Remote Sensing Image Super-Resolution via Residual Aggregation and Split Attentional Fusion Network

Long Chen, Hui Liu, Minhang Yang, Yurong Qian, Zhengqing Xiao, and Xiwu Zhong

***Abstract*—Remote sensing images contain various land surface scenes and different scales of ground objects, which greatly increases the difficulty of super-resolution tasks. The existing deep learning-based methods cannot solve this problem well. To achieve high-quality super-resolution of remote sensing images, a residual aggregation and split attentional fusion network (RASAF) is proposed in this article. It is mainly divided into the following three parts. First, a split attentional fusion block is proposed. It uses a basic split–fusion mechanism to achieve cross-channel feature group interaction, allowing the method to adapt to various land surface scene reconstructions. Second, to fully exploit multiscale image information, a hierarchical loss function is used. Third, residual learning is used to reduce the difficulty of training in super-resolution tasks. However, the respective residual branch features are used quite locally and fail to represent the real value. A residual aggregation mechanism is used to aggregate the local residual branch features to generate higher quality local residual branch features. The comparison of RASAF with some classical super-resolution methods using two widely used remote sensing datasets showed that the RASAF achieved better performance. And it achieves a good balance between performance and model parameter number. Meanwhile, the RASAF's ability to support multilabel remote sensing image classification tasks demonstrates its usefulness.**

***Index Terms*—Remote sensing image, residual aggregation, split attentional fusion, super-resolution (SR).**

I. Introduction

REMOTE sensing images are commonly used in environmental monitoring, military, agriculture, and other fields due to their wide shooting range, free shooting time, and rich information. However, some factors such as weather and sensor noise in the imaging phase eventually affect remote sensing images, resulting in poor image quality and lowering the application value of remote sensing images. Considering the high cost and long period in improving hardware, the use of super-resolution (SR) methods to achieve an improved remote sensing image resolution has become the focus of research.

Single-image super-resolution (SISR) is the use of a low-resolution (LR) image to generate a high-resolution (HR) image using certain methods. However, SISR is a pathological inverse problem, where multiple corresponding HR images could exist for a single LR image. Many different algorithms have been created to solve this problem over years. The existing SISR methods can be classified as three types, namely, 1) interpolation-based methods; 2) reconstruction-based methods; and 3) learning-based methods. The interpolation-based methods include bilinear interpolation, Lanczos resampling [1], and edge-guided image interpolation [2]. These methods are very quick in reconstructing images and can almost meet real-time requirements, but they may lose high-frequency information and generate blurring images. The reconstruction-based method [3] uses efficient prior knowledge of LR and HR image pairs to minimize the solution space size. These methods generate sharp texture details, but they suffer from computational complexity and performance degradation when the magnification increases. The reconstruction precision is greatly improved when using learning-based methods. Freeman *et al.* [4], [5] incorporated new high-frequency information in LR images by modeling the relationship between HR and LR images using Markov networks. Chang *et al.* [6] proposed an SR method based on domain embedding. Yang *et al.* [7] proposed a sparse representation prior-based instance learning SR method. Ni *et al.*[8] proposed a vector regression-based instance learning method. Deep learning-based methods [9]–[11] have recently viewed the SR problem as an image-to-image regression problem, and have obtained results that are superior to conventional methods by using vast quantities of training data to create neural networks that directly learn the mapping relationship from LR to HR.

Some recent deep learning-based algorithms for remote sensing SR images [12]–[15] have shown promising results. The residual module plays a key role in these methods because the idea of residuals proposed by He *et al.* [16] effectively reduces the training difficulty of the model and allows a greater extension

Manuscript received June 8, 2021; revised September 5, 2021; accepted September 15, 2021. Date of publication September 20, 2021; date of current version October 6, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61966035, in part by the National Science Foundation of China under Grant U1803261, in part by the Xinjiang Uygur Autonomous Region Innovation Team under Grant XJEDU2017T002, and in part by the Autonomous Region Graduate Innovation Project under Grant XJ2020G074. *(Corresponding author: Yurong Qian.)*

Long Chen, Minhang Yang, Yurong Qian, and Xiwu Zhong are with the College of Software, Xinjiang University, Ürümqi 830008, China, with the Key Laboratory of Signal Detection and Processing in Xinjiang Uygur Autonomous Region, Ürümqi 830008, China, and also with the Key Laboratory of Software Engineering, Ürümqi 830008, China (e-mail: ry19chenlong@stu.xju.edu.cn; yangminhang@stu.xju.edu.cn; qyr@xju.edu.cn; xw.zhong@siat.ac.cn).

Hui Liu is with the College of Information Science and Engineering, Xinjiang University, Ürümqi 830014, China, with the Key Laboratory of Signal Detection and Processing in Xinjiang Uygur Autonomous Region, Ürümqi 830014, China, and also with the Key Laboratory of Software Engineering, Ürümqi 830008, China (e-mail: 903123414@qq.com).

Zhengqing Xiao is with the College of Mathematics and System Sciences, Xinjiang University, Ürümqi, China (e-mail: xiaozq@xju.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2021.3113658

of the depth of the network. The residual features progressively aggregate various aspects of the input image as the network depth increases, which is useful for reconstructing the spatial information of the image. However, the current method neglects the use of each residual branching feature. We use an improved residual aggregation mechanism to enhance feature extraction to solve this problem. Previous methods for natural images, on the other hand, do not perform as well with remote sensing images. Second, the remote sensing images are much larger in terms of visual angle. Different land surface scenes, such as houses, airports, beaches, and farmlands, could be included in one remote sensing picture. LR and HR images have vastly different texture and structure details, resulting in inconsistencies in mapping relationships between LR and HR images for various scenes. For the abovementioned problem, we propose the split attentional fusion block (SAFB). The channel interaction across feature groups is achieved by splitting the channel to generate feature groups of the same size and then using the attentional feature fusion mechanism to achieve two-by-two fusion of the feature groups. To complete the adaptive reconstruction of remote sensing images of various scenes, we improved the model's feature representation and generalization abilities. Meanwhile, we constructed a backbone network based on DRN [17] and generated images at multiple scales in upsampling step by step. A hierarchical loss is used to cope with the complex spatial distribution of remote sensing images and to reduce the difficulty of reconstruction. To sum up, our contribution consists of the following three aspects.

1) We proposed a new SAFB to replace the conventional spatial and channel attention layers which can effectively boost the model's performance with a few additional parameters.
2) We introduced the residual aggregation mechanism into the remote sensing image SR to aggregate the branches of residuals to generate better quality features and we defined a hierarchical loss to reduce the complexity of reconstruction.
3) We conducted the experiments to compare the residual aggregation and split attentional fusion network (RASAF) with some state-of-the-art models using the UCMerced_LandUse and PatternNet datasets in order to validate the improvement generated by the RASAF.

II. Related Work

A. Deep Learning in Single-Image SR

In recent years, SR methods based on deep convolutional neural networks (CNNs) have made significant progress. Dong *et al.* [18] are the first to introduce deep learning into SISR. They proposed the SRCNN, a three-layer CNN, which outperformed conventional approaches. Following that, FSRCNN [19] was proposed to make further changes by postsampling the upsampling, resulting in a significant reduction in computational workload. Meanwhile, the model's feature extraction capability was also improved by using smaller convolutional kernels and more mapping layers. LapSRN [20] predicted the residuals hierarchically by using stepwise upsampling. When performing large-scale reconstruction, it might yet produce intermediate-scale images. DBPN [21] proposed an iterative upsampling method that used the upper and lower sampling layers to provide an error feedback mechanism at each point to boost the SR effect. DRN [17] built a dual regression network based on UNet's [22] idea of encoder and decoder, and it created better reconstruction results.

Residual connections are crucial in the various CNN-based SR methods, which are mentioned above. ResNet [16] was proposed to deal with the issue of deeper network structure and the presence of untrained nodes, as well as to improve the model's efficiency. In addition, the residual block is commonly used in a variety of tasks. VDSR [23] indicated the similarity in low-frequency information between the input LR image and the output HR image. Only the high-frequency partial residuals between LR and HR should be learned. SRDenseNet [24] built a network based on dense blocks in DenseNet [25]. All subsequent layers were fed with the features from each layer of the dense block. This architecture solved the network's gradient disappearance problem and encouraged feature reuse to improve feature propagation. RDN [26] proposed residual dense block (RDB), which combined residual block and dense block and improved them. ESRGAN [27], on the other hand, employed residual in residual dense block (RRDB), which was a more advanced version of RDB with a greater number of parameters and higher calculation cost, yet improved the efficiency. Aggregating local residual features in deep residual networks using a feature aggregation model, RFANet [28], yielded a better feature representation, and experimentally outperformed RRDB and RDB structures.

B. Attention Mechanism

CNN is multidimensional, and by assigning different weights in spatial dimension and channel dimension, the neural network can be made to focus on the focal features and ignore the irrelevant information. The attention mechanism was originally applied in machine translation. It is now widely used in computer vision [28], [29]. Hu *et al.* [30] proposed a squeeze–excitation block. By modeling the interdependencies between channels, it adaptively recalibrated the feature response in terms of channels. Zhang *et al.* [29] brought this idea to the SR domain to construct a very deep residual network (RCAN), which achieved better SR performance. CBAM [31] considered a tandem spatial and channel attention module to achieve simultaneous focusing on both spatial and channel dimensions of the network, helping networks understand "what" to focus on and "where" to focus. Attention mechanisms are used to enhance the feature representation capability of neural networks: focus on important features and discourage unnecessary ones. Using the features of the two branches of channel attention fusion, Li *et al.* [32] proposed a new way of manipulating the attention process. Dai *et al.* [33] suggested a multiscale attentional feature fusion module to replace the conventional addition and contact to achieve feature fusion. ViT [34] introduced a transformer from the NLP domain to computer vision. It used a self-attentive framework instead of the conventional convolutional architecture to construct a network. The use of attention mechanism is expanding.

C. Deep Learning in Remote Sensing SR

Remote sensing SISR has attracted a lot of attention in the last several years. One of the low-level computer vision tasks, remote sensing SISR, can be applied as a preprocessing phase for high-level computer vision tasks (such as remote sensing image classification, changes detection, and semantic segmentation). For high-level tasks, the reconstructed high-quality output can effectively improve visual identification and understanding. Some early work directly used the SISR algorithm for simple images to remote sensing images, and the results were promising. Ducournau *et al.* [15] applied SRCNN [18] on ocean remote sensing data and reaped great benefits in terms of PSNR compared to traditional methods. Because they have been proven and have a strong reconstruction capability, several classical SISR algorithms [23], [29] are used as baseline methods by some recent remote sensing SISR [35], [36]. However, simple images, such as skyscrapers with nearly identical individual windows, have more repeating texture information. Some scenes such as faces have obvious prior information. Remote sensing images, on the other hand, are more complicated, with significant changes in image information for different scenes and larger scale differences between individual objects. Therefore, the characteristics of the remote sensing images themselves are taken into account in the later work. LGCNet [14] uses a "multi-fork" structure to learn multiscale information from remote sensing data. WTCRR [13] considered image reconstruction from the frequency domain, using wavelet transform for LR and then recursive ResNet for SR prediction. SMSR [35] took into account the highly complex spatial distribution of remote sensing images, captured multiscale information by aggregating features of different depths in a single path, and proposed a second-order learning mechanism that reuses large- and small-discrepancy features both globally and locally. Some recent work has started to build base modules based on remote sensing image characteristics. RDBPN [37] based on the proposed residual inverse projection block for the construction of the network, combining the advantages of global and local learning and achieved good results in the SR task with large-scale factors. Zhang *et al.* [36] designed a mixed high-order attention network (MHAN), which introduced high-order attention (HOA) to the SR domain and achieved a large improvement inaccuracy. Growing work has begun to explore the characteristics of remote sensing images and refine the feature representation of the network.

III. Method

A. Network Architecture

Fig. 1 depicts the network, which is functionally divided into four sections: 1) heads, 2) downsampling, 3) upsampling, and 4) reconstruction.

Heads: Using $f_{\text{up}}(\cdot)$, the I_{LR} is upsampled to the target image scale, i.e., bicubic interpolation, and then fed into the network for shallow feature extraction. $\text{Conv}^{(c,k,f)}$ is a convolution layer with c equaling the number of input channels, k equaling the convolutional kernel size, and f equaling the number of convolutional kernels. To complete the initial feature extraction $f_{\text{initial}}(\cdot)$

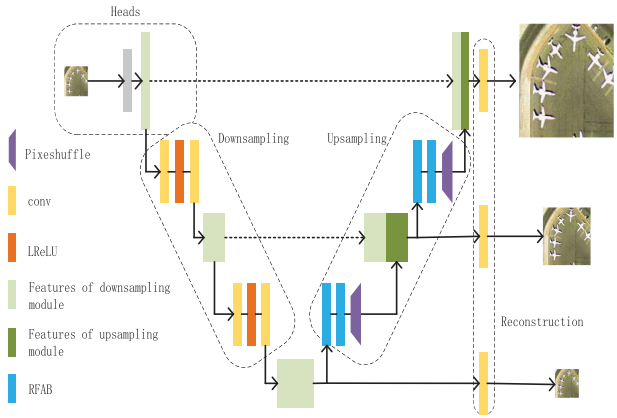

Fig. 1. Architecture of the proposed RASAF.

of the input image to output feature F_0, a 3×3 convolution was used

$$F_0 = f_{\text{up}}\left(f_{\text{intial}}\left(I_{\text{LR}}\right)\right) = f_{\text{up}}\left(\text{Conv}^{(c,k,f)}\left(I_{\text{LR}}\right)\right). \quad (1)$$

Downsampling: The output features F_1 and F_2 are downsampled in two stages after the initial feature extraction is completed, $f_{\text{down}}(\cdot)$ is used to implement each step, which consists of two 3×3 convolutions and a $\text{ReLU}(\cdot)$ activation function

$$\begin{aligned} F_1 &= f_{\text{down1}}\left(F_0\right) \\ &= \text{Con}\,v^{(32,3,32)}\left(\text{Re}\,\text{LU}\left(\text{Conv}^{(16,3,32)}\left(F_0\right)\right)\right) \end{aligned} \quad (2)$$

$$\begin{aligned} F_2 &= f_{\text{down2}}\left(F_1\right) \\ &= \text{Con}\,v^{(32,3,32)}\left(\text{Re}\,\text{LU}\left(\text{Conv}^{(16,3,32)}\left(F_1\right)\right)\right). \end{aligned} \quad (3)$$

Upsampling: The residual aggregation block (RFAB) is used to perform deep feature learning, which is then output to the upsampling block. The number of base blocks RFAB can be changed to adjust the network's depth

$$\begin{aligned} F_3 &= f_{\text{up2}}\left(\text{RFAB}_t\left(F_{t-1}\right)\right) \\ &= f_{\text{up2}}\left(\text{RFAB}_t\left(\text{RFAB}_{t-1}\left(\ldots\left(\text{RFAB}_0\left(F_2\right)\right)\right)\right)\right) \end{aligned} \quad (4)$$

$$\begin{aligned} F_4 &= f_{\text{up2}}\left(\text{RFAB}_t\left(F_{t-1}\right)\right) \\ &= f_{\text{up2}}\left(\text{RFAB}_t\left(\text{RFAB}_{t-1}\left(\cdots\text{RFAB}_0\left(F_3 + F_2\right)\right)\right)\right). \end{aligned} \quad (5)$$

Finally, $f_{\text{up2}}(\cdot)$, which is a subpixel convolution layer [11], upsamples the performance of F_t. Upsampling is performed using two stages to output features F_3 and F_4, respectively. The input of the first stage is the downsampled feature F_2. To reduce the difficulty of training, while taking into account the rich structural information contained in the LR images, adding a residual connection for interaction between HR and LR images allows the feature extraction part of the method to concentrate on extracting high-frequency information from the image. As a result, the second stage's input features are the sum of features F_2 and F_3.

Reconstruction: Differently from traditional networks, RASAF can generate multiscale images, which helps to utilize the information in the intermediate scale of images and reduces

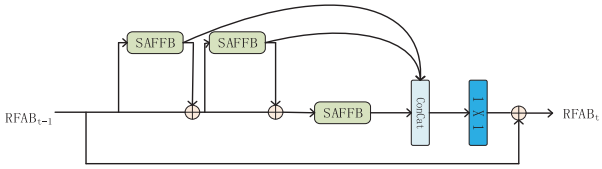

Fig. 2. Architecture of the improved RFAB.

the difficulty of reconstruction

$$I_{\mathrm{SR1}} = f_{\mathrm{re}}\left(F_2\right) = \mathrm{Con}\, v^{(64,3,3)}\left(F_2\right) \tag{6}$$

$$I_{\mathrm{SR2}} = f_{\mathrm{re}}\left(F_3 + F_2\right) = \mathrm{Con}\, v^{(64,3,3)}\left(F_3 + F_2\right) \tag{7}$$

$$I_{\mathrm{SR4}} = f_{\mathrm{re}}\left(F_4 + F_3\right) = \mathrm{Con}\, v^{(32,3,3)}\left(F_4 + F_3\right). \tag{8}$$

The reconstruction is completed using 3×3 convolution. $I_{\mathrm{SR1}}, I_{\mathrm{SR2}}$, and I_{SR4} are the generated SR image, the reduced twice and four times SR images, respectively.

B. Residual Aggregation Block

Since the advent of ResNet [17], the residual block has been increasingly used in SISR. A common method for designing networks is to stack residual blocks. The low-level residual features, on the other hand, must travel a long way before reaching the final part of the network. A large number of convolution and addition operations in this process make these features no longer significant so that the residual features are not fully utilized.

To overcome the problem of moving shallow residual features, ESRGAN [27] and RDN [26] combined dense connectivity and residual modules, but this method greatly increased the number of parameters. We used an improved residual feature aggregation block inspired by RFANet to solve this problem, adding just a few parameters to achieve results that outperform RRDB and RDB. Fig. 2 depicts the block's details, which are made up of three residual blocks and a 1×1 convolution, with split attentional feature fusion block (SAFFB) standing for our changed residual block, which is defined in Section III-C. This block first outputs the first two blocks' residual branch features directly to the end of the block, then the third residual block overwrites the output of the first two residual blocks, then the features of the three residual branches are stitched together and downscaled and aggregated using a 1×1 convolution. And finally, the aggregated features are fused using a skip join. In this way, shallow features can be passed directly to the end without losses due to intermediate operations.

C. Split Attentional Fusion

Multipath representation has been successful on GoogleNet [38]. On separate routes, multiscale convolutional kernels are used, and the final stitching completes the fusion of features at different scales. ResNeXt [39] combines ResNet and Inception to transform multiple paths into a single operation by using group convolution, then using the same topology at each branch. SKNet [32] applied attention to two branches to achieve cross-feature graph attention. ResNeSt [40] introduced this way of attention across feature maps to group convolution with good results. The abovementioned approach performed

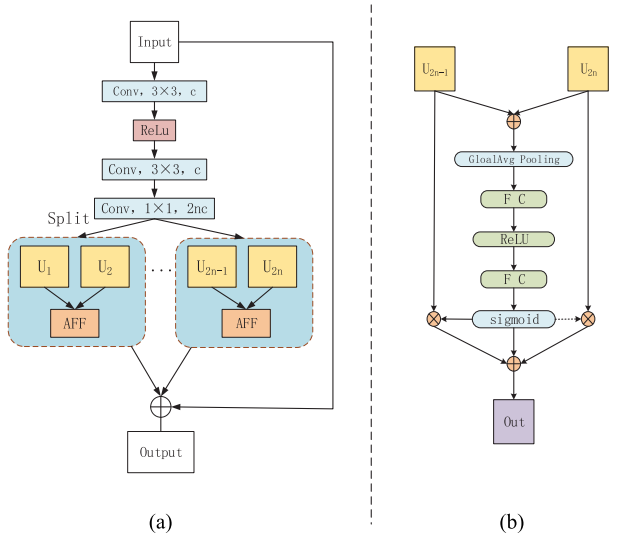

Fig. 3. Architecture of SAFFB, based on the union of our proposed split attentional feature fusion (SAFF) and residual blocks.

the corresponding convolutional operations in each branch and required a large width of the network to have good results. In the field of SR, however, RCAN [29] has shown that for the same number of parameters, depth has a greater effect on the network than width. To achieve this kind of attention to feature maps around the pass, we consider using a simple split–fusion technique, which can achieve an efficient boost with just a slight increase in the number of parameters.

Fig. 3(a) depicts the proposed SAFFB. There are three main components: 1) the base convolution block, 2) the splitting block, and 3) the attentional feature fusion block. The underlying convolutional block is the same as what in a traditional residual network, i.e., a 3×3-sized convolutional kernel and $\mathrm{ReLU}(\cdot)$ activation feature. F_{in} is used as the initial feature input and F_{base} is the output feature

$$F_{\mathrm{base}} = \mathrm{Conv}^{(c,k,f)}\left(\mathrm{ReLU}\left(\mathrm{Conv}^{(c,k,f)}\left(F_{\mathrm{in}}\right)\right)\right). \tag{9}$$

Split block: Similar to ResNeSt [40], we first increased the number of feature channels using a 1×1 convolution, then split the features into $2n$ feature groups, and then combined every two feature groups into a base group

$$F_{\mathrm{sp}} = \mathrm{Split}\left(\mathrm{Conv}^{(c,1,2\mathrm{nc})}\left(F_{\mathrm{base}}\right)\right) = U_1, U_2, \ldots U_{2n} \tag{10}$$

where $\mathrm{Split}(\cdot)$ is the splitting function, which is used to achieve the splitting of the feature map in the channel latitude, and the split U_{2i} has the same size as F_{base} .

Attentional feature fusion block: The detailed architecture is shown in Fig. 3(b). Different from SKNET [32] and ResNeSt [40], we used attentional feature fusion to achieve attentional interaction across channels. It is a unified and universal feature fusion scheme proposed by Dai [34] to replace the traditional feature fusion methods such as plus as well as contact. We improved the design and used it as a replacement for the split attention block. Our attentional feature fusion block, as compared to split attentional block, is more capable of achieving cross-channel attentional interaction with two branches and

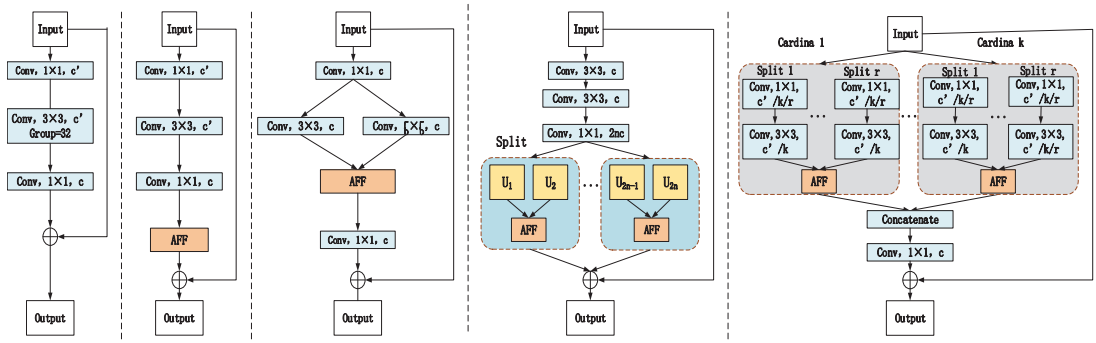

Fig. 4. Architectures of ResNeXt, SENet, SKNet, SAFFB, and ResNeSt.

requires less computational effort

$$\partial = f_{\mathrm{CA}}\left(U_{2n-1} + U_{2n}\right)$$
$$= f_{\mathrm{sigmoid}}\left(f_{\mathrm{FC}}\left(\mathrm{ReLU}\left(f_{\mathrm{FC}}\left(f_{\mathrm{gap}}\left(U_{2n-1} + U_{2n}\right)\right)\right)\right)\right) \quad (11)$$

$$f_{\mathrm{AFF}}\left(U_{2n-1}, U_{2n}\right) = U_{2n-1} \times \partial + U_{2n} \times (1 - \partial) \quad (12)$$

$$F_{\mathrm{filsion}} = \sum_{i=1}^{n} f_{\mathrm{AFF}}\left(U_{2i-1}, U_{2i}\right) \quad (13)$$

where $F_{\mathrm{CA}}(\cdot)$ represents the channel attention mechanism, $F_{\mathrm{gap}}(\cdot)$ is the global average pooling, $F_{\mathrm{FC}}(\cdot)$ is the fully connected layer, and $F_{\mathrm{signoid}}(\cdot)$ is the sigmoid activation function. In (11), we used the channel attention mechanism to obtain the weight (∂) coefficients of the fused image at each channel position. In (12), we used this coefficient to assign weights to the features on the input double branch. As (1-∂) is used, it removes the need to calculate channel weights for features on both branches, resulting in a significant reduction in computational effort.

Fig. 4 compares SAFFB to ResNeXt [39], SENet [30], SKNet [32], and ResNeSt [40], with the squeeze–excitation module of SENet being a particular case of AFF with only one branch module. Both SAFFB and SENet calculate the weights of different channels as a way to achieve focus on different channels. SAFFB improves SENet's squeeze–excitation mechanism by refining it to feature map groups, allowing it to pay attention to different channels at higher latitudes and allowing feature map groups to interact. In ResNeXt, group convolution reduces computational cost, but it also tends to diminish accuracy. SAFFB absorbs the advantages of grouped convolution and combines the channel attention mechanism and feature map groups to retain the advantages of parallel computation and improve feature representation capability. SKNet uses different size convolutional kernels in the two branches and achieves adaptive adjustment of the receptive domain to the input information by fusing the two branches through an attention mechanism. The difference between this multiscale information is not obvious because the perceptual fields on the two branches do not differ much, while there are cases where the remote sensing images differ greatly in the scale of different ground objects. To achieve multiscale feature extraction, we combine residual modules of various depths on a single path. We employ a hierarchical loss function to achieve multiscale information utilization at the same time. As a result, SAFFB may concentrate more on local learning. ResNeSt optimizes the residual module from the perspective of the feature map group; thus, SKNet can be seen as a subset of ResNeSt. In terms of feature map groups, ResNeSt has more convolutional calculations, which raises the computational cost of the model significantly, whereas SAFFB is more focused on feature refinement extraction and channel latitude representation.

D. Loss Function

Most of the previous models use the generated SR and HR for loss calculation, but RASAF can generate multiple task images at once, and therefore uses a hierarchical loss function for network optimization. Comparing the $\times 4$, $\times 2$, and $\times 1$ SR images and the target images to calculate the losses, respectively, and constraining the generated images at different scales can effectively speed up the model convergence. Considering also that the ground objects of remote sensing images have large-scale differences, feature extraction at different scales can help to improve the reconstruction effect of small objects. Our loss function is defined as follows for a given training dataset containing a number of (denoted by *N*) LR image patches

$$\ell(\Theta) = \frac{1}{\mathrm{N}} \sum_{j=1}^{N} \left\| G\left(I_{\mathrm{LR}}^{j}\right) - I_{\mathrm{HR1}}^{j} - I_{\mathrm{HR2}}^{j} - I_{\mathrm{HR4}}^{j} \right\| \quad (14)$$

where $G(\cdot)$ is RASAF and $I_{\mathrm{LR}}^{j}(j = 1, 2, 3, \ldots, n)$ is the jth LR image. I_{HR1}^{j}, I_{HR2}^{j}, and I_{HR4}^{j} are HR images downsampled twice and four times, respectively.

IV. Experiment

This section specifies the details and results of the experiments. We evaluated the performance of the models using the peak signal-to-noise ratio (PSNR) and structural similarity ratio (SSIM), which are commonly used in image quality evaluation [12]–[15]. The time taken for an epoch in the training process (Time) is used to measure the running speed of the model, and the number of parameters of the model (Parameters) is used to measure the size of the model.

A. Settings

Dataset: We used the UCMerced_LandUse and PatternNet datasets in our experiments, which are common dataset used

TABLE I
ABLATION STUDIES TO VERIFY THE EFFECTIVENESS OF RESIDUAL AGGREGATION MECHANISM AND SPLIT ATTENTIONAL FUSION

Name	RB	RB-CA	RB-ESA	RB-SAFF	RB-RFA	RB-RFA-CA	RB-RFA-ESA	RB-RFA-SAFF	RB-RFA-SAFF
Residual Block(RB)	√	√	√	√	√	√	√	√	√
Residual Feature Aggregation(RFA)					√	√	√	√	√
Enhanced Spatial Attention Block(ESA)			√				√		
Channel Attention Block(CA)		√				√			
Split Attentional Feature Fusion Block(SAFF)				√				√	√
Hierarchical Loss	√	√	√	√	√	√	√		√
PSNR(dB)	29.31	29.31	29.25	29.37	29.35	29.32	29.31	29.38	**29.41**
SSIM	0.795	0.7955	0.792	0.7962	0.7964	0.7943	0.7946	0.7962	**0.7987**
Time(s)	**62.5**	68.7	69.6	72.15	66.7	71.1	74.25	76	76.75

in previous remote sensing SR methods. In addition, given the limited number of available remote sensing images, the training dataset is expanded by randomly rotating 90°, 180°, 270°, and mirroring. The HR images were downsampled with a scale factor of 4 using a bicubic interpolation algorithm in the MATLAB setting to produce LR images.

1) *UCMerced_LandUse:* The dataset was released by the University of California in 2010, according to the source GoogleEarth. It includes 21 categories of remote sensing scenes such as agriculture, aircraft, baseball, diamonds, and beaches. There are 100 images for each category. All images are 256 × 256 pixels with a spatial resolution of 0.3 m/pixel. We randomly selected 40% of the images for training and 5% for testing.
2) *PatternNet:* The dataset was released by Wuhan University in 2018 and the data source is GoogleMap. It includes 38 categories of remote sensing scenes such as overpasses, golf courses, oil wells, parking lots, and railroads. There are 800 pictures for each category. All images have 256 × 256 pixels and a spatial resolution of 0.06 4.7 m/pixel. We randomly selected 100 training images and five test images from each scene.

Details of the experiment: By randomly extracting 96 × 96 image patches from LR images as input, all inputs and outputs are RGB images with each small batch size of 16. The parameters of the model were updated using the Adam optimization method, where $\beta 1$ was 0.9 and $\beta 2$ was 0.999. The initial learning rate is 5×10^{-4} and was reduced by half every 100 epochs, for a total of 500 epochs trained. We implemented all the experiments using PyTorch and tested them on a Tesla V100 device.

B. Ablation Study

Based on the UCMerced_LandUse dataset, we conducted ablation experiments to verify the efficacy of all blocks in this section, and we used PSNR and SSIM as evaluation metrics. The baseline model is the DRN [17], which includes 60 RCAB blocks in the upsampling part and has the regression blocks removed. For a fair comparison, we used 20 RFA blocks to maintain the same amount of residual blocks as DRN-Baseline. As shown in Table I, we compared the CA, ESA, and SAFF blocks without using the residual aggregation mechanism, and SAFF block achieved higher PSNR and SSIM, with PSNR 0.06 dB higher than that of the CA block, demonstrating the SAFF block's powerful feature representation capability. Second, when the RFA block was used alone, the PSNR increases by 0.04 dB if compared to the case without the RFA block. When RFA is combined with CA or ESA, the PSNR were improved by 0.01 and 0.06 dB, respectively, when compared to the original time before the RFA block was added. This shows that the residual aggregation mechanism can enable the effective aggregation of local residual features to produce better feature representations by enhancing the effective utilization of residual features. Further, when RFA and SAFF were combined, the best PSNR was obtained, showing that the two together can effectively improve network performance. By comparing the time, we can see that the SAFF's running speed is inefficient. In comparison to the CA, SAFF must calculate channel weights twice and adds a 1 × 1 convolution operation. However, the time difference is not significant, at about 5% when compared to the CA. Finally, we show that when the hierarchical loss function is used, the model's performance improves by 0.03 dB, and the influence of the hierarchical loss function on training time is essentially negligible; thus, it is required to apply the hierarchical loss function cost-effectively.

TABLE III
COMPARISON WITH OTHER SIMILAR METHODS

	ResNeXt	SENet	SKNet	ResNeSt	Our
Parameters(M)	3.66	3.32	**1.25**	6.75	6.06
Time(s)	72.03	**53.28**	58.53	76.48	76.75
PSNR(dB)	29.14	29.20	29.25	29.17	**29.41**
SSIM	0.7893	0.7910	0.7911	0.7902	**0.7987**

TABLE II
ABLATION STUDIES TO VERIFY THE EFFECT OF CARDINALITY ON MODEL PERFORMANCE

Name	Cardinality	Memory	Parameters	Time	PSNR
SAFF2	2	6033M	5.53M	68.87s	29.38dB
SAFF4	4	7467M	6.06M	76.75s	29.41dB
SAFF6	6	8795M	6.59M	100.61s	**29.44dB**
SAFF8	8	10133M	7.11M	125.96s	29.37 dB

To verify the effect of cardinality on model performance in the SAFB, an ablation study was set up based on RASAF. The cardinality is set to 2, 4, 6, 8, and the rest was kept unchanged, and the final results are shown in Table II. It is discovered that the model's performance was increased steadily at first, but that this improvement had a peak at cardinality 6, and the PSNR began to decline at cardinality 8. This shows that more cardinality is not better, because an increase in cardinality also increases the difficulty of training. In addition, as the number of branches grows, the amount of time required for training grows as well, limiting the number of branches available to some extent. Although the best PSNR was obtained with a final cardinality of 6, we ultimately chose 4 as the final cardinality of the model, based on the number of parameters, memory consumption, time, and performance, and the subsequent experiments were performed on this basis.

TABLE IV
AVERAGE PSNR AND SSIM FOR VARIOUS CLASSES OF GROUND OBJECTS BY DIFFERENT METHODS USING UCMERCED_LANDUSE DATASET WITH SCALE FACTOR $\times 4$

Method	Bicubic	SRCNN	SRResnet	RDN	EDSR	RCAN	DRN	RASAF	RASAF+
	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
agricultural	25.98/0.4665	26.29/0.4905	27.97/0.6885	28.23/0.6820	27.37/0.6590	28.28/0.6822	27.41/0.6425	28.33/0.6860	**28.34/0.6863**
airplane	26.62/0.7473	28.47/0.7896	30.46/0.8251	30.72/0.8293	30.64/0.8264	30.80/0.8300	30.88/0.8317	31.12/0.8341	**31.21/0.8346**
baseballdiamond	33.59/0.8533	34.48/0.8689	35.26/0.8845	35.32/0.8851	35.23/0.8845	35.40/0.8860	35.47/0.8871	35.49/0.8868	**35.58/0.8882**
beach	34.06/0.8512	34.63/0.8638	35.19/0.8756	35.19/0.8764	35.18/0.8759	35.20/0.8774	35.20/0.8780	35.22/0.8786	**35.29/0.8790**
buildings	24.69/0.7223	26.26/0.7792	28.17/0.8340	28.28/0.8356	28.22/0.8346	28.39/0.8385	28.31/0.8382	28.61/0.8431	**28.74/0.8432**
chaparral	25.24/0.5903	25.93/0.6437	26.70/0.6899	26.57/0.6900	26.66/0.6906	26.55/0.6915	26.76/0.6964	26.78/0.6986	**26.84/0.7010**
denseresidential	25.70/0.7360	27.14/0.7923	28.85/0.8468	28.90/0.8482	28.91/0.8483	28.98/0.8509	29.12/0.8540	29.17/0.8559	**29.26/0.8572**
forest	27.64/0.5714	27.95/0.6102	28.06/0.6280	27.98/0.6267	28.05/0.6288	28.03/0.6293	28.08/0.6317	**28.10/0.6323**	**28.10**/0.6320
freeway	29.01/0.8032	30.24/0.8363	32.58/0.8804	32.83/0.8849	32.73/0.8816	32.73/0.8839	32.95/0.8900	32.82/0.8853	**33.15/0.8925**
golfcourse	31.57/0.7881	32.61/0.8115	33.17/0.8255	33.19/0.8269	33.23/0.8271	33.26/0.8277	33.31/0.8287	33.33/0.8294	**33.36/0.8294**
harbor	20.93/0.6823	22.18/0.7576	24.03/0.8408	24.31/0.8520	24.24/0.8461	24.40/0.8527	24.57/0.8588	24.64/0.8626	**24.82/0.8650**
intersection	26.15/0.7196	26.90/0.7529	27.99/0.8015	28.09/0.8083	28.00/0.8035	28.15/0.8095	28.20/0.8099	28.18/0.8113	**28.43/0.8147**
mediumresidential	24.95/0.6518	25.92/0.7036	26.73/0.7425	26.77/0.7465	26.75/0.7450	26.81/0.7475	26.94/0.7512	26.98/0.7537	**27.12/0.7552**
mobilehomepark	22.50/0.6214	23.86/0.6915	25.19/0.7443	25.28/0.7483	25.21/0.7465	25.34/0.7503	25.55/0.7587	25.60/0.7598	**25.68/0.7616**
overpass	25.45/0.6882	26.31/0.7278	28.21/0.7953	28.51/0.8061	28.53/0.8035	28.59/0.8063	28.51/0.8068	28.64/0.8070	**28.94/0.8134**
parkinglot	21.09/0.6426	22.01/0.7095	23.61/0.7926	24.08/0.8084	23.89/0.8023	24.10/0.8092	24.16/0.8139	24.31/0.8170	**24.36/0.8177**
river	27.65/0.6618	28.14/0.6949	28.39/0.7124	28.40/0.7133	28.38/0.7129	28.33/0.7126	28.44/0.7147	28.45/0.7148	**28.51/0.7158**
runway	30.12/0.7801	31.36/0.8037	33.96/0.8426	33.99/0.8422	34.09/0.8439	34.18/0.8460	34.18/0.8458	34.19/0.8467	**34.29/0.8471**
sparseresidential	28.24/0.6954	28.99/0.7279	29.69/0.7567	29.78/0.7598	29.76/0.7584	29.77/0.7603	29.89/0.7620	29.95/0.7634	**30.02/0.7641**
storagetanks	25.30/0.6674	26.22/0.7110	27.46/0.7664	27.61/0.7744	27.53/0.7694	27.75/0.7785	27.72/0.7779	27.75/0.7765	**27.88/0.7800**
tenniscourt	27.53/0.7319	28.48/0.7668	29.54/0.8163	29.68/0.82	29.67/0.8227	29.75/0.8252	29.87/0.8286	29.95/0.8311	**30.05/0.8329**
All	26.86/0.6987	27.83/0.7397	29.11/0.7899	29.22/0.7937	29.16/0.7911	29.28/0.7951	29.31/0.7955	29.41/0.7987	**29.52/0.8005**

In order to compare SAFFB with several other similar methods: ResNeXt [39], SENet [30], SKNet [32], and ResNeSt [40], SAFFB was replaced using these excellent residual modules and experiments were performed based on the UCMerced_LandUse dataset. In comparison to the other four methods, RASAF achieves optimality in PSNR and SSIM, as shown in Table III. The difference between RASAF and SKNet model parameters is due to the usage of group convolution, which reduces the number of parameters greatly. RASAF is also more time consuming than other methods. Overall, RASAF does not have an advantage in terms of time and number of parameters, but can achieve better accuracy gains.

C. Comparison With Other State-of-the-Art Methods

We compared RASAF with six state-of-the-art methods to validate its performance, which included the SRCNN [18], SRResNet [41], RDN [26], EDSR [42], RCAN [29], and DRN [17]. We trained and evaluated all the models under the same conditions using the UCMerced_LandUse dataset and the PatternNet dataset to validate their output according to the open-source code, and we used PSNR and SSIM as the evaluation metrics. In the testing section, we use a new strategy (RASAF+) where we flip and rotate each test image to obtain eight images and input them to the network in turn, invert the corresponding generated images, and then filter these eight images to obtain the final SR results. This strategy is similar to self-ensemble [42], but we do not generate new images by manipulating the generated images, which is more representative of the performance of the network.

Table IV shows the quantitative comparison results of the different models on the UCMerced_LandUse dataset for the scale $\times 4$ case. RASAF achieved the best results in every category of the scene. In the airplane scenes, it is at least 0.33 dB better than the other models, and at least 0.35 dB better than the overpass scenes. This shows that RASAF can adjust itself to different categories of scenes and establish a good LR–HR mapping relationship for each type of scenario. On the average evaluation of the last column, RASAF improves 0.2 dB on PSNR and 0.05 on SSIM over the other best models, which indicates that RASAF recovers better structural information.

Fig. 5 shows the results of the qualitative comparison. It can be seen that for the image airplane_81, the tail of the aircraft generated by other models is blurred, while RASAF can recover the details of the image without blurring. As for the image tenniscourt_71, due to the poor quality of the image itself, there is a noise effect, except for DRN and RASAF, other models are unable to recover the lines on the court. Compared with DRN, RASAF can go further to recover the net on the court more completely, showing a better reconstruction quality. This is due to our hierarchical loss function, which can effectively use multiple-scale image information and better accomplish the reconstruction of small features.

Table V shows the quantitative comparison results of all models on the PatternNet dataset scale $\times 4$ case, where RASAF still achieved the best evaluation results. This includes an improvement of at least 0.09 dB in PSNR and 0.013 dB in SSIM. This indicated that RASAF has good generalization ability and can achieve reconstruction results that exceed other models for different data.

The visual effect comparison is shown in Fig. 6. It can be seen that all the models except SRCNN can recover the zebra lines on runway_479, and the images generated by RASAF are the clearest among all the methods. For the shippingyard_655 image, RASAF generated a clearer centerline of the road. Compared with other methods, the shadow areas generated by RASAF are more compliant with HR, the boundaries are clearer, the structural information in the image can be well reconstructed, and the recovery of detail information is more prominent.

D. Comparison With Other Remote Sensing SISR

The abovementioned comparisons are all algorithms oriented to simple images. In order to better demonstrate the advantages of RASAF on remote sensing images, we choose three classical algorithms: LGCNet [14], MHAN [36], and SMSR [35] for comparison. We experiment on the UCMerced_LandUse dataset

Fig. 5. Visualization comparison of various algorithms in the case of the UCMerced_LandUse with scale factor ×4.

TABLE V
QUANTITATIVE COMPARISON OF VARIOUS METHODS USING PATTERNNET DATASET WITH SCALE FACTOR ×4

	Bicubic	SRCNN	SRResnet	RDN	EDSR	RCAN	DRN	RASAF	RASAF+
PSNR	26.46	26.99	27.88	27.99	28.01	28.07	28.05	28.08	**28.16**
SSIM	0.6610	0.6887	0.7254	0.7301	0.7306	0.7332	0.7317	0.7332	**0.7345**

TABLE VI
COMPARISON OF REMOTE SENSING SR ALGORITHMS BY USING THE UCMERCED_LANDUSE ×4

	Bicubic	LGCNet	MHAN	SMSR	Ours
PSNR	26.86	28.37	28.81	29.25	**29.41**
SSIM	0.6987	0.7592	0.7777	0.7918	**0.7987**

for the ×4. Table VI shows the results of the quantitative comparison, which reveal that RASAF has the best PSNR and SSIM, with a 0.16-dB improvement above the second best SMSR. The results of the qualitative comparison are shown in Fig. 7, where RASAF is the best for the reconstruction of road gaps and can also be reconstructed more clearly for the smaller gaps, while SMSR, the second best, does not perform well in the location of the smaller gaps. For storagetanks_54, RASAF also obtained a better visual performance. It is demonstrated that RASAF can solve the remote sensing image SR problem rather efficiently by comparing the above to the previous leading remote sensing SR methods.

E. Model Complexity Analysis

Fig. 8 shows the comparison of various methods in terms of model parameters, time, and performance. RASAF adds more in time, only less than RCAN, and does not have an advantage in runtime. In comparison to the three models, RCAN, RDN, and EDSR, the performance of RASAF was better with fewer parameters. RASAF has a few more parameters than DRN, but it achieved a huge improvement in PSNR. RASAF strikes a better balance between performance and the size of the model.

F. Multilabel Remote Sensing Image Classification Performance

To validate the impact of SR algorithms on multiclassification tasks of remote sensing images. We used ResNet-50 [16] as an evaluation model with the same configuration of training and testing data as RASAF. The original test image (256 × 256 pixels) was downsampled to 64 × 64 pixels using bicubic interpolation, and then various SR methods (RCAN, DRN, etc.) were used to recover it to the original target size and feed it into the qualified ResNet-50 for evaluation. As an evaluation metric, we used the average overall precision (OP), and we also evaluated the outcomes of the top three labels for a fair comparison. Table VII shows the final results. RASAF has the highest OP and is in the Top-3 OP, proving its outstanding feature representation capacity. We analyzed the classification accuracy of five of these classes and discovered that the classification network was able

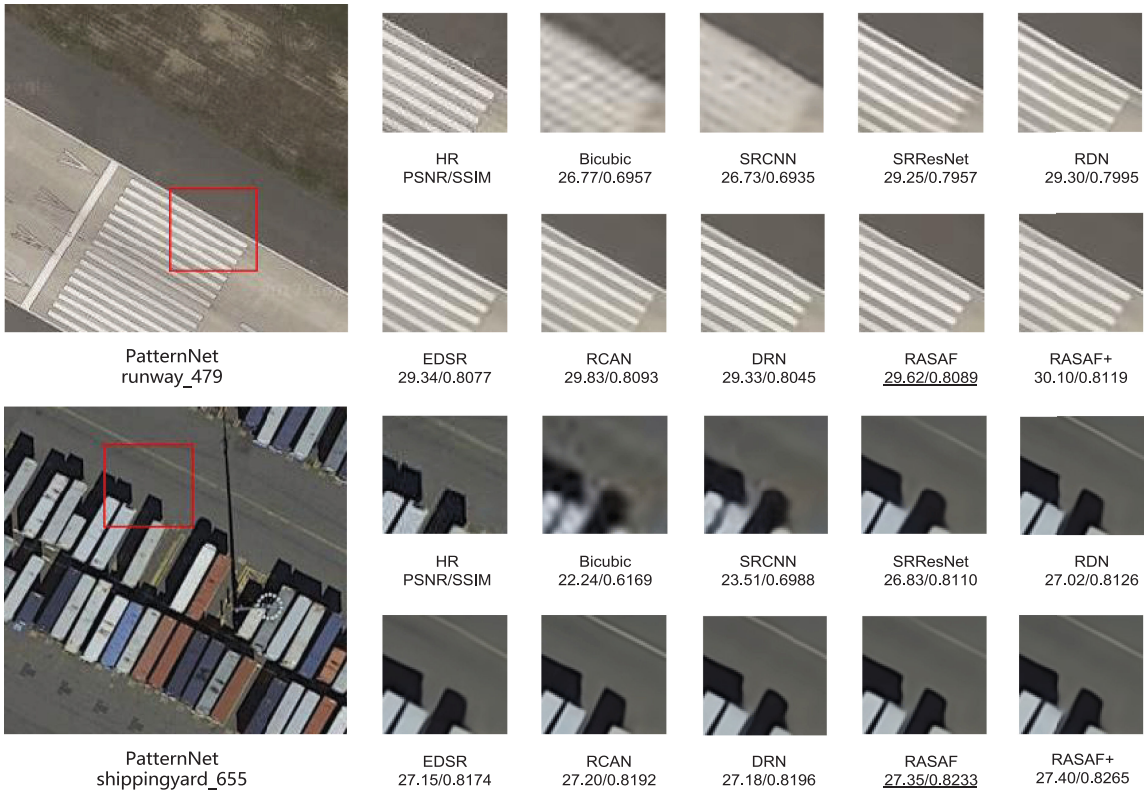

Fig. 6. Visualization comparison of various algorithms using PatternNet with scale factor × 4.

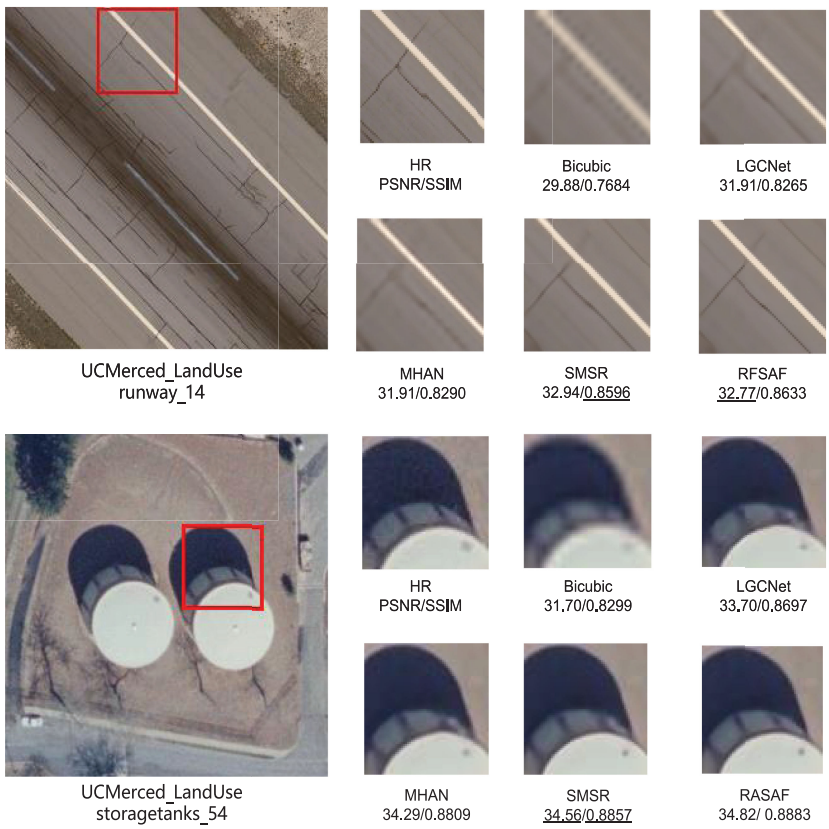

Fig. 7. Visualization comparison of remote sensing SISR algorithms in the case of the UCMerced_LandUse with scale factor ×4.

TABLE VII
MULTILABEL REMOTE SENSING IMAGE CLASSIFICATION PERFORMANCE

Evaluation	Bicubic	SRCNN	SRResnet	RDN	EDSR	RCAN	DRN	Our	Our+	Baseline
Top1	0.8719	0.8757	0.881	0.8687	0.8789	0.8793	0.8725	0.8808	**0.8818**	0.8994
Top3	0.8805	0.8763	0.8889	0.9024	0.8927	0.9028	0.9053	0.9088	**0.9123**	0.9567
AP(airplane)	1	1	1	1	1	1	1	1	1	1
AP(cars)	0.9121	0.9139	0.9334	0.9335	0.9384	0.9383	**0.9418**	0.9334	0.9417	0.9427
AP(grass)	0.9517	0.9381	0.9429	0.9503	0.9472	0.9464	0.9511	0.9446	**0.9527**	0.9639
AP(mobile-home)	0.9429	1	0.9667	1	1	0.9429	1	1	**1**	1
AP(trees)	0.9721	0.9704	0.9831	0.9856	0.985	0.9865	0.985	**0.9879**	0.9867	0.9901

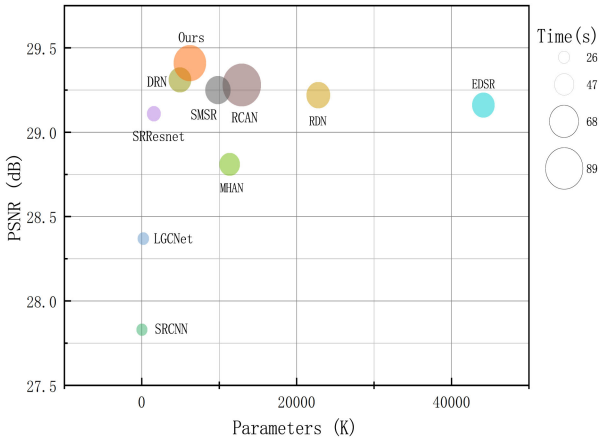

Fig. 8. Performance, time, and number of parameters. Results are evaluated on UCMerced_LandUse with scale factor ×4.

to accurately identify huge items such as "airplane" without much correlation with the SISR. For some smaller objects, such as "mobile-home," it starts to be affected by SISR, and the generated results in individual methods such as RCAN perform badly. The SISR improvement is higher for smaller objects, such as "cars," with a difference of 2.96% between RASAF+ and Bicubic.

V. CONCLUSION

We proposed a RASAF for SR of remote sensing images with varying scenes and spatial complexity. The SAFB enhances cross-channel interaction between feature groups. It achieved better feature representation capability than spatial or channel attention blocks as illustrated by the experiments. In the upsampling stage, the RASAF model can produce multiscale images and create hierarchical loss functions from them. It allows the use of multiscale image information without adding additional parameters. Furthermore, we used an improved RFAB by adding a few parameters. It aggregates local residual features to generate high-quality local features. It alleviates feature loss caused by interference in the backward transfer phase, such as convolution and summation of low-level residual features. The proposed method produces a lightweight and high-performance SISR model. The experiments on the UCMerced_LandUse and PatternNet datasets showed a better performance of the proposed blocks and model. The RASAF is a strong candidate for enhancing high-level vision tasks as it supports multilabel remote sensing image classification tasks.

REFERENCES

[1] C. E. Duchon, "Lanczos filtering in one and two dimensions," *J. Appl. Meteorol. Climatol.*, vol. 18, no. 8, pp. 1016–1022, 1979.
[2] L. Zhang and X. Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2226–2238, Aug. 2006.
[3] K. Zhang, X. Gao, D. Tao, and X. Li, "Single image super-resolution with non-local means and steering kernel regression," *IEEE Trans. Image Process.*, vol. 21, no. 11, pp. 4544–4556, Nov. 2012.
[4] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Comput. Graph. Appl.*, vol. 22, no. 2, pp. 56–65, Mar./Apr. 2002.
[5] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low-level vision," *Int. J. Comput. Vis.*, vol. 40, no. 1, pp. 25–47, 2000.
[6] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2004, pp. 1–1.
[7] C.-Y. Yang and M.-H. Yang, "Fast direct super-resolution by simple functions," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 561–568.
[8] K. S. Ni and T. Q. Nguyen, "Image superresolution using support vector regression," *IEEE Trans. Image Process.*, vol. 16, no. 6, pp. 1596–1610, Jun. 2007.
[9] F. Nan, Q. Zeng, Y. Xing, and Y. Qian, "Single image super-resolution reconstruction based on the ResNeXt network," *Multimedia Tools Appl.*, vol. 79, no. 45, pp. 34459–34470, 2020.
[10] S. Lei, Z. Shi, and Z. Zou, "Coupled adversarial training for remote sensing image super-resolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3633–3643, May 2020.
[11] W. Shi *et al.*, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1874–1883.
[12] N. Zhang *et al.*, "A multi-degradation aided method for unsupervised remote sensing image super resolution with convolution neural networks," *IEEE Trans. Geosci. Remote Sens.*, early access, Dec. 21, 2020, doi: 10.1109/TGRS.2020.3042460.
[13] W. Ma, Z. Pan, J. Guo, and B. Lei, "Achieving super-resolution remote sensing images via the wavelet transform combined with the recursive ResNet," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3512–3527, Jun. 2019.
[14] S. Lei, Z. Shi, and Z. Zou, "Super-resolution for remote sensing images via local-global combined network," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 8, pp. 1243–1247, Aug. 2017.
[15] A. Ducournau and R. Fablet, "Deep learning for ocean remote sensing: An application of convolutional neural networks for super-resolution on satellite-derived SST data," in *Proc. IEEE 9th IAPR Workshop Pattern Recognit. Remote Sens.*, 2016, pp. 1–6.
[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[17] Y. Guo *et al.*, "Closed-loop matters: Dual regression networks for single image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5407–5416.
[18] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 184–199.
[19] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 391–407.
[20] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep Laplacian pyramid networks for fast and accurate super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 624–632.

[21] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1664–1673.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2015, pp. 234–241.

[23] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1646–1654.

[24] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4799–4807.

[25] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.

[26] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2472–2481.

[27] X. Wang *et al.*, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2018, pp. 63–79.

[28] J. Liu, W. Zhang, Y. Tang, J. Tang, and G. Wu, "Residual feature aggregation network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2359–2368.

[29] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 286–301.

[30] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.

[31] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.

[32] X. Li, W. Wang, X. Hu, and J. Yang, "Selective Kernel networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 510–519.

[33] Y. Dai, F. Gieseke, S. Oehmcke, Y. Wu, and K. Barnard, "Attentional feature fusion," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2021, pp. 3560–3569.

[34] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[35] X. Dong, L. Wang, X. Sun, X. Jia, L. Gao, and B. Zhang, "Remote sensing image super-resolution using second-order multi-scale networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 4, pp. 3473–3485, Apr. 2021.

[36] D. Zhang, J. Shao, X. Li, and H. T. Shen, "Remote sensing image super-resolution via mixed high-order attention network," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 5183–5196, Jun. 2021.

[37] Z. Pan, W. Ma, J. Guo, and B. Lei, "Super-resolution of single remote sensing image based on residual dense backprojection networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7918–7933, Oct. 2019.

[38] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[39] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1492–1500.

[40] H. Zhang *et al.*, "ResNeSt: Split-attention networks," 2020, *arXiv:2004.08955*.

[41] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4681–4690.

[42] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 136–144.

Long Chen received the bachelor's degree in geographic information science from the Shandong University of Science and Technology, Qingdao, China, in 2018. He is currently working toward the master's degree in software engineering with Xinjiang University, Ürümqi, China.

His research interests include deep learning and single image super-resolution.

Hui Liu received the B.S. degree in software engineering in 2014, the M.E. degree in software engineering from the College of Software, Xinjiang University, Ürümqi, China, in 2017, where she is currently working toward the Ph.D. degree in computer science and technology.

Her research interests include deep learning and opportunistic networks and the processing of remote sensing image data.

Minhang Yang received the bachelor's degree in software engineering from the Xi'an University of Technology, Xi'an, China, in 2019. She is currently working toward the master's degree in software engineering with Xinjiang University, Ürümqi, China.

Her research interests include deep learning and multilabel image classification.

Yurong Qian received bachelor's and master's degrees in computer science and technology from Xinjiang University, Ürümqi, China, in 2000, and the doctorate degree in biology from Nanjing University, Nanjing, China, in 2010.

From 2012 to 2013, she was a Postdoctoral Fellow with the Department of Electronics and Computer Engineering, Hanyang University, South Korea. She is currently a Professor with the School of Software, Xinjiang University. Her research interests include computational intelligence such as big data processing, image processing, and artificial neural networks.

Dr. Qian is a Senior Member of the Chinese Computer Federation. In 2015, she was trained as a Young Scientific and Technological Innovation Talent by the Science and Technology Department, Xinjiang, China.

Zhengqing Xiao received the Ph.D. degree in geographic information system from Beijing Normal University, Beijing, China, in 2011.

He is currently working with the College of Mathematics and System Sciences, Xinjiang University, Ürümqi, China. His research interests include big data analysis, image processing, and complex system modelling.

Xiwu Zhong received the bachelor's degree in network engineering from Huizhou University, Huizhou, China, in 2018. He is currently working toward the master's degree, majoring in software engineering with Xinjiang University, Ürümqi, China.

His research interests include deep learning and remote sensing image processing.