale factor ×3. The best and second best results are highlighted in red and blue respectively.

Methods	Degrad.	Set5		Set14		B100		Urban100		Manga109	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SRCNN [4]	BD	32.05	0.8944	28.80	0.8074	28.13	0.7736	25.70	0.7770	29.47	0.8924
	DN	25.01	0.6950	23.78	0.5898	23.76	0.5538	21.19	0.5737	23.75	0.7148
VDSR [10]	BD	33.25	0.9150	29.46	0.8244	28.57	0.7893	26.61	0.8136	31.06	0.9234
	DN	25.20	0.7183	24.00	0.6112	24.00	0.5749	22.22	0.6096	24.20	0.7525
IRCNN_G [44]	BD	33.38	0.9182	29.63	0.8281	28.65	0.7922	26.77	0.8154	31.15	0.9245
	DN	25.70	0.7379	24.45	0.6305	24.28	0.5900	22.90	0.6429	24.88	0.7765
IRCNN_C [44]	BD	29.55	0.8246	27.33	0.7135	26.46	0.6572	24.89	0.7172	28.68	0.7701
	DN	26.18	0.7430	24.68	0.6300	24.52	0.5850	22.63	0.6205	24.74	0.7701
SRMDNF [43]	BD	34.09	0.9242	30.11	0.8364	28.98	0.8009	27.50	0.8370	32.97	0.9391
	DN	27.74	0.8026	26.13	0.6924	25.64	0.6495	24.28	0.7092	26.72	0.8590
RDN [8]	BD	34.57	0.9280	30.53	0.8447	29.23	0.8079	28.46	0.8581	33.97	0.9465
	DN	28.46	0.8151	26.60	0.7101	25.96	0.6573	24.92	0.7362	28.00	0.8590
CASGCN [2]	BD	34.62	0.9283	30.60	0.8458	29.30	0.8196	28.68	0.8611	34.27	0.9476
	DN	–	–	–	–	–	–	–	–	–	–
DiVANet-S (Ours)	BD	34.45	0.9263	30.40	0.8420	29.11	0.8048	28.26	0.8529	33.90	0.9448
	DN	28.41	0.8154	26.16	0.6933	25.87	0.6599	24.88	0.7356	28.13	0.8600
DiVANet (Ours)	BD	34.64	0.9286	30.63	0.8460	29.31	0.8198	28.70	0.8613	34.30	0.9479
	DN	28.49	0.8159	26.22	0.6939	25.93	0.6605	24.94	0.7361	28.18	0.8605
DiVANet+ (Ours)	BD	34.70	0.9291	30.69	0.8469	29.39	0.82.06	28.78	0.8621	34.38	0.9486
	DN	28.57	0.8164	26.29	0.6945	26.01	0.6611	24.99	0.7369	28.26	0.8611

also demonstrate the powerful representational ability of our methods. Due to page limits, more qualitative results are provided as supplementary material.

*4.3.2. Results with **BD** and **DN** Degradation Models*

Following [8, 13], we also provide the results after applying **BD** and **DN** degradation models. The proposed DiVANet-S, DiVANet, and DiVANet+ are compared with state-of-the-art methods including SRCNN [4], VDSR [10], IRCNN_G [44], IRCNN_C [44], SRMDNF [43], RDN [8], and CASGCN [2]. As shown in Table 7, our methods achieve better PSNR and SSIM scores compared to other SR methods, in all datasets. The consistently better results of our methods indicate that they adapt well to scenarios with multiple degradation models.

In Figures 6 and 7 we provide some visual results for the **BD** and **DN** degradation models for ×4 scale factor from the standard benchmark datasets. For **BD** degradation, other methods were unable to remove blurring artifacts. In

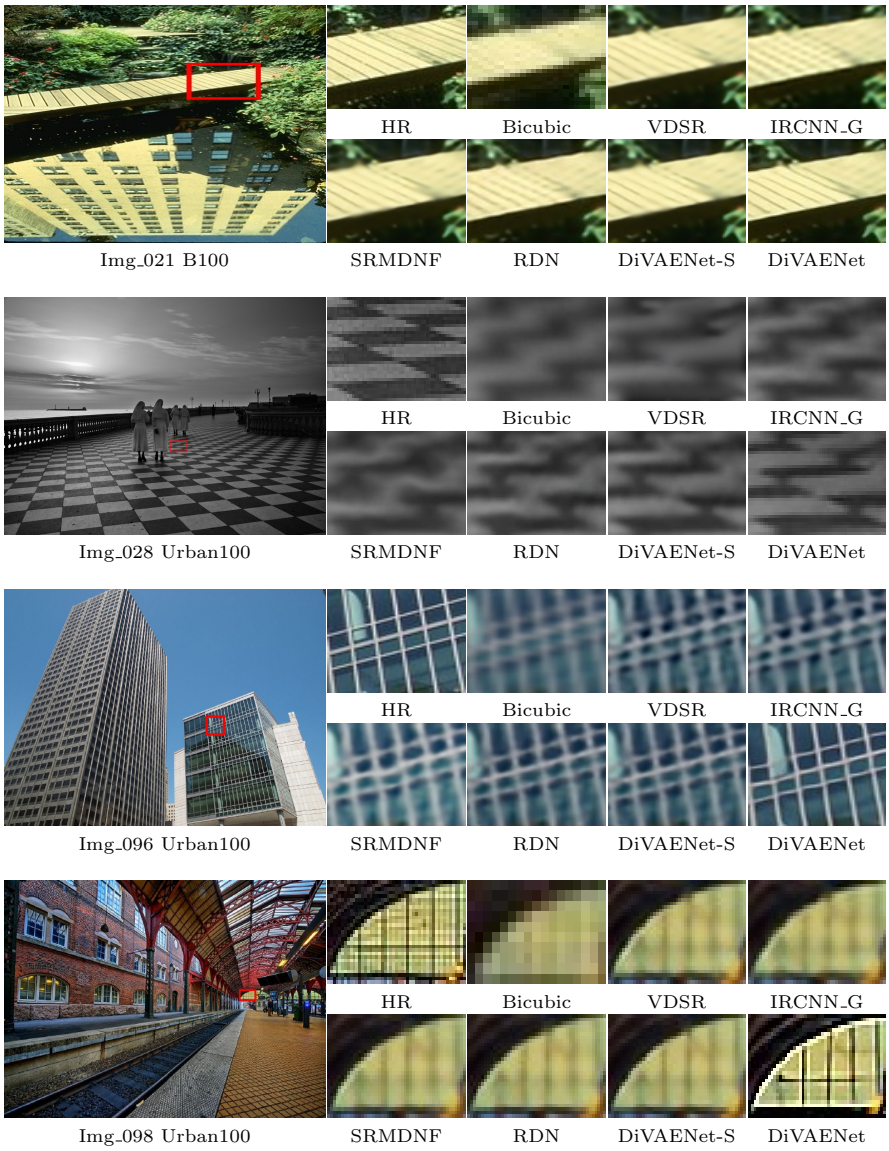


Figure 6: Visual results of **BD** degradation model for $\times 3$ scale factor.

contrast, DiVANet-S and DiVANet are able to recover structured details that were missing in the LR image, by efficiently exploiting the feature hierarchy. Regarding the **DN** degradation, we observe that recovering details becomes difficult with other methods. However, ours deliver good results by removing

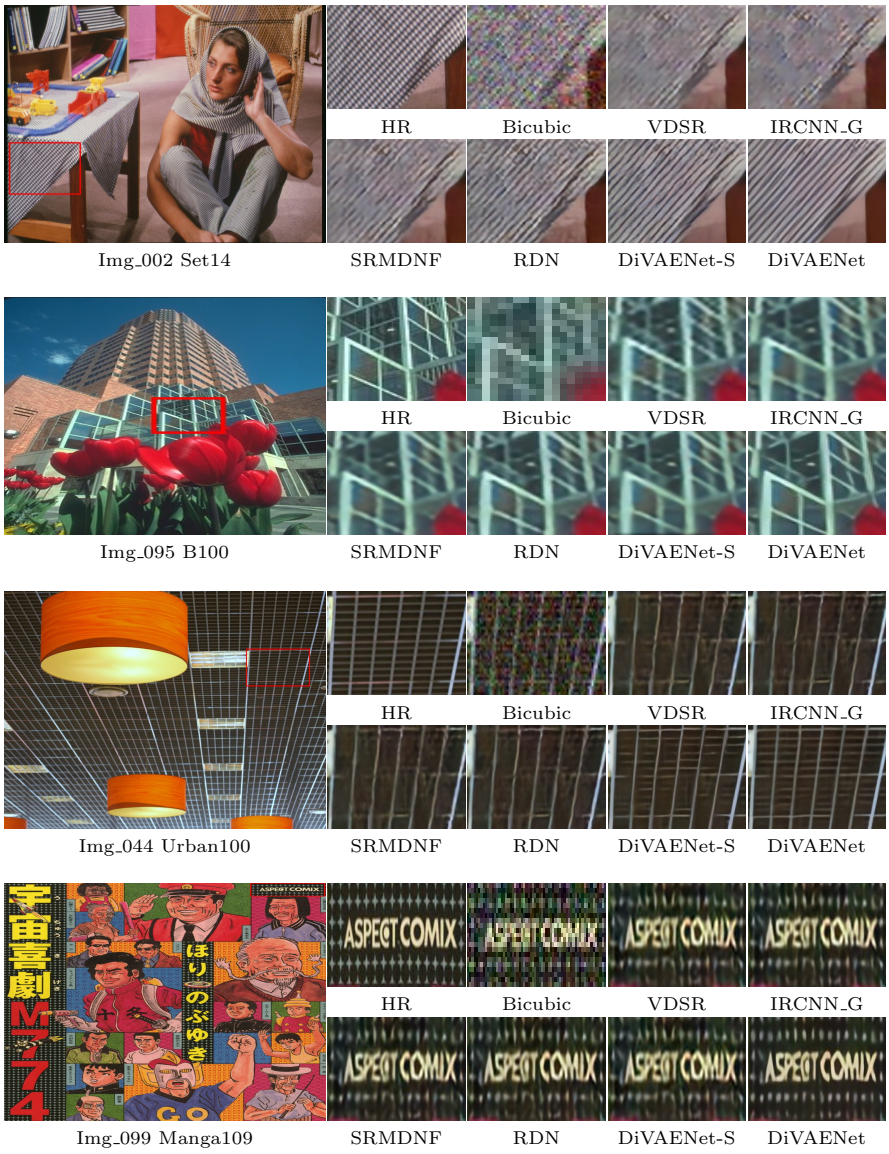


Figure 7: Visual results of **DN** degradation model for $\times 3$ scale factor.

additional noise and enhancing the details. From these comparisons, we further indicate the robustness and effectiveness of our methods in handling **BD** and **DN** degradation models.

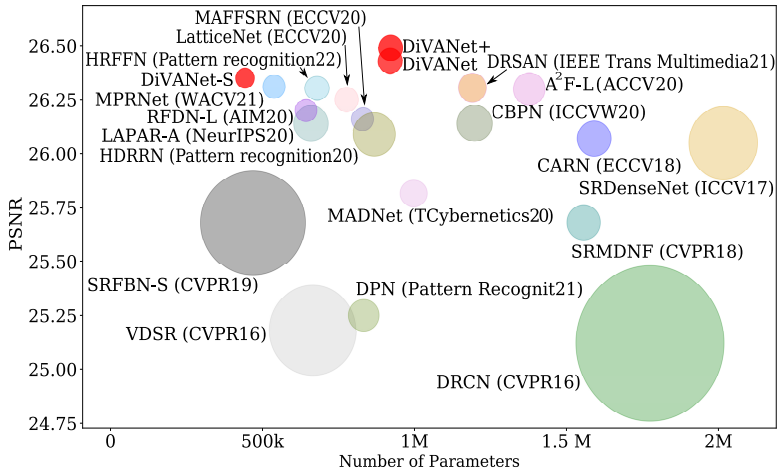


Figure 8: Comparing capacity vs performance for lightweight state-of-the-art SISR models on Urban100 (×4). Circle sizes are set proportional to the number of multiplications and additions (Multi-Adds).

4.3.3. Model complexity analysis

In this section, we compare the trade-off between performance and number of parameters for our methods (DiVANet-S, DiVANet and DiVANet+) and existing lightweight networks. Figure 8 shows the PSNR performances of several lightweight models, namely VDSR [10], DRCN [11], SRDenseNet [7], CARN [14], SRFBN-S [13], CBPN [24], SRMDNF [43], LAPAR-A [23], MAFFSRN [28], LatticeNet [20], MPRNet [29], RFDN-L [19], MADNet [33], HDRN [17], DPN [26], and A^{2}F-L [22]. versus their number of parameters, with results evaluated on Urban100 for ×4. As shown in Figure 8, our models achieve state-of-the-art results with less parameters and Multi-Add operations. This demonstrates that our proposals achieve a better trade-off between model size and reconstruction performance.

In addition, we compare our models with large networks such as EDSR [5], MDSR [5], MSRN [12], RDN [8], RCAN [27], SRFBN [13], CSFM [6], RFANet [19], S^{2}TSR-NAAN [1], S^{2}TSR-RRDB [1], and CASGCN [2]. The results are given in Figure 9 in terms of network parameters and reconstruction effects (PSNR). For example, S^{2}TSR-RRDB, CASGCN, and RFANet respectively have pa-

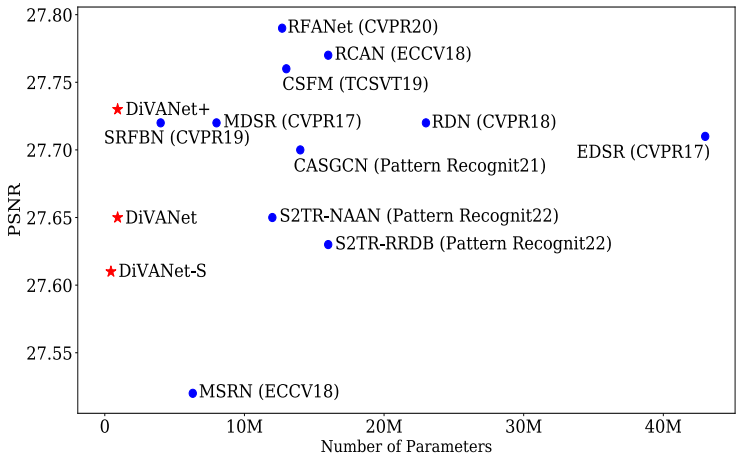


Figure 9: Comparing capacity vs performance for non-lightweight state-of-the-art SISR models in the B100 dataset (×4). The red stars represent our proposed methods.

rameters/PSNR ratios of 16M/27.63dB, 14M/27.70dB, and 12M/27.76, under ×4 setting on the B100 dataset. On the other hand, the proposed DiVANet (0.9M/27.65) and DiVANet+ (0.9M/27.73) achieve competitive or better results, while only needing the 5%, 6% and 7% parameters of S^{2}TSR-NAAN, S^{2}TSR-RRDB and CASGCN, respectively. The DiVANet-S model also shows comparable results to the heavy models. In particular, the DiVANet-S model outperforms MSRN by a large margin of 0.10dB. It is worth noting that while MSRN has 8M parameters, DiVANet-S only has 0.4M parameters. Thus, the proposed networks are lightweight and more efficient than other state-of-the-art methods.

4.3.4. Memory Complexity and Running Time Analysis

Table 8 illustrates the superiority of the proposed DiVANet-S and DiVANet architectures in terms of Inference Time (s) and Memory Consumption (MB), when compared to recent light- and heavy-weight state-of-the-art approaches on Urban100 ×4. For a fair comparison, we use a single NVIDIA RTX 3090 GPU for evaluation, and their official source code implementations. It can be observed that our models have the fastest running time, while also using the least memory per image. Especially, our networks are highly efficient but

Table 8: Average running time (s) and memory consumption (MB) comparison on Urban100 for ×4.

Methods	Params	Memory	Running Time(s)	PSNR
CARN[14]	1.5M	1,116	0.032	26.07
SRFBN-S[13]	0.5M	2,154	0.031	25.71
SRDenseNet[7]	2M	5,531	0.221	26.05
RFDN-L[19]	0.6M	3,015	0.033	26.22
A^2F-L[22]	1.3M	3,015	0.032	26.32
RCAN[27]	16M	1,531	0.297	26.82
EDSR[5]	43M	2,731	0.085	26.64
SAN[13]	16M	3,015	0.224	26.79
RDN[8]	23M	5,015	0.172	26.82
DiVANet-S (Ours)	0.4M	671	0.012	26.35
DiVANet (Ours)	0.9M	875	0.019	26.42

RCAN [27], EDSR[5], RFANet [19], and RDN [8] which are 16×, 5×, 12×, and 10× slower than DiVANet, respectively. These networks mainly leverage much deeper network designs to achieve more accurate SR results. This comparison demonstrates that our methods effectively balance performance and running time.

4.3.5. Perceptual Metrics

Perceptual metrics better reflect the human judgment of image quality. In this paper, Perceptual Index (PI) [41] is chosen as the perceptual metric. Table 9 shows the PI for those works with publicly available source code, and the same order of magnitude in terms of parameters. We observe that our proposed models obtains better results than all the compared baselines. This demonstrates the

Table 9: Perceptual index comparison of the proposed methods with recent lightweight state-of-the-art methods on five datasets for ×4. The lower is better. All of the output SR images are provided officially.

Methods	Params	Set5	Set14	B100	Urban100	Manga109
CARN[14]	1.5M	6.297	5.775	5.700	5.540	5.132
SRFBN-S[10]	0.6M	6.451	5.775	5.702	5.549	5.010
SRDenseNet[7]	2M	6.128	5.615	5.653	5.526	4.762
RFDN_L[19]	0.6M	6.124	5.644	5.659	5.531	4.810
A^2F_L[22]	1.3M	6.084	5.499	5.532	5.179	4.771
DiVANet-S (Ours)	0.4M	5.550	5.490	5.430	5.168	4.676
DiVANet (Ours)	0.9M	**5.511**	**5.361**	**5.163**	**5.149**	**4.480**

ability of the proposed DiVANet and DiVANet-S for generating realistic images.

5. Conclusions and Future Work

In this paper, we have introduced a novel and efficient architecture called directional variance attention network (DiVANet) for modeling the process of single image super-resolution. We propose a directional variance attention mechanism (DiVA), specifically related to SR, which encodes spatial and inter-channel information simultaneously by considering higher-order feature statistics. DiVANet is able to extract and preserve fine details from the whole feature hierarchy in order to reduce the degradation caused by successive residual aggregations. This is achieved through a novel Residual Attention Feature Group (RAFG), which combines an efficient connectivity pattern with a DiVA module that is processed in parallel to the main residual computational path. Through a series of ablation experiments, we have demonstrated the effectiveness of the proposed DiVA and RAFG schemes. We have empirically shown that our proposal attains better PSNR, SSIM, and perceptual scores than previous lightweight state-of-the-art models on all benchmarks while having a similar or fewer amount of parameters.

Although the proposed methods achieve better performance compared to other SR methods, it can be observed that the reconstructed results of realistic fine textures and details are obviously worse than the quality of the ground-truth images. This is mainly because the models trained on datasets conducted by manual degradation can only simulate limited patterns, thus performing poorly in real-world scenes.

In our future work, we will attempt to improve the SR quality of natural images by using the new datasets obtained by different resolution cameras with real-world scenarios. Moreover, we will extend our proposals to other low-level restoration tasks e.g., denoising, dehazing, and JPEG deblocking. Lastly, we wish to further develop this work by applying our technique to video data. Many streaming services require a large storage to provide high-quality videos. In

conjunction with our approach, one may devise a service that stores low-quality videos that go through our SR system to produce high-quality videos on the fly.

Acknowledgements

This work was supported by the Spanish Ministry of Economy and Competitiveness (MINECO) and the European Regional Development Fund (ERDF) under Grants PID2020-120311RB-I00 funded by MCIN/AEI/10.13039/501100011033, and TIN2015-65464-R. Isabelle Hupont's work is supported by the HUMAINT project of the European Commission's Joint Research Centre.

References

[1] L. Wang, K.-J. Yoon, Semi-supervised student-teacher learning for single image super-resolution, Pattern Recognition 121 (2022) 108206.

[2] Y. Yang, Y. Qi, Image super-resolution via channel attention and spatial graph convolutional network, Pattern Recognition 112 (2021) 107798.

[3] R. Chen, H. Zhang, J. Liu, Multi-attention augmented network for single image super-resolution, Pattern Recognition 122 (2022) 108349.

[4] C. Dong, C. C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE transactions on pattern analysis and machine intelligence 38 (2015) 295–307.

[5] B. Lim, S. Son, H. Kim, S. Nah, K. Mu Lee, Enhanced deep residual networks for single image super-resolution, in: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2017, pp. 136–144.

[6] Y. Hu, J. Li, Y. Huang, X. Gao, Channel-wise and spatial feature modulation network for single image super-resolution, IEEE Transactions on Circuits and Systems for Video Technology 30 (2019) 3911–3927.

[7] T. Tong, G. Li, X. Liu, Q. Gao, Image super-resolution using dense skip connections, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4799–4807.

[8] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, Y. Fu, Residual dense network for image super-resolution, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 2472–2481.

[9] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.

[10] J. Kim, J. K. Lee, K. M. Lee, Accurate image super-resolution using very deep convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1646–1654.

[11] J. Kim, J. K. Lee, K. M. Lee, Deeply-recursive convolutional network for image super-resolution, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1637–1645.

[12] J. Li, F. Fang, K. Mei, G. Zhang, Multi-scale residual network for image super-resolution, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 517–532.

[13] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, W. Wu, Feedback network for image super-resolution, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3867–3876.

[14] N. Ahn, B. Kang, K.-A. Sohn, Fast, accurate, and lightweight super-resolution with cascading residual network, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 252–268.

[15] J. Qin, F. Liu, K. Liu, G. Jeon, X. Yang, Lightweight hierarchical residual feature fusion network for single-image super-resolution, Neurocomputing (2022).

[16] P. Behjati, P. Rodriguez, A. Mehri, I. Hupont, C. F. Tena, J. Gonzalez, Overnet: Lightweight multi-scale super-resolution with overscaling network, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 2694–2703.

[17] K. Jiang, Z. Wang, P. Yi, J. Jiang, Hierarchical dense recursive network for image super-resolution, Pattern Recognition 107 (2020) 107475.

[18] X. Chu, B. Zhang, H. Ma, R. Xu, Q. Li, Fast, accurate and lightweight super-resolution with neural architecture search, in: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021, pp. 59–64.

[19] J. Liu, J. Tang, G. Wu, Residual feature distillation network for lightweight image super-resolution, in: European Conference on Computer Vision, Springer, 2020, pp. 41–55.

[20] X. Luo, Y. Xie, Y. Zhang, Y. Qu, C. Li, Y. Fu, Latticenet: Towards lightweight image super-resolution with lattice block, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16, Springer, 2020, pp. 272–289.

[21] Z. Lu, H. Liu, J. Li, L. Zhang, Efficient transformer for single image super-resolution, arXiv preprint arXiv:2108.11084 (2021).

[22] X. Wang, Q. Wang, Y. Zhao, J. Yan, L. Fan, L. Chen, Lightweight single-image super-resolution network with attentive auxiliary feature learning, in: Proceedings of the Asian Conference on Computer Vision, 2020.

[23] W. Li, K. Zhou, L. Qi, N. Jiang, J. Lu, J. Jia, Lapar: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond, Advances in Neural Information Processing Systems 33 (2020).

[24] F. Zhu, Q. Zhao, Efficient single image super-resolution via hybrid residual feature learning with compact back-projection network, in: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019, pp. 0–0.

[25] X. Zhu, K. Guo, S. Ren, B. Hu, M. Hu, H. Fang, Lightweight image super-resolution with expectation-maximization attention mechanism, IEEE Transactions on Circuits and Systems for Video Technology (2021).

[26] Y. Liang, R. Timofte, J. Wang, S. Zhou, Y. Gong, N. Zheng, Single-image super-resolution-when model adaptation matters, Pattern Recognition 116 (2021) 107931.

[27] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, Y. Fu, Image super-resolution using very deep residual channel attention networks, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 286–301.

[28] A. Muqeet, J. Hwang, S. Yang, J. Kang, Y. Kim, S.-H. Bae, Multi-attention based ultra lightweight image super-resolution, in: European Conference on Computer Vision, Springer, 2020, pp. 103–118.

[29] A. Mehri, P. B. Ardakani, A. D. Sappa, Mprnet: Multi-path residual network for lightweight image super resolution, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 2704–2713.

[30] K. Park, J. W. Soh, N. I. Cho, Dynamic residual self-attention network for lightweight single image super-resolution, IEEE Transactions on Multimedia (2021).

[31] Y. Mei, Y. Fan, Y. Zhou, L. Huang, T. S. Huang, H. Shi, Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5690–5699.

[32] Y. Mei, Y. Fan, Y. Zhou, Image super-resolution with non-local sparse attention, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3517–3526.

[33] R. Lan, L. Sun, Z. Liu, H. Lu, C. Pang, X. Luo, Madnet: A fast and lightweight network for single-image super resolution, IEEE transactions on cybernetics 51 (2020) 1443–1453.

[34] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, L. Zhang, Second-order attention network for single image super-resolution, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 11065–11074.

[35] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, L. Zhang, Ntire 2017 challenge on single image super-resolution: Methods and results, in: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2017, pp. 114–125.

[36] M. Bevilacqua, A. Roumy, C. Guillemot, M. L. Alberi-Morel, Low-complexity single-image super-resolution based on nonnegative neighbor embedding (2012).

[37] R. Zeyde, M. Elad, M. Protter, On single image scale-up using sparse-representations, in: International conference on curves and surfaces, Springer, 2010, pp. 711–730.

[38] P. Arbelaez, M. Maire, C. Fowlkes, J. Malik, Contour detection and hierarchical image segmentation, IEEE transactions on pattern analysis and machine intelligence 33 (2010) 898–916.

[39] J.-B. Huang, A. Singh, N. Ahuja, Single image super-resolution from transformed self-exemplars, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 5197–5206.

[40] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, K. Aizawa, Sketch-based manga retrieval using manga109 dataset, Multimedia Tools and Applications 76 (2017) 21811–21838.

[41] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, L. Zelnik-Manor, The 2018 pirm challenge on perceptual image super-resolution, in: Proceedings of the European Conference on Computer Vision (ECCV) Workshops, 2018, pp. 0–0.

[42] R. Timofte, R. Rothe, L. Van Gool, Seven ways to improve example-based single image super resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1865–1873.

[43] K. Zhang, W. Zuo, L. Zhang, Learning a single convolutional super-resolution network for multiple degradations, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3262–3271.

[44] K. Zhang, W. Zuo, S. Gu, L. Zhang, Learning deep cnn denoiser prior for image restoration, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 3929–3938.