RESIDUAL ATTENTION BLOCK SEARCH FOR LIGHTWEIGHT IMAGE SUPER-RESOLUTION

Wenrui Liao[1,2,3], *Zhong-Qiu Zhao*[1,2,3*], *Hao Shen*[1,2,3], *Weidong Tian*[1,2,3]

[1] School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, 230009, China
[2] Intelligent Interconnected Systems Laboratory of Anhui Province (Hefei University of Technology)
[3]Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
{z.zhao,wdtian}@hfut.edu.cn, 2019170980@mail.hfut.edu.cn, haoshenhs@gmail.com

2021 IEEE International Conference on Multimedia and Expo (ICME) | 978-1-6654-3864-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICME51207.2021.9428408

ABSTRACT

Recently, lightweight neural networks with different manual designs have presented a promising performance in single image super-resolution (SR). However, these designs rely on too much expert experience. To address this issue, we focus on searching a lightweight block for efficient and accurate image SR. Due to the frequent use of various residual blocks and attention mechanisms in SR methods, we propose the residual attention search block (RASB) which combines an operation search block (OSB) with an attention search block (ASB). The former is used to explore the suitable operation at the proper position, and the latter is applied to discover the optimal connection of various attention mechanisms. Moreover, we build the modified residual attention network (MRAN) with stacked found blocks and a refinement module. Extensive experiments demonstrate that our MRAN achieves a better trade-off against the state-of-the-art methods in terms of accuracy and model complexity.

Index Terms— Lightweight super-resolution, deep learning, residual search attention block, modified residual attention network

1. INTRODUCTION

Single image super-resolution (SISR) aims to reconstruct a visually pleasing high-resolution (HR) image from its degraded low-resolution counterpart. However, SISR is a typical ill-posed problem since multiple HR solutions can be degraded to one LR input. To tackle this issue, recently numerous methods [1, 2, 3, 4, 5] based on deep convolutional neural networks have been proposed. However, they have new growing requirements in terms of model capacity and computational complexity, which is not conducive to practical applications.

*Corresponding Author. This work was supported in part by the National Natural Science Foundation of China under Grants 61976079 & 61672203, in part by Anhui Natural Science Funds for Distinguished Young Scholar under Grant 170808J08, and in part by Anhui Key Research and Development Program under Grant 202004a05020039.

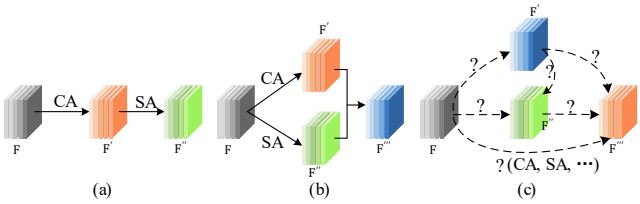


Fig. 1. Examples of various connection patterns of attention mechanism, where CA and SA denote channel attention and spatial attention. (a) Two sequential attentions. (b) Two parallel attentions. (c) Our attention search space contains potential connection patterns of various attentions. (a) and (b) are special cases of (c).

To reduce model parameters, some existing SR methods concentrate on the architecture design, such as cascading mechanism [6] and multi-distillation block [7]. These designs may be lightweight but rely on overweight expert experience, which leads to spending considerable consumption on unnecessarily repetitive designs. Other methods propose various attention mechanisms such as channel attention [1] and spatial attention [8]. As shown in Fig.1 (a) and (b), channel attention and spatial attention can be organized in a sequential or parallel manner. The capability of attention mechanism for feature representation is subject to previous artificial arrangements. Inspired by neural architecture search (NAS) which is a process of automating architecture engineering, we design an operation search block (OSB) to integrate all possible operation types (*i.e.*, a residual block). It helps to discover the optimal operation at the appropriate location. For the design of attention search space, we have two options: one of which is to embed various independent attention modules in an OSB to form a multi-branch structure, and the other is to build an attention search block (ASB) which can emphasize meaningful features in a progressive way. Obviously, the former just adds more branches to the connection as shown in Fig.1 (b), and the latter helps to discover potential forms of connection and inner correlation of various attention mechanisms. Therefore, an ASB structure depicted in Fig.1 (c) should be the better

978-1-6654-3864-3 / 21 / $ 31.00©2021 IEEE

choice. To our best knowledge, the combination of residual blocks and attention mechanisms allows low-frequency features to frequently propagate to the next layer via skip connections, while allowing high-frequency features to be enhanced in some dimension, respectively. Therefore, based on OSB and ASB, we design the residual attention search block (RASB) to build a search network, as shown in Fig.2 (a). After the search phase, we can get the modified version of the RASB named modified residual attention block (MRAB).

Besides, sub-pixel convolution is adopted to expand resolution for the reconstruction process, which is not conducive to a lightweight SR network due to a large number of parameters. Moreover, this convolution can not take advantage of the high-level content information extracted from previous residual blocks. To address this issue, we propose the cross-scale refinement module (CSRM) which extracts multi-scale features through parallel pixel-wise convolution modules of different filter sizes. To further enhance the information communication between multi-scale features, we utilize a butterfly structure [9] for the proposed CSRM, which generates various linear combination of multi-scale features. Based on MRAB and CSRM, we build a lightweight SR network named modified residual attention network (MRAN), as depicted in Fig.2 (b). In summary, the main contributions of this paper are: (1) We firstly utilize differentiable architecture search algorithm to search a residual attention search block (RASB) which combines an operation search block (OSB) with an attention search block (ASB); (2) We propose the cross-scale refinement module which can be embedded in MRAN to obtain better SR performance without additional parameters; (3) Extensive experiments on four benchmark datasets demonstrate the superiority of all mentioned search blocks and our proposed MRAN.

2. RELATED WORK

Our work is most closely related to DARTS [10] and PC-DARTS [11]. DARTS is an algorithm based on the continuous relaxation of architecture representation, allowing an efficient search of the architecture using gradient descent. PC-DARTS proposes a simple approach that samples a subset of channels into the operation selection block while bypassing the rest directly in a shortcut. In this paper, we combine this sampling strategy with the gradient-based algorithm as our search strategy. Besides, we employ various lightweight residual blocks with dense connections to build our search space. There are two more relevant works. FALSR [12] introduces elastic neural architecture search to SR and achieves excellent results. ESRN [13] proposes an efficient residual dense block search algorithm to obtain better SR performance with multiple objectives. However, both of these methods search on discrete structures, which leads to a relatively large amount of computations and a high demand for search time. In contrast, our proposed search method can discover an efficient architecture of residual attention block with one-shot

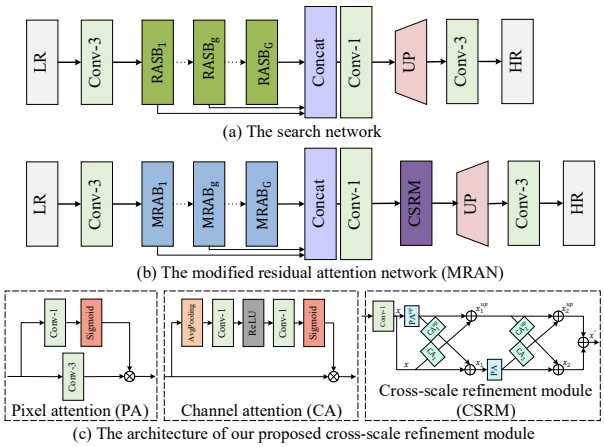


Fig. 2. The overall architecture of the proposed search network and the modified residual attention network (MRAN). The 'Up' modules denote sub-pixel convolution [16].

search period for $\times 2$ SR, and then the searched architecture is adopted to SR tasks of other scales (*e.g.*, $\times 3$ and $\times 4$).

Nowadays, the attention mechanism is employed in numerous computer vision tasks. SENet [14] adaptively recalibrates channel-wise feature responses by explicitly modeling codependency between channels. CANet [15] proposes pixel-wise attention to filter pixel spatial information. CBAM [8] utilizes spatial attention module to learn the inner-spatial relationship of features. In this paper, we explore the potential combinations and correlations between various attention mechanisms by building the attention search block without getting bogged down in the details of their internal implementation.

3. PROPOSED METHOD

3.1. Residual attention Search block

It has been demonstrated that the residual block with proper attention mechanisms can be used to improve performance in SR tasks [1, 3, 4, 5]. However, few previous methods consider the connection of residual blocks or various attention modules. As shown in Fig.3, our RASB mainly consists of the operation search block (OSB) and the attention search block (ASB). We denote O_{g-1} and O_g as the input and output of the RASB at the g-th layer. The process of feature propagation in the RASB can be formulated as

$$F_0 = OSB_g(O_{g-1}) \tag{1}$$

$$O_g = ASB_g(F_0) + O_{g-1}, \tag{2}$$

where F_0 denotes the output and input for the g-th OSB and the g-th ASB.

3.1.1. Operation search block

The OSB adopts 1×1 convolution layer at the beginning namely $Conv(\cdot)$. Given the input feature O_{g-1}, we have:

$$S_0 = Conv(O_{g-1}), \tag{3}$$

Then, we denote N_1 operation nodes with the index from 0 to $N_1 - 1$. Each node takes the outputs of all previous nodes as input and outputs a feature map. Note that the output of the node with index 0 equals to S_0. Taking the i-th node as an example, the output of this node is calculated as follows:

$$S_j = \sum_{i=0}^{j-1} O_{(i,j)}(S_i), 0 < j \leq N_1 - 1 \tag{4}$$

where S_i is the output of the i-th node. $O_{(i,j)}(\cdot)$ represents the operation flow that transforms S_i from the i-th node to j-th node, where $i < j$. Here, to implement this transformation, we adopt the channel sampling strategy [11] to sample a subset of channels into the operation flow as the following:

$$O_{(i,j)}(S_i) = Conact((1-M_{(i,j)})*S_i, \sum_{k=1}^{R} \omega_{(i,j)}^{OP_k} \cdot OP_k(M_{(i,j)}*S_i)), \tag{5}$$

where $Concat(\cdot)$ denotes the concatenation operation, $\{OP_1, OP_2, \cdots, OP_R\}$ denote R possible operation types, and $\omega_{(i,j)}^{OP_k}$ corresponds to the weight of operation OP_k. $M_{(i,j)}$ denotes the mask which assigns 1 to the selected channels and 0 to the remaining ones. Therefore, $M_{(i,j)} * S_i$ and $(1 - M_{(i,j)}) * S_i$ represent the selected and remaining channels, respectively.

Finally, the outputs of all nodes are fused by the concatenation operation followed by a 1×1 convolution layer. We denote F_0 as the output of OSB. For the SR task, we redesign the set of candidate operations, *i.e.*, $\{OP_1, OP_2, \cdots, OP_R\}$. By introducing the design of residual dense block (RDB) [2], we assume the convolution number is 2 and the growth rate is 16. The basic convolution is replaced with residual block, shallow residual block [17] and lightweight residual block [18], respectively. Consequently, we get 5 types of candidate operations while adding skip connection and none [10], which correspond to $OP_1 \sim OP_5$ as depicted in Fig.3.

3.1.2. Attention search block

In this subsection, we first present our proposed attention search block. As illustrated in Fig.3, the ASB is a directed acyclic graph containing a sequence of N_2 nodes. Each node is a potential representation of features, and each directed edge is regarded as an attention flow. Taking the flow from the i-th node to the j-th node for example, where $i < j$, the core idea of an attention flow is to formulate the features propagated from i to j as a weighted summation of T candidate attentions,

$$A_{(i,j)}(F_i) = \sum_{k=1}^{T} \mu_{(i,j)}^{AT_k} \cdot AT_k(F_i), \tag{6}$$

where $\{AT_1, AT_2, \cdots, AT_T\}$ denote T possible attention types. F_i denotes the output of the i-th attention node. We mix candidate attentions in a continuous relaxation way by weighting $AT_k(\cdot)$ with $\mu_{(i,j)}^{AT_k}$. The output of each node in ASB

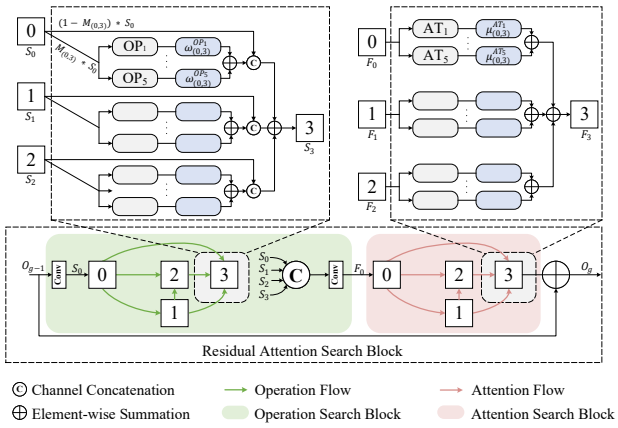


Fig. 3. The architecture of the residual attention search block (RASB) and its components: operation search block (OSB) and attention search block (ASB). Each node with an index outputs a latent representation (*e.g.,* a feature map). In OSB, an operation flow from the i-th node to j-th node is formulated by weighting candidate operations (denoted as OP_k) with a set of hyper-parameters, namely, $\left\{\omega_{(i,j)}^{OP_k}\right\}$. We only sample part of input features with a mask $M_{i,j}$ in the channel dimension. In ASB, the mixing process of our candidate attentions (denoted as AT_k) with $\mu_{(i,j)}^{AT_k}$ is denoted as an attention flow from the i-th node to j-th node.

is the summation of all associated attention flows, which is:

$$F_j = \sum_{i=0}^{j-1} A_{(i,j)}(F_i), 0 < j \leq N_2 - 1 \tag{7}$$

Note that the output of the first node equals to F_0. The output of ASB is denoted as F_{N_2-1}. Due to the residual structure in RASB, we compute the output of RASB by summing F_{N_2-1} with O_{g-1}.

We introduce a variety of attention mechanisms to build attention search space. These attention mechanisms can be divided into three categories. The first is based on the channel level, such as channel-wise attention [1]. The second type is spatial including SAM [8] and ESA [5]. The third type is based on pixel-wise, such as CEA [19]. Besides, skip connection is added to attention search space. These attention mechanisms and skip connection are all convenient to be embedded in our attention search block, namely $AT_1 \sim AT_5$, as shown in Fig.3.

3.2. From search to evaluate

The architecture of the search network is shown in Fig.2 (a). The goal of search stage is to determine the best sets of hyper-parameters, *i.e.* $\left\{\omega_{(i,j)}^{OP_k}\right\}$ and $\left\{\mu_{(i,j)}^{AT_k}\right\}$. After the search stage is finished, for each operation flow $O_{(i,j)}(\cdot)$ and attention flow $A_{(i,j)}(\cdot)$, we select the operation and attention with the maximum value which is determined by $\omega_{(i,j)}^{OP_k}$ and $\mu_{(i,j)}^{AT_k}$, respec-

tively. Thereby, we obtain the exact architecture of modified residual attention block (MRAB). Based on these blocks, we build a lightweight SR network named modified residual attention network (MRAN), as illustrated in Fig.2 (b).

To alleviate the burden of the increased number of parameters due to sub-pixel convolution, we propose a cross-scale refinement module (CSRM) which reduces the channel dimension without losing contextual information of deep features. As shown in Fig.2 (c), our CSRM is a multi-branch structure with pixel-attention modules [20] in each branch and contains several channel attention modules between two branches. We employ a 1×1 convolution layer at the beginning to reduce feature dimension by half. Let x denote the features after the reduction operation. We can denote the linear combination of feature propagation as the following equations:

$$x_1^{up} = PA^{up}(x) + CA_1(x) \tag{8}$$

$$x_1 = CA_1^{up}(PA^{up}(x)) + x \tag{9}$$

$$x_2^{up} = CA_2(PA(x_1)) + x_1^{up} \tag{10}$$

$$x_2 = CA_2^{up}(x_1^{up}) + PA(x_1) \tag{11}$$

where the superscript ^{up} denotes a module in the upper branch, and the subscripts $_1$ or $_2$ denote the order in which the modules appear on the same branch. Notably, we replace the 3×3 convolution with 5×5 convolution in the PA^{up} module to extract multi-scale spatial information. Finally, x_2^{up} and x_2 are summed to obtain the refinement feature named x', which is propagated to the upsampling module to generate SR images.

4. EXPERIMENTS

4.1. Datasets and implementation details

The DIV2K [21] dataset is adopted in both the search and evaluation stages. We denote its training images as W and its validation images as A. The input size of the LR image is set to 64 × 64. The paired data is augmented by random flipping and rotation. In the search stage, W and A are used to optimize the kernels of convolution layers and the hyper-parameters $\omega_{(i,j)}^{OP_k}$ and $\mu_{(i,j)}^{AT_k}$, respectively. To be fair in comparison, we test our model on four public SR benchmark datasets: Set5 [22], Set14 [23], B100 [24], and Urban100 [25]. We employ peak signal to noise ratio (PSNR) and structural similarity (SSIM) on the Y channel of the YCbCr color space as the evaluation metrics. We adopt Mult-Adds to evaluate the computational complexity of a model, which denote the number of composite multiply-accumulate operations for a single image. The HR image size is set to 1280×720.

The numbers of G in the search network and MRAN are set to 4 and 5, respectively. The number of filters in candidate operations and attentions are set to 16 and 64, respectively. We train the search network for 100 epochs with batch size of 16. To learn the kernels and hyper-parameters, we use two

Table 1. Comparison of the number of parameters and mean values of PSNR obtained by All-RB, All-LRB, All-SRB, and MRB. We record the best results for ×4 SR in 500 epochs.

Operation type	Params	Set5	Set14	B100	Urban100
All-RB	631K	31.89	28.42	27.44	25.64
All-LRB	484K	31.74	28.33	27.38	25.48
All-SRB	520K	31.80	28.37	27.41	25.57
MRB	588K	**31.91**	**28.44**	**27.47**	**25.69**

Table 2. Comparison of the number of parameters and mean values of PSNR obtained by RB, RB-CA, RB-SA, RB-PA and RB-MA. We record the best results for ×4 SR in 500 epochs.

Attention type	Params	Set5	Set14	B100	Urban100
RB	610K	31.60	28.26	27.34	25.37
RB-CA	615K	31.67	28.27	27.36	25.44
RB-SA	617K	31.74	28.40	27.40	25.53
RB-PA	627K	31.89	28.42	27.44	25.64
RB-MA	651K	**31.95**	**28.48**	**27.48**	**25.80**

Adam optimizer with β_1 = 0.9, β_2 = 0.999. Both learning rate and weight decay are fixed at 0.0001. In the evaluation stage, we only use W to train our modified models for 1000 epochs with batch size of 16. We use Adam optimizer with β_1 = 0.9, β_2 = 0.999. The initial learning rate is set as 0.0001 and then decreases to half for every 200 epochs. We utilize L1 Loss as our loss function and PyTorch framework to implement our models with a GTX 2080Ti GPU.

4.2. Effects of OSB

To demonstrate the effect of operation search block, we obtain the modified residual block (MRB) from the MRAB. Then, we replace the operations between paired nodes in the MRB with residual blocks (All-RB), lightweight residual blocks (All-LRB) and shallow residual blocks (All-SRB), respectively. In this experiment, we use the search network as the basic network, and then replace the RASBs with the same amount of All-RBs, All-LRBs, All-SRBs and MRBs, respectively. From Table 1, it is obviously observed that our MRB outperforms the other three types of operation. Compared with All-RB, our MRB could improve the PSNR by 0.02dB, 0.02dB, 0.03dB and 0.05dB on Set5, Set14, B100 and Urban100 with fewer parameters. It indicates that the operation search block has found the optimal architecture from candidate operations.

4.3. Effects of ASB

We obtain the modified attention (MA) from MRAB. To evaluate the effect of MA, we use the search network as the basic network. Then, we replace the RASBs with the same number of residual blocks (RB), residual blocks with channel attention (RB-CA), residual blocks with spatial attention (RB-SA), residual blocks with pixel attention (RB-PA) and residual blocks with modified attention (RB-MA), respectively.

Table 3. Investigations of MRAB and CSRM. We record the best PSNR for $\times 4$ SR in 500 epochs.

MRAB	✗	✓	✗	✓
CSRM	✗	✗	✓	✓
Params	610K	629K	411K	430K
PSNR on Set5	31.60	31.98	31.79	**32.04**
PSNR on Set14	28.27	28.52	28.35	**28.54**

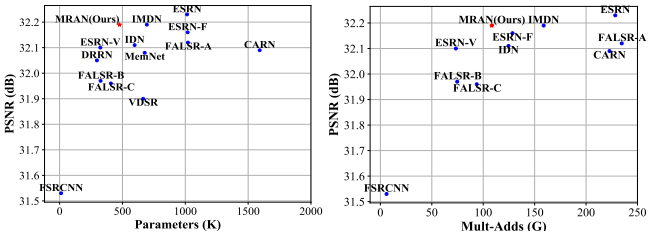
Note that all mentioned attention layers are embedded in the tail of RB. From Table 2, RB-SA and RB-PA could improve the PSNR by 0.14dB and 0.16dB on Set14 ($\times 4$), respectively. MA outperforms the other attentions and significantly improves the PSNR compared with the baseline. It indicates that we have found an effective attention module based on the attention search block.

Table 4. Quantitative results on benchmark datasets.The best and second best results are in red and blue, respectively.

Method	Scale	Params	Mult-Adds	Set5 PSNR/SSIM	Set14 PSNR/SSIM	B100 PSNR/SSIM	Urban100 PSNR/SSIM
Bicubic	× 2	-	-	33.66/0.9299	30.24/0.8688	29.56/0.8431	26.88/0.8403
FSRCNN [26]		12K	6.0G	37.05/0.9560	32.66/0.9090	31.53/0.8920	29.88/0.9020
VDSR [27]		665K	612.6G	37.53/0.9590	33.05/0.9130	31.90/0.8960	30.77/0.9140
DRRN [28]		297K	6,796.9G	37.74/0.9591	33.23/0.9136	32.05/0.8973	31.23/0.9188
MemNet [29]		677K	2,662.4G	37.78/0.9597	33.28/0.9142	32.08/0.8978	31.31/0.9195
IDN [30]		579K	124.6G	37.85/0.9598	33.58/0.9178	32.11/0.8989	31.95/0.9266
CARN [6]		1,592K	222.8G	37.76/0.9590	33.52/0.9166	32.09/0.8978	31.92/0.9256
FALSR-A [12]		1,021K	234.7G	37.82/0.9595	33.55/0.9168	32.12/0.8987	31.93/0.9256
FALSR-B [12]		326K	74.7G	37.61/0.9585	33.29/0.9143	31.97/0.8967	31.28/0.9191
FALSR-C [12]		408K	93.7G	37.66/0.9586	33.26/0.9140	31.96/0.8965	31.24/0.9187
IMDN [7]		694K	158.8G	38.00/0.9605	33.63/0.9177	32.19/0.8996	32.17/0.9283
ESRN [13]		1,014K	228.4G	38.04/0.9607	33.71/0.9185	32.23/0.9005	32.37/0.9310
ESRN-F [13]		1,019K	128.5G	37.93/0.9602	33.56/0.9171	32.16/0.8996	31.99/0.9276
ESRN-V [13]		324K	73.4G	37.85/0.9600	33.42/0.9161	32.10/0.8987	31.79/0.9248
MRAN (Ours)		476K	108.4G	38.05/0.9608	33.63/0.9180	32.19/0.8997	32.17/0.9284
Bicubic	× 3	-	-	30.39/0.8682	27.55/0.7742	27.21/0.7385	24.46/0.7349
FSRCNN [26]		12K	5.0G	33.18/0.9140	29.37/0.8240	28.53/0.7910	26.43/0.8080
VDSR [27]		665K	612.6G	33.67/0.9210	29.78/0.8320	28.83/0.7990	27.14/0.8290
DRRN [28]		297K	6,796.9G	34.03/0.9244	29.96/0.8349	28.95/0.8004	27.53/0.8378
MemNet [29]		677K	2,662.4G	34.09/0.9248	30.00/0.8350	28.96/0.8001	27.56/0.8376
IDN [30]		588K	56.3G	34.24/0.9260	30.27/0.8408	29.03/0.8038	27.99/0.8489
CARN [6]		1,592K	118.8G	34.29/0.9255	30.29/0.8407	29.06/0.8034	28.06/0.8493
IMDN [7]		703K	71.5G	34.36/0.9270	30.32/0.8417	29.09/0.8046	28.17/0.8519
ESRN [13]		1,014K	115.6G	34.46/0.9281	30.43/0.8439	29.15/0.8072	28.42/0.8579
ESRN-F [13]		1,019K	71.7G	34.32/0.9268	30.35/0.8410	29.09/0.8046	28.11/0.8512
ESRN-V [13]		324K	36.2G	34.23/0.9262	30.27/0.8400	29.03/0.8039	27.95/0.8481
MRAN (Ours)		522K	52.9G	34.37/0.9272	30.37/0.8424	29.09/0.8047	28.16/0.8519
Bicubic	×4	-		28.42/0.8104	26.00/0.7027	25.96/0.6675	23.14/0.6577
FSRCNN [26]		12K	4.6G	30.72/0.8660	27.61/0.7550	26.98/0.7150	24.62/0.7280
VDSR [27]		665K	612.6G	31.35/0.8830	28.02/0.7680	27.29/0.7260	25.18/0.7540
DRRN [28]		297K	6,796.9G	31.68/0.8888	28.21/0.7720	27.38/0.7284	25.44/0.7638
MemNet [29]		677K	2,662.4G	31.74/0.8893	28.26/0.7723	27.40/0.7281	25.50/0.7630
IDN [30]		600K	32.3G	31.99/0.8928	28.52/0.7794	27.52/0.7339	25.92/0.7801
CARN [6]		1,592K	90.9G	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837
IMDN [7]		715K	40.9G	32.21/0.8948	28.58/0.7811	27.56/0.7353	26.04/0.7838
ESRN [13]		1,014K	66.1G	32.26/0.8957	28.63/0.7818	27.62/0.7378	26.24/0.7912
ESRN-F [13]		1,019K	41.4G	32.15/0.8940	28.59/0.7804	27.59/0.7354	26.11/0.7851
ESRN-V [13]		324K	20.7G	31.99/0.8919	28.49/0.7779	27.50/0.7331	25.87/0.7782
MRAN (Ours)		513K	35.6G	32.21/0.8949	28.60/0.7812	27.59/0.7355	26.04/0.7839

4.4. Effects of MRAB and CSRM

To a fair comparison, we replace the RASB in the search network with the RB to build a baseline model. In Table 3, the baseline achieves the lowest PSNR value on Set14 ($\times 4$). When MRAB or CSRM is adopted, the PSNR values are increased by $+0.25$dB and $+0.08$dB compared with the baseline on Set14 ($\times 4$), respectively. Note that the model only with CSRM has 1/3 fewer parameters than the baseline.



(a) PSNR *vs.* Parameters. (b) PSNR *vs.* Mult-Adds.

Fig. 4. Trade-off between performance *vs.* capacity and computational complexity of the model on B100 ($\times 2$) dataset.

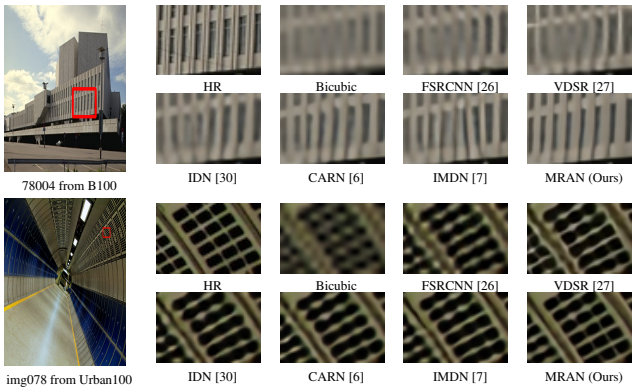


Fig. 5. Visual comparison for $4\times$ SR.

Moreover, the results of the last column demonstrate that the combination of our proposed MRAB and CSRM achieves a comprehensive balance of the number of parameters and performance.

4.5. Comparisons with the state-of-the-arts

We compare our MRAN with 9 state-of-the-art methods: FSRCNN [26], VDSR [27], DRRN [28], MemNet [29], IDN [30], CARN [6], FALSR [12], IMDN [7], and ESRN [13]. Table 4 shows quantitative comparisons for $\times 2$, $\times 3$, and $\times 4$ SR. We can observe that our MRAN achieves comparable or better performance compared with the state-of-the-art lightweight models. As shown in Fig.4 (a), though ESRN outperforms all previous methods, it has more parameters than most of the lightweight models. On the contrary, our MRAN achieves comparable performance compared with ESRN with half of its parameters. To get a more comprehensive understanding of the model complexity, we also demonstrate the comparison of PSNR *vs.* Mult-Adds on B100 ($\times 2$) dataset in Fig.4 (b). Our MRAN obtains the same PSNR values against IMDN, which has approximately 1/3 fewer Mult-Adds than IMDN. These results indicate that the proposed MRAN makes a better trade-off between performance and model complexity than other SR models. The visual comparison for $\times 4$ SR on B100 and Urban100 are shown in Fig.5. We can see that our method can recover the correct texture well than other methods.

5. CONCLUSIONS

In this paper, we propose the residual attention search block which has the potential of multiple connections of the predefined operations and attentions. Based on the search of these blocks, we obtain the modified residual attention block which is adopted to build a lightweight SR network. Moreover, we propose a cross-scale refinement module to enhance multi-scale features while reducing the number of parameters in the reconstruction process. Extensive experiments demonstrate that our MRAN can achieve comparable or better performance than the existing methods.

6. REFERENCES

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

[2] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu, "Residual dense network for image super-resolution," in *CVPR*, 2018.

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

[4] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu, "Residual non-local attention networks for image restoration," in *ICLR*, 2019.

[5] Jie Liu, Wenjie Zhang, Yuting Tang, Jie Tang, and Gangshan Wu, "Residual feature aggregation network for image super-resolution," in *CVPR*, 2020.

[6] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *ECCV*, 2018.

[7] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang, "Lightweight image super-resolution with information multi-distillation network," in *ACM MM*, 2019.

[8] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon, "CBAM: convolutional block attention module," in *ECCV*, 2018.

[9] Bodong Li and Xieping Gao, "Lattice structure for regular linear phase paraunitary filter bank with odd decimation factor," *IEEE Signal Processing Letters*, 2013.

[10] Hanxiao Liu, Karen Simonyan, and Yiming Yang, "DARTS: differentiable architecture search," in *ICLR*, 2019.

[11] Yuhui Xu, Lingxi Xie, Xiaopeng Zhang, Xin Chen, Guo-Jun Qi, Qi Tian, and Hongkai Xiong, "PC-DARTS: partial channel connections for memory-efficient architecture search," in *ICLR*, 2020.

[12] Xiangxiang Chu, Bo Zhang, Hailong Ma, Ruijun Xu, Jixiang Li, and Qingyuan Li, "Fast, accurate and lightweight super-resolution with neural architecture search," *arXiv preprint arXiv:1901.07261*, 2019.

[13] Dehua Song, Chang Xu, Xu Jia, Yiyi Chen, Chunjing Xu, and Yunhe Wang, "Efficient residual dense block search for image super-resolution.," in *AAAI*, 2020.

[14] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," in *CVPR*, 2018.

[15] Tian YingJie, Wang YiQi, Yang LinRui, and Qi ZhiQuan, "Concatenated attention neural network for image restoration," *arXiv preprint arXiv:2006.11162*, 2020.

[16] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *CVPR*, 2016.

[17] Jie Liu, Jie Tang, and Gangshan Wu, "Residual feature distillation network for lightweight image super-resolution," *arXiv preprint arXiv:2009.11551*, 2020.

[18] Chunwei Tian, Ruibin Zhuge, Zhihao Wu, Yong Xu, Wangmeng Zuo, Chen Chen, and Chia-Wen Lin, "Lightweight image super-resolution with enhanced cnn," *arXiv preprint arXiv:2007.04344*, 2020.

[19] Abdul Muqeet, Jiwon Hwang, Subin Yang, JungHeum Kang, Yongwoo Kim, and Sung-Ho Bae, "Multi-attention based ultra lightweight image super-resolution," *arXiv preprint arXiv:2008.12912*, 2020.

[20] Hengyuan Zhao, Xiangtao Kong, Jingwen He, Yu Qiao, and Chao Dong, "Efficient image super-resolution using pixel attention," in *ECCVW*, 2020.

[21] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *CVPRW*, 2017.

[22] Marco Bevilacqua, A. Roumy, Christine Guillemot, and Marie-Line Alberi-Morel, "Low-complexity single image super-resolution based on nonnegative neighbor embedding," in *BMVC*, 2012.

[23] Roman Zeyde, Michael Elad, and Matan Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*, 2012.

[24] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *TPAMI*, 2011.

[25] Jia Bin Huang, Abhishek Singh, and Narendra Ahuja, "Single image super-resolution from transformed self-exemplars," in *CVPR*, 2015.

[26] Chao Dong, Chen Change Loy, and Xiaoou Tang, "Accelerating the super-resolfsrcnnution convolutional neural network," in *ECCV*, 2016.

[27] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *CVPR*, 2016.

[28] Ying Tai, Jian Yang, and Xiaoming Liu, "Image super-resolution via deep recursive residual network," in *CVPR*, 2017.

[29] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: A persistent memory network for image restoration," in *ICCV*, 2017.

[30] Zheng Hui, Xiumei Wang, and Xinbo Gao, "Fast and accurate single image super-resolution via information distillation network," in *CVPR*, 2018.