Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **9**, Number **3**, August **2024**

An Amalgamated CNN-Transformer Network for Lightweight Image Super-Resolution

Jinsheng Fang*, Hanjiang Lin

School of Computer Science and Engineering
Minnan Normal University, Zhangzhou 363000, China
Key Laboratory of Data Science and Intelligence Application
Fujian Province University, Fujian Zhangzhou, 363000, China
fjs1867@mnnu.edu.cn, 2961094656@qq.com

Kun Zeng*

School of Computer and Control engineering
Fujian Provincial Key Laboratory of Information Processing and Intelligent Control
Minjiang University, Fuzhou 350108, China
zengkun@mju.edu.cn

*Corresponding author: Jinsheng Fang and Kun Zeng

Received September 8, 2023, revised December 9, 2023, accepted February 23, 2024.

Abstract. *Recently, Transformer-based methods for single image super-resolution (SISR) have achieved better performance advantages than the methods based on convolutional neural network (CNN). Exploiting self-attention mechanism to model global context definitely improves the SR results. However, the neglect of local information will bring inevitable reduction of the network performance. In this work, we propose an Amalgamated CNN-Transformer network for lightweight SR, namely ACTSR. Specifically, an amalgamated CNN-Transformer block (ACTB) is developed to extract the useful information of both local and global features. By employing stacked ACTBs, our ACTSR extracts more informative deep features beneficially for super-resolution reconstruction to improve network performance while keeps lightweight and flexible enough. Extensive experiments on commonly used benchmark datasets validate our ACTSR outperforms the advanced competitors. Our codes are available at: https://github.com/ginsengf/ACTSR.*
Keywords:super-resolution, self-attention, spatial attention, Transformer, lightweight network

1. **Introduction.** Single image super-resolution (SISR) is a typical issue that devotes to rebuilding high-resolution (HR) images from corresponding low-resolution (LR) ones [1, 2]. Recently, the convolutional neural network (CNN) has been demonstrated as a powerful tool in various fields of computer vision [3, 4, 5]. As for SR task, the incorporation of CNN-based methods has achieved impressive performance with respect to the traditional methods [6].

Dong et al. proposed the first convolutional neural network for SR (SRCNN), a simple yet efficient three-layer network reconstruct high-quality HR image from LR image [1]. From then on, a large plenty of effective networks, such as very deep convolutional network (VDSR) [7], enhanced deep residual network (EDSR) [8], residual dense network (RDN) [9], residual channel attention network (RCAN) [10], are sequentially proposed to achieve better network performance. These proposed models indicate constructing a deeper and/or wider network architecture can obtain better SR results. However, as

the model size of network increases, the parameter and computational cost explode as well, and it is challenging to apply these methods on resource-limited devices [11, 12]. Moreover, the convolutional kernels used in CNN have limited receptive fields, leading to information loss on non-local feature information.

To this end, a novel architecture derived from natural language processing, known as Transformer [13], is proposed to provide a self-attention mechanism to capture long-term information. Locality vision transformer (LocalViT) [14] introduces CNN to bring local information into Transformers. Then, Transformer using shifted windows (Swin Transformer) [15] is proposed with greater efficiency of self-attention computation and an improved model for SR called SwinIR [16] is proposed thereafter. In SwinIR, a convolutional layer is added to several Swin Transformer layers to extract more features. Through an integration of Transformer and CNN, SwinIR outperforms contemporary state-of-the-art (SOTA) SR methods. However, embedding simple convolutional layers in Transformer-based models cannot fully extract the local information and more delicate CNN structures are required to improve the network performance [17].

In this work, we propose a novel Amalgamated CNN-Transformer network (ACTSR) for lightweight SR, combining CNN with Transformer to simultaneously obtain local and long-term priors. Our ACTSR is composed of shallow feature extraction (SFE) module, amalgamated CNN and Transformer blocks (ACTBs), dense feature fusion (DFF) module and up-sampling module. Specifically, we use a 3×3 convolutional layer in SFE to extract shallow features that comprises rich low-frequency information. Then, series of ACTBs are utilized to extract hierarchical features, which are concatenated and fused with the output of SFE. In the end, SR images are reconstructed with the reconstruction module. The involving of spatial attention and self-attention enables our ACTSR to extract more effective features.

Our contributions are summarized below:

1) We propose an Amalgamated CNN-Transformer network for lightweight image SR (ACTSR), which outperforms other advanced lightweight methods.

2) We design an amalgamated CNN and Transformer block (ACTB) that exploits both local and long-range information to extract advantageous features for SR. Equipping self-attention and spatial attention mechanisms improve the efficiency of ACTB to capture informative features and further improve SR performance.

3) Extensive experiments have shown that our ACTSR outperforms advanced lightweight methods. Qualitative and quantitative comparisons show that our method generates more accurate SR results.

2. **Related work.** Recently, CNN-based methods have made impressive improvements in image SR research, especially with the introduction of attention mechanism [18, 19], including self-attention mechanism [13]. Hence, we will make a brief review on some of the typical CNN-based methods and attention mechanisms.

CNN-based networks. Dong et al. firstly proposes the SRCNN [1] by using a three-convolutional-layer CNN and obtains satisfying results. Then, VDSR [7] and Deeply-recursive convolutional network (DRCN) [20] further enhance SR results by building larger networks with residual scheme and recursive learning, respectively. Deep recursive residual network (DRRN) [21] employs recursive learning strategies and performs better even with smaller amount of parameters. A persistent memory network (MemNet) [22] is proposed to dispose the long-term dependency issue by mining persistent memory. Laplacian pyramid network (LapSRN) [23] reconstructs SR image by progressively upscaling image resolution and reconstructing sub-band residuals of HR images. Based on residual network (ResNet) [24], both SRResNet [25] and EDSR [8] stack a number

of residual blocks to boost network performance. Residual dense network (RDN) [9] introduces dense connection to completely utilize the features produced by each previous layer.

The performances of these methods commonly base on large model size, which is not convenient to deploy on mobile platforms. Therefore, lightweight models are proposed without compromising much of the model performance. A lightweight model cascading residual network (CARN-M) [26] exploiting group convolution operation has achieved comparable results with other advanced methods with fewer computation complexity and parameters. Information distillation network (IDN) [27] gradually extracts features from different paths and distill more informative features for SR reconstruction. Information multi-distillation network (IMDN) [28] proposes information multi-distillation strategy and achieves SOTA (state-of-the-art) performance. Residual feature distillation network (RFDN) [29] further devises an efficient module with feature distillation connections and shallow residual block, whose parameters are fewer but obtains better performance than IMDN.

Attention-based networks. Attention mechanism in deep learning network is to imitate the visual system of human beings to focus on significant features automatically, which achieves great success in varieties of vision tasks [30, 31]. RCAN [10] introduces a channel attention mechanism into simplified residual block to focus on the most important channels. Residual feature aggregation network (RFANet) proposes an enhanced spatial attention (ESA) module [29] to efficiently exploit spatial information in larger receptive field. Non-local recurrent network (NLRN) [32] , residual non-local attention network (RNAN) [33], and efficient non-local contrastive attention network (ENLCN) [34], introduce non-local attention mechanism to achieve performance improvement. Recently, Transformer-based SR models like [16, 35] introduce self-attention mechanism, to model long-range dependencies to further improve SR performance. Specially, SwinIR [16], a SR model based on Swin Transformer [15] has achieved excellent SR performance and outperforms the lightweight CNN-based methods. Hence, attention mechanism enables the deep learning network focusing on important information to improve the performance.

3. Method.

3.1. Overall network architecture.

Figure 1(a) shows the proposed Amalgamated CNN-Transformer network (ACTSR) that consists of shallow feature extraction (SFE), amalgamated CNN and Transformer blocks (ACTBs), dense feature fusion (DFF) and up-sampling module. Suppose the input LR is I_{LR}, we then extract shallow features

$$F_0 = H_{SF}(I_{LR}), \tag{1}$$

where H_{SF} denotes the 3×3 convolutional layer. F_0 is then fed into the stacked ACTBs. Supposed there are M ACTBs, the output of the m-th ACTB F_m $(1 \leq m \leq M)$ is expressed as

$$F_m = f^m_{ACTB}(f^{m-1}_{ACTB} \cdots ((f^1_{ACTB}(F_0)))), \tag{2}$$

where f^m_{ACTB} and F_m are the function and output of m-th ACTB, respectively. ACTBs is capable of extracting higher-level features and we will provide more details of ACTB in Section 3.2.

Then, the output features from ACTBs are concatenated and refined by DFF that comprises a 1×1 convolutional layer to fuse all the features and a 3×3 convolutional layer, and global residual connection is added to help training. Hence, the output of DFF is expressed as

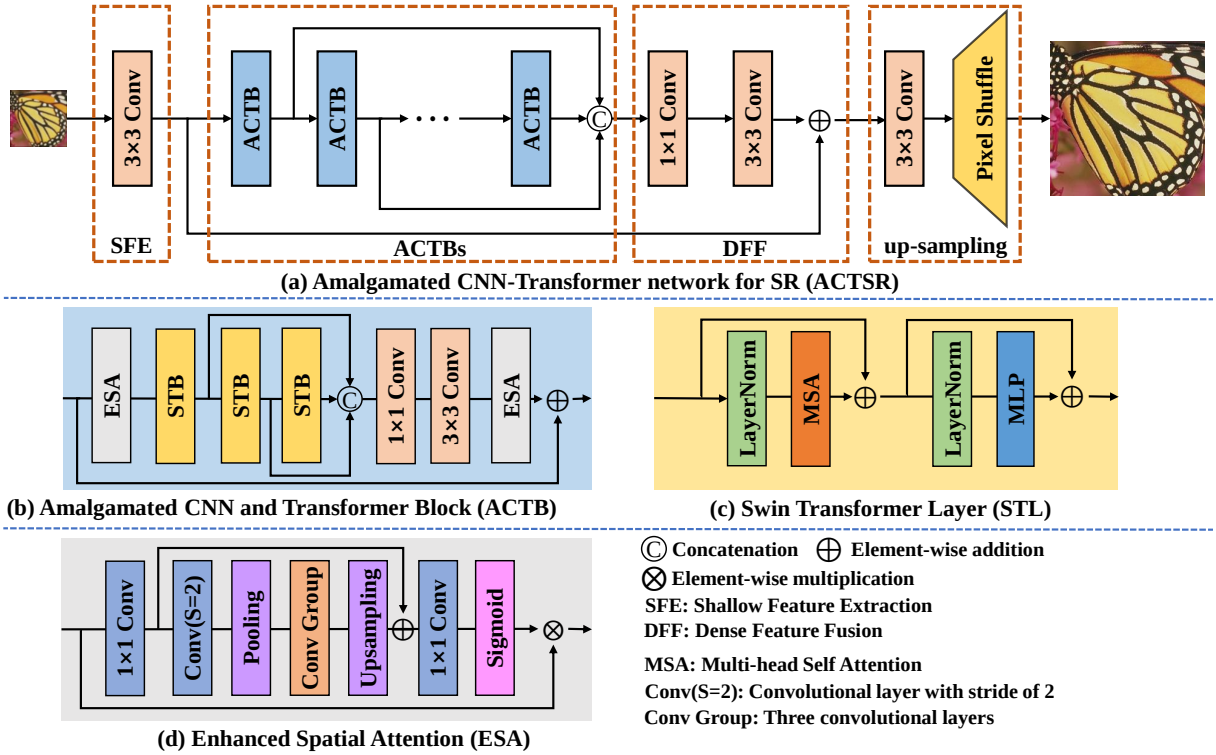

FIGURE 1. The overall framework of the proposed ACTSR. It is worth noting that An STB contains two STLs.

$$F_{DFF} = H_1 * (H_2 * Cat(F_1, F_2, \cdots, F_M)) + F_0, \tag{3}$$

where Cat$(\cdot)$ is the concatenation operation. H_1 and H_2 are the 3×3 and 1×1 convolutional layers, respectively.

Finally, SR image I_{SR} is reconstructed with up-sampling block constructed by a 3×3 convolutional layer and a pixel-shuffle layer [36], which is formulated as follows

$$I_{SR} = F_{UP}(H_3 * F_{DFF}), \tag{4}$$

where H_3 and F_{UP} denote the convolution layer and pixel-shuffle operation, respectively. To optimize our ACTSR, l_1 loss is employed and formulated as

$$L(\theta) = \frac{1}{N} \sum_{n=1}^{N} \|I^i_{SR} - I^i_{HR}\|_1, \tag{5}$$

where θ denotes the parameters to be optimized in ACTSR, N is the number of image patches for training. I^i_{SR} and I^i_{HR} are the i-th images reconstructed by the network and the corresponding ground-truth images, respectively.

3.2. **Amalgamated CNN and Transformer block.** As shown in Figure 1(b), the ACTB consists of several Swin Transformer blocks (STBs), a 1×1 convolutional layer, a 3×3 convolutional layer, two enhanced spatial attention (ESA) modules and cross residual connections. The feature maps of $(m-1)$-th ACTB are transferred to the m-th ACTB. The input features F_{m-1}, are first processed by the m-th ACTB to extract important features with an ESA module [29] and further refined by several STBs. The process of ACTB is formulated as

$$F_m = f^m_{ACTB}(F_{m-1}) = H_{ESA}(F_{STB}(H_{ESA}(F_{m-1}))), \tag{6}$$

where $H_{ESA}(\cdot)$ is the function of ESA. First, supposed there are N STBs in an ACTB, the output of the n-th STB $F_{m,n}$ $(1 \leq n \leq N)$ in d-th ACTB is formulated as

$$F_{m,n} = f^{m,n}_{STB}(f^{m,n-1}_{STB} \cdots ((f^{m,1}_{STB}(F_{m-1})))), \tag{7}$$

where $f^{m,n}_{STB}(\cdot)$ and $F_{m,n}$ are the function of n-th STB and the output of n-th STB in m-th ACTB, respectively. Then, F_{STB} uses features from all preceding STB layers and the output can be expressed as

$$F_{STB} = H_4 * (H_5 * Cat(F_{d,1}, F_{d,2}, \cdots, F_{d,N})), \tag{8}$$

where H_4 and H_5 are the 3×3 and 1×1 convolutional layers, respectively.

3.3. **Swin Transformer Block (STB).** As shown in Figure 1(c), we adopt the Swin Transformer layer (STL) proposed in [13], which involves local attention and shifted window mechanism. In ACTSR, we use two STLs to construct an STB to make a good trade-off between the network performance and complexity. As for an input image with size of $h \times w \times c$, Swin Transformer computes the matrices of query, key and value Q, K and $V \in R^{M^2 \times c}$ in a given local window feature $F^{swt}_{in} \in R^{M^2 \times c}$ as

$$Q = F^{swt}_{in} W_Q, K = F^{swt}_{in} W_K, V = F^{swt}_{in} W_V, \tag{9}$$

where W_Q, W_K and W_V denote shared learnable projection matrices across different windows, and d is the query dimension. The attention matrix $Attn(Q, K, V)$ is calculated from expression (10),

$$Attn(Q, K, V) = SoftMax(\frac{QK^T}{\sqrt{d}} + b)V, \tag{10}$$

where b is the relative positional encoding. The outputs of multi-head self-attention (MSA) are added with the inputs, which is denoted as F^{swt}_{inter}. Then, F^{swt}_{inter} is processed with a Layer Normalization (LN) layer and a multi-layer perceptron (MLP), the results of which is added with F^{swt}_{inter}t to generate the output F^{swt}_{out}. The whole function of Transformer is formulated as

$$\begin{cases} F^{swt}_{inter} = H_{MSA}(H_{LN}(F^{swt}_{in})) + F^{swt}_{in}, \\ F^{swt}_{out} = H_{MLP}(H_{LN}(F^{swt}_{inter})) + F^{swt}_{inter}, \end{cases} \tag{11}$$

where H_{LN}, H_{MSA}, and H_{MLP} denote LN, MSA and MLP functions, respectively. Since the step is of the shifted window is half of the window size, even number of stacked STLs are usually employed.

4. **Experiments.** In this section, we conduct a comprehensive analysis on the validation of our ACTSR. We compare ACTSR with other advanced lightweight models both quantitatively and qualitatively. Meanwhile, we conduct ablation studies to better understand the key components proposed in ACTSR.

4.1. **Experimental Setup. Datasets and Evaluation Metrics.** 800 images from DIV2K [37] dataset are used for training, from which LR-HR image pairs are generated by employing bicubic down-sampling algorithm. Meanwhile, Set5 [38] is used to validate our ACTSR during training. We randomly rotate the original dataset by 90°, 180°, 270° and flipping horizontally to implement data augmentation. Five commonly used public benchmark datasets including Set5 [38], Set14 [39], BSD100 [40], Urban100 [41] and Manga109 [42] are used as the testing datasets. Peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) are the quantitative evaluation metrics, which are calculated from the following expressions (12-13):

$$PSNR(x,y) = 10log_{10}(\frac{MAX^2}{MSE}), \tag{12}$$

where MSE is the mean square error between two images (x,y) and MAX is the maxinum pixel value.

$$SSIM = (\frac{2u_xu_y + C_1}{u_x^2 + u_y^2 + C_1})(\frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}), \tag{13}$$

where C_1, and C_2 are small constants. u_x and u_y are the average gray values of the two input images (x, y), respectively. σ_x and σ_y are standard deviations of x,y, respectively.

Training settings. In the training stage, there are 16 patches with size 64×64 in a mini-batch, which are obtained with random cropping from LR images. We exploit Adam optimizer to train ACTSR with $\beta1$=0.9, $\beta2$=0.999, and ϵ=1e-8, and the total training epoch is 1200. The learning rate is initialized to 5e-4, and halved after each 200 epochs at the first 1000 epochs. The window size, embedding dimension and head number of multi-head self-attention in STL are set as 8, 55 and 5, respectively. Specifically, the final architecture of ACTSR consists of four ACTBs and each STB contains two STLs.

4.2. **Ablation study. Effectiveness of ACTB.** To study the effectiveness of our proposed ACTB, we first replace the ESAs with 3×3 vanilla convolutional layers, which is marked as ACTSR-VC. We then replace the Transformer layers of ACTB with the original structure proposed in SwinIR [16], which is marked as ACTSR-OS. Table 1 shows that our ACTSR achieves best results, comparing with ACTSR-VC and ACTSR-OS, indicating the effectiveness of the structure of ACTB.

TABLE 1. Studies on the effectiveness of ACTB under Manga109 ×4 dataset. The best result is highlighted in bold.

Model	ACTSR	ACTSR-VC	ACTSR-OS
PSNR/SSIM	**31.17/0.9168**	31.17/0.9168	31.13/0.9158

Effectiveness of the number of ESAs in ACTB. We evaluate the effectiveness of the number of ESAs involved in ACTB on Manga109 dataset under scaling factor of 4, the results of which is listed in Table 2. For clarity, as shown in Figure 1(b), the first ESA of ACTB is marked as ESA-1 and the one at the end is ESA-2. The results show that when none of the ESAs is involved in ACTB, the network performs worst, and ACTB contains either of the ESA, the network performance is improved. The network achieves best performance if both ESAs are involved. Hence, the ESAs substantively contribute to the improvement of the network performance.

Effectiveness of the skip connection and feature concatenation. As shown in Figure 1(b), ACTB contains a skip connection to transfer the original feature information to the end of the block so as to avoid vanishing gradient problem, and connections to

TABLE 2. Studies on the effectiveness of the ESAs in ACTB under Manga109 ×4 dataset. The best result is highlighted in bold.

Model	ESA-1	ESA-2	PSNR/SSIM
ACTSR	✓	✓	**31.17/0.9168**
Model 1	✗	✗	31.02/0.9154
Model 2	✗	✓	31.07/0.9158
Model 3	✓	✗	31.10/0.9150

concatenate intermediate features extracted from STBs to keep the rich information for further excavation. The study results are shown in Table 3. We can observe that the proposed ACTSR performs best when both skip connection and feature concatenation are used in ACTB, indicating the effectiveness of these two connections.

TABLE 3. Studies on the effectiveness of skip connection and feature concatenation in ACTB under Manga109 ×4 dataset. The best result is highlighted in bold.

Model	Skip Connection	Feature Concatenation	PSNR/SSIM
ACTSR	✓	✓	**31.17/0.9168**
Model 4	✗	✗	31.06/0.9156
Model 5	✗	✓	31.03/0.9141
Model 6	✓	✗	31.13/0.9158

Effectiveness of the number of STB in ACTB. To study the effectiveness of the number of STB on the performance of the network, we vary the number of STB from 1 to 3 and test the performance of the corresponding models. Table 4 shows that the performance of our ACTSR is improved as the number of STB increases. To better balance the performance and mode size, we use 3 STBs in ACTB.

TABLE 4. Studies on the number of STB in ACTB under Manga109 ×4 dataset. The best result is highlighted in bold.

Model	Number of STB	Parameters	PSNR/SSIM
Model 7	1	457K	31.06/0.9156
Model 8	2	677K	31.03/0.9141
ACTSR	3	896K	**31.17/0.9168**

4.3. **Comparison with state-of-the-arts.** We compare our ACTSR with several advanced lightweight SR methods, including traditional Bicubic and deep learning models, such as SRCNN [1], VDSR [7], MemNet [22], IDN [27], CARN [26], LAPAR-A [43], IMDN [28], RFDN [29], LatticeNet [44], ELAN [46], ESRGCNN [45] and SwinIR [16]. Table 5 shows the PSNR/SSIM on the five benchmark datasets under scaling factors of 2, 3 and

TABLE 5. PSNR/SSIM on five benchmark datasets under scaling factors of 2, 3 and 4. The best and the second-best results are highlighted in red and blue, respectively.

Method	Scale	Params	FLOPs	Set5	Set14	BSD100	Urban100	Manga109
				PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Bicubic	×2	-	-	33.66/0.9299	30.24/0.8688	29.56/0.8431	26.88/0.8403	30.80/0.9339
SRCNN[1]		8K	52.7G	36.66/0.9542	32.45/0.9067	31.36/0.8879	29.50/0.8946	35.60/0.9663
VDSR[7]		666K	612.6G	37.53/0.9587	33.03/0.9124	31.90/0.8960	30.76/0.9140	37.22/0.9750
MemNet[22]		678K	2662.4G	37.78/0.9597	33.28/0.9142	32.08/0.8978	31.31/0.9195	37.72/0.9740
IDN[27]		553K	124.6G	37.83/0.9600	33.30/0.9148	32.08/0.8985	31.27/0.9196	38.01/0.9749
CARN[26]		1592K	222.8G	37.76/0.9590	33.52/0.9166	32.09/0.8978	31.92/0.9256	38.36/0.9765
IMDN[28]		694K	158.8G	38.00/0.9605	33.63/0.9177	32.19/0.8996	32.17/0.9283	38.88/0.9774
RFDN[29]		534K	95.0G	38.05/0.9606	33.68/0.9184	32.16/0.8994	32.12/0.9278	38.88/0.9773
LAPAR-A[43]		548k	171.0G	38.01/0.9605	33.62/0.9183	32.19/0.8999	32.10/0.9283	38.67/0.9772
LatticeNet[44]		756K	169.5G	38.15/0.9610	33.78/0.9193	32.25/0.9005	32.43/0.9302	38.94/0.9773
SwinIR[16]		878K	195.6G	38.14/0.9611	33.86/0.9206	32.31/0.9012	32.76/0.9340	39.12/0.9783
ESRGCNN[45]		1238K	312G	37.79/0.9589	33.48/0.9166	32.08/0.8978	32.02/0.9222	-/-
ELAN[46]		582K	168.4G	38.17/ 0.9611	33.94/0.9207	32.30/0.9012	32.76/0.9340	39.11/0.9782
ACTSR(Ours)		878K	190.2G	38.21/0.9612	33.92/0.9210	32.32/0.9013	32.75/0.9340	39.27/0.9781
Bicubic	×3	-	-	30.39/0.8682	27.55/0.7742	27.21/0.7385	24.46/0.7349	26.95/0.8556
SRCNN[1]		8K	52.7G	32.75/0.9090	29.30/0.8215	28.41/0.7863	26.24/0.7989	30.48/0.9117
VDSR[7]		666K	612.6G	33.66/0.9213	29.77/0.8314	28.82/0.7976	27.14/0.8279	32.01/0.9340
MemNet[22]		678K	2662.4G	34.09/0.9248	30.00/0.8350	28.96/0.8001	27.56/0.8376	32.51/0.9369
IDN[27]		553K	56.3G	34.11/0.9253	29.99/0.8354	28.95/0.8013	27.42/0.8359	32.71/0.9381
CARN[26]		1592K	118.8G	34.29/0.9255	30.29/0.8407	29.06/0.8034	28.06/0.8493	33.50/0.9440
IMDN[28]		703K	71.5G	34.36/0.9270	30.32/0.8417	29.09/0.8046	28.17/0.8519	33.61/0.9445
RFDN[29]		541K	42.2G	34.41/0.9273	30.34/0.8420	29.09/0.8050	28.21/0.8525	33.67/0.9449
LAPAR-A[43]		544k	114.0G	34.36/0.9267	30.34/0.8421	29.11/0.8054	28.15/0.8523	33.51/.09441
LatticeNet[44]		765K	76.3G	34.53/0.9281	30.39/0.8424	29.15/0.8059	28.33/0.8538	33.63/0.9441
SwinIR[16]		886K	87.2G	34.62/0.9289	30.54/0.8463	29.20/0.8082	28.66/0.8624	33.98/0.9478
ESRGCNN[45]		1500K	179G	34.24/0.9252	30.29/0.8413	29.05/0.8036	28.14/0.8512	-/-
ELAN[46]		590K	75.7G	34.61/0.9288	30.55/0.8463	29.21/0.8081	28.69/0.8624	34.00/0.9478
ACTSR(Ours)		886K	86.1G	34.67/0.9290	30.58/0.8467	29.23/0.8088	28.66/0.8626	34.21/0.9484
Bicubic	×4	-	-	28.42/0.8104	26.00/0.7027	25.96/0.6675	23.14/0.6577	24.89/0.7866
SRCNN[1]		8K	52.7G	30.48/0.8626	27.50/0.7513	26.90/0.7101	24.52/0.7221	27.58/0.8555
VDSR[7]		666K	612.6G	31.35/0.8838	28.01/0.7674	27.29/0.7251	25.18/0.7524	28.83/0.8870
MemNet[22]		678K	2662.4G	31.74/0.8893	28.26/0.7723	27.40/0.7281	25.50/0.7630	29.42/0.8942
IDN[27]		553K	32.3G	31.82/0.8903	28.25/0.7730	27.41/0.7297	25.41/0.7632	29.41/0.8942
CARN[26]		1592K	90.9G	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837	30.47/0.9084
IMDN[28]		715K	40.9G	32.21/0.8948	28.58/0.7811	27.56/0.7353	26.04/0.7838	30.45/0.9075
RFDN[29]		550K	23.9G	32.24/0.8952	28.61/0.7819	27.57/0.7360	26.11/0.7858	30.58/0.9089
LAPAR-A[43]		659K	94.0G	32.15/0.8944	28.61/0.7818	27.61/0.7366	26.14/0.7871	30.42/0.9074
LatticeNet[44]		777K	43.6G	32.30/0.8962	28.68/0.7830	27.62/0.7367	26.25/0.7873	30.54/0.9073
SwinIR[16]		897K	49.6G	32.44/0.8976	28.77/0.7858	27.69/0.7406	26.47/0.7980	30.92/0.9151
ESRGCNN[45]		1530K	113G	32.02/0.8920	28.57/0.7801	27.57/0.7348	26.10/0.7850	-/-
ELAN[46]		601K	43.2G	32.43/0.8975	28.78/0.7858	27.69/0.7406	26.54/0.7982	30.92/0.9150
ACTSR(Ours)		896K	49.4G	32.53/0.8988	28.85/0.7865	27.72/0.7410	26.55/0.8001	31.17/0.9168

4. The best and the second-best results are highlighted in red and blue, respectively. It is worth noting that the results of all the comparison methods are obtained from the original paper or codes provided by the authors. It shows that our proposed ACTSR performs best under all scaling factors on all datasets, except on Urban100 and Manga109 under scaling factor ×2. The quantitative comparison listed in Table 5 shows that SwinIR has similar parameters and computational complexity with our ACTSR under all cases, but its performance is inferior to ours. Hence, benefitting from the combination of CNN and Transformer, our proposed ACTSR substantially obtains the best results with smallest model size. Figure 2 shows qualitative visual comparisons between ACTSR and the other advanced lightweight models on BSD100, Urban100 ×3 and Urban100 ×4 datasets. From the zoomed-in views, we can observe clear textures and sharp edges reconstructed

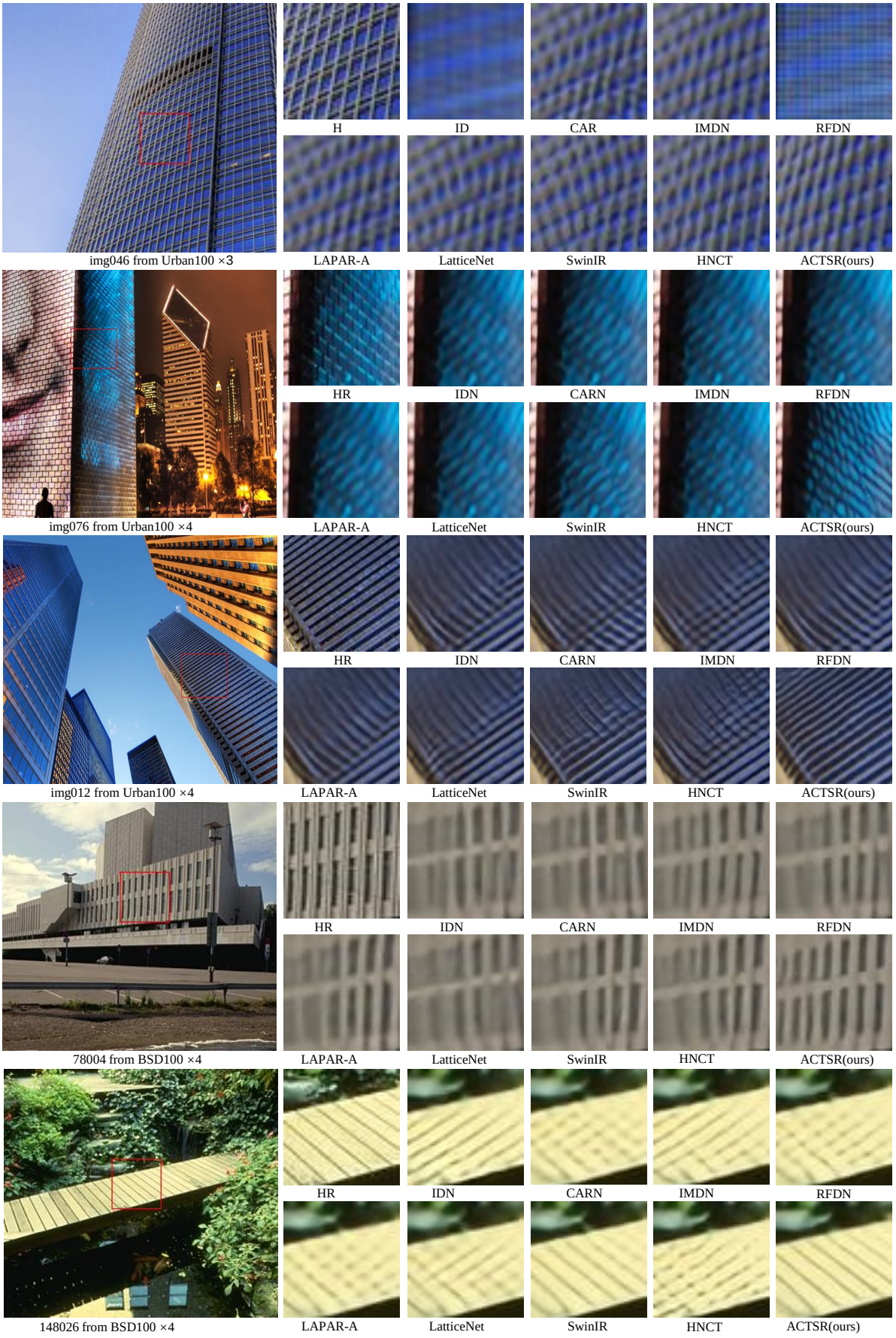

FIGURE 2. Visual comparisons of ACTSR with other advanced lightweight methods on Urban100 ×3, Urban100 and BSD100 ×4 datasets.

by ACTSR, which demonstrates the SR results of our method are more accurate than the comparison methods.

5. **Conclusion.** In this work, we have proposed an amalgamated CNN-Transformer network (ACTSR) for lightweight image super-resolution. By integrating CNN and Transformer, ACTSR is capable of exploiting both local and long-term priors and extracting features for image SR reconstruction. Extensive experiments demonstrate that ACTSR outperforms the lightweight competitors. However, since our ACTSR exploits the self-attention mechanism, it suffers from the intrinsic issue of heavy computation complexity as the other Transformer-based methods have confronted. Hence, our future work will focus on reducing the inference time.

Acknowledgment. This work was partly supported by Natural Science Foundation of Fujian Province (2021J011005) and Open Project of the Key Laboratory of Plasma and Magnetic Resonance in Fujian Province, Xiamen University (No.20191201).

REFERENCES

[1] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 184–199.

[2] Z. Wang, J. Chen, and S. C. Hoi, "Deep learning for image super-resolution: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3365–3387, 2020.

[3] Y. Ma, Y. Peng, and T.-Y. Wu, "Transfer learning model for false positive reduction in lymph node detection via sparse coding and deep learning," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 2, pp. 2121–2133, 2022.

[4] F. Zhang, T.-Y. Wu, Y. Wang, R. Xiong, G. Ding, P. Mei, and L. Liu, "Application of quantum genetic optimization of lvq neural network in smart city traffic network prediction," *IEEE Access*, vol. 8, pp. 104 555–104 564, 2020.

[5] T.-Y. Wu, X. Fan, K.-H. Wang, J.-S. Pan, and C.-M. Chen, "Security analysis and improvement on an image encryption algorithm using chebyshev generator," *Journal of Internet Technology*, vol. 20, no. 1, pp. 13–23, 2019.

[6] R. Wen, Z. Yang, T. Chen, H. Li, and K. Li, "Progressive representation recalibration for lightweight super-resolution," *Neurocomputing*, vol. 504, pp. 240–250, 2022.

[7] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654.

[8] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 136–144.

[9] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2472–2481.

[10] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel networks," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 286–301.

[11] Y. Qiu, R. Wang, D. Tao, and J. Cheng, "Embedded block residual network: A recursive restoration model for single-image super-resolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4180–4189.

[12] D. Song, Y. Wang, H. Chen, C. Xu, C. Xu, and D. Tao, "Addersr: Towards energy efficient image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 648–15 657.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.

[14] Y. Li, K. Zhang, J. Cao, R. Timofte, and L. Van Gool, "Localvit: Bringing locality to vision transformers," *arXiv preprint arXiv:2104.05707*, 2021.
[15] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 012–10 022.
[16] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "Swinir: Image restoration using swin transformer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1833–1844.
[17] J. Fang, H. Lin, X. Chen, and K. Zeng, "A hybrid network of cnn and transformer for lightweight image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2022, pp. 1103–1112.
[18] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.
[19] X. Zhu, D. Cheng, Z. Zhang, S. Lin, and J. Dai, "An empirical study of spatial attention mechanisms in deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6688–6697.
[20] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1637–1645.
[21] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3147–3155.
[22] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: A persistent memory network for image restoration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4539–4547.
[23] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 624–632.
[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
[25] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4681–4690.
[26] N. Ahn, B. Kang, and K.-A. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 252–268.
[27] Z. Hui, X. Wang, and X. Gao, "Fast and accurate single image super-resolution via information distillation network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 723–731.
[28] Z. Hui, X. Gao, Y. Yang, and X. Wang, "Lightweight image super-resolution with information multi-distillation network," in *Proceedings of the ACM International Conference on Multimedia*, 2019, pp. 2024–2032.
[29] J. Liu, J. Tang, and G. Wu, "Residual feature distillation network for lightweight image super-resolution," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 41–55.
[30] M. Guo, T. Xu, J. Liu, Z. Liu, P. Jiang, T. Mu, S. Zhang, R. Martin, M. Cheng, and S. Hu, "Attention mechanisms in computer vision: A survey," *Computational Visual Media*, vol. 8, no. 3, pp. 331–368, 2022.
[31] E. K. Wang, X. Zhang, F. Wang, T.-Y. Wu, and C.-M. Chen, "Multilayer dense attention model for image caption," *IEEE Access*, vol. 7, pp. 66 358–66 368, 2019.
[32] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang, "Non-local recurrent network for image restoration," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 31, 2018, pp. 1680–1689.
[33] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, "Residual non-local attention networks for image restoration," *arXiv preprint arXiv:1903.10082*, 2019.
[34] B. Xia, Y. Hang, Y. Tian, W. Yang, Q. Liao, and J. Zhou, "Efficient non-local contrastive attention for image super-resolution," *arXiv preprint arXiv:2201.03794*, 2022.
[35] W. Li, X. Lu, J. Lu, X. Zhang, and J. Jia, "On efficient transformer and image pre-training for low-level vision," *arXiv preprint arXiv:2112.10175*, 2021.

[36] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883.

[37] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 114–125.

[38] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," *British Machine Vision Conference*, pp. 1–10, 2012.

[39] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proceedings of the International Conference on Curves and Surfaces*, 2010, pp. 711–730.

[40] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings of the International Conference on Computer Vision*, 2001, pp. 416–423.

[41] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5197–5206.

[42] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa, "Sketch-based manga retrieval using manga109 dataset," *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21 811–21 838, 2017.

[43] W. Li, K. Zhou, L. Qi, N. Jiang, J. Lu, and J. Jia, "Lapar: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond," in *Proceedings of the Advances in Neural Information Processing Systems*, 2020, pp. 20 343–20 355.

[44] X. Luo, Y. Xie, Y. Zhang, Y. Qu, C. Li, and Y. Fu, "Latticenet: Towards lightweight image super-resolution with lattice block," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 272–289.

[45] C. Tian, Y. Yuan, S. Zhang, C.-W. Lin, W. Zuo, and D. Zhang, "Image super-resolution with an enhanced group convolutional neural network," *Neural Networks*, vol. 153, pp. 373–385, 2022.

[46] X. Zhang, H. Zeng, S. Guo, and L. Zhang, "Efficient long-range attention network for image super-resolution," in *Proceedings of the European Conference on Computer Vision*, 2022, pp. 649–667.