A Lightweight Local-Global Attention Network for Single Image Super-Resolution*

Zijiang Song[0000−0002−0225−5547] and Baojiang Zhong (✉)

School of Computer Science and Technology,
Soochow University, Suzhou 215008, China
bjzhong@suda.edu.cn

Abstract. For a given image, the self-attention mechanism aims to capture dependencies for each pixel. It has been proved that the performance of neural networks which employ self-attention is superior in various image processing tasks. However, the performance of self-attention has extensively correlated with the amount of computation. The vast majority of works tend to use local attention to capture local information to reduce the amount of calculation when using self-attention. The ability to capture information from the entire image is easily weakened on this occasion. In this paper, a *local-global attention block* (LGAB) is proposed to enhance both the local features and global features with low calculation complexity. To verify the performance of LGAB, a lightweight *local-global attention network* (LGAN) for single image super-resolution (SISR) is proposed and evaluated. Compared with other lightweight state-of-the-arts (SOTAs) of SISR, the superiority of our LGAN is demonstrated by extensive experimental results. The source code can be found at https://github.com/songzijiang/LGAN.

1 Introduction

For a given low-resolution (LR) image, single image super-resolution (SISR) is a task aiming at generating a high-resolution (HR) one. Among the current mainstream SISR methods (e.g., [5, 8–10, 12, 13, 15, 17, 18, 20, 25]), the SwinIR [17] achieves the impressive performance, and the fundamental idea of SwinIR is self-attention. The self-attention captures *long-range dependencies* (LRDs) for each pixel in an image, and the greatest advantage of self-attention is producing large receptive fields. However, the advantage of self-attention comes with the huge computation. Therefore, lots of works (e.g., [7, 16, 24]) attempt to reduce the computation with less negative impact. Among them, reducing the number of pixels involved to calculate the LRDs is the most salient means. In SwinIR [17],

* This work was supported in part by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant 21KJA520007, in part by the National Natural Science Foundation of China under Grant 61572341, in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions, in part by Collaborative Innovation Center of Novel Software Technology and Industrialization.

dividing the image into non-overlapped windows and executing the self-attention operation in each window. Next, the operation of shifting windows is performed to enlarge the receptive fields. By repeating the operations above, the receptive fields are expanded to the whole image. However, only local attention is used in SwinIR, and it is hard to acquire information efficiently from long-range targets directly. Thus, SwinIR suffers from the lack of long-range relationships to generate the image. Note that, it has been reported in [7] that calculating pixels on the specific path can also effectively reduce the amount of calculation and the receptive fields are able to be expanded to the whole image without repeating. However in CCNet [7], too much attention is paid to long-range relationships, and local relationships are neglected, it is fatal for CCNet [7]. Therefore, networks like CCNet, which reduces computation only by reducing the pixels involved to calculate LRDs, are difficult to achieve an outperforming result.

Motivated by the SwinIR [17] and CCNet [7], we aim to combine both local and global features in an efficient way and reduce the complexity of the self-attention computation. For that, we use self-attention both in the local features and global features in our proposed *local-global attention block* (LGAB). In our proposed LGAB, there are three attention parts: 1) window attention (WA), 2) shifted window attention (SWA), and 3) long-range attention (LRA). Dividing each image into non-overlapped windows is used in WA to extract the local features effectively. Due to the resolution of each window is small, the computational complexity is low in WA. The local features suffer from the lack of long-range relationships, which is expressed in the form of receptive fields. Therefore, we use SWA to build the relationship between neighbor windows and use LRA to expand the receptive fields to the whole image. LGAB can extract both local features and global features. To evaluate the performance of the LGAB we proposed, a *local-global attention network* (LGAN) using LGAB is therefore developed for SISR. For accommodating objects of multiple sizes and further enhancing information of receptive fields, multi-scale method is used in our LGAN.

The rest parts of this paper are organized as follows. Related works are described in Section 2 and our proposed LGAN and LGAB are described in Section 3. Extensive experiments are shown in Section 4 for the performance evaluation of our proposed LGAN. Lastly in Section 5, conclusion is drawn.

2 Related Work

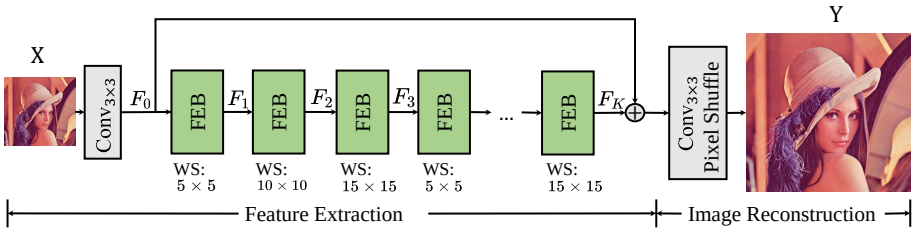
The existing deep learning methods usually enlarge the receptive fields using self-attention for better performance. Non-local block and transformer block are two outstanding blocks using self-attention. For ease of understanding, a brief description of non-local neural networks and transformer-based networks in computer vision (CV) is given.

2.1 Non-local Neural Networks

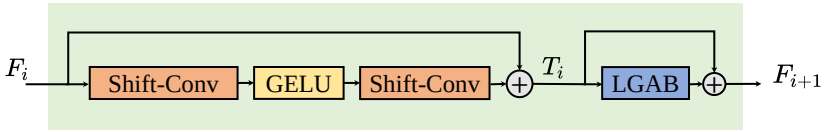
Non-local neural network, whose basic idea is self-attention. This idea was first proposed in [29] as a generic block for capturing LRDs. The non-local operation in [29] can be seen as computing the weighted sum of other positions for each pixel. For each pixel, densely computing LRDs required by pixel-wise dot-products over the entire image has a high complexity. Therefore, multiple non-local blocks can not be added to the network, due to the unbearable amount of calculation. To address this issue, Huang *et al.* [7] suggested reducing the LRD computation by limiting the number of pixels involved. For that, *recurrent criss-cross attention* (RCCA) was proposed in [7], and it computes the LRDs at each pixel position along a specific criss-cross path. A network called *criss-cross network* (CCNet) using RCCA was then proposed for image segmentation [7]. The receptive fields can be easily expanded to the entire image without having to be repeated using non-local-based blocks. Additionally, Mei *et al.* [24] investigated the combinations of non-local operation and sparse representation, and proposed a novel *non-local sparse attention* (NLSA) with a dynamic sparse attention pattern for SISR. However, these approaches suffer from a huge computational burden and do not balance well the local and global representation capabilities.

2.2 Transformer in Computer Vision

Transformer was proposed in [27] firstly, whose outperforming results quickly swept through many tasks in natural language processing (NLP). Due to the difference in the dataset between CV and NLP, the number of pixels in the image is far greater than the number of words in the sentence. Repeating self-attention operations on each pixel in a transformer-based network causes unaffordable expenses in CV task. Therefore, dividing the image into non-overlapped 16×16 windows is used in ViT [3] for CV task. Each window is seen as a token and fed into the transformer block. With the operation above, the information of position in image is destroyed, and position embedding is therefore added to the network. ViT is effective to reduce the computation complexity compared to treating each pixel as a token. However, considering that taking each 16×16 window as a token could lose low-level information, especially for image restoration. To solve this issue, Swin [19] was proposed. The same as ViT [3], the input images are divided into non-overlapped windows. Different from ViT [3], the self-attention operation is only performed in each window. We can think of this operation as a local attention block. However, global information is lost as a result. To get the information from the entire image, a shifting window operation was proposed in [19]. With the repetition of the transformer operations, the receptive fields can be expanded to the global image. Due to the impressive performance of Swin, the network structure was applied to SISR and named SwinIR [17]. SwinIR achieved state-of-the-art (SOTA) result compared to the previous networks on the vast majority of benchmark datasets. Although the shifting operation is added in the network, SwinIR is not efficient enough at obtaining information from long-range targets.



(*a*) Local-global attention network (LGAN)



(*b*) Feature extract block (FEB)

Fig. 1. The overall of the proposed LGAN. The window size of each feature extract block (FEB) is denoted by WS, and '$\oplus$' denotes the operation of residual plus.

3 The Proposed Approach

In this section, the structures of 1) the proposed *local-global attention network* (LGAN), 2) *feature extract block* (FEB) and 3) *local-global attention block* (LGAB) are described.

3.1 Network Structure

As shown in Fig. 1, our proposed LGAN contains two stages: 1) feature extraction and 2) image reconstruction. A set of feature maps denoted by F_0 is generated by 3×3 convolution based on the given image X in the feature extraction stage as follows:

$$F_0 = \mathrm{Conv}_{3\times3}(X), \tag{1}$$

where $\mathrm{Conv}_{3\times3}(\cdot)$ denotes the convolution operation with 3×3 kernel size. As shown in Fig. 1, the generated F_0 is further enhanced by several FEBs. In our LGAN, it is empirically determined that the number of FEBs is 24 (i.e., $K = 24$). Finally in the image reconstruction stage, based on the 'coarse' feature maps F_0 and the residual part F_K, HR image Y will be generated via the *pixel shuffle* [26] as follows:

$$Y = \mathrm{Conv}_{3\times3}(\mathrm{U}(F_0 + F_K)), \tag{2}$$

where the pixel shuffle operator is denoted by $\mathrm{U}(\cdot)$, which is used in LGAN for upsampling.

When training our LGAN, a set of image pairs $\{X^{(n)}, H^{(n)}\}_{n=1}^{N}$ is used. The $X^{(n)}$ denotes the LR image and the $H^{(n)}$ denotes the ground-truth (GT) image

correspondingly. Minimizing the loss is performed for training between $Y^{(n)}$ and $H^{(n)}$; i.e.,

$$\Theta^* = \arg\min_{\Theta} \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}\left(Y^{(n)}, H^{(n)}\right), \tag{3}$$

where Θ denotes the set of parameters to be learned in our LGAN, and $\mathcal{L}(\cdot)$ stands for the smooth ℓ_1 loss function [4].

3.2 Feature Extract Block (FEB)

Different from the existing transformer blocks, our FEBs excavate both local and global information simultaneously with low computational complexity as depicted in Fig. 1(b). In our proposed FEB, there are a local-global attention block (LGAB) and two shift-conv blocks [30]. For shift-conv blocks, smaller computational parameters are required when expanding the receptive fields compared with 3×3 convolutions. GELU is chosen to be the activation function. For F_i, the mathematical definitions in FEB are depicted as follows to generate F_{i+1}:

$$\begin{aligned} T_i &= \mathrm{SC}(\mathrm{GELU}(\mathrm{SC}(F_i))) + F_i, \\ F_{i+1} &= \mathrm{LGAB}(T_i) + T_i, \end{aligned} \tag{4}$$

where $\mathrm{SC}(\cdot)$ denotes the shift-conv blocks [30], $\mathrm{GELU}(\cdot)$ is the activation function, and $\mathrm{LGAB}(\cdot)$ represents LGAB which defined as shown in the following section.

3.3 Local-Global Attention Block (LGAB)

As depicted in Fig. 2, there are three attention parts in an LGAB: 1) window attention (WA), 2) shifted window attention (SWA), and 3) long-range attention (LRA). For a given set of feature maps T_i, it is splited into three parts on the channel dimension, denoted by $x^{(1)}$, $x^{(2)}$ and $x^{(3)}$, respectively. The number of channels of $x^{(k)}$ $(k = 1, 2, 3)$ is one third of the number of channels of T_i. $x^{(1)}$, $x^{(2)}$ and $x^{(3)}$ are fed into the three attention parts to achieve $\mathrm{LGAB}(T_i)$ as follows:

$$\mathrm{LGAB}(T_i) = \mathrm{Conv}_{1\times1}(\mathrm{CAT}(\mathrm{WA}(x^{(1)}), \mathrm{SWA}(x^{(2)}), \mathrm{LRA}(x^{(3)}))), \tag{5}$$

where $\mathrm{WA}(\cdot)$, $\mathrm{SWA}(\cdot)$ and $\mathrm{LRA}(\cdot)$ denote window attention, shifted window attention and long rang attention, respectively; concatenate operation on channel dimension is denoted by $\mathrm{CAT}(\cdot)$.

Window attention. Motivated by SwinIR [17], the original images are divided into non-overlapped windows. In order to adapt to objects of different scales, multi-scale resolutions of windows (i.e., 5×5, 10×10, and 15×15) are used. The reason that we not set the sizes of windows to power of 2 (e.g 4×4, 8×8, and 16×16), is to seek larger receptive fields. For each pixel, the range capturing

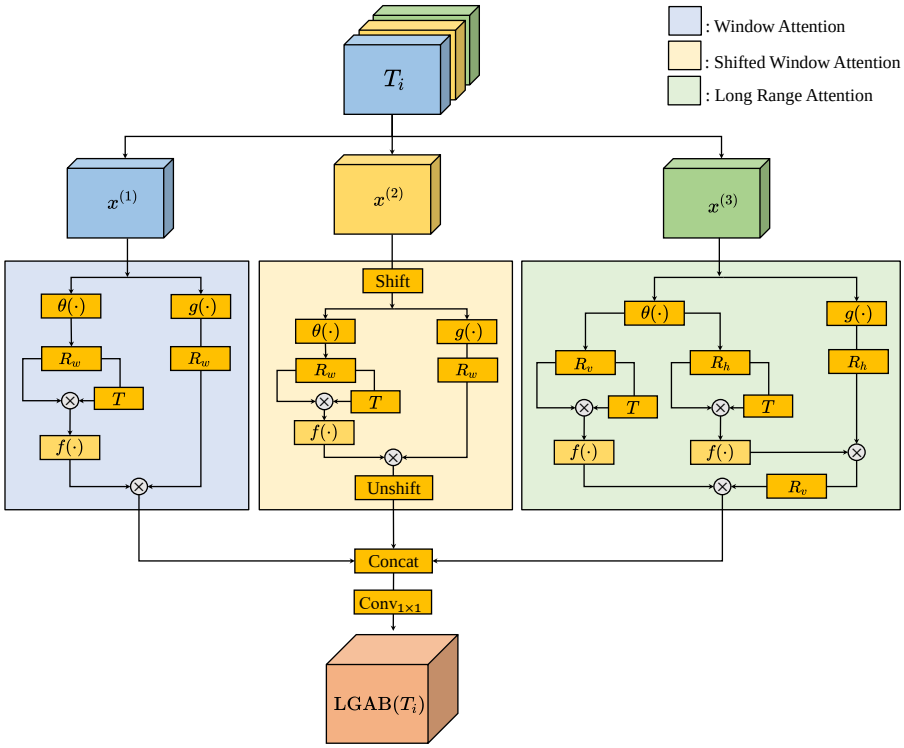


Fig. 2. The overview of our proposed LGAB. The input set of feature maps is divided on channels into 3 sets of feature maps. Three attention parts are performed on these three divided feature maps, respectively.

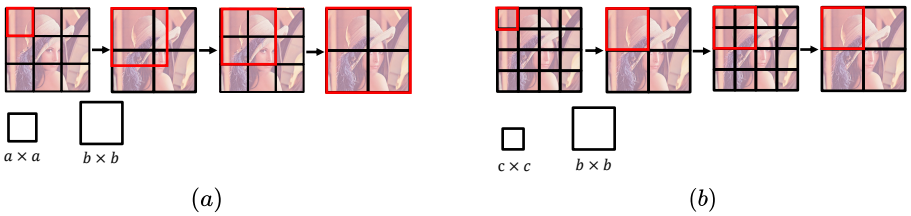


Fig. 3. Receptive fields in windows of different sizes. The windows are denoted in the big black square and the receptive fields are represented by the red box. Each small black box denotes a divided window.

information is determined by the size of its corresponding receptive fields. For example, only considering WA, the size of receptive fields is determined by the least common multiple of the sizes of windows. As shown in Fig. 3(a), there are two window sizes $a \times a$ and $b \times b$, and there is $3a = 2b$. Therefore, the least

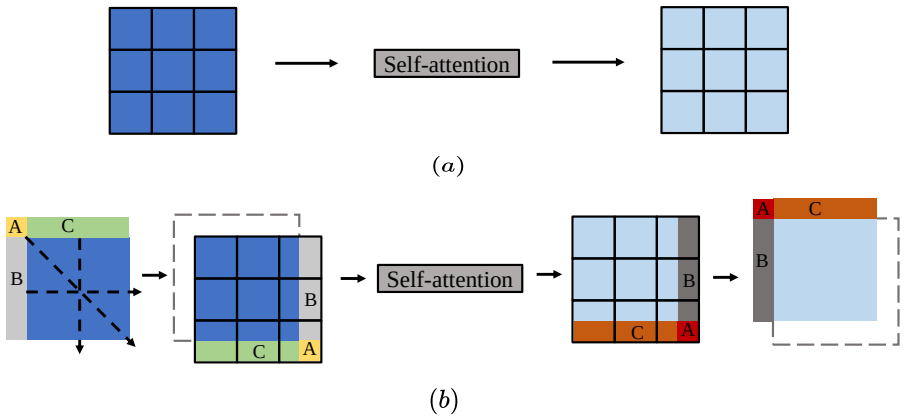


Fig. 4. A comparison of two parts in LGAB: (a) window attention and (b) shifted window attention.

common multiple of these window sizes is $2b \times 2b$, which is the same size as the entire image. On the other hand in Fig. 3(b), the window sizes are $c \times c$ and $b \times b$. The least common multiple of these window sizes is b and the receptive fields is $b \times b$. These two cases have the approximate amount of computation, but the receptive fields of case (a) are quadruple of case (b).

Let θ and g denote two 1×1 convolutions. For the given feature maps $x^{(1)}$, there are $\theta(x^{(1)}) = W_\theta x^{(1)}$ and $g(x^{(1)}) = W_g x^{(1)}$, where W_θ and W_g are weight matrices to be learned. The operation of reshaping feature maps into windows is denoted by R_w. For the given feature maps $x^{(1)}$ of shape $B \times C \times H \times W$, shape of $R_w(x^{(1)})$ is $B \cdot n \cdot n \times nh \cdot nw \times C$, where $nw \times nh$ denotes the pre-set window size; and $n = \lceil W/nw \rceil$, $n = \lceil H/nh \rceil$. The window where $x^{(1)}_{i,j}$ is located is denoted by $\Omega^{(1)}(i,j)$, and $\mathrm{WA}(x^{(1)})_{i,j}$ is performed as follows:

$$\mathrm{WA}(x^{(1)})_{i,j} = \sum\nolimits_{(s,t)\in\Omega^{(1)}(i,j)} f(\theta(x^{(1)}_{i,j}) \cdot \theta(x^{(1)}_{s,t})^T) \cdot g(x^{(1)}_{s,t}), \tag{6}$$

where $f(\cdot)$ denotes the softmax operation. WA can fully utilize the local features with low computation complexity.

Shifted window attention. WA only captures information within each individual window, and the information in the neighbor window is established by SWA [17]. As shown in Fig. 4(b), the black boxes denote the windows and the shift size is set to the half of the window size to expand the receptive fields to a greater extent. As for a given set of feature maps in WA, it is divided into several windows and self-attention is executed in each window in Fig. 4(a). Comparatively speaking between Fig. 4(a) and (b), the yellow pixels in area A are shifted from top left to bottom right, and pixels in area B and area C are processed correspondingly in SWA. Following, the shifted image is divided into windows

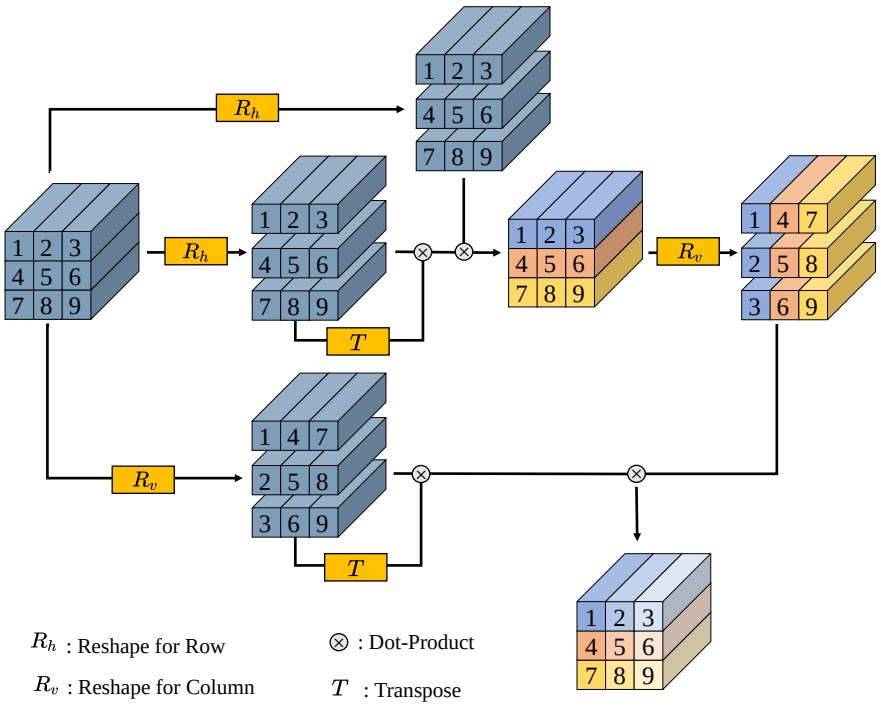


Fig. 5. Operations of reshaping feature maps in LRA. There are two reshaping operations in LRA: (a) R_h, and (b) R_w. Each number denotes a pixel in the feature maps. For the convenience of presentation, the parts of convolution and softmax operation are omitted.

as shown in the WA part. Next, the self-attention operation is performed in each shifted window. Inverse operation of shifting windows is used to restore the original image. The feature maps $x^{(2)}$ are calculated as follows:

$$\mathrm{SWA}(x^{(2)}) = \mathrm{UnShift}(\mathrm{WA}(\mathrm{Shift}(x^{(2)}))), \tag{7}$$

where $\mathrm{WA}(\cdot)$ denotes the window attention described in previous subsection and $\mathrm{Shift}(\cdot)$, $\mathrm{UnShift}(\cdot)$ denote the shifting windows, and the inverse operation of shifting windows as shown in Fig. 4(b), respectively. SWA enhances the relationship between neighbor windows, which is lacking in WA.

Long-range attention. In both WA and SWA, self-attention is calculated densely between the current pixel and other pixels in the same window. Although receptive fields can be expanded to the entire image by shifted windows, we use LRA to enhance the ability of the network to catch information from the whole image in a more efficient way. For a given set of feature maps $x^{(3)}$, whose shape

is $B \times C \times H \times W$. There are R_h and R_v denote the operation of reshaping feature maps along horizontal and vertical direction, respectively, as shown in Fig. 5. After R_h, the shape of the set of feature maps is $B \cdot W \times H \times C$ and after R_v is $B \cdot H \times W \times C$, where H and W denote the height and width of the set of the feature maps $x^{(3)}$, respectively. As depicted in Fig. 5, the set of the feature maps $x^{(3)}$ is reshaped by R_h and executed self-attention in the same row for each pixel (e.g., for pixel 1, the pixels involved to calculate are pixel 1, pixel 2, and pixel 3). Then the set of the feature maps is reshaped by R_w, and only pixels in the same column are considered into self-attention calculating (e.g., for pixel 1, pixels involved in calculating are pixel 1, pixel 4, and pixel 7 in Fig. 5). The definition of $\mathrm{LRA}(x^{(3)})$ is shown as follows:

$$M = f(R_h(\theta(x^{(3)})) \cdot R_h(\theta(x^{(3)}))^T) \cdot R_h(g(x^{(3)})), \tag{8}$$

$$\mathrm{LRA}(x^{(3)}) = f(R_v(\theta(x^{(3)})) \cdot R_v(\theta(x^{(3)}))^T) \cdot R_v(g(M)). \tag{9}$$

LRA can catch the LRDs from the entire image with low computation complexity and make up for the lack of global features in WA and SWA.

4 Experiments

In this section, the settings of the experiment and training steps are shown. To verify the outperforming structure of our LGAN, the ablation experiments are shown. Lastly, The experiments of performance are shown in indicators and visual results for a comparison between proposed LGAN and other lightweight SOTAs for SISR.

4.1 Experimental Setup

Our LGAN is trained on DIV2K [1], which is the standard benchmark dataset for SISR. Set5 [2], Set14 [31], B100 [22], Urban100 [6], and Manga109 [23] are used to evaluate the performance of LGAN. Window sizes are set to 5×5, 10×10, and 15×15 for 3 continuously FEBs. There are 24 FEBs in our proposed LGAN and the number of channels is set to 60. And 64 images are used in a batch for training fairly. Adam [11] is selected as our optimizer and smooth ℓ_1 loss [4] is selected as our loss function. All experiments were running on Nvidia Titan XP GPUs and implemented by PyTorch based on ELAN [32][1]. The results are evaluated by PSNR and SSIM metrics on the Y channel (i.e., luminance) of YCbCr space. The LR images generated from the HR images (i.e., GT images) by downsampling are fed into the network. Data augmentation including random flips and rotations is applied when training our proposed LGAN. We cut each LR image into a 60×60 patch for training. Lastly, the $\times2$ model was trained with $1,000$ epochs, for each epoch, the training dataset are repeated 80 times. The initial learning rate was 2×10^{-4} and was reduced by half at epoch 500,

[1] `https://github.com/xindongzhang/ELAN`

Table 1. An ablation study using PSNR and SSIM for LGAN of scale $\times 2$ trained with 100 epochs.

Case	Attention Block	Activation Function	BSDS100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
(a)	WA+SWA+LRA	GELU	**32.13**/**.8994**	**32.00**/**.9274**	**38.34**/**.9764**
(b)	WA	GELU	32.12/.8991	31.83/.9260	38.24/.9260
(c)	WA+SWA	GELU	**32.13**/.8992	31.91/.9262	38.29/**.9764**
(d)	LRA	GELU	32.04/.8976	31.73/.9239	38.04/.9758
(e)	WA+SWA+LRA	ReLU	32.11/.8992	31.95/.9267	38.19/.9762

Table 2. A comparison of total FLOPs in attention operations between three attention blocks in ablation experiments. For the convenience of count, the calculation of convolution and softmax are discarded. Total channels of these cases are all set to 60. The column of channels in the table is denoted by $a \times b$, where a denotes the channels of each block and b denotes the number of blocks.

Case	WA	SWA	LRA	Channels	FLOPs (M)
(a)	✓	✓	✓	20×3	506.9
(b)	✓	×	×	60×1	604.8
(c)	✓	✓	×	30×2	604.8
(d)	×	×	✓	60×1	311.0

800, 900, and 950. Larger magnification factors (i.e., $\times 3$ and $\times 4$) were trained for 500 epochs from the starting based on the trained $\times 2$ network. The learning rate was reduced by half at epoch 250, 400, 450, and 475, the initial learning rate was also inited to 2×10^{-4}.

4.2 Ablation Study

To prove that our LGAN is an effective structure, we made an ablation experiment on the scale of $\times 2$ and the results are shown in Tab. 1. To save time and computational resources, all ablation experiments were trained with 100 epochs, and the LR images are cropped into 30×30 patches for training, and the batchsize is set to 64. Due to the different attention parts occupying a part of channels alone, respectively. The more attention parts are added to the LGAB, the fewer channels are distributed to each attention part. Therefore as shown in Tab. 2, cases with more attention parts are not necessarily more computationally expensive than cases with fewer attention parts. In Tab. 1, case (a) serves as the baseline and other cases are built based on case (a). In case (b), only WA part is used in block and the result of PSNR reduces from 32.00dB to 31.83dB on Urban100. Same, the performance also deteriorates on both BSD100 and Manga109. And then in case (c), WA part and SWA part are added to the block. Compared with case (b), performance has improved (i.e., 31.91dB vs 31.83dB) but there is still a certain gap with case (a) (i.e., 31.91dB vs 32.00dB)

Table 3. Angles of the line segment produced by LSD [28] in five benchmark datasets and the changing PSNR and SSIM of SwinIR compared to LGAN.

	Method	Set5	Set14	BSDS100	Urban100	Manga109
$0^\circ \pm 10^\circ$	–	13.60%	13.86%	13.19%	19.70%	18.18%
$90^\circ \pm 10^\circ$	–	10.00%	19.71%	21.37%	19.81%	15.82%
Total	–	23.60%	33.57%	34.56%	39.51%	34.60%
PSNR	SwinIR	-0.04dB	-0.06dB	-0.02dB	-0.16dB	-0.15dB
SSIM	SwinIR	-0.0008	-0.0006	-0.0010	-0.0042	-0.0000

on Urban100. In case (d), only LRA part is added in LGAB. Compared with case (a), case (d) yields a worse result on Urban100 (i.e., 31.73dB vs 32.00dB). As shown in case (a), (b), (c) and (d), it is demonstrated that these three attention parts in LGAB can compensate each other for local and global relationships. To further explore higher performance, GELU is replaced by ReLU in case (e). It turns out that GELU is better than ReLU for our network (i.e., case(a) vs case (e)).

4.3 Compared with SwinIR

In this subsection, the reason LGAN outperforms SwinIR [17], which is the most effective SISR network of the currently accepted papers in mainstream opinion, is discussed.

'Local attention only' versus 'local and global attention'. In SwinIR [17], the image is split into several non-overlapped windows, and attention operation is executed in each window. Only local information is captured and it is a fatal problem. In our LGAN, LRA is proposed to obtain the global information at the same time in one block.

Better spatial location. Spatial information is very important for CV tasks. However, in original ViT [3] or SwinIR [17], spatial information is handled by position embedding, which can not explore the potential of spatial information well. Our proposed LRA could utilize spatial information more effectively.

More reasonable organization of blocks. Although SwinIR and LGAN both include base units window attention (WA) and shifted window attention (SWA), the ways these base units are organized are quite different. In SwinIR [17] or in transformer [27], multi-head attention (MHA) is designed to extract features. In each block, channels are split into several heads. Each head is executed by the same attention operation (e.g., WA). In our LGAN, channels are split into 3 parts to execute 3 different attention operations (i.e., WA, SWA, LRA). Experimental results demonstrated that, compared with SwinIR's single function, our LGAB

Table 4. Comparison of PSNR and SSIM with other lightweight SISR methods. The results highlighted in red are the best, and the second best results are in blue. Data not given in the corresponding paper is identified using '–'.

Method	Scale	Params.	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSDS100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
MSICF [5]	x2	–	37.89/.9605	33.41/.9153	32.15/.8992	31.47/.9220	–/–
SRNIF [15]		–	38.05/.9607	33.65/.9181	32.19/.9002	32.14/.9286	–/–
NLRN [18]		–	38.00/.9603	33.46/.9159	32.19/.8992	31.81/.9249	–/–
LapSRN [12]		–	37.52/.9591	33.08/.9130	31.80/.8949	30.41/.9101	37.27/.9740
MSRN [13]		–	38.08/.9605	33.74/.9170	32.23/.9013	32.22/.9326	38.82/.9771
MIPN [20]		–	38.12/.9609	33.73/.9188	32.25/.9006	32.42/.9312	38.88/.9773
AMNet [10]		–	38.13/.9608	33.77/.9191	32.27/.9008	32.52/.9320	39.02/.9779
LatticeNet [21]		756K	38.06/.9607	33.70/.9187	32.20/.8999	32.25/.9288	–/–
LAPAR-A [14]		548K	38.01/.9605	33.62/.9183	32.19/.8999	32.10/.9283	38.67/.9772
IDN [9]		579K	37.83/.9600	33.30/.9148	32.08/.8985	31.27/.9196	38.01/.9749
IMDN [8]		694K	38.00/.9605	33.63/.9177	32.19/.8996	32.17/.9283	38.88/.9774
HRFFN [25]		646K	38.12/.9608	33.80/.9192	32.24/.9005	32.52/.9319	39.05/.9797
SwinIR [17]		878K	38.14/.9611	33.86/.9206	32.31/.9012	32.76/.9340	39.12/.9783
LGAN (Ours)		650K	38.13/.9612	33.95/.9221	32.32/.9017	32.81/.9343	39.13/.9777
MSICF [5]	x3	–	34.24/.9266	30.09/.8371	29.01/.8024	27.69/.8411	–/–
SRNIF [15]		–	34.42/.9274	30.36/.8426	29.06/.8047	28.23/.8541	–/–
NLRN [18]		–	34.27/.9266	30.16/.8374	29.06/.8026	27.93/.8453	–/–
LapSRN [12]		–	33.82/.9227	29.87/.8320	28.82/.7973	27.07/.8270	32.21/.9343
MSRN [13]		–	34.38/.9262	30.34/.8395	29.08/.8041	28.08/.8554	33.44/.9427
MIPN [20]		–	34.53/.9280	30.43/.8440	29.15/.8061	28.38/.8573	33.86/.9460
AMNet [10]		–	34.51/.9281	30.47/.8445	29.18/.8074	28.51/.8595	34.10/.9474
LatticeNet [21]		765K	34.40/.9272	30.32/.8416	29.10/.8049	28.19/.8513	–/–
LAPAR-A [14]		544K	34.36/.9267	30.34/.8421	29.11/.8054	28.15/.8523	33.51/.9441
IDN [9]		588K	34.11/.9253	29.99/.8354	28.95/.8013	27.42/.8359	32.71/.9381
IMDN [8]		703K	34.36/.9270	30.32/.8417	29.09/.8046	28.17/.8519	33.61/.9445
HRFFN [25]		654K	34.49/.9279	30.41/.8433	29.13/.8061	28.43/.8574	33.82/.9459
SwinIR [17]		886K	34.62/.9289	30.54/.8463	29.20/.8082	28.66/.8624	33.98/.9478
LGAN (Ours)		658K	34.56/.9286	30.60/.8463	29.24/.8092	28.79/.8646	34.19/.9482
MSICF [5]	x4	–	31.91/.8923	28.35/.7751	27.46/.7308	25.64/.7692	–/–
SRNIF [15]		–	32.34/.8970	28.66/.7838	27.62/.7380	26.32/.7935	–/–
NLRN [18]		–	31.92/.8916	28.36/.7745	27.48/.7306	25.79/.7729	–/–
LapSRN [12]		–	31.54/.8863	28.19/.7720	27.32/.7262	25.21/.7548	29.09/.8890
MSRN [13]		–	32.07/.8903	28.60/.7751	27.52/.7273	26.04/.7896	30.17/.9034
MIPN [20]		–	32.31/.8971	28.65/.7832	27.61/.7375	26.23/.7906	30.67/.9107
AMNet [10]		–	32.28/.8962	28.71/.7841	27.66/.7392	26.37/.7951	31.04/.9136
LatticeNet [21]		777K	32.18/.8943	28.61/.7812	27.57/.7355	26.14/.7844	–/–
LAPAR-A [14]		569K	32.15/.8944	28.61/.7818	27.61/.7366	26.14/.7871	30.42/.9074
IDN [9]		677K	31.82/.8903	28.25/.7730	27.41/.7297	25.41/.7632	29.41/.8942
IMDN [8]		715K	32.21/.8948	28.58/.7811	27.56/.7353	26.04/.7838	30.45/.9075
HRFFN [25]		666K	32.33/.8960	28.69/.7830	27.62/.7378	26.32/.7928	30.73/.9107
SwinIR [17]		897K	32.44/.8976	28.77/.7858	27.69/.7406	26.47/.7980	30.92/.9151
LGAN (Ours)		669K	32.48/.8984	28.83/.7864	27.71/.7416	26.63/.8022	31.07/.9151

has more powerful functions with fewer parameters and complexity as shown in Tab. 2.

Adaptivity to object size. Multi-scale strategy is applied in our LGAN to adapt to objects with different sizes. In the image, the size of pixels occupied by objects is different. Window sizes of $5, 10, 15$ are used in our LGAN.

Higher quality of restoring line segments. At the same time, we observe that on the Urban100 dataset, our LGAN has huge superiority to restore the

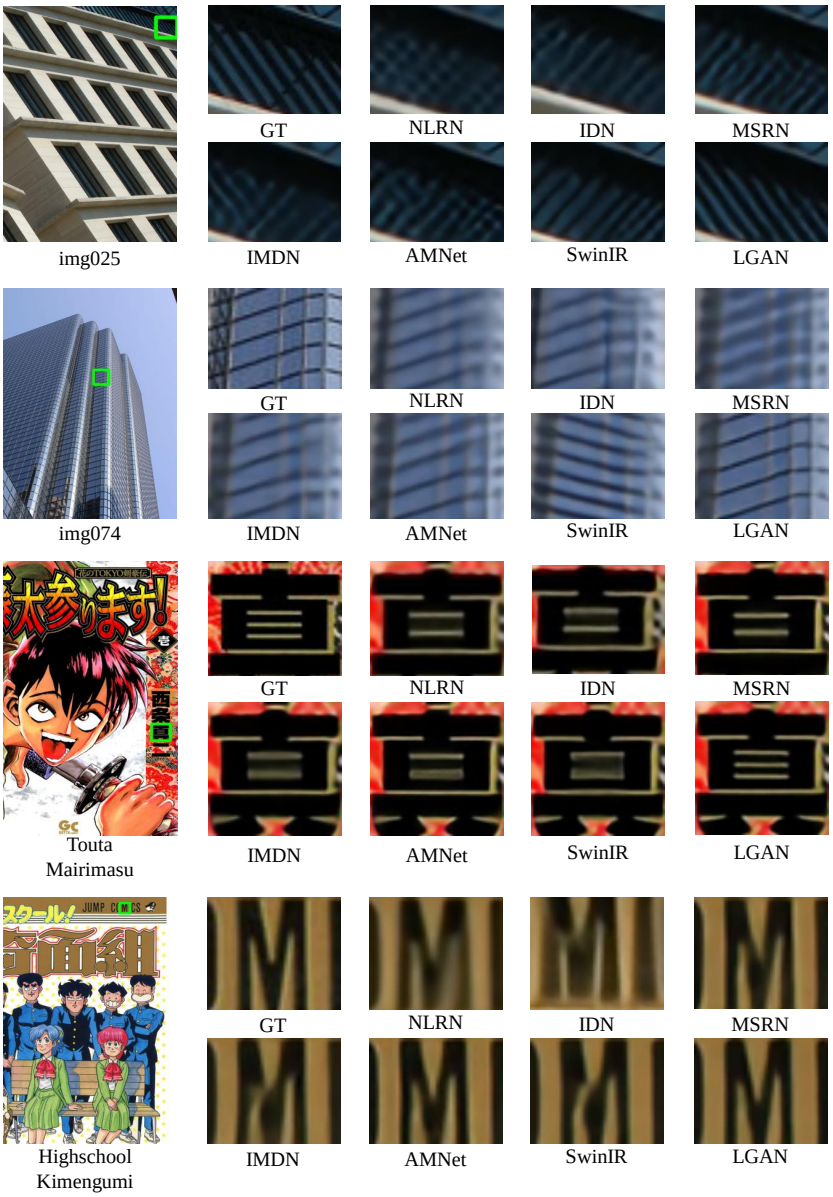


Fig. 6. A visual comparison of $\times 4$ scale with other lightweight SISR networks on Urban100 and Manga109.

horizontal and vertical line segments. To evaluate the reinforcement of our LGAN of line segments, Tab. 3 is shown. Straight line segment detection by LSD [28] is firstly performed on the five datasets, and then the detected straight line

segments are divided into from $-10°$ to $170°$ according to their slopes. The line segments with slopes from $-10°$ to $10°$ are considered as horizontal ones, and line segments from $80°$ to $100°$ are considered as vertical straight line segments. The percentage of horizontal and vertical line segments are 19.70% and 19.81% in Urban100 respectively, and significantly ahead of the average (percentage of per $20°$ is 11.11%). Comparing SwinIR and our proposed LGAN, PSNR is reduced by 0.16dB correspondingly in Urban100. In contrast in Set5, only 13.60% and 10.00% of line segments are horizontal or vertical. As a result, PSNR has been reduced only by 0.04dB. From this, we can speculate that our LGAN can yield outperforming results compared with SwinIR on the datasets regardless of the ratio of vertical and horizontal line segments, and if on a dataset with more vertical and horizontal line segments like Urban100, our network can achieve considerably increased performance.

4.4 Performance Evaluation

The results of indicators are shown in Tab. 4. We choose several lightweight SISR networks to compare, including MSICF [5], SRNIF [15], NLRN [18], LapSRN [12], MSRN [13], MIPN [20], AMNet [10], LatticeNet [21], LAPAR-A [14], IDN [9], IMDN [8], HRFFN [25], and SwinIR [17]. If there are multiple versions of the same network to choose from, the lightweight version is chosen to be compared in this paper fairly. Since some networks did not provide the number of parameters, only a part of parameters are listed in Tab. 4. Our proposed LGAN achieves impressive results on most benchmark datasets and most scales.

In Fig. 6, subjective comparisons on Urban100 and Manga109 are shown. Due to space limitations, we have only selected six networks developed in recent years for comparison, including NLRN [18], IDN [9], MSRN [13], IMDN [8], AMNet [10], and SwinIR [17]. Note that, SwinIR is the most effective of the currently accepted papers in mainstream opinion. From the visual results, the reconstructions of existing methods are of low quality and have obvious errors, while our LGAN delivers an outstanding image quality for SISR.

5 Conclusion

In this paper, a block called *local-global attention block* (LGAB) is proposed. In each LGAB, there are three different attention parts: 1) window attention (WA), 2) shifted window attention (SWA), and 3) long-range attention (LRA). Then an efficient network called *local-global attention network* (LGAN) is proposed for single image super-resolution (SISR). Extensive experiments demonstrate that our proposed LGAN can yield outperformance on the most benchmark datasets over the existing lightweight state-of-the-arts for SISR. This work focuses on the mechanism of self-attention blocks, and may enlighten some insights for further studies.

References

1. Agustsson, E., Timofte, R.: NTIRE 2017 challenge on single image super-resolution: Dataset and study. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop. pp. 126–135 (2017)
2. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: Proc. Brit. Mach. Vis. Conf. pp. 1–10 (2012)
3. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Int. Conf. Learn. Represent. (2020)
4. Girshick, R.: Fast r-cnn. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (2015)
5. Hu, Y., Gao, X., Li, J., Huang, Y., Wang, H.: Single image super-resolution with multi-scale information cross-fusion network. Signal Process. **179**, 107831 (2021)
6. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. pp. 5197–5206 (2015)
7. Huang, Z., Wang, X., Wei, Y., Huang, L., Shi, H., Liu, W., Huang, T.S.: CCNet: Criss-cross attention for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell., early access (2020)
8. Hui, Z., Gao, X., Yang, Y., Wang, X.: Lightweight image super-resolution with information multi-distillation network. In: Proc. ACM Int. Conf. Multimedia. pp. 2024–2032 (2019)
9. Hui, Z., Wang, X., Gao, X.: Fast and accurate single image super-resolution via information distillation network. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. pp. 723–731 (2018)
10. Ji, J., Zhong, B., Ma, K.K.: Single image super-resolution using asynchronous multi-scale network. IEEE Signal Process. Lett. **28**, 1823–1827 (2021)
11. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Int. Conf. Learn. Represent. (2015)
12. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep Laplacian pyramid networks for fast and accurate super-resolution. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. pp. 624–632 (2017)
13. Li, J., Fang, F., Mei, K., Zhang, G.: Multi-scale residual network for image super-resolution. In: Proc. Eur. Conf. Comput. Vis. pp. 517–532 (2018)
14. Li, W., Zhou, K., Qi, L., Jiang, N., Lu, J., Jia, J.: Lapar: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond. Adv. Neural Inf. Process. Syst. (2020)
15. Li, X., Chen, Z.: Single image super-resolution reconstruction based on fusion of internal and external features. Multimed. Tools. Appl. pp. 1–17 (2021)
16. Li, Y., Jin, X., Mei, J., Lian, X., Yang, L., Xie, C., Yu, Q., Zhou, Y., Bai, S., Yuille, A.L.: Neural architecture search for lightweight non-local networks. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. pp. 10297–10306 (2020)
17. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: Swinir: Image restoration using swin transformer. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. pp. 1833–1844 (2021)
18. Liu, D., Wen, B., Fan, Y., Loy, C.C., Huang, T.S.: Non-local recurrent network for image restoration. In: Adv. Neural Inf. Process. Syst. pp. 1673–1682 (2018)

19. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proc. IEEE Int. Conf. Comput. Vis. pp. 10012–10022 (2021)
20. Lu, T., Wang, Y., Wang, J., Liu, W., Zhang, Y.: Single image super-resolution via multi-scale information polymerization network. IEEE Signal Process. Lett. **28**, 1305–1309 (2021)
21. Luo, X., Xie, Y., Zhang, Y., Qu, Y., Li, C., Fu, Y.: Latticenet: Towards lightweight image super-resolution with lattice block. In: Proc. Eur. Conf. Comput. Vis. (2020)
22. Martin, D., Fowlkes, C., Tal, D., Malik, J., et al.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. IEEE Int. Conf. Comput. Vis. pp. 416–423 (2001)
23. 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 (2017)
24. Mei, Y., Fan, Y., Zhou, Y.: Image super-resolution with non-local sparse attention. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. pp. 3517–3526 (2021)
25. Qin, J., Liu, F., Liu, K., Jeon, G., Yang, X.: Lightweight hierarchical residual feature fusion network for single-image super-resolution. Neurocomputing (2022)
26. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. pp. 1874–1883 (2016)
27. 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)
28. Von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: Lsd: A line segment detector. Image Process. Line pp. 35–55 (2012)
29. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. pp. 7794–7803 (2018)
30. Wu, B., Wan, A., Yue, X., Jin, P., Zhao, S., Golmant, N., Gholaminejad, A., Gonzalez, J., Keutzer, K.: Shift: A zero flop, zero parameter alternative to spatial convolutions. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. pp. 9127–9135 (2018)
31. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: Int. Conf. Curves and Surf. pp. 711–730 (2010)
32. Zhang, X., Zeng, H., Guo, S., Zhang, L.: Efficient long-range attention network for image super-resolution. In: Proc. Eur. Conf. Comput. Vis. (2022)