RAM: Residual Attention Module for Single Image Super-Resolution

Jun-Hyuk Kim, Jun-Ho Choi, Manri Cheon and Jong-Seok Lee
School of Integrated Technology, Yonsei University, Korea
{junhyuk.kim, idearibosome, manri.cheon, jong-seok.lee}@yonsei.ac.kr

Abstract

Attention mechanisms are a design trend of deep neural networks that stands out in various computer vision tasks. Recently, some works have attempted to apply attention mechanisms to single image super-resolution (SR) tasks. However, they apply the mechanisms to SR in the same or similar ways used for high-level computer vision problems without much consideration of the different nature between SR and other problems. In this paper, we propose a new attention method, which is composed of new channel-wise and spatial attention mechanisms optimized for SR and a new fused attention to combine them. Based on this, we propose a new residual attention module (RAM) and a SR network using RAM (SRRAM). We provide in-depth experimental analysis of different attention mechanisms in SR. It is shown that the proposed method can construct both deep and lightweight SR networks showing improved performance in comparison to existing state-of-the-art methods.

1. Introduction

Single image super-resolution (SR), the process of inferring a high-resolution (HR) image from a single low-resolution (LR) image. It is one of the computer vision problem progressing rapidly with the development of deep learning. Recently, convolutional neural network (CNN)-based SR methods [4, 13, 14, 23, 17, 25, 16, 19, 26, 29, 12, 5, 6, 34, 18, 1, 33] have shown better performance compared with previous hand-crafted methods [3, 24, 31, 27, 11].

Stacking an extensive amount of layers is a common practice to improve performance of deep networks [22]. After Kim et al. [13] first apply residual learning in their very deep CNN for SR (VDSR), this trend goes on for SR as well. Ledig et al. [17] propose a deeper network (SRResNet) than VDSR based on the ResNet architecture. Lim et al. [19] modify SRResNet and propose two very large networks having superior performance: deeper one and wider one, i.e., enhanced deep ResNet for SR (EDSR) and multi-scale deep SR (MDSR), respectively. In addition, there have been approaches adopting DenseNet [10] for SR, e.g., [29, 34].

While a huge size of SR networks tends to yield improved performance, this also has a limitation. Typically, most CNN-based methods internally treat all types of information equally, which may not effectively distinguish the detailed characteristics of the content (e.g., low and high frequency information). In other words, the networks have limited ability to selectively use informative features.

Recently, the attention mechanism is one of the notable network structures in various computer vision problems [8, 30]. It allows the network to recalibrate the extracted feature maps, so that more adaptive and efficient training is possible. A few recent SR methods also employ attention mechanisms. Zhang et al. [33] simply adopt an existing channel attention mechanism used for a high-level vision problem [8] without modification. Hu et al. [9] also use the same channel attention mechanism and also a spatial attention mechanism that is similar to that used in high-level vision problems [30]. It should be noted that the attention mechanisms applied to SR in these works are borrowed from other vision problems such as classification, and thus they may not be optimal for SR. Furthermore, how to combine the channel and spatial attention mechanisms effectively also remains unresolved.

To tackle these issues, in this paper, we propose a new attention-based SR method that effectively integrates two new attention mechanisms, i.e., channel attention (CA) and spatial attention (SA). These mechanisms, which are optimized for SR, are attached to a ResNet-based structure, resulting in our proposed residual attention module (RAM) and consequently our proposed SR using RAM model (SRRAM). The proposed RAM exploits both inter- and intra-channel relationship by using the proposed CA and SA, respectively. We demonstrate both the effectiveness and efficiency of our proposed method via thorough analysis of the proposed attention mechanism and fair comparison with the state-of-the-art SR methods.

In summary, our main contributions are as follows:

- Through careful analysis, we propose two attention mechanisms (CA and SA) optimized for SR.

- We combine the two mechanisms and propose a residual attention module (RAM) based on ResNet.

- We show that our approach can build successfully both lightweight networks (aiming at efficiency) and heavy (very deep) ones (aiming at high image quality), whereas existing methods focus only on one direction. This also enables us to make fair comparison with the state-of-the-art SR methods to demonstrate the effectiveness and efficiency of our method.

2. Related works

In this section, we describe in detail the existing attention mechanisms applied to ResNet-based networks in the previous works [33, 30, 9]: residual channel attention block (RCAB) [33], convolutional block attention module (CBAM) [30], and channel-wise and spatial attention residual (CSAR) block [9]. Targeting SR tasks in [33, 9], the RCAB and CSAR block are residual blocks where one or more attention mechanisms are applied. CBAM is the attention module that can be used in combination with a residual block, which is used for classification and detection tasks in [30].

For mathematical formulation, we denote the input and output feature maps of an attention mechanism as $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$ and $\hat{\mathbf{X}} \in \mathbb{R}^{H\times W\times C}$, where H, W, and C are the height, width, and number of channels of $\mathbf{X}$. We also denote the sigmoid and ReLU functions as $\sigma(\cdot)$ and $\delta(\cdot)$. For simplicity, bias terms are omitted.

The attention mechanisms can be divided into two types depending on the dimension to which they are applied: channel attention (CA) and spatial attention (SA). CA and SA can be further divided into three processes:

- *squeeze*: it is a process to extract one or more statistics $\mathbf{S}$ by the channel (CA) or spatial region (SA) from $\mathbf{X}$. The statistics are extracted by using pooling methods, and 1×1 convolution can be used for SA.

- *excitation*: using the extracted statistics, the *excitation* process captures the interrelationship between channels (CA) or spatial regions (SA) and generates an attention map $\mathbf{M}$, having a size of $1 \times 1 \times C$ (CA) or $H \times W \times 1$ (SA). Two fully connected (FC) layers are used for CA in all methods, which has a bottleneck structure with a reduction ratio of r. For SA, one or two convolutions are used.

- *scaling*: to recalibrate the input feature maps, the generated attention map is normalized through a sigmoid function between a range from 0 to 1, and then used for channel or spatial-wise multiplication with $\mathbf{X}$. The same *scaling* process is applied to all methods.

The attention mechanism of each study is illustrated in Figure 1. In addition, we formulate each method mathematically in Table 1, which is explained below in detail.

2.1. RCAB

The attention mechanism used in RCAB [33] is equal to that of the squeeze-and-excitation (SE) block [8], which is used for classification and detection tasks. The mechanism aims to recalibrate filter responses by exploiting inter-channel correlation, i.e., CA. The average pooling is applied in the *squeeze* process and the *excitation* process is performed as follows:

$$\mathbf{M} = f_{ex}(\mathbf{S}_{avg}) = fc_2(fc_1(\mathbf{S}_{avg})) = W_2\delta(W_1\mathbf{S}_{avg}), \tag{1}$$

where $\mathbf{S}_{avg} = f_{sq}(\mathbf{X}) \in \mathbb{R}^{1\times 1\times C}$, $fc_1(\cdot)$ and $fc_2(\cdot)$ are FC layers, and $W_1 \in \mathbb{R}^{\frac{C}{r}\times C}$ and $W_2 \in \mathbb{R}^{C\times\frac{C}{r}}$ are the parameters of the FC layers.

2.2. CBAM

While the attention mechanism in RCAB uses only inter-channel relation for refining feature maps, CBAM [30] exploits both inter-channel and inter-spatial relationship of feature maps through its CA and SA modules. In the CA module, the difference from that of RCAB is that max pooling is additionally performed in the *squeeze* process, and the two kinds of statistics are used for the *excitation* process. For the SA module, the results of the average and max pooling are also applied in the *squeeze* process, resulting in generating two 2D statistics. They are concatenated and undergo the *excitation* process using one 7×7 convolution. To combine the two attention mechanisms, CBAM sequentially performs CA and then SA.

2.3. CSAR block

Similarly to CBAM, the CSAR block [9] includes both CA and SA. The former is equal to that of RCAB. For SA, in contrast to CBAM, the input feature map proceeds to the *excitation* process without going through the *squeeze* process. The *excitation* process employs two 1×1 convolutions, where the first one has $C \times \gamma$ filters and the second one has a single filter. Here, γ is the increase ratio. While CBAM combines the two attention mechanisms sequentially, the CSAR block combines them in a parallel manner using concatenation and 1×1 convolution.

3. Proposed methods

3.1. Network architecture

The overall architecture of our SRRAM, which is inspired by EDSR [19], is illustrated in Figure 2. It can be

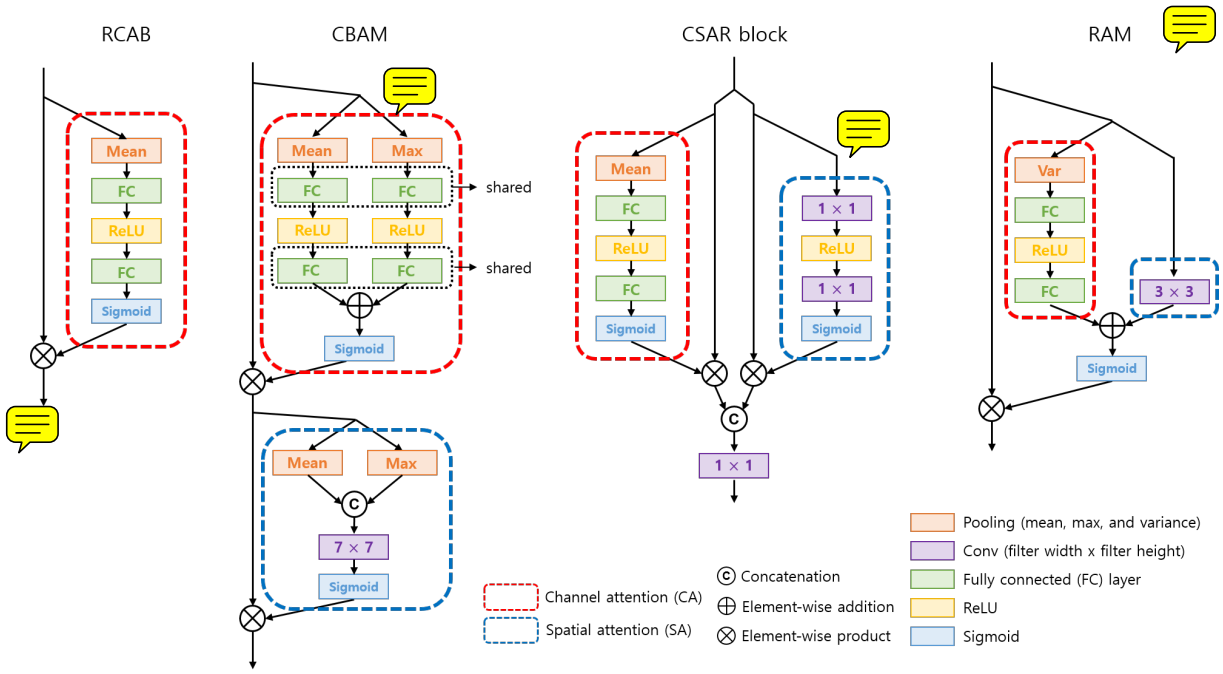


Figure 1: Structures of various attention mechanisms.

	methods	*squeeze*: $f_{sq}(\cdot)$	*excitation*: $f_{ex}(\cdot)$		*scaling*: $f_{sc}(\cdot)$
Channel attention: $f^{CA}(\cdot)$	RCAB [33]	$\mathbf{S}_{avg} = pool_{avg}(\mathbf{X})$	$\mathbf{M} = fc_2(fc_1(\mathbf{S}_{avg}))$		$\hat{\mathbf{X}} = \sigma(\mathbf{M}) \otimes \mathbf{X}$
	CBAM [30]	$\mathbf{S}_{avg} = pool_{avg}(\mathbf{X})$	$\mathbf{M}_1 = fc_2(fc_1(\mathbf{S}_{avg}))$	$\mathbf{M} = \mathbf{M}_1 + \mathbf{M}_2$	
		$\mathbf{S}_{max} = pool_{max}(\mathbf{X})$	$\mathbf{M}_2 = fc_2(fc_1(\mathbf{S}_{max}))$		
	CSAR [9]	$\mathbf{S}_{avg} = pool_{avg}(\mathbf{X})$	$\mathbf{M} = fc_2(fc_1(\mathbf{S}_{avg}))$		
	RAM	$\mathbf{S}_{var} = pool_{var}(\mathbf{X})$	$\mathbf{M} = fc_2(fc_1(\mathbf{S}_{var}))$		
Spatial attention: $f^{SA}(\cdot)$	RCAB [33]	-	-		-
	CBAM [30]	$\mathbf{S}_{avg} = pool_{avg}(\mathbf{X})$	$\mathbf{M} = conv_{7\times7}([\mathbf{S}_{avg}, \mathbf{S}_{max}])$		$\hat{\mathbf{X}} = \sigma(\mathbf{M}) \otimes \mathbf{X}$
		$\mathbf{S}_{max} = pool_{max}(\mathbf{X})$			
	CSAR [9]	-	$\mathbf{M} = conv_{1\times1}(conv_{1\times1}(\mathbf{X}))$		
	RAM	-	$\mathbf{M} = conv_{3\times3,depth}(\mathbf{X})$		
Fused attention: $f^{FA}(\cdot)$	RCAB [33]	-			
	CBAM [30]	$\hat{\mathbf{X}} = f^{SA}(f^{CA}(\mathbf{X}))$			
	CSAR [9]	$\hat{\mathbf{X}} = conv_{1\times1}([f^{CA}(\mathbf{X}), f^{SA}(\mathbf{X})])$			
	RAM	$\hat{\mathbf{X}} = f_{sc}(f_{ex}^{CA}(f_{sq}^{CA}(\mathbf{X})) \oplus f_{ex}^{SA}(\mathbf{X}))$			

Table 1: Mathematical formulations of various attention mechanisms. $\mathbf{X}$: intput feature maps. $\hat{\mathbf{X}}$: output feature maps. $pool(\cdot)$: pooling layer. $fc(\cdot)$: fully connected layer. $conv(\cdot)$: convolutional layer. $\sigma(\cdot)$: sigmoid activation. $\oplus$: element-wise addition. $\otimes$: element-wise product.

divided into two parts: 1) feature extraction part, and 2) upscaling part. Let $\mathbf{I}^{LR}$ and $\mathbf{I}^{HR}$ denote the input LR image and the corresponding output HR image, respectively. At the beginning, one convolution layer is applied to I^{LR} to extract initial feature maps, i.e.,

$$\mathbf{F}_0 = f_0(\mathbf{I}^{LR}), \tag{2}$$

where $f_0(\cdot)$ denotes the first convolution and F_0 means the extracted feature maps to be fed into the first residual attention module (RAM), which is described in detail in Section 3.2. $\mathbf{F}_0$ is updated through R RAMs and one convolution, and then the updated feature maps are added to $\mathbf{F}_0$ by using global skip connection:

$$\mathbf{F}_f = \mathbf{F}_0 + f_f(R_R(R_{R-1}(...R_1(\mathbf{F}_0)...))), \tag{3}$$

where $R_r(\cdot)$ denotes the r-th RAM and $f_f(\cdot)$ and $\mathbf{F}_f$ are the last convolution and feature maps of the feature extraction part, respectively.

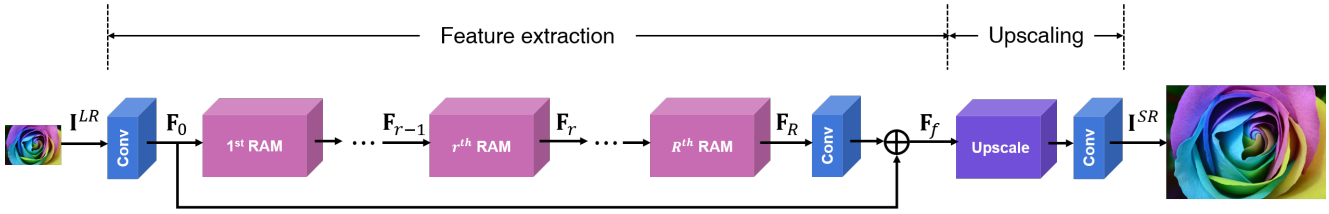


Figure 2: Overall architecture of our proposed network.

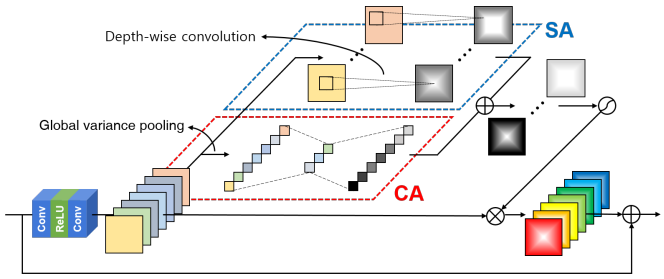


Figure 3: Residual attention module (RAM).

For the upscaling part, we use the sub-pixel convolution layers [23], which is followed by one convolution for reconstruction:

$$\mathbf{I}^{SR} = f_{recon}(f_{up}(\mathbf{F}_f)), \tag{4}$$

where $f_{up}(\cdot)$ and $f_{recon}(\cdot)$ are the functions for upscaling and reconstruction, respectively, and $\mathbf{I}^{SR}$ is the output super-resolved image.

3.2. Residual attention module

The structure of RAM is illustrated in Figure 3. First, we present new CA and SA mechanisms, respectively, and additionally propose a way to fuse the two mechanisms, which is termed fused attention (FA) mechanism. Then, we propose residual attention module (RAM) by integrating the FA mechanism after the last convolution of the residual block.

Let $\mathbf{F}_{r-1}$ and $\mathbf{F}_r$ be the input and output feature maps of the r-th RAM. Then, $\mathbf{F}_r$ can be formulated as

$$\mathbf{F}_r = R_r(\mathbf{F}_{r-1}) = \mathbf{F}_{r-1} + f^{FA}(f_{trans}(\mathbf{F}_{r-1})), \tag{5}$$

where $f_{trans}(\cdot)$ denotes the function consisting of convolution, ReLU, and convolution, and $f^{FA}(\cdot)$ means our proposed FA mechanism.

Channel attention (CA). The *squeeze* process of the CA mechanism in the previous works [33, 9] simply adopts the popular average pooling method used in high-level computer vision problems such as image classification and object detection without modification. However, since SR ultimately aims at restoring high-frequency components of images, it is more reasonable for attention maps to be determined using high-frequency statistics about the channels. To this end, we choose to use the variance rather than the average for the pooling method, i.e.,

$$\mathbf{S}_{r-1} = pool_{var}(\mathbf{X}_{r-1}), \tag{6}$$

where $pool_{var}(\cdot)$ denotes variance pooling, $\mathbf{S}_{r-1} \in \mathbb{R}^{1\times1\times C}$ is the output statistic, and $\mathbf{X}_{r-1} = f_{trans}(\mathbf{F}_{r-1})$. The *excitation* and *scaling* processes are performed in the same way as in [33].

Spatial attention (SA). Each channel in the feature maps $\mathbf{X}_{r-1}$ has a different meaning depending on the role of the filter used. For example, some filters will extract the edge components in the horizontal direction, and some filters will extract the edge components in the vertical direction. From the viewpoint of SR, the importance of the channels varies by the spatial region. For example, in the case of edges or complex textures, more detailed information, i.e., those from complex filters, is more important. On the other hand, in the case of the region having almost no high-frequency components such as sky or homogeneous areas of comic images, relatively less detailed information is more important and needs to be attended. In this regard, the SA map for each channel needs to be different. Therefore, unlike CBAM [30], which performs the *squeeze* process in its SA module, our proposed method does not squeeze information per channel to preserve channel-specific characteristics. In addition, for the *excitation* process, in contrast to other SA mechanisms [30, 9] generating a single 2D SA map, we obtain different SA maps for each channel $\mathbf{M}^{SA} \in \mathbb{R}^{H\times W\times C}$ using depth-wise convolution [7], i.e.,

$$\mathbf{M}^{SA}_{r-1} = conv_{3\times3,depth}(\mathbf{X}_{r-1}), \tag{7}$$

where $conv_{3\times3,depth}(\cdot)$ denotes the 3×3 depth-wise convolution.

Fused attention (FA). The proposed CA and SA mechanisms exploit information from inter-channel and intra-channel relationship, respectively. Therefore, in order to exploit the benefits of both mechanisms simultaneously, we combine them by adding the CA and SA maps and then perform the *scaling* process is performed using the sigmoid function, whose result is used for recalibrating the feature map $\mathbf{X}_{r-1}$:

$$\hat{\mathbf{X}}_{r-1} = f^{FA}(\mathbf{X}_{r-1}) = \sigma(\mathbf{M}^{CA}_{r-1} \oplus \mathbf{M}^{SA}_{r-1}) \otimes \mathbf{X}_{r-1}. \tag{8}$$

where, $\oplus$ and $\times$ denote element-wise addition and product, respectively. The analysis of each proposed module is covered in detail in Section 4.3.

4. Experiments

4.1. Datasets and metrics

In the experiments, we train all our models using the training images from the DIV2K dataset [28]. It contains 800 RGB HR training images and their corresponding LR training images for three downscaling factors ($\times 2$, $\times 3$, and $\times 4$). For evaluation, we use five datasets commonly used in SR benchmarks: Set5 [2], Set14 [32], BSD100 [20], Urban100 [11], Manga109 [21]. The Set5, Set14, and BSD100 datasets consist of natural images. The Urban100 dataset includes images related to building structures with complex and repetitive patterns., which are challenging for SR. The Manga109 dataset consists of images taken from Japanese manga, which are computer-generated images and have different characteristics from natural ones.

To evaluate SR performance, we calculate peak signal-to-noise ration (PSNR) and structural similarity (SSIM) index on the Y channel after converting to YCbCr channels.

4.2. Implementation Details

To construct an input mini-batch for training, we randomly crop a 48×48 patch from each of the randomly selected 16 LR training images. For data augmentation, the patches are randomly horizontal flipped and rotated ($90°$, $180°$, and $270°$). Before feeding the mini-batch into our networks, we subtract the average value of the entire training images for each RGB channel of the patches.

We set the size and number of filters as 3×3 and 64 in all convolution layers except those for the upscaling part. All our networks are optimized using the Adam optimizer [15] to minimize the L1 loss function, where the parameters of the optimizer are set as $\beta 1 = 0.9$, $\beta 2 = 0.999$, and $\epsilon = 10^{-8}$. The learning rate is initially set to 10^{-4}, which decreases by a half at every 2×10^5 iterations. We implement our networks using the Tensorflow framework with NVIDIA GeForce GTX 1080 GPU[1].

4.3. Model analysis

Ablation study. The first four columns of Table 2 show the ablation study of our proposed attention mechanisms. For fair comparison, we set up four networks having the same numbers of filters (64) and residual blocks (16). Employing each mechanism increases the number of model parameters, and as the network becomes deeper, the number of such additional parameters becomes large. We want to minimize the possibility to obtain improved performance simply due to such an increased number of parameters and observe the effect of the mechanisms well. Therefore, we configure the networks to be relatively shallow. We experiment with the easiest case, i.e., $\times 2$ SR, to enable sufficient convergence even for shallow networks. In addition, the comparison is done on the five datasets with different characteristics to check the generalization capability of the four networks.

First, without the CA and SA mechanisms, the network exhibits the worst performance, which implies that channel-wise and spatial information is not effectively exploited without the mechanisms. Then, we add the proposed CA and/or SA mechanisms to the baseline. The CA mechanism leads to performance improvement (+0.1 dB on average) only by adding 9K parameters (which is an increase of 0.6%). This improvement is more prominent for Urban100 and Manga109. Since Urban100 contains challenging images and the images in Manga109 have unique characteristics of computer-generated images, which are quite different from those in the training images, it can be interpreted that the network generalizes well for diverse images. The SA mechanism also yields the performance similar to or slightly better (+0.05 dB for Set5) than the baseline. We finally add the proposed FA, a fused mechanism of CA and SA, to the baseline, which further improves performance. This shows that the FA mechanism not only considers the relationship between channels, but also effectively exploits spatial information in each channel. Note that the performance improvement is achieved only with additional 19K parameters (which is an increase of 1.4%).

Effectiveness and efficiency of RAM. To get a closer look at effectiveness and efficiency of our method, we compare it with the other attention mechanisms [33, 30, 9] illustrated in Section 2. For fairness, we construct networks in the form of implementing the attention mechanisms in each residual block of the baseline network.

The right part of Table 2 shows their $\times 2$ SR performance. Overall, all the cases show lower performance than our method and thus we can confirm the efficiency of the proposed method, especially by considering that the second best network (the last column of Table 2) has more parameters than our network (+257K). By comparing performance of the different CAs (the second, fifth, sixth, and ninth columns of Table 2), we can examine which pooling methods of the *squeeze* process is suitable for SR. It can be observed that our method extracting channel-specific statistics with variance produces the best performance for challenging images, i.e., Urban100 and Manga109. Comparing the SA mechanisms through the third, seventh, tenth columns of Table 2, we can see that their performance is lower than that of the baseline, except for ours. This shows that preserving the channel-specific information through depth-wise convolution is an effective way of dealing with spatial

[1] Our code is made publicly available at http://[anonymous submission].

	Baseline	RAM			RCAB [33]	CBAM [30]			CSAR block [9]		
CA	×	✓	×	✓	✓	✓	×	✓	✓	×	✓
SA	×	×	✓	✓	×	×	✓	✓	×	✓	✓
# params.	1370K	1379K	1380K	1389K	1379K	1379K	1371K	1381K	1379K	1505K	1646K
Set5	37.90	37.93	37.95	37.98	37.96	37.91	37.84	37.89	37.96	37.91	37.96
Set14	33.58	33.55	33.59	33.57	33.58	33.51	33.52	33.45	33.58	33.56	33.57
BSD100	32.17	32.17	32.17	32.17	32.17	32.14	32.12	32.11	32.17	32.14	32.16
Urban100	32.13	32.26	32.13	32.28	32.24	32.14	31.93	32.01	32.24	32.02	32.29
Manga109	38.47	38.67	38.46	38.72	38.60	38.19	38.31	38.20	38.60	38.33	38.62
All	34.40	34.50	34.40	34.53	34.48	34.30	34.27	34.25	34.48	34.31	34.50

Table 2: Performance of $\times 2$ SR by our proposed RAM and existing methods in terms of PSNR(dB). Note that the CSAR block only with CA is equivalent to RCAB. Red and blue colors indicate the best and second best performance, respectively.

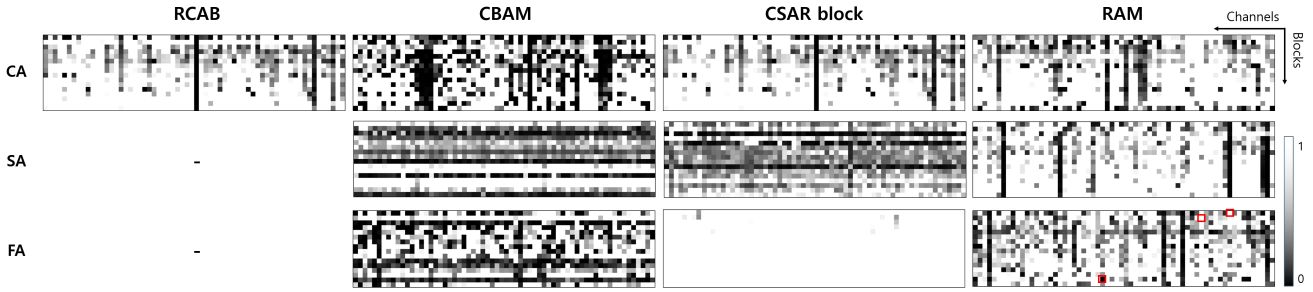


Figure 4: Variances calculated from each channel in each attention mechanism. The x and y axes represent different channels and blocks, respectively.

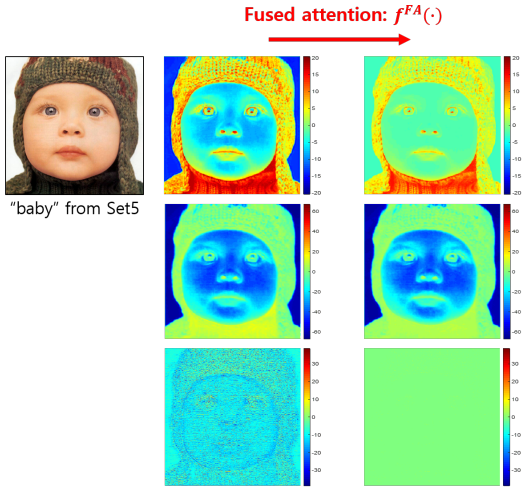
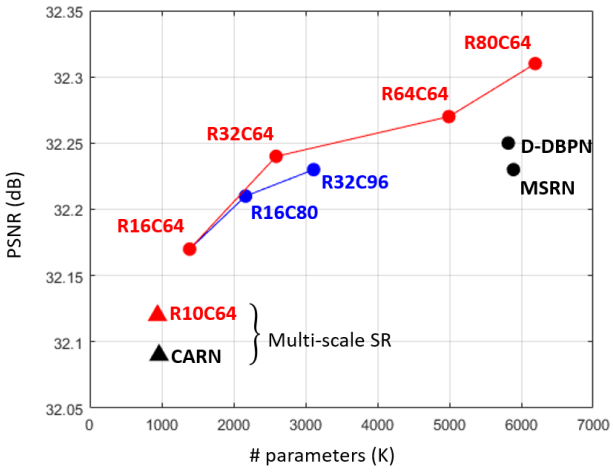


Figure 5: Example feature maps before and after our FA mechanism.

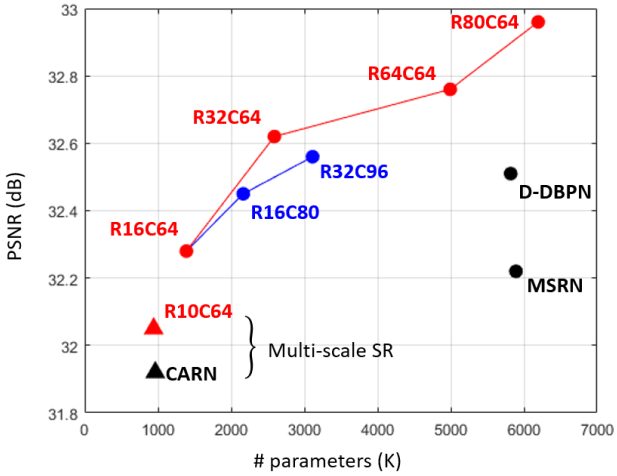
attention while maintaining the performance of the baseline. Lastly, using two mechanisms in CBAM is less effective for SR than using its CA alone. The performance of the CSAR block improves by 0.02 dB when its SA mechanism is additionally used, but at the cost of an increased number of parameters by 19.4%. Our FA mechanism is much more efficient than that, yielding performance improvement by 0.03 dB using only 0.7% more parameters than CA.

We further analyze the role of attention mechanisms by observing statistics of intermediate feature maps. To this end, we simply obtain the variance over the spatial axes of each intermediate feature map of the residual blocks. The networks consist of 16 blocks, each having 64 feature maps, so 16×64 variances are obtained, which are shown in Figure 4. We have three observations: 1) The CA mechanisms showing higher performance have higher variances (i.e., more white pixels appear.) 2) The other SA methods show almost black horizontal lines in some blocks, indicating the corresponding blocks only suppress the activations of all channels, which is not observed in our SA mechanism. 3) Our FA mechanism, the best performing model, shows clear differences across different areas, showing that the roles are well divided among the filters.

We further observe the changes in feature maps before and after the proposed FA mechanism to analyze its role. Figure 5 shows an example of three different characteristics, which are the feature maps corresponding to the three red boxes shown in the bottom left panel of Figure 4. In (5), the feature maps obtained by our FA mechanism are "added" to the input feature maps, which means that no update oc-



(a) BSD100



(b) Urban100

Figure 6: PSNR (dB) vs. the number of parameters for $\times 2$ SR. Those marked with black color indicate existing methods, and those marked with red or blue color are our models with varying the number of RAMs (R) or the number of channels (C), respectively. The networks trained for single-scale SR are marked with $\circ$, and those trained for multi-scale SR are marked with $\triangle$.

curs in the pixel where the value of the feature map is zero. In the first row, the feature map is activated in both positive (red color) and negative (blue color) directions, which correspond to the low-frequency components (white background) and the high-frequency components (texture of the woolen hat), respectively, and our FA mechanism kills low-frequency component values to zero. It can be interpreted that the feature map is recalibrated so that it can pay "attention" to the high-frequency components. In this second row, the feature map before FA focuses on low-frequency components and it is not changed after the FA mechanism. In the bottom row, the feature map before FA seems to have almost no active information and no clear meaning. In this case, the FA mechanism kills the activated values.

Study of R and C. The structure of our SRRAM is determined by the number of RAMs (R) and the number of channels (C) used in each RAM. In this experiment, we examine the effect of these two variables on performance. Starting from the case with $R = 16$ and $C = 64$ ($R16C64$), we increase R or C, whose results are shown as the red and blue lines in Figure 6, respectively. A larger value of R or C leads to performance improvement and it appears that increasing R is more effective for the same number of additional parameters.

4.4. Comparison with state-of-the-art methods

We finally evaluate our proposed SRRAM by comparing with the state-of-the art SR methods, including VDSR [13], LapSRN [16], DRRN [25], MemNet [26], DSRN [5], IDN [12], CARN [1], MSRN [18], and D-DBPN [6]. MSRN and D-DBPN are selected to verify the effectiveness of RAM on large networks, and the remainder corresponds to the opposite case. We select SRRAM with $R = 64$ and $C = 64$ (SRRAM_$R64C64$) and that with $R = 10$ and $C = 64$ (SRRAM_$R10C64$) as our final large and lightweight networks, respectively. For large networks, only $\times 2$ SR is considered for meaningful comparison of feature extraction in different SR methods, since MSRN uses different upscaling methods using more parameters for the other scaling factors. In the cases of CARN and SRRAM_$R10C64$, 64×64 patches and multi-scale ($\times 2$, $\times 3$, and $\times 4$) SR are used for training. The results are summarized in Table 3, Table 4, and Figure 6.

In Table 3, SRRAM shows the highest PSNR values for all datasets, and the performance gap with the other methods widens further in Urban100 and Manga109. Note that our model has only about 85% and 86% of parameters of MSRN and D-DBPN, respectively. We also provide the visual results of challenging images in Figure 7, where only our model successfully restores complex patterns.

In Table 4, our model yields similar or better performance with fewer parameters than CARN, which demonstrates the efficiency of ours, considering that CARN is the best performing model in terms of efficiency.

5. Conclusion

We have proposed an attention-based SR network based on new attention methods (CA and SA). The two mechanisms are integrated in our FA method, based on which RAM was proposed. The experimental results demonstrated that our attention methods provide improved attention mechanisms for SR and, consequently, the proposed SRRAM model can achieve improved SR performance as both large and lightweight networks.

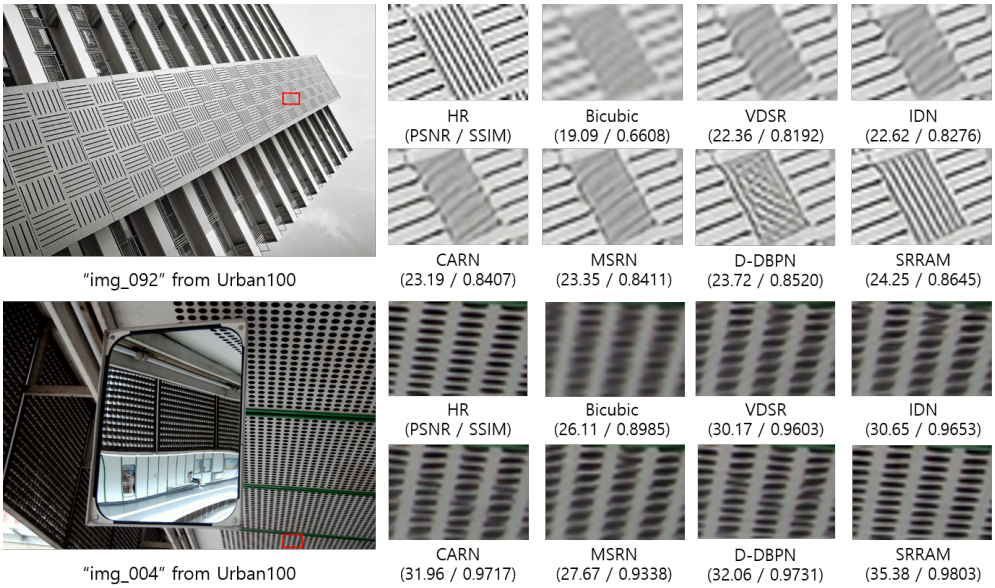


Figure 7: Comparison of $\times 2$ SR results on challenging images of Urban100 [11].

Method	# params.	Set5	Set14	BSD100	Urban100	Manga109
		PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM
MSRN [18]	5891K	38.08 / 0.9605	33.74 / 0.9170	32.23 / 0.9013	32.22 / 0.9326	38.82 / 0.9868
D-DBPN [6]	5819K	38.05 / 0.960	33.79 / 0.919	32.25 / 0.900	32.51 / 0.932	38.81 / 0.976
SRRAM	4993K	38.10 / 0.9601	33.90 / 0.9195	32.27 / 0.9003	32.76 / 0.9330	39.10 / 0.9769

Table 3: Quantitative evaluation results of large SR models for a scaling factor of 2. Red and blue colors indicate the best and second best performance, respectively.

Scale	Method	# params.	Set5	Set14	BSD100	Urban100
			PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM
$\times 2$	VDSR [13]	666K	37.53 / 0.9587	33.03 / 0.9124	31.90 / 0.8960	30.76 / 0.9140
	LapSRN [16]	407K	37.52 / 0.959	33.08 / 0.913	31.80 / 0.895	30.41 / 0.910
	DRRN [25]	298K	37.74 / 0.9591	33.23 / 0.9136	32.05 / 0.8973	31.23 / 0.9188
	MemNet [26]	686K	37.78 / 0.9597	33.23 / 0.9142	32.08 / 0.8978	31.31 / 0.9195
	DSRN [5]	>1,000K	37.66 / 0.959	33.15 / 0.913	32.10 / 0.897	30.97 / 0.916
	IDN [12]	553K	37.83 / 0.9600	33.30 / 0.9148	32.08 / 0.8985	31.27 / 0.9196
	CARN [1]	964K	37.76 / 0.9590	33.52 / 0.9166	32.09 / 0.8978	31.92 / 0.9256
	SRRAM	942K	37.82 / 0.9592	33.48 / 0.9171	32.12 / 0.8983	32.05 / 0.9264
$\times 3$	VDSR [13]	666K	33.66 / 0.9213	29.77 / 0.8314	28.82 / 0.7976	27.14 / 0.8279
	DRRN [25]	298K	34.03 / 0.9244	29.96 / 0.8349	28.95 / 0.8004	27.53 / 0.8378
	MemNet [26]	686K	34.09 / 0.9248	30.00 / 0.8350	28.96 / 0.8001	27.56 / 0.8376
	DSRN [5]	>1,000K	33.88 / 0.922	30.26 / 0.837	28.81 / 0.797	27.16 / 0.828
	IDN [12]	553K	34.11 / 0.9253	29.99 / 0.8354	28.95 / 0.8013	27.42 / 0.8359
	CARN [1]	1,149K	34.29 / 0.9255	30.29 / 0.8407	29.06 / 0.8034	28.06 / 0.8493
	SRRAM	1,127K	34.30 / 0.9256	30.32 / 0.8417	29.07 / 0.8039	28.12 / 0.8507
$\times 4$	VDSR [13]	666K	31.35 / 0.8838	28.01 / 0.7674	27.29 / 0.7251	25.18 / 0.7524
	LapSRN [16]	814K	31.54 / 0.885	28.19 / 0.772	27.32 / 0.728	25.21 / 0.756
	DRRN [25]	298K	31.68 / 0.8888	28.21 / 0.7720	27.38 / 0.7284	25.44 / 0.7638
	MemNet [26]	686K	31.74 / 0.8893	28.26 / 0.7723	27.40 / 0.7281	25.50 / 0.7630
	SRDenseNet [29]	2,015K	32.02 / 0.8934	28.50 / 0.7782	27.53 / 0.7337	26.05 / 0.7819
	DSRN [5]	>1,000K	31.40 / 0.883	28.07 / 0.770	27.25 / 0.724	25.08 / 0.717
	IDN [12]	553K	31.82 / 0.8903	28.25 / 0.7730	27.41 / 0.7297	25.41 / 0.7632
	CARN [1]	1,112K	32.13 / 0.8937	28.60 / 0.7806	27.58 / 0.7349	26.07 / 0.7837
	SRRAM	1,090K	32.13 / 0.8932	28.54 / 0.7800	27.56 / 0.7350	26.05 / 0.7834

Table 4: Quantitative evaluation results of lightweight SR models. Red and blue colors indicate the best and second best performance, respectively.

References

[1] N. Ahn, B. Kang, and K.-A. Sohn. Fast, accurate, and, lightweight super-resolution with cascading residual network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 1, 7, 8

[2] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2012. 5

[3] H. Chang, D.-Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. 1

[4] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. 1

[5] W. Han, S. Chang, D. Liu, M. Yu, M. Witbrock, and T. S. Huang. Image super-resolution via dual-state recurrent networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 7, 8

[6] M. Haris, G. Shakhnarovich, and N. Ukita. Deep backprojection networks for super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 7, 8

[7] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 4

[8] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[9] Y. Hu, J. Li, Y. Huang, and X. Gao. Channel-wise and spatial feature modulation network for single image super-resolution. *arXiv preprint arXiv:1809.11130*, 2018. 1, 2, 3, 4, 5, 6

[10] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[11] J.-B. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 5, 8

[12] Z. Hui, X. Wang, and X. Gao. Fast and accurate single image super-resolution via information distillation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 7, 8

[13] J. Kim, J. Lee, and K. Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 7, 8

[14] J. Kim, J. Lee, and K. Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[15] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[16] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 7, 8

[17] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[18] J. Li, F. Fang, K. Mei, and G. Zhang. Multi-scale residual network for image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 1, 7, 8

[19] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017. 1, 2

[20] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2001. 5

[21] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76(20):21811–21838, 2017. 5

[22] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio. On the number of linear regions of deep neural networks. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2014. 1

[23] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 4

[24] J. Sun, Z. Xu, and H.-Y. Shum. Image super-resolution using gradient profile prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 1

[25] Y. Tai, J. Yang, and X. Liu. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 7, 8

[26] Y. Tai, J. Yang, X. Liu, and C. Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 7, 8

[27] R. Timofte, V. De, and L. Van Gool. Anchored neighborhood regression for fast example-based super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. 1

[28] R. Timofte, S. Gu, J. Wu, L. Van Gool, L. Zhang, M.-H. Yang, et al. Ntire 2018 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018. 5

[29] T. Tong, G. Li, X. Liu, and Q. Gao. Image super-resolution using dense skip connections. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 8

[30] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 1, 2, 3, 4, 5, 6

[31] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 19(11):2861–2873, 2010. 1

[32] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *Proceedings of the International Conference on Curves and Surfaces*, 2010. 5

[33] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 1, 2, 3, 4, 5, 6

[34] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1