Context Reasoning Attention Network for Image Super-Resolution

Yulun Zhang[1], Donglai Wei[2], Can Qin[1], Huan Wang[1,*], Hanspeter Pfister[3], Yun Fu[1]
[1]Northeastern University, [2]Boston College, [3]Harvard University

Abstract

Deep convolutional neural networks (CNNs) are achieving great successes for image super-resolution (SR), where global context is crucial for accurate restoration. However, the basic convolutional layer in CNNs is designed to extract local patterns, lacking the ability to model global context. With global context information, lots of efforts have been devoted to augmenting SR networks, especially by global feature interaction methods. These works incorporate the global context into local feature representation. However, recent advances in neuroscience show that it is necessary for the neurons to dynamically modulate their functions according to context, which is neglected in most CNN based SR methods. Motivated by those observations and analyses, we propose context reasoning attention network (CRAN) to modulate the convolution kernel according to the global context adaptively. Specifically, we extract global context descriptors, which are further enhanced with semantic reasoning. Channel and spatial interactions are then introduced to generate context reasoning attention mask, which is applied to modify the convolution kernel adaptively. Such a modulated convolution layer is utilized as basic component to build the blocks and networks. Extensive experiments on benchmark datasets with multiple degradation models show that CRAN obtains superior results and favorable trade-off between performance and model complexity.

1. Introduction

Image super-resolution (SR) aims to reconstruct an accurate high-resolution (HR) image given its low-resolution (LR) counterpart [14]. Image SR plays a fundamental role in various computer vision applications, ranging from security and surveillance imaging [71], medical imaging [48], to object recognition [45]. However, image SR is an ill-posed problem, since there exists multiple solutions for any LR input. To tackle such an inverse problem, lots of deep convolutional neural networks (CNNs) have been proposed to learn mappings between LR and HR image pairs.

Deep CNNs have achieved remarkable successes for image SR [10, 12, 26, 36, 62, 18, 66, 1, 23, 31, 67]. In

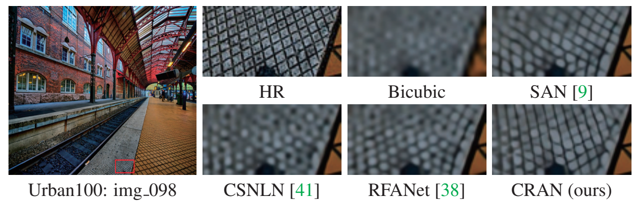


Figure 1. Visual examples for 4× SR with Bicubic (BI) degradation on Urban100 [22]. SAN, CSNLN, and RFANet recover parts of local textures. Global context guided convolution enables CRAN to recover more structural textures with proper directions.

CNNs, convolution extracts local patches by a sliding window, making it only capable of capturing local patterns. However, recent advances in neuroscience reveal that neurons' awareness of global context is essential for us to process complex perceptual tasks effectively [34, 15]. The sliding window mechanism in convolution limits its ability to utilize global context, being crucial for accurate image SR.

To alleviate this limitation, many SR methods have been recently proposed to introduce global context modeling modules into SR networks [64, 9, 38, 65, 41]. Zhang et al. proposed residual channel attention network [64], where the global context was modelled with global average pooling and used to rescale each feature channel. Dai et al. proposed second-order channel attention by considering higher order feature statistics in SAN [9]. Different from channel attention, Liu et al. proposed an enhanced spatial attention block in FRANet [38] to make the residual features be more focused on critical spatial contents.

Zhang et al. further proposed residual non-local attention network [65] to rescale hierarchical features with mixed channel and spatial attentions adaptively. Such a non-local attention mechanism was further developed in cross-scale non-local attention (CSNLN) [41]. Mei et al. proposed a self-exemplar mining cell to exhaustively mine all the possible intrinsic priors by combining local and in-scale/cross-scale non-local feature correlations in CSNLN [41]. As shown in Figure 1, SAN, RFANet, and CSNLN could recover some kind of local textures. But, it seems that the directions of those textures are not faithful to the ground truth. This is mainly because these methods mainly incorporate the global context into the local features.

However, as investigated in neuroscience [15], the func-

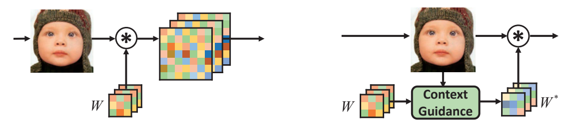
*Corresponding author: Huan Wang (wang.huan@northeastern.edu)

tion of neurons should be adaptively changed according to the behavioral context. Therefore, we can dynamically modify the convolution kernels based on context information [37]. Image SR has not witnessed works exploiting such a modulation mechanism, which was tentatively investigated in other computer vision applications. Zhu et al. proposed to adaptively set the offset of each element in a convolution kernel and the gather value for each element in the local feature patch [70]. However, such an operation only changes the input features fed into the convolutional (Conv) layer. Wu et al. proposed to generate the convolution kernel weights dynamically by taking local segments as inputs only [55]. Similar works in [24, 25] extracted features from the input image with another network and then generated convolution kernel weights. The feature extraction process could be time-consuming, making it impractical for very deep CNNs in image SR. Lin et al. proposed context-gated convolution to introduce context-awareness to Conv layers [37]. However, most of them neglected to mine the relationship among context information, which could also be important for high-quality image SR.

Motivated by the observations and analyses above, we propose a context reasoning attention network (CRAN) for image SR. This is the first attempt in image SR to modulate the convolution kernel according to the global context adaptively to the best of our knowledge (see Figure 2). Specifically, we project the input feature into latent representations and extract global context descriptors. The context relationship descriptors are further enhanced by using the descriptor relationship with semantic reasoning. Channel and spatial interactions [37] are then introduced to generate context reasoning attention mask, which is applied to modify the convolution kernel adaptively. We use the modulated convolution layer as a basic component to build blocks and the whole networks. Consequently, our CRAN can achieve much superior SR results (e.g., in Figure 1) against recent leading methods and favourable efficiency trade-off.

In summary, the main contributions of this work can be concluded in three parts:

- We propose a context reasoning attention network for accurate image SR. Our CRAN can adaptively modulate the convolution kernel according to the global context enhanced by semantic reasoning. CRAN achieves superior SR results quantitatively and visually.
- We propose to extract context information into latent representations, resulting in a bag of global context descriptors. We further enhance the descriptors by using their relationship with semantic reasoning.
- We introduce channel and spatial interactions to generate context reasoning attention mask used to modify convolution kernel. We finally obtain the context reasoning attention convolution, which further serves as a base to build blocks and networks for image SR.



(a) Previous Conv (b) Context Guided Conv

Figure 2. Conv layers. Motivated by [37], we modify Conv kernel W as W^* with context. $\circledast$ denotes Conv operation.

2. Related Works

Deep CNN for Image SR. The pioneering work was done by Dong et al. [10], who proposed SRCNN with three convolutional (Conv) layers for image SR. By introducing residual learning to ease the training difficulty, Kim et al. proposed VDSR [26] and DRCN [27] with 20 layers and achieved significant performance improvement. Lim et al. proposed EDSR [36] by simplifying residual block, which allows to build deeper and wider networks with more parameters. Zhang et al. proposed RDN [66] to reduce the model size and keep accurate performance. However, those methods neglect to utilize different importance across different feature channels and/or spatial positions. The attention mechanism was then utilized to tackle those limitations. Zhang et al. proposed residual channel attention network (RCAN) [64] by considering interdependencies among feature channels. Then, more and more works have been proposed to investigate efficient attention mechanisms for image SR. Dai et al. proposed a second-order attention network (SAN) [9] for more powerful feature expression and feature correlation learning. In those methods, the convolution kernels are not adaptive to the specific context in the inference phase, hindering the representation ability of networks. Those observations motivate us to modify convolution kernels adaptively according to the input.

Context Information in CNN. Tentative works have augmented CNNs with context information and can be briefly categorized into three types. First, similar as humans' visual processing system, backward connections were incorporated in CNNs [59, 57] to model the top-down influence [15]. But, it is still hard to understand how the feedback mechanism can perform effectively and efficiently in CNNs. Second, attention mechanism was utilized to modify intermediate feature representations in CNNs [50, 52, 54, 5]. They usually utilized the global context information (e.g., self-attention mechanism) to modify the local features [51, 52, 21, 54, 5, 2]. However, this kind of methods only consider changing the input feature maps. Third, many works attempted to dynamically changing the convolutional layer parameters by considering local or global information [24, 8, 25, 6, 55, 70, 37]. Some of them neglected to consider the Conv weight tensor [70], only took local segments and inputs [55], or suffered from expensive feature extraction process [24, 25]. Plus, most of them neglected to mine the relation among context information, which could be bone with semantic reasoning.

Semantic Reasoning. Relational reasoning is initially introduced into the artificial intelligence community as symbolic methods [42]. As an active research area, graph-based methods have been prevalent in recent years and shown to be an efficient way of relational reasoning. Inspired by the great success of CNNs in computer vision area [19], [29] proposed graph convolution networks (GCNs) for semi-supervised classification. [56] utilized GCNs to encode the prior knowledge into a deep reinforcement learning framework to improve semantic navigation in unseen scenes and novel objects. [5, 32] incorporated GCN into the design of visual encoding and learn relationship enhanced features end-to-end towards the task of interest, such as image classification and image-text matching. [58] trained a visual relationship detection model on Visual Genome dataset [30] and used a GCN-based image encoder to encode the detected relationship information.

3. Context Reasoning Attention Network for Image Super-Resolution (SR)

In image super-resolution (SR), the original input is the low-resolution (LR) image I_{LR}, which would be extracted as deep features by convolutional layers. For a convolutional layer, the input is a feature map $F_{in} \in \mathbb{R}^{c_{in} \times h \times w}$, where c_{in}, h, w are the channel number, height, and width of the feature map, respectively. To conduct the convolution operation, we slide window to extract a local feature patch of size $c_{in} \times k_1 \times k_2$. Then, we multiply the feature patch with the convolutional kernel $W \in \mathbb{R}^{c_{out} \times c_{in} \times k_1 \times k_2}$, where c_{out}, k_1, k_2 are the output channel number, height, and width of the kernel. Here, each convolutional operation only extracts local information, which would not affect the kernels adaptively in the inference phase.

3.1. Context Information Extraction

To tackle the above drawback of traditional convolution, we propose a context reasoning attention convolution (see Figure 3). We attempt to incorporate the global context information into the convolution process. On the other hand, the input LR image size can be arbitrarily large, so as the feature maps. To extract context information, we first reduce the spatial size of input feature F_{in} to $h' \times w'$ by using a pooling layer. Then, for each feature channel, we extract a latent representation of the global context by considering all the spatial positions. Specifically, we use a shared linear layer with weight $W_E \in \mathbb{R}^{h' \times w' \times e}$ to project each channel to a latent vector of size e. Following the bottleneck design in [50, 21, 52, 37], we set the vector size as $e = \frac{k_1 \times k_2}{2}$. Consequently, we obtain a new feature with global context information and denote it as $F_C \in \mathbb{R}^{c_{in} \times e}$.

Let's write the global context information as a set of vectors $F_C = [\mathbf{f}_1, \cdots, \mathbf{f}_e] \in \mathbb{R}^{c_{in} \times e}$. It gives a new perspective on the context information extraction results, which are actually a bag of global context descriptors.

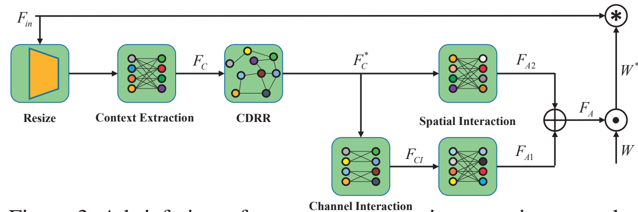


Figure 3. A brief view of our context reasoning attention convolution (CRAC). CDRR denotes context descriptor relationship reasoning. $\odot$ and $\circledast$ denote element-wise multiplication and convolution operations, respectively. Eq. (4) describes operation $\oplus$.

3.2. Context Descriptor Relationship Reasoning

We first obtain the global context descriptor set F_C. Then, the relationship among each context descriptor $\mathbf{f}_i$ enables further enhancement. Recently, visual reasoning based methods [46, 4, 68, 33, 63] have been investigated in deep learning to make better use of the relationship among visual components. Motivated by those works, we construct a relationship reasoning model among the context descriptors. Specifically, with weight parameters W_φ and W_ϕ, we embed the context descriptors into two embedding spaces. Then, the pairwise affinity can be calculated via

$$R(\mathbf{f}_i, \mathbf{f}_j) = (W_\varphi \mathbf{f}_i)^T (W_\phi \mathbf{f}_j), \tag{1}$$

which obtains the relationship between every two learned context descriptors $\mathbf{f}_i$ and $\mathbf{f}_j$, resulting in a graph.

We then denote the graph as $G(F_C, R)$, where F_C is the set of graph nodes (i.e., context descriptors) and R is the set of graph edges (i.e., context descriptor relationships). Based on Eq. (1), the affinity matrix R can be obtained by measuring the affinity edge of each context descriptor pair. For a graph edge, high affinity score indicates strong semantic relationship among the corresponding context descriptor pair. We then bridge F_C and original input with residual learning

$$F_C^* = \sigma(\left[\left(R{F_C}^T W_g\right) W_r\right]^T) \odot F_C + F_C, \tag{2}$$

where $\sigma(\cdot)$ is sigmoid activation function. R is the $e \times e$ affinity matrix. W_g is a $c_{in} \times c_{in}$ weight matrix of the GCN layer and W_r is the weight matrix of the residual structure. $\odot$ denotes element-wise multiplication.

3.3. Context Reasoning Attention Convolution

Inspired by [37], we try to update the convolution kernel with attention by adopting the enhanced global context information F_C^*. The attention mask has size of $F_A \in \mathbb{R}^{c_{out} \times c_{in} \times k_1 \times k_2}$, same as the convolution kernel weight.

Kernel Decomposition. For deep CNN based image SR methods, the feature input and output channels c_{in}, c_{out} are usually large (e.g., 128 in CSNLN [41] and 256 in EDSR [36]), which could make the kernel modulation time consuming. To break down the computation complexity, we follow the previous works about convolution kernel decomposition [20, 7, 37] and attempt to generate two tensors $F_{A1} \in \mathbb{R}^{c_{out} \times k_1 \times k_2}$ and $F_{A2} \in \mathbb{R}^{c_{in} \times k_1 \times k_2}$.

We aim to reduce the computation complexity further to adjust for a very deep network and large feature size in image SR. Motivated by the design in depth-wise separable convolutions [20, 7, 37], we turn to achieve such two tensors F_{A1}, F_{A2} and the final F_A by modelling the channel interaction and spatial interaction separately.

Channel Interaction. To fit the size of kernel weight, we first project the global context information $F_C^* \in \mathbb{R}^{c_{in} \times e}$ to the output dimension space c_{out}. Inspired by [17, 37], we introduce a grouped linear layer with weight $W_{ci} \in \mathbb{R}^{\frac{c_{in}}{g} \times \frac{c_{out}}{g}}$ for the projection, where g is the number of groups. We denote the output as $F_{CI} \in \mathbb{R}^{c_{out} \times e}$.

Spatial Interaction. Then we conduct spatial interaction onto F_{CI} and F_C to get the corresponding tensors F_{A1} and F_{A2}. Specifically, we utilize two linear layers with weights $W_{A1} \in \mathbb{R}^{e \times k_1 \times k_2}$ and $W_{A2} \in \mathbb{R}^{e \times k_1 \times k_2}$. W_{A1} and W_{A2} are shared across different feature maps in F_{CI} and F_C^*, respectively. Consequently, we generate two tensors $F_{A1} = F_{CI} W_{A1}$ and $F_{A2} = F_C^* W_{A2}$.

Context Reasoning Attention Convolution. After conducting channel and spatial interaction [37], we generate $F_{A1} \in \mathbb{R}^{c_{out} \times k_1 \times k_2}$ and $F_{A2} \in \mathbb{R}^{c_{in} \times k_1 \times k_2}$. Then, we form the final context reasoning attention mask F_A via

$$F_A = F_{A1} \oplus F_{A2}, \tag{3}$$

where $F_A \in \mathbb{R}^{c_{out} \times c_{in} \times k_1 \times k_2}$ has the same size of the convolution kernel W. The operation $\oplus$ can be expressed in an element-wise way. Specifically, each element $(F_A)_{h,i,j,k}$ of F_A can be determined by

$$(F_A)_{h,i,j,k} = \sigma((F_{A1})_{h,j,k} + (F_{A2})_{i,j,k}), \tag{4}$$

where $\sigma(\cdot)$ denotes the sigmoid function. In this way, we get the attention mask F_A by considering the global context.

Then, we can apply the attention mask F_A to modulate the convolution kernel weight W as follows

$$W^* = W \odot F_A, \tag{5}$$

where the operation $\odot$ denotes element-wise multiplication.

With the modulated convolution kernel W^*, the traditional convolution process on the input feature maps could dynamically capture representative local patterns under the guidance of global context. We name it as context reasoning attention convolution (CRAC), whose primary process is shown in Figure 3. We will further show visualization results about the diversity of W^* with respect to different inputs in Section 4.6. Then, we can use CRAC further to form the basic network modules for image SR.

3.4. CRAN for Image SR

Our proposed context reasoning attention convolution (CRAC) can be easily used to replace traditional convolution. We use CRAC to build the basic block and network.

Context Reasoning Attention Block. Lim *et al.* [36] proposed simplified residual block in EDSR [19] for image SR. Such a simplified residual block has shown pretty promising performance in image SR and served as a basic building module in many follow-up works. Here, we simply follow the same block design in EDSR [36] by replacing

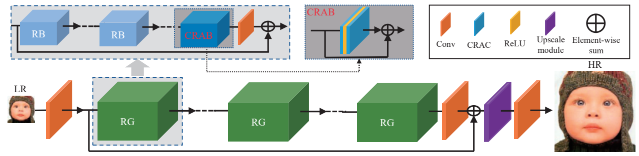


Figure 4. The pipeline of our CRAN. We use RCAN [64] as the backbone, where RG, RB, and CRAB denote residual group, residual block, and context reasoning attention block.

the traditional convolution with our proposed CRAC, resulting in context reasoning attention block (CRAB). Following the design of basic residual block [19, 36], we formulate the function of CRAB via

$$F_{out} = W_2 \sigma(W_1 F_{in}) + F_{in}, \tag{6}$$

where F_{in}, F_{out} are the input and output feature. $\sigma(\cdot)$ denotes the ReLU [16] activation function. W_1, W_2 are the weights of our proposed CRAC layer, where the bias terms are omitted for simplicity.

Context Reasoning Attention Network. We then follow the network design of RCAN [64] to build our context reasoning attention network (CRAN) in Figure 4. It should be noted that our proposed CRAC and CRAB can be used for other image SR networks. Here, we mainly focus on very deep networks and want to compare with recent related state-of-the-art (SOTA) SR methods. Specifically, we use RCAN [64] as a backbone and replace all the residual channel attention blocks [64] with the simplified residual block (RB) [36] or our proposed CRABs, resulting in the context reasoning attention network (CRAN). The super-resolved output I_{SR} of CRAN can be obtained by

$$I_{SR} = \mathcal{F}_{CRAN}(I_{LR}), \tag{7}$$

where $\mathcal{F}_{CRAN}(\cdot)$ denotes the function of our CRAN.

3.5. Implementation Details

Now we specify the implementation details of our proposed CRAN. For the CRAC, we use average pooling with h'=k_1 and w'=k_2 to resize the feature maps. In grouped linear layer, we set the group number as g=16. For network configuration, same as the backbone RCAN [64], we set residual group number as 10 in the residual in residual (RIR) [64] structure. To keep similar parameter numbers and FLOPs as RCAN, in each residual group, we set RB number as 19 and CRAB number as 1. We place one CRAB as the last block in each residual group. We set c_{in}=64, c_{out}=64, k_1=3, and k_2=3 for convolution kernel $W \in \mathbb{R}^{c_{out} \times c_{in} \times k_1 \times k_2}$ in all convolutional (Conv) layers, except for those in the input, final output Conv layers, and upscaling module. For Conv layers with kernel size 3×3 (regardless of channel dimensions), zero-padding strategy is used to keep size fixed. For upscaling module in the backbone, we follow [47, 36, 66, 64] and use ESPCNN [47] to upscale the coarse resolution features to fine ones. The final Conv layer has 3 filters, as we output color images. While, our network can also process gray-scale images.

Block Type	RB [36]	RCAB [64]	CRAB (w/o CDRR)	CRAB
PSNR (dB)	37.15	37.19	37.28	37.34

Table 1. Performance of the EDSR baseline with different block types. Networks with CRAB (w/ or w/o CDRR) perform better.

4. Experiments

4.1. Experimental Settings

Data. Following [49, 36, 18, 66, 62], we use DIV2K dataset [49] and Flickr2K [36] as training data. For validation, we use the first 10 validation images in DIV2K. For testing, we use five standard benchmark datasets: Set5 [3], Set14 [60], B100 [39], Urban100 [22], and Manga109 [40].

Image Degradation Models. We conduct experiments with Bicubic (BI), blur-downscale (BD) [61, 62, 66], and downscale-noise (DN) [61, 62] degradation models. For BD degradation model, the HR image is first blurred by a Gaussian kernel of size 7×7 with standard deviation 1.6 and then downscaled with scaling factor ×3. For DN degradation model, the HR image is first downscaled with scaling factor ×3 and then added Gaussian noise (noise level=30).

Evaluation Metrics. The SR results are evaluated with PSNR and SSIM [53] on Y channel (i.e., luminance) of transformed YCbCr space. We also compare with several leading SR methods in terms of network parameter number, FLOPs, and GPU memory usage.

Compared Methods. We compare with numerous image SR methods: SRCNN [11], FSRCNN [12], VDSR [26], IRCNN [61], EDSR [36], SRMDNF [62], DBPN [18], RDN [66], RCAN [64], RNAN [65], SRFBN [35], SAN [9], CSNLN [41], RFANet [38], HAN [43], IGNN [69], and NSR [13]. All the results are either provided by the authors, or produced by their officially released codes.

Training Settings. Data augmentation is performed on the training images, which are randomly rotated by 90°, 180°, 270° and flipped horizontally. In each training batch, 16 LR color patches with the size of 48 × 48 are extracted as inputs. To keep fair comparisons, we choose to optimize L_1 loss function, same as other compared works c[36, 66]. Our model is trained by ADAM optimizor [28] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The initial learning rate is set to 10^{-4} and then decreases to half every 2×10^5 iterations of back-propagation. We use PyTorch [44] to implement our models with Titan Xp GPUs.

4.2. Ablation Study

We study the effects of our proposed context descriptor relationship reasoning (CDRR) and context reasoning attention block (CRAB). We further investigate the effects of channel interaction, spatial interaction, and position of CRAB. We use EDSR baseline [36] as the backbone, where the residual block (RB) number and feature number are 16 and 64. We observe the best performance on validation data under BI model for ×2 SR in 200 epochs.

Spatial interaction	F_{A1}		√	√	√
	F_{A2}	√		√	√
Channel interaction	F_{C1}		√		√
PSNR (dB)		37.22	37.26	37.28	37.34

Table 2. Ablation study about spatial interaction and channel interaction in image SR. Validation performance of EDSR baseline.

CRAB Position	Baseline [36]	1-st	4-th	8-th	16-th
PSNR (dB)	37.15	37.17	37.18	37.22	37.27

Table 3. Performance of the EDSR baseline, where there are 15 RBs and one CRAB is inserted in different positions. Higher-level position helps obtain better performance.

Effects of CDRR and CRAB. In EDSR baseline, we replace all RBs with RCAB [64], our CRAB with or w/o CDRR. In Table 1, we find that RCAB achieves slight performance gain. However, our CRAB w/o CDRR obtains the obvious improvement over the baseline. These comparisons indicate that adaptively modulating the Conv kernel according to global context contribute to accurate image SR greatly. With CDRR, our CRAB achieves further improvement, which demonstrates the effectiveness of CDRR.

Channel Interaction and Spatial Interaction. We investigate channel interaction and spatial interaction [37] in image SR. As shown in Figure 3, channel interaction produces F_{C1}. Spatial interaction consists of two branches F_{A1} and F_{A2}. We provide several combinations of spatial interaction and channel interaction components and report results in Table 2. We find that each component contributes to the performance. The best result is achieved by using them all, showing the reasons why we choose them.

Effects of CRAB Position. As analyzed above, we utilize one CRAB to replace the 1-st, 4-th, 8-th, 16-th RB respectively in EDSR baseline, resulting in four cases. In Table 3, CRAB in lower-level (e.g., 1-st and 4-th) would contribute to the performance gain slightly. When we insert the CRAB into a higher-level position, we can obtain more obvious gains. Such observation helps us to set up the final configuration of a deeper network. Consequently, for our CRAN, we keep the first 19 RBs and place CRAB as the last block for all 20 residual groups. We then compare with other larger networks under different degradation models.

4.3. Results with BI Degradation Model

We compare our proposed CRAN with 13 recent image SR methods. Similar to [36, 66, 64, 9, 43], we also introduce self-ensemble strategy to improve our CRAN further and denote the self-ensembled one as CRAN+. However, we mainly compare our CRAN with others for fairness.

Quantitative Results. Table 4 shows quantitative comparisons for ×2, ×3, and ×4 SR. When compared with all previous methods, our CRAN+ performs the best on all the datasets with all scaling factors, except for SSIM value on Set5 (×2). Even without self-ensemble, our CRAN also outperforms other compared methods in all cases, except for SSIM value (copied from SAN) on Set5 (×2). Compared

Method	Scale	Set5		Set14		B100		Urban100		Manga109	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
EDSR [36]	×2	38.11	0.9602	33.92	0.9195	32.32	0.9013	32.93	0.9351	39.10	0.9773
SRMDNF [62]	×2	37.79	0.9601	33.32	0.9159	32.05	0.8985	31.33	0.9204	38.07	0.9761
DBPN [18]	×2	38.09	0.9600	33.85	0.9190	32.27	0.9000	32.55	0.9324	38.89	0.9775
RDN [66]	×2	38.24	0.9614	34.01	0.9212	32.34	0.9017	32.89	0.9353	39.18	0.9780
RCAN [64]	×2	38.27	0.9614	34.12	0.9216	32.41	0.9027	33.34	0.9384	39.44	0.9786
RNAN [65]	×2	38.17	0.9611	33.87	0.9207	32.31	0.9014	32.73	0.9340	39.23	0.9785
SRFBN [35]	×2	38.11	0.9609	33.82	0.9196	32.29	0.9010	32.62	0.9328	39.08	0.9779
SAN [9]	×2	38.31	0.9620	34.07	0.9213	32.42	0.9028	33.10	0.9370	39.32	0.9792
HAN [43]	×2	38.27	0.9614	34.16	0.9217	32.41	0.9027	33.35	0.9385	39.46	0.9785
NSR [13]	×2	38.23	0.9614	33.94	0.9203	32.34	0.9020	33.02	0.9367	39.31	0.9782
IGNN [69]	×2	38.24	0.9613	34.07	0.9217	32.41	0.9025	33.23	0.9383	39.35	0.9786
CSNLN [41]	×2	38.28	0.9616	34.12	0.9223	32.40	0.9024	33.25	0.9386	39.37	0.9785
RFANet [38]	×2	38.26	0.9615	34.16	0.9220	32.41	0.9026	33.33	0.9389	39.44	0.9783
CRAN (ours)	×2	38.31	0.9617	34.22	0.9232	32.44	0.9029	33.43	0.9394	39.75	0.9793
CRAN+ (ours)	×2	38.36	0.9619	34.37	0.9243	32.48	0.9033	33.61	0.9405	39.89	0.9798
EDSR [36]	×3	34.65	0.9280	30.52	0.8462	29.25	0.8093	28.80	0.8653	34.17	0.9476
SRMDNF [62]	×3	34.12	0.9254	30.04	0.8382	28.97	0.8025	27.57	0.8398	33.00	0.9403
RDN [66]	×3	34.71	0.9296	30.57	0.8468	29.26	0.8093	28.80	0.8653	34.13	0.9484
RCAN [64]	×3	34.74	0.9299	30.65	0.8482	29.32	0.8111	29.09	0.8702	34.44	0.9499
RNAN [65]	×3	34.66	0.9290	30.53	0.8463	29.26	0.8090	28.75	0.8646	34.25	0.9483
SRFBN [35]	×3	34.70	0.9292	30.51	0.8461	29.24	0.8084	28.73	0.8641	34.18	0.9481
SAN [9]	×3	34.75	0.9300	30.59	0.8476	29.33	0.8112	28.93	0.8671	34.30	0.9494
HAN [43]	×3	34.75	0.9299	30.67	0.8483	29.32	0.8110	29.10	0.8705	34.48	0.9500
NSR [13]	×3	34.62	0.9289	30.57	0.8475	29.26	0.8100	28.83	0.8663	34.27	0.9484
IGNN [69]	×3	34.72	0.9298	30.66	0.8484	29.31	0.8105	29.03	0.8696	34.39	0.9496
CSNLN [41]	×3	34.74	0.9300	30.66	0.8482	29.33	0.8105	29.13	0.8712	34.45	0.9502
RFANet [38]	×3	34.79	0.9300	30.67	0.8487	29.34	0.8115	29.15	0.8720	34.59	0.9506
CRAN (ours)	×3	34.80	0.9304	30.73	0.8498	29.38	0.8124	29.33	0.8745	34.84	0.9515
CRAN+ (ours)	×3	34.89	0.9309	30.82	0.8508	29.42	0.8131	29.50	0.8768	35.06	0.9525
EDSR [36]	×4	32.46	0.8968	28.80	0.7876	27.71	0.7420	26.64	0.8033	31.02	0.9148
SRMDNF [62]	×4	31.96	0.8925	28.35	0.7787	27.49	0.7337	25.68	0.7731	30.09	0.9024
DBPN [18]	×4	32.47	0.8980	28.82	0.7860	27.72	0.7400	26.38	0.7946	30.91	0.9137
RDN [66]	×4	32.47	0.8990	28.81	0.7871	27.72	0.7419	26.61	0.8028	31.00	0.9151
RCAN [64]	×4	32.63	0.9002	28.87	0.7889	27.77	0.7436	26.82	0.8087	31.22	0.9173
RNAN [65]	×4	32.43	0.8977	28.83	0.7871	27.72	0.7410	26.61	0.8023	31.09	0.9149
SRFBN [35]	×4	32.47	0.8983	28.81	0.7868	27.72	0.7409	26.60	0.8015	31.15	0.9160
SAN [9]	×4	32.64	0.9003	28.92	0.7888	27.78	0.7436	26.79	0.8068	31.18	0.9169
HAN [43]	×4	32.64	0.9002	28.90	0.7890	27.80	0.7442	26.85	0.8094	31.42	0.9177
NSR [13]	×4	32.55	0.8987	28.79	0.7876	27.72	0.7414	26.61	0.8025	31.10	0.9145
IGNN [69]	×4	32.57	0.8998	28.85	0.7891	27.77	0.7434	26.84	0.8090	31.28	0.9182
CSNLN [41]	×4	32.68	0.9004	28.95	0.7888	27.80	0.7439	27.22	0.8168	31.43	0.9201
RFANet [38]	×4	32.66	0.9004	28.88	0.7894	27.79	0.7442	26.92	0.8112	31.41	0.9187
CRAN (ours)	×4	32.72	0.9012	29.01	0.7918	27.86	0.7460	27.13	0.8167	31.75	0.9219
CRAN+ (ours)	×4	32.79	0.9022	29.07	0.7929	27.91	0.7470	27.30	0.8197	32.02	0.9239

Table 4. Quantitative results with BI degradation model. Best and second best results are colored with red and blue.

with attention-based methods (e.g., RCAN, SAN, RNAN, HAN, and CSNLN), especially the backbone RCAN used in our work, our CRAN achieves higher PSNR/SSIM values in most cases. This comparison indicates that our proposed CRAN can further improve the performance by modulating Conv layer kernels with global context reasoning attention.

Visual Results. In Figure 5, we further show visual comparisons on scale ×4. Here, we mainly provide some representative challenging cases about texture and small details (e.g., tiny lines). In image "img_034", there has some brick textures according to the HR image. Most compared methods can hardly recover such textures, but suffer from some blurring artifacts. In contrast, our CRAN can alleviate the blurring artifacts better to some degree and recover parts of textures. In image "img_044", most of the compared methods cannot recover the tiny horizontal lines clearly. However, our CRAN produces much sharper structural details, being more faithful to the ground truth.

In image "img_092", there are several groups of strips in different directions. All the compared methods cannot reconstruct recover the top-right strips correctly. They either suffer from heavy blurring artifacts (e.g., EDSR, DBPN, RDN, RCAN, and SAN) or output strips with wrong direction (e.g., CSNLN and RFANet). However, our CRAN handles this challenge better and recovers shaper strips. This is mainly because we consider the global context information and encode it into the Conv layer kernel modulation. Those obvious visual comparisons with most recent SOTA methods further demonstrate the effectiveness of our CRAN.

Method	S	Set5	Set14	B100	Urban100	Manga109
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Bicubic	×3	28.78/0.8308	26.38/0.7271	26.33/0.6918	23.52/0.6862	25.46/0.8149
SRCNN [11]	×3	32.05/0.8944	28.80/0.8074	28.13/0.7736	25.70/0.7770	29.47/0.8924
FSRCNN [12]	×3	26.23/0.8124	24.44/0.7106	24.86/0.6832	22.04/0.6745	23.04/0.7927
VDSR [26]	×3	33.25/0.9150	29.46/0.8244	28.57/0.7893	26.61/0.8136	31.06/0.9234
IRCNN [61]	×3	33.38/0.9182	29.63/0.8281	28.65/0.7922	26.77/0.8154	31.15/0.9245
SRMDNF [62]	×3	34.01/0.9242	30.11/0.8364	28.98/0.8009	27.50/0.8370	32.97/0.9391
RDN [66]	×3	34.58/0.9280	30.53/0.8447	29.23/0.8079	28.46/0.8582	33.97/0.9465
RCAN [64]	×3	34.70/0.9288	30.63/0.8462	29.32/0.8093	28.81/0.8647	34.38/0.9483
SRFBN [35]	×3	34.66/0.9283	30.48/0.8439	29.21/0.8069	28.48/0.8581	34.07/0.9466
SAN [9]	×3	34.75/0.9290	30.68/0.8466	29.33/0.8101	28.83/0.8646	34.46/0.9487
HAN [43]	×3	34.76/0.9294	30.70/0.8475	29.34/0.8106	28.99/0.8676	34.56/0.9494
RFANet [38]	×3	34.77/0.9292	30.68/0.8473	29.34/0.8104	28.89/0.8661	34.49/0.9492
CRAN (ours)	×3	34.90/0.9302	30.79/0.8485	29.40/0.8115	29.17/0.8706	34.97/0.9512
CRAN+ (ours)	×3	34.93/0.9305	30.86/0.8493	29.43/0.8121	29.34/0.8727	35.16/0.9519

Table 5. Quantitative results with BD degradation model. Best and second best results are colored with red and blue.

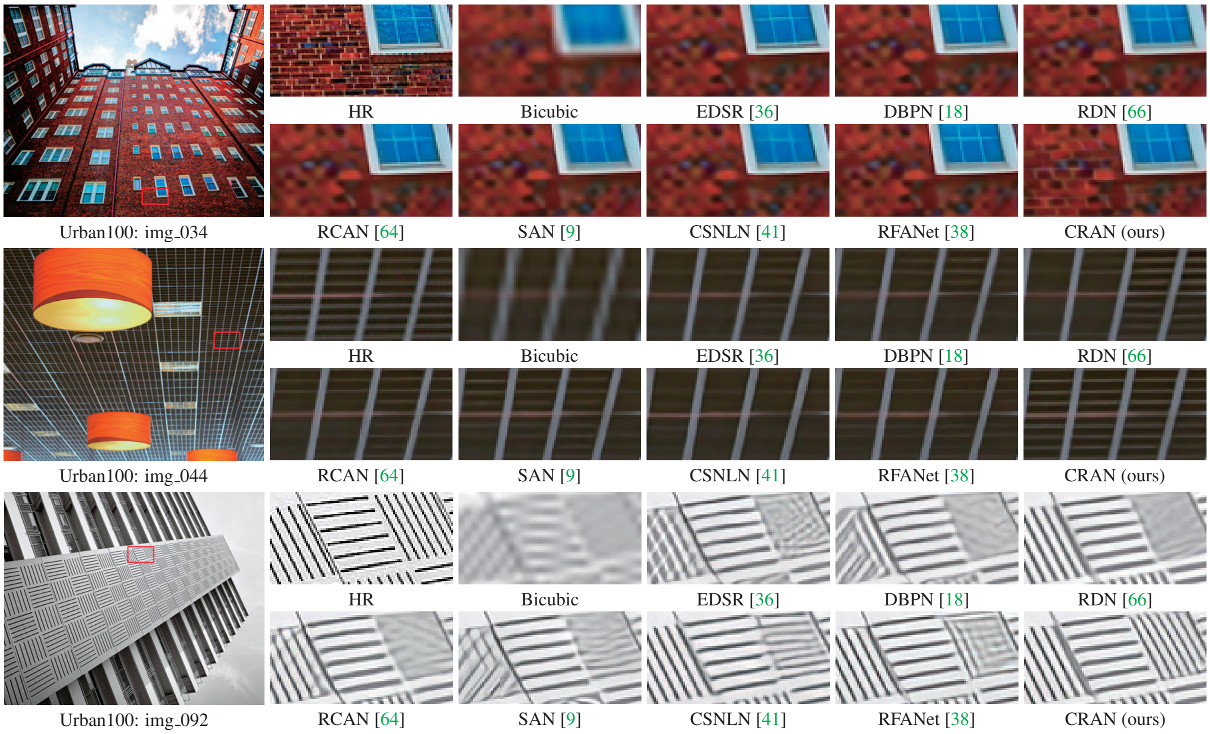


Figure 5. Visual comparison for 4× SR with BI model on Urban100 dataset.

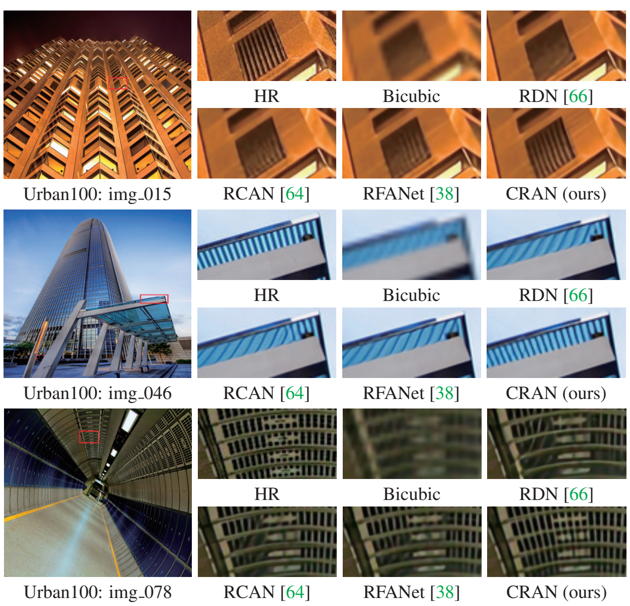


Figure 6. Visual comparison for 3× SR with BD model.

4.4. Results with BD Degradation Model

We apply our method to super-resolve images with blur-down (BD) degradation model, which is also commonly used in recent image SR works [61, 62, 66, 64, 38].

Quantitative Results. In Table 5, RFANet has achieved very high performance on each dataset. However, our proposed CRAN can obtain notable gains over RFANet. We can achieve even better results with self-ensemble (i.e., CRAN+). Our CRAN achieves larger gains compared with attention-based SR methods (e.g., RCAN and SAN). This comparison also indicates that adaptively modulating the Conv layer kernels with context information could perform better than those modifying local features.

Method	S	Set5	Set14	B100	Urban100	Manga109
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Bicubic	×3	24.01/0.5369	22.87/0.4724	22.92/0.4449	21.63/0.4687	23.01/0.5381
SRCNN [11]	×3	25.01/0.6950	23.78/0.5898	23.76/0.5538	21.90/0.5737	23.75/0.7148
FSRCNN [12]	×3	24.18/0.6932	23.02/0.5856	23.41/0.5556	21.15/0.5682	22.39/0.7111
VDSR [26]	×3	25.20/0.7183	24.00/0.6112	24.00/0.5749	22.22/0.6096	24.20/0.7525
IRCNN_G [61]	×3	25.70/0.7379	24.45/0.6305	24.28/0.5900	22.90/0.6429	24.88/0.7765
IRCNN_C [61]	×3	27.48/0.7925	25.92/0.6932	25.55/0.6481	23.93/0.6950	26.07/0.8253
RDN [66]	×3	28.47/0.8151	26.60/0.7101	25.93/0.6573	24.92/0.7364	28.00/0.8591
CRAN (ours)	×3	28.74/0.8235	26.77/0.7178	26.04/0.6647	25.43/0.7566	28.44/0.8692
CRAN+ (ours)	×3	28.76/0.8240	26.80/0.7186	26.06/0.6652	25.51/0.7587	28.55/0.8708

Table 6. Quantitative results with DN degradation model. Best and second best results are colored with red and blue.

Visual Results. We also provide visual comparisons in Figure 6, where the LR images are further blurred. For challenging details in images "img_015" and "img_078", most methods either suffer from heavy blurring artifacts or recover parts of the columns. CRAN deblurs them to a deeper degree and can recover more columns. In image "img_046", most compared methods produce some column-like details with wrong direction. In contrast, our CRAN obtains much better results by recovering the correct components. These comparisons indicate that kernel modulation with context reasoning attention would alleviate the blurring artifacts.

4.5. Results with DN Degradation Model

We further provide comparisons under the more challenging DN degradation model [61, 66], where the LR images are further added with heavy noise (noise level=30).

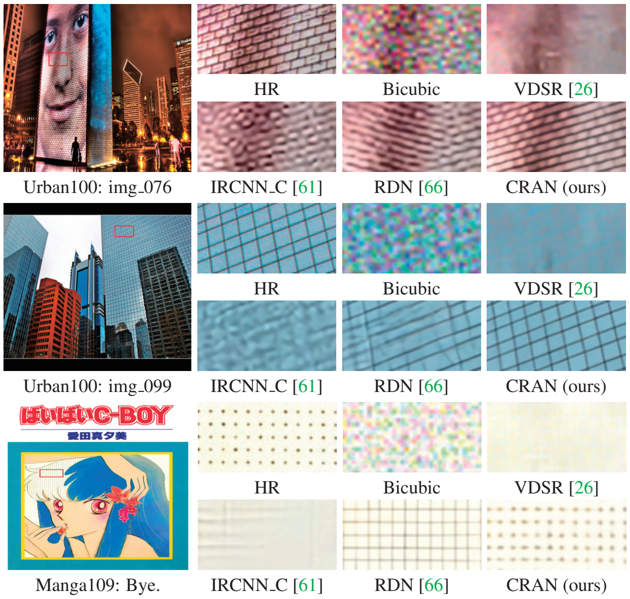


Figure 7. Visual comparison for 3× SR with DN model on Urban100 and Manga109 datasets.

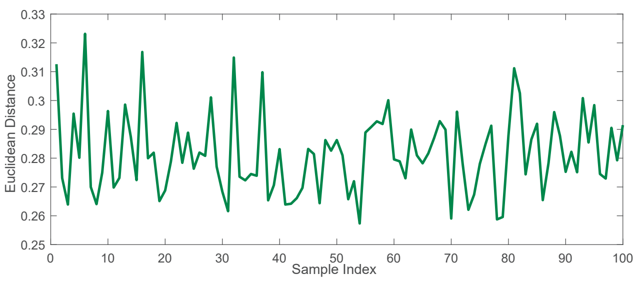


Figure 8. W^* diversity investigation according to different inputs.

Quantitative Results. As shown in Table 6, RDN has achieved very high PSNR/SSIM values on each dataset. While, our CRAN can further achieve notable performance gains over RDN. Compared with the usage of hierarchical features in RDN, our CRAN shows promising potential to deal with noisy images with context reasoning attention.

Visual Results. We further show visual comparisons for pretty challenging cases in Figure 7. In image "img_076", where the textural structures are noisy, the compared methods would either fail to recover the texture or generate obviously different structures (e.g., RDN). Our CRAN removes noise and obtains better textural structures. We also show some grid-like cases in images "img_099" and "Bye.", where the heavy noise could lead most SR methods to over-smooth the results (e.g., VDSR and IRCNN). RDN may even produce wrong structures (e.g., in image "Bye."). However, having a global sense of the noisy texture with context information, our CRAN obtains much better visual results, showing stronger ability to suppress noise.

4.6. Diversity of Convolution Kernel W^*

We show how much the convolution kernel W in Eq. (5) would be modulated to W^* according to different inputs.

	EDSR [36]	RCAN [64]	SAN [9]	CSNLN [41]	CRAN
Parameters (M)	40.73	15.44	15.67	3.06	14.94
FLOPs (G)	1,042.74	391.86	400.46	2,245.98	372.99
GPU mem. (Mb)	1,089	661	8,177	8,099	669
Running Time (s)	0.37	0.85	1.45	7.14	0.96
PSNR (dB)	39.10	39.44	39.32	39.37	39.75

Table 7. Number of parameters, FLOPs, GPU memory, and performance on Manga109 with scaling factor ×2 (BI model). When we calculate FLOPs and time, we use input size of 3×160×160.

Namely, how diverse would W^* be? To investigate the diversity of W^*=$W \odot F_A$, we consider the average Euclidean distance between F_A and the all-ones matrix I. We randomly forward 100 images into the network and calculate distance for each sample. We show the visualization results in Fig. 8. We can see that W^* is diverse based on different input, indicating the adaptive modification of W^*.

4.7. Model Complexity Analyses

We further show comparisons with recent representative image SR works about model complexity in terms of model size, FLOPs, GPU memory, running time, and performance in Table 7. It shows that EDSR [36] has the largest model size. Our CRAN has slightly less parameter number than that in RCAN [64] and SAN [9]. CSNLN [41] has much smaller model size in a recurrent framework, which actually costs huge computation operations. Specifically, when the input size is 3×160×160, CSNLN would use over 2.2×10^3 G FLOPs, being over 6 times as ours. Our CRAN also needs much less running time than CSNLN. Both SAN and CSNLN would consume over 8×10^3 Mb GPU memory, being over 12 times as ours. Although RCAN, as our backbone, has similar model size, FLOPs, GPU memory, and running time as ours, our CRAN obtains notable SR performance gain over RCAN. Those comparisons and analyses indicate that our CRAN achieves a better efficiency trade-off between model complexity and performance.

5. Conclusion

Global context information is crucial for accurate image super-resolution (SR). Recent works in neuroscience motivate us to modify the convolution kernel according to the global context dynamically. Therefore, we propose a context reasoning attention network (CRAN) for image SR. Specifically, we project the input feature into latent representations and extract global context descriptors. The context relationship descriptors are further enhanced by using the descriptor relationship with semantic reasoning. Channel and spatial interactions are then introduced to generate context reasoning attention mask, which is applied to modify the convolution kernel adaptively. We use modulated convolution layers as basic components to build blocks and networks. Consequently, our CRAN achieves superior SR results under different degradation models and a favourable trade-off between performance and model complexity.

Acknowledgments. The work was partially supported by NSF award ECCS-1916839 and NSF award IIS-1835231.

References

[1] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *ECCV*, 2018. 1

[2] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *ICCV*, 2019. 2

[3] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *BMVC*, 2012. 5

[4] Xinlei Chen, Li-Jia Li, Li Fei-Fei, and Abhinav Gupta. Iterative visual reasoning beyond convolutions. In *CVPR*, 2018. 3

[5] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *CVPR*, 2019. 2, 3

[6] Changmao Cheng, Yanwei Fu, Yu-Gang Jiang, Wei Liu, Wenlian Lu, Jianfeng Feng, and Xiangyang Xue. Dual skipping networks. In *CVPR*, 2018. 2

[7] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017. 3, 4

[8] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 2

[9] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *CVPR*, 2019. 1, 2, 5, 6, 7, 8

[10] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, 2014. 1, 2

[11] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 2016. 5, 6, 7

[12] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, 2016. 1, 5, 6, 7

[13] Yuchen Fan, Jiahui Yu, Yiqun Mei, Yulun Zhang, Yun Fu, Ding Liu, and Thomas S Huang. Neural sparse representation for image restoration. In *NeurIPS*, 2020. 5, 6

[14] William T Freeman, Egon C Pasztor, and Owen T Carmichael. Learning low-level vision. *IJCV*, 2000. 1

[15] Charles D Gilbert and Wu Li. Top-down influences on visual processing. *Nature Reviews Neuroscience*, 2013. 1, 2

[16] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *AISTATS*, 2011. 4

[17] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. In *ICLR*, 2017. 4

[18] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *CVPR*, 2018. 1, 5, 6, 7

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 4

[20] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 3, 4

[21] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 2, 3

[22] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, 2015. 1, 5

[23] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *ACM MM*, 2019. 1

[24] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *NeurIPS*, 2016. 2

[25] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *CVPR*, 2018. 2

[26] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016. 1, 2, 5, 6, 7, 8

[27] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *CVPR*, 2016. 2

[28] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 5

[29] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 3

[30] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017. 3

[31] Wonkyung Lee, Junghyup Lee, Dohyung Kim, and Bumsub Ham. Learning with privileged information for efficient image super-resolution. In *ECCV*, 2020. 1

[32] Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. Visual semantic reasoning for image-text matching. In *ICCV*, 2019. 3

[33] Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. Visual semantic reasoning for image-text matching. In *ICCV*, 2019. 3

[34] Wu Li, Valentin Piëch, and Charles D Gilbert. Perceptual learning and top-down influences in primary visual cortex. *Nature neuroscience*, 2004. 1

[35] Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu. Feedback network for image super-resolution. In *CVPR*, 2019. 5, 6

[36] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, 2017. 1, 2, 3, 4, 5, 6, 7, 8

[37] Xudong Lin, Lin Ma, Wei Liu, and Shih-Fu Chang. Context-gated convolution. In *ECCV*, 2020. 2, 3, 4, 5

[38] Jie Liu, Wenjie Zhang, Yuting Tang, Jie Tang, and Gangshan Wu. Residual feature aggregation network for image super-resolution. In *CVPR*, 2020. 1, 5, 6, 7

[39] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001. 5

[40] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 2017. 5

[41] Yiqun Mei, Yuchen Fan, Yuqian Zhou, Lichao Huang, Thomas S Huang, and Humphrey Shi. Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining. In *CVPR*, 2020. 1, 3, 5, 6, 7, 8

[42] Allen Newell. Physical symbol systems. *Cognitive science*, 1980. 3

[43] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Haifeng Shen. Single image super-resolution via a holistic attention network. In *ECCV*, 2020. 5, 6

[44] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[45] Mehdi SM Sajjadi, Bernhard Schölkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *ICCV*, 2017. 1

[46] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *NeurIPS*, 2017. 3

[47] 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. 4

[48] Wenzhe Shi, Jose Caballero, Christian Ledig, Xiahai Zhuang, Wenjia Bai, Kanwal Bhatia, Antonio M Simoes Monteiro de Marvao, Tim Dawes, Declan ORegan, and Daniel Rueckert. Cardiac image super-resolution with global correspondence using multi-atlas patchmatch. In *MICCAI*, 2013. 1

[49] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, Lei Zhang, Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, Kyoung Mu Lee, et al. Ntire 2017 challenge on single image super-resolution: Methods and results. In *CVPRW*, 2017. 5

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 3

[51] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *CVPR*, 2017. 2

[52] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 2, 3

[53] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004. 5

[54] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, 2018. 2

[55] Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. In *ICLR*, 2019. 2

[56] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. *ICLR*, 2019. 3

[57] Yibo Yang, Zhisheng Zhong, Tiancheng Shen, and Zhouchen Lin. Convolutional neural networks with alternately updated clique. In *CVPR*, 2018. 2

[58] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *ECCV*, 2018. 3

[59] Amir R Zamir, Te-Lin Wu, Lin Sun, William B Shen, Bertram E Shi, Jitendra Malik, and Silvio Savarese. Feedback networks. In *CVPR*, 2017. 2

[60] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Proc. 7th Int. Conf. Curves Surf.*, 2010. 5

[61] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *CVPR*, 2017. 5, 6, 7, 8

[62] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *CVPR*, 2018. 1, 5, 6, 7

[63] Yulun Zhang, Kai Li, Kunpeng Li, and Yun Fu. Mr image super-resolution with squeeze and excitation reasoning attention network. In *CVPR*, 2021. 3

[64] 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. 1, 2, 4, 5, 6, 7, 8

[65] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. In *ICLR*, 2019. 1, 5, 6

[66] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *CVPR*, 2018. 1, 2, 4, 5, 6, 7, 8

[67] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image restoration. *TPAMI*, 2020. 1

[68] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018. 3

[69] Shangchen Zhou, Jiawei Zhang, Wangmeng Zuo, and Chen Change Loy. Cross-scale internal graph neural network for image super-resolution. In *NeurIPS*, 2020. 5, 6

[70] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, 2019. 2

[71] Wilman WW Zou and Pong C Yuen. Very low resolution face recognition problem. *TIP*, 2012. 1