




Upsampling Attention Network for Single Image Super-resolution

Zhijie Zheng^{1,2,3}^a, Yuhang Jiao⁴^b and Guangyou Fang^{1,2,3}^c

¹*Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China*

²*Key Laboratory of Electromagnetic Radiation and Sensing Technology, Chinese Academy of Sciences, Beijing, China*

³*University of Chinese Academy of Sciences, Beijing, China*

⁴*The University of Tokyo, Tokyo, Japan*

Keywords: Single Image Super-resolution, Convolutional Neural Network, Attention Mechanism.

Abstract: Recently, convolutional neural network (CNN) has been widely used in single image super-resolution (SISR) and made significant advances. However, most of the existing CNN-based SISR models ignore fully utilization of the extracted features during upsampling, causing information bottlenecks, hence hindering the expressive ability of networks. To resolve these problems, we propose an upsampling attention network (UAN) for richer feature extraction and reconstruction. Specifically, we present a residual attention groups (RAGs) based structure to extract structural and frequency information, which is composed of several residual feature attention blocks (RFABs) with a non-local skip connection. Each RFAB adaptively rescales spatial- and channel-wise features by paying attention to correlations among them. Furthermore, we propose an upsampling attention block (UAB), which not only applies parallel upsampling processes to obtain richer feature representations, but also combines them to obtain better reconstruction results. Experiments on standard benchmarks show the advantage of our UAN over state-of-the-art methods both in objective metrics and visual qualities.

1 INTRODUCTION


Single image super-resolution (SISR) (Freeman et al., 2000) is a classical task in computer vision that aims at reconstructing an accurate high-resolution (HR) image from its low-resolution (LR) image. However, SISR is an ill-posed problem since there are many HR solutions exist for each LR input. Therefore, numerous methods have been proposed to deal with this inverse problem.


Deep convolutional neural network (CNN) based methods have obtained significant improvements over conventional SISR methods recently. Due to the powerful feature representation ability, CNN-based SISR methods can learn the mapping function from LR image to its corresponding HR image, and achieved state-of-the-art performances. SRCNN (Dong et al., 2014) was first proposed as a three layers CNN for SISR. VDSR (Kim et al., 2016) was then proposed to learn the residual of the interpolation image and HR image. LapSRN(Lai et al., 2017) was further proposed as a multi-phase method to learn intermediate


residual features. FSRCNN (Dong et al., 2016) replaced the predefined interpolation with a learnable upsampling method. Inspired by ResNet(He et al., 2016), EDSR(Lim et al., 2017) introduced a very deep network by using residual connections. DBPN (Muhammad et al., 2018) was designed by using iterative up and downsampling to gain richer features. Channel attention was introduced in RCAN (Zhang et al., 2018) to enhance the extracted features and SAN (Dai et al., 2019) used second-order attention to further improve accuracy.

However, existing CNN-based methods still have several limitations: (1) most of CNN-based SISR methods equally treat extracted features, hindering the representation of models. (2) most existing CNN-based SISR methods use extracted low-resolution information by a single upsampling process, which loses the flexibility to process different information and causes information bottlenecks. Learnable upsampling has proven to be better than interpolation upsampling (Shi et al., 2016), but using single upsampling still limits the possibility of image reconstruction.

To resolve these problems, we propose an upsampling attention network (UAN) to obtain richer feature representation and make better use of features for the

^a  <https://orcid.org/0000-0001-8849-4997>

^b  <https://orcid.org/0000-0003-0629-3345>

^c  <https://orcid.org/0000-0003-2052-4808>

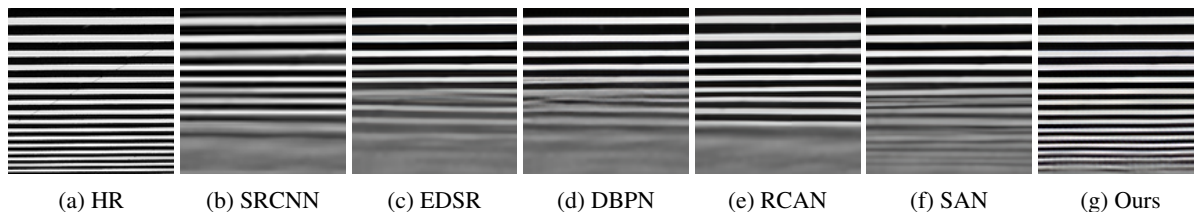


Figure 1: Visual results for $4\times$ SISR on “img_011” from Urban100 benchmark. Our method obtains more details and higher visual quality than other state-of-the-art methods.

upsampling process. Specifically, we pay more attention to useful feature information using a set of residual attention groups (RAGs). RAG preserves abundant information and further enhances information through a non-local skip connection. In each RAG, the residual feature attention block (RFAB) serves as the basic part and a local skip connection passes residual features at a fine level. Moreover, an upsampling attention block (UAB) combines different upsampling features as an ensemble method, and reconstructs the final high-resolution image through convolutional layers. Compared with other state-of-the-art methods as shown in Figure 1, our method recovers more details and obtains higher visual quality.

In summary, our contributions are three-fold:

- We propose an upsampling attention network (UAN) for SISR tasks. We show the superiority of our UAN over state-of-the-art methods both in objective and subjective qualities by experiments on public benchmarks;
- We propose a residual attention group (RAG) structure to extract deep and rich features. We use non-local and local skip connections in RAG to help the main network learn abundant information;
- We propose an upsampling attention block (UAB) to adaptively choose better upsampling features by paying attention to extracted channels. Such mechanism further enhances the representational ability of our network.

We organize our paper as follows. Related work on SISR using convolutional networks and attention mechanism is reviewed in Section 2. UAN architecture is described in Section 3. Experimental settings, ablations and results are discussed in Section 4. We summarize the paper in Section 5.

2 RELATED WORK

A lot of SISR models have been proposed recently in the computer vision field. Attention mechanism is popular in high-level vision tasks. In this section, we

focus on works related to CNN-based SISR models and attention mechanism.

2.1 CNN-based SISR Models

Recently, CNN-based models, which have strong nonlinear representational power, have been widely studied in SISR. These models can be primarily divided into two types based on upsampling methods as follows.

Interpolation-based upsampling uses interpolation (e.g., Bicubic) as the upsampling operator to increase the spatial resolution, then uses convolutional layers to add details. SRCNN (Dong et al., 2014) was first introduced to SISR, which achieved superior performance than previous works. By introducing residual learning to train deeper layers, VDSR (Kim et al., 2016) achieved better performance. LapSRN (Lai et al., 2017) was further proposed as a multi-stage residual learning method to learn intermediate features. These interpolation-based methods can easily keep low-frequency information such as region colors, but lose high-frequency information such as edges and produce new noise.

Learning-based upsampling uses learnable upsampling methods (e.g., deconvolution) to increase the spatial resolution. This approach was firstly proposed by FSRCNN (Dong et al., 2016) to accelerate SRCNN. An enhanced deep network EDSR (Lim et al., 2017) was proposed by introducing residual blocks. Later, DBPN (Muhammad et al., 2018) used iterative up and downsampling to gain richer features. To further improve the performance, RCAN (Zhang et al., 2018) considered feature correlations in channel dimension. These learning-based methods simultaneously reconstruct low- and high-frequency information, usually need bigger model capacity and longer training times.

However, these methods only perform a single upsampling process of LR images. Since upsampling is a key step to reconstruct HR images from LR images, improper upsampling will cause information bottlenecks. We introduce an upsampling attention method that focuses on more important upsampling feature channels to generate integrated HR images.

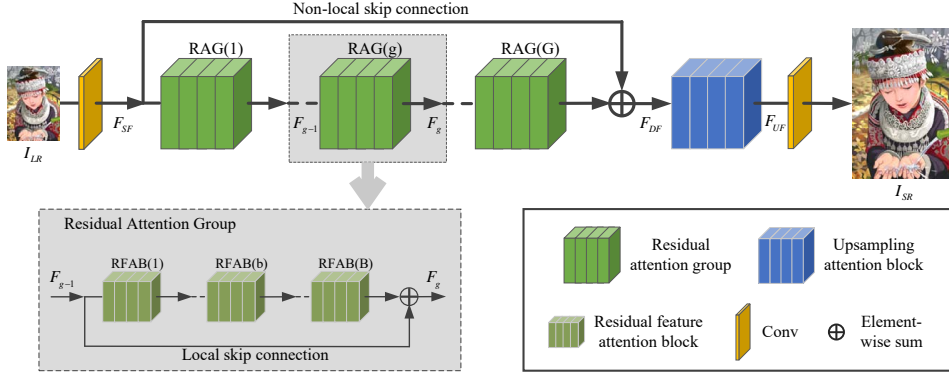


Figure 2: Network architecture of our upsampling attention network (UAN). It consists of four parts: a shallow feature extraction part, a residual attention group (RAG) based deep feature extraction part, an upsampling attention block (UAB), and a reconstruction part.

2.2 Attention Mechanism

Recently, attention mechanism has been proved useful in high-level vision tasks such as image classification. Squeeze-and-excitation network (SENet) (Hu et al., 2018) was proposed to build channel-wise attention to gain significant performance improvement for image classification. Beyond channel, CBAM (Woo et al., 2018) introduced spatial attention in a similar way.

In recent years, attention mechanism was introduced to SISR to improve accuracy (Zhang et al., 2018)(Dai et al., 2019), but they only focused on channel attention. For the extracted feature, different channels carry different frequency information, and different spatial pixels carry different position information. As a low-level vision task, SISR not only requires the recovery of low- and high-frequency information, but also requires accurate position information. If our model pays attention to more informative information, it should be promising to get better performance. To introduce such mechanism in deep CNNs, we propose an upsampling attention network, which will be detailed in the next section.

3 UPSAMPLING ATTENTION NETWORK (UAN)

In this section, we introduce an upsampling attention network (UAN), a new model architecture for SISR tasks, and explain the details of the whole network.

3.1 Model Architecture

As shown in Figure 2, the UAN can be divided into four parts: a shallow feature extraction part, a residual

attention group (RAG) based deep feature extraction part, an upsampling attention block (UAB) module, and a reconstruction part. Let the I_{LR} and I_{SR} as the input and output of UAN. Following (Zhang et al., 2018)(Dai et al., 2019), we use a single convolutional layer to extract the shallow feature F_{SF} from the LR input:

$$F_{SF} = H_{SF}(I_{LR}), \quad (1)$$

where $H_{SF}(\cdot)$ denotes the convolution operation to extract the shallow feature. Then F_{SF} is passed to RAG based feature extraction, which extracts the deep feature as F_{DF} :

$$F_{DF} = H_{RAG}(F_{SF}), \quad (2)$$

where $H_{RAG}(\cdot)$ represents our proposed RAG based deep feature extraction part, which consists of G RAGs to extract the deep feature and a non-local skip connection to enhance the feature. Each RAG consists of B residual feature attention blocks (RFABs). Then the deep feature F_{DF} is upsampled via a UAB module:

$$F_{UF} = H_{UAB}(F_{DF}), \quad (3)$$

where $H_{UAB}(\cdot)$ and F_{UF} represent our UAB part and upsampled feature respectively. There're several methods to be chosen as the upsampling method, such as deconvolution and ESPCN (Shi et al., 2016). However, these methods only use a single upsampling process, which hinders the delivery of rich features. Our UAB module adaptively adjusts concerned features based on attention of upsampling information. Without much computational burden, it achieves higher performance than previous SISR methods. The upsampled feature is then reconstructed to high-resolution image through a single convolutional layer:

$$I_{SR} = H_R(F_{UF}) = H_{UAN}(I_{LR}), \quad (4)$$

where $H_R(\cdot)$ and $H_{UAN}(\cdot)$ respectively denote the reconstruction convolutional layer and our UAN.

Then a certain loss function will be optimized for the model. Several loss functions have been used, such as L_2 loss and L_1 loss. We choose the same loss function as previous works (e.g., L_1 loss). Given a training set contains N LR images and related HR images represents as $\{I_{LR}^i, I_{HR}^i\}_{i=1}^N$. The goal of training UAN is to optimize the L_1 loss function:

$$L(\Theta) = \frac{1}{N} \sum_{i=1}^N \|H_{UAN}(I_{LR}^i) - I_{HR}^i\|_1, \quad (5)$$

where Θ denotes the parameter set of our UAN. The loss function is minimized by gradient descent algorithm. In the following subsections, we pay attention to our proposed RAG, RFAB and UAB modules.

3.2 Residual Attention Group (RAG)

We now show more details of our proposed RAG (see Figure 2), which consists of several RFAB modules and a local skip connection. The RFAB exploits the abundant feature information which will be introduced in Section 3.3. The local skip connection helps to keep more structure cues. Such residual structure allows extracting deep and rich features with more details.

It has been proven that stacked residual blocks are useful to construct deep CNNs in (He et al., 2016). However, building a very deep network in such way for SISR would cause training difficulty and introduce gradient exploding and vanishing to hinder performance improvement. Inspired by works in (Zhang et al., 2018), we proposed RAG as the fundamental block for our network. A RAG in the g -th group is represented as:

$$F_g = H_g(F_{g-1}), \quad (6)$$

where $H_g(\cdot)$ represents the function of the g -th RAG. F_{g-1} and F_g denote the input and output of the g -th RAG. It is known that naively stacking repeated blocks would fail to gain better performance. To this end, we use a non-local skip connection to keep abundant information and facilitate training. The deep feature then can be obtained as:

$$F_{DF} = W_{NSC}F_{SF} + F_G, \quad (7)$$

where W_{NSC} denotes the weight to the non-local skip connection. It can not only help to learn residual information from shallow features, but also stabilize the training of deep networks.

As a low-level visual task, there are abundant information in the LR images and the goal of SISR task is to reconstruct more useful information. Due to the non-local skip connection, the rich low-frequency information can be passed. To get better use of residual information, we stack B residual feature attention

blocks (RFABs) in each RAG. The b -th RFAB in the g -th RAG can be represented as:

$$F_{g,b} = H_{g,b}(F_{g,b-1}), \quad (8)$$

where $H_{g,b}(\cdot)$ is the function of the b -th RFAB in the g -th RAG. $F_{g,b-1}$ and $F_{g,b}$ denote the corresponding input and output. A local skip connection is introduced to gain the block output to extract more informative features via:

$$F_g = W_{LSC}F_{g-1} + F_{g,B}, \quad (9)$$

where W_{LSC} denotes the weight to the local skip connection. It keeps more abundant residual information. To extract more discriminative representations, we introduce our RFAB by channel- and spatial-wise feature rescaling with attention mechanism.

3.3 Residual Feature Attention Block (RFAB)

Most previous CNN-based SISR models treat features equally. To extract more informative features, channel attention (Zhang et al., 2018)(Dai et al., 2019) was introduced to better use the channel-wise features for SISR. However, they ignored exploiting spatial-wise information, thus hindering the discriminative ability of the network.

Inspired by the above observations, we propose a residual feature attention block (RFAB) module (see Figure 3). For the b -th RFAB in the g -th RAG, we get:

$$F_{g,b} = F_{g,b-1} + S_{g,b}(C_{g,b}(F'_{g,b-1})), \quad (10)$$

where $C_{g,b}(\cdot)$ and $S_{g,b}(\cdot)$ denote the function of channel attention and spatial attention respectively. $F_{g,b-1}$ and $F_{g,b}$ denote the input and output of RFAB. The middle feature $F'_{g,b-1}$ is gained by two stacked convolutional layers:

$$F'_{g,b-1} = W_{g,b}^2 \delta(W_{g,b}^1 F_{g,b-1}), \quad (11)$$

where $W_{g,b}^1$ and $W_{g,b}^2$ denote the weights of convolutional layers, and $\delta(\cdot)$ denotes the Mish activation function. Then we will introduce how to exploit channel and spatial information next.

Channel Attention. As shown in Figure 3, given the input feature map $F'_{g,b-1} = [f_1, \dots, f_C]$ which has C channel numbers with size of $H \times W$. We use global average pooling as the channel descriptor for simplicity. It shrinks $F'_{g,b-1}$ through spatial dimensions $H \times W$ to produce the channel-wise statistic $Z_C \in \mathbb{R}^C$, the c -th scalar of Z_C can be computed as:

$$z_c = H_{AP}(f_c) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W f_c(i, j), \quad (12)$$

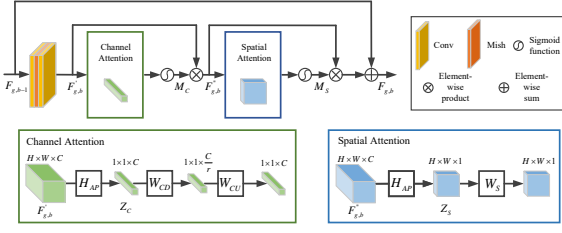


Figure 3: A RFAB receives a feature map $F_{g,b-1}$ of the size $H \times W \times C$ as an input and outputs a feature map $F_{g,b}$ of the same size. Channel attention and spatial attention are introduced to make better use of abundant feature information. Residual connections help to build a deep and efficient network.

where $H_{AP}(\cdot)$ and $f_c(i, j)$ denote the average pooling function and the value of c -th feature f_c at position (i, j) respectively.

Then a gating mechanism is introduced to exploit channel interdependencies from the aggregated information. Following (Hu et al., 2018), we use a simple sigmoid function as the gating function:

$$M_C = \sigma(W_{CU}f(W_{CD}Z_C)), \quad (13)$$

where $\sigma(\cdot)$ and $f(\cdot)$ denote the sigmoid and ReLU function. W_{CD} is the weight of channel-downscaling convolutional layer with reduction ratio r . W_{CU} is the weight of channel-upscaling convolutional layer with ratio r . Then we rescale the input with the channel attention map M_C :

$$F''_{g,b-1} = M_C \cdot F'_{g,b-1} \quad (14)$$

Spatial Attention. The process of spatial attention is similar to channel attention. Since we've obtained the channel reweighted feature $F''_{g,b-1} = [f'_1, \dots, f'_C]$, we use channel average pooling as the spatial descriptor. The spatial-wise statistic $Z_S \in \mathbb{R}^{H \times W}$ can be gained by shrinking channel dimension C . Then the (i, j) element of Z_S is computed by:

$$z_s(i, j) = H_{AP}(F''_{g,b-1}(i, j)) = \frac{1}{C} \sum_{c=1}^C f'_c(i, j), \quad (15)$$

where $H_{AP}(\cdot)$ and $f'_c(i, j)$ denote the average pooling function and the value of c -th feature f'_c at position (i, j) respectively. Then we use the same gating mechanism as channel attention:

$$M_S = \sigma(W_S Z_S), \quad (16)$$

where $\sigma(\cdot)$ and W_S denote the sigmoid function and the weight of a convolutional layer. Then we reweight the spatial dimension using map M_S and obtain the final output as discussed before:

$$F_{g,b} = F_{g,b-1} + M_C \cdot F''_{g,b-1} \quad (17)$$

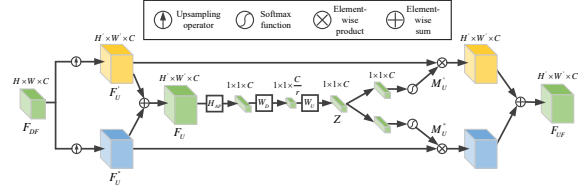


Figure 4: A UAB receives a low-resolution feature map of the size $H \times W \times C$ as an input and outputs a high-resolution feature map of the size $H' \times W' \times C$. We use parallel upsampling processes to obtain integrated reconstruction results. We only show a two-branch case, but it is easy to extend to multiple branches case.

3.4 Upsampling Attention Block (UAB)

To gain better reconstruction results, we propose an upsampling attention block (UAB) among multiple upsampling processes. Specifically, we produce several high-resolution features from the deep extracted feature, fuse them and output the ensemble feature using attention mechanism. Figure 4 shows a two-branch case. Therefore in this case, there are only two upsampling branches, but it's easy to extend to multiple branches cases.

For the extracted feature map $F_{DF} \in \mathbb{R}^{H \times W \times C}$, we apply P ($P = 2$ in Figure 4) upsampling transformations $F_{DF} \rightarrow F'_U \in \mathbb{R}^{H' \times W' \times C}$ and $F_{DF} F'_U \in \mathbb{R}^{H' \times W' \times C}$, respectively. Our goal is to ensemble features to adaptively obtain more accurate reconstructions, so we use the gating mechanism to control the information flows from multiple branches carrying different upsampled information. The gate fuses information from all branches to achieve this goal via an element-wise summation:

$$F_U = F'_U + F''_U \quad (18)$$

Then we embed the parallel information by a soft attention across channels. We compute the compact feature descriptor $Z \in \mathbb{R}^{1 \times 1 \times C}$ using the same way as described in Section 3.3, and apply a softmax operator on the channel-wise elements:

$$M'_U = \frac{e^{W'_U Z}}{e^{W'_U Z} + e^{W''_U Z}}, M''_U = \frac{e^{W''_U Z}}{e^{W'_U Z} + e^{W''_U Z}}, \quad (19)$$

where $W'_U, W''_U \in \mathbb{R}^{C \times C}$. In the case of two branches, the final upscaled feature map F_{UF} is obtained through the attention weights on parallel features:

$$F_{UF} = W'_U \cdot F'_U + W''_U \cdot F''_U \quad (20)$$

Note that we introduce the formulas for the two-branch case and one can easily generalize to more branches by extending Eqs. 18, 19 and 20.

3.5 More Details

In this subsection we show more implementation details of our proposed UAN. We use the RAG number

Table 1: Effectiveness of RAG structure. We show the best PSNR (dB) values on Set5 (4×) in 5.5×10^5 iterations.

local skip connection	×	✓	×	×	✓	✓	×	✓
channel attention	×	×	✓	×	✓	×	✓	✓
spatial attention	×	×	×	✓	×	✓	✓	✓
PSNR on Set5 (2×)	31.97	32.08	32.04	32.06	32.12	32.14	32.07	32.21

as $G = 10$. In each RAG, we use RFAB number as $B = 20$. We set the upsampling branches $P = 3$ in UAB. We use 1×1 as the size of convolution layers during downscaling and upscaling in channel attention, where the reduction ratio r is set as 16. For other convolutional filters, the number and size of filters are respectively set as 64 and 3×3 , and the same padding strategy is used to keep size fixed. For the upsampling method, we follow the work (Zhang et al., 2018)(Dai et al., 2019) and use ESPCNN (Shi et al., 2016) to reconstruct fine resolution features from coarse ones. One convolutional layer was used in the tail, which has 3 features to output RGB-channel color images.

4 EXPERIMENTS

4.1 Setups

Now we introduce experimental setups.

Datasets and Degradation Models. We use HR images from DIV2K (Timofte et al., 2017) datasets as training sets following (Zhang et al., 2018). Four standard benchmarks datasets are used for testing: Set5 (Bevilacqua et al., 2012), Set14 (Zeyde et al., 2010), BSDS100 (Martin et al., 2002), and Urban100 (Huang et al., 2015). We use Bicubic (BI) degradation models for experiments.

Training Settings. During training, we apply random rotation by 90° , 180° , 270° and horizontal flipping as data augmentation. We use 16 LR color patches with size 48×48 as inputs in training batches. Our model is trained by Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. We initialize the learning rate as 10^{-4} and then decrease one half every 500 epochs. All the experiments have been implemented on PyTorch on four Titan V GPUs.

Evaluation Metrics. All the results are evaluated using PSNR and SSIM metrics. In order to compare with state-of-the-art models, we transform the results from RGB space to YCbCr space, and evaluate them on Y channel.

4.2 Ablation Study

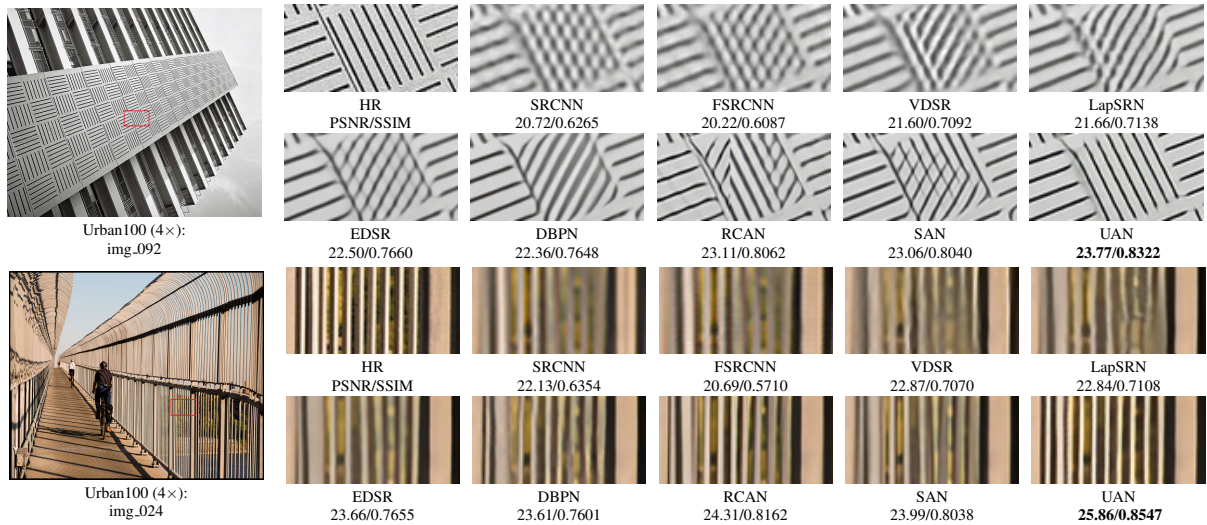
In this subsection, we study the effectiveness about two main components of our UAN, residual attention group (RAG) and upsampling attention block (UAB).

Residual Attention Group (RAG). We train RAG with its variants and test them on Set5 benchmark to demonstrate the effects of different structures, and the evaluation results are listed in Table 1. We set a basic baseline model that only contains convolutional layers, which has the same layer number as all the other control models. We can see that the baseline model obtains PSNR = 31.97 dB on Set5 ($\times 2$). When we add different structures to the module, the performance improves to varying degrees. Specifically, the performance is improved to 32.08 dB when the local skip connection is applied alone. We further observe that we can always obtain better performance when we combine it with other methods. These results show that local skip connection is essential for building a deep network. We also study the effect of channel attention and spatial attention. When we add channel attention or spatial attention, the performance move to 32.12 dB and 32.14 dB. It indicates that paying more attention to the extracted features is more important than naively deepening the network. When we put all the structures together, the performance reaches 32.21 dB. These comparisons demonstrate the effectiveness of our RAG.

Upsampling Attention Block (UAB). We further study the number of parallel upsampling processes on final performance and model complexity. Specifically, we set the parameter P , the number of upsampling transformations, to 1, 2, 3 and 4 respectively. We show the PSNR values under different conditions and mark the parameters of different networks in Table 2. We observe that when there are more parallel paths, the model can converge to a higher PSNR value. This shows that although parallel upsampling processes share the same feature extraction part, they can produce different high-resolution features. Integrating these features by attention mechanism can effectively improve performance at the cost of a small increase in parameters. In order to compare with other models, we set $P = 3$ in this paper, which makes our UAN has similar parameters as SAN.

Table 2: Investigations of UAB structure. All the models are evaluated on Set5 (2×) in 5×10^4 iterations.

Branch	1	2	3	4
PSNR	37.90	37.94	37.96	37.97
Para.	15.4M	15.6M	15.7M	15.9M

Figure 5: Visual comparisons for 4× SISR on Urban100 dataset. The best results are **highlighted**.

4.3 Results on Single Image Super-resolution

To show the advantage of our UAN, we compare our model with 8 state-of-the-art models: SRCNN (Dong et al., 2014), FSRCNN (Dong et al., 2016), VDSR (Kim et al., 2016), LapSRN (Lai et al., 2017), EDSR (Lim et al., 2017), DBPN (Muhammad et al., 2018), RCAN (Zhang et al., 2018), and SAN (Dai et al., 2019). Following RCAN, we also apply self-ensemble strategy to our model denoted as UAN+ to further improve performance.

Objective Metrics. Quantitative results by PSNR/SSIM are shown in Table 3. Our UAN+ obtains the best results on all the datasets on different scaling factors compared with other models. Even without self-ensemble, our UAN performs comparable results as RCAN and SAN and outperforms other methods. We observe that there is a performance gap between attention based models and other models, and UAN performs best among attention based models. It’s because our model can simultaneously utilize channel- and spatial-wise feature correlations for stronger feature expressions. We further notice that our model performs better on higher scaling factors (e.g., the PSNR gains of UAN over SAN for Set5 are 0.13 dB on 8× and 0.02 dB on 2×). This is because when the factor is larger, the LR image can provide less information. If the model does not learn the appropriate features during upsampling, it will cause information bottlenecks.

Visual Qualities. We show visual results compared to other methods in Figure 5. We can observe that most of the SISR models fail to recover the patterns and suffer from severe blur artifacts. In contrast, our

Table 3: Performance comparison with state-of-the-art algorithms for SISR. Best results are **bold numbers**.

Method	Scale	Set5		Set14		BSDS100		Urban100	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	×2	33.66	0.9299	30.24	0.8688	29.56	0.8431	26.88	0.8403
SRCNN	×2	36.66	0.9542	32.45	0.9067	31.36	0.8879	29.50	0.8946
FSRCNN	×2	37.05	0.9560	32.66	0.9090	31.53	0.8920	29.88	0.9020
VDSR	×2	37.53	0.9590	33.05	0.9130	31.90	0.8960	30.77	0.9140
LapSRN	×2	37.52	0.9591	33.08	0.9130	31.08	0.8950	30.41	0.9101
EDSR	×2	38.11	0.9602	33.92	0.9195	32.32	0.9013	32.93	0.9351
D-DBPN	×2	38.09	0.9600	33.85	0.9190	32.27	0.9000	32.55	0.9324
RCAN	×2	38.27	0.9614	34.12	0.9216	32.41	0.9027	33.34	0.9384
SAN	×2	38.31	0.9620	34.07	0.9213	32.42	0.9028	33.10	0.9370
UAN (ours)	×2	38.33	0.9623	34.11	0.9214	32.42	0.9030	33.35	0.9389
UAN+ (ours)	×2	38.35	0.9624	34.17	0.9223	32.51	0.9040	33.52	0.9401
Bicubic	×4	28.42	0.8104	26.00	0.7027	25.96	0.6675	23.14	0.6577
SRCNN	×4	30.48	0.8628	27.50	0.7513	26.90	0.7101	24.52	0.7221
FSRCNN	×4	30.72	0.8660	27.61	0.7550	26.98	0.7150	24.62	0.7280
VDSR	×4	31.35	0.8830	28.02	0.7680	27.29	0.7251	25.18	0.7540
LapSRN	×4	31.54	0.8850	28.19	0.7720	27.32	0.7270	25.21	0.7560
EDSR	×4	32.46	0.8968	28.80	0.7876	27.71	0.7420	26.64	0.8033
D-DBPN	×4	32.47	0.8980	28.82	0.7860	27.72	0.7400	26.38	0.7946
RCAN	×4	32.63	0.9002	28.87	0.7889	27.77	0.7436	26.82	0.8087
SAN	×4	32.64	0.9003	28.92	0.7888	27.78	0.7436	26.79	0.8068
UAN (ours)	×4	32.66	0.9005	28.92	0.7889	27.79	0.7439	26.86	0.8107
UAN+ (ours)	×4	32.71	0.9016	28.97	0.7922	27.83	0.7453	27.02	0.8129
Bicubic	×8	24.40	0.6580	23.10	0.5660	23.67	0.5480	20.74	0.5160
SRCNN	×8	25.33	0.6900	23.76	0.5910	24.13	0.5660	21.29	0.5440
FSRCNN	×8	20.13	0.5520	19.75	0.4820	24.21	0.5680	21.32	0.5380
VDSR	×8	25.93	0.7240	24.26	0.6140	24.49	0.5830	21.70	0.5710
LapSRN	×8	26.15	0.7380	24.35	0.6200	24.54	0.5860	21.81	0.5810
EDSR	×8	26.96	0.7762	24.91	0.6420	24.81	0.5985	22.51	0.6221
D-DBPN	×8	27.21	0.7840	25.13	0.6480	24.88	0.6010	22.73	0.6312
RCAN	×8	27.31	0.7878	25.23	0.6511	24.96	0.6058	22.97	0.6452
SAN	×8	27.22	0.7829	25.14	0.6476	24.88	0.6011	22.70	0.6314
UAN (ours)	×8	27.35	0.7880	25.22	0.6509	24.98	0.6053	22.96	0.6433
UAN+ (ours)	×8	27.41	0.7909	25.25	0.6518	25.00	0.6060	22.98	0.6460

UAN recovers more details and reconstructs accurate results. For “img_092”, most compared models generate the lines with wrong directions or even cannot produce lines, and only our UAN recovers the right result. For “img_024”, the early models (e.g., SRCNN, FSRCNN, VDSR and LapSRN) fail to recover the main structure. The recent proposed EDSR and DBPN can reconstruct the basic contour but lose details. Compared with HR image, RCAN, SAN and UAN gain sharp results and reconstruct more details, but UAN obtains higher qualities. These comparisons demonstrate that our UAN can better utilize spatial and channel features for recovering more finer results.

5 CONCLUSION

In this work, we propose a deep upsampling attention network (UAN) for accurate SISR. Specifically, the residual attention groups (RAGs) based structure allows UAN to capture the structure and frequency information by rescaling spatial- and channel-wise features. Meanwhile, RAG allows abundant residual information to be bypassed through non-local skip connections, making the network more effective. Furthermore, in addition to improve the ability of our model, we propose an upsampling attention block (UAB) to adaptively combine parallel upsampled features by considering correlations among them. Experimental results on standard benchmarks show that our UAN achieves better accuracy and visual quality over state-of-the-art methods.

ACKNOWLEDGEMENTS

This work is funded by the National Key Research and Development Program of China under the program number 2017YFC0810200.

REFERENCES

- Freeman, W.T., Pasztor, E.C., and Carmichael, O.T. (2000). Learning low-level vision. In *International Journal of Computer Vision (IJCV)*, pages 25–47. Springer.
- Dong, C., Loy, C., He, K., and Tang, X. (2014). Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision (ECCV)*, pages 184–199. Springer.
- Kim, J., Lee, J., and Lee, K. (2016). Accurate image super-resolution using very deep convolutional networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1646–1654. IEEE.
- Lai, W., Huang, J., Ahuja, N., and Yang, M. (2017). Deep laplacian pyramid networks for fast and accurate super-resolution. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 624–632. IEEE.
- Dong, C., Loy, C., and Tang, X. (2016). Deeply recursive convolutional network for image super-resolution. In *European Conference on Computer Vision (ECCV)*, pages 1637–1645. Springer.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE.
- Lim, B., Son, S., Kim, H., Nah, S., and Lee, K. (2017). Enhanced deep residual networks for single image super-resolution. In *Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 136–144. IEEE.
- Muhammad, H., Shakhnarovich, G., and Ukita, N. (2018). Deep back-projection networks for super-resolution. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1664–1673. IEEE.
- Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., and Fu, Y. (2018). Image super-resolution using very deep residual channel attention networks. In *European Conference on Computer Vision (ECCV)*, pages 286–301. Springer.
- Dai, T., Cai, J., Zhang, Y., Xia, S., and Zhang, L. (2019). Second-order attention network for single image super-resolution. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11065–11074. IEEE.
- Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A., Bishop, R., Rueckert, D., and Wang, Z. (2016). Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883. IEEE.
- Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-excitation networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7132–7141. IEEE.
- Woo, S., Park, J., Lee, J., and S.K. (2018). Cbam: Convolutional block attention module. In *European Conference on Computer Vision (ECCV)*, pages 3–19. Springer.
- Timofte, R., Agustsson, E., Gool, L., Yang, M., and Zhang, L. (2017). Ntire 2017 challenge on single image super-resolution: Methods and results. In *Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 114–125. IEEE.
- Bevilacqua, M., Roumy, A., Guillemot, C., and Morel, A. (2012). Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *The British Machine Vision Conference (BMVC)*, pages 135.1–135.1. Springer.
- Zeyde, R., Elad, M., and Protter, M. (2010). On single image scale-up using sparse-representations. In *7th Int. Conf. Curves Surf.*, pages 711–730.
- Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2002). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *International Conference on Computer Vision (ICCV)*, pages 416–423. IEEE.
- Huang, J., Singh, A., and Ahuja, N. (2015). Single image super-resolution from transformed self-exemplars. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5197–5206. IEEE.