The Visual Computer (2024) 40:619–635
https://doi.org/10.1007/s00371-023-02805-1

ORIGINAL ARTICLE

Relation-consistency graph convolutional network for image super-resolution

Yue Yang[1] · Yong Qi[1] · Saiyu Qi[1]

Accepted: 6 February 2023 / Published online: 6 April 2023

Abstract
Convolutional neural networks (CNNs) have been widely exploited in single image super-resolution (SISR) due to their powerful feature representation and the end-to-end training paradigm. Recent CNN-based SISR methods employ attention mechanism to enrich the feature representation and achieve notable performance. However, most of them use attention mechanism to model channel-wise dependencies, and the global relations of image features are not fully explored, thus hindering the discriminative learning capability. To amplify the feature representation, we propose a relation-consistency graph convolutional network (RGCN) for high-quality image rendering. Specifically, we introduce a spatial graph attention (SGA) to encode feature correlations in spatial dimension. Within SGA, the parameter-free Gram matrix is adopted to construct the global dependencies of pixel features, which dynamically measure the pixel-wise spatial relation. Furthermore, we embed a spatial pyramid pooling scheme into SGA to reduce the high complexity of correlation modeling between two pixels. Such an operation efficiently constructs the spatial relations through pixel and region-pooled features. Moreover, we propose a relation-consistency loss to retain the invariant of global relationship across all feature layers. The proposed loss regularizes the consistency between the low-resolution input and its corresponding high-resolution output in terms of the spatial relationships, enabling our network to learn a reasonable mapping and reconstruct more realistic images. Qualitative and quantitative comparison against state-of-the-art SISR methods on benchmark datasets under various degradation models demonstrate the superior performance of our RGCN.

Keywords Image super-resolution · Graph convolutional · Attention mechanism · Gram matrix

1 Introduction

Single image super-resolution (SISR) is a fundamental low-level vision task, which aims to generate a high-resolution (HR) image from its low-resolution (LR) counterparts. SISR is an ill-posed problem as multiple HR solutions can map to any LR inputs. Hence, plenty of image SR approaches have been proposed to tackle this inverse issue, ranging from early interpolation-based [1,2] to the latest deep learning-based methods [3–11].

✉ Yong Qi
qiy@xjtu.edu.cn

Yue Yang
yang_yue2015@stu.xjtu.edu.cn

Saiyu Qi
saiyu-qi@xjtu.edu.cn

1 School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, Shaanxi, China

Benefiting from the powerful feature representation and end-to-end training paradigm of convolutional neural networks (CNNs), a flurry of CNN-based SISR methods have been presented to learn a mapping from LR input image to its corresponding HR output, obtaining impressive improvement over the conventional approaches. As a pioneer work, Dong et al. introduced CNN to the image SR field and proposed SRCNN [3] with three convolutional layers. To explore more high-level information, SAN [12] and HAN [13] incorporated attention mechanism into SR methods to capture long-range features interdependencies, which achieved noticeable improvement. Later, SwinIR [14] combined the advantage of CNN and Transformer [15] to process large-scale image and model the global relationship simultaneously, it obtained better performance with less parameters. More advanced and complex SISR methods [9,11,16–18] are proposed to promote the quality of the reconstructed image.

Although remarkable performance has been achieved by these image SR methods, they still have some limitations.

Few of them have been able to draw on the spatial correlations of image features to obtain contextual information, resulting in unpleasant super-resolved outputs. In particular, the pixels of image features have different relations with each other, exploration of these contextual relations is helpful for better discriminative learning. Tracing back to the early classical self-example studies [19,20], they captured self-similar patterns among the whole image to provide the relations between pixels in a global view, showing the significance of spatial relations in image SR field. Thus, how to model the feature spatial dependencies is presented as a key issue.

In this paper, we propose a novel relation-consistency graph convolutional network (RGCN), exploiting spatial information to enhance feature representation. To be specific, a spatial graph attention (SGA) is proposed for spatial feature encoding, which dynamically models the feature relations with awareness of global information. We first calculate all-region similarities in the feature space and then update features based on the global similarities in SGA. In this way, each pixel feature is learned from the whole image representation through graph convolutional operations. We employ the Gram matrix in this paper to construct global correlations without introducing extra parameters. These correlations formulate the adjacency matrix in SGA to update pixel-wise features through all-region relationships. As shown in Fig. 1, due to the similarity and regularity characteristics of the floor texture, capturing the long-range information provides more clues to recover finer image details. It is noteworthy that modeling the spatial relations between two pixels usually consumes heavily computational resources, especially when the image size is large. We thereby embed a spatial pyramid pooling scheme into SGA to account for this issue. The pyramid pooling constructs the spatial relationships via pixels and grid features, reducing the computational overhead.

In addition, [21] found that the commonly used pixel-wise loss (e.g., $\mathcal{L}_1$ loss) tends to produce over-smoothed results since it is oblivious to the semantic information of the image. As the spatial relation is a natural and static characteristic, that is, the super-resolved SR image should share similar semantic relations with its corresponding LR input image. We hence introduce the relation-consistency loss to maintain the spatial relationships between low-level features and high-level features. Specifically, we minimize the discrepancy between adjacent matrices of the first and last feature layers. It regularizes the model to retain consistent spatial relations after image super-resolution. Overall, our network learns a more reasonable mapping with the above designs for reconstructing images with finer details.

Our main contributions can be summarized as follows:

- We propose a novel relation-consistency graph convolutional network (RGCN) to enhance the learning ability through contextual information modeling, thus recovering images with finer details.
- We propose a spatial graph attention (SGA) to encode feature spatial correlations. Within SGA, the adjacency matrix is calculated by the Gram matrix without learnable parameters. Meanwhile, a spatial pyramid pooling scheme is embedded into SGA to reduce computational costs.
- We propose a simple yet effective loss term, namely relation-consistency loss, to maintain the consistency of spatial information between the LR input image and its corresponding SR output.
- Extensive experiments on various degradation models demonstrate the superiority of our RGCN in terms of quantitative metrics and visual quality.

The rest of this paper is organized as follows. In Sect. 2, we mainly review the related works about CNN-based SISR methods and describe some relative approaches about attention mechanism and graph convolutional networks. In Sect. 3, we introduce our proposed SR network in detail. To verify the effectiveness of our method, the experimental comparisons and evaluations are presented in Sect. 4. Finally, we conclude our work in Sect. 5.

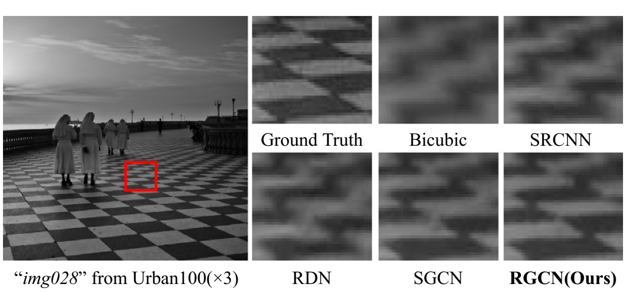


Fig. 1 Visual comparisons for scaling factor ×3 on "img_028" from Urban100. Our proposed RGCN obtains better visual quality with sharper edges compared with other state-of-the-art SISR methods

2 Related work

2.1 CNN-based SISR methods

Recently, deep convolutional neural networks (CNNs) have been extensively studied in various computer vision communities. The powerful representational ability and end-to-end training paradigm of CNN make it widely used in the SISR field. The pioneering work was done by Dong et al. who proposed a shallow convolutional network (SRCNN) [3] to predict the non-linear relationship between the interpolated LR image and HR image, achieving considerable improvement over the traditional methods. Later, Kim et al. designed deeper networks VDSR [5] and DRCN [22] to capture more high-level information based on residual learning [23] and recursive learning. To control the number of model parameters and maintain persistent memory, Tai et al. introduced DRRN [24] with a novel recursive block and further designed MemNet [25] with memory blocks and dense connections.

For the described methods above, the LR images need to be interpolated to coarse HR images with the desired size, which inevitably increases the computational costs and produces side effects (e.g., noise amplification and blurring). To overcome these drawbacks, post-upsampling architecture is proposed and has soon become the mainstream framework in image SR task. Lim et al. introduced a very deep and wide network EDSR [26] by stacking simplified residual blocks in which the unnecessary layers are removed. Similarly, Zhang et al. proposed a residual dense network (RDN) [8] to facilitate effective feature learning through a continuous memory mechanism. Li et al. built SRFBN [6] that utilizes recurrent neural network and feedback mechanism to refine low-level information with high-level image details. Vassilo et al. [27] incorporated multi-agent reinforcement learning and proposed an ensemble GAN-based SR network to increase the quality of reconstructed image. Fang et al. [18] introduced an accurate and efficient soft-edge assisted network, which employed the image prior knowledge into the network for better image reconstruction. Furthermore, Niu et al. [28] proposed CSN with an efficient channel segregation block that attempts to enlarge the size of receptive fields to capture informative information, thus promoting the quality of super-resolved image.

Compared to these CNN-based methods limited to local relations constraints, in this work, we adopt a spatial graph attention mechanism to model contextual information in spatial dimension.

2.2 Attention mechanism

The attention mechanism was initially proposed by Sutskever et al. in machine translation [29] via giving different weights to the input. Coupled with deep networks, attention mechanism has gained popularity in a variety of high-level vision tasks [30–32]. Hu et al. proposed a "squeeze-and-extraction" (SE) [33] block to enhance the learning ability by modeling the channel-wise inter-dependencies. Woo et al. introduced a convolutional block attention module (CBAM) [34] which captures the feature relations along the channel and spatial dimension, respectively. Recent image SR studies have been conducted using attention mechanism and shown remarkable performance gain. Zhang et al. integrated SE block into residual learning and established a deeper network RCAN [35]. The channel-wise attention mechanism utilizes global average pooling to selectively highlight the channel map. Hu et al. presented a CSFM [36] network that combined channel-wise and spatial attention to construct the feature dependencies to enhance the quality of output HR images. Besides, Dai et al. proposed the second-order attention network (SAN) [12] to exploit more powerful feature expressions by using second-order feature statistics. A recent SR approach HAN [13] proposed a layer attention module to model the relationships of features, thus enabling the network to produce the high-quality image. Later, SwinIR [14] utilized several residual Swin Transformer blocks to extract deep features, which obtained impressive performance with less parameters on various low-level vision tasks. To reduce computation costs while maintaining the reconstruction performance, Mei et al. [11] combined sparse feature representation with nonlocal to capture long-range dependencies.

Though the above attention-based SR methods attained noticeable performance, they pay less attention to feature spatial relations modeling. In our work, we aim at modeling the contextual dependencies via graph convolutional operation.

2.3 Graph convolutional network

The concept of graph neural network (GNN) [37] was first proposed by Gori et al. which well-processes the graph-structured non-Euclidean data. The GNN collectively aggregates the node features in a graph and properly embed the graph in a new discriminative space. However, as for regular Euclidean data like images and text, it is hard to apply GNNs straightforwardly [38]. Therefore, defining a convolution-like operation for regular structure data is a major challenge. The graph convolutional networks (GCNs) provide a well-solution to solve this problem. Bruna et al. [39] developed the operation of "graph convolution" based on spectral property, which convolved on the neighborhood of every graph node and produces a node-level output, but led to expensive computational costs. After that, a flurry of graph convolutional studies has been presented. Kipf et al. [40] introduced a fast approximation localized convolution on image classification, which not only simplified the convolution operation but also alleviated the problem of overfitting. Li et al. [41] designed a residual graph convolutional broad network for emotion recognition, which extracts features and abstract features via employing the GCN-based residual block. It not only improves the performance of the network but also extracts higher-level information. In addition, bease on the attention mechanism, Wei et al. [42] constructed a cascade framework between the graph convolutional layers via dense connections, which further enhancing the graph representation capability.

With the property of graph convolution, we propose a spatial graph attention mechanism to exploit global relations of image features. Instead of directly modeling the pairwise relationships of features, we further embed a pyramid pooling scheme in graph convolutional operation, which effectively reduces the computational resources.

3 Proposed method

In this section, we first introduce an overview of our proposed network for image SR. We then describe the details of the designed spatial graph attention and relation-consistency loss, which are the core of our network.

3.1 Network architecture

The overall architecture of our relation-consistency graph convolutional network (RGCN) is shown in Fig. 2 given a low-resolution (LR) image I_{LR} and its corresponding super-resolved (SR) image I_{SR} as the input and output of our RGCN. As explored in [12,35], we first use a convolutional layer to extract shallow feature F_0 from the initial LR input

$$F_0 = \mathcal{H}_{\mathrm{SF}}(I_{\mathrm{LR}}), \tag{1}$$

where $\mathcal{H}_{\mathrm{SF}}(\cdot)$ represents the convolution operation. F_0 is then served as an input for a series of attention-based feature refinement modules (AFRMs). Supposing we have N stacked AFRMs, thus the output F_n of n-th AFRM is formulated as

$$F_n = \mathcal{H}_{\mathrm{AFRM}}(F_{n-1}), \tag{2}$$

where $\mathcal{H}_{\mathrm{AFRM}}(\cdot)$ stands for the function of AFRM. After obtaining informative features with a set of AFRMs, global feature fusion is further applied to extract global feature F_{glo} by fusing features from all AFRMs

$$F_{glo} = \mathcal{H}_{GFF}(F_1, \cdots, F_N), \tag{3}$$

where $\mathcal{H}_{\mathrm{GFF}}(\cdot)$ denotes the convolutional layer with the kernel size of 1×1 to aggregate features from all modules. We utilize global residual learning before conducting upscale operation by

$$F_{\mathrm{DF}} = F_0 + F_{\mathrm{glo}}, \tag{4}$$

where F_{DF} denotes the obtained deep feature. Finally, the feature F_{LR} is upscaled via the upsampler to generate SR image I_{SR}. Inspired by [43], we adopt the sub-pixel layer with one convolutional layer followed

$$I_{\mathrm{SR}} = \mathcal{H}_{\uparrow}(F_{\mathrm{LR}}), \tag{5}$$

where $\mathcal{H}_{\uparrow}(\cdot)$ stands for the operation of upsampler.

3.2 Attention-based feature refinement module

As shown in Fig. 2b, the attention-based feature refinement module (AFRM) contains two parts: an Inception-style feature extraction, and a two-stream attention.

3.2.1 Inception-style feature extraction

Several studies [44] have demonstrated that multi-scale features carry rich information, which are beneficial for accurate SR image reconstruction. To this end, we employ the well-known *Inception* module [45] into our AFRM as a multi-scale feature extractor, and simplify its structure by remaining two different convolution kernel sizes (*i.e.,* 3×3 and 5×5). Besides, we leverage dense connections in feature extraction to reuse the features from preceding layers.

3.2.2 Two-stream attention

As shown in Fig. 2b, the two-stream attention is constructed by two-paralleled attention, aiming at modeling the feature dependencies in channel and spatial dimensions, respectively. Within the two-stream attention, we separately learn the feature relations between channels (*i.e.,* channel-wise attention) and pixels (*i.e.,* spatial graph attention) and then aggregate their corresponding outputs to strengthen the feature representation. Specifically, the channel-wise attention (CA) explores the inter-dependencies across feature channels by [33] to adaptively rescale each channel-wise feature, while the spatial graph attention (SGA) dynamically models the feature relations with awareness of global information. In order to take full utilization of the learned information from channel and spatial dimensions, we place CA and SGA in a parallel manner. Moreover, we investigate different arrangements (*i.e.,* parallel and sequential) of CA and SGA in Sect. 4, which experimentally found that parallel arrangement gives a better result than doing in a sequential way.

3.3 Spatial graph attention

In recent CNN-based SISR studies [4–6,26], most of them mainly focus on deeper or wider network architectural design, and the feature dependencies in spatial dimension are rarely explored, thus limiting the learning ability of the network. Thereby, a spatial graph attention (SGA) is designed to build the spatial relationships of the local features, which also can be regarded as a complementary to the channel-wise attention. The SGA encodes the feature spatial relations according to the semantic associations, enhancing the discriminative representation for better image generation.

Given an input feature $F_n^e \in \mathbb{R}^{C\times H\times W}$, which has C channels with size of $H \times W$. As shown in the green rectangle of Fig. 2b, the proposed SGA is composed by a graph convolu-

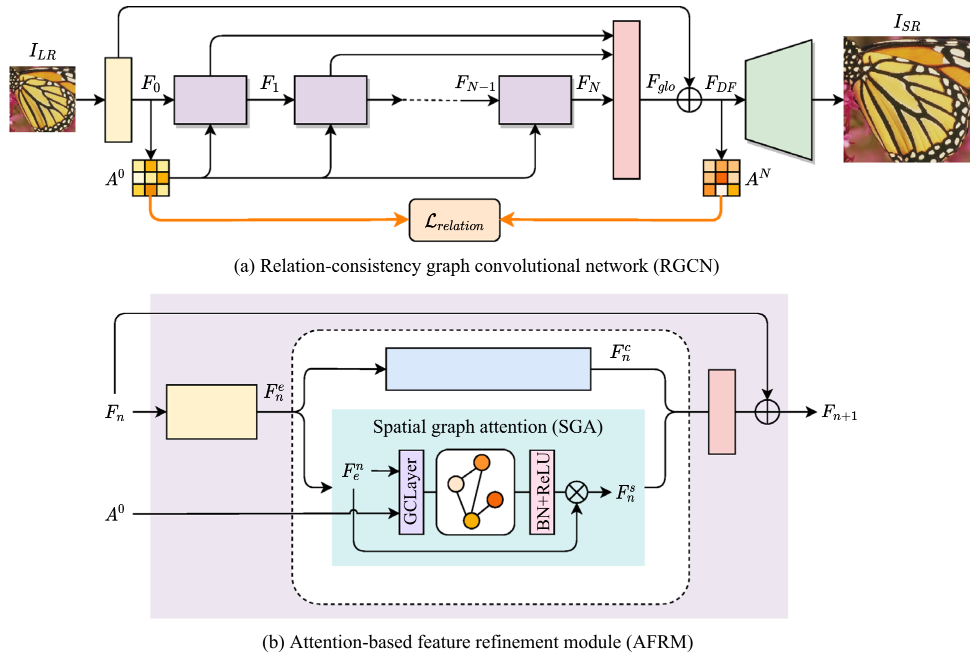


Fig. 2 **a** Architecture of relation-consistency graph convolutional network (RGCN). **b** The attention-based feature refinement module (AFRM) contains two parts: feature extraction and two-stream attention. Within two-stream attention, the proposed spatial graph attention (SGA) focuses on modeling the relationships between any two pixels, which is the core of our proposed network

tional layer and a BatchNorm and a ReLU activation function are followed. A matrix multiplication is further performed to obtain the output F_n^s

$$F_n^s = F_n^e[\sigma(BN(\mathcal{H}_{\mathrm{GC}}(F_n^e)))], \tag{6}$$

where $\sigma(\cdot)$, $BN(\cdot)$ and $\mathcal{H}_{GC}(\cdot)$ represent the function of ReLU, BatchNorm and graph convolutional layer, respectively.

3.3.1 Graph convolutional layer

Among image SR task, the standard convolution operation extracts features over the local areas via a predefined filter size (e.g., typically 3×3), while neglecting the global information of features. On the other hand, the graph convolution has been widely employed in recent works [40,42,46], which has a capability to capture the global similarity between image pixels at arbitrary areas. We thereby combine these two types of convolution and propose a graph convolutional layer (GCLayer) for feature correlations learning, as shown in Fig. 3a.

Unlike the classical convolution that operates on local Euclidean structure, graph convolution tries to learn a function $\mathcal{H}_{GC}(\cdot,\cdot)$ by defining edges $\mathcal{E}$ among nodes $\mathcal{V}$ in a global graph $\mathcal{G}$. Given a local feature $F_n^e \in \mathbb{R}^{C\times H\times W}$ and an adjacency matrix $A \in \mathbb{R}^{HW\times HW}$ that is calculated from the shallow feature F_0. We first feed feature F_n^e into a standard convolutional layer to generate feature F_{con}. Then we reshape the feature F_n^e to $\mathbb{R}^{C\times HW}$. The graph convolution operation is performed on the reshaped feature F_n^e and adjacency matrix A, a new feature F_{gra} is thus acquired by

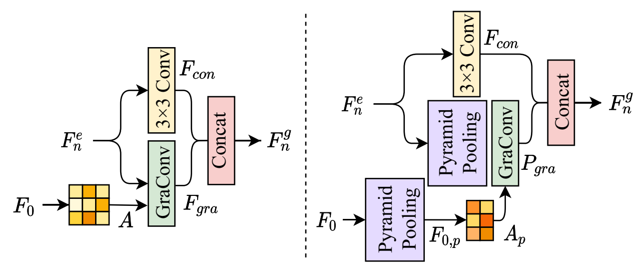


Fig. 3 Details of **a** graph convolutional layer and **b** graph convolution with an embedded pyramid pooling scheme. Compared to (**a**), the pyramid pooling is added before graph convolutional operation and adjacency matrix generation process, which decreases the computational complexity of matrix multiplication without sacrificing the overall performance

$$F_{gra} = \mathcal{H}_{GC}(F_n^e, A) = \hat{A}F_n^e W, \tag{7}$$

with

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}, \tag{8}$$

where $\tilde{A} = A + I_n$ is the adjacency matrix A of graph $\mathcal{G}$ with self loops. I_n is an identity matrix, and $\tilde{D}$ is a diagonal matrix where the element is the sum of $\tilde{A}$ in each row. W is a layer-specific trainable weight parameter.

3.3.2 Adjacency matrix

The relationship between any two pixels is characterized by the adjacency matrix A, thus enabling the generated feature F_{gra} containing the information in nonlocal areas. It is shown in [40] that GCN-based methods propagate information based on the adjacency matrix, which describes the correlations between different nodes. As a result, constructing a proper adjacency matrix is critical for GCN-based methods. Rather than using complicated nonlocal [47] to model the global relations of features, we prefer to generate A via the Gram matrix. The Gram matrix is commonly used in image neural style transfer fields to capture the summary statistics of an entire image, which can be treated as second-order statistics [48]. In our work, we calculate the Gram matrix from shallow feature F_0 as our adjacency matrix

$$A = < F_l^T, F_l >, \tag{9}$$

where A is the inner product between the feature F and its transposed feature F^T in the l-th layer. We here set $l = 0$, which represents the shallow feature F_0.

Several advantages can be brought under the above operations: 1) The Gram matrix is calculated with free learnable parameters so that it is easy to calculate and reproduce. 2) The adjacency matrix can be acquired from the arbitrary size of an input image. 3) Sharing the adjacency matrix among the network can decrease the computational burden without a performance drop, as validated in Sect. 4.

3.3.3 Pyramid pooling scheme

Since the graph convolution models the relationship between any two pixels, it requires high GPU memory occupation and expensive computational costs, especially when the image size is large. Considering this, we are concerned about whether there is an efficient way to solve this issue without sacrificing performance. By observing the computing process of graph convolution, we could clearly see that Eq. (7) has two matrix multiplications. More importantly, the former matrix multiplication dominates the computation, in which the computational complexity is $\mathcal{O}(CH^2W^2)$. We draw a conclusion that the key point of reducing the computational overhead should be on changing H and W. We thus embed a pyramid pooling scheme into graph convolutional layer.

The detailed process of pyramid pooling in a given example feature is depicted in Fig. 4, in which several pooling layers are parallel-placed to produce the pooled features with varied sizes of 1×1, 3×3, 6×6 and 8×8. The features with 8×8 are omitted for brevity.

As shown in Fig. 3b, the pyramid pooling is added before the adjacency matrix A and features F_n^e, respectively. To be specific, the shallow feature F_0 first through the pyramid pooling and generate the pooled feature $F_{0,p}$ with the size of $C \times S$, in which S is the total number of the sampled points in pyramid pooling (*i.e.,* $S = 1^2+3^2+6^2+8^2 = 110$). The shallow feature F_0 is then reshaped and transposed to $\mathbb{R}^{HW\times C}$.

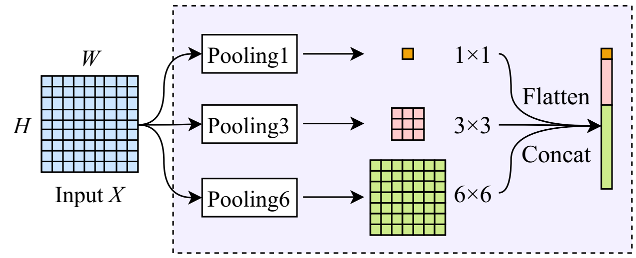


Fig. 4 Detailed process of pyramid pooling scheme. The "Pooling1," "Pooling3" and "Pooling6" represent the size of pooling layer. In our model, we set pooling size $\subseteq \{1, 3, 6, 8\}$. The pooling size of 8 is omitted for brevity

The Eq. (9) is performed to calculate the adjacency matrix $A_p \in \mathbb{R}^{HW\times S}$. Similarly, the feature F_n^e is feed to the pyramid pooling, obtaining the pooled result $F_{n,p}^e \in \mathbb{R}^{S\times HW}$. Thus, the formulation of Eq. (7) is rewritten as

$$P_{gra} = \mathcal{H}_{GC}(F_{n,p}^e, A_p) = \hat{A}_p F_{n,p}^e W, \tag{10}$$

where the output $P_{gra} \in \mathbb{R}^{C\times H\times W}$ is kept the same size as F_{gra} in Eq. (7).

By virtue of the spatial pyramid pooling, the computational complexity of the former matrix multiplication in Eq. (7) is decreased to $\mathcal{O}(CHWS)$, lower than the original $\mathcal{O}(CH^2W^2)$. In addition to reducing the computations, the spatial pyramid pooling is also parameter-free. Consequently, the pyramid pooling efficiently lowers the computational overhead and maintains the overall performance simultaneously, as demonstrated in Sect. 4.

3.4 Relation-consistency loss

The pixel-wise loss (e.g., $\mathcal{L}_1$ loss) is generally used in most CNN-based SISR methods, which aims to minimize the distance between the super-resolved result I_{SR} and the ground-truth image I_{HR}. Although such loss assists the networks to gain higher performance, it only measures the discrepancy on an entire image at pixel level, and the difference in semantic level is rarely considered, thus resulting in poor visual-quality on image details.

Moreover, as the SISR is an image-to-image task, the semantic relation between input LR image and reconstructed SR image is similar. Ideally, according to the spatially invariant of the global relations, the obtained features from different levels share the similar contextual relations in the training process. For example, when super-resolving a "face" image, the features from one "eye" should be highly related to the other one and be less correlated with the features from the "nose." This kind of feature dependency does not easily change and is independent of distance, since it is an inherent characteristic of the image, which can be regarded as prior knowledge of the image. Besides modeling the spatial correlation by SGA, the feature relation coherence throughout the entire network also needs to be considered for generating visual pleasing images. To achieve this, we propose a relation-consistency loss to enhance the visual quality by minimizing the discrepancy in semantic level.

As described previously in Sect. 3.3, the Gram matrix in SGA captures global statistics across the entire image. We thereby implement the relation-consistency loss by Gram matrix. The proposed relation-consistency loss tries to encourage the spatial relations to be consistent among different layers, the selection of features from a specific layer thus seems to be a key point. In Table 1, the comparative experiments are conducted that generate Gram matrix from various layers of the network. It can be seen from the experimental results that there is no apparent improvement in frequently calculating Gram matrix from different layers compared to the Gram matrix only generated from low-level feature. This phenomenon indicates the property of contextual relation consistency in an image, which also exactly validates the motivation of our proposed relation-consistency loss. Based on the experimental findings, we utilize the relation-consistency loss to give a constraint between low-level feature and high-level feature, the loss function can be formally given as

$$\mathcal{L}_{relation} = \left\| A_p^0 - A_p^N \right\|_1, \tag{11}$$

where A_p^0 and A_p^N denote the corresponding adjacency matrix of feature F_0 and F_N, respectively.

As a consequence, the relation-consistency loss encourages the network to reconstruct a more realistic image by maintaining the contextual relation consistency between low-level feature and high-level feature.

3.5 Implementation details

3.5.1 Full objective

Similar to [6,8,35], we employ $\mathcal{L}_1$ loss to optimize the proposed network via minimizing the difference between the reconstructed image I_{SR} and the ground truth image I_{HR}. Given a training dataset with M image pairs $\{I_m^{LR}, I_m^{HR}\}_{i=1}^{M}$, the reconstruction loss is represented as

$$\mathcal{L}_{\text{rec}} = \left\| \mathcal{H}_{\text{RGCN}}(I_{\text{LR}}^m; \theta) - I_{\text{HR}}^m \right\|_1, \tag{12}$$

where $\mathcal{H}_{\text{RGCN}}(\cdot)$ is the function of our proposed RGCN. The total loss function is expressed by

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rec}} + \lambda \mathcal{L}_{\text{relation}}. \tag{13}$$

where λ is the hyperparameter to control the weights of different losses. The performance of RGCN with different losses is compared in Sect. 4, which verifies the importance of each loss.

3.5.2 Training details

We set the AFRMM number as $N = 20$ in our proposed SR network, and each AFRM has 64 filters (*i.e.,* C=64). Within AFRM, we use 3×3 and 5×5 convolutional layers in feature extraction. For channel-wise attention in two-stream attention, we adopt 1×1 convolutional layer with reduction ratio $r = 16$, which is as similar as [35]. And the hyperparameter λ in the loss function is set as 1, this setting brings a more stable training process and better results.

4 Experiments

4.1 Settings

4.1.1 Datasets and metrics

Timofte [49] has released a high-quality dataset DIV2K for image restoration tasks, which contains 800 training images, 100 validation images and 100 test images. Following [6,8,12], we use DIV2K dataset as our training set. For testing stage, we evaluate our SR model on five benchmark datasets: Set5 [50], Set14 [51] BSD100 [52], Urban100 [20], and Manga109 [53]. All the SR results are evaluated with peak signal to noise ratio (PSNR) and the structural similarity index (SSIM) [54] metrics on Y channel (*i.e.,* luminance) of transformed YCbCr space.

4.1.2 Degradation models

To fully demonstrate the effectiveness of our proposed RGCN, three degradation models are used to simulate LR images. The first one is bicubic downsampling by adopting the Matlab function *imresize* with the option *bicubic* (denoted as BI for short). Similar to [8,55], the second one is to blur HR image by Gaussian kernel size 7×7 with standard deviation 1.6. The blurred image is then downsampled with a scaling factor $\times3$ (denoted as BD for short). We finally produce LR images in a more challenging way. The HR image is first downscaled by bicubic with scaling factor $\times3$, and then we add Gaussian noise with noise level 30 on the downsampled HR image (denoted as DN for short).

4.1.3 Training details

During training, data augmentation is performed by flipping horizontally and rotating 90°, 180° and 270°. In each batch, 16 LR image patches with size of 48×48 are extracted as inputs for image SR, and 1,000 iterations of back-propagation

Table 1 The impact of adjacency matrix A computed from different locations

Description	Inference time(s)	#Param(M)	PSNR(dB)
A from each lLayer	0.9042	14.5	32.38
A from feature F_0	0.7820	14.5	32.41

The results are reported on Set5 with scaling factor $\times 4$

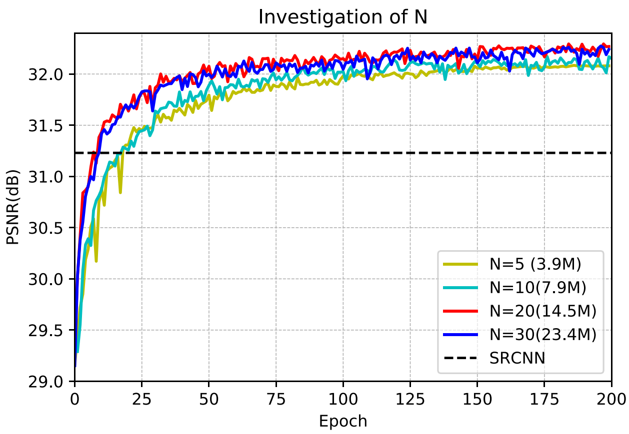


Fig. 5 Convergence analysis of RGCN with different number (N) of attention-based feature refinement modules (AFRM). The performance curves are plotted on DIV2K with scaling factor $\times 4$ in 200 epochs

Table 2 The impact of different scale of pooling in SGA on Set5 with scaling factor $\times 4$

Scale	FLOPs(G)	Memory(MB)	PSNR(dB)
Baseline	49.75	8.682	32.43
Pooling (1)	18.50	5.987	32.32
Pooling (3)	18.62	5.987	32.35
Pooling (6)	19.03	5.987	32.36
Pooling (8)	19.44	5.992	32.39
Pooling (24)	26.93	6.034	32.38
Pooling (1368)	32.69	6.047	32.41

constitute an epoch. Our model is trained by AdamW optimizer [56] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The cosine learning rate scheduler [57] is adopted by initializing the learning rate as 10^{-4}. We use the PyTorch framework to implement our model with Titan V GPUs.

4.2 Ablation study

In this section, we investigate the effectiveness of different components in our proposed method, including attention-based feature refinement module (AFRM), pyramid pooling scheme and relation-consistency loss. All the comparative experiments are trained on DIV2K with scaling factor $\times 4$ in 200 epochs and further tested on Set5.

4.2.1 Number of *N*

We first investigate the basic parameter in our network: the number of AFRM (denoted as N for short), which is directly related to the model size and overall performance. For a clarity comparison, the performance of SRCNN [3] is set as a reference. The convergence curves of AFRM with different numbers of N are presented in Fig. 5. From the results, we observe that RGCN with larger N (*i.e.,* N=20 and N=30) obtain better performance, mainly because the network goes deeper with more AFRMs stacking. Instead, RGCN with smaller N (*i.e.,* N=5 and N=10) suffers some performance drop but still outperforms SRCNN. More importantly, when increasing the number of N from 20 to 30, the capacity of the network goes larger (14.5 M $\rightarrow$ 23.4 M) without an obvious performance improvement. To better trade-off performance and model size, we thus adopt N=20 as our final RGCN model.

4.2.2 Graph convolutional layer

In order to evaluate the efficiency of graph convolutional layer in SGA, we conduct some comparative experiments, including the impact of sharing adjacency matrix and the embedded pyramid pooling scheme, respectively.

Adjacency Matrix: In our network, the adjacency matrix in graph convolution is calculated from the low-level feature F_0 and is shared across the whole network. To verify the efficacy of these strategies, we perform a comparative experiment on computing adjacency matrix from different locations. Two networks are introduced: calculate the adjacency matrix from feature F_0 and from each output of AFRM, respectively. Table 1 shows that computing the adjacency matrix from each output of AFRM is time-consuming with no obvious performance gain. We can draw a conclusion that frequently updating the adjacency matrix does not lead to higher performance, instead, reusing the adjacency matrix performs better. This intriguing finding could be due to the contextual relation of image is consistent and spatially invariant, which exactly verifies the motivation of our proposed relation-consistency loss in our RGCN. Considering the balance between efficacy and efficiency, we finally opt to calculate the adjacency matrix from a shallow feature and share it across the whole network to ensure the consistency of feature spatial information.

Table 3 Comparative results achieved by our RGCN trained with different losses for scaling factor ×4 in 200 epochs

Method	$\mathcal{L}_{rec}$	$\mathcal{L}_{relation}$	PSNR(dB)
RGCN	✓		32.28
	✓	✓	32.41 (+0.13)

Pyramid pooling scheme: As discussed in Sect. 3.3, the adoption of pyramid pooling aims to reduce the computational overhead from the vanilla graph convolution. We thus give a quantitative comparison to validate its contribution via the following metrics: FLOPs,[1] GPU memory usage[2] and the performance in PSNR, which are evaluated on Set5 with a 48×48 input image patch. As shown in Table 2, we compare several settings of pyramid pooling, including pooling with one single scale and multiple scales. The graph convolutional layer without pooling is set as a baseline.

In detail, when using single scale pooling in SGA (e.g., pooling(3)), the FLOPs and GPU memory occupation are reduced with 31.25 G and 2,695MB. When we further utilize multiple sizes of pooling (e.g., pooling(1368)), we obtain comparative performance with baseline, and decrease the computational overhead and GPU memory simultaneously. Moreover, when adopting a large size of pooling (e.g., pooling(24)), although the computational resources are effectively reduced, the performance gets worse. Thus, four-scales pooling is selected into SGA for decreasing computational resources.

4.2.3 Relation-consistency loss

Following, we conduct ablation experiments to verify the effectiveness of the proposed relation-consistency loss for training our RGCN. It can be observed from Table 3 that the PSNR values of our RGCN decreases from 32.41dB to 32.28dB if the network is trained without relation-consistency loss. This is mainly because, with reconstruction loss, the network only learns to optimize the difference between the generated output and the ground truth in pixel level, while neglecting the relations correspondence of high-level and low-level features in a global view. When further employing relation-consistency loss into our network, better performance is achieved with 0.13dB increased.

[1] We evaluate the theoretical amount of multiply-add operations, which referred to the work [58].

[2] We evaluate the GPU memory usage on Titan V with PyTorch 1.2.0 and CUDA 10.1.

Table 4 Comparative results of different arrangement of CA and SGA on Set5 with an upscaling factor ×4

Arrangement	Sequential		Parallel
	CA-SGA	SGA-CA	SGA+CA
PSNR(dB)	32.37	32.38	32.41
Param(M)	14.3	14.3	14.5

Table 5 Comparisons with other spatial attention on Set5 with scaling factor ×4

Component	PSNR(dB)	#Param(M)
CA (Baseline)	32.35	13.8
+ Nonlocal [47]	32.36	14.2
+ CBAM [34]	32.38	14.1
+ SGA	32.41	14.5

4.2.4 Stream of CA and SGA

Since CA and SGA have different functions in our network, the placement of them affects the overall performance. We here explore the influence of different arrangements (*i.e.,* parallel and sequential) between CA and SGA. As shown in Table 4, it is clear that the parallel arrangement of CA and SGA infers better representations (PSNR = 32.41dB) than doing sequential and brings only 0.2 M parameters increased, which is brought by the 1×1 convolutional layer for the feature fusion. Therefore, utilizing both CA and SGA is crucial while the best-arranging strategy further pushes the overall performance.

4.2.5 Comparisons with other spatial attention methods

In order to evaluate our spatial graph attention (SGA) effectively, we conduct comparative experiments with two related attention methods: nonlocal [47] and spatial attention in CBAM [34]. These two attention methods are used to replace our SGA, and the network only contains the channel-wise attention set as the baseline. Training and testing settings are kept the same as our RGCN for a fair comparison. The results are listed in Table 5. One can clearly see that all methods with spatial attention achieve higher performance over the baseline, which indicates their effectiveness for image SR. Compared with equipping the two well-known spatial attention methods (+ Nonlocal and + CBAM), the performance of our network (+ SGA) is increased by 0.05dB and 0.03dB with 0.3 M and 0.4 M extra parameters.

4.2.6 Visualization of local attribution maps

Recently, Gu et al. incorporated attribution analysis into image SR methods and proposed a novel attribution approach

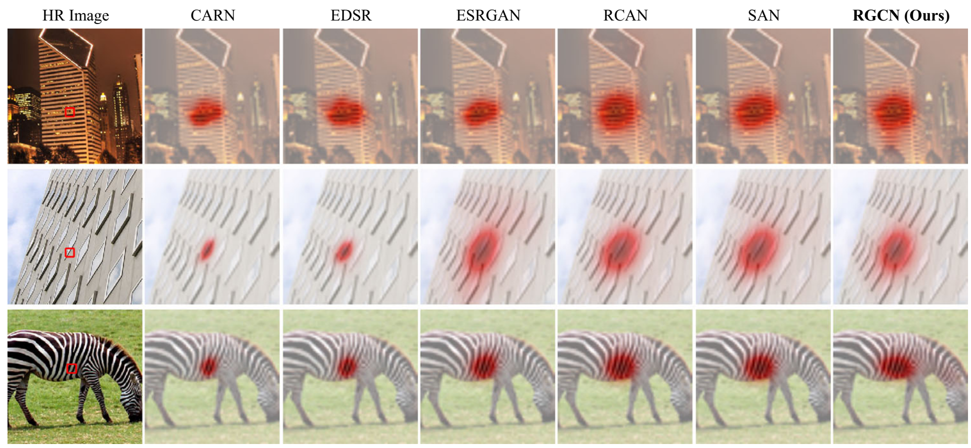


Fig. 6 Visualization results of the LAM on different SR approaches with scaling factor ×4. The LAM illustrates the contribution of each pixel in the selected image patch (the red box in HR image). The larger the red area, the more pixels are utilized in feature extraction

Table 6 Investigation of each component in RGCN

Case index	Baseline	1	2	3	4	5	6
FE	✓	✓	✓	✓	✓	✓	✓
CA		✓			✓		✓
SGA			✓		✓	✓	✓
GFF				✓		✓	✓
PSNR(dB)	32.28	32.34	32.36	32.32	32.38	32.39	32.41

We observe the best result (PSNR) on Set5 with scaling factor ×4 in 200 epochs

Table 7 Quantitative results with scaling factor ×2 on BI degradation model

Type	Method	Scale	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
CNN-based	Bicubic	×2	33.66/0.9299	30.24/0.8688	29.56/0.8431	26.88/0.8403	30.80/0.9339
	SRCNN (2014)	×2	36.66/0.9542	32.45/0.9067	31.36/0.8879	29.50/0.8946	35.60/0.9663
	VDSR (2016)	×2	37.53/0.9590	33.05/0.9130	31.90/0.8960	30.77/0.9140	37.22/0.9750
	EDSR (2017)	×2	38.11/0.9602	33.92/0.9195	32.32/0.9013	32.93/0.9351	39.10/0.9773
	RCAN (2018)	×2	38.27/0.9614	34.11/0.9216	32.41/0.9026	33.34/0.9385	39.43/0.9786
	NLRN (2018)	×2	38.00/0.9603	33.46/0.9159	32.19/0.8992	31.81/0.9246	–/–
	SRFBN(2019)	×2	38.11/0.9609	33.82/0.9196	32.29/0.9010	32.62/0.9328	39.08/0.9779
	SAN (2019)	×2	**38.31/0.9620**	34.07/0.9213	32.42/0.9028	33.10/0.9370	39.32/0.9792
	RDN (2020)	×2	38.24/0.9614	34.01/0.9212	32.34/0.9017	32.89/0.9353	39.18/0.9780
	USRNet (2020)	×2	37.77/0.9592	33.49/0.9156	32.10/0.8981	31.79/0.9255	38.37/0.9760
	HAN (2020)	×2	38.27/0.9614	**34.16/0.9217**	32.41/0.9027	**33.35/0.9385**	**39.46/0.9785**
	SRGAT (2021)	×2	38.20/0.9610	33.93/0.9201	32.34/0.9014	32.90/0.9359	39.30/0.9785
	RGCN (Ours)	×2	38.30/0.9616	34.10/0.9213	**32.44/0.9030**	33.15/0.9377	39.38/0.9784
Transformer-based	SCET (2022)	×2	38.06/0.9615	33.78/0.9198	32.24/0.9006	32.38/0.9299	**39.86/0.9821**
	SwinIR (2021)	×2	**38.35/0.9620**	**34.14/0.9215**	**32.44/0.9030**	**33.40/0.9393**	39.60/0.9792
	RGCN (Ours)	×2	38.30/0.9616	34.10/0.9213	**32.44/0.9030**	33.15/0.9377	39.38/0.9784

The best results are **highlighted**

Table 8 Quantitative results with scaling factor ×3 on BI degradation model

Type	Method	Scale	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
CNN-based	Bicubic	×3	30.39/0.8682	27.55/0.7742	27.21/0.7385	24.46/0.7349	26.95/0.8556
	SRCNN (2014)	×3	32.75/0.9090	29.30/0.8215	28.41/0.7863	26.24/0.7989	30.48/0.9117
	VDSR (2016)	×3	33.67/0.9210	29.78/0.8320	28.83/0.7990	27.14/0.8290	32.01/0.9340
	EDSR (2017)	×3	34.65/0.9280	30.52/0.8462	29.25/0.8093	28.80/0.8653	34.17/0.9476
	RCAN (2018)	×3	34.74/0.9299	30.65/0.8482	29.32/0.8111	29.09/0.8702	34.44/0.9499
	NLRN (2018)	×3	34.27/0.9266	30.16/0.8374	29.06/0.8026	27.93/0.8453	–/–
	SRFBN (2019)	×3	34.70/0.9292	30.51/0.8461	29.24/0.8084	28.73/0.8641	34.18/0.9481
	SAN (2019)	×3	34.75/0.9300	30.59/0.8476	29.33/0.8112	28.93/0.8671	34.30/0.9494
	RDN(2020)	×3	34.71/0.9296	30.57/0.8468	29.26/0.8093	28.80/0.8653	34.13/0.9484
	USRNet (2020)	×3	34.43/0.9279	30.51/0.8446	29.18/0.8076	28.38/0.8575	34.05/0.9466
	HAN (2020)	×3	34.75/0.9299	**30.67**/0.8483	29.32/0.8110	**29.10/0.8705**	**34.48**/0.9500
	SRGAT (2021)	×3	34.75/0.9297	30.63/0.8474	29.29/0.8099	28.90/0.8666	34.42/0.9495
	RGCN (Ours)	×3	**34.77/0.9301**	**30.67/0.8486**	**29.33/0.8114**	28.99/0.8679	34.47/**0.9501**
Transformer-based	SCET (2022)	×3	34.53/0.9278	30.43/0.8441	29.17/0.8075	28.38/0.8559	34.29/0.9503
	ESRT (2022)	×3	34.42/0.9268	30.43/0.8433	29.15/0.8063	28.46/0.8574	33.95/0.9455
	LBNet (2022)	×3	34.47/0.9277	30.38/0.8417	29.13/0.8061	28.42/0.8559	33.82/0.9460
	SwinIR (2021)	×3	**34.89/0.9312**	**30.77/0.8503**	**29.37/0.8124**	**29.29/0.8744**	**34.74/0.9518**
	RGCN (Ours)	×3	34.77/0.9301	30.67/0.8486	29.33/0.8114	28.99/0.8679	34.47/0.9501

The best results are **highlighted**

Table 9 Quantitative results with scaling factor ×4 on BI degradation model

Type	Method	Scale	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
CNN-based	Bicubic	×4	28.42/0.8104	26.00/0.7027	25.96/0.6675	23.14/0.6577	24.89/0.7866
	SRCNN (2014)	×4	30.48/0.8628	27.50/0.7513	26.90/0.7101	25.52/0.7221	27.58/0.8555
	VDSR (2016)	×4	31.35/0.8830	28.02/0.7680	27.29/0.0726	25.18/0.7540	28.83/0.8870
	EDSR (2017)	×4	32.46/0.8968	28.80/0.7876	27.71/0.7420	26.64/0.8033	31.02/0.9148
	RCAN (2018)	×4	32.63/0.9002	28.87/0.7889	27.77/0.7436	26.82/0.8087	31.22/0.9173
	NLRN (2018)	×4	31.92/0.8916	28.36/0.7745	27.48/0.7346	25.79/0.7729	–/–
	SRFBN (2019)	×4	32.47/0.8983	28.81/0.7868	27.72/0.7409	26.60/0.8015	31.15/0.9160
	SAN (2019)	×4	32.64/0.9003	**28.92**/0.7888	27.78/0.7436	26.79/0.8068	31.18/0.9169
	RDN (2020)	×4	32.47/0.8990	28.81/0.7871	27.72/0.7419	26.61/0.8028	31.00/0.9151
	USRNet (2020)	×4	32.42/0.8978	28.83/0.7871	27.69/0.7404	26.44/0.7976	31.11/0.9154
	HAN (2020)	×4	32.64/0.9002	28.90/0.7890	**27.80/0.7442**	**26.85**/0.8094	**31.42/0.9177**
	SRGAT (2021)	×4	32.57/0.8997	28.86/0.7879	27.77/0.7421	26.76/0.8052	31.41/0.9181
	RGCN (Ours)	×4	**32.65/0.9005**	28.91/**0.7892**	27.79/0.7440	**26.85/0.8089**	31.24/0.9176
Transformer-based	SCET (2022)	×4	32.27/0.8963	28.72/0.7847	27.67/0.7390	26.33/0.7915	31.10/0.9155
	ESRT (2022)	×4	32.19/0.8947	28.69/0.7833	27.69/0.7379	26.39/0.7962	30.75/0.9100
	LBNet (2022)	×4	32.29/0.8960	28.68/0.7832	27.62/0.7382	26.27/0.7906	30.76/0.9111
	SwinIR (2021)	×4	**32.72/0.9021**	**28.94/0.7914**	**27.83/0.7459**	**27.07/0.8164**	**31.67/0.9226**
	RGCN (Ours)	×4	32.65/0.9005	28.91/0.7892	27.79/0.7440	26.85/0.8089	31.24/0.9176

The best results are **highlighted**

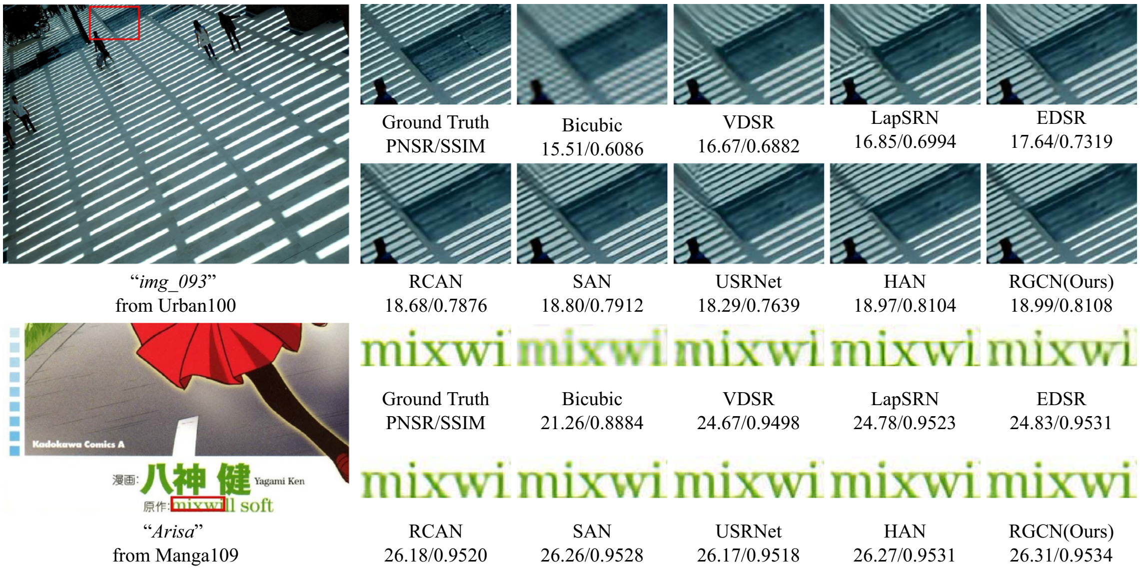


Fig. 7 Visualization comparison of BI degradation on Urban100 and Manga100 with scaling factor ×4

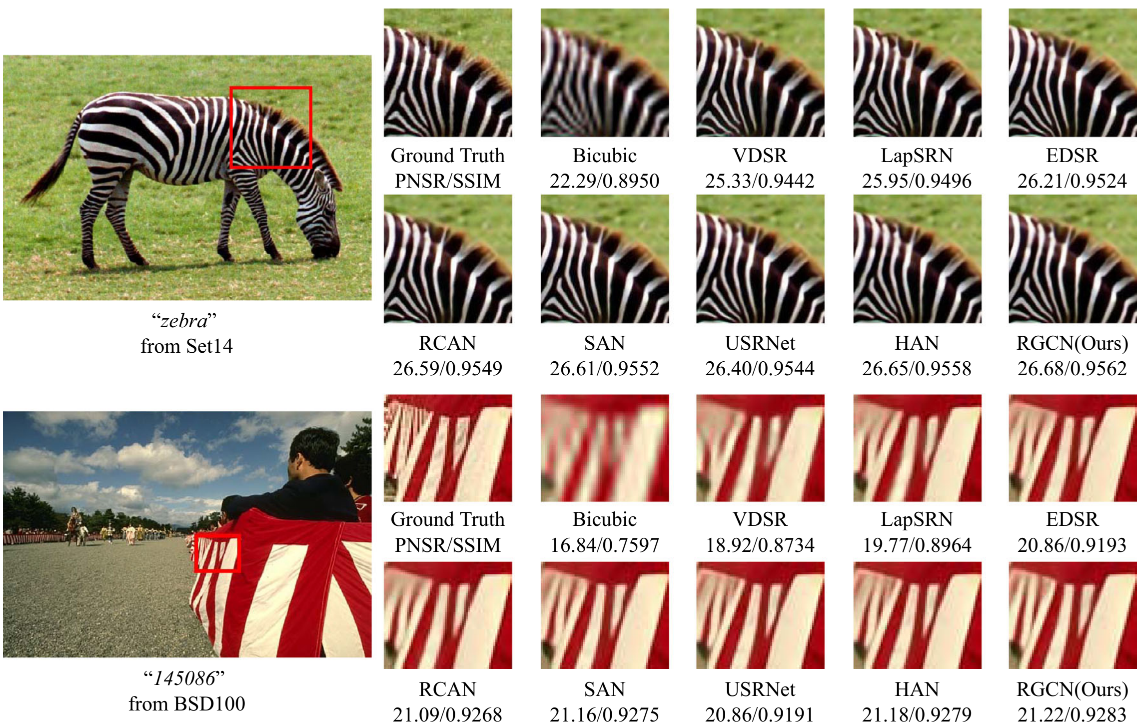


Fig. 8 Visualization comparison of BI degradation on Set14 and BSD100 with scaling factor ×4

called local attribution map (LAM) [59]. The goal of LAM is to find the input features that strongly influence the network outputs, which visualize the results via LAM.

Figure 6 shows the results of LAM in some representative image SR approaches, including CARN [60], EDSR [26], ESRGAN [61], RCAN [35] and SAN [12]. As can be seen, our RGCN involves more pixels with larger receptive fields while CARN and EDSR only extracts few information under a limited region. It is implied that our model could capture long-range feature dependencies for enriching the representational ability of the network, which generates better super-resolved image.

4.2.7 Other components

As stated in Sect. 3, our RGCN mainly contains channel-wise attention (CA), spatial graph attention (SGA) and global feature fusion (GFF). We perform various combinations to verify the effectiveness of each component. *Baseline* refers to the network only containing feature extraction with 3×3 and 5×5 convolutional layers, which has a similar size as our RGCN to ensure a fair comparison. As shown in Table 6, the baseline achieves relatively low performance, indicating that blindly stacking more layers cannot lead to better performance. When adding CA, SGA and GFF individually to the baseline, resulting in *Case1*, *Case2* and *Case3*, each of them improves the overall performance efficiently.

We further equip CA and SGA simultaneously, leading to *Case4*, the method obtains consistently improvement with 0.04dB and 0.02dB gain as compared with *Case1* and *Case2*. Similar phenomena can be found in using GFF to form our final network *Case6*, and the overall performance is boosted from 32.39dB to 32.41dB.

4.3 Results with BI degradation

Simulating LR image with a bicubic degradation (BI) model is widely used in image SR settings. For BI degradation model, we compare our RGCN with 16 state-of-the-art SISR methods: SRCNN [3], VDSR [5], EDSR [26], NLRN [62], RCAN [35], RDN [8], SRFBN [6], SAN [12], USRNet [17], HAN [13], SRGAT [16], SCET [63], ESRT [64], LBNet [65] SwinIR [14]. All the quantitative results for three scaling factors over five benchmark are reported in Tables 7, 8 and 9.

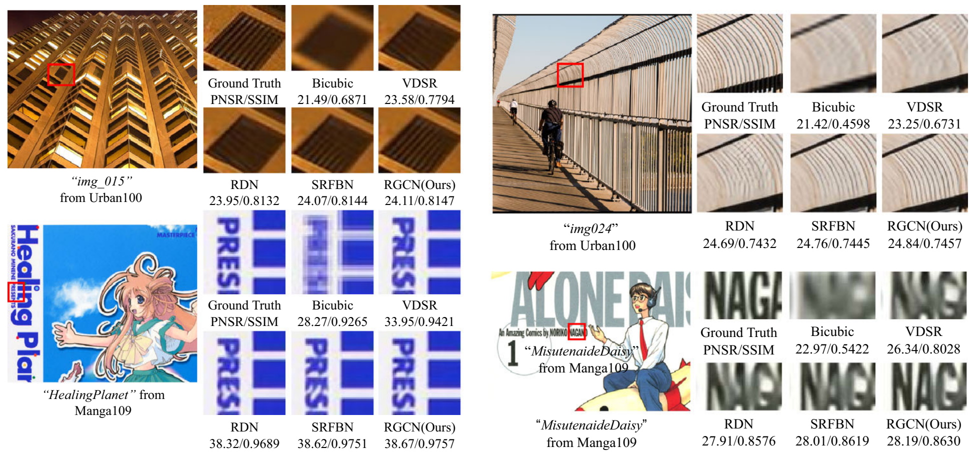
Compared with the CNN-based SR methods, our RGCN achieves the best performance on most benchmark datasets for all scaling factors. Note that on Set14 for scaling factor ×3, our RGCN and HAN [13] both obtain the best PSNR while our SSIM value is higher than HAN, which indicates our method can reconstruct better result, the same phenomenon can be found in Urban100 with scaling factor ×4. However, all these CNN-based methods perform worse than the Transformer-based image SR approaches, demonstrating the strong representation ability of the Transformer. Although our method obtains superior performance to most CNN-based methods, we have a large margin with the Transformer-based approaches. In future work, this will be considered to improve our method by combining graph convolutional and Transformer.

We also present the visual comparisons on different benchmark datasets with scaling factor ×4 in Figs. 7 and 8, respectively. As shown in Fig. 7, the “img_09” from Urban100 has an amount of structured texture. Some CNN-based SR methods cannot recover clearer edges and fine details from the LR images, such as VDSR [5] and LapSRN [66]. And the methods employed $\mathcal{L}_1$ loss ((e.g., RCAN [35] and SAN [12])

Table 10 Quantitative results with BI and DN degradation models

Dataset	Model	Bicubic	SRCNN [3]	VDSR [4]	IRCNN_G [67]	IRCNN_C [67]	SRMD [55]	RDN [8]	SRFBN [6]	RGCN (Ours)
Set5	BD	28.34/0.8161	31.63/0.8888	33.30/0.9159	33.38/0.9182	29.55/0.8246	34.09/0.9242	34.57/0.9280	34.66/0.9283	**34.69/0.9286**
	DN	24.14/0.5445	27.16/0.7672	27.72/0.7872	24.85/0.7205	26.18/0.7430	27.74/0.8026	28.46/0.8151	28.53/0.8182	**28.59/0.8194**
Set14	BD	26.12/0.7106	28.52/0.7924	29.67/0.8269	29.73/0.8292	27.33/0.7135	30.11/0.8364	30.53/0.8447	30.48/0.8439	**30.58/0.8457**
	DN	23.14/0.4828	25.49/0.6580	25.92/0.6786	23.84/0.6091	24.68/0.6300	26.13/0.6974	26.60/0.7101	26.60/0.7144	**26.71/0.7167**
BSD100	BD	26.02/0.6733	27.76/0.7526	28.63/0.7903	28.65/0.7922	26.46/0.6572	28.98/0.8009	29.23/0.8079	29.21/0.8069	**29.33/0.8102**
	DN	22.94/0.4461	25.11/0.6151	25.52/0.6345	23.89/0.5688	24.52/0.5850	25.64/0.6495	25.93/0.6573	25.95/0.6625	**26.00/0.6637**
Urban100	BD	23.20/0.6661	25.31/0.7612	26.75/0.8145	26.77/0.8154	24.89/0.7172	27.50/0.8370	28.46/0.8581	28.48/0.8581	**28.65/0.8614**
	DN	21.63/0.4701	23.32/0.6500	23.83/0.6797	21.96/0.6018	22.63/0.6205	24.28/0.7092	24.92/0.7362	24.99/0.7424	**25.25/0.7525**
Manga109	BD	25.03/0.7987	28.79/0.8851	31.66/0.9260	31.15/0.9245	28.68/0.8574	32.97/0.9391	33.97/0.9465	34.07/0.9466	**34.24/0.9484**
	DN	23.08/0.5448	25.78/0.7889	26.41/0.8130	23.18/0.7466	24.74/0.7701	26.72/0.8424	28.00/0.8590	28.02/0.8618	**28.25/0.8669**

The best results are **highlighted**



(a) Visual comparisons of BD degradation (b) Visual comparisons of DN degradation

Fig. 9 Visual comparisons of **a** BD degradation and **b** DN degradation with scaling factor $\times 3$

Table 11 Model size, inference times and performance compare results on Set5 with upscaling factor $\times 4$

	EDSR [26]	RCAN [35]	HAN [13]	RDN [8]	SAN [12]	SRGAT [16]	SwinIR [14]	RGCN(Ours)
Param(M)	43	16	17.4	22.1	15.6	6.6	13.8	14.3
PSNR(dB)	32.46	32.63	32.64	32.47	32.64	32.57	32.72	32.65
Time(s)	1.64	1.15	1.34	1.56	1.49	1.21	1.39	1.42

generate over-smoothed results and less fine image details. In contrast, our RGCN reconstructs the HR result with clear structure information and textural details, such as the lines of floor.

4.4 Results with BD and DN degradations

Following [8,55], we also conduct comparisons on more challenging degradations: BD and DN. Our RGCN is compared with SRCNN [3], IRCNN_C [67], IRCNN_G [67], SRMD [55], RDN [8] and SRFBN [6]. All the results on $\times 3$ are listed in Table 10, from which we can observe that our network achieves better performance on all datasets. For quantity comparisons, we show the super-resolved results with BD degradation in Fig. 9a. One can see that, for BD degradation, most compared methods recover blurring artificial. Instead, our RGCN suppresses the blurs and recovers texture information. And for the DN degradation model, it can be found that our network removes the noise of corrupted images and recovers more details compared to other methods, as shown in Fig. 9b. These comparative results of BD and DN degradation models demonstrate that our network can be well-adapted to multiple degradation models.

4.5 Model size and inference time

Table 11 shows the comparison results of performance, inference time and model size. PSNR results and inference time are evaluated on Set5 with upscaling factor $\times 4$. Our RGCN outperforms CNN-based image SR networks (e.g., SAN [12], RCAN [35] and HAN [13]) in terms of performance and model parameters with faster inference times. Despite SRGAT [16] being much smaller than RGCN, the performance is still underperformed. Moreover, the PSNR value of RGCN is slightly lower than that of the Transformer-based method; however, our model has a comparable model size and costs less inference time than SwinIR [14]. Thus, the experimental result provides an implication that our RGCN has a good balance on model size and performance.

5 Conclusion

In this paper, we propose a relation-consistency graph convolutional network (RGCN) for accurate image SR, which captures contextual information in the spatial dimension. To be specific, we utilize a spatial graph attention (SGA) to dynamically model global dependencies via graph convolutional. To draw the pairwise relationships of image features, the Gram matrix is adopted to calculate the adjacency matrix in SGA, and then share it across the entire network. We further embed a pyramid pooling scheme in SGA to reduce expensive computational costs and memory occupation without sacrificing the overall performance. Additionally, a relation-consistency loss is introduced, which gives a constraint on spatial relationships between low-level feature and high-level feature in the semantic level. Extensive experiments demonstrate the superiority of our RGCN in terms of quantitative and visual quality.

Acknowledgements This work was supported by the National Natural Science Foundation of China under Grant (No.61672421) and Ministry of Education of the People's Republic of China (No. 2020KJ010801).

Declarations

Conflict of interest Authors declare that they have no conflicts of interest.

References

1. Zhang, L., Wu, X.: An edge-guided image interpolation algorithm via directional filtering and data fusion. IEEE Trans. Image Process. **15**, 2226–2238 (2006)
2. Zhang, Y., Fan, Q., Bao, F., Liu, Y., Zhang, C.: Single-image super-resolution based on rational fractal interpolation. IEEE Trans. Image Process. **27**, 3782–3797 (2018)
3. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: European Conference on Computer Vision (ECCV), pp. 184–199 (2014)
4. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. In: European Conference on Computer Vision (ECCV), pp. 391–407 (2016)
5. Kim, J., Lee, J., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1646–1654 (2016)
6. Li, Z., Yang, J., Liu, Z., Yang, X., Jeon, G., Wu, W.: Feedback network for image super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3862–3871 (2019)
7. Ma, T., Tian, W.: Back-projection-based progressive growing generative adversarial network for single image super-resolution. Vis. Comput. **37**, 925–938 (2020)
8. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image restoration. IEEE Trans. Pattern Anal. Mach. Intell. **43**, 2480–2495 (2020)
9. Shi, W., Du, H., Mei, W., Ma, Z.: (sarn)spatial-wise attention residual network for image super-resolution. Vis. Comput. **37**, 1569–1580 (2020)
10. Yang, X., Zhu, Y., Guo, Y., Zhou, D.: An image super-resolution network based on multi-scale convolution fusion. The Visual Computer, 1–11 (2021)
11. Mei, Y., Fan, Y., Zhou, Y.: Image super-resolution with non-local sparse attention. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3517–3526 (2021)
12. Dai, T., Cai, J., Zhang, Y.-B., Xia, S., Zhang, L.: Second-order attention network for single image super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11057–11066 (2019)
13. Niu, B., Wen, W., Ren, W., Zhang, X., Yang, L., Wang, S., Zhang, K., Cao, X., Shen, H.: Single image super-resolution via a holistic attention network. In: European Conference on Computer Vision (ECCV), pp. 191–207 (2020)
14. Liang, J., Cao, J., Sun, G., Zhang, K., Gool, L.V., Timofte, R.: Swinir: Image restoration using swin transformer. IEEE International Conference on Computer Vision Workshops (ICCVW), 1833–1844 (2021)
15. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Adv. Neural Inf. Process. Syst. **30** (2017)
16. Yan, Y., Ren, W., Hu, X., Li, K., Shen, H., Cao, X.: Srgat: Single image super-resolution with graph attention network. IEEE Trans. Image Process. **30**, 4905–4918 (2021)
17. Zhang, K., Gool, L.V., Timofte, R.: Deep unfolding network for image super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3217–3226 (2020)
18. Fang, F., Li, J., Zeng, T.: Soft-edge assisted network for single image super-resolution. IEEE Trans. Image Process. **29**, 4656–4668 (2020)
19. Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 349–356 (2009)
20. Huang, J.-B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5197–5206 (2015)
21. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. ArXiv arXiv:1603.08155 (2016)
22. Kim, J., Lee, J., Lee, K.M.: Deeply-recursive convolutional network for image super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1637–1645 (2016)
23. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)
24. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2790–2798 (2017)
25. Tai, Y., Yang, J., Liu, X., Xu, C.: Memnet: A persistent memory network for image restoration. In: IEEE International Conference on Computer Vision (ICCV), pp. 4549–4557 (2017)
26. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPWR), 1132–1140 (2017)
27. Vassilo, K., Heatwole, C., Taha, T., Mehmood, A.: Multi-step reinforcement learning for single image super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 512–513 (2020)
28. Niu, Z.-H., Lin, X.-P., Yu, A.-N., Zhou, Y.-H., Yang, Y.-B.: Lightweight and accurate single image super-resolution with channel segregation network. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1630–1634 (2021)
29. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Conference and Workshop on Neural Information Processing Systems (NeurIPS) (2014)
30. Ma, F., Zhu, L., Yang, Y., Zha, S., Kundu, G., Feiszli, M., Shou, Z.: Sf-net: Single-frame supervision for temporal action localization. In: European Conference on Computer Vision (ECCV), pp. 420–437 (2020)
31. Li, M., Zhao, L., Zhou, D., Nie, R., Liu, Y., Wei, Y.: Aems: An attention enhancement network of modules stacking for low-light image enhancement. Vis. Comput 1–17 (2021)
32. Bai, J., Chen, R., Liu, M.: Feature-attention module for context-aware image-to-image translation. Vis. Comput. **36**, 2145–2159 (2020)
33. Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E.: Squeeze-and-excitation networks. IEEE Trans. Pattern Anal. Mach. Intell. **42**, 2011–2023 (2020)

34. Woo, S., Park, J., Lee, J.-Y., Kweon, I.-S.: Cbam: Convolutional block attention module. In: European Conference on Computer Vision (ECCV), pp. 3–19 (2018)
35. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: European Conference on Computer Vision (ECCV), pp. 286–301 (2018)
36. Hu, Y., Li, J., Huang, Y., Gao, X.: Channel-wise and spatial feature modulation network for single image super-resolution. IEEE Trans. Circuits Syst. Video Technol. **30**, 3911–3927 (2020)
37. Gori, M., Monfardini, G., Scarselli, F.: A new model for learning in graph domains. In: International Joint Conference on Neural Networks (IJCNN), pp. 729–734 (2005)
38. Zhang, Z., Cui, P., Zhu, W.: Deep learning on graphs: A survey. ArXiv arXiv:1812.04202 (2018)
39. Bruna, J., Zaremba, W., Szlam, A.D., LeCun, Y.: Spectral networks and locally connected networks on graphs. CoRR arXiv:1312.6203 (2014)
40. Kipf, T., Welling, M.: Semi-supervised classification with graph convolutional networks. ArXiv arXiv:1609.02907 (2017)
41. Li, Q., Zhang, T., Chen, C.P., Yi, K., Chen, L.: Residual gcb-net: Residual graph convolutional broad network on emotion recognition. IEEE Transactions on Cognitive and Developmental Systems (2022)
42. Wei, L., Liu, Y., Feng, K., Li, J., Sheng, K., Wu, Y.: Graph convolutional neural network with inter-layer cascade based on attention mechanism. In: IEEE International Conference on Cloud Computing and Intelligent Systems, pp. 291–295 (2021)
43. Caballero, J., Ledig, C., Aitken, A.P., Acosta, A., Totz, J., Wang, Z., Shi, W.: Real-time video super-resolution with spatio-temporal networks and motion compensation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2848–2857 (2017)
44. Li, J., Fang, F., Mei, K., Zhang, G.: Multi-scale residual network for image super-resolution. In: European Conference on Computer Vision (ECCV), pp. 517–532 (2018)
45. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9 (2015)
46. Cheng, K., Zhang, Y., He, X., Chen, W.-H., Cheng, J., Lu, H.: Skeleton-based action recognition with shift graph convolutional network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 180–189 (2020)
47. Wang, X., Girshick, R.B., Gupta, A., He, K.: Non-local neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7794–7803 (2018)
48. Li, Y., Wang, N., Liu, J., Hou, X.: Demystifying neural style transfer. ArXiv arXiv:1701.01036 (2017)
49. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 1122–1131 (2017)
50. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.-L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: British Machine Vision Conference (BMVC) (2012)
51. Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution via sparse representation. IEEE Trans. Image Proces. **19**, 2861–2873 (2010)
52. Martin, D., Fowlkes, C.C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: IEEE International Conference on Computer Vision (ICCV), pp. 416–423 (2001)
53. Matsui, Y., Ito, K., Aramaki, Y., Fujimoto, A., Ogawa, T., Yamasaki, T., Aizawa, K.: Sketch-based manga retrieval using manga109 dataset. Multimed. Tools Appl. **76**, 21811–21838 (2016)
54. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Trans. Image Process. **13**, 600–612 (2004)
55. Zhang, K., Zuo, W., Zhang, L.: Learning a single convolutional super-resolution network for multiple degradations. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3262–3271 (2018)
56. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (ICLR) (2019)
57. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)
58. Molchanov, P., Tyree, S., Karras, T., Aila, T., Kautz, J.: Pruning convolutional neural networks for resource efficient inference (2017)
59. Gu, J., Dong, C.: Interpreting super-resolution networks with local attribution maps. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9199–9208 (2021)
60. Li, Y., Agustsson, E., Gu, S., Timofte, R., Gool, L.: Carn: Convolutional anchored regression network for fast and accurate single image super-resolution. In: European Conference on Computer Vision Workshops (ECCVW) (2018)
61. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y.-H., Dong, C., Loy, C.C., Qiao, Y., Tang, X.: Esrgan: Enhanced super-resolution generative adversarial networks. In: European Conference on Computer Vision Workshops (ECCVW) (2018)
62. Liu, D., Wen, B., Fan, Y., Loy, C.C., Huang, T.: Non-local recurrent network for image restoration. In: Conference and Workshop on Neural Information Processing Systems (NeurIPS) (2018)
63. Zou, W., Ye, T., Zheng, W., Zhang, Y., Chen, L., Wu, Y.: Self-calibrated efficient transformer for lightweight super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 930–939 (2022)
64. Lu, Z., Li, J., Liu, H., Huang, C., Zhang, L., Zeng, T.: Transformer for single image super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 457–466 (2022)
65. Gao, G., Wang, Z., Li, J., Li, W., Yu, Y., Zeng, T.: Lightweight bimodal network for single-image super-resolution via symmetric cnn and recursive transformer. arXiv preprint arXiv:2204.13286 (2022)
66. Lai, W.-S., Huang, J.-B., Ahuja, N., Yang, M.-H.: Deep laplacian pyramid networks for fast and accurate super-resolution, 5835–5843 (2017)
67. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep cnn denoiser prior for image restoration. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2808–2817 (2017)

Yue Yang received the PhD degree in computer science and technology from Xi'an Jiaotong University, China. She is currently lecturer at Xi'an Jiaotong University. Her research interests include image restoration and data privacy preserving.

Yong Qi received the PhD degree from Xi'an Jiaotong University, China. He is currently a full-time professor at Xi'an Jiaotong University. His research interests include operating systems, distributed systems, and cloud computing.

Saiyu Qi received the BS degree in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 2008, and the PhD degree in computer science and engineering from the Hong Kong University of Science and Technology, Hong Kong, in 2014. He is currently an associate professor with the School of Computer Science and Technology, Xi'an Jiaotong University, China. His research interests include applied cryptography, cloud security, distributed systems, and pervasive computing.